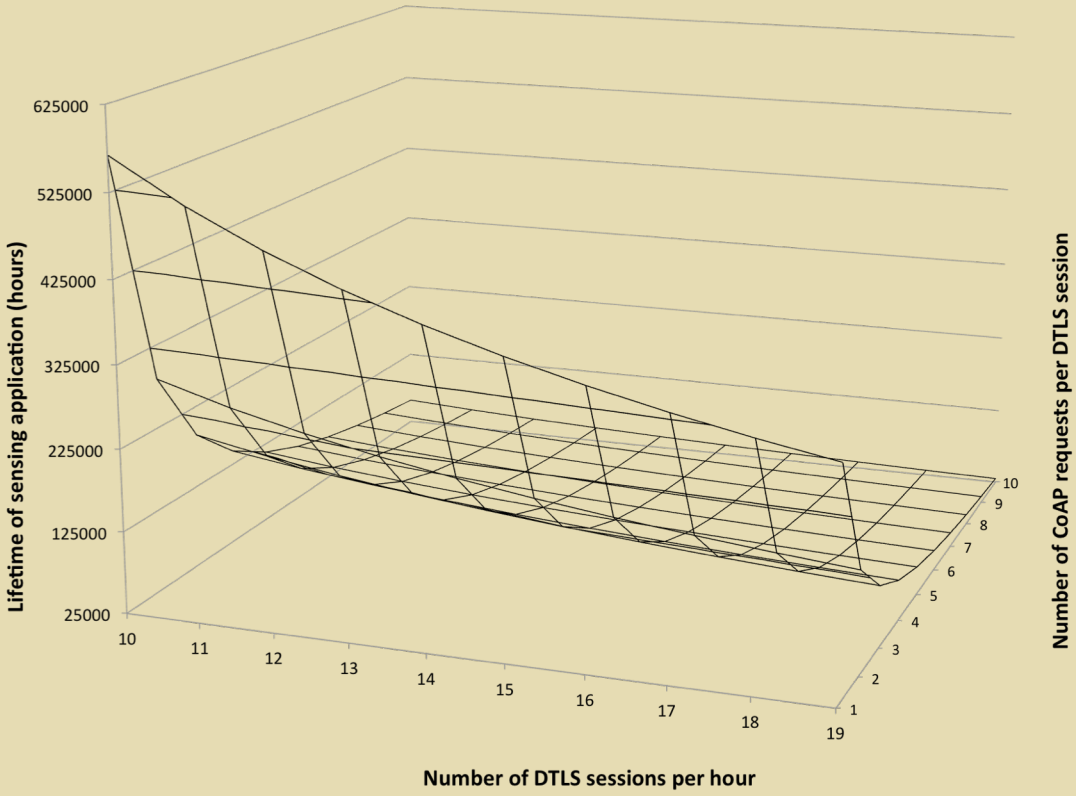

António Jorge da Costa Granjal

END-TO-END SECURITY SOLUTIONS FOR INTERNET-INTEGRATED WIRELESS SENSOR NETWORKS

Tese de Doutoramento em Programa Doutoral em Ciências e Tecnologias da Informação, orientada pelo Doutor Edmundo Monteiro e pelo Doutor Jorge Sá Silva, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Dezembro de 2014

· U C ·

UNIVERSIDADE DE COIMBRA

UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

END-TO-END SECURITY SOLUTIONS FOR INTERNET-INTEGRATED WIRELESS SENSOR NETWORKS

António Jorge da Costa Granjal

Tese submetida à Universidade de Coimbra para obtenção do grau de Doutor em
Programa Doutoral em Ciências e Tecnologias da Informação

COIMBRA
dezembro de 2014

Tese realizada sob orientação do

Doutor Edmundo Heitor da Silva Monteiro

Professor Catedrático do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

e do

Doutor Jorge Sá Silva

Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

This investigation was partially supported by the Portuguese Research Agency FCT through scholarship number BD/48861/2008, and by the research projects GINSENG (FP7/2007-2013, under agreement 224282), iCIS (CENTRO-07-ST24-FEDER-002003) and COST (European Cooperation in Science and Technology) Action WiNeMo (Wireless Networking for Moving Objects, IC0906).

ACKNOWLEDGMENTS

I would like to thank Professors Edmundo Monteiro and Jorge Sá Silva for their valuable guidance and advices during the last years. Professor Edmundo is also a friend since my early days at the University of Coimbra, whom I thank for the support in this important step of my life. To my family and friends, I really appreciate your support throughout the last years. To Eunice, my love and soul mate, thank you for your encouragement and for our journey together.

FOREWORD

The research efforts and results discussed throughout this thesis were supported by the research projects GINSENG (Performance Control in Wireless Sensor Networks, FP7/2007-2013 under agreement 224282) and iCIS (Intelligent Computing in the Internet of Services, CENTRO-07-ST24-FEDER-002003), as well as by the COST (European Cooperation in Science and Technology) Action WiNeMo (Wireless Networking for Moving Objects, IC0906). The goals of these research projects, as well as our contribution, are discussed next, while later we present the peer-reviewed publications resulting from our research efforts.

Participation in Research Projects

The goal of GINSENG was the design of solutions to enable the employment of wireless sensor networks in performance-critical communication environments, particularly in the context of industrial sensing and control applications. In this project we contributed to the design of the architecture, as well as with the identification of its main requirements in terms of security. Our contribution to this project is discussed in Chapter 3 and also provided the ground for the design of the reference model for end-to-end security considered throughout the thesis.

We must also note that the research proposals described throughout the thesis were not developed nor evaluated in the context of this project, since due to time constraints security was subsequently left out of the project, as other tasks required the allocation of more resources. In this context, our research efforts evolved to the consideration of the usage of communication technologies currently being designed to enable Internet communications on WSN environments. We must also observe that the reference model for end-to-end security was from the start designed to consider the usage of such technologies.

The iCIS project is currently ongoing and its goal is to support research efforts in intelligent computing for the Internet of Things and Services, as well as the formation of research and industry consortiums targeting national and international research framework programmes. The research efforts discussed in the thesis were also supported by the COST WiNeMo Action, which supports research in the area of wireless networking for moving objects, with the goal of advancing the state-of-the-art concerning networking aspects of scenarios integrating sensing objects into the IoT.

The research proposals discussed throughout the thesis resulted in various peer-reviewed publications in the literature. In the following list we identify peer-reviewed publications in international journals, international conferences, and as book chapters and technical reports. With each publication we also identify the number of citations in the literature at the time of writing of the thesis.

Publications in International Journals

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, “A survey on Security Mechanisms for the Internet of Things.” *IEEE Surveys & Tutorials* (2014) (status: pending, revised version submitted).

Jorge Granjal, Edmundo Monteiro and Jorge Sá Silva, “A survey on the secure integration of low-power Wireless Sensor Networks with the Internet.” *Elsevier Ad Hoc Networks* (2015).

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, "Network-layer security for the Internet of Things using TinyOS and BLIP." *International Journal of Communication Systems*, 2013 (*Cited by 4 publications*).

Publications in International Conferences

Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva and Fernando Boavida, "Why is IPSec a viable option for wireless sensor networks." *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*. IEEE, 2008 (*Cited by 24 publications*).

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva. "Enabling network-layer security on IPv6 wireless sensor networks." *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010 (*Cited by 15 publications*).

Vasco Pereira, Jorge Sá Silva, Jorge Granjal, Ricardo Silva, Edmundo Monteiro and Qiang Pan. "A taxonomy of wireless sensor networks with QoS." *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*. IEEE, 2011 (*Cited by 7 publications*).

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, "A secure interconnection model for IPv6 enabled wireless sensor networks." *Wireless Days (WD), 2010 IFIP*. IEEE, 2010 (*Cited by 6 publications*).

Jorge Granjal, Edmundo Monteiro and Jorge Sá Silva, “On the effectiveness of end-to-end security for Internet-integrated sensing applications.” *The IEEE International Conference on Internet of Things 2012 (iThings 2012)* (*Best Paper Award, Cited by 4 publications*).

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, "On the feasibility of secure application-layer communications on the Web of Things." *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*. IEEE, 2012 (*Cited by 5 publications*).

Jorge Granjal, Edmundo Monteiro and Jorge Sá Silva, “End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication”, *IFIP Networking 2013* (*Cited by 3 publication*).

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, "Application-Layer Security for the WoT: Extending CoAP to Support End-to-End Message Security for Internet-Integrated Sensing Applications." *Wired/Wireless Internet Communication*. Springer Berlin Heidelberg, 2013. 140-153.

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, "A framework towards adaptable and delegated end-to-end transport-layer security for Internet-integrated Wireless Sensor Networks." *2nd Joint ERCIM eMobility and MobiSense Workshop*.

André Riker, Jorge Granjal, Marília Curado and Edmundo Monteiro, "Middleware Group Communication Mechanisms in M2M environments." *2nd Joint ERCIM eMobility and MobiSense Workshop*.

Publications as Book Chapters

Iva Bojic, Jorge Granjal, Edmundo Monteiro, Damjan Katusic, Pavle Skocir, Mario Kusek and Gordan Jezic, "Communication and Security in Machine-to-Machine Systems", Wireless Networking for Moving Objects Book, LNCS Springer, 2014.

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva, "Security Issues and Approaches on Wireless M2M Systems." *Wireless Networks and Security*, Springer Berlin Heidelberg, 2013. 133-164.

Publications as Technical Reports

Jorge Granjal and Edmundo Monteiro, "WiNeMO Security Provision - Security in M2M environments", June 2012.

RESUMO EM PORTUGUÊS

A investigação em soluções tecnológicas para as Redes de Sensores Sem Fios (RSSF) despertou grande interesse e inúmeros esforços ao nível da investigação em anos recentes. O objetivo inicial de tais redes foi o de providenciar uma base tecnológica que permitisse dispor de aplicações sensoriais distribuídas, desenhadas com propósitos bem específicos nas mais diversas áreas de investigação e aplicação. Uma característica distintiva das RSSF é a utilização de dispositivos com capacidade para comunicar por radiofrequência e de "sentir" e "atuar" no meio físico que os rodeia. Tal capacidade permite, na prática, o desenvolvimento e a utilização de soluções verdadeiramente inovadoras implementadas com recurso a aplicações distribuídas capazes de interagir com o mundo físico.

Outra característica fundamental das RSSF é a utilização de dispositivos sensoriais desenhados para utilização em grande número, razão pela qual tais dispositivos são projetados para serem pouco dispendiosos, um requisito que se traduz por sua vez em restrições ao nível de diversas características e recursos fundamentais. Para além do seu baixo custo, os Sensores Sem Fios (SFF) são projetados para disporem da capacidade de comunicar por radiofrequência e serem autónomos ao nível energético. Tais características são essenciais para permitir a utilização dos SSF em áreas remotas e o suporte de aplicações com tempos de vida elevados. Dado o seu baixo custo, os SSF normalmente utilizam baterias com capacidade energética limitada. Dadas as características e restrições dos SSF, bem como a sua utilização em aplicações com tempos de vida alargados, os mecanismos projetados e utilizados nas RSSF são normalmente muito optimizados no que diz respeito à sua utilização de recursos computacionais e energéticos ao seu dispor nas plataformas sensoriais.

As aplicações originais das RSSF visavam essencialmente a construção de soluções eficientes para problemas bem delimitados e, como consequência, tais redes não eram projetadas com o objetivo de suportar diferentes tipos de aplicações ou mecanismos de comunicação adaptáveis a diferentes propósitos de utilização. Podemos igualmente verificar que os mecanismos de comunicação e segurança utilizados em tais aplicações eram desenhados de acordo com o seu propósito específico de utilização. Por conseguinte, a heterogeneidade ao nível das aplicações suportadas e dos seus requisitos não foi considerado um aspecto prioritário no desenho das soluções clássicas de comunicação e segurança em ambientes de RSSF. Tais soluções consideram portanto as RSSF como sendo vocacionadas essencialmente para o suporte de aplicações especializadas e isoladas, sem a capacidade de suportar aplicações e equipamentos heterogéneos, bem como comunicações com dispositivos externos ao seu ambiente de comunicações. Tais características são fundamentais na evolução e no sucesso da infraestrutura de comunicações e segurança da Internet, e motivam atualmente uma mudança de paradigma ao nível da utilização das RSSF, um aspeto que motiva as abordagens de investigação descritas na presente tese.

No contexto da nossa discussão anterior, é atualmente possível verificar que os objetivos de investigação das RSSF têm vindo a evoluir em anos recentes para a consideração da necessidade da sua integração com a Internet. De facto, a visão tradicional das RSSF enquanto ambientes de comunicações isolados tem vindo a sofrer alterações, no sentido da aceitação da necessidade e das vantagens inerentes à sua integração com ambientes externos de comunicação e, em particular, com a infraestrutura de comunicações da Internet. Esta evolução está igualmente relacionada com a atual visão da Internet do futuro nas suas mais diversas formas, no contexto da qual se espera que os dispositivos sensoriais e de atuação possam fazer parte de forma natural e transparente da arquitetura global de comunicações. Esta visão é atualmente materializada em conceitos tais como a Internet dos Objetos (IoT, *Internet of Things*) e a Web dos Objetos (WoT, *Web of Things*), ou em padrões de comunicação reconhecidos como fundamentais para materializar esta visão, tal como as comunicações máquina-a-máquina (M2M, *Machine to Machine*). A integração das RSSF com a Internet afigura-se portanto como necessária e vantajosa, suscitando atualmente um interesse crescente por parte da comunidade científica.

Tal como teremos a oportunidade de analisar de forma aprofundada no Capítulo 2, a integração das RSSF com a Internet pode na prática ser conseguida mediante diferentes técnicas ou estratégias, algumas das quais traduzem mesmo soluções já disponíveis comercialmente. Tais estratégias envolvem por exemplo a utilização de dispositivos intermediários especializados, e em alguns casos a utilização de serviços de comunicação e computação baseados em soluções *Cloud*. Estas soluções apresentam a vantagem de permitir construir, de forma relativamente simples, aplicações complexas baseadas na obtenção e processamento de informação sensorial proveniente de dispositivos sensoriais em RSSF. Por outro lado, a utilização nas RSSF de soluções universais de comunicação e segurança permitirá integrá-las com a arquitetura da Internet, bem como dispor de comunicações mais diretas entre dispositivos sensoriais e entidades externas aos ambientes das RSSF. Tal como teremos a oportunidade de discutir, este é o contexto de integração que motiva as propostas descritas ao longo da presente tese.

A viabilidade da maioria das aplicações dos ambientes RSSF, quer isolados ou integrados com a Internet, dependerá fortemente da utilização de mecanismos adequados de segurança. Embora a segurança seja fundamental no contexto das várias estratégias de integração analisadas na presente tese, pode ser considerada particularmente relevante no contexto da exposição das RSSF à infraestrutura de comunicações global da Internet. De facto, para além das ameaças de segurança inerentes à utilização de comunicações sem fios e às restrições dos próprios dispositivos sensoriais, a exposição das RSSF à infraestrutura de comunicações Internet, ainda que limitada e controlada, motivará necessariamente riscos e ameaças acrescidas que importa prevenir e combater através da adopção de mecanismos de segurança apropriados.

Os trabalhos de investigação apresentados na presente tese abordam a problemática da segurança no contexto da integração das RSSF com a Internet, em particular no que diz

respeito à proteção das comunicações fim-a-fim no contexto da integração das RSSF com a infraestrutura de comunicações e segurança global. Esta estratégia de integração está presentemente a ganhar protagonismo através da adopção de mecanismos Internet de comunicações optimizados para ambientes RSSF. Um objetivo central das propostas de investigação discutidas na presente tese é precisamente o de contribuir para os ambientes de RSSF que utilizam tais mecanismos de comunicação, em particular no que concerne à proteção de comunicações fim-a-fim entre dispositivos SSF e sistemas externos às RSSF.

Os mecanismos de segurança propostos ao longo da presente tese abordam várias técnicas e focam-se em diferentes níveis protocolares, para a obtenção de segurança fim-a-fim com RSSF integradas com a Internet. Estes mecanismos procuram responder, em particular, à questão da viabilidade de obtenção, de forma eficiente, de segurança efetiva no contexto da utilização de comunicações fim-a-fim Internet entre SSF e outros sistemas externos à RSSF ou na Internet. Tais mecanismos de segurança e comunicações poderão contribuir de forma decisiva para a viabilidade de aplicações sensoriais distribuídas que dependam ou beneficiem de comunicações Internet diretas entre sistemas Internet e plataformas sensoriais com restrições ao nível de recursos tais como os SSF. De notar que, apesar do foco particular da presente tese nas soluções de segurança para comunicações fim-a-fim, a integração das RSSF com a Internet motivará igualmente a utilização de outros tipos de mecanismos desenhados para fazer face a requisitos de segurança muitas vezes transversais aos vários protocolos de comunicação. Tais mecanismos podem garantir funcionalidades de segurança tais como o controlo de acessos, a gestão de chaves ou a detecção de intrusões, entre outros, ou suportar mecanismos para garantia de aspectos de segurança tal como a privacidade e a confiança.

Os mecanismos de segurança descritos na presente tese são desenvolvidos e avaliados no contexto de um modelo de referencia para a integração de RSSF com a Internet, que teremos a oportunidade de expor no Capítulo 3. Não traduzindo uma concepção definitiva sobre como esta integração irá ser assegurada no futuro, este modelo reflete na prática a estratégia de integração atual alicerçada na utilização de tecnologias Internet de comunicação e segurança existentes ou atualmente em desenvolvimento para as RSSF. Estas tecnologias são na realidade bastante recentes, nalguns casos encontrando-se ainda na sua fase de desenvolvimento. A utilização de tais soluções tecnológicas no contexto da integração das RSSF com a arquitetura de comunicações da Internet motiva a nossa proposta de soluções de segurança e a metodologia de avaliação experimental considerada ao longo da presente tese.

No contexto da investigação desenvolvida e descrita na presente tese, a eficácia das soluções propostas reporta necessariamente à sua capacidade em não comprometerem o tempo de vida das aplicações sensoriais, ao mesmo tempo garantindo níveis aceitáveis de segurança. Um aspeto essencial é portanto a sua capacidade em cumprir requisitos predefinidos de segurança e funcionais, avaliados de acordo com métricas e critérios objetivos. Por outro lado, tais soluções devem poder suportar aplicações com requisitos de

segurança e funcionais diversos, contribuindo para o suporte de aplicações heterogéneas, um aspeto que distingue claramente as propostas de investigação descritas na presente tese das abordagens tradicionais à segurança em ambientes de RSSF.

Os mecanismos desenhados e avaliados ao longo da presente tese consideram diferentes aproximações ao problema da obtenção de segurança fim-a-fim em RSSF integradas com a Internet. Estes mecanismos são avaliados de acordo com a sua capacidade de cumprirem requisitos de segurança predefinidos, ao mesmo tempo fazendo uma utilização eficiente dos recursos críticos e limitados ao seu dispor nas RSSF. O trabalho de investigação descrito na presente tese refere-se ao desenho de mecanismos para obtenção de segurança fim-a-fim ao nível das camadas protocolares de rede, transporte e aplicação. À semelhança da arquitetura atual de comunicações da Internet, a utilização de mecanismos complementares de comunicações e segurança também promove o suporte efetivo de aplicações e cenários de utilização com diferentes características e requisitos.

Ao nível da camada de rede, os mecanismos propostos e avaliados refletem a filosofia da arquitetura atual de segurança da Internet, no contexto da qual cabeçalhos de segurança são adicionados ao protocolo IP com o intuito de suportar segurança na camada de rede de forma completamente transparente às comunicações em níveis protocolares superiores. Em particular, são propostos e avaliados cabeçalhos de segurança que permitem garantir autenticação, integridade, não-repúdio e confidencialidade em comunicações ao nível da camada de rede em RSSF integradas com a Internet. No desenho destas soluções considerou-se igualmente o interesse da sua integração futura com a arquitetura de segurança da Internet.

No que diz respeito à camada de transporte, os mecanismos propostos abordam a utilização de técnicas de delegação de operações de segurança dispendiosas dos SSF em dispositivos com mais recursos. Em particular, são propostas e avaliadas técnicas que implementam a mediação transparente e a delegação da autenticação por chave pública no contexto da fase inicial de autenticação e negociação de chaves em protocolos de segurança fim-a-fim na camada de transporte. Estas propostas abordam igualmente outras vantagens inerentes à mediação transparente da segurança fim-a-fim em RSSF integradas com a Internet, por exemplo ao nível da detecção atempada de ataques ao nível protocolar.

De forma complementar às propostas anteriores, ao nível da camada de aplicação a presente tese aborda a integração da segurança no contexto do próprio protocolo de comunicação. A abordagem considerada a este nível apresenta as vantagens de suportar políticas de segurança flexíveis e mais granulares, a utilização de mecanismos de segurança diferentes dependentes da semântica do protocolo de aplicação ou do conteúdo das comunicações, ou o suporte de múltiplos domínios de segurança e métodos de autenticação.

Tal como observado anteriormente, a complementaridade dos mecanismos de segurança propostos e avaliados ao longo da presente tese traduz-se, não apenas ao nível protocolar,

mas igualmente ao nível da forma como abordam a segurança fim-a-fim. Ao nível protocolar, a existência de mecanismos de segurança a diversos níveis pode contribuir para a integração segura das RSSF com a Internet, em particular através da adopção de tais mecanismos no contexto da arquitetura de segurança da Internet. A utilização de diversas abordagens no suporte de segurança fim-a-fim pode, por sua vez, contribuir para o suporte de RSSF com diversas características e tipos de dispositivos, bem como de aplicações com diferentes requisitos e graus de exposição às comunicações Internet. Diferentes mecanismos de segurança podem, de forma complementar, adaptar-se aos requisitos funcionais e de segurança de diferentes aplicações e cenários de utilização. Esta flexibilidade é uma característica fundamental da arquitetura atual de segurança da Internet, e poderá igualmente contribuir para a sua evolução no sentido da inclusão das RSSF que utilizam protocolos de comunicação Internet.

ABSTRACT

The area of Wireless Sensor Networks (WSN) has motivated great interest and numerous research efforts in the recent years. The initial purpose of these networks was to provide a technological basis on top of which new distributed sensorial applications can be built. One main distinctive characteristic of WSN is the employment of sensing devices that have the capability of communicating wirelessly, and also of "feeling" and "actuating" with the physical world. Such capabilities enable the development of truly innovative solutions, based on applications that are designed to benefit from or require interactions with the physical world. Most traditional WSN approaches target particular research goals and applications with very focused purposes, rather than the support of heterogeneous applications and devices as in traditional Internet communication environments.

Another important characteristic of WSN is the employment of constrained wireless sensing platforms. The constraints of such platforms are mostly due to cost restrictions, given that such devices are designed to support cost-effective applications that may require the employment of large amounts of devices in potentially large geographical areas. These cost restrictions usually motivate that most sensing platforms are constrained in term of critical resources such as memory, energy and computational capabilities. Sensing devices usually also run on batteries, since WSN applications frequently target remote and unattended deployment environments without continual energy sources. In conclusion, we may observe that the constraints and characteristics of WSN devices and applications determine that communications and security mechanisms be designed to be very optimized and to use the limited available resources very frugally.

As previously observed, the initial applications designed for WSN targeted very particular goals and application areas. Due to the characteristics and constraints of WSN sensing devices, the communication and security technologies designed for such applications were optimized according to the particular requirements of the application at hand, rather than to support heterogeneous applications and devices, as is the traditional Internet communications environment. In the same context, communications with external networks or with the Internet was also not an issue. As research in WSN evolves, we currently observe that this perception is changing, and that the advantages of integrating WSN with the Internet are currently being realized and motivating further research efforts.

The integration of WSN with the Internet can potentially support transparent end-to-end communications involving constrained wireless sensing devices and other external or Internet hosts. The support of such communications may also contribute to materialize current visions of the Internet of the future, as the IoT (Internet of Things) or the WoT (Web of Things), in which communications with sensing devices of various types and possessing diverse capabilities are transparently supported as required for sensing applications.

As discussed in Chapter 2, the integration of WSN with the Internet may in practice be accomplished according to different strategies, some of which are materialized in existing research and commercial proposals. Many of such proposals employ proprietary intermediary systems (gateways) or cloud-based computational services. Despite the pragmatism and practicality of such approaches, we in general realize that they lack the support of pure end-to-end Internet communications enabling the full integration of WSN with the Internet communications infrastructure. This is due to the fact that in such approaches WSN are isolated from the global Internet communications, despite the WSN data and devices being reachable via interconnection gateways. As we will observe later, the full integration of WSN with the Internet at the protocol level provides various benefits and motivates the research solutions discussed throughout the thesis.

As in traditional WSN applications, security will be a fundamental enabling factor of future sensorial applications employing sensing devices integrated with the Internet communications infrastructure. This applies to all the existing integration approaches, and will constitute a particularly relevant and challenging aspect for the integration of WSN employing Internet communication technologies. In such WSN environments, security threats will be present not only because of aspects which are inherent of WSN environments, for example the employment of wireless communications and the constraints and physical exposure of sensing devices, but also because of the threats which may be present from the day we start exposing WSN communications to the Internet. If on the one side security mechanisms, such as traffic filtering or intrusion detection, may help in preventing such threats, on the other applications may require or benefit from the employment of true end-to-end communications involving constrained sensing devices. Security will thus be of paramount importance for the enabling of such applications.

In the present thesis we describe and evaluate research proposals designed to target the problem of security in the context of WSN integrated with the Internet using Internet communication technologies designed and optimized for such environments. These technologies enable end-to-end Internet communications between WSN devices, and also between WSN devices and external or Internet hosts, and provide the context for the research proposals described in the present thesis. We target different approaches in supporting end-to-end security in the context of Internet-integrated WSN, with the goal of investigating the viability of supporting end-to-end security with communication technologies developed for such environments, and providing complementary solutions to support heterogeneous applications and deployment environments. We must also note that, despite our particular focus on end-to-end communications and security, the full integration of WSN with the Internet will in fact require efforts towards the design of appropriate mechanisms targeting other important security aspects. Such mechanisms may possibly be designed in a cross-layer fashion and support fundamental security-related operations such as key management and intrusion detection, or the enforcement of security requirements such as privacy and trust, among others.

The security mechanisms described throughout the thesis are proposed and evaluated in the context of a reference model supporting the integration of WSN with the Internet at the protocol level, which we discuss in Chapter 3. Rather than providing a definitive conception of how this integration approach may be supported, this model supports a reference framework for the employment of the Internet communication technologies currently being designed with this purpose. These communication technologies provide the ground for the development of the security mechanisms proposed throughout the thesis. We evaluate such proposals experimentally, as we consider this approach to provide various benefits in comparison with its validation in simulation environments, as we discuss later.

The security proposals discussed in the thesis seek to investigate the viability of enabling security for end-to-end communications with sensing devices using Internet WSN communication technologies. For this purpose, we propose solutions to protect communications using technologies currently being designed without proper security mechanisms, and propose alternatives to existing security approaches that we find to be inappropriate or insufficient. The research solutions proposed and evaluated throughout the thesis also aims to support heterogeneous devices and applications, as security is addressed at different protocol layers and by implementing different approaches to the support of security-related procedures. The effectiveness of new security mechanisms may be measured according to their ability to not compromise the lifetime of sensing applications, in the light of the previously discussed characteristics and constraints of WSN applications and devices, which we may measure according to specific metrics and evaluation criteria. On the other hand, applications with different functional and security requirements must also be appropriately supported by the proposed mechanisms, in line with our goal of securing communications supporting heterogeneous applications and deployment environments.

The security solutions proposed and evaluated throughout the thesis are materialized in security mechanisms implementing different approaches to the problem of end-to-end security with Internet-integrated WSN. We evaluate the proposed solutions against its ability to cope with predefined security requirements, while at the same time being able to employ the limited resources available on constrained sensing platforms in an efficient and controlled manner. As with the current Internet architecture, the complementarity of the approaches considered for the design of such mechanisms may promote the support of applications and deployment scenarios with different characteristics and requirements in terms of security. As we discuss next, we target the usage of end-to-end security at the network, transport and application layers.

At the network layer, the proposed solutions in practice inherit some of the characteristics of the current approach of the Internet security architecture to network-layer security. In particular, we consider the employment of security headers employed side-by-side with the headers of the network layer, with the goal of supporting end-to-end security in a transparent fashion to communication protocols and applications at upper layers of the

communications stack. The design of the proposed security headers also considers its future adoption in the Internet security architecture, as we discuss later in the thesis.

In what respects the transport layer, we consider the employment of delegation techniques to offload costly security-related computations from constrained sensing platforms to more powerful network entities. In particular, such entities may support public-key cryptography in the context of the authentication and key agreement phase, which is particularly costly for the support of transport-layer security with Internet-integrated WSN as currently proposed. The proposed solutions to address security at the transport-layer are also able to guarantee total transparency from the point of view of the two ends of the transport-layer secure communications, while adapting to sensing applications and devices with different requirements and characteristics. The proposed mechanisms also support further security functionalities such as intrusion detection, and network operations such as mobility of sensing devices between sensing domains.

Regarding security at the application layer, we consider yet a different approach complementing network-layer security and transport-layer security as previously described. We investigate the benefits of the integration of security in the communications protocol itself, rather than being transparently supported by mechanisms designed at lower layers of the communications stack. Such an approach may enable the support of granular security policies or of various authentication methods and multiple security domains, thus complementing other security mechanisms for sensing applications with such requirements.

As we discuss throughout the thesis, the various research proposals offer effective solutions to the problem of securing end-to-end communications in the context of Internet-integrated sensing applications. Other that the security of such communications, the proposed solutions also lay the ground for the design of further mechanisms accomplishing important security-related goals for the protection of WSN devices against Internet-originated threats and attacks. One important requirement of the discussed research solutions is to be able to complementarily adapt to the functional and security requirements of different applications and deployment scenarios. The complementary nature of the various security approaches is an important property of the current Internet security architecture, and one that may also be fundamental in a future Internet security architecture supporting communications with Internet-integrated WSN. As previously discussed, this aspect also differentiates the research proposals discussed in the thesis from traditional approaches to security in WSN environments.

TABLE OF CONTENTS

TABLE OF ACRONYMS

3GPP	3rd Generation Partnership Project
6LBR	6LoWPAN Border Router
6LoWPAN	IPv6 over Low-power Wireless Personal Area Networks
AAA	Authentication, Authorization and Accounting
AC	Access Control Server
ACK	Acknowledgment message
ACL	Access Control List
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
API	Application Programmer Interface
APS	Application Support Sub-layer (ZigBee)
ASN	Absolute Slot Number
BLIP	Berkeley Low-power IP stack
CA	Certification Authority
CBC	Cypher block chaining mode of encryption
CC	Consistency Check (RPL)
CCM	Counter (CTR) with CBC-MAC encryption mode
CCM*	CCM with support of integrity-only and encryption-only
CDMA	Code Division Multiple Access
CGA	Cryptographically Generated Addresses
CIA	Confidentiality, Integrity and Availability
CISUC	Centre for Informatics and Systems of the University of Coimbra
CoRE	Constrained RESTful Environments
COST	European Cooperation in Science and Technology
CRT	Counter encryption mode
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSS	Chirp Spread Spectrum
CTR	Counter mode of encryption

CTS	Clear-to-send message
DAO	Destination Advertisement Object (RPL)
DAO-ACK	DAO Acknowledgment (RPL)
DDoS	Distributed Denial of Service
DIO	DODAG Information Object (RPL)
DIS	DODAG Information Solicitation (RPL)
DODAG	Destination Oriented Directed Acyclic Graph
DODAGID	DODAG Identifier
DoS	Denial of Service
DSSS	Direct Sequence Spread Spectrum
DTLS	Datagram Transport Layer Security
ECC	Elliptic Curve Cryptography
ECDHE	Elliptic Curve Diffie-Hellman Algorithm with Ephemeral keys
ECDSA	Elliptic Curve Digital Signature Algorithm
ETSI	European Telecommunications Standard Institute
FFD	Full-function device
FTSP	Flooding Time Synchronization Protocol
GPRS	General packet radio service
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
ICV	Integrity Check Value
IDS	Intrusion Detection Systems
IEEE	Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IoT	Internet of Things
IPSec	Internet Protocol Security
ISM	Industrial, Scientific and Medical radio band
ITU	International Telecommunications Union
ITU-T	Telecommunication Standardization Sector of the ITU

IV	Initialization Vector
LCT	Laboratório de Comunicações e Telemática do CISUC
LoWPAN	Low-Power Wireless Personal Area Network
LTE	Long Term Evolution
LTE-A	LTE Advanced
M2M	Machine-to-Machine communications
MAC	Medium Access Control / Message Authentication Code
MIC	Message Integrity Code
MP2P	Multipoint-to-Point
ND	Neighbor Discovery
NFC	Near-Field Communications
NGN	Next Generation Networks
OCSP	Online Certificate Status Protocol
P2MP	Point-to-Multipoint
P2P	Point-to-Point
PRF	Pseudorandom Function
RBS	Reference Broadcast Synchronization
REST	Representational State Transfer
RF	Radio Frequency
RFD	Reduced-function device
RFID	Radio Frequency Identification
ROLL	Routing Over Low-power and Lossy Networks
RPL	Routing Protocol for Low power and Lossy Networks
RSSF	Rede de Sensores Sem Fios
SCADA	Supervisory Control and Data Acquisition
SDH	Synchronous Digital Hierarchy
SEND	SEcure Neighbor Discovery
SIA	Secure Information Aggregation
SKKE	Symmetric-Key Key Exchange
SNEP	Secure Network Encryption Protocol

SoA	State of the Art
SOA	Service Oriented Architecture
SONET	Synchronous Optical Networking
SSF	Sensor sem Fios
STCP	Sensor Transmission Control Protocol
STS	Security Token Service
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TESLA	Timed, Efficient, Streaming, Loss-tolerant Authentication protocol
TLS	Transport Layer Security
TLV	Type-Length-Value format
TPSN	Timing-sync Protocol for Sensor Networks
TSMP	Time Synchronized Mesh Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
URI	Universal Resource Identifier
USN	Ubiquitous Sensor Networks
UWB	Ultra-Wideband
VPN	Virtual Private Networks
WiMax	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WoT	Web of Things
WS	Web Services
WSDA	Wireless Sensor Data Aggregator
WSN	Wireless Sensor Network
ZDO	Zigbee Device Object

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

This thesis is the result of research work performed in the area of security for Wireless Sensor Networks (WSN) from October 2007 to July 2013, in the Communications and Telematics Services Group of the Centre for Informatics and Systems of the University of Coimbra (CISUC), in Portugal. From the start, research work focused on security issues for WSN environments, in particular considering the future employment of communication technologies enabling the integration of such environments with the Internet. The communication technologies enabling this integration are very recent, and the research work described in this thesis has accompanied the design and experimental validation of such technologies, while focusing on important and open issues in what regards security with its usage.

In this opening chapter, we start by discussing the main objectives of the research work discussed throughout the thesis. We also discuss our research approach, which encompasses the main motivations for the design and experimental evaluation of the various research proposals discussed throughout the thesis. A brief summary of the contributions is also presented, and finally the global structure of this document is outlined at the end of the chapter.

1.1 CONTEXT AND MOTIVATION

Research in WSN has traditionally approached the design of communication and security mechanisms to support applications very focused in terms of its purpose, goals and requirements, rather than to flexibly support heterogeneous applications and sensing devices. In consequence, such mechanisms are usually highly optimized for the application at hand, and the integration of WSN with external communication environments such as the Internet has not been considered in most of the previous research proposals in the area.

The resource constraints of the wireless sensing devices dictate many challenges in the development of communications and security mechanisms appropriate to support distributed sensorial applications. Its autonomous nature and its usage in deployments without continual energy sources determine that such devices must support wireless communications and be powered by batteries. In general, the characteristics and constraints of WSN devices represent many challenges for the enabling of mechanisms that typically are very computationally demanding, as is the case of security.

As a consequence of the previous aspects, the communications and security mechanisms currently empowering the Internet architecture have traditionally been considered to be inadequate for wireless sensing applications, and the integration of WSN with the Internet was not considered as a research goal. As we observe throughout the thesis, this perception is currently changing and communications mechanisms are starting to appear that may

support Internet communications over constrained WSN environments, and in its context the necessity of protecting such communications using appropriate security mechanisms.

Contrary to the previous conception of WSN as isolated communication environments, its integration with the Internet is currently raising an increasing interest in the research community and industry. As technologies become available to support Internet communications on WSN environments, various challenges arise in respect to the appropriate support of security. In fact, the support of security in such environments must cope not only with the limitations and constraints of its communications and devices (as in traditional approaches to security in WSN), but also with new aspects related with the exposure of WSN domains to Internet-originated security threats. In the same context, the heterogeneity of applications in terms of its requirements and deployment characteristics must also be taken into account. The research efforts described in this thesis have a particular focus in this context, and address the problem of how to support viable and effective security mechanisms for Internet-integrated WSN, in particular regarding the support of end-to-end communications with such environments. Our research efforts are also motivated by the fact that the integration of WSN with the Internet may represent an important step on the evolution of the Internet communications and security architectures to encompass WSN applications.

As in the traditional approaches to security in WSN environments, security mechanisms designed for Internet-integrated WSN must cope with and appropriately balance various important aspects. The constraints and characteristics of WSN devices and communications call for the usage of mechanisms that are able to optimize the usage of the limited resources available for energy, while on the other hand such mechanisms must not compromise the support of heterogeneous applications and devices. Thus, appropriate compromises must be achieved between security and its impact on the lifetime of sensing applications, among other functional aspects that may be appropriately considered. This approach also motivates the consideration of predefined requirements describing the applications in terms of various functional and security aspects, as we observe throughout the thesis.

1.2 SECURITY APPROACHES FOR ISOLATED WSN ENVIRONMENTS

As we observe in detail in Chapter 2, traditional research approaches to security in WSN environments targeted mainly the design of mechanisms with very specific goals in mind. Such proposals also materialize a previous conception of WSN as vehicles to enable new distributed sensorial applications employing devices that are unable to support communications with external environments. This also applies to how security is addressed in such proposals, and in consequence security mechanisms are designed with very particular requirements in mind, rather than to support heterogeneous applications and Internet communication technologies.

In the context of classic WSN security approaches, we also observe that most proposals address only communications technologies at lower layers of the protocol stack. This is a consequence of the fact that in initial WSN applications only link-layer (hop-by-hop) communications were required, possibly accompanied by routing solutions designed and optimized with particular goals. In this context, many previous WSN applications deemed communications at higher layers of the stack to be unnecessary or even unfeasible, the same applying to security mechanisms designed to go along with such communication technologies. As we observe in subsequent chapters of the thesis, the integration of WSN with the Internet is currently motivating the design of communication technologies supporting Internet communication at the network, transport and applications layers of the stack. The absence or the insufficiency of security mechanisms to protect communications using such technologies motivates the research efforts and proposals presented in the thesis.

In Chapter 2 of the thesis we present a State of the Art (SoA) study on security in WSN, which we divide in two main contexts. First we discuss previous research and standardization approaches for isolated WSN applications, which provide an important insight and guidance into what are the fundamental security problems on WSN and on how they may be addressed by research. In the context of such proposals, we also analyze previous approaches on defining security architectures for WSN environments. Next in this chapter we analyze how security is addressed in recent research proposals targeting the integration of WSN with the Internet, thus more in line with the research proposals presented and evaluated throughout the thesis.

1.3 SECURITY APPROACHES FOR INTERNET-INTEGRATED WSN ENVIRONMENTS

As communication technologies appear that facilitate the interconnection of WSN with the Internet, a change in the perception on how security must be addressed in WSN environments is also taking place. Current concepts such as the Internet of Things (IoT) and the Web of Things (WoT), or those that refer to particular communication approaches such as Machine-to-Machine (M2M) communications, also play a part in this change of perception, bringing light into what could be the requirements of future sensing applications employing Internet communication and security technologies. In this context, security mechanisms may be developed providing solutions to address security as an enabling factor of new sensing applications requiring WSN integrated with the Internet. We may also expect this integration to be facilitated and evolve at the pace of research results on appropriate communications and security technologies.

The integration of WSN with the Internet may also motivate new approaches to the addressing of security, previously not considered for such environments. For example, end-to-end Internet communications with constrained wireless sensing devices were previously considered unnecessary or unfeasible, due to the possibly large impact of Internet communications on the resources of WSN devices and communications. Also, the addressing

of security aspects such as key management, intrusion detection, trust or anonymity, among others, may require new (possibly cross-layer) approaches, compatible with the employment of WSN Internet communication technologies.

One important property of the Internet communications and security architectures is the support of heterogeneous applications, devices and communication technologies. In the same context, we may expect that such architectures evolve to support Internet communication technologies designed to support WSN environments, as we consider in the research proposals presented in the thesis. We also consider that such proposals may contribute with this purpose, by providing solutions to address end-to-end security for WSN Internet communications at various protocol layers.

1.4 RESEARCH OBJECTIVES

The objectives motivating the research work described in the thesis have been initially related with the design of security mechanisms for performance-controlled environments, in the context of the GINSENG EU FP7 [1] research project, and later evolved to the consideration of security in the context of the integration of WSN with the Internet, particularly using communication technologies being designed to support Internet communications in WSN environments. The following are the **main research objectives** of the research work described in the subsequent chapters of the thesis:

- Propose new research solutions for the **support of security in the context of end-to-end Internet communications in WSN environments**. In particular, our goal is to approach different strategies for the support of end-to-end security, in what respects the protocol layer at which security is enforced, how security is implemented in constrained wireless sensing devices, and the support of techniques to protect WSN devices from external security threats. The usage of different approaches to end-to-end security also facilitates the support of heterogeneous applications and devices.

- Propose a **reference model for end-to-end security mechanisms in Internet-integrated WSN**, in particular supporting communications at the network, transport and application layers. This reference model considers the Internet communication technologies currently being designed for WSN environments, as well as the various approaches to end-to-end security discussed throughout the thesis.

- The proposed research approaches are **evaluated experimentally**, in order to measure the impact of the proposed mechanisms on critical resources of constrained wireless sensing platforms supporting the applications. For this purpose, we consider an evaluation methodology and the description of applications and deployment scenarios in terms of its functional and security requirements. It is also our goal that this approach

supports applications that may statically or dynamically opt for the most appropriate end-to-end security mechanism.

The research proposals described in the thesis target the addressing of security requirements that are fundamental for the successful interconnection of WSN with the Internet, in particular in what respects the enabling of security for end-to-end communications. Such communications are enabled by Internet communication technologies currently being designed for WSN environments, which we analyze in detail in Chapter 2.

1.5 RESEARCH APPROACH

For the development and evaluation of the proposals discussed throughout the thesis we consider the integration of the proposed mechanisms into a reference operating system, which supports the experimental evaluation of the impact of security for applications with particular security and functional requirements. The WSN Internet communication technologies in the context of which the proposed research solutions are implemented and evaluated are also available, modified or implemented as appropriate using the same operating system and reference wireless sensing platforms. The considered approach is also motivated by the fact that some of such communication technologies are currently work in progress, and as such the evaluation of new security mechanisms in its context may provide useful insight on how such technologies may be designed or evolve, other than by contributing to its evaluation.

The experimental evaluation of the research proposals is deemed to be fundamental to prove its effectiveness and efficiency, and it is our conviction that the experimental evaluation of new solutions for WSN environments provides more valuable and precise insight into the problem at hand when compared with simulation approaches, in which it is difficult to properly capture various effects that in practice have a direct impact on the evaluation of the effectiveness of new mechanisms. This is particularly true in what respects the impact of security on resources that are critical for the normal operation of wireless sensing platforms, such as memory, energy and computational capabilities.

1.6 RESEARCH CONTRIBUTIONS

From any work, there are always a disparate number of results and accomplishments. The following contributions have also resulted in a number of research publications, as previously discussed. Looking back, probably the major achievements resulting from the research work described in this thesis are the following:

- The **proposal and experimental evaluation of complementary approaches to end-to-end security for WSN Internet communications at different layers** of the communications stack. The complementary nature of the various approaches to

end-to-end security contributes to the support of heterogeneous applications and sensing platforms, as previously discussed.

- The proposal of a **reference model for end-to-end security with Internet-integrated WSN**. This model enables the integration of WSN with the Internet using appropriate communication technologies at the various protocol layers and the complementary end-to-end security approaches proposed throughout the thesis. This model also supports components designed with the purpose of supporting the management of **application security and functional profiles**, key management and intrusion detection. Application security and functional profiles identify requirements of applications that have a direct impact on the resources required to support security, and the evaluation and monitoring of security in respect to the various mechanisms employed may be based in such profiles and appropriate performance metrics. We also identify the possibility of using dynamic security, meaning that applications may determine the most appropriate end-to-end security configuration or mechanisms according to various functional parameters and conditions.

- The proposal of **compressed security headers** to support **network-layer security** for end-to-end communications in the context of Internet-integrated WSN. Our approach to network-layer security in such environments also considers the integration of the proposed mechanisms in the existing IP Security architecture, in order to support secure end-to-end communications with devices in Internet-integrated WSN environments.

- The proposal of **transport-layer security with mutual and delegated public-key authentication**, consisting of mechanisms to transparently intercept and mediate the authentication and key agreement phase of secure transport-layer communications. This approach proposes a solution to the problem of supporting costly cryptographic computations in constrained sensing platforms, which we find to be particularly critical in the case of authentication and key agreement as currently defined for transport-layer security. The proposed solution also supports the usage of further security mechanisms to protect WSN environments from external security threats, and mobility of sensing devices from the point of view of end-to-end security.

- The proposal of **application-layer security integrated with the application-layer protocol**. This approach is complementary to our research proposals at the network and transport layers, and supports the usage of flexible and granular security policies, as well as of different authentication mechanisms and multiple trust domains.

As we discuss throughout the thesis, WSN may be effectively integrated with the Internet communications infrastructure in what respects the support of security for end-to-end communications with external or Internet devices, in contrast with the classic perception of

research on security for WSN environments. We also consider a quantifiable (measurable) approach to security, and that mechanisms may be in place to enforce and monitor end-to-end security as appropriate. Other than the previously discussed research solutions, various aspects certainly remain to be addressed regarding security in Internet-integrated WSN environments, which also motivates future research work. We discuss research opportunities throughout the thesis, and also in Chapter 7.

1.7 STRUCTURE OF THE THESIS

The thesis is divided in seven chapters, starting with an analysis on the SoA on security in WSN and proceeding to the discussion of the proposed research solutions. In detail, the following are the goals of the various chapters forming the thesis:

- **Chapter 1** (this chapter) presents the motivation for the undergone investigation, the research objectives and approach, together with an outline of the main research contributions described in the thesis.

- **Chapter 2** presents a SoA analysis on security on WSN environments. The research proposals are discussed considering two distinct and complementary contexts. First the chapter discusses previous research proposals targeting isolated WSN environments, and next we focus on more recent works targeting security in the context of Internet-integrated WSN. In this chapter we also identify the current approaches enabling the integration of WSN with the Internet, an analysis we find useful in order to contextualize the research mechanisms proposed throughout the thesis.

- **Chapter 3** presents a reference model for end-to-end security in Internet-integrated WSN, which supports the employment of the research solutions proposed in the thesis. We also discuss the methodology considered for the experimental evaluation of the various research proposals, as well how end-to-end security can be statically or dynamically reconfigured by applications employing Internet-integrated WSN.

- **Chapter 4** discusses security at the network-layer using Internet-integrated sensing devices. The research solutions proposed and experimentally evaluated in this chapter support end-to-end secure communications with constrained WSN sensing devices, in various configurations and operational modes. We also discuss the effectiveness of extending the IP Security architecture to encompass WSN applications and devices using the proposed mechanisms.

- **Chapter 5** discusses security at the transport-layer using Internet-integrated sensing devices, particularly on the employment of solutions enabling the transparent interception and mediation of end-to-end authentication and key agreement in the context of transport-layer security. Such mechanisms enable the offloading of costly

security-related operations from constrained sensing devices to more powerful Internet entities and the support of further security mechanisms to protect communications on the WSN domain and WSN devices against Internet-originated security threats.

- **Chapter 6** discusses security at the application-layer using Internet-integrated sensing devices. In particular, the proposed research solutions target the design of security directly in the context of the application protocol, with the purpose of enabling granular and semantic security policies, together with the support of different client authentication methods and the transversal of multiple security domains.

- **Chapter 7** concludes the thesis, summing up the major results from the various research proposals and identifying future research opportunities.

We also note that our discussion in Chapter 2 is complemented in the various chapters of the thesis, while focusing on research proposals for particular layers and communication technologies. In the same vein, research opportunities are also identified throughout our discussion and complemented by the discussion in Chapter 7.

2 SECURITY FOR WIRELESS SENSOR NETWORKS[1]

In this chapter we present a study on the State of the Art (SoA) on research proposals addressing security in WSN environments. We analyse the proposals targeting isolated WSN environments and also more recent works addressing security in the context of Internet-integrated WSN environments. In order to properly contextualize such research solutions, we also identify the main approaches for the integration of WSN with the Internet.

We start by identifying the fundamental aspects of WSN security, namely the applicable attack and threat model, its main security requirements and why traditional approaches to security may not be appropriate to constrained WSN environments. We next analyse the research proposals targeting security in WSN environments, namely those that focus on isolated WSN applications and more recent works applicable to Internet-integrated WSN.

2.1 SECURITY IN WSN ENVIRONMENTS

Our following discussion focuses on fundamental aspects to consider when addressing security in WSN environments, some of which are also inherent of most wireless communication environments. We begin by identifying the applicable attack and threat model, which subsequently enables the identification of the security requirements to consider in such environments. Finally, we discuss why traditional approaches to security on Internet wireless and wired environments are usually inadequate to such resource-constrained communication environments.

2.1.1 ATTACK AND THREAT MODEL

In addition to the security threats that are inherent to the characteristics and constraints of low-power wireless communication, WSN environments may also be targeted by attacks and threats due to the employment of wireless sensing platforms. Also, the exposure of WSN to global Internet communication may also promote new threats and attacks. In this context, our following discussion applies to isolated WSN environments and also for WSN that are integrated with the Internet communications infrastructure.

---

[1] *This chapter has supported the following publications:*

- Granjal J, Monteiro E, Silva J. *Security Issues and Approaches on Wireless M2M Systems*. *Wireless Networks and Security*. Springer Berlin Heidelberg, 2013. 133-164.
- Granjal J, Monteiro E, Silva J. *A survey on Security Mechanisms for the Internet of Things*. *IEEE Surveys & Tutorials, 2014* (status: pending, revised version submitted).
- Granjal J, Monteiro E, Silva J. *Security in the integration of low-power Wireless Sensor Networks with the Internet: a Survey*. *Elsevier Ad Hoc Networks 2014* (status: pending, revised version submitted).

In what respects its level of access to WSN devices and communications, attackers against the normal functioning of WSN may be classified as either internal or external. Regarding how attacks may be performed and perceived by legitimate communicating entities, attacks may on the other hand by either passive or active. The following are the main characteristics of this classification:

- An internal attacker is able to compromise a node and subsequently participate in a communications session as a fully legitimate entity. This implies that, even if security or cryptographic mechanisms are in place, the attacker may have access to the secret keying material required to process security, and thus participate in communications as a normal entity of the network. When considering Internet-integrated WSN, end-to-end communications may take place between external devices and constrained sensing devices on the WSN domain, and thus an internal attacker may also be a compromised external or Internet device.

- Contrary to internal attackers, an external attacker is usually only able to listen on the wireless communications channel and try to obtain or derive knowledge about the functioning of the network. Therefore, an external attacker is usually not in the possession of the secret keying material required to interpret encrypted communications. When compared with external attackers, internal attackers are usually more difficult to defend against.

- A passive attack is one in which the attacker does not interact with other devices on the network, and which consequently may be able to perform its actions without being noticed by other network entities. Since WSN employ wireless communications, passive attacks may consist on the listening of communications and on the breaking of the security based on the collected packets.

- Contrary to passive attacks, an active attacker may attempt to compromise the security of the network using any mechanisms, without concerns about its actions being noticed. For example, an active attacker may try to compromise the availability of the network by injecting bogus packets on the wireless communication channel or by trying to physically compromise a sensing device to extract useful information from its internal memory.

According to our previous classification, attacks against WSN may be either active or passive, and may be perpetrated by both internal and external attackers. Attacks of such types may target both isolated WSN environments and Internet-integrated environments. As we discuss later in this chapter, numerous surveys currently exist analyzing research works focusing on security for isolated WSN environments [2][3]. Later in the chapter we also focus on more recent proposals targeting security in the context of Internet communication technologies designed for Internet-integrated WSN environments.

2.1.2 ATTACKS AGAINST WSN

Numerous aspects may be identified that in practice enable security threats against WSN environments, such as the resource constraints of sensing devices, their physical exposure in many deployment scenarios or the employment of wireless communications and particular communication protocols, among others. The exposure of wireless sensing devices may also promote its physical compromise by non-authorized or malicious individuals, and as a consequence the data (including security-related data) stored in such devices may be available for attackers to use.

An eavesdropping attack is a passive and external attack, consisting in an attacker listening and possibly recording wireless network traffic with the goal of obtaining or deriving any type of useful information. This attack is purely passive, since the attacker may be able to obtain privileged information without interfering with the normal operations of the network. Eavesdropping may also be conjugated with packet insertion, facilitating the access of the attacker to network resources without authorization. Eavesdropping and insertion are a problem of wireless communication environments in general, not only of WSN.

A spoofing or masquerading attack is an active attack perpetrated by an internal or external attacker, in which an attacker masquerades as another in order to achieve some form of illegitimate advantage. This attack can also include the deleting and replaying of networks packets, and thus is an active attack.

Attacks of the Denial of Service (DoS) type consist in general of an attacker performing some type of malicious action in order to prevent a legitimate user from being able to access a service or network functionality. DoS attacks are particularly pernicious in low-energy WSN communication environments, since they may target the exhaustion of the limited resources and of the energy available on wireless sensing devices, or the prevention of the normal functioning of wireless communications between devices. In the same context, Distributed Denial of Service (DDoS) attacks consist in the simultaneous action of many attacking nodes flooding a target with requests. As examples of DoS attacks we may consider jamming attacks against the normal operation of communications at the physical layer, which enables an attacker to disrupt wireless communications by overwhelming the radio carrier with bogus data, or attacks against the MAC (Media Access Control) layer, consisting of an attacker purposely creating collisions by sending its own packet when a legitimate user's packet is being transmitted. In WSN, DoS attacks may enable an attacker to disable sensor nodes by draining their battery by continuously transmitting bogus packets destined to that node.

The design of Internet communication technologies for WSN will also enable security threats and attacks against mechanisms designed for higher layers of the stack, and also against security mechanisms which may be transversal to the communications stack, for example mechanisms supporting authentication, authorization and key agreement. In this context, authorization violation may take place when some entity is able to use services without

having proper authorization. This threat may in practice involve the insertion of forged packets containing authentication information, thus also representing an active attack, and may also be facilitated by the wireless nature of WSN communications. Threats may also be present against mechanisms designed to support security-related requirements such as anonymity or trust. For example, repudiation happens when an entity is able to falsely claim that it is not responsible for some action, by compromising mechanisms designed to support identification, authentication and trust.

The previously discussed security threats apply also in general to other communication environments, in particular those that depend on wireless communications. In this context, the fact that many of such threats can potentially cause a higher impact on WSN is deeply related to the constraints on resources available on wireless sensing devices and to the low bandwidth available on low-energy WSN communication environments.

The integration of WSN with the Internet may enable end-to-end communications between wireless sensing devices and external or Internet hosts, and in this context insider attackers may be either sensing devices or external or Internet hosts. An outsider attacker may target devices and communications also in the WSN or Internet communication domains. We observe that the integration of WSN with the Internet may contribute to amplify the previously discussed security threats and attacks. Wireless sensing devices in Internet-integrated WSN may be more vulnerable to Internet-originated attacks, and as such security mechanisms will be of prime importance to guarantee the feasibility of this integration.

The classic approach against external attacks consists in the employment of cryptography-based security protocols and procedures, which are designed to guarantee fundamental security properties as confidentiality, integrity, authentication and non-repudiation to the communications taking place between devices. In this context, effective security through encryption requires appropriate authentication and key management mechanisms in place, since encryption algorithms are only effective as long as the keys employed are refreshed periodically. On the other hand, protection against internal attackers usually requires other mechanisms, and in this case prevention is usually the key for success. Internal attacks can be prevented with security procedures such as security perimeter enforcement via access control mechanisms, or intrusion prevention and detection systems. In many WSN deployments sensing devices are physical exposed, and in this context mechanisms against the tampering of such devices may also be useful in preventing attacks.

2.1.3 SECURITY REQUIREMENTS

From our previous analysis on the attacks and threats against WSN environments, we are able to identify a group of fundamental security requirements that should be enforced by appropriate security mechanisms. While realizing that the level of security may depend on the application at hand, we are able to identify the following general security requirements applicable to WSN applications and WSN Internet communication technologies:

- Confidentiality of the information exchanged between WSN sensing devices and/or of the data stored on such devices. Mechanisms must be in place to guarantee that this information is only available to authorized entities. As WSN environments employs wireless communications, they are vulnerable to attacks such as traffic analysis, and cryptography based on symmetric or asymmetric solutions may provide a solution in this context.

- Integrity of the information exchanged between WSN sensing devices and/or of the data stored on such devices. Integrity allows a node to confirm that the data received from other devices was not modified in transit, either intentionally or by accident. We again observe that wireless communications may facilitate eavesdropping and false data injections attacks, which may be conducted to compromise the integrity of the communications. Integrity is also a particularly important requirement with respect to data aggregation and time synchronization operations, and various research proposals for WSN environments focus on this aspect [4]–[6]. The detection of false or corrupted values allows the discard of such data from the data aggregation computations. As with confidentiality, cryptography-based protocols may be employed that associate error correcting codes, hash values or digital signatures to the transmitted or stored data, allowing to validate the data and detect illegal modifications.

- Freshness of the data exchanged between devices, implying that mechanisms may be required to protect against data replay attacks. Replay attacks can enable an attacker to assume a false identity in an ongoing or new data session, for example by retransmitting packets containing authentication or authorization information. Data freshness may also be supported by the same mechanisms supporting data integrity and authentication.

- Authentication of the communicating entities, which implies that mechanisms may be in place that allow identifying and authenticating the communicating parties and consequently the true origin of the received data. Authentication may be conjugated with integrity verification mechanisms, to enable the detection of spoofed or maliciously injected messages. Encrypted hashed or digital signatures may serve this purpose, by providing a mechanism that relates integrity information with a secret that only the true sender is supposed to know. Authentication is also a major aspect in classic research proposals, for example to protect clustering management operations and communications between cluster heads

- Accountability of communications and other relevant WSN operations, which may require mechanisms to identify the entity requesting a particular service, triggering an action or sending a message. Accountability mechanisms can also be designed to work side-by-side with traffic control and quality of service (QoS) mechanisms.

- Availability, implying that that legitimate entities should be able to access a particular service or information, while also benefiting from the proper operation of that service. This is also a relevant requirement for WSN environments, as it is related to the capability of detecting and adjusting to security threats or attacks. Resilience against attacks targeting the availability of the network is a desired property, and in this context graceful degradation mechanisms may also play again an important part.

- Access control, meaning that accesses to particular services or data may be restricted only to authorized entities. This requirement is also related with the problem of resources allocation and verification of service application bounds.

- Robustness and resilience against outsider attacks, which ideally implies that mechanisms are in place to enforce resistance of the network against attacks such as eavesdropping, packet injection and node compromise and failure. For example, cryptographic-based security mechanisms can help in detecting eavesdropping and injection attacks, while node failures can be addressed by designing mechanisms and protocols that are able to identify failed nodes and adjust dynamically.

- Adjusting in the presence of internal attacks, which may imply being able to detect compromised nodes and act accordingly, for example by revoking cryptographic keys. Resilience against node compromise is also a possible approach, guaranteeing graceful degradation with respect to performance and delivery of data.

- Secure management, which is related to the employment of mechanisms to support operations such as key distribution, routing security and security clustering, among others. As previously discussed, this requirement is also relevant in the context of Internet-integrated WSN, and mechanisms may be designed in a cross-layer fashion to implement such security procedures.

As we discuss later in this chapter, various mechanisms have been proposed for isolated WSN environments with the purpose of supporting one or more of the previously discussed security requirements. Security architectures may also play an important role in relating complementary security mechanisms, and in contextualizing their usage according to the requirements of the applications and deployment scenarios.

The integration of low-power WSN with the Internet will also require appropriate security mechanisms to support the previous discussed security requirements. As in isolated Internet environments, end-to-end communications involving WSN sensing devices will require appropriate security assurances in terms of confidentiality, integrity, authentication and non-repudiation of the transmitted messages. End-to-end security may be addressed in the context of the communication protocol itself, or on the other hand by external mechanisms.

Another class of security requirements may be targeted with cross-layer security approaches, for example in what respects threats due to the exposure of WSN environments to the global Internet communications infrastructure, in integration approaches employing Internet communication technologies designed for WSN. In this context, availability and resilience against Internet-originated attacks may be particularly important requirements for many sensing applications. Other requirements as privacy, anonymity, liability and trust may also be considered fundamental for the acceptance of most of the envisioned applications on the IoT employing Internet-integrated WSN communication environments, and may be targeted also (by following possibly a cross-layer approach) by appropriate mechanisms. We also note that the research proposals described in the thesis focus on the protection of end-to-end communications with sensing devices in respect to the confidentiality, integrity, freshness and authentication of such communications, when WSN Internet communication technologies are in place.

2.1.4 CHALLENGES TO CLASSIC SECURITY APPROACHES ON WSN ENVIRONMENTS

The characteristics and constraints of WSN application and devices typically difficult the employment of classic Internet security approaches to WSN environments. As we have previously discussed, the limitations of low-power wireless sensing devices in terms of critical resources such as memory, computational capability and energy usually dictate the design and adoption of highly optimized mechanisms to support communications and security on WSN. The same reasoning applies to a WSN interconnected with the Internet, and in this scenario we have also to consider additional threats that may be present due the possible exposure of constrained sensing devices to global Internet communications.

Many research protocols do exist targeting security for WSN and ad hoc networks, and from the start researchers recognized the necessity of addressing security and cryptography differently, due to the inherent resource and computational constraints of these networks, which pose particular challenges to the implementation of security mechanisms. The constraints of wireless sensing platforms usually dictate that performance and security must be balanced against the available computational and storage capability. For example, wireless communications between sensor nodes may consume a large percentage of the available energy, more than sensing and computation operations. In the same context, cryptographic operations may not only be costly in terms of the computational power required to process security, but also because the security protocols introduce an extra overhead on communications, as more messages need to be exchanged for key management purposes, and messages become larger as authentication, initialization and encryption data must also be transported. In this context, the proportionality between the data to be transmitted and the overhead of the new security mechanisms must be carefully considered, also because such data may only occupy a small percentage of the total payload space. The same security solutions may also impact on the available storage space on sensing devices, for example to store large cryptographic keys or digital certificates.

There are various limitations of WSN devices that in practice pose difficulties to the design of appropriate security mechanisms. For example, its finite energy budget may open new types of DoS attacks, as victim nodes are forced to exhaust their energy budget quickly and die, also because attackers can have much more energy at their disposal than sensor nodes. Node exposure is also typically an issue, as capture may be impossible or at least very difficult to prevent due to the large number and geographical distribution of devices in many WSN deployments. Rather than assuming that the devices are physically protected as in many Internet security environments, the reasonable posture may be to plan security solutions with this aspect in mind. The capture of a node allows the attacker to perform various types of internal attacks, as previously analyzed. Special secure memory devices are available but will probably remain unfeasible for most WSN applications employing inexpensive sensing platforms, meaning that resilience may instead be integrated in the security protocols.

Other characteristics of WSN are accepted by researchers as posing challenges to the development of WSN security solutions. For example, the random topology of sensing devices may difficult the employment of encryption between groups of neighboring sensor nodes. Even if a key management solution is in place, such keys may be important for the support of a secure bootstrap procedure of the sensing devices. Research may thus target the development of key agreement protocols that do not require the previous pre-deployment of such keys, nor any type of previous knowledge about neighborhood relationships.

Data aggregation is a fundamental operation for many WSN applications and one that may also pose challenges to the development of security solutions. In data aggregation intermediate nodes typically need to access and modify the information contained in the packets, which may be incompatible with the employment of data integrity verification procedures. The hierarchical nature of the network and employment of tree structured routing protocols may also permit an attacker to determine the position and attack the root nodes or other nodes nearby, in order to disrupt the normal operations of the network, as such nodes are the ones with access to messages containing the most important information, for example aggregated collection of readings.

In regard to the usage of existing Internet security mechanisms to protect Internet-integrated WSN, we are able to verify that the constraints of WSN devices and applications motivating previous research efforts on WSN security [2][3] also apply to WSN environments integrated with the Internet. This implies that the limitations of sensing devices in terms of critical resources will also guide the design and adoption of highly optimized mechanisms to support communications and security on such environments. On the other hand, contrary to most classic WSN security approaches, newer approaches to security on Internet-integrated WSN may also consider the employment of heterogeneous sensing platforms and the support of applications with different characteristics and requirements.

The employment of Internet communication technologies designed for WSN environments may not only facilitate its integration with the Internet at the protocol level, but also promote WSN communications as a transparent and agnostic communications medium, in the same vein as how data communications are supported in the current Internet communications infrastructure. Despite the existence of big challenges, the design of new security mechanisms for WSN in the context of a global communications and security architecture provides the almost unique chance to take into account security issues from the beginning.

2.2 PROPOSALS ON SECURITY FOR ISOLATED WSN ENVIRONMENTS

In our following discussion security on WSN environments is analyzed from the point of view of the possible threats and attacks at the various protocol layers, thus following a layered approach as in existing analysis in the literature [7][8]. For each protocol layer, we discuss in detail the possible threats to security in WSN and the corresponding research proposals on solutions to address such threats. Our analysis follows a bottom-up layered approach, with cross-layer security threats and proposals being discussed at the end.

Although the research proposals discussed throughout the thesis apply to WSN environments which are integrated with the Internet, we consider that the analysis of the security issues and approaches in isolated WSN environments provides valuable information and guidance on the design of new security mechanisms appropriate to Internet communication technologies designed for WSN. We must nevertheless note that our following discussion is necessarily bounded in time, since it relates to a study performed in the initial phase of our research work. As previously observed, our research focus later evolved to consider communications technologies enabling Internet communications on WSN environments, and our analysis of the proposals in this context is performed later in the chapter.

2.2.1 PROPOSALS AT THE PHYSICAL LAYER

The main attacks against security at the physical layer in WSN environments are jamming, tampering and traffic analysis. The first two attacks belong to the Denial of Service (DoS) category, while traffic analysis involves listening to the wireless communication channel in order to gather information on how the network operates. The jamming attack consists on an attacker interfering with WSN Radio Frequency (RF) communications and being able to disrupt normal network communications.

Classic approaches against jamming attacks include the employment of spread spectrum wireless communications, with the purpose of requiring the attacker to spend more energy while trying to jam the network. Example of this approach are found in the usage of code spreading in the Global System for Mobile Communications (GSM) [9] technology, and the employment of frequency hopping modulation techniques. The cost of such approaches

usually makes them unfeasible for WSN environments. In alternative, techniques could be employed that allow nodes to automatically switch to low power cycles during jamming attacks, or switch to other communication medium if available. A solution has also been proposed in [10] that enables the mapping of a jammed region, in which the network is able to route around the jammed region.

The tampering attack consists in the physical compromise of a sensing device, and is usually easy to perform since such platforms do not provide resistance against this threat. In this situation, the attacker may be able to extract cryptographic keys or other security-related data, or exploit shortcomings of the software implementation of functionalities supported by the captured device. Costly solutions such as tamper resistant packaging or sensor nodes prepared to automatically erase data after capture are technologically available, but are in principle also too costly to be employed in real WSNs scenarios. In a real scenario the loss of a sensor node must be tolerated by the network and compensated by other (possibly redundant) sensing devices. In this scenario the critical component to protect may be the data and not necessarily physical devices themselves, as discussed in [11].

A promising research approach against tampering attacks is in the exploration of code attestation techniques, both at hardware and software levels. At the software level, the employment of software-based attestation mechanisms in WSN has been discussed in [12]. On the other hand, hardware attestation techniques to be adopted can possibly be based on the proposals by the Trusted Computing Group [13] and the Next Generation Secure Computing Base [14]. Due to the inherent cost of such solutions, WSN may preferably adopt algorithmic solutions, which are able to support resilience based on redundancy. Alternative approaches would be to replicate state among the nodes or use majority-voting techniques to detect inconsistencies.

Regarding traffic analysis attacks in WSN, in this case adversaries may explore known traffic patterns in WSN applications, for example many-to-one or many-to-a-few communications implemented by applications where the nodes send their data back to a base or sink node. By observing the traffic pattern and volume of information transported on the network, an adversary can infer about the topology of the network and determine the location of strategic devices such as base stations. Probabilistic routing schemes and fake messages can be employed against this threat in WSN environments. Probabilistic routing schemes have been proposed in [15] and consist of choosing the next node from among a number of candidate nodes, taking into consideration the link quality and residual energy of such nodes. On the other hand, mechanisms employing fake messages may contribute to protect legitimate traffic patterns, but may be problematic for WSN since they introduce additional overhead in terms of traffic and energy consumption.

As in other wireless communication environments, security threats against WSN communications at the physical layer can be difficult to prevent or avoid. Although various research proposals have been produced addressing such issues in WSN environments, as in

other communication environments security mechanisms designed for upper layer communication technologies or protocols can indirectly contribute to avoid the undesired effects of physical layer security attacks.

2.2.2 PROPOSALS AT THE DATA LINK LAYER

Communications at the link layer in WSN support mechanisms for neighboring nodes to access a shared wireless communication medium, using access rules defined in the context of techniques such as Time Division Multiple Access (TDMA) or Carrier Sense Multiple Access with collision avoidance (CSMA/CA). At this layer, the attacker seeks to exploit vulnerabilities of the MAC protocol, which may allow him to induce malicious collisions, to exhaust the energy of nodes by continuously forcing retransmissions, or to get an unfair share of the communications medium by simply not following the MAC rules, as analyzed in [8].

Induced collisions may be considered a type of link layer jamming, in which the attacker usually explores particularities of the functioning of the MAC. The insertion of bogus ACK messages can lead to an exponential back off of the MAC protocol, effectively denying access of legitimate sensor nodes to the WSN. Frames injected can also provoke collisions and contention. In this context, Wood and Stankovic [8] propose the employment of error correction codes against collision attacks, and another possibility is to employ collision detection techniques. The former involves the use of extra bits and is not immune to corruption, while the later is currently not proved to be an effective solution for WSN.

The exhaustion attack consists in forcing a node to continuously retransmit frames due to collisions until its battery is exhausted. A variant of this attack is when a self-sacrificing node continuously sends acknowledgment (ACK) messages, forcing neighbors to respond with clear-to-send (CTS) messages. Possible solutions against this type of attack consist on the usage of time division multiplexing techniques to avoid indefinite postponements during collisions, and the modification of the MAC protocol to limit the rate of requests [8]. An attacker may unfairly obtain access to the communications medium by abusing the MAC priority schemes. One possible and partial solution against this threat consists in the usage of small frames in order to capture the channel for smaller periods of time.

One may consider that research concerning prevention or avoidance of attacks against the physical and link layers can also result in the design of new secure MAC protocols. These protocols can employ mechanisms that allow the detection and isolation of regions targeted by jamming attacks, allowing the network to route around the compromised area [8]. On the other hand, the design of Internet communication technologies for low-energy WSN environments calls for the adoption of particular MAC solutions that may not support many of the proposed mechanisms against security threats at the MAC layer. This is visible in the current adoption of IEEE 802.15.4 for the design of 6LoWPAN-based communication technologies, which we discuss later in the present chapter. Also in this context, security mechanisms designed for upper layers of the stack can contribute to indirectly avoid security threats against the normal functioning of the MAC layer.

2.2.3 PROPOSALS AT THE NETWORK AND ROUTING LAYERS

In the following discussion we focus on research proposals targeting security mechanisms designed for communications at the network and routing layers in isolated WSN environments. As the network-layer may enable end-to-end communications with devices external to the WSN, we also analyze research proposals on network-layer security in the context of Internet-integrated WSN later in this chapter. As for other communication technologies, network-layer communication technologies and routing protocols designed for WSN have to be energy and memory efficient, while at the same time providing resistance against security attacks or sensing node failures. In the context of previous research proposals for WSN, there have been many power-efficient routing protocols proposed for WSN environments as discussed in [16], although many of them haven't been designed with security in mind.

As discussed in [17], the main attacks against the security and normal functioning of routing protocols in WSN are black holes, wormholes, spoofing, selective forwarding, sinkholes, hello floods and acknowledgment spoofing attacks, that we proceed to analyze. In black hole attacks a compromised node advertises a very low or zero cost route to its neighbors, with the purpose of attracting and discarding network packets. Black hole attacks are particularly effective with distance vector routing protocols. In the wormhole attack messages are tunneled over a low latency link to another part of the network and then replayed there. It is usually considered that geographical routing protocols are in principal resistant to these attacks, and tight time synchronization is important to fight them. A solution to this attack has been proposed in the form of packet leashes [18], which consists in adding additional information to the packet in order to restrict the maximum distance that the packet can travel in a given amount of time. Another solution has been proposed in [19] where a graph theoretic framework for modeling wormholes allows the detection and defense against wormhole attacks. Nevertheless, in this work the authors don't present technical details for the proposed solution.

In spoofing attacks packets containing routing information are altered and replayed, with the purpose of creating routing loops or of increasing the end-to-end delay of communications. Link layer encryption and authentication helps against outsider attackers trying to inject such falsified messages, but as previously discussed is not effective against insider attacks. In selective forwarding attacks the attacker is able to include himself in the data flow and to chose which packets should be forwarded or dropped, thus creating a black hole or preventing data from reaching specific nodes. An example of a selective forwarding attack is the DoS attack against broadcast messages in WSN. Proposed solutions include the employment of redundancy, for example via the employment of multi-path routing techniques.

In the sinkhole attack an adversary tries to attract traffic towards the compromised nodes, with the goal of discarding it. This attack can work as the launching block for selective

forwarding, making a node attractive to its neighbors by adverting high quality routes or low latency links. Attacks of the hello flood type consist of an attacker with a high power antenna convincing a large number of other nodes that he is its neighbor [8]. This attacks works as a broadcast wormhole, convincing nodes to send their packets to nowhere. Routing protocols dependent on localized information are vulnerable to this class of attack. Proposed solutions are to employ mechanisms that employ verification of the bi-directionally of links, such as with the Needham-Schroeder Symmetric Key Protocol [20], and limitation of the number of verified neighbors by the base station. Finally, in the acknowledgment spoofing attack an attacker spoofs a bad link or a dead node using the link layer acknowledgment for the packets it overhears for those nodes. The solution to this threat is to employ authentication and encryption of all transmitted packets, including of packet headers.

In general, malicious packets at the network layer can be detected by using appropriate authentication mechanisms, and in the same context message freshness can provide protection against replayed messages. Attacks can be perpetrated against network-layer communications but also against routing protocols. Multipath routing techniques have been proposed in [16] against WSN routing attacks. The basic idea behind multipath routing is to use multiple disjoint paths to route a message such that it is unlikely that all wireless sensor nodes in the path are compromised. Another approach is to use secure localization determination mechanisms in securing geographic routing protocols. Secure localization mechanisms also allow the detection of nodes compromised by wormhole and Sybil attacks.

Other than the previously discussed proposals to target security attacks at the network and routing layers in WSN environments, the integration of WSN with the Internet via WSN Internet communication technologies will also call for mechanisms designed to work in tandem with such technologies. Mechanisms designed in this context can provide resistance against DoS attacks, injection of malicious routing information, replay of routing messages and node capture, among other security threats. New routing protocols may also be designed to be data-centric, and currently there seems to be no such protocol available supporting security mechanisms. On the other hand, proposals for ad hoc networks such as Ariadne [21] are usually considered too heavy for WSN environments.

2.2.4 PROPOSALS AT THE TRANSPORT LAYER

The goal of the transport layer is to manage end-to-end connections for different applications in the network. Research proposals of transport layer protocols for isolated WSN environments consists in simple solutions designed to cope with the limitations of WSN environments, as is the case with the Sensor Transmission Control Protocol (STCP) [22]. The two types of attacks targeted at the transport layer in WSN are flooding and de-synchronization, as we proceed to discuss.

The flooding attack seeks to exhaust the memory resources of victim sensor node, by sending too many connection requests or half open and half close transport-layer network segments. The employment of IP on sensor networks can also possibly potentiate the

appearance of this class of attack on WSN. One classic research solution to this threat is to employ client puzzles [8], which consists on clients showing their commitment to the resources they require before being authorized to communicate. Regarding the constraints of WSN environments, this approach presents the disadvantage of forcing legitimate nodes to also spend more resources.

On the other hand, in de-synchronization attacks the adversary forges one or both ends of a transport-layer connection using different sequence numbers on the packets. Authentication of packets and of the related control fields, together with and client puzzles [9], have been proposed against this threat.

The integration of WSN with the Internet will also facilitate the employment of new transport-layer communication protocols, which may support end-to-end communications between sensing devices and also with external or Internet devices. The addressing of security in the context of such transport-layer protocols is currently motivating numerous research efforts, as we discuss later in the context of our SoA on security proposals for Internet-integrated WSN.

2.2.5 CROSS-LAYER THREATS AND SECURITY APPROACHES

Various threats and proposals regarding security in WSN environments do not relate to a specific protocol layer, and may also apply or operate in a cross-layer fashion, as we proceed to discuss. One classic attack in this class is the Sybil Attack [23], which consists of a malicious node taking on multiple identities. Using such identities the attacker is able to impersonate legitimate nodes in the network and to simultaneously compromise the normal functioning of communication mechanisms at the various protocol layers. The Sybil attack can also affect different protocols and fault tolerant schemes, such as distributed storage, multipath routing, topology maintenance, data aggregation, voting, fair resource allocation and misbehavior detection techniques, among others.

Due to the complexity of the Sybil attack, currently there is no completely secure solution to circumvent it. Possible mechanisms against Sybil attacks include the registration of node identities at a central base station for validation purposes, secure localization verification techniques to detect compromised nodes, limitation of the number of verified neighbors per node using key pre-distribution techniques and radio resource testing assuming that each sensor uses only one radio interface. All such approaches present limitations due to the particularities of the various implementations and devices. For example, many WSN require mobility and sensing devices may in practice employ more than one radio interface. Also, mechanisms based on particular key distribution techniques may not always be easy to implement in practice.

Other than protection against threats with the characteristics of the Sybil attack, a few important security aspects in practice require the employment of cross-layer approaches. Important goals in this context include key management, broadcast and multicast

authentication, reputation management, security in data aggregation, secure time management and intrusion detection, as we discuss next. We also analyze how such aspects can be addressed in the context of Internet-integrated WSN, in our discussion later in the chapter.

2.2.5.1 Key management

Key distribution is a fundamental aspect of the effective support of security in any communications environments, the same certainly applying to WSN. Key management is fundamental because it enables the initial negotiation of cryptographic keys, and also its periodic renewal as appropriate for the maintenance of long-term security.

From the start, key management was a hot research topic regarding security for WSN environments. Numerous key management solutions have been proposed and are extensively analyzed in the literature [24][25][26][27]. One limitation of such approaches is that the proposed solutions do not provide resistance against physical node capture. This aspect implies that, for proposals using pair-wise keys, the capturing of a small number of nodes may suffice to compromise the key management protocol itself. We also verify that most of the key management protocols proposed are not resistant against an attacker that observes the initial key discovery process and uses the gathered information to attack sensing nodes in the network, as demonstrated in [28]. One particularly interesting proposal in classic approaches to key management in WSN is that of random key pre-distribution protocols. In general, the proposed algorithms could benefit from further research to enable its improvement in terms of scalability, resilience to node compromise, memory requirements and communications overhead, as discussed in [29].

Regarding the proposed approaches to implement key management in WSN, network shared keying can be considered insecure and as such does not provide an acceptable level of security. The configuration of key pairs at the link level between any two communicating nodes is also a very limited solution, because of its lack of scalability. Cryptographic keys can also be configured via the base station or sink node, in which case each node establishes and shares a key with the sink node. The disadvantage of this approach is the exposure of the base station as a single point of failure. This problem can be somehow alleviated by the usage of tamper-resistant hardware to store the keying material on the sink node.

The feasibility of employing public-key cryptographic mechanisms on WSN has never been consensual and still motivates numerous research efforts in the area. Although many consider public-key cryptography to be unfeasible in such environments due to limited computational and energy resources of wireless sensing devices, there are some preliminary results that defend otherwise [30]. As we explore later in the thesis, acceptable compromises in this situation may be established by delegating part of the effort required to support public-key cryptography to devices with less resource constraints.

Research in key management solutions for WSN is still ongoing, and many strategies may be followed in this context. One is to design better random key pre-distribution schemes that are able to resist to node compromise. Regarding public-key cryptography, hardware support could be integrated in new wireless sensing platforms to improve its effectiveness, similarly to existing platforms providing support for AES/CCM cryptography at the hardware. Alternative methods of application of public key cryptography can also be studied in order to conclude on the effective applicability of this technology to WSN, as discussed in [30][31]. Other approach is to design mechanisms that use asymmetric cryptography protocols where most of the burden falls on the base station or sink node, rather than on the sensing devices. Also in the context of key management, alternative schemes as Elliptic Curve Cryptography (ECC) currently deserve attention, as we discuss later.

Other than the previous proposals on key management for isolated WSN environments, it will remain a major aspect to be addressed in the context of Internet-integrated WSN. In this scenario, new mechanisms can be developed not only to cope with the limitations and particularities of WSN environments, but also to be compatible with Internet key management approaches and Internet communication technologies developed for WSN environments. Key management can be supported by mechanisms employed in wireless sensor nodes and on the Internet devices they communicate with, or in alternative can be partially or fully delegated to more powerful devices, similarly to our approach to transport-layer security described later in the thesis.

2.2.5.2 Broadcast and multicast authentication

Broadcast and multicast are essential operations in the majority of WSN deployments, and as such authentication of such communications is an important requirement. The main problem here is the implementation of sender authentication. Broadcast communications may also be incompatible with end-to-end security, since it requires cryptographic keys established for each pair of devices. The usage of a network wide shared key configured at the link layer simplifies key setup and supports broadcast, but with this approach intermediate nodes may easily eavesdrop or alter messages.

Previous research proposals targeting broadcast and multicast authentication include the delayed key disclosure and one-way function key chains techniques. One protocol that employs these techniques for secure broadcast authentication is µTesla [32], a variant of the TESLA (Timed, Efficient, Streaming, Loss-tolerant Authentication) protocol. The µTesla is part of SPINS [33], which also implements SNEP (Secure Network Encryption Protocol) to support data confidentiality, two-party data authentication and data freshness. We may observe that broadcast and multicast security remains a research issue with more recent communication technologies enabling the integration of WSN with the Internet. This applies to recent proposals on communication technologies proposed for upper layers of the communications stack, as in the current research efforts targeting security for multicast communications at the application-layer using the CoAP protocol [34], as we discuss later in the chapter.

2.2.5.3 Reputation assignment schemes

Centralized reputation systems were made popular by the internet, but the model considered to be the most suited for WSN environments in previous research approaches is that of a decentralized method such as the CORE reputation system [35] or the CONFIDANT protocol [36]. Both approaches propose that sensing devices supports a watchdog module to monitor the forwarding rate of its neighbors. If a neighbor node is found to not forward a message its reputation consequently decreases, and this information is propagated throughout the network. Each node also uses reputation information from other nodes in order to determine the overall reputation of a particular device. The main goal in such proposals is thus that, over time, less trusted nodes are not used to form reliable paths for routing purposes.

The two previous research proposals differ in how they use reputation information from other nodes, in how they punish bad behavior and how they attribute trust to nodes that temporarily misbehave. A high level framework for trust and reputation management in sensor networks is proposed in [37], while the authors do not propose any specific mechanism to manage reputation in a WSN. In this proposal, the authors simply suggest the employment of a watchdog mechanism to compute the reputation of each node, and state that the design of such a mechanism is dependent on the practical application of the WSN.

Researchers working on reputation assignment schemes for WSN environments have addressed two main challenges. One is that many of the existing proposals in this context simply assume that the information available about an entity of the network is correct, which can clearly be wrong in many WSN environments being targeted by numerous types of attacks. On the other hand, research proposals employing watchdog mechanisms such as in [35][36] may be incompatible with more dynamic applications like mobility tracking. Such proposals could evolve to support better such applications, for example by taking into account and correlating the signals strengths of the various neighboring nodes in order to be able to cope with mobility and adjust accordingly.

Considering that Internet-integrated WSN may enable the employment of wireless sensing devices in the context of global and distributed IoT sensing applications, it is fair to consider that research for reputation assignment schemes will find new ground in such environments, and that cross-layer security mechanisms may be designed for wireless sensing devices and interconnection gateways in order to implement new trust management approaches in the future.

2.2.5.4 Data aggregation Protocols

Most of the original WSN applications employed data aggregation mechanisms as a vital feature, which also provided new opportunities to security attacks. In respect to the compromise of the normal operation of data aggregation, an attacker may inject faulty data into the network, which may result in corrupted aggregated information and consequently in

the compromise the overall goal of the application. Examples of applications depending on data aggregation mechanisms are object tracking as discussed in [38] and directed diffusion routing [39].

Research proposals to implement security in the context of data aggregation operations include the employment of statistical properties to filter non-relevant data and reduce the effects of attacks on the aggregation process [40]. In [41] the authors propose the employment of secure hierarchies of node clusters in data aggregation operations, using cryptographic keys at each level of the hierarchy to secure communications between nodes in each cluster. The solution presented in [42] also uses cryptographic keys with the same purpose. Another proposed protocol to secure data aggregation is the Secure Information Aggregation (SIA) Protocol [5], which works under appropriate trust assumptions, randomly sampling a small fraction of nodes and checking that they have behaved properly to detect several types of attacks. Future research in this context can also address the employment of reputation systems to secure data aggregation in WSN. An alternative approach would be to develop statistical methods for estimation of sensing data that are robust against the corruption of results by attackers injecting false data.

2.2.5.5 Time synchronization protocols

The importance of time synchronization protocols in the support of security operations is well known, for example in supporting authentication mechanisms and protection against various types of message replay attacks. In this context, various protocols have been proposed in the literature for WSN. Well known proposals in this class are the Reference Broadcast Synchronization (RBS) [6] protocol, the Timing-sync Protocol for Sensor Networks (TPSN) [43] and the Flooding Time Synchronization Protocol (FTSP) [44]. None of such proposals have been designed with security in mind, in fact assuming to operate in a trusted environment. This implies that by capturing some of the sensing devices an adversary would be able to easily inject false synchronization messages and disrupt the normal operations of time synchronization. Time-synchronization is also important for several sensor networks applications, as for example in state estimation for position tracking.

As for the previous proposals targeting isolated WSN environments, we may expect that time synchronization will play a fundamental role in achieving security with Internet-integrated WSN. Authentication and protection against packet relay attacks will also be an issue in such environments, particularly in what regards the employment of end-to-end communications with Internet devices. Research may also target the design of time synchronization protocols that operate with other mechanisms in a cross-layer fashion, such as key management.

2.2.5.6 Intrusion detection

Despite the importance of Intrusion Detection Systems (IDS) for the security of traditional computing environments, research on IDS systems for WSN environments is relatively scarce and recent. We may argue that security will never be complete in WSN without the employment of appropriate failure detection and recovery mechanisms. Failure recovery can allow extending the lifetime of a WSN by restarting or reprogramming failed or misbehaving nodes, or by circumventing affected areas. This may also apply to Internet-integrated WSN, for which intrusion detection can help in identifying and reacting to external threats and attacks, for example regarding the data transported by end-to-end communications with WSN sensing devices.

Authors in [45] present some interesting reflection about the dual purpose of detecting and recovering from a node compromise. They also discuss how to implement such recovery mechanisms in wireless sensing devices. In [46] the authors discuss why intrusion detection solutions proposed for ad hoc networks are not appropriate to WSN environments. They also introduce general guidelines for the application of IDS architectures and techniques to sensor networks without mobility. In this work the authors propose the employment of spontaneous watchdog mechanisms in sensing devices, which enable them to actively monitor their neighbors.

Another approach considered in research is to decentralize IDS systems for WSN environments. Similar architectures are presented in [47] and [48], where sensing devices monitor their neighbors without collaboration between monitoring devices. These research proposals also discuss on how buffer size is relevant for the accomplishment of the monitoring tasks, while not specifying how the system should operate in detail, instead focusing only on the proposal of algorithms to detect specific security attacks. In [49] the authors approach the problem of modeling traffic in a WSN in order to test if IDS techniques are applicable to specific environments.

Intrusion detection can also play an important part in the context of WSN integrated with the Internet. End-to-end communications between sensing devices and external or Internet hosts may open a plethora of new threats, which may originate both inside or outside the WSN domain. For example, intrusion detection systems and techniques can be extended to interpret new communications based on 6LoWPAN, or on the other hand new detection techniques can be developed that are appropriate and optimized WSN wireless communication environments.

2.2.6 SECURITY ARCHITECTURES FOR NON-INTERNET WSN ENVIRONMENTS

A lesson learned from the Internet is that security may be efficiently supported in the context of an appropriately designed architecture supporting various complementary security mechanisms. Security architectures may also enable the enforcement of security policies using appropriate mechanisms designed to ensure fundamental security properties

such as confidentiality, integrity, freshness, availability, and authentication, among others. In our following discussion we analyze proposals on architectures for WSN environments. These architectures represent either closed or previous proposals on security architectures for WSN environments not focused on the support of Internet communication technologies.

2.2.6.1 ZigBee

The ZigBee [50], [51] specification provides a suite of communication protocols for wireless radios based on 802.15.4, particularly by designing mechanisms for the network and application layers, and also a security protocol with support for application profiles. At the network layer, ZigBee supports mechanisms that allow nodes to join and leave the network, the application of security to messages, and the routing of packets towards their destination. ZigBee also supports key exchange and authentication security mechanisms, two important components of its security protocol.

The specification provides three security levels in par with the security mechanisms available at the IEEE 802.15.4 MAC. Communications may employ no security, security via access control lists or 32-bit to 128-bit Advanced Encryption Standard (AES) encryption with authentication. ZigBee also supports a set of security services that include key establishment, key transport, frame protection and device management [50]. Frame protection is achieved through data freshness, message integrity, authentication and encryption. In particular, data freshness is achieved using incoming and outgoing counters, and message integrity employs 32, 64 or 128-bit cryptographic keys. Authentication is available at the network layer using a shared network key or at link level using pairwise keys. Network shared keys offers security against outsider attacks, while pairwise keys offer additional security against insider attacks, at an added extra cost in terms of resources on sensing devices. Encryption can also be turned off without sacrificing freshness, integrity or authentication by using appropriate Message Authentication Codes (MAC). Broadcast communications may be protected using a 128-bit network key that is distributed to nodes when they join the network, while unicast communications are protected using pairwise link keys.

The ZigBee specification defines also services for the establishment and maintenance of security relationships at the Application Support Sub-layer (APS), allowing systems designers to select the appropriate level of security for the application at hand. The Zigbee Device Object (ZDO) is another component of the architecture, and is responsible of managing security policies and configuration of devices. The 2007 specification of the ZigBee architecture also defines the usage of the Symmetric-Key Key Exchange (SKKE) Protocol. This protocol allows the secure establishment of link keys between devices. In respect to the evolution of this specification, we are also able to verify that ZigBee is more recently adopting IP [52].

2.2.6.2 SPINS

SPINS [33] is a suite of security protocols optimized for WSN employing two security building blocks, the Secure Network Encryption Protocol (SNEP) and µTESLA [32], which were evaluated using the TinyOS [53] operating system. SNEP is used to provide confidentiality (through encryption and authentication) and data freshness, while µTESLA supports broadcast authentication. SNEP provides a number of security and functional properties, namely low overhead, data authentication, replay protection and weak freshness. Using SNEP, communicating nodes must share a secret master key, from which independent keys are derived when necessary for encryption and authentication operations. Data authentication is achieved through the use of a Message Authentication Code (MAC), together with the use of a counter to provide replay protection and weak freshness. Strong freshness is also provided if both parties agree on authenticating the data packets exchanged after the generation of a nonce value by one side of the communications channel.

The µTESLA [32] Protocol is a smaller version of the Timed Efficient Stream Loss-tolerant Authentication (TESLA) Protocol [54], also proposed by Perrig *et al*. This protocol employs delayed disclosure of symmetric keys in order to emulate asymmetric cryptography, and serves as the broadcast authentication service of SNEP. Some limitations can be identified for SPINS. One is that µTESLA requires strict synchronization between each node and the base station, which in some deployment situations cannot be easily guaranteed. Limitations have also been shown in the flexibility of symmetric key exchange mechanisms using key disclosure techniques as a result of their energy efficiency. Other problem is that mechanisms based on delayed key disclosure are easily targeted with DoS attacks, leading to buffer overflow and battery exhaustion as fake messages are exchanged among the nodes. Finally, Karlof *et al*. state that SNEP was unfortunately neither fully specified nor fully implemented [55], a factor that also motivated the design of TinySec, that we discuss next.

2.2.6.3 TinySec

The design of TinySec [55] was motivated by the unfinished state of SNEP, as previously discussed. TinySec proposed the employment of security extensions to the packet protocol of the TinyOS [56] operating system, and its security mechanisms are in practice similar to the ones implemented by SNEP [33]. TinySec supports access control, message integrity through authentication and confidentiality through encryption. Semantic security is also assured through the use of a unique nonce or Initialization Vector (IV) value in each invocation of the encryption algorithm. The protocol doesn't implement replay protection, due to the limited amount of state information maintained by each recipient. The authors also defend that replay protection belongs in the higher layers of the stack, and not at the link-layer [55].

The TinySec proposal defines two security modes, the TinySec-Auth mode for authentication only and the TinySec-AE for authentication and encryption [55]. TinySec employs CBC-MAC codes for authentication in both security modes using a 4-byte MAC, considerably smaller than the usually employed 8 or 16-byte MAC. The TinySec authors argue that, in the context of sensor networks this does not represent a security problem. Encryption is supported by the Skipjack algorithm, which employs a variant of the CBC mode of block cipher operation. The Skipjack algorithm preceded the AES standard adopted by IEEE 802.15.4.

One major limitation of the TinySec security architecture is that it does not specify a key management protocol, and thus it is up to the application to choose and support the keying solution considered to be appropriate. Despite various limitations, TinySec provided a significant contribution to security in WSN, and was one of initial research proposals proving that efficient and secure communications in WSN are indeed possible. TinySec has also been used as the secure link layer basis in a number of research and commercial projects in the area of WSN.

When comparing the previously discussed security architecture, we are able to observe shared characteristics in terms of the security functionalities supported by ZigBee, SPINS and TinySec. For example, Zigbee and SPINS provide data freshness for communications, and the three proposals provide authentication via CBC MAC codes. Confidentiality (through encryption) is optional in TinySec (using AES in the CBC mode) and mandatory in ZigBee (using AES) and SPINS (with AES in CTR mode). The block cypher is AES with 128-bit keys for ZigBee, RC5 for SPINS and Skipjack for TinySec. Regarding key management, TinySec provides no solution, SPINS supports delayed disclosure and master keys, and in ZigBee it is supported by the SKA Trust Center. ZigBee is undoubtedly the most complete architecture from the point of view of the security mechanisms supported, as well as related to its support of application and security profiles.

We finally note that, although IEEE 802.15.4 could be considered in the previous analysis, we defer its discussion until later in the chapter, given its significance in the context of the communication technologies currently being designed to support of the integration of WSN with the Internet.

2.3 PROPOSALS ON SECURITY FOR INTERNET-INTEGRATED WSN ENVIRONMENTS

In our following discussion we proceed by analysing how WSN may be integrated with the Internet communications infrastructure, by using Internet communication technologies currently being designed for WSN environments. Such technologies are being developed to operate on top of the IEEE 802.15.4 PHY and MAC, and in practice provide the basis for the design of the research solutions described throughout the thesis. We also analyse the security solutions and mechanisms currently available to protect communications using such technologies. In our following analysis we start by identifying the technologies that contribute to this integration approach.

2.3.1 INTEGRATION SUPPORT TECHNOLOGIES

The integration of WSN with the Internet is currently being enabled by technologies being designed to support Internet communications on low-power wireless personal area networks (LoWPAN) environments such as WSN. In our following classification, WSN communications belong to the context of capillary communications, in the sense that they support the final hop in the communications path towards the physical sensing and actuating devices. Other communication technologies are already available or are currently being designed that may be part of a future IoT communications architecture, and in this context that may support the integration of WSN with the global communications infrastructure. Figure 2.1 illustrates such technologies, which are categorized as backbone, backhaul and capillary communications technologies. In the same figure we also illustrate possible interactions between communication technologies at different categories, which may be supported by devices implementing translation mechanisms (for example specialized devices or gateways supporting the interconnection of different communication domains) or supporting various communication technologies in simultaneous [57].

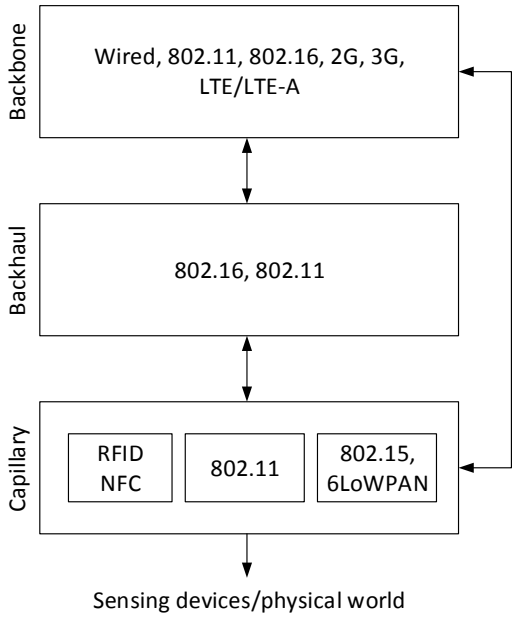

Figure 2.1 - Categories of communication technologies for IoT applications

While recognizing the subjectivity of the classification illustrated in the previous figure, we note that its main purpose is to enable the contextualization of the WSN communication technologies that provide the experimental ground for the evaluation of the research proposals described in the thesis. As we observe later, IEEE 802.15.4 assumes a particularly relevant role in the context of Internet-integrated WSN. In our following discussion we analyze the various technologies considered in the classification illustrated in Figure 2.1.

2.3.1.1 Backbone communication technologies

As in the current Internet communications infrastructure, backbone communications can be supported by both wired and wireless communication technologies. Wired communication technologies may include IEEE 802.3 [58] Ethernet-based communications, as well as other technologies such as Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) [59] fiber optic-based communication technologies. A particularly important role in the context of backbone communications will be also played by broadband wireless communication technologies, considering the increasing adoption of mobile devices requiring pervasive broadband Internet communications. In this context, the technologies may include second-generation GSM [9] from ETSI (European Telecommunications Standard Institute) [60], third generation UMTS (Universal Mobile Telecommunication System) [61] from 3GPP (3rd Generation Partnership Project) [62] and fourth generation LTE (Long Term Evolution) and LTE-A (LTE Advanced) [63] also from 3GPP.

We may also consider the employment of IEEE 802.11 [64] and IEEE 802.16 [65] technologies to support wireless backbone communications. IEEE 802.11 [64] provides communications focused mostly on wireless local area network (WLAN) applications, but may also support applications designed for larger geographical areas, and the same reasoning may apply also to IEEE 802.16 [65] WiMax (Worldwide Interoperability for Microwave Access), which targets wireless metropolitan area network (WMAN) applications.

2.3.1.2 Backhaul communication technologies

Backhaul communication technologies support communications between the capillary and backbone communication domains, and also provide a bridge between different capillary communication domains and technologies. As in other categories, the backhaul communication technologies employed may depend on factors such as the geographical coverage of applications, its communication requirements and the types of devices employed.

As we illustrate in Figure 2.1, the candidate technologies to support backhaul communications on the IoT may include WLAN IEEE 802.11 [64] and WMAN 802.16 [65] WiMax. Such technologies may provide adequate geographical coverage for distributed sensing applications employing multiple capillary domains, and also support communications between such capillary domains and the global Internet communications infrastructure. Internet-integrated WSN are a form of capillary domain, which may be enabled by IEEE 802.15.4-based Internet communication technologies, as discussed next.

2.3.1.3 Capillary communication technologies

The communication technologies identified at the capillary category may support the final hop in the communications path towards the sensing/actuating devices interfacing with the

physical world. This is also the context of application of WSN, which may be progressively integrated with the Internet via the usage of low-power Internet communication technologies optimized for such environments. In this context, of particular interest is IEEE 802.15.4, as we consider throughout the thesis, since it provides the support for the design of Internet communication technologies for WSN. WSN are in general able to facilitate the enabling of data-collection systems using constrained low-power autonomous wireless sensing devices, and the integration of such systems with the Internet communications infrastructure promises to dramatically improve the usefulness and pervasiveness of sensorial applications.

Other technologies are also expected to play an important part in the capillary category, for example Radio Frequency Identification (RFID), which is becoming widely used for authentication and goods tracking, and Near-Field Communications (NFC), which are increasingly being adopted to support applications such as contactless payments and ticketing, among others.

Other than RFID and NFC, we may identify two main classes of capillary IoT communication technologies, as illustrated in Figure 2.1. IEEE 802.11-based WLAN may support communications with less constrained devices, for example embedded devices with continuous power sources, smartphones or devices supporting 802.11 side-by-side with low-power wireless communications [57]. 802.11 is also being optimized to support low-power wireless communications in 802.11ah [66], which will support sub 1-Ghz communications for sensor network and smart metering applications. It is thus possible that future versions of the 802.11 standard may include support for applications using devices with characteristics similar to current wireless sensing platforms employed in WSN.

Finally, IEEE 802.15 [67] provides communication technologies for LoWPAN environments, as is currently the most representative approach to support Internet communications designed for WSN environments integrated with the Internet. In its context, of particular interest are IEEE 802.15.6 [68] and IEEE 802.15.4 [51]. The former is designed to support wireless body area networks (WBAN) applications, while IEEE 802.15.4 supports low-power and short-range wireless communications as employed in WSN environments. The IEEE 802.15.4 physical (PHY) and medium access control (MAC) communications provide the ground for the design of Internet communications and security protocols for WSN, as we proceed to discuss.

2.3.2 A Protocol Stack for Internet-integrated WSN

Communication technologies are currently being designed for constrained WSN environments that promise to enable the integration of WSN with the Internet communications infrastructure. Thus, in this integration scenario WSN devices are able to communicate directly with external or Internet entities, at diverse protocol layers. This vision is currently becoming a reality thanks to communication technologies developed

based on the 6LoWPAN [69][70][71] adaptation layer. The employment of Internet communication technologies on WSN environments can also require the evolution of existing Internet security mechanisms and solutions to encompass WSN environments and sensing devices, an aspect that deeply motivates the research efforts described throughout the thesis.

The communication technologies currently being designed to enable Internet communications on WSN environments are also a result of efforts from working groups of organizations such as the Internet Engineering Task Force (IETF). Also relevant are efforts conducted in the context of the ETSI Technical Committee on M2M communications [60], which is working to develop an end-to-end high-level architecture for M2M and also standards fulfilling the gaps where other standards bodies or groups are unable to do so. The ITU-T (Telecommunication Standardization Sector of the International Telecommunication Union) [72] is working on recommendations related to USN (Ubiquitous Sensor Networks) and NGN (next generation networks), with the goal of designing a conceptual network built over existing physical networks, which provides knowledge services by making use of sensorial data.

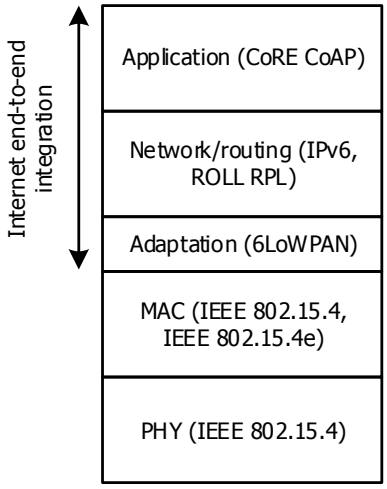

Figure 2.2 - A standardized protocol stack for the Internet of Things [73]

Internet communication technologies for WSN are being designed accordingly to the constraints and characteristics of low-energy sensing devices and low-rate wireless communications that are typical of such environments. Although such characteristics have also influenced previous designs of applications employing WSN isolated from the Internet, the new solutions are being designed to guarantee interoperability with existing Internet standards and guarantee that sensing devices are able to communicate with other Internet entities in the context of future IoT distributed applications. The communication protocols available or currently being designed with this purpose already enable a reference protocol

stack for the employment of Internet communication technologies [73], which is illustrated in Figure 2.2.

The communication technologies at particular layers of the protocol stack illustrated in Figure 2.2 are designed to be appropriate to the employment of low-energy devices and wireless communications, while providing acceptable reliability and not compromising the lifetime of sensing applications. As previously discussed, many sensing devices are powered by batteries and, in consequence, new communication and security solutions developed for WSN environments are required to carefully balance the communications rate, reliability and energy usage. From a bottom-up perspective, the following are the main characteristics of the various standard protocols forming the stack illustrated in Figure 2.2:

- Low-energy communications at the physical (PHY) and Medium Access Control (MAC) layers may be supported by IEEE 802.15.4 [74], including more recent addendums to the standard as IEEE 802.15.4e [75]. IEEE 802.15.4 sets the rules for communications at the lower layers and lays the ground for the development of WSN Internet communication technologies at higher layers of the stack.

- Low-energy communication environments using IEEE 802.15.4 support at most 102 bytes for the transmission of data at higher layers of the stack, a value much less than the maximum transmission unit (MTU) of 1280 bytes required for IPv6. Addressing this issue, 6LoWPAN [69][70][71] provides an adaptation layer for the transmission of IPv6 packets over IEEE 802.15.4, by implementing fragmentation and reassembly of IPv6 packets, among other required mechanisms, as we detail later.

- Routing over 6LoWPAN WSN environments may be supported by the Routing Protocol for Low-power and Lossy Networks (RPL) [76]. RPL provides a framework that may be adapted to the requirements of particular applications. Application-specific profiles are defined to identify the corresponding routing requirements and optimization goals.

- The Constrained Application Protocol (CoAP) [34] supports communications at the application layer. CoAP is currently being designed to provide interoperability at the application layer, in conformance with the REST architecture prevalent on the web.

The communication technologies forming the protocol stack of Figure 2.2 enable communications between wireless sensing devices and external hosts at the various protocol layers. Such technologies may enable more full integration approaches, and also motivate new challenges in respect to the fulfillment of appropriate security guarantees in the context of Internet-integrated WSN. This aspect raises the question of what mechanisms are available to guarantee security in the presence of end-to-end communications using the technologies illustrated in Figure 2.2.

The complexity of protecting WSN domains in such more full integration approaches is also related to the fact that end-to-end communications with sensing devices may take place

from the network-layer up, and as such WSN devices and communications may be more open to a plethora of threats and attacks originated at external communication environments or the Internet. As most WSN devices are expected to remain constrained, end-to-end communication and security technologies must be employed parsimoniously, and one possible strategy in this context is to complement such mechanisms with appropriate security mechanisms supported by more resourceful devices, as we consider later. On the other hand, end-to-end security is in reality only part of the problem, as many security aspects may require appropriate cross-layer approaches. Later in the chapter we discuss such aspects in greater detail, together with the employment of 6LoWPAN-based communication technologies in WSN environments.

The gradual adoption of 6LoWPAN-based communication technologies for WSN is also visible in existing commercial offerings. For example, the popular ZigBee-2006 [50] specification is evolving to adopt the ZigBee IP stack [52], which also provides support for 6LoWPAN, RPL [76] and CoAP [34]. Despite the adoption of a networking stack oriented towards 6LoWPAN, we must observe that ZigBee remains a commercial and closed specification, in the sense that communications related with ZigBee applications that are transported over the Internet remain restricted to such applications. Other proposals such as those from Sensinode [77] also adopt IP-based 6LoWPAN communication technologies. Sensinode currently offers the NanoStack 6LoWPAN protocol stack and the NanoRouter platform, which supports applications requiring 6LoWPAN-Internet routing infrastructures.

We may also observe that a few research proposals have contributed to the idea of integrating WSN with the Internet via the Internet communication technologies developed for such environments. As in other research proposals, the exploratory nature of such works justifies the absence of appropriate security solutions in such proposals. The research proposals in [78] [79] do not address security, instead focusing on the intelligent placement of gateways in order to reduce data latency in scalable and sustainable WSN deployments. A few initial research proposals also focus on the integration of WSN with the Internet via web services, in particular exploring the usage of web services directly on constrained sensing devices. As in the previous works, we also observe the lack of appropriate security solutions in such proposals. In [80] and [81] the authors focus only on the communication aspects of the integration. In the StreamFeeds proposal [82] the authors discuss that applications may be able to inherit security mechanisms supporting authentication and privacy services from the web services technology employed in the Internet, while doesn't specifying how this may be achieved in practice. In conclusion, given the preliminary nature of such research proposals, security is either absent or mostly undefined.

The integration of WSN with the Internet via Internet WSN communication technologies was initially addressed in the design of mechanisms to enable communications with web services running directly on constrained WSN sensing devices [80][81]. Such proposals thus represent an evolution and complement other proposals focused on the integration via front-end gateways, which we discuss later. The employment of web services on constrained sensing

devices was initially proposed and evaluated in [80], and in [81] the authors describe a RESTful web service architecture allowing external servers to communicate directly over TCP with IP-enabled sensor devices using web services. The architecture described in this proposal employs a session-aware power-saving MAC protocol running over X-MAC [83], which synchronizes wake-up periods of devices with TCP control messages, and on the employment of the HTTP (Hypertext Transfer Protocol) conditional mechanisms to avoid the transmission of non-changing data from the server to the client. Stream Feeds [82] identifies streams of data from sensing devices using URLs that may be hyperlinked to other objects on the web, thus enabling such streams to be indexed by search engines.

In our following discussion we analyze the communication technologies identified in Figure 2.2, and also the security technologies and approaches that may be considered in the context of its employment. This analysis also serves our purpose of identify the currently open issues regarding security, which motivate the research solutions described and evaluated later in the thesis.

2.3.3 PHY AND MAC COMMUNICATIONS AND SECURITY

The IEEE produces standards to facilitate a common platform of rules for new technological developments. This is also the goal of the IEEE 802.15.4 standard [74], which is designed to implement a healthy trade-off between energy-efficiency, range and data rate of communications. IEEE 802.15.4 supports low-energy communications at the physical (PHY) and Medium Access Control (MAC) layers, with a short-range of roundly 10 meters at 250kbit/s.

The original IEEE 802.15.4 standard from 2006 was recently updated in 2011, mainly to include a discussion on the market applicability and practical deployments of the standard. Other amendments were recently introduced for the standard, namely IEEE 802.15.4a [84] specifying additional PHY layers, IEEE 802.15.4c [84] to support recently opened frequency bands in China and IEEE 802.15.4d [85] with a similar goal for Japan. Also of particular interest is the IEEE 802.15.4e [75] addendum, which defines modifications to the MAC layer to support time-bounded multi-hop communications. In the following description we begin by discussing how communications operate using IEEE 802.15.4 and IEEE 802.15.4e, and later we address security is its context.

2.3.3.1 IEEE 802.15.4-2011 PHY

Due to its suitability to low-energy wireless communication environments, IEEE 802.15.4 lays the ground for the design of standardized technologies such as 6LoWPAN and CoAP at higher layers of the stack, as previously illustrated in Figure 2.2. IEEE 802.15.4 was also previously adopted as the foundation of industrial WSN standards such as ZigBee-2006 [50], ZigBee PRO (2007) [50], ISA 100.11a [86] and WirelessHART [87]. ZigBee defines application profiles targeting market areas such as home automation and smart energy while, on the

other hand, WirelessHART and ISA 100.11a target the industrial automation and control market. The IEEE 802.15.4e addendum to the IEEE 802.15.4 standard was recently introduced to enable support for the critical environments supported by these industry specifications. Therefore, Internet communications can be employed in the future by such applications, which were in the past only supported by closed specifications.

The IEEE 802.15.4 PHY manages the physical Radio frequency (RF) transceiver of the sensing device, and also channel selection, energy and signal management. The standard supports 16 channels in the 2.4 GHz Industrial, Scientific and Medical (ISM) radio band. Reliability is introduced at the PHY by employing the Direct Sequence Spread Spectrum (DSSS), Direct Sequence Ultra-Wideband (UWB) and Chirp Spread Spectrum (CSS) modulation techniques. DSSS was introduced in the original 2006 version of the standard, while UWB and CSS were added later in 2007 by the IEEE 802.15.4a addendum. The main goal of such modulation techniques is to achieve reliability by transforming the information being transmitted, so that it occupies more bandwidth at a lower spectral power density. This allows the achievement of less interference along the frequency bands, together with an improved Signal to Noise (SNR) ratio at the receiver.

Regarding security, we observe that no mechanisms have been designed in the standard to operate in the context of PHY communications. On the other hand, IEEE 802.15.4 provides security services at the MAC layer, as we discuss next. MAC security services are available for upper layer protocols and are also employed by existing specifications as ZigBee, as we have previously discussed.

2.3.3.2 IEEE 802.15.4-2011 MAC

The IEEE 802.15.4 standard supports the transmission at the PHY of data frames occupying a maximum of 128 bytes. This limited size of the packets is due to the support of low-energy wireless communications environments, particularly to minimize the probability of errors taking place in such communications. The MAC layer manages, besides data transportation, other operations, such as accesses to the physical channel, network beaconing, validation of frames, guaranteed time slots, node association and security.

The IEEE 802.15.4 standard defines two main types of devices, a full-function device (FFD) and a reduced-function device (RFD). An FFD is a device that is able to coordinate a network of devices, while an RFD is only able to communicate with other RFD or FFD devices. By using RFD and FFD devices, IEEE 802.15.4 can support various types of topologies, such as peer-to-peer, star and cluster networks. IEEE 802.15.4 devices may be identified using either a 16-bit short identifier or a 64-bit IEEE EUI-64 [74] identifier. Short identifiers are usually employed in restricted environments, while larger identifiers are obtained from the IEEE EUI-64 identifier of devices. 6LoWPAN provides mechanisms to map standard Internet IPv6 addresses to 16-bit and 64-bit identifiers.

IEEE 802.15.4 defines four types of frames: data frames, acknowledgment frames, beacon frames and MAC command frames. Collisions during data communications are managed in the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access method or, in alternative, the coordinator may establish a super frame in the context of which applications with predefined bandwidth requirements may reserve and use one or more exclusive time slots. In this situation, beacon frames act as the limits of the super frame and provide synchronization to other devices, as well as configuration information.

2.3.3.3 802.15.4e multi-channel MAC

The employment of single-channel communications as enabled by the 2006 and 2011 versions of the IEEE 802.15.4 standard in practice delivers unpredictable communications in terms of reliability, particularly in multi-hop usage scenarios. As a consequence, IEEE 802.15.4 is not well suited to support applications with strict timing constraints, for example to support industrial monitoring and control applications. Applications in such areas are traditionally supported by specifications such as WirelessHART and ISA 100.11a, and IEEE 802.15.4 has been traditionally avoided in such environments. This situation is changing thanks to the design of the IEEE 802.15.4e [75] addendum to the standard.

The IEEE 802.15.4e approach to the problem of unreliability in multi-hop usage scenarios was originally proposed in the form of the Time Synchronized Mesh Protocol (TSMP) [88]. TSMP employs time synchronized frequency channel hopping to combat multipath fading and external interferences, and has also provided the technological foundation for WirelessHART communications. Considering its state as an addendum to the standard, the mechanisms introduced by IEEE 802.15.4e are expected to become part of the next revision of IEEE 802.15.4.

In IEEE 802.15.4e devices synchronize to a slot frame structure, a group of slots repeating over time. For each active slot, a schedule indicates with which neighbor a given device communicates with, and on which channel offset. Although IEEE 802.15.4e enables the definition of how the MAC layer executes a given schedule, it does not define how such a schedule is built. IEEE 802.15.4e channel hopping also requires synchronization between devices, which may be acknowledgment-based or frame-based. In the former, the receiver calculates the difference between the expected time of arrival of the frame and its actual time of arrival, and provides this information to the sender in the corresponding acknowledgment, thus enabling the sender to synchronize its clock to the clock of the receiver. In the latter, the receiver adjusts its own clock by the same difference, thus synchronizing to the clock of the sender. IEEE 802.15.4e also introduces a few modifications to the security services supported at the MAC layer, as we discuss later in the chapter.

2.3.3.4 Security in the IEEE 802.15.4-2011 standard

The IEEE 802.15.4 standard defines security services at the MAC layer, which can be employed to secure WSN link-layer communications. At the same time, such services are valuable in supporting security mechanisms designed for higher layers of the protocol stack illustrated in Figure 2.2. This cross-layer usage of security may be also promoted by the availability in most sensing platforms of efficient symmetric cryptography at the hardware using the Advanced Encryption Standard (AES) [89], as defined in IEEE 802.15.4. An example of this may be found in sensing platforms employing the cc2420 single-chip [90]. The security modes supported at the IEEE 802.15.4 MAC are described in Table 2.1. AES as employed by IEEE 802.15.4 uses 128-bit keys to support access control, confidentiality, data authenticity and replay protection.

Table 2.1 - Security modes at the IEEE 802.15.4 MAC

Security mode	Security properties supported at the MAC layer
No Security	No data encryption, no data authenticity validation
AES-CBC-MAC-32	No data encryption, data authenticity using a 32-bit MIC
AES-CBC-MAC-64	No data encryption, data authenticity using a 64-bit MIC
AES-CBC-MAC-128	No data encryption, data authenticity using a 128-bit MIC
AES-CTR	Data encrypted, without data authenticity
AES-CCM-32	Data encrypted, data authenticity using a 32-bit MIC
AES-CCM-64	Data encrypted, data authenticity using a 64-bit MIC
AES-CCM-128	Data encrypted, data authenticity using a 128-bit MIC

As may be observed in Table 2.1, security as supported at the IEEE 802.15.4 MAC is optional, given that in practice applications may opt for no security or for security at others layers of the protocol stack. Confidentiality is achieved by encrypting the transmitted data using AES in the Counter (CTR) mode, while data authenticity is achieved with encryption in the Cypher Block Chaining (CBC) mode to produce a Message Integrity Code (MIC) or Message Authentication Code (MAC) of variable size, which is appended to the transmitted data. IEEE 802.15.4 also defines the support of the CTR and CBC modes jointly using the combined Counter with CBC-MAC AES/CCM encryption mode. IEEE 802.15.4 platforms such as the TelosB usually support AES in the CCM* variant, which offers the added possibility of using security in the integrity-only and encryption-only modes.

The application of security to an IEEE 802.15.4 link-layer data frame is illustrated in Figure 2.3. As this figure illustrates, a protected frame is identified by the *Security Enabled Bit* field of the *Frame Control* field being set at the beginning of the IEEE 802.15.4 header. The

Auxiliary Security Header is employed only when security is used, and identifies how security is applied to the frame. In the *Auxiliary Security Header*, the *Security Control* field identifies the *Security Level* mode, according to the modes defined in the standard and described in Table 2.1, and how the cryptographic key required to process security for the link-layer frame is to be determined by the sender and receiver. This key may be known implicitly by the two communication parties, or on the other hand determined from information transported in the *Key Source* and *Key Index* subfields of the *Key Identifier* field. The *Key Source* subfield specifies the group key originator, while the *Key Index* subfield identifies a specific key from a specific *Key Source*. The *Frame Counter* field is set by the sender and transports a unique message identifier providing semantic security and message replay protection.

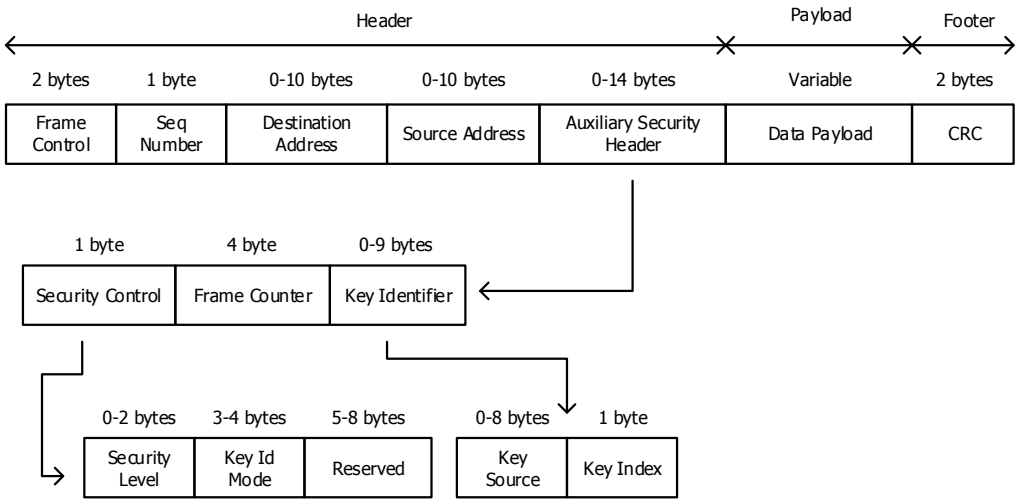

Figure 2.3 - Security data and control fields in IEEE 802.15.4

Depending on the security mode employed, the data payload can have three different configurations, as illustrated in Figure 2.4. In the AES-CTR security mode only confidentiality is provided and the encrypted payload contains a *Frame Counter* and *Key Control* fields. The *Frame Counter* sets the unique message ID and the key counter (*Key Control* field) is under the control of the application, which may increment it if the maximum value for the *Frame Counter* is reached. The sender breaks the original packet into 16-byte blocks, with each block identified by its own block counter. In order to support semantic security and replay protection, each block is encrypted using a different nonce or Initialization Vector (IV).

As illustrated in Figure 2.5, the Frame Counter and Key Counter fields, together with a static 1-byte *Flags* field, plus the sender's address and a 2-byte *Block Counter* field, constitute the IV. The *Block Counter* is not transmitted with the message, since the receiver can infer its value for each block. The IV is also employed for encryption with the modes based on AES/CCM. In security modes based on AES-CBC-MAC the unencrypted payload is followed by

a MAC code. This MAC is created encrypting information from the 802.15.4 MAC header and the data payload. As security modes based on AES-CCM provide confidentiality and data authenticity, they transport all the required fields plus the encrypted payload, as illustrated in Figure 2.4.

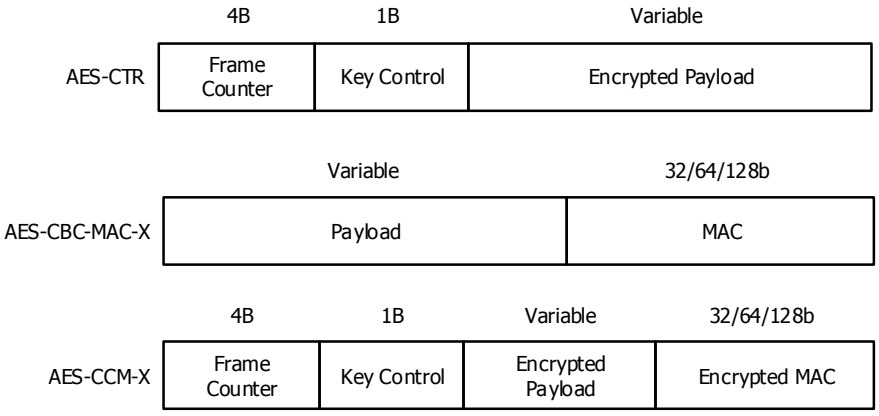

Figure 2.4 - Formatting of the payload data with IEEE 802.15.4 security

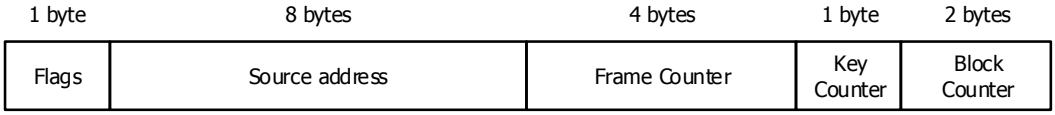

Figure 2.5 - Format of the Initialization Vector for AES-CTR and AES-CCM in IEEE 802.15.4

Other than confidentiality, data authenticity and replay protection, the IEEE 802.15.4 standard provides support for control of accesses. A sensing device may use the source and destination addresses of the frame to search for information on the security mode and security-related information required to process security for a given message. The 802.15.4 radio chips of the device stores access control lists (ACLs) with a maximum of 255 entries, each containing the information required for the processing of security for communications with a particular destination device. A default ACL entry may also be present, which is used to process security for packets not corresponding to specific ACL entry.

In Figure 2.6 we illustrate the format of an ACL entry as defined by IEEE 802.15.4. An ACL entry contains an IEEE 802.15.4 address, a *Security Suite* identifier field and security material required to process security for communications with the device identified in the *Address* field, namely the cryptographic *Key* and, for suites supporting encryption, the *Nonce* (IV) that must be preserved across different packet encryption invocations. When replay protection is active, the ACL also stores a high water mark of the most recently received packet's identifier in the *Replay Counter* field of the ACL entry.

Figure 2.6 - Format of an ACL entry in IEEE 802.15.4

The IEEE 802.15.4e [75] addendum to the standard introduces a few small modifications required to adapt MAC security mechanisms to time-synchronized channel-hopping communications. IEEE 802.15.4e defines the possibility of using null or 5-byte *Frame Counter* values, which in the latter case shall be set to the global Absolute Slot Number (ASN) of the network. The ASN stores the total number of timeslots that have elapsed since the start of the network and is beaconed by devices already connected to the network, thus allowing new devices to synchronize. The usage of the ASN as a global frame counter value allows for time-dependent security, replay protection and semantic security.

To enable the usage of a 5-byte *Frame Counter* value, IEEE 802.15.4e introduces modifications to the *Security Control* field illustrated in Figure 2.3, which in addition to the *Security Level* and the *Key Identifier Mode* now employs two bits from the reserved space, bit 5 to enable suppression of the *Frame Counter* field and bit 6 to distinguish between a *Frame Counter* field occupying 4 or 5 bytes. In consequence, the *Auxiliary Security Header* illustrated in Figure 2.3 may now transport a null, a 4-byte or a 5-byte *Frame Counter* field. The CCM* IV for AES encryption may now contain a 5-byte *Frame Counter*, instead of a 4-byte *Frame Counter* followed by a 1-byte *Key Control* as illustrated in Figure 2.5.

We observe that IEEE 802.15.4e adapts replay protection and semantic security to time-synchronized network communications, as supported by the addendum to the standard. Other than such small modifications, the remaining security services provided by the IEEE 802.15.4 base specification still apply to applications employing IEEE 802.15.4e MAC communications. Although IEEE 802.15.4 does not support end-to-end communications with entities external to the local LoWPAN, it provides security services at the link-layer and sensing platforms implementing the standard offer efficient AES/CCM hardware cryptography that security mechanisms designed at upper layers may benefit from.

2.3.3.5 Research proposals and directions on security with IEEE 802.15.4

Despite the maturity of the IEEE 802.15.4 standard, various limitations may be identified in respect to how it implements security services at the MAC layer. Such limitations may be addressed in future versions of the standard, or on the other avoided by adopting security at upper layers of the stack in Figure 2.2. In the following discussion we analyze the main limitations of IEEE 802.15.4 security mechanisms as currently available at the MAC, which we have previously analyzed:

- As discussed in the current version of the specification [74], IEEE 802.15.4 does not specify any keying model, due to the fact that the most appropriate keying model is

considered to be dependent on the threat model applicable to a particular application, and on the resources available on sensing devices to support key management operations. As previously analyzed, IEEE 802.15.4 provides the support for the storage and usage of cryptographic keys, while key negotiation and management is considered to be a problem to be dealt with in the context of applications.

- The management of IV values on IEEE 802.15.4 ACL entries may be problematic, in case the same key is used in two or more ACL entries. In this situation, it is possible that the sender accidentally reuses the nonce value. This situation is potentially dangerous with stream ciphers encrypting in the CRT mode, as is the case of AES/CCM, since this may enable an adversary to recover plaintexts from cipher texts. The reuse of nonce values is also possible due to the loss of ACL state after a power interruption, or when a node wakes up from a low-power mode.

- Tables storing ACL entries in IEEE 802.15.4 may not provide adequate support for all keying models, in particular group keying and network-shared keying. Group keying is in fact difficult to implement, since each ACL entry may be associated with a single destination address. Thus, the support of group keying requires various ACL entries using the same key, again promoting nonce reuse and the breaking of confidentiality, as previously discussed. On the other hand, network shared keying is incompatible with replay protection. This mode may be supported only through the usage of the default ACL entry, and as such transmitter nodes would have to somehow coordinate their usage of replay counter space.

- As currently defined, IEEE 802.15.4 is unable to protect acknowledgment messages in respect to integrity or confidentiality. An adversary may therefore forge acknowledgments, for which it only needs to learn the sequence number that is sent in the clear of the packet to be confirmed, thus being able to perform DoS attacks.

The issues previously identified may also be dealt with at higher layers of the communications stack, since IEEE 802.15.4 security only protects communications at the link-layer from one hop to the next. On the other hand, many IoT applications may require secure end-to-end communications established between sensing devices and external Internet entities, which are enabled by the technologies forming the stack illustrated in Figure 2.2. Standalone AES/CCM hardware encryption provides an efficient cryptographic basis for research proposals addressing security at the network and higher layers, which we discuss next in the chapter.

2.3.4 END-TO-END NETWORK-LAYER COMMUNICATIONS AND SECURITY USING 6LOWPAN

One fundamental characteristic of the Internet architecture is that it enables packets to traverse interconnected networks using heterogeneous link-layer technologies, and the

mechanisms and adaptations required to transport IP packets over particular link-layer technologies are defined in appropriate specifications. With a similar goal, the IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) working group of the IETF was formed in 2007 to produce a specification enabling the transportation of IPv6 packets over low-energy IEEE 802.15.4 and similar wireless communication environments.

The 6LoWPAN specifications currently form a key technology for the integration of WSN with the Internet, and one that has changed a previous perception of IPv6 as being impractical for constrained low-energy wireless communication environments. The 6LoWPAN adaptation layer materializes a good example of how cross-layer mechanisms and optimizations may enable Internet communication protocols on constrained WSN communication environments, and enables IPv6 end-to-end communications between constrained sensing devices and other similar or more powerful Internet entities, thus providing the required support for the building of future IPv6-based distributed sensing applications on the IoT. The 6LoWPAN adaptation layer maps the services required by the IP layer on the services provided by the IEEE 802.15.4 MAC layer. The characteristics of IEEE 802.15.4 previously discussed strongly determine the usage of very-optimized adaptation mechanisms at the adaptation layer, as we discuss next.

2.3.4.1 6LoWPAN frame format and header compression

The employment of IEEE 802.15.4 at the PHY and MAC layers enables the transportation of data from communication protocols at higher layers of the stack using a limited data payload of 102-bytes, in the absence of link-layer security, as illustrated in Figure 2.7. Given the limited available payload space, the 6LoWPAN adaptation layer is required to optimize payload space usage through packet header compression. 6LoWPAN also defines mechanisms for the support of operations that are required for the employment of IPv6, in particular neighbor discovery and address auto-configuration.

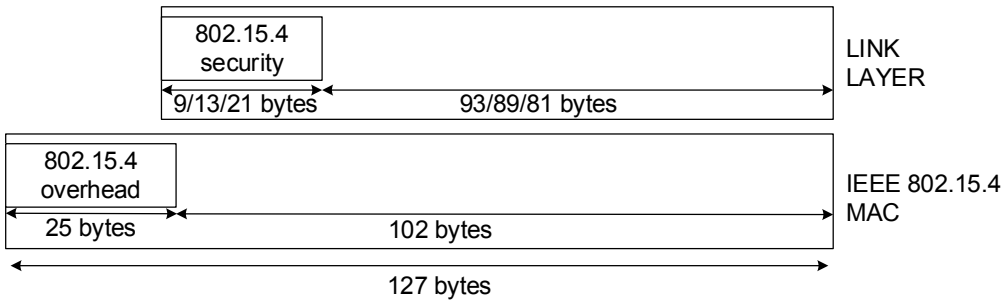

Figure 2.7 - Payload space availability in IEEE 802.15.4 environments

The first document discussing 6LoWPAN is RFC 4919 [69], which discusses the group's general goals and assumptions. RFC 4944 [70] defines mechanisms for the transmission of IPv6 packets over IEEE 802.15.4 networks, with header compression being defined in RFC

6282 [71]. Header compression is implemented by using information from the link and adaptation layers to jointly compress network and transport protocol headers. RFC 6282 also specifies how User Datagram Protocol (UDP) headers may be compressed in the context of the 6LoWPAN adaptation layer.

Other documents relevant for 6LoWPAN standardization are RFC 6568 [91] discussing design and application spaces for 6LoWPAN, RFC 6606 [92] discussing the main requirements for 6LoWPAN routing, and RFC 6775 [93] defining optimizations for the usage of ND (Neighbor Discovery) mechanisms on 6LoWPAN communication networks. All 6LoWPAN encapsulated datagrams transported over IEEE 802.15.4 MAC frames are prefixed by a stack of 6LoWPAN headers. A *type* field occupying the first two bits of the header identifies the 6LoWPAN header, and the standard currently defines four header types employed with the following purposes:

- A *no 6LoWPAN* header indicates that a given packet is not for 6LoWPAN processing. This header in practice enables the coexistence of 6LoWPAN communications with sensing devices employing other communication technologies.

- A *dispatch* header supports IPv6 header compression and link-layer multicast and broadcast communications.

- A *mesh addressing* header supports forwarding of IEEE 802.15.4 frames at the link-layer, as required for the formation of multi-hop networks.

- A *fragmentation* header supports fragmentation and reassembly operations required to transmit IPv6 datagrams over IEEE 802.15.4 networks.

The presence of each 6LoWPAN header is optional, and headers are required to appear in a specific order, starting from the *mesh addressing* header, followed by the *broadcast*, *fragmentation* and *dispatch* headers. The *dispatch* header identifies the compression method applied to a given packet, in particular one of the following compression mechanisms defined for 6LoWPAN:

- LOWPAN_HC1 is the original compression scheme defined in RFC 4944 [70] and supports compression of link-local IPv6 addresses. This compression scheme does not support global IPv6 addresses, thus being suboptimal for the support of applications employing Internet-integrated WSN communication environments.

- LOWPAN_HC1g and LOWPAN_HC2 [94] provide an initial approach to support compression of global IPv6 addresses and UDP headers, respectively. LOWPAN_HC1g assumes that a given network of WSN devices is assigned a compressible 64-bit global IPv6 prefix.

- LOWPAN_IPHC replaces the previous compression methods and is standardized in RFC 6282 [71]. LOWPAN_IPHC compression is based on shared states and enables compression, not only of link-local addresses, but also of global and multicast IPv6 addresses. RFC 6282 also defines the LOWPAN_NHC scheme to compress IPv6 next headers and how UDP header compression may be accomplished. For compatibility with previous implementations, 6LoWPAN implementations are required to support decompression using LOWPAN_HC1.

It is important to note that, although 6LoWPAN is currently defined only for IEEE 802.15.4, other PHY and MAC communication technologies may be adopted in the future. The support of alternative technologies at the lower layers of the protocol stack illustrated in Figure 2.2 will enable IoT applications employing different types of sensing and actuating devices. Proposals currently exist with this goal, particularly regarding the support of Bluetooth Low Energy (BLE) [95] and of Digital Enhanced Cordless Telecommunications Ultra Low Energy (DECT-ULE) [96]. Also, the support of ITU-T G. 9959 networks was recently proposed [97].

In conclusion, we realize the significance of 6LoWPAN as a convergence technology supporting an increasingly growing ecosystem of PHY/MAC communications technologies optimized for particular communication environments and applications. Devices such as RFID tags that are unable to run software applications currently require different approaches to security, as considered in [98], although they may evolve to support 6LoWPAN in the future or on the other hand be supported by future standard communication mechanisms designed in the context of the protocol stack previously illustrated in Figure 2.2.

2.3.4.2 Security in the 6LoWPAN standard

The Internet Protocol Security (IPSec) [99]–[101] enables the authentication and encryption, at the network-layer, of the IP packets exchanged in the context of Internet communication sessions. IPSec supports end-to-end security providing support for the usage of Virtual Private Networks (VPN) in various network configurations. The employment of end-to-end network-layer secure communications may also find useful usage scenarios in future IoT applications, in the context of which constrained sensing devices will be required to communicate with backend devices or other Internet entities. Despite the advantages of end-to-end network-layer security, we observe that no security mechanisms are currently adopted in the context of the 6LoWPAN adaptation layer, as we proceed to discuss:

- The informational RFC 4919 [69] discusses the addressing of security at various complementary protocol layers of the stack illustrated in Figure 2.2, considering that the most appropriate approach may depend on the application requirements and on the constraints of particular sensing devices. This document also identifies the possibility of employing security at the network-layer using IPSec, together with the interest in investigating its applicability in the transport and tunnel usage modes.

- The discussion regarding security on RFC 4944 [70] is related to the possibility of forging or accidentally duplicating EUI-64 interface addresses, which may consequently compromise the global uniqueness of 6LoWPAN interface identifiers. This document also discusses that Neighbor Discovery (ND) and mesh routing mechanisms on IEEE 802.15.4 environments may be susceptible to security threats, and that AES security as available at the link-layer may provide a basis for the development of mechanisms protecting against such threats, particularly for very constrained sensing devices. Nevertheless, this document doesn't propose any specific security solution to address such issues. Other interesting discussion is on the possibility of employing more powerful 6LoWPAN devices in order to support heavy security-related operations, also because such devices may also support existing Internet security protocols, as such representing strategic places for the enforcement of security control mechanisms.

- The discussion concerning security on RFC 6282 [71] focuses on the security issues posed by the usage of a mechanism inherited from RFC 4944, which enables the compression of a particular range of 16 UDP port numbers down to 4 bits. This document discusses that the overload of ports in this range, if employed with applications not honoring the reserved set for port compression, may increase the risk of an application getting the wrong type of payload or of an application misinterpreting the content of a message. As a result, RFC 6282 recommends that the usage of such ports be associated with a security mechanism employing MIC codes.

- The discussion on security contained in the informational RFC 6568 [91] again focuses on the possible approaches to adopt security in the light of the characteristics and constraints of wireless sensing devices. This document discusses threats due to the physical exposure of such devices, which may pose high demands for its resiliency and survivability. It also discusses how wireless IEEE 802.15.4 communications may facilitate attacks against the confidentiality, integrity, authenticity and availability of 6LoWPAN devices and communications.

- Rather than providing a specific approach to routing in 6LoWPAN environments, RFC 6606 [92] provides guidelines that are useful in designing specific routing approaches. As with the previous standard documents, RFC 6606 identifies the importance of security and the usefulness of AES/CCM available at the link-layer. This document also discusses the importance of designing security mechanisms able to adapt to changes in the network topology and devices, rather than employing a static security configuration, given that many 6LoWPAN applications may employ networks that are dynamic in such respects. This document also identifies the importance of time synchronization, self-organization and security localization in providing security for data and multi-hop routing control packets. Other important security requirements identified in RFC 6606 are the support of authenticated broadcasts and multicasts, and the verification of bidirectional links.

- RFC 6775 [93] defines optimizations to enable Neighbor Discovery (ND) operations in 6LoWPAN environments. This document identifies the threat model for IPv6 ND operations defined in RFC 4861 [102] as applicable to 6LoWPAN, and the possibility of adapting the SEcure Neighbor Discovery (SEND) [103] and Cryptographically Generated Addresses (CGA) [104] mechanisms to 6LoWPAN environments.

An important security requirement that is discussed throughout the current 6LoWPAN specification documents is key management. Key management is in fact a cross-layer security issue and one that is interrelated with authentication, since security mechanisms designed to protect communications require that keys are negotiated in the context of the initial authentication of the communicating devices, and periodically refreshed in order to guarantee effective and long-term security, independently of the layer at which communications take place.

While not proposing any specific key management approach, RFC 6568 [91] identifies the possibility of adopting simplified versions of current Internet key management solutions, such as the minimal IKEv2 proposed in [105]. This document describes the requirements for minimal implementations and various optimizations that can be done to adapt IKEv2 to constrained sensing environments, while maintaining compatibility with the Internet key exchange standard. Other approaches are to compress the IKE headers and payload information using 6LoWPAN IPHC compression, as proposed in [106], or to adopt new lightweight key management mechanisms appropriate to the IoT [107].

2.3.4.3 Research proposals and directions on network-layer security using 6LoWPAN

As previously discussed, the current 6LoWPAN specification only discusses general security threats and requirements, despite RFC 4944 [70] clearly identifying the interest of adopting appropriate security mechanisms in the context of the 6LoWPAN adaptation layer. The adoption of security mechanisms at the adaptation layer could enable the employment in WSN environments of network-layer security in a transparent fashion, as we address in Chapter 4. Security in 6LoWPAN is discussed in an initial contribution in the form of an I-D [108] that, while not proposing any particular approach or security mechanisms, identifies the main difficulties in adopting standard network-layer solutions as IPSec and Internet Key Exchange (IKE) in 6LoWPAN environments. The challenges in the adoption of standard network-layer security approaches such as IPSec and IKE in 6LoWPAN environments have also been discussed in previous contributions [109], [110]

The design of appropriate security mechanisms to work in tandem with the mechanisms at the 6LoWPAN adaptation layer would enable secure end-to-end communications at the network-layer for IoT applications, and a few research proposals current exist with this purpose which may contribute to the future design and adoption of standard 6LoWPAN security solutions, as we discuss in detail in Chapter 4. The support of 6LoWPAN network-layer security will also require appropriate support from external Internet entities,

either by introducing support for compressed security headers and related security mechanisms in existing IPSec stacks, or in the other end by designing mechanisms to translate between IPSec and 6LoWPAN security at specialized devices or security gateways.

Regarding other works related with security in 6LoWPAN, authors in [111] discuss the consequences of packet fragmentation attacks against the 6LoWPAN fragmentation and reassembly mechanisms. As such mechanisms render buffering, forwarding and processing of fragmented packets challenging on resource-constrained devices, a malicious or misconfigured node sending forged, duplicate or overlapping fragments may threat the normal functioning or the availability of such devices. This is due to the lack of authentication at the 6LoWPAN adaptation layer, since recipients are unable to distinguish undesired fragments from legitimate ones when performing packet reassembly. The effects of fragmentation attacks include receiver buffer overflow and misusage of the available computational capability, among others. The authors propose the addition of new fields to the 6LoWPAN fragmentation header to deal with such threats, namely of a timestamp providing protection against unidirectional fragment replays and of a nonce providing protection against bidirectional fragment replays.

Also regarding fragmentation attacks against 6LoWPAN-enabled WSN, a more recent contribution [112] proposes the usage of mechanisms supporting per-fragment sender authentication and purging of messages from the receiver's buffer for transmitter devices considered suspicious. The former employs hash chains enabling a legitimate sender to add an authentication token to each fragment during the 6LoWPAN fragmentation procedure, while in the later the receiver decides on which fragments to discard in case a buffer overload occurs, based on the observed sending behavior. This decision is based on per-packet scores, which capture the extent to which a packet is completed along with the continuity in the sending behavior. While this proposal does not require any modification to the 6LoWPAN packet formats, the proposed security mechanisms would have to be adopted in the context of the adaptation-layer.

2.3.5 SECURITY FOR LOW-POWER ROUTING PROTOCOLS

The Routing Over Low-power and Lossy Networks (ROLL) working group of the IETF was formed with the goal of designing routing solutions for IoT applications. The current approach to routing in 6LoWPAN environments is materialized in the Routing Protocol for Low power and Lossy Networks (RPL) [113] Protocol. Rather than providing a generic approach to routing, RPL provides in reality a framework that is adaptable to the requirements of particular classes of applications. We proceed by discussing the main mechanisms of RPL, which are relevant in contextualizing our discussion of security later in the chapter.

2.3.5.1 LoWPAN routing using the ROLL RPL protocol

The design of appropriate routing strategies for 6LoWPAN environments is a very challenging task, due to the inherent specificities of each application and of the constrained sensing devices employed. In consequence, RPL assumes that routing must adapt to the requirements of particular application areas. For each application area, an appropriate RFC documents an objective function that maps the optimization requirements of the target scenario. Requirements are defined in RFC 5548 [114] for urban low-power applications, in RFC 5673 [115] for industrial applications, in RFC 5826 [116] for home automation applications and in RFC 5867 [117] for building automation applications. RPL also employs metrics that are appropriate to 6LoWPAN environments, as currently specified in RFC 6551 [118].

Considering that in the most typical setting LoWPAN nodes are connected through multi-hop paths to a small set of root devices responsible for data collection and coordination, RPL builds a Destination Oriented Directed Acyclic Graph (DODAG) identified by an identifier (DODAGID) for each root device, by accounting for link costs, node attributes, note status information, and its respective objective function. The topology is set up based on a rank metric, which encodes the distance of each node with respect to its reference root, as specified by the objective function. According to the gradient-based approach, the rank should monotonically decrease along the DODAG and towards the destination node.

The simplest RPL topology is made by a single DODAG with just one root, but more complex scenarios are possible. Multiple instances of RPL may run concurrently on the network, possibly with different optimization objectives, as traduced by the correspondent objective function. RPL is designed to support three fundamental traffic topologies: Multipoint-to-Point (MP2P), Point-to-Multipoint (P2MP) and Point-to-Point (P2P). MP2P traffic is routed towards nodes that support the DODAG root role and that may also support gateway functions towards the Internet or other external IP networks. P2MP can be used for networks requiring the usage of actuating devices, in addition to sensors. P2P involve a packet flowing from the source towards the common ancestor of the two communicating devices and then downward to the destination device. The three topologies require RPL to discover both upward routes to support MP2P and P2P traffic, and downward routes to support P2P and P2MP traffic. Tree-based topologies also map well with time-synchronized schedule-based MAC communications using IEEE 802.15.4e [75].

The RPL protocol supports various types of control messages, in particular DIO (DODAG Information Object), DIS (DODAG Information Solicitation), DAO (Destination Advertisement Object), DAO-ACK (DAO acknowledgment) and CC (Consistency Check) messages. A node transmits DIO messages containing information required for other nodes to compute their own rank, to join an existing DODAG and to select a set of parents and the preferred parent in that DODAG among all possible neighbors. DIO messages may be requested by sending a DIS (DODAG Information Solicitation) message. DIO and DIS messages are employed for the

establishment of upward routes in the RPL routing tree, while downward paths are established by having DAO messages to back-propagate routing information from leaf nodes to the roots. A DAO message is triggered by the reception of a DIO message, and its recipient may send a DAO-ACK message to a DAO parent or the DODAG root. Finally, CC messages are used for synchronization of counter values among communicating nodes and provide a basis for the protection against packet replay attacks. All RPL control messages are encapsulated in ICMPv6 packets [43] and are identified by an ICMPv6 type of 155. Regarding security, RPL defines secure versions of the various routing control messages and three basic security modes, as we proceed to discuss.

2.3.5.2 Security in the RPL standard

The current RPL specification [113] defines secure versions of the various routing control messages that we have previously analyzed. In Figure 2.8 we illustrate the format of a secure RPL control message, which transports a *Security* field following the 4-byte ICMPv6 message header. Data related to security is transported between the *Checksum* and the *Base* fields.

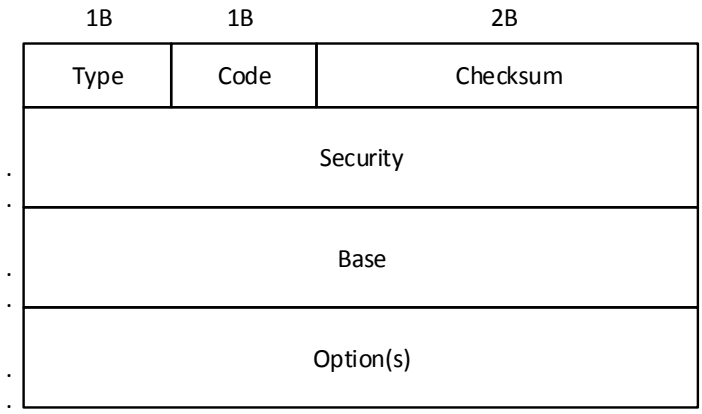

Figure 2.8 - Secure RPL Control Message

The secure variants of RPL control messages may support integrity and replay protection, as well as optional confidentiality and delay protection. The high order bit of the RPL *Code* field identifies whether or not security is applied to a given RPL message. In particular, a message code of 0x80 identifies a secure DIS message, while a secure DIO message is identified by 0x81, a secure DAO by 0x82 and a secure DAO-ACK by 0x83. A Consistency Check (CC) control message is also used to support security and is identified by the code 0x8A. CC messages allow nodes to issue a challenge-response to validate a node's current counter value. One usage of CC control messages is when a received message has an initialized (zero value) counter value and the receiver has an incoming counter currently maintained for the message originator. In this case the receiver initiates counter resynchronization by sending a CC message to the message source.

The *Security* field of a protected RPL control message is illustrated in Figure 2.9. The information in this field indicates the level of security and the cryptographic algorithms employed to process security for the message, while not transporting security-related data such as a Message Integrity Code (MIC) code or a signature. Instead, the security transformation itself states how the cryptographic fields should be employed in the context of the secure message.

In respect to the fields illustrated in Figure 2.9, the *T* bit indicates if the counter field transports a timestamp, and otherwise this field is treated simply as an incrementing counter. The next byte identifies the security suite employed to provide security. The current RPL specification [76] defines the employment of AES/CCM with 128-bit keys for encryption and MAC generation, and of RSA with SHA-256 for digital signatures. The *KIM* (Key Identifier Mode) field indicates whether the cryptographic key required to process security for this message may be determined implicitly or explicitly. RFC 6550 [76] currently defines different values for this field to support group keys, per-pair keys, and signatures. The *LVL* (Security Level) field indicates the provided packet protection and allows for varying levels of data authentication and, optionally, of data confidentiality. RFC 6550 also defines various values to identify the usage of confidentiality, integrity and data authenticity using MAC-32 and MAC-64 authentication codes, and of 2048 and 3072-bit signatures using RSA.

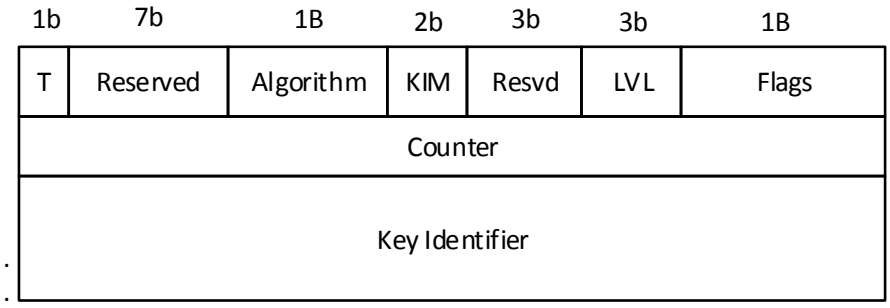

Figure 2.9 - Security section of a secure RPL Control Message

Also relating the message illustrated in Figure 2.9, the *Flags* field is currently reserved, and the *Counter* field transports a non-repeating value used to support semantic security and protection against packet replay attacks. The *Key Identifier* field indicates which key should be used to process security for the packet. This field supports various levels of granularity of packet protection. It is represented as indicated by the key identifier mode field and is divided in a *key source* and *key index* subfields. The *key source* subfield indicates the logical identifier of the originator of a group key. The *key index* subfield, when present, allows unique identification of keys with the same originator.

Regarding the employment of cryptographic algorithms in RPL, AES/CCM is adopted as the basis to support security in the current specification [76], while we note that other

algorithms may be adopted in the future and appropriately identified in the security section of a secure RPL control message, as illustrated in Figure 2.9. RPL control messages may be protected using both an integrated encryption and authentication suite, such as supported by AES/CCM, as well as schemes that employ separate algorithms to support message encryption and authentication.

The entire RPL message is within the scope of RPL Security. MAC codes and signatures are calculated over the entire unsecured IPv6 packet, considering the mutable IPv6 fields to be all zeros. When an RPL ICMPv6 message is encrypted, encryption starts at the first byte after the *Security* section and continues to the last byte of the packet. The IPv6 header, ICMPv6 header and RPL message, up to the start of the *Security* field are not encrypted, since these fields are required to correctly decrypt the packet. Other than defining how security is applied to routing control messages, RPL also defines three security modes, with the following purposes:

- The *unsecured* mode corresponds to the transmission and reception of routing control messages without security applied. Thus, this mode corresponds to the support of RPL routing in 6LoWPAN communication domains without any security mechanisms applied in the context of the routing protocol.

- The *preinstalled* security mode may be employed by a device using a preconfigured symmetric key to join an RPL instance, either as a host or a router. This key is employed to support confidentiality, integrity and data authentication for routing control messages.

- A device operating as a router may employ the *authenticated* security mode. The device initially joins the network using the preconfigured key and the *preinstalled* security mode, and next obtains a different cryptographic key from a key authority with which it may start functioning as a router. The key authority is responsible for authenticating and authorizing the device for this purpose.

The current RPL specification [76] states that the *authenticated* security mode must not be supported by symmetric cryptography, although not specifying how asymmetric cryptography is to be employed to support node authentication and key retrieval by the device intending to operate as a router. A more clear definition of such mechanisms may thus be defined for future versions of the routing protocol specification.

While not introducing additional security mechanisms, other RFC documents that are relevant to RPL also discuss security. The informational RFC documents discussing routing requirements for the various application areas [114]–[118] discuss security from the perspective of the protection of routing control messages with appropriate security mechanisms supporting confidentiality, authentication and integrity, as is possible using the secure versions of the RPL control messages previously discussed. RFC 6551 [118] specifies a

set of link and node routing metrics appropriate to the characteristics and constraints of 6LoWPAN environments, and discusses the necessity of handling such metrics in a secure and trustful manner, including protection against nodes being able to falsify or lie in the advertisement of metrics, as a way to protect against attacks on normal routing operations.

We observe that, other than the secure versions of the routing control messages and the security modes previously discussed, no further security mechanisms are designed in the current version of the RPL Protocol standard [76]. The remaining standard documents produced in the context of IETF ROLL only identify general security requirements and goals, without introducing additional security mechanisms.

2.3.5.3 Research proposals and directions on routing-layer security with RPL

As previously discussed, RPL defines secure versions of routing control messages, together with a few basic security operations. On the other hand, RPL currently lacks mechanisms to support important security procedures such as the secure bootstrapping of devices, key management and management of routing security policies, among others. The current specification [76] only addresses the handling of keys with applications employing device pre-configuration, discussing how such devices should be able to join a network using a preconfigured default shared group key or a key learned from a received DIS configuration message, while not defining how authentication and secure joining mechanisms may be designed to support other more dynamic or security-critical application contexts.

Similarly to routing profiles defined for particular application areas, research and standardization may target the definition of security policies stating how security must be applied to protect routing operations in particular application contexts. Such policies may identify the requirements of applications in terms of confidentiality, integrity, authenticity and replay protection for control messages, among others.

A discussion on the open issues in respect to security in RPL is expressed in [119], which performs an analysis on the main threats against ROLL routing mechanisms, together with recommendations on how to address security. This document identifies such threats by employing the ISO 7498-2 security reference mode [120], which include Authentication, Access Control, Data Confidentiality, Data Integrity and Non-Repudiation, and to which Availability is added. This model enables the identification of the assets to protect, of its security needs, and of the points of access through which security may be compromised. The model enables the categorization and discussion of the threats and of the specific attacks regarding confidentiality, integrity and availability of routing message exchanges in the context of ROLL routing protocols. This document also proposes a security framework for ROLL routing protocols, which is built upon previous work on security for routing and adapting the assessments to the constraints of 6LoWPAN environments, In the context of this framework, security measures are identified that can be activated in the context of the RPL routing protocol, together with system security aspects that may impact routing but that

also require considerations beyond the routing protocol, as well as potential approaches in addressing them. The assessments in this document may provide the basis of the security recommendations for incorporation into ROLL routing protocols as RPL. We also note that the implications of the various security requirements, defined as appropriate for each application, to the routing protocol itself, is also a topic for future research and standardization work.

Other important aspect of RPL security, as currently proposed, is that the services defined in the current specification [76] offer security against external attacks only. An internal attacker is in possession of a node and in consequence of the required security keys, and as such may selectively inject routing messages with malicious purposes. Authors in [121] discuss the issue of internal attacks on RPL, particularly on the rank concept as employed by the protocol. The rank serves the purposes of route optimization, loop prevention and management of routing control overhead. This work discusses various possible attacks against the rank property, together with its impact on the performance of the network. Authors also discuss that this limitation in RPL is due to the fact that a child node receives parent information through control messages, but is unable to check the services provided by the parent, so it will follow a bad quality route if it has a malicious parent. While not proposing specific measures or mechanisms for this purpose, this work discusses that mechanisms could be adopted in RPL to allow a node to monitor the behavior of its parents and defend against such threats.

Internal attacks against RPL are also discussed in [122], particularly that an internal attacker is able to compromise a node in order to impersonate a gateway (the DODAG root) or a node that is in the vicinity of the gateway. The authors propose a version number and rank authentication security scheme based on one-way hash chains, which binds version numbers with authentication data (MAC codes) and signatures. This scheme offers protection against internal attackers that are able to send DIO messages with higher version number values or that are able to publish a high rank value. The former attack enables an attacker to impersonate the DODAG root and initiate the reconstruction of the routing topology, while in the later a large part of the network may be forced to connect to the DODAG root via the attacker, thus providing the ability to eavesdrop and manipulate part of the network traffic. The security data enable intermediate nodes to validate DIO messages containing new version numbers and rank values. While an evaluation is performed against the impact of these mechanisms on computational time, this work doesn't discuss its impact on aspects such as energy or memory of constrained sensing devices. The same mechanisms are also proposed in the form of a recent I-D [123].

In another contribution focusing on internal attacks against RPL [124], the authors discuss the effects of sinkhole attacks on the network, particularly regarding its end-to-end data delivery performance in the presence of an attack. A sinkhole consists of a compromised node that purposely captures and drops messages. The authors propose the combination of a parent fail-over mechanism with a rank authentication scheme and, based on simulation

results, argue that the combination of the two approaches produces good results, and also that by increasing the network density the penetration of sinkholes may be combated without needing to identify the sinkholes. The rank-verification technique is also based on one-way hash chains as in [122][123], while the parent fail-over scheme employs an end-to-end acknowledgment scheme controlled by the DODAG root node.

The previous research proposals represent approaches to address open security issues in RPL, particularly regarding the presence of internal threats and attackers. Such proposals may provide contributions to the adoption of future security mechanisms in the context of RPL. As extensive research has been performed in the area of security for routing protocols for sensor networks and ad hoc networks in the past, approaches in such proposals may also guide future research efforts regarding RPL security, as long as appropriately designed to cope with the characteristics of 6LoWPAN devices and the mechanisms of RPL.

2.3.6 TRANSPORT-LAYER COMMUNICATIONS AND SECURITY MECHANISMS

The 6LoWPAN adaptation layer currently supports only UDP [125] transport-layer communications, although it is possible to envision the support of alterative transport-layer protocols in the future, possibly by adopting mechanisms from the TCP [126] protocol. UDP is currently the adopted transport-layer protocol for 6LoWPAN, due to its simplicity and low impact on the limited packet payload space available at the adaptation layer. In the context of the employment of UDP, the Datagram Transport Layer Security (DTLS) [127] protocol appears as a natural candidate to provide security for transport-layer communications in WSN environments. DTLS is in practice the Transport Layer Security (TLS) [128] protocol with added features to deal with the unreliable nature of transport-layer communications.

Despite the apparent appropriateness of DTLS to WSN environments, the effectiveness of its employment in constrained low-energy WSN environments is currently not consensual among researchers in the area. In consequence, research efforts are currently targeting the investigation of the impact of DTLS in wireless sensing devices, together with the design of mechanisms to adapt or optimize the protocol for WSN communication environments. Other aspects currently being investigated include the impact of public key cryptography on sensing platforms to support authentication for DTLS, the modification of the protocol to support multicast communications and group keying, and the usage of DTLS with reverse proxies in CoAP. The employment of DTLS to protect transport-layer communications in the context of Internet-integrated WSN are further discussed later in this chapter, in the context of CoAP security.

In what respects the usage of alternative transport-layer communication protocols for LoWPAN environments, TCP is also currently being considered. Existing research proposals [129][130] target the employment of TCP on WSN environments, although not considering yet the usage of 6LoWPAN. If TCP is ever adopted for Internet-integrated WSN environments, SSL is a natural candidate to support end-to-end transport-layer security. We

extend our discussion on proposals focusing on the adaptation of SSL for constrained WSN environments later in the thesis.

2.3.7 APPLICATION-LAYER COMMUNICATIONS AND SECURITY MECHANISMS

As previously discussed, the currently supported transport-layer protocol is UDP [125], since it provides a good trade-off between reliability and energy-cost. The adoption of transport-layer approaches with characteristics more close to protocols such as the Transmission Control Protocol (TCP) [126] is still open to debate, and research is ongoing addressing the adaptation of TCP for 6LoWPAN environments [129]. Transport protocols with such mechanisms are currently considered to be too expensive for 6LoWPAN environments, given its requirements in terms of the exchange of traffic control information and the maintenance of status information on constrained sensing devices. The adoption of UDP on 6LoWPAN networks also dictates aspects of the design of the CoAP application-layer protocol at the Constrained RESTful Environments (CoRE) working group of the IETF, as we proceed to analyze.

2.3.7.1 IETF CoAP application-layer communications

The CoAP [34] protocol implements a set of techniques to compress application-layer protocol metadata without compromising application inter-operability, in conformance with the REST architecture of the web. Application-layer communications may enable IoT sensing applications to interoperate with existing Internet applications without requiring specialized application oriented code or translation mechanisms. In practice, CoAP restricts the HTTP dialect to a subset that is well suited to the constraints of 6LoWPAN sensing devices, and may enable abstracted communications between users, applications and such devices, in the context of IoT applications.

The CoAP protocol provides a request and response communications model between application endpoints and enables the usage of key concepts of the web, namely the usage of URI addresses to identify the resources available on constrained sensing devices. The protocol may support end-to-end communications at the application-layer between constrained IoT sensing devices and other Internet entities purely using CoAP or in alternative by translating HTTP to CoAP at a reverse or forward proxy.

The messages of the CoAP protocol are exchanged asynchronously between two endpoints and are used to transport CoAP requests and responses. Since such messages are transported over unreliable UDP communications, CoAP implements a lightweight reliability mechanism. The CoAP messages may be marked as *Confirmable*, for which the sender activates a simple stop-and-wait retransmission mechanism with exponential back off. The receiver must acknowledge a *Confirmable* message with a corresponding *Acknowledge* message or, if it lacks context to process the message properly, reject it with a *Reset* message. The *Acknowledge* or *Reset* message is related to a *Confirmable* message by means

of a Message ID, along with additional information on the address of the corresponding endpoint. CoAP messages may also be transmitted less reliably if marked as *Non-Confirmable*, in which case the recipient does not acknowledge the message. Similarly to HTTP, CoAP defines a set of method and response codes available to applications.

Other than a basic set of information, most of the information in CoAP is transported using options. Options defined for the CoAP Protocol may be critical, elective, safe or unsafe. A critical option is one that an endpoint must understand, while an elective option may be ignored by an endpoint not recognizing it. Safe and unsafe options determine how an option may be processed by an intermediary entity. An unsafe option needs to be understood by the proxy in order to be safely forwarded, while a safe option may be forwarded even if the proxy is unable to process it.

Figure 2.10 illustrates the CoAP header and message format as proposed in the current specification of the Protocol [34]. The message header starts with a 4-byte fixed header, formed by the *Version* field (2 bits), the *T* (message type) field (2 bits), the *TKL* (Token Length) field (4 bits), the *Code* field (8 bits) and the *Message ID* (16 bits). The token enables a CoAP entity to perform matching of CoAP requests and replies, while the message ID supports duplicate detection and optional reliability.

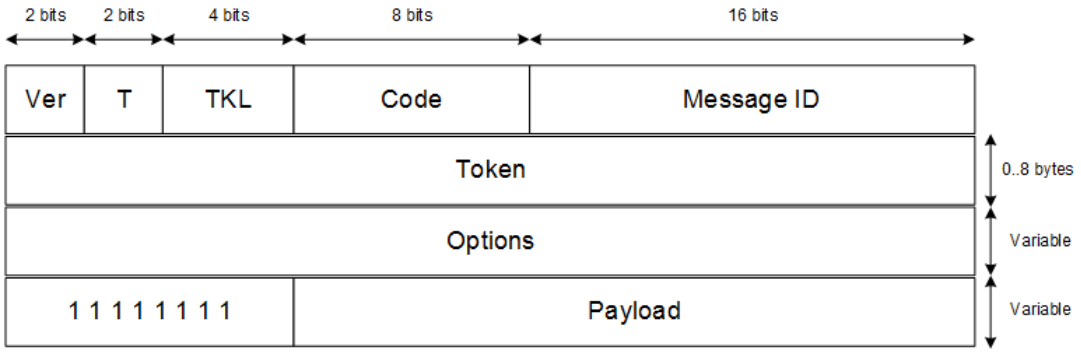

Figure 2.10 - Format of a CoAP message header

The CoAP options are defined in the Type-length-value (TLV) format, by specifying its option number followed by the corresponding length and value. The current specification of CoAP defines options such as *Uri-Host*, *Uri-Port*, *Uri-Path* and *Uri-Query* allowing to specify the target resource of a request sent to a CoAP server, *Content-Format* to specify the representation format of the message payload, and *Max-Age* to indicate the maximum time a CoAP response may be cached before being considered not fresh, among others [34].

Regarding security, CoAP currently adopts the DTLS security protocol to support transparent security at the transport-layer, rather than implementing security at the application-layer. This is again in line with the adoption of UDP as the preferred transport-layer communications protocol, and with the fact that transparent security in the context of

pre-established security session is considered to be an appropriate strategy to protect individual CoAP messages.

2.3.7.2 Security in the CoAP Protocol

The CoAP Protocol [34] defines bindings to the DTLS (Datagram Transport-Layer Security) [131] protocol in order to protect CoAP messages, along with a few minimal configurations that are mandatory to implement and appropriate to constrained WSN environments. DTLS is in practice TLS [128] with added features to deal with the unreliable nature of the UDP transport. The impact of supporting DTLS on constrained wireless sensing devices may be due to the cost of supporting the initial handshake, and also the processing of security for the various CoAP messages exchanged between client and server.

Figure 2.11 illustrates the availability of payload space for applications in IEEE 802.15.4 and 6LoPWAN communication environments in the presence of CoAP and DTLS. As we may observe in this figure, DTLS adds a limited per-datagram overhead of 13 bytes, not counting any initialization vectors, integrity check values or the padding that may be required by the cipher suite employed. Shared-context 6LoWPAN header compression requires 10 bytes for an UDP/IPv6 header, while the CoAP fixed header requires 4 bytes.

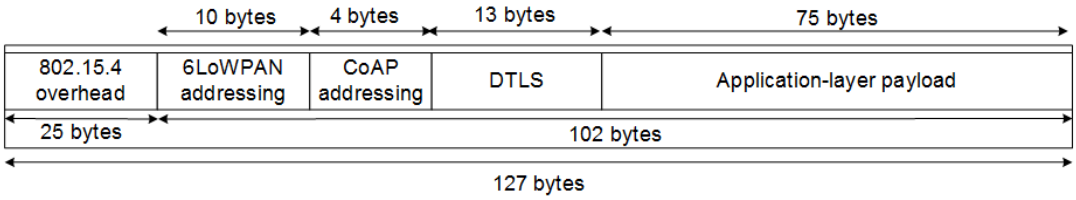

Figure 2.11 - Payload space availability for 6LoWPAN-based technologies

The adoption of DTLS implies that security is supported at the transport-layer, rather than being designed in the context of the CoAP protocol itself. The establishment of a DTLS security session at the transport-layer between a CoAP client and a CoAP server provides confidentiality, integrity and data origin authentication for the CoAP messages exchanged between the two entities. CoAP also adopts AES/CCM as the cryptographic mechanisms of reference, and security against replay attacks may be achieved by using a different nonce value for each protected CoAP packet.

In addition to the adoption of DTLS, the current CoAP specification defines four complementary security modes, which differ on how authentication and key negotiation is performed, as we proceed to analyze:

- The *NoSec* security mode corresponds to CoAP messages being sent and received without security, thus by applications that do not require security properties to be guaranteed for application-layer communications.

- The *PreSharedKey* security mode may be employed by sensing devices that are pre-programmed with the symmetric cryptographic keys required supporting secure communications with other devices or groups of devices. This mode may be appropriate to applications employing devices which are unable to support public-key cryptography, or for which it is convenient to pre-configure security for the devices. Applications may use one key per destination device or a single key for a group of destination devices.
- The *RawPublicKey* security mode is appropriate for devices requiring authentication based on public keys, but which are unable to participate in public-key infrastructures. A given device must be preprogrammed with an asymmetric key pair that may be validated using an out-of-band mechanism [132] and possibly programmed as part of the manufacturing process, while without a certificate. The device has an identity calculated from its public key and a list of identities and public keys of the nodes it can communicate with. This security mode is mandatory to implement.
- The *Certificates* security mode also supports authentication based on public-keys but for applications that are able to participate in a certification chain for certificate validation purposes. This security mode thus assumes the availability and usage of a security infrastructure. The device has an asymmetric key pair with an X.509 certificate that binds it to its Authority Name and is signed by some common trusted root. The device also has a list of root trust anchors that can be used for validating a certificate.

The *RawPublicKey* and *Certificates* security modes are supported by Elliptic Curve Cryptography [133]. ECC supports device authentication using the Elliptic Curve Digital Signature Algorithm (ECDSA) and key agreement using the ECC Diffie-Hellman counterpart, the Elliptic Curve Diffie-Hellman Algorithm with Ephemeral keys (ECDHE). The *NoSec* security mode corresponds to a device sending packets without security, using the “coap” scheme in URI addresses identifying resources available on CoAP servers. On the other hand, accesses to resources with DTLS use the “coaps” scheme, and in this case a security association at the transport-layer using DTLS must exist between the CoAP client and the CoAP server. The current CoAP specification defines a mandatory-to-implement cipher suite for each of the previous security modes, based on the usage of AES/CCM and ECC cryptographic operations, as follows:

- Applications supporting the *PreSharedKey* security mode are required to support at least TLS_PSK_WITH_AES_128_CCM_8 [134]. This cipher suite supports authentication using pre-shared symmetric keys and 8-byte nonce values, to encrypt and produce 8-byte integrity codes.
- Applications supporting the *RawPublicKey* CoAP security mode are required to support the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 [128][135] security suite using ECDSA-capable public keys. This security mode also makes use of SHA-256 to compute hashes.

- Applications supporting the *Certificates* security mode are also required to support the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 cipher suite. Regarding the usage of public-keys transported in X.509 certificates, the *SubjectPublicKeyInfo* field in a X.509 certificate defines how the corresponding public key must be employed for ECC computations. The certificate must also contain a signature created using ECDSA and SHA-256. Applications using devices with a shared key plus a certificate must also support TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA.

In addition to the cipher suites previously discussed, we may expect that further security suites may be adopted in future versions of CoAP, as this would enable a better adaptation of the various security modes to different applications and types of sensing platforms. As with the remaining protocols illustrated in the stack of Figure 2.2, CoAP doesn't currently define or adopt any solution to address key management, other than the assumption that initial keys are available resulting from the DTLS authentication handshake.

2.3.7.3 Research proposals and directions on application-layer security using CoAP

Despite the adoption of DTLS to protect CoAP messages, work is ongoing in the CoRE working group and the CoAP specification is not yet completely defined. In this context, we may also identify various aspects that are particularly relevant in this context of addressing security for CoAP communications, and that may guide future research and standardization efforts:

- The impact of DTLS as currently proposed for CoAP must be experimentally evaluated considered the various classes of sensing devices currently available. If it is true that AES/CCM is efficiently available at the hardware in IEEE 802.15.4 sensing platforms, the DTLS handshake (for authentication and key agreement) can pose a significant impact on the resources of constrained devices, particularly considering the adoption of ECC public-key cryptography to support authentication. In this context, research efforts are being conducted in investigating optimizations for DTLS in IoT environments, and also on conducting interoperability testing of DTLS implementations using CoAP [136].

- The support of ECC public-key cryptographic on 6LoWPAN environments also motivates further investigation. The viability of ECC cryptography on constrained sensing platforms is not currently clear, and optimizations may be designed at the hardware of new sensing platforms to support ECC computations, similarly to the support of AES/CCM in IEEE 802.15.4 platforms.

- Sensing devices employed in the context of future IoT applications may require mechanisms supporting the online verification of the validity of X.509 certificates, particularly for the CoAP *Certificates* security mode. The design and adoption of mechanisms with this purpose also requires further investigation. Possible approaches

may consist in investigating the applicability of current Internet approaches such as the Online Certificate Status Protocol (OCSP) [137] or OCSP stapling through the TLS Certificate Status Request extension defined in RFC 6066 [138]. OCSP stapling enables the presenter of a certificate to bear the resource cost involved in serving OCSP validation requests, instead of the issuing Certification Authority (CA), as with OCSP. Such approaches may also be simplified for 6LoWPAN environments.

- The usage of DTLS is not appropriate to group keying as required to support security with multicast communications. The current CoAP specification [34] discusses the applicability of DTLS as a component of a future group key management protocol.

- The employment of DTLS is not well suited to the usage of CoAP proxies in forward or reverse mode. Although end-to-end communications are at the hearth of IPv6, the exposure of constrained IoT devices to the Internet may call for security mechanisms based on the usage of security gateways, which may also support the roles of border routers for 6LoWPAN and CoAP.

Two main research approaches regarding the employment of DTLS to protect CoAP communications in the context of the architecture illustrated in Figure 2.2 consist on the investigation of the impact of DTLS on existing sensing platforms, and on the employment of alternative approaches to provide security for application-layer communications. The former may include the employment of mechanisms enabling the delegation of costly operations to other devices or of hardware-assisted security. On the other hand, the later may involve the design of security mechanisms in the context of the application-layer protocol.

2.3.7.3.1 Proposals on the impact of DTLS on sensing devices

The impact of the DTLS protocol on the resources of constrained wireless sensing platforms is currently under investigation and has also motivated the formation of the DTLS In Constrained Environments (dice) working group of the IETF, in 2013. Various features of the protocol have been identified in the literature as complicating its adoption in constrained sensing environments, as we discuss next.

The DTLS handshake as currently defined in the protocol [131] is problematic to support in constrained sensing environments, as large messages cause fragmentation at the 6LoWPAN adaptation layer and the cost of the computation of the *Finished* message at the end of the handshake is high, as discussed in [139]. Fragmentation implies that retransmission and reordering of handshake messages at the DTLS communicating entities may result in added complexity and reliability. Research approaches for such problems may include the design of appropriate reliability mechanisms to support the transportation of DTLS handshake messages, or of alternative transport-layer approaches.

As discussed in [140], other problem is that DTLS is unable to support multicast communications, which will be required in many IoT environments. Secure CoAP multicast communications will also require the establishment of appropriate session keys among the various participating devices. This can be achieved either by designing an external key management solution appropriate to applications using CoAP and DTLS, or on the other hand by modifying the DTLS handshake to support session key negotiation for a group of devices. In a previous approach to the support of multicast security using DTLS authors in [141] proposed the setup of multicast groups via a gateway, with each sensing device performing the initial DTLS handshake with the gateway and receiving the required keying material. Authors in [142] propose the adaptation of the DTLS record layer to enable multiple senders in a multicast group to securely send CoAP messages using a common group key, while providing confidentiality, integrity and replay protection to group messages. This proposal considers that the required group keying material is already available in the context of a given group security association, particularly the appropriate client and server read and write MAC keys, encryption keys and IV values. How the required keying material is configured or obtained prior to normal multicast communications is currently an open issue, thus also representing an opportunity for research.

Other features of the protocol may be inappropriate to IoT applications and devices, and as such a suitable DTLS profile may be identified and adopted. In [142] the authors discuss various issues that may difficult the usage of DTLS in constrained sensing devices, as the inadequateness of the timers for message retransmission as defined in the DTLS standard, which may require large buffers on the receiver to hold data for retransmission purposes, and the size of the code required to support DTLS in constrained sensing platforms. The same document also discusses the usage of stateless compression of the DTLS headers with the goal of reducing the overhead of DTLS records and handshake messages. Authors in [143] followed this approach, by proposing the compression of the DTLS headers using LOWPAN_IPHC 6LoWPAN header compression. Similarly to IPSec compressed security headers, the compression of DTLS headers in the context of 6LoWPAN requires appropriate support from existing implementations of DTLS, or on the other hand the design of mechanisms to map between DTLS and compressed DTLS.

Other proposals do exist based on the employment of a gateway to support security-related mechanisms. As discussed in [140], one issue to be addressed for CoAP security is the inexistence of mechanisms for mapping between TLS and DTLS, which may be supported by such a gateway. Authors in [140] propose a mechanism for mapping between TLS and DTLS at a security gateway that also supports CoAP to HTTP mappings. An alternative approach to having constrained sensing platforms fully supporting DTLS is to offload costly operations to a more capable device. A few proposals consider this approach, focusing particularly on the delegation of operations performed in the context of the DTLS handshake. In [144] a mechanism is proposed also based on a proxy to support sleeping devices, using a mirroring mechanism to serve data on behalf of sleeping smart objects. In [145] the authors propose

an end-to-end architecture supporting mutual authentication with DTLS, using specialized trusted-platform modules (TPM) supporting RSA cryptography on sensing devices, rather than ECC public-key cryptography as currently required for CoAP. This proposal is also described and more thoroughly evaluated in [146] using an experimental wireless sensor network.

The impact of the processing of certificates using current sensing platforms is an aspect that also requires proper evaluation studies in a near future. Authors in [147] discuss possible design approaches to address the computational burden of supporting certificates in constrained sensing platforms, also by considering the usage of a security intermediary. The proposed approaches are certificate pre-validation and session resumption. Certificate pre-validation involves a security gateway supporting the validation of certificates in the context of the handshake, before forwarding the handshake messages to the final sensing device. Session resumption allows communication peers to maintain minimal session state after session teardown, which they may use to later resume secure communications without the need of performing again the DTLS handshake. For very constrained sensing devices, this proposal addresses the full delegation of the DTLS handshake to a proxy using a mechanism based on TLS session resumption without server-side state.

2.3.7.3.2 Proposals on alternative approaches to CoAP security

Recent research and standardization work is also considering the employment of alternative approaches to DTLS in order to guarantee the security of CoAP communications. One of such approaches is to use CoAP to support costly DTLS handshake operations. In [147] the authors propose the usage of a RESTful DTLS handshake to deal with the problem of message fragmentation at the 6LoWPAN adaptation layer. The proposed mechanism enables the efficient transmission of DTLS handshake messages in the payload of CoAP messages using CoAP block-wise transfers [148] for larger messages. In this proposal a DTLS session is modeled as a CoAP resource and a well-known URI path is used to identify a collection resource that models the set of active security sessions.

An alternative approach consists in designing security to be integrated into to CoAP protocol itself. This approach was first discussed in an I-D [149], which proposes new CoAP security options for the activation and deactivation of security contexts between a CoAP client and server, and for the identification of CoAP messages with security applied. In this proposal a CoAP client and server maintain a shared security context, in a similar fashion to security sessions maintained by the DTLS protocol. As an alternative approach, security could be applied in a more granular form, according to each particular message or its contents, among other approaches, as we explore later in the thesis.

The I-D in [150] also proposes the addition of two new CoAP options for security, the *Profile* and *Sec-flag* options. Contrary to [149], such options complement DTLS security rather than representing an alternative. The *Profile* option enables the attribution of a CoAP message to

a particular application and the processing of security at an intermediary entity accordingly. On the other hand, the *Sec-flag* option enables the usage of lower layer security (rather than DTLS) in a particular segment of the communications path. This document also proposes an initial security negotiation scheme using CoAP messages transporting the *Sec-flag* option.

The main characteristics of the proposals previously discussed are summarized in Table 2.2. In this table we refer to proposals applying to the 6LoWPAN adaptation layer, transport-layer, routing-layer and application-layer, in the context of the reference stack previously illustrated in Figure 2.2. We also include references to security mechanisms and solutions adopted or currently being designed in the context of the various standardization groups.

Our following analysis is focused on aspects of security that in principle require or benefit from cross-layer security approaches, and thus that are not related to a particular layer of the stack illustrated in Figure 2.2. We thus proceed by discussing the existing works focusing on security aspects that do not belong in the context of a single protocol layer.

2.3.8 CROSS-LAYER SECURITY ASPECTS

Other security aspects must be considered that do not apply necessarily to a specific protocol layer, and that in consequence may be targeted with cross-layer approaches. An essential security aspect in the context of the integration of WSN with the Internet is *key management*, and one that will play a fundamental role in the support of end-to-end security mechanisms. Key management may be considered a cross-layer security issue and one that is interrelated with authentication, since security mechanisms designed to protect communications require that keys are negotiated in the context of the initial authentication of the communicating devices and periodically refreshed in order to guarantee effective and long-term security, independently of the layer at which communications take place.

While not proposing any specific key management solution, RFC 6568 [91] identifies the possibility of adopting simplified versions of current Internet key management solutions, such as the minimal IKEv2 proposed in [105]. RFC 6568 describes the requirements for minimal implementations of IKEv2, together with possible optimizations promoting its adaptability to constrained WSN environments. Other approaches may be pursuit to adapt IKEv2 to Internet-integrated low-power WSN environments. One is to compress the IKE headers and related payload data using 6LoWPAN IPHC compression, as proposed in [106]. The other is to adopt new lightweight key management mechanisms that are more close to the capabilities of WSN environments and to the characteristics of IoT applications [107].

Table 2.2 – Security proposals for 6LoWPAN-based communication technologies

Research proposal	Operational Layer	Security properties and functionalities supported	Application context of security	Implementation details
[151][152]	6LoWPAN adaptation layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (network layer) security	6LoWPAN IPHC compression of AH and ESP security headers; preprogrammed 128-bit keys
[111]	6LoWPAN adaptation layer	Resistance against fragmentation attacks	End-to-end 6LoWPAN with fragmentation	Addition of a timestamp plus a nonce to the 6LoWPAN fragmentation header to support security against unidirectional and bidirectional fragment replays
[112]	6LoWPAN adaptation layer	Resistance against fragmentation attacks	End-to-end 6LoWPAN with fragmentation	Usage of mechanisms to support per-fragment sender authentication using hash chains and purging of messages from suspicious senders based on the observed behavior
[143]	Transport-layer	Confidentiality, integrity and replay protection	Security for CoAP multicast communications	Adaptation of the DTLS record layer to enable multiple senders in a multicast group to securely send CoAP messages using a common group key
[143]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (transport layer) security	Compression of the DTLS headers in the context of 6LoWPAN using IPHC
[140]	Transport-layer	TLS and DTLS mapping for end-to-end secure communications	Transparent end-to-end (transport-layer) security	Mapping between TLS and DTLS using a gateway also providing HTTP to CoAP mapping
[144]	Transport-layer	Support of end-to-end transport-layer security for sleepy devices	Transparent end-to-end (transport-layer) security for inactive devices	Usage of a proxy to support secure end-to-end communications and data retrieval from devices that may be inactive
[145][146]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (transport layer) security	End-to-end DTLS using mutual authentication with hardware support provided by specialized trusted-platform modules (TPM) supporting RSA cryptography
[147]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	End-to-end (transport layer) security	Usage of the certificate pre-validation and session resumption to offload public key authentications to the gateway. Certificates and sessions managed by the gateway
[141]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	Support secure multicast communications on sensing devices	Setup of multicast groups by a gateway, each sensing device performs the initial DTLS handshake with the gateway and receives the required keying material

[148]	Transport-layer	Support of DTLS handshake with block-wise communications	Support authentication and initial key agreement with sensing devices	DTLS handshake messages transported in the payload of CoAP application-layer messages using CoAP blockwise transfers to reduce 6LoWPAN fragmentation
[76]	Routing layer	Confidentiality, integrity, authentication, non-repudiation	Protection of RPL routing control messages	Definition of secure versions of the RPL routing control messages, together with two security modes to protect routing updates
[120]	Routing layer	Security framework for ROLL routing protocols	Identification of security measures appropriate to the RPL routing protocol	Identification of security measures that can be activated in the context of RPL, and of the system aspects that may impact on routing, as well as potential approaches in addressing them
[122][123]	Routing layer	Resistance against internal attacks	Protection of RPL routing operations against falsified routing updates	Usage of a version number and rank authentication security scheme based on one-way hash chains providing security against internal attackers
[124]	Routing layer	Resistance against internal attacks	Protection of RPL routing operations against falsified routing updates	Usage of a security mechanism combining parent fail-over with a rank authentication scheme to combat sinkhole attacks
[34]	Application layer	Confidentiality, integrity, authentication, replay protection	Protection of CoAP application-layer messages using DTLS	Definition of bindings to DTLS to protect CoAP messages, together with three security modes with different approaches to cryptographic key management
[149]	Application layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (application layer) security	CoAP security options allow for the setup of security contexts between CoAP communicating entities and protection of CoAP messages
[150]	Application layer	Confidentiality, integrity, authentication, non-repudiation	Application layer security with application identification and support for link-layer security	CoAP security options complement DTLS security, enabling the identification of particular applications and the employment of link-layer security when appropriate

The gateway supporting the integration of WSN with the Internet can also support standard Internet key negotiation mechanisms with Internet hosts, while abstracting such key negotiation operations from the constraints and characteristics of WSN devices. The gateway may deal with the identification and authorization of sensing devices prior to key management, therefore acting as a trusted broker for end-to-end key negotiation purposes. Alternatively, key negotiation may be performed in a truly end-to-end fashion, as such having key management mechanisms dealing with the constraints and characteristics of sensing devices and applications.

The applicability of existing key management mechanisms designed to support link-layer security on sensor networks [153] to Internet-integrated WSN can also be investigated. Proposals based on mathematical techniques such as linear algebra, combinatory or algebraic geometry may be of interest, as they may contribute or at least provide the ground for the adoption of new key-management mechanisms [107]. Research work may also target the extension of such proposals to global environments in the context of their integration with the IKE standard, or their adaptation to the usage of a trusted third-party, as this would provide support for the usage a of security infrastructure supporting authentication and key negotiation for Internet-integrated WSN.

Other important security services in the context of the integration architecture previously discussed and illustrated in Figure 2.2 are those to guarantee *authentication*, *authorization* and *access control*. Such services will be fundamental, as not all services provided by an Internet-integrated WSN in the context of the IoT will be public, and some applications may require that accesses to data available on sensing devices be carefully controlled. Mechanisms for control of accesses may be designed to operate on packet header information related with 6LoWPAN-based communication protocols, as this would enable a fine-grained control of communications between the Internet and WSN domains. In the same context, compressed 6LoWPAN security headers, DTLS headers and CoAP security options can be inspected and processed in cooperation with security-mapping and key management mechanisms.

The creation of a worldwide object network will require a security infrastructure to support mutual object authentication and operations related with identity management, anonymization, authentication and authorization. While not all IoT applications will require or be able to access such an infrastructure, research and standardization work will be required for its design and integration with current certification infrastructures. Authentication and authorization mechanisms will also be dependent on the adoption of suitable and scalable identification mechanisms to provide unique identifiers and virtual identifiers to users, sensors and other types of devices [154][155].

As with any other Internet device, it is fair to expect that a low-power WSN sensing device exposed to Internet communications will be targeted by malicious entities trying to hinder the availability of its services. In this context, *fault tolerance* in Internet-integrated WSN

devices may involve making all objects secure by default, giving all objects to know the state of the network and its services, and making objects able to defend themselves against network failures and attacks. Despite such desirable properties, the employment of a security gateway as in the reference integration architecture illustrated in Figure 2.2 will be important and in many situations may be unavoidable, as the gateway may provide valuable support in the enforcement of appropriate security perimeters.

Other fundamental aspect related with fault-prevention is intrusion detection. Despite the existence of preliminary works on intrusion detection systems for 6LoWPAN WSN environments [156], further research still needs to be performed in this area. Intrusion detection mechanisms can be extended to understand possible attacks against 6LoWPAN-based communications and security technologies, and be developed symbiotically with other mechanisms required to guaranteeing the availability and robustness of the WSN, such as load balancing.

2.4 PROPOSALS ON SECURITY FOR OTHER WSN INTERCONNECTION APPROACHES

In our following discussion we focus on alternative approaches to support communications between the WSN and Internet domains and on how security is addressed in this context. These interconnection approaches provide alternatives to the full integration via the usage of Internet WSN communication technologies, which we have previously analyzed.

2.4.1 INTEGRATION VIA CLOUD-BASED TECHNOLOGIES

The analysis of the current approaches to integrate WSN environments with the Internet [157][158] also enables the identification of the open issues regarding security in its context. In practice, different integration strategies may serve different applications and approaches. For example, the best approach in offering services supported by sensing and control devices in a SCADA (Supervisory Control and Data Acquisition) industrial control network may be to support indirect accesses to data in such services, via a gateway that may also enforce adequate security controls. Other applications may benefit from more direct communications between wireless sensing devices on different WSN domains or between such devices and backend/Internet hosts. In practice, the various integration approaches also correspond to different degrees and strategies of integration of WSN communications with the Internet communications infrastructure.

A currently popular integration approach is via cloud-based web services [159][160]. Proposals in this category usually offer a platform as a service, in which the user may be able to customize the tools at its disposal with the goal of building a custom product. The main goal of the cloud-based integration approach is to enable the usage of high-performance computing and storage facilities in the processing of sensing data retrieved from WSN devices. This approach may enable applications targeting diverse areas and providing advanced analysis tools, for example based on business intelligence algorithms. This

approach is greatly motivated by the easiness of quickly developing specialized products, and part of its success is due to the fact that it hides the communication technologies employed in the WSN domain from the outside.

Existing cloud-based proposals and products usually employ tailor-made middleware solutions and Application Programmer Interfaces (API) designed according to the Service Oriented Architecture (SOA) principles. The middleware simplifies the development of new applications, since it abstracts applications from the characteristics of the sensing devices and the complexity of the WSN communications. The data gathered from WSN sensing devices may be uploaded to cloud-based servers via a gateway, a device that may also support operations related with data aggregation, protocol translation, remote management and security, among others.

Other characteristic of proposals following this integration approach is the virtualization of physical sensors, which enables a single physical sensor to be used by multiple applications. A virtualized sensor abstracts the physical device from its particular characteristics, capabilities and location, and cloud-based proposals may also support mechanisms to manage the service infrastructure and to publish the services available on the various devices using service templates [160].

Various research and industry proposals may be identified that materialize this integration approach. One example may be found in Xively Could Services from LogMeIn [161], a product formerly called Pachube, which consists in an IoT cloud service providing web-based tools and developer resources to facilitate the development and deployment of connected products using heterogeneous services. Another is SensorCloud [162] from MicroStrain, a cloud-based data storage, visualization and remote management platform supporting user-programmable data analysis via a math engine also supported by a specialized cloud application. SensorCloud may also be complemented by specialized WSDA (Wireless Sensor Data Aggregator) gateways supporting data aggregation and remote management of the devices and data.

SensaTrack [163] offers a turnkey solution for monitoring services and supports a large variety of sensors and mobile devices, together with gateways supporting backhaul Internet communications using CDMA (Code division multiple access), GSM (Global System for Mobile Communications), Ethernet and WiMax communications. A free cloud-based service is proposed by Nimbits [164], which may be used to record and share sensing data on the cloud using a free service, and also provides a server platform available for users to deploy applications on their own servers. In Nimbits the user creates data points in the cloud to which the sensing data is sent to trigger diverse types of calculations, alerts and statistics. Also in the context of free-based services, ThingSpeak [165] is an open source IoT application offering an API to store sensing data on the cloud, and also to support numeric data processing operations on the data.

Despite the advantages of this integration approach, such proposals do not contribute to the evolution of the Internet communications and security infrastructures to encompass transparent communications with WSN devices, which also provide the context of our research efforts. In such proposals, wireless sensing devices aren't able to communicate directly with external entities and WSN environments may even employ proprietary communication technologies. Aspects such as the mobility of the sensing devices in the context of future IoT applications must also be considered, and are not targeted by such proposals. Cloud-based integration proposals do support a practical and effective strategy for the gathering and processing of sensing data, but do not promote a technological basis capable of supporting richer communication patterns involving wireless sensing devices. Regarding how the previous proposals address security, we are able to observe that they lack, in general, important security mechanisms and assurances, as we proceed to analyze.

In Xively [161] devices write and read data from cloud-based applications using various API provided by the platform. One security service provided in this solution is the secure provisioning of devices for their initial boot up in the context of a given application. Each device is provisioned with a Feed Identifier and an API key to be able to send data to the cloud-based application, after contacting a device activation API. In order to obtain the Feed ID and API key, the device submits a secure activation request constructed by producing an HMAC-SHA1 hash of the device's serial number plus a secret key associated with the application in the context of which the device is being activated. After this initial procedure, the Feed Identifier and the API key are stored also on the application. At the end, the feed ID enables the device and the application to communicate with each other and with Xively. Xively also employs keys to control accesses to all API resources. A key may be associated with a particular permission of accessing a resource or feed, also by a particular user or machine with a particular IP address. Keys are sent in API requests, either in the HTTP request header or as part of the URL. Of course, the usage of keys in this way is inherently insecure if not using encryption, and Xively also supports TLS/SSL to support end-to-end security for communications between sensing devices and the cloud servers, while we must notice that HTTPS is optional.

The support of TLS/SSL in SensorCloud [162] is mandatory across the platform, including for data uploads and downloads via HTTPS, as in the previous proposal. The platform also provides mechanisms to identify the entities that are authorized to access sensing data stored on the cloud, in the context of a given application. All sensing data is private by default, and data owners can also send invitations to other users they want to bring to the application, for example to assist in analyzing and building custom-tailored data processing applications. SensaTrack [163] provides mechanisms for the setup of user accounts and corresponding security access privileges, and some of the provided gateways also support IPSec VPN accesses to the cloud servers.

Regarding free cloud-based integration solutions, Nimbits [164] also supports HTTPS protected requests to web services. Access keys can also be created and employed in access

URLs to get access to protected resources. The administrator of a given application may create a key and associate it with a particular data point or with all of his data points. Access permission may also determine read-only accesses, rather than read and write accesses.

ThingSpeak [165] supports management of privileges to control accesses to data, as well as to define who is able to build and use applications, providing control of accesses to data and applications considered private. In this proposal data channels are used to store and retrieve data, and each channel has private and public views. Accesses to the private view are controlled via authorized accesses to the web server, while the public view is what other viewers see when they visit the channel. The administrator of a channel is able to define the information that is available on each view, customize the view with plugins, or even disable the public view. Accesses to resources may also be controlled via write or read API keys. By default, a channel is private and requires a read API key to access its feed. ThingSpeak also supports HTTPS accesses to API web services.

Table 2.3 resumes the main characteristics in terms of security of the previously analyzed WSN cloud-based integration proposals. Based on the previous discussion and in the resume provided by this table, we are able to observe that the security properties provided by such solutions are focused on the support of secure communications between the WSN gateway and the cloud-based web server, and on security mechanisms designed in the context of the supported cloud-based services. In the context of such services, security may involve the usage of access control mechanisms to control accesses to web services and to the API via security keys.

Table 2.3 - Security mechanisms on cloud-based integration proposals

Proposal	Secure communications between gateways and cloud services	Secure provisioning of devices	Access control mechanisms	API Security
Xively [26]	Optional (TLS/SSL)	Yes (authentication hashes)	Yes (by user and IP client address)	Yes (using access control keys)
SensorCloud [27]	Yes (TLS/SSL)	No	Yes (by user)	No
SensaTrack [28]	Optional (IPSec VPN)	No	Yes (by user)	No
NimBits [29]	Optional (TLS/SSL)	No	Yes (by user and data points)	Yes (using access control keys)
ThingSpeak [30]	Optional (TLS/SSL)	No	Yes (by user and public or private keys)	Yes (using access control keys)

We may also observe that security aspects such as privacy, trust and anonymity are not addressed in the previously discussed cloud-based proposals, nor is security in the context of end-to-end communications with wireless sensing devices. We proceed by analyzing proposals for the integration of WSN with the Internet via front-end gateways.

2.4.2 INTEGRATION VIA FRONT-END GATEWAYS

One initial integration approach reflected in literature proposals consisted in the employment of a specialized gateway operating as a front-end proxy for services available on the WSN domain, while isolating and abstracting WSN communications from the Internet. In such proposals the gateway offers the services of sensing devices to the outside, particularly at the application-layer via Web Services (WS). We may thus observe that such proposals precede the previously analyzed integration approach via 6LoWPAN-based communication technologies.

Considering how the proxy obtains data from sensing devices, two main strategies are considered by research proposals in this category. One consists in the data being obtained from a sensing device upon the arrival of a request from an Internet client. In this situation the data may also be cached at the proxy, if required. The other is to employ a subscription and push protocol that enables sensing devices to update sensorial data on the proxy upon changes on the measured physical variable. As we observe next, despite not supporting direct communications between the WSN and Internet communication domains, research proposals in this category have also pioneered the idea of employing web services based on the Representational State Transfer (REST) architecture, to support communications for data available on WSN devices.

An initial research proposal in this integration category is discussed in [166][167], in which an architecture is proposed where embedded sensing devices support web services and the HTTP protocol, although communications on the WSN domain do not run over IP, rather over a proprietary communications protocol. This architecture is more recently discussed and evaluated in greater depth in [168]. Another proposal in this category is SenseWeb [169] from Microsoft Research, which supports multiple gateways serving different WSN islands. SensorMap [170] is a practical implementation of the SenseWeb architecture, and mashes up sensing data from multiple sources on a map interface and provides interactive tools to selectively query sensors and visualize the data. SensorMap also supports authenticated accesses to sensor management functionalities. It is also interesting to observe that, considering that SenseWeb and SensorMap support a platform to share and support computations over data obtained from WSN devices, in this sense precede the more recent cloud-based integration proposals that we have previously analyzed.

In [171] the authors propose the integration of a WSN with the Internet also via a WS API supported by a front-end proxy, which supports virtual counterparts on the Web of WSN physical sensing devices. In this proposal the authors also discuss the advantages of supporting WS directly on the sensing devices in the future. Other research proposal is [172],

which proposes the interconnection of a WSN with the Internet via mobile communication networks, in particular using General Packet Radio Service (GPRS) communications. This proposal also employs a specialized gateway in the support of mechanisms for protocol conversion and control of WSN sensing devices.

As previously discussed, the front-end proxy integration approach may enable the indirect integration of WSN with the Internet at the application-layer, particularly via a WS API interface. Thus, from the point of view of entities external to the WSN, this approach enables a standard communications interface, despite the fact that on the WSN domain communications are possibly supported by proprietary technologies. From the point of view of security, this integration approach offers the immediate advantages of isolating WSN communications from Internet-originated threats and attacks, as in fact WSN applications delegate all Internet-related communications to the front-end proxy.

In respect to security, the gateway may also behave as a normal Internet citizen, and in consequence support standard Internet security mechanisms to protect communications with Internet hosts. A characteristic that is shared with cloud-based integration proposals is that communications are not extended to the WSN domain, and as such end-to-end communications with WSN devices are not supported.

We observe that most of the previously discussed research proposals are not focused on security. As previously discussed, research proposals using web mashups [166][167][168] focus on device abstraction and on making sensing data available via a simplified web services API, while not addressing particular security threats nor the design of security mechanisms. SenseWeb [169] identifies the importance of addressing security issues, as the trustworthiness of the data, the privacy of the users and the reliability and verifiability of the shared data against malicious intervention or inadvertent errors. The authors also discuss the challenges of addressing security and trust, and of building a sensing infrastructure out of shared resources, while doesn't proposing or defining any specific mechanisms to target such aspects. In [171] and [172] the authors discuss the interest of designing security management functions for the proposed IoT gateway in future work, while not proposing any specific solution.

In general, we may observe that the exploratory nature of research proposals in this category motivated a focus of researchers primarily on the communication aspects of the proposed solutions, rather than on security. Nevertheless, as previously discussed such proposals have provided an important contribution in paving the way to the acceptance of the viability of the interconnection of constrained low-power WSN with the Internet, even if in this context indirectly via services supported by a front-end proxy.

2.4.3 ARCHITECTURE FRAMEWORKS

Various research projects on WSN include in its goals the design of architecture frameworks for the support of WSN applications integrated with the Internet. Such projects implement

different strategies to enable communications between separate WSN domains over the Internet, while do not focusing necessarily on the employment of WSN Internet communication technologies. As such, the employment of interconnecting gateways and of specialized middleware layers abstracting operations on sensing devices and data from the particularities of WSN communications may also be found in the proposals in this approach. We may also observe that the technologies proposed in the context of such architectures are primarily focused on the support of complex applications over distributed WSN domains. A consequence of this design approach and of the purpose of such architectures, security is designed according to the particular goals of the project and not to support Internet WSN communications.

One important architecture framework in this category was proposed in the context of the SENSEI EU FP7 research project [173]. SENSEI targeted the design of an architectural framework and of related technological solutions to enable the easy plug and play integration of distributed WSN domains into a global system, while providing support for fundamental operations such as network and information management, security, privacy, trust and accounting. In order to enable interoperability of sensing devices on different WSN domains, SENSEI supports REST communications in the WSN parts of the system. An extensive set of security mechanisms were also designed in this project, namely to support secure code updates, jamming mitigation, secure routing, and detection of node capture and replication. The SENSEI architecture also supports the employment of a trusted hardware component to defend against a broad range of security threats resulting from compromise attacks, and introduces the middleware component FAIR [4], which supports resilient in-network data processing.

The SENSEI architecture [173] introduces the notion of a community, which is formed by various actors taking up one or more business roles. Actors may be resource providers (the owners of the resources), framework providers (the owners of framework components), service providers (the owners of the services that use the resources and support services), or resource users (who are the main users of such resources and services). The proposed framework also offers community management functions, which include user account management, identity management, security and privacy functions, among others.

In order to support secure interactions between different entities of SENSEI, the architecture supports authentication, authorization and accounting (AAA), as well as privacy and trust management mechanisms. In particular, the AAA component of SENSEI supports a security token service (STS), which provides entities with the security assertions (tokens) required to access resources on the network. The auditing and billing service supports accounting in the context of the AAA architecture, while the resource access proxy service supports authentication, token request and resource access on behalf of the user.

Regarding privacy, the SENSEI architecture addresses real world privacy issues and electronic privacy issues. The former includes the privacy of personal information collected by sensors,

and access to this information is controlled by use of the AAA architecture previously described. Electronic privacy issues include people leaving digital traces of their movement and actions in various places, and the architecture provides a range of features to allow users to control how difficult it is to link their traces to them, for example the use of pseudonyms or attributes instead of recognizable identities. The SENSEI research project also produced work regarding the secure programming of sensing devices [174][175], [176], resilience in-network data processing [4] and mechanisms against capture attacks [177].

In the SmartSantander [178] project a city-scale experimental research facility is being enabled to support applications and services for a smart city. This project builds on results from SENSEI [173] and on the WISEBED test bed facilities [179]. WISEBED is a research effort of nine academic and research institutes across Europe, aiming to provide a multi-level infrastructure of interconnected test beds of large-scale WSN for research purposes. The architecture considered in the SmartSantander project supports the controlling of sensing devices through a set of low-level API, the running of experiments through a web portal and the support of applications using web services.

One of the goals of the SmartSantander [178] project is to implement and evaluate security as one of the key building blocks of the IoT architecture. The architecture currently being designed includes security requirements related with the AAA (authentication, user account management and authorization) model. Trust and privacy requirements are also being considered in the context of session management in test bed servers, gateways and sensing devices. Researchers may access the test bed provided by the project via a specialized web portal, and the control of authorizations and accesses to the test bed is supported both by this portal and in the set of low-level API supported by the architecture. The administrator of the experimental facility will be able to grant and revoke user access privileges. As we have previously discussed, SmartSantander is an ongoing research project and as such work related with the design of appropriate security mechanisms is ongoing and results may be expected in the future.

Other project relevant in this context is the IoT-A project [180], which builds on the results from the previous projects and targets the design of an architectural reference model for the interoperability of IoT systems. Among the goals of this project are the outline of principles and guidelines for the technical design of protocols, interfaces and algorithms, and the design of mechanisms for the integration of the proposed architecture into the service layer of the Internet of the future. Also, the project includes the design of a novel resolution infrastructure, of novel platform components and the experimentation of the proposed mechanisms using real implementation scenarios.

The main results of the IoT-A [180] research project in terms of security are related with the resolution infrastructure that is being designed to allow scalable look up and discovery of IoT resources, entities, and their associations. Mechanisms are being designed to support privacy and security in the resolution infrastructure. The original architecture was extended

with a security component to ensure privacy and security for the resolution functions, as well as to offer the basis for other security functionalities outside the resolution infrastructure.

Table 2.4 - Security properties of integration architecture frameworks

Architecture framework	User and privilege management	Privacy and trust management	Authentication, Authorization and Accounting	Other security properties
SENSEI [38]	Yes	Yes (privacy via user pseudonyms)	Yes (via security tokens)	End-to-end security; secure device reprogramming; secure data aggregation; resistance against sensing device capture
SmartSantander [40]	Yes	Yes (trust and privacy planned for all components)	Yes	End-to-end security
IoT-A [42]	Yes	Yes (user privacy via pseudonyms, privacy on resolution mechanisms)	Yes	End-to-end and hop-by-hop authentication and security; key exchange and management; reputation management.

A set of components are introduced in the IoT-A architecture to support security, namely an authorization component to perform access control decisions based on access control policies, an authentication component, an identity management component that manages pseudonyms and accessory information to trusted subjects so that they can operate anonymously, and a key exchange and management component. The IoT-A project is also designing a trust and reputation architecture and the relationships of the various security-related components to the other mechanisms of the architecture.

Table 2.4 resumes the main characteristics in terms of security of the previously discussed proposals, which have as its main goal the support of complex services and applications based on distributed WSN domains. Rather than designing mechanisms to enable Internet communications over such domains, the proposed architectures again employ specialized middleware approaches and Internet communications are employed only to support communications between gateways interfacing with the various WSN islands.

As previously discussed, the main focus of the research proposals presented in this thesis is on security for communication technologies being designed to enable Internet communications on WSN environments, thus based on the 6LoWPAN communication technologies previously discussed and contextualized by the reference protocol stack illustrated in Figure 2.2.

3 A REFERENCE MODEL FOR END-TO-END SECURITY[2]

This chapter reflects our contribution in the context of the GINSENG research project [1], and presents the reference model for end-to-end security considered in the subsequent chapters of the thesis. This model enables the employment of the various research proposals in the context of end-to-end communications using 6LoWPAN-based communication technologies, which we analyzed in the previous chapter.

As previously observed, our initial research efforts in the context of the GINSENG research project later evolved to the consideration of the usage of communication technologies currently being designed to enable Internet communications on WSN environments. This change in context was also motivated by the perceived importance of such communication technologies for the enabling of future Internet sensing applications, for which security will be fundamental. We also note that the reference model for end-to-end security was from the start designed to support such communication technologies.

In our following discussion we begin by analyzing our preliminary research efforts in the context of GINSENG, particularly the identification of its main requirements for security, the consideration of security as a performance metric and the proposal of application security profiles. We also discuss our preliminary approaches on the addressing of security at the MAC layer designed in this research project. Such contributions provided the ground for the design of our model for end-to-end security, which we discuss later in the chapter. Also, notions such as security metrics and application security profiles have been considered firstly in the context of GINSENG and are also present in our evaluation strategy throughout the thesis. We also discuss the methodology considered in the following chapters for the experimental evaluation of the various research proposals, as well as how applications may statically or dynamically reconfigure end-to-end security.

3.1 SECURITY IN PERFORMANCE-CONTROLLED WSN ENVIRONMENTS

The goal of the GINSENG research project [1] was the design of technologies to enable the employment of wireless sensor networks that meet application-specific performance

[2] *This chapter has supported the following publications:*

- Granjal J, Monteiro E, Silva J. *A framework towards adaptable and delegated end-to-end transport-layer security for Internet-integrated Wireless Sensor Networks*, Second Joint ERCIM eMobility and MobiSense workshop, WWIC 2013
- Granjal J, Monteiro E, Silva J. *End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication*, IFIP Networking 2013
- Granjal J, Monteiro E, Silva J. *A secure interconnection model for IPv6 enabled Wireless Sensor Networks*, IFIP Wireless Days 2010

targets, while also focusing on the integration of such technologies with industry resource management systems. As previously discussed, the following discussion reflects our contribution in the initial stages of this research project.

3.1.1 TIME-CRITICAL DATA COMMUNICATIONS IN WSN

The GINSENG approach to performance-controlled communications in WSN environments considers that such networks are dimensioned, deployed and operated in order to ensure reliable and timely data delivery. In this context, one important component of GINSENG is the GinMAC [1] MAC layer, which is single-channel and operates in a TDMA fashion. GinMAC is employed with off-line dimensioning, which allows a transmission schedule to be planned for the whole network before network deployment [181] in order to ensure timely and collision-free data communications. GinMAC also employs reliability control mechanisms during network operation, to cope with the fluctuating characteristics of the wireless channel and still guarantee transmissions according to predefined delay bounds. This applies to upstream communications (from sensing devices to the sink) and also to downstream communications (from the sink to actuators).

Another feature implemented in GinMAC is topology control, which enables a node to determine in which slots it may become active. A new node starts by obtaining time synchronization by listening to the messages transmitted in the network, and this also allows the new node to find its position in the topology, since data packets in GinMAC transport information about available positions. The new node can thus claim an advertised position by transmitting a data packet in the slot allocated for this purpose and, after receiving the corresponding acknowledgement, can start using this position in the topology for normal communications.

The slots forming a GinMAC frame may be of types basic, additional and unused. Basic slots are attributed exclusively to a given node and are defined such that within the frame the node can forward one message to the sink or the sink can transmit one message to a given actuator. Additional slots are used to improve transmission reliability, by supporting temporal and spatial transmission diversity. Such slots are added in the schedule directly after the respective basic slots for the upstream and downstream directions. If a node fails to transmit data in a basic slot, it can use an additional slot for a retransmission. To determine the number of additional slots needed for reliability control, the worst-case link characteristics may be obtained from measurements in the target deployment area before deployment. Finally, unused slots may be added to improve the duty cycle of nodes, enabling nodes to turn the transceiver off for the duration of these slots. The TDMA slots in GinMAC are fixed in size and large enough to accommodate a data transmission of a maximum length and an acknowledgement from the receiver. As we observe later, our proposal for the design of security in GinMAC involves modifications to how transmission slots are employed to support communications at the MAC layer.

3.1.2 REQUIREMENTS FOR SECURITY

Our approach to security in the context of GINSENG considered that appropriate mechanisms should be developed to support fundamental security properties for WSN communications, and also that security can benefit from the availability of deterministic communications at the MAC layer. The fundamental security requirements applicable to WSN communications in the context of GINSENG applications are those of *confidentiality*, *integrity*, *non-repudiation* and *authentication*. On the other hand, the usage of a performance-controlled MAC enables the reservation of communications bandwidth for security *a priori*, and also to determine and control the energy required for security operations accordingly to each application scenario.

Regarding the support of fundamental security requirements, applications should be able to enable *confidentiality* for all communications through the encryption of data reported from sensor nodes and also of data sent from the sink to actuators, thus enabling security against eavesdroppers. Regarding *authentication*, a sink device must be able to authenticate that communications are from a particular sensor node and, similarly, a sensor node must be able to authenticate that a packet arrives from a particular sink node. Authentication mechanisms may thus offer protection from both sensor node and sink node spoofing. Communications in the WSN may also be protected in terms of *integrity* and *non-repudiation*, meaning that the sink node and the sensing devices must have a mechanism to allow the verification that each data packet has not being modified in transit. We may also note that the previous security requirements are general enough to also apply to the research proposals targeting end-to-end communications in Internet-integrated WSN environments, which we discuss throughout the thesis.

Other requirements were also considered in our previous approach to security in GINSENG, which also influenced the design of the reference model for end-to-end security discussed later in the present chapter. One important aspect to consider is that security mechanisms may be designed to support heterogeneous sensing devices, and when appropriate support the *delegation of costly security operations* from very constrained devices to more powerful network entities. For example, the initial authentication and key agreement phase is a particularly costly phase of end-to-end security protocols, and may thus benefit from such an approach.

Other than end-to-end security, the enforcement of *security perimeters* is also a desired property, as it may provide security for WSN devices and communications against external threats. Security perimeters may be enforced with the help of mechanisms designed for *intrusion detection* and *control of accesses* via *filtering* or *front-end proxies*, for example. As sensing applications can in practice employ various WSN domains, we may also consider the *mobility* of sensing devices between different domains, from the point of view of end-to-end communications. Finally, *resilience* of critical communications and network operations against security attacks is a desired property.

The previously discussed security requirements may apply, on the one end, to communications at the MAC layer using GinMAC, and also to communications at upper layers of the stack in the context of Internet-integrated WSN, which may thus support end-to-end communications between WSN devices and external or Internet devices. The broad scope of the previously identified security requirements is also motivated by our focus on 6LoWPAN-based WSN communication technologies, which were considered from the start of our research efforts.

3.1.3 SECURITY AS A PERFORMANCE METRIC

The GinMAC layer is designed in such a way that specific bounds are defined for message transport delay and reliability. Our approach to security in this context considered that security could also beneficiate from such an approach, since the energy and communications bandwidth required for security operations may be taken into account in the design of the GinMAC MAC layer and when planning for the duration of TDMA epochs. We may thus consider that security constraints (or required resources) are measurable metrics, in the sense that the energy, computational effort and time required for security can be considered in the dimensioning phase of the network, and be measured to ensure proper operation during the functioning phase. In this context, the ability to cope with security in a performance controlled network requires firstly the availability of proper security metrics, and secondly the existence of procedures to measure those metrics. Such metrics can either be calculated directly from data received from sensing devices by the sink node, or inferred by combining various data values or metrics.

With this approach, security can be monitored side-by-side with other performance-related metrics, in the sense that security metrics may allow to discern the effectiveness of various components on security and also to measure the level of risk in not taken a specific (corrective) action, and also in prioritizing corrective actions. For example, the decision of selecting appropriate cryptographic components and appropriate usage and configuration parameters for those components may be supported by appropriate metrics. In this context, quantifiable measurements of how much specific security attributes (or combination of attributes) an entity possesses may be considered. A security metric can be measured also from lower–level physical measures, such as the behavior of a specific cryptographic protocol with pre-defined characteristics and configuration under specific types of security attacks. Considering security to be a quantifiable measure allows us to obtain deterministic feedback for the behavior of the network in different application scenarios and environments.

The usage of security metrics also enables the possibility of defining specific levels of security, and the establishment of compromises between security and aspects such as reliability, the lifetime of applications or the maximum communications rate, as appropriate for particular deployment scenarios. Therefore, with this approach we may manage the interdependence between the level of reliability and how it influences the configuration and

operation of security mechanisms such as key management and the cryptographic algorithms employed. In the context of GINSENG, reliability and security can thus be considered two interrelated and orthogonal requirements. Our approach to security thus considers that in a WSN where performance and reliability is guaranteed we must also include the definition and testing of security metrics mechanisms that are able to guarantee the desired level of security. Security metrics can thus appear as one of the key aspects to guarantee specific application service bounds in respect to reliability.

3.1.4 APPLICATION SECURITY PROFILES

Application security profiles allow the definition of quantifiable security parameters that have a direct impact on the energy expended with security operations, and allow the quantification and control of the energy required for security operations accordingly to each deployment scenario. Application security profiles are also related to mechanisms that can adapt to the security requirements of specific deployment environments. This means that we must be able to adapt operational aspects that have a direct impact on the energy expended with security operations, for example the cryptographic algorithms employed and its corresponding key size. The main goal of application security profiles is to define specific functional security parameters that determine how security is implemented on each specific deployment scenario. Our previous approach to application security profiles in the context of GINSENG considered, in particular, the following functional parameters:

- The security algorithms employed in order to guarantee confidentiality, non-repudiation, integrity and authentication of the communications and of the communicating parties.
- The frequency of refreshment of the key employed with each cryptographic algorithm, as appropriate.
- The size of the cryptographic keys employed with each cryptographic algorithm.
- The security metrics to consider for the measurement and monitoring of security, as applicable to a particular sensing application.

Our original approach to application security profiles considered the definition of a security matrix quantifying each of the previous aspects for each application. Overall, the main goal of application security profiles is to have a mechanism that allows us to quantify the requirements of the applications in terms of the resources required to support security in constrained wireless sensing platforms. Application security profiles are also considered in the context of the research proposals for end-to-end security described throughout the thesis, related to the necessity of properly measuring the impact of the proposed research solutions on the critical resources of wireless sensing devices.

3.1.5 SECURITY APPROACHES FOR GINMAC

Our research in the context of GINSENG enabled the identification of preliminary approaches for the integration of security in GinMAC, as we proceed to discuss. Modifications were proposed with the goal of having security benefit from the presence of a MAC layer with deterministic behaviour, since it may facilitate the quantification of the overall impact of security on the lifetime of the sensing devices and applications, thus supporting the definition of appropriate security levels. If accompanied by appropriate monitoring mechanisms, it may also support the detection of failing or misbehaving devices.

For the support of security in GinMAC, we proposed the usage of two slots in each TDMA epoch reserved for security operations. The goal with this approach is to guarantee the availability of communication slots during which security operations can be performed. Such operations are related to security management, for example data communications in the context of key management or intrusion detection operations. In each epoch, the device that is authorized to use the security slot to upload or download a security message is determined accordingly to the node's identification. One slot may be reserved for upstream security management communications, and the other for downstream security management communications. We also consider that security-related communications can be transmitted in broadcast mode from the sink to all sensing devices in the network. When not using the security slots, a node listens in case some other node or the sink node transmits a security management packet with its destination address or in broadcast mode. When transmitting in such slots, the node can send a security management packet to a specific sensor node or sink node, or transmit in broadcast mode. The reservation of slots for security provides the benefit of supporting security management communications without interfering with normal data transmissions. Thus, security management operations may be defined considering the available communications bandwidth without ever compromising normal data communications.

Another aspect to consider is that the employment of cryptographic algorithms should not compromise performance-controlled communications, meaning that for example the time required to perform encryption or decryption on a message should not compromise its timely transmission from a sensor node to the sink, or from the sink to a sensor node. In general, security algorithms must be adopted that still allow the transmission of a sensor reading to the sink in one TDMA epoch, as per the goals of GINSENG. Our proposal was for security to be processed for a packet before its transmission, more precisely using a pre-slot time window reserved for message pre-processing in the TDMA epoch defined in GinMAC. This time window may thus support the computational time required for encryption and generation of MAC code for the message. Similarly, decryption and verification of the MAC code can be performed using the post-slot time-window reserved for post-processing in the same epoch.

During the pre-planning phase of the network the computational time required to support pre-processing and post-processing of security may be taken into account, according to the security algorithms employed by each application. Security may thus be considered when planning the duration of a slot, and consequently the TDMA epoch may be dimensioned considering the need to perform security operations for each forwarded packet. The consideration of security during the dimensioning phase may thus enable a performance-controlled MAC layer with deterministic security. We also note that our proposal for the addressing of security in the context of GinMAC also involved the adaptation of the GinMAC header in order to identify the presence of security. This enables the identification of the type of security applied to a given message, and also of messages transporting security management information.

3.2 A REFERENCE MODEL FOR END-TO-END SECURITY IN INTERNET-INTEGRATED WSN

As previously discussed, our initial approach to security in the context of the GINSENG research project enabled also the identification of strategies to address security in the context of Internet-integrated WSN environments, which was a major goal of our research efforts from the start. We proceed by discussing our reference model for end-to-end security in Internet-integrated WSN, together with the operation of its main components.

3.2.1 FUNCTIONAL OVERVIEW OF THE REFERENCE MODEL

In Figure 3.1 we provide a functional perspective of end-to-end security using the reference integration model considered in the thesis. As illustrated, we consider that a sensing application may encompass multiple WSN domains, which are interconnected via specialized gateways. The gateways route traffic between the WSN domains and the Internet, and also provide strategic places for the usage of specialized security mechanisms, as we discuss in later chapters. Such devices are assumed to be without the constraints of WSN devices in terms of critical resources such as energy, memory and computational power.

The other components of the model are a Certification Authority (CA) and an Access Control (AC) entity. The CA manages identification and certification information for the entities participating in end-to-end communications, namely external hosts, security gateways and AC servers on the WSN domain. As illustrated in Figure 3.1, an AC server is employed in each of the various WSN domains to support authentication and authorization in the context of end-to-end communications involving WSN devices on the correspondent domain. Such communications may take place between devices on different WSN domains or between WSN devices and (external) Internet hosts. We also assume that communications between WSN gateways and AC servers may use the backhaul communication technologies previously discussed in Chapter 2, thus not being limited to low-power WSN wireless communications. Such communications may support security-control messages, in the context of mechanisms such as key management, intrusion detection and control of accesses, among others.

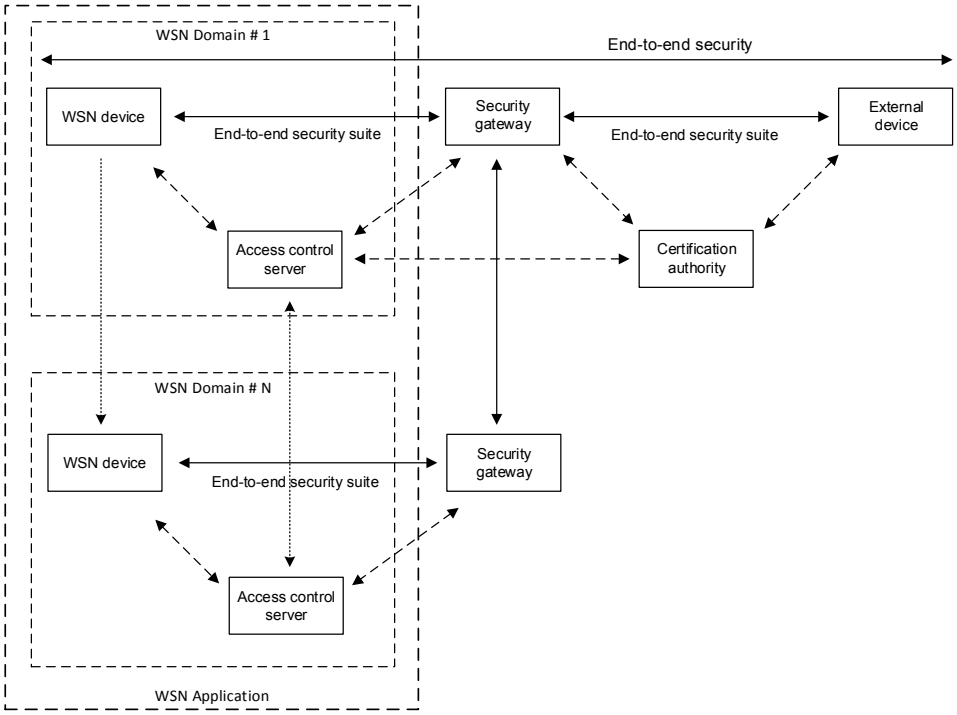

Figure 3.1 – Functional overview of end-to-end security with Internet-integrated WSN

In line with classic WSN deployment scenarios, security gateways can also operate as sink nodes for the corresponding WSN domains. WSN domains operate with IPv6 communications and addresses accordingly to the rules defined in 6LoWPAN [92], and gateways may support both IPv6 and 6to4 tunneling on the external interface. As described in the literature, WSN gateways also assume the role of 6LoWPAN border routers (6LBR), thus supporting the mechanisms required for communications to run between the WSN and Internet domains. As also illustrated in Figure 3.1, the gateway may transparently intercept and mediate security between the two communicating parties, while supporting costly security operations on behalf of constrained WSN devices.

The AC server may store security information related with the various WSN communicating entities (sensing devices and 6LBR), which may include cryptographic keys identifying such devices and rules for controlling accesses to resources on the WSN domain. This information may support authentication and authorization in the context of mechanisms designed to guarantee appropriate end-to-end security in Internet-integrated WSN, as we explore later for transport-layer security. In the system model of Figure 3.1 we also consider that WSN devices may roam between WSN domains and, in this context, that trust relationships between AC servers on different domains may support transparent mobility from the point of view of end-to-end secure communications. Finally, we also note that in other deployments end-to-end secure communications may be established without the usage of an intermediary, in a truly end-to-end fashion. Without the support of mobility, in such situations the AC server may not be required to support security.

3.2.2 OPERATIONAL COMPONENTS OF THE REFERENCE MODEL

The optimization of communications and security mechanisms according to the scarcity of the energy and other resources available in most WSN sensing devices has motivated most of the research efforts in the field of WSN [7] in the past. An aspect motivated by this fact is that traditional communication architectures proposed for WSN usually collapse networking layers in order to optimize energy usage. Many proposals have been presented where the layers have been turned upside down, intermingled, or where the network itself processes the data produced by the sensor nodes. Such proposals are usually optimized for isolated and specific deployments, and are usually closed and non-scalable. As we have previously discussed, the research community is more recently leaning toward more layered and open network architectures, and particularly IP, also due to the added benefits of modularity, separation or concerns and global Internet communications that it supports.

Although still not consensual, the employment of IP on WSN environments presents several benefits. The IP Protocol enables a common network-programming interface to support communications between sensing devices or between such devices and external or Internet hosts. IP may also facilitate the integration of existing applications with sensing devices, and is able to bring physical sensing capabilities to the Internet as we know it today, thus enabling WSN as an important component of new ubiquitous and heterogeneous communication environments. Even for WSN that do not require direct integration with the Internet communications infrastructure, IP can enable ubiquitous communications between heterogeneous sensor nodes.

As we have previously observed, 6LoWPAN-based communication technologies may support the integration of WSN with the Internet via the employment of Internet communication technologies in such environments, consequently enabling sensing applications to transparently appear as part of the Internet communications infrastructure. The fact that many aspects are still open in how to address security in the context of this integration approach motivates our usage of the reference integration model previously described, as well as of its functional components, which we proceed to discuss.

Figure 3.2 illustrates the operational components of the considered reference model. This figure considers the employment of the model both in a WSN Gateway (6LBR) and wireless sensing devices. The previously discussed WSN Internet communication technologies are employed side-by-side with specific management and security-related components, which are considered to be transversal to end-to-end communications taking place at the various protocol layers. We also illustrate interactions between such components. The research solutions presented in the thesis belong in the context of end-to-end communications taking place at the various protocol levels of the stack illustrated in Figure 3.2 and employ one or various of the functional components considered in this model.

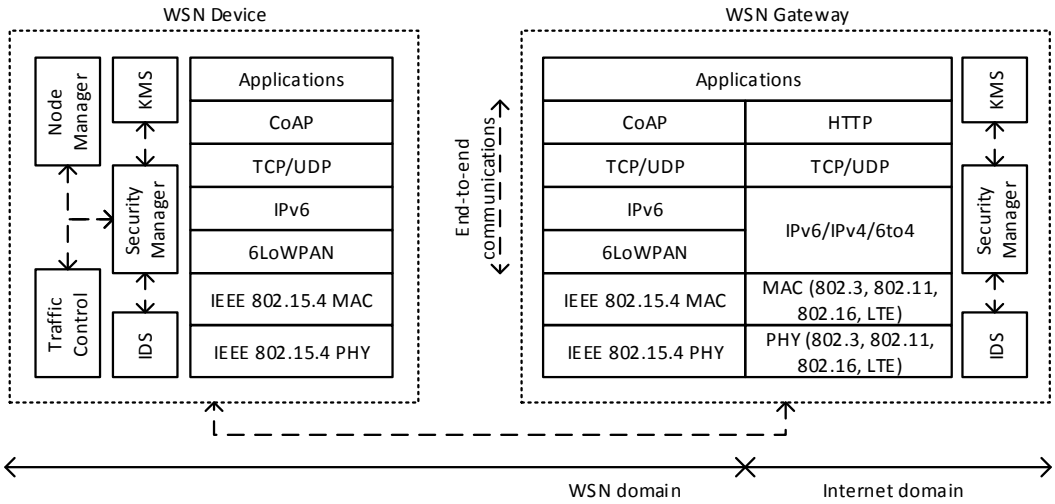

Figure 3.2 – Model cross-layer operation and functional components

We may note that, other than the integration of WSN with the Internet via WSN Internet (6LoWPAN-based) communication technologies, this reference integration model also supports the previously discussed alternative integration approaches, via cloud-based services and front-end proxies. From the perspective of the communication technologies employed, such approaches are based on the usage of a web services interface supported by the WSN gateway, together with specialized middleware components.

As already noted, our initial approach to the design of the model illustrated in Figure 3.2 [182] included the employment of the GinMAC MAC layer. The employment of a TDMA approach to MAC communications also provides opportunities to the design of particular security approaches, as previously considered. As may be observed in Figure 3.2, our reference integration model considers instead the employment of IEEE 802.15.4 for PHY and MAC communications the in WSN domain, in line with the various WSN (6LoWPAN-based) Internet communication technologies supporting the research proposals described in the thesis. The Security Manager (SM), Key Management System (KMS), Intrusion detection (IDS) and Node Manager (NM) components of this model are considered to be cross-layer, and are analyzed in our following discussion.

3.2.2.1 Security Manager

The Security Manager (SM) component is intended to support the enforcement of application security and functional profiles, both in WSN gateways and wireless sensing platforms. As such, this component operates in a layer-independent fashion, providing support for the usage of the procedures necessary for the enforcement of such profiles. Application security and functional profiles enable the description of the requirements of sensing applications in terms of communications and security, as we observe in detail later in the chapter, and in practice may determine the employment of specific key management and intrusion detection approaches.

When employed in a wireless sensing device, the SM is related to the Node Manager (NM) and Traffic Control (TC) components. In our previous approach to the integration model [182], the goal of the TC module was to disconnect misbehaving sensing devices, upon notifications received from the Intrusion Detection System (IDS) module. In particular, with GinMAC the offending device may be disconnected by marking its communications slot as invalid or unavailable, in the context of the TDMA communications schedule. The SM module also supports the authentication and authorization mechanisms in the WSN Device and Gateway, which are related to the AC server as previously discussed in the context of the reference integration model illustrated in Figure 3.1. We also consider that the SM module on the WSN Gateway supports the communications required with the CA server, in the context of the same model.

3.2.2.2 Key Management

The Key Management System (KMS) component supports key management mechanisms designed to support security in the context of end-to-end communications with WSN devices. Various approaches to key management are possible in this context, from simpler approaches involving the pre-deployment of cryptographic keys in sensing devices to more complex solutions, for example providing compatibility with existing Internet key management approaches such as the Internet Key Exchange (IKE) protocol or mechanisms designed to support authentication and key negotiation in the context of particular end-to-end communication protocols, as we explore later.

We have previously discussed the possibility of employing the KMS component on the WSN gateway to support IKE negotiations with Internet hosts [182]. In this situation the KMS could transmit ECC public-keys to wireless sensing devices after negotiation, thus acting as a broker for key negotiation purposes. The gateway can also transparently intercept and mediate end-to-end key negotiations, as we discuss for transport-layer security in Chapter 5. If required by the deployment scenario, keys may also be preprogrammed in sensing devices or transmitted in the bootstrap phase using some form of secure channel.

One fundamental aspect to consider is that keys must be periodically renewed for effective end-to-end security. For example, the AES/CCM algorithm completely loses its security if the same Initialization Vector (IV) is reused with the same key. This implies that the previous approaches to support key management in the context of end-to-end security may also support periodic key renegotiation. Other than the support of different key negotiation strategies, the KMS component may also control the size of the cryptographic keys and its frequency of renewal. Such aspects impact directly on the resources required from constrained sensing devices to support key management, and may be defined according to the application security and functional profile at hand, as managed by the security manager. For example, for key pre-deployment the KMS component on a wireless sensing device may receive the initial key to support the secure bootstrap of the device on the network, and to

afterwards support the renegotiation of a new cryptographic key in the context of the initial authentication of a particular external or Internet client.

3.2.2.3 Intrusion detection

Intrusion detection is a fundamental enabling aspect of the effective integration of WSN with the Internet via WSN Internet communication technologies based on 6LoWPAN, given that end-to-end security at the network and above layers alone cannot provide complete security against internal and external attacks. Intrusion detection for WSN environments is a vast area of research per se, and the IDS component in the reference model illustrated in Figure 3.2 was originally planned to operate also in the context of GinMAC. In the context of TDMA communications, a failing node may also be a node that somehow doesn't follow the temporal and synchronization requirements defined by the TDMA operational model.

As illustrated in Figure 3.2, IDS is based on monitoring components supported by wireless sensing devices and also by the gateway. The main goal of this approach is to employ IDS components in wireless sensing devices that operate as simple probes, scanning network traffic and applying basic filtering operations in order to identify relevant data and events. This information may then be sent to the main IDS component running on the WSN gateway, which supports the most computationally demanding analysis operations. Complex IDS algorithms are implemented exclusively on security gateways and allow the identification and disconnection of misbehaving sensor nodes from the WSN, via the SM component on the gateway and on relevant sensing devices. As previously discussed, in TDMA communication environments the disconnection of particular nodes may also be supported with the help of the TC component.

3.2.2.4 Node Manager

The Node Manager (NM) runs exclusively on WSN devices, and its goal is to provide auxiliary information regarding the operational status of the node to the other components of the model. This operational information may enable the selection of the most appropriate end-to-end security mode in the context or the application, or the dynamic adaptation of security in the light of the available resources. As an example, as long as an application allows it, security may dynamically adapt to employ smaller cryptographic keys or a different symmetric or asymmetric algorithm, in order to save resources.

Other possible application of this component is that the Security Manager on a device can use the information provided by the NM to inform the SM on the WSN gateway about the availability of critical resources on the WSN device. This knowledge may support the clean shutdown of sensing devices reaching the end of its lifetime, or the reconfiguration of particular communication and security mechanisms.

3.3 EMPLOYMENT AND EXPERIMENTAL EVALUATION OF END-TO-END SECURITY

Contrary to the current Internet security architecture, in the context of which mechanisms and protocols are usually designed for devices without serious resource constraints, mechanisms appropriate to Internet-integrated WSN must be carefully designed to cope with the characteristics and limitations of WSN devices and low-energy wireless communications. On the other hand, such mechanisms must be able to support appropriate security requirements, as defined for particular sensing applications. Such aspects dictate that new research solutions must be evaluated with these two aspects in mind, in order to search for acceptable compromises between resource usage and appropriate security.

With the previous aspects in mind, we approach the design of a framework for reconfigurable end-to-end security with Internet-integrated WSN, which accompanies the design, evaluation and employment of new security mechanisms supporting measurable and controllable end-to-end security in the context of Internet-integrated sensing applications. The framework illustrated in Figure 3.3 enables the static configuration, as well as the dynamic reconfiguration of end-to-end security, as required for applications with particular functional and security requirements. We also consider that such requirements may be described by appropriate application security and functional profiles. As previously discussed, the reconfiguration of security may also take place upon particular events from the SM and IDS components, as we consider in our reference model for end-to-end security.

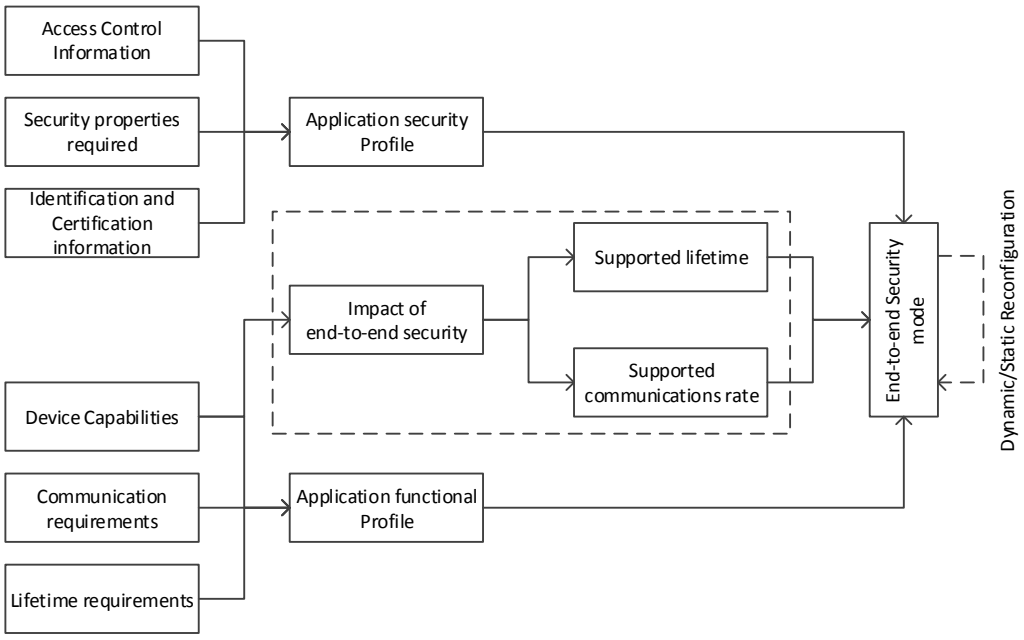

Figure 3.3 - A framework for reconfigurable end-to-end security with Internet-integrated WSN

As Figure 3.3 illustrates, the impact of end-to-end security may be measured considering the functional requirements of sensing applications and the characteristics of the employed

wireless sensing devices. In particular, applications are characterized by particular requirements in terms of how communications are to be supported and their expected lifetime, factors that directly influence the overall cost of end-to-end security. Applications can decide on the security mode to be employed, in a per-device basis, and considering requirements predefined for the application at hand.

As previously discussed, application security profiles may describe the requirements of the application in terms of security and security-related configurations such as the cryptographic suites to employ and the size of cryptographic keys, among others. Application functional profiles identify the type of devices employed and its capabilities, and the communication and lifetime requirements of the application. We also consider that the remaining information required for particular end-to-end security mechanisms must be available, namely access control information and public-keys or certificates identifying the communicating entities.

The research solutions described in the following chapters of the thesis provide different approaches to end-to-end security in the context of Internet-integrated WSN. Such mechanisms also support complementary approaches to security, which may enable applications to statically or dynamically configure end-to-end security for particular devices, as the framework in Figure 3.3 illustrates. Very-constrained wireless sensing devices may employ mechanisms with delegation of security operations to more powerful devices, while more capable sensing devices may support more functionalities or even full end-to-end security.

As previously discussed, the measurement of the impact of end-to-end security is an important component of the framework illustrated in Figure 3.3. In Figure 3.4 we illustrate how the effectiveness and efficiency of the research proposals is evaluated experimentally in subsequent chapter of the thesis. In this methodology we consider the impact of the proposed mechanisms both on the lifetime of sensing applications and on the maximum achievable communications rate, two fundamental requirements for the effectiveness of sensing applications employing Internet-integrated sensing devices. Memory is also an important aspect to be considered, given the limited RAM and ROM memory available on wireless sensing platforms.

As Figure 3.4 illustrates, the impact of security on the (limited) energy available on WSN sensing devices influences the achievable lifetime of the application. This is a particularly important aspect to consider, given that security-related operations may be particularly expensive in current sensing platforms. Energy is required to support authentication and key agreement in the context of the initial end-to-end authentication phase, and also to process security for normal communications afterwards. We must also consider the impact of the processing and transmission of information required for the support of security, for example new security headers or authentication and integrity codes. Considering the communications rate of applications, we must also consider the computational time required to support

authentication and key agreement, and also the delay introduced on communications by the processing of security and the transmission of security-related data, as in the case of energy.

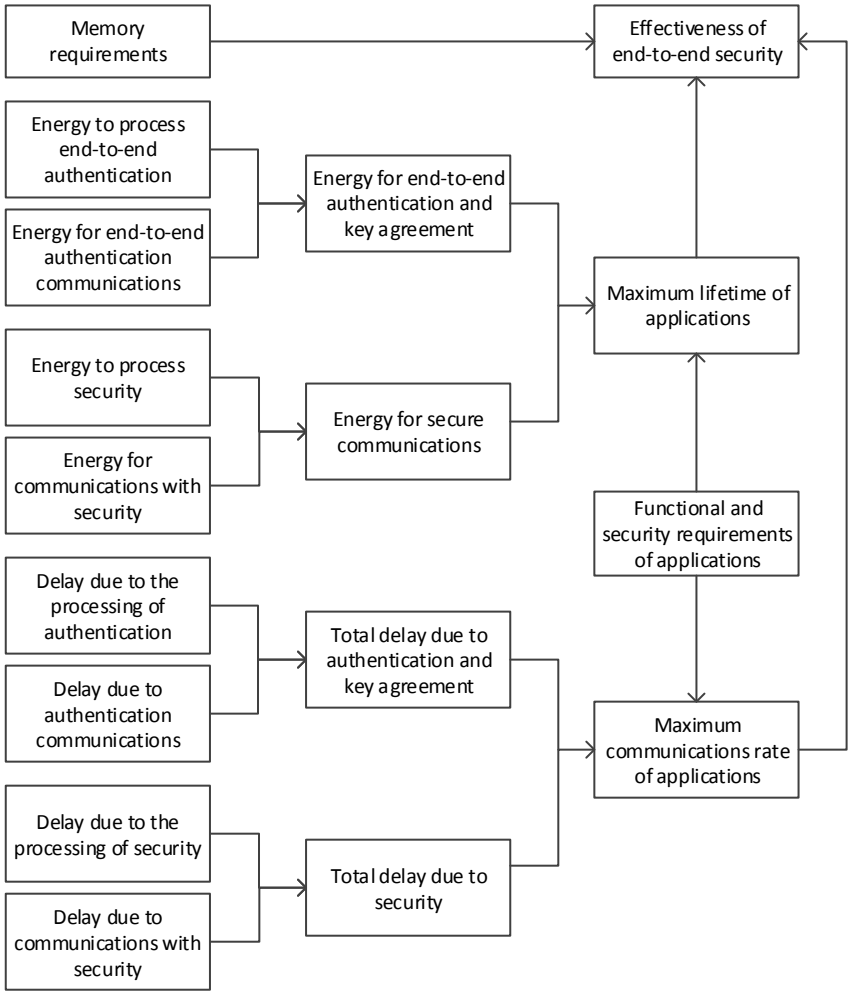

Figure 3.4 – Methodology for the experimental evaluation of end-to-end security

In conclusion, the previously described approaches target the identification and employment of end-to-end security solutions that are controllable from the point of view of its requirements of resources on constrained WSN devices, while guaranteeing appropriate security for sensing applications employing Internet-integrated WSN. In this context, in the following chapters of the thesis we propose and evaluate research solutions to support security with 6LoWPAN-based end-to-end communications at the network, transport and application layers.

3.4 SUMMARY

In the present chapter we begin by discussing our preliminary research efforts in the context of the GINSENG research project. As discussed, despite the absence of particular research proposals or mechanisms, this work provided the ground for the design of the reference

model for end-to-end security, which we discuss later in the chapter. As we have previously discussed, our research efforts later evolved to focus on security for end-to-end communications with Internet-integrated WSN and M2M environments. We have also noted that our model has considered from the start the usage 6LoWPAN-based technologies to enable such end-to-end communications. The same applies to the consideration of metrics and profiles for the evaluation of the impact of security.

We have also presented the methodology considered in the following chapters for the experimental evaluation of the various research proposals. Among other aspects, we consider the impact of the proposed mechanisms on the energy and on the computational time required from constrained sensing platforms, two aspects that directly influence the lifetime of sensing applications and the communications rate that such applications may effectively sustain over time. Other aspect we discuss is that applications may dynamically or statically reconfigure or select the most appropriate end-to-end security mode from among a set of available mechanisms. This approach may be useful in the context of a security architecture supporting Internet-integrated WSN and end-to-end security mechanisms side-by-side with other required functionalities. Other than functional or security requirements, applications may also adapt according to external conditions or particular deployment characteristics.

4 END-TO-END SECURITY FOR 6LOWPAN[3]

The present chapter describes our research proposal to address security in the context of end-to-end communications at the network layer involving Internet-integrated WSNs. As we have previously discussed, communications with such characteristics may be enabled by the 6LoWPAN adaptation layer on the WSN domain, and are in reality transparent to communication mechanisms at higher layers of the stack. Our research proposal is also motivated by the fact that end-to-end security may provide a transparent solution to address security for Internet communications involving WSN sensing devices and external or Internet hosts.

We start by addressing the benefits and goals of network-layer security in the context of 6LoWPAN, and next we describe the proposed compressed security headers for the 6LoWPAN adaptation layer. The effectiveness of the usage of the proposed security mechanisms is also experimentally evaluated, considering the methodology discussed in the previous chapter and the reference model for the employment of end-to-end security.

4.1 INTRODUCTION

The design of standard communication and security mechanisms for resource-constrained sensing applications and devices may provide an important contribution for its integration with the Internet and consequently towards the realization of what we nowadays identify as the Internet of Things (IoT). This vision will only be realizable if appropriate security mechanisms are available, and in this context we target the design and experimental evaluation of security mechanisms for end-to-end communications at the network-layer with sensing devices (smart objects) using the standard IPv6 protocol.

Our work proposes the employment of new compressed security headers for the 6LoWPAN adaptation layer, which we also experimentally evaluate using the TinyOS operating system and the BLIP (Berkeley Low-power IP) networking stack. As we discuss later, various employment scenarios are identified as viable for end-to-end network-layer security in WSN environments using the proposed extensions to the 6LoWPAN adaptation layer, particularly

[3] *This chapter has supported the following publications:*

- Granjal J, Monteiro E, Silva J. *Network-layer security for the Internet of Things using TinyOS and BLIP*, International Journal of Communication Systems, 2013
- Granjal J, Monteiro E, Silva J. *Enabling network-layer security on IPv6 wireless sensor networks*, IEEE Globecom 2010
- Granjal J, Monteiro E, Silva J. *Why is IPSec a viable option for wireless sensor networks*, 5th IEEE International on Mobile Ad Hoc and Sensor Systems (MASS) 2008

if security mechanisms are designed to benefit from cross-layer interactions enabling the optimization of expensive cryptographic operations.

Our initial proposal of the introduction of security in the 6LoWPAN adaptation layer was published in [183], and next with our reference model for the interconnection of WSN with the Internet in [182][184], in the context of which the proposed mechanisms are also theoretically validated. A more complete experimental evaluation of the research proposals discussed in this chapter is also available in [185].

The research work described in this chapter (and throughout the thesis for that matter) is strongly motivated by the realization that strong assurances will be required in terms of security for many applications on the Internet of Things (IoT), which are expected to process and communicate sensitive data using wireless communications [159][3], and that in this context WSN environments which can be integrated with the existing Internet communications and security architecture will play an important part. Security mechanisms should thus be designed and adopted for the IoT that are flexible to the requirements of applications, while providing acceptable security guarantees. In the context of the experimentation of new research solutions, standardization represents an effective ground for the design of compatible security solutions [157], also because the materialization of the IoT will strongly depend on the design and acceptance of appropriate communication and security technologies.

While we may accept that not all smart objects on the IoT will have the capability or be required to support IPv6, the availability of secure end-to-end communications at the network layer with other sensing devices or with Internet hosts may enable a much richer integration of sensing applications with the Internet. It may also enable new types of sensing applications where smart objects are able to cooperate transparently, remotely and securely using Internet communications.

As previously discussed, despite the current design of 6LoWPAN for IEEE 802.15.4 at the MAC and PHY, other technologies are expected to be supported in the future, enabling a myriad of heterogeneous sensing and actuating devices to communicate using standard IP protocols [186][187]. Other than the adaptation layer, the 6LoWPAN group has also defined mechanisms such as neighbor discovery and address auto-configuration that allow a sensing device to activate its presence on an existing IPv6 network of smart objects. As previously analyzed in Chapter 2, header compression is omnipresent in all 6LoWPAN solutions, given the extremely limited payload space to transmit data using LoWPAN technologies such as IEEE 802.15.4.

Although the successful integration of 6LoWPAN networks with the Internet will require security to be properly addressed from the start [188], we note that it has not been properly addressed in 6LoWPAN, as only generic considerations and recommendations [108] have been produced. At the time of publication of our research proposal, no solution to enable secure end-to-end communications with IPv6-enabled smart objects using the adaptation

layer existed, also due to the assumption that security should be addressed at higher layers of the stack. On the other hand, 6LoWPAN enables many useful practical usage scenarios, for example two smart objects on remote locations that are able to communicate over the Internet in the context of a distributed sensing application, or Internet hosts that are able to obtain information directly from sensing devices by communicating directly with such devices.

As network-layer security provides an important contribution to security in the current Internet security architecture, we may also expect that it may also play an important part in a future security architecture encompassing Internet-integrated WSN environments and communications. This aspect motivates our research efforts towards the design of security mechanisms compatible with the internal workings of the 6LoWPAN adaptation layer [69], which we present and evaluate in the present chapter, and also other proposals in subsequent chapters. Although we have previously discussed security at the network-layer using 6LoWPAN communications in our SoA study in Chapter 3, our following analysis complements our discussion, by identifying other proposals focused on security for end-to-end communications with constrained sensing platforms.

4.2 PREVIOUS APPROACHES TO END-TO-END SECURITY

The following discussion complements the analysis previously performed in Chapter 2, where security in WSN environments was discussed, both in regards to isolated WSN environments and also to Internet-integrated WSN environments. We are able to verify that previous proposals on the implementation of secure end-to-end communications between smart objects and Internet hosts mostly target the transport layer, in particular by proposing modified versions of the SSL (Secure Sockets Layer) protocol. One of such research proposals is SSNAIL [189], which proposes a light-weighted version of SSL to be supported by Internet hosts and smart objects. Other proposals do exist providing only partial end-to-end security, for example Sizzle [190] which employs SSL to secure communications between an Internet host and a security gateway protecting the network of smart objects from the Internet, with such communications being translated to a proprietary communications protocol in the network of smart objects. The support of different security modes that can be related to the security requirements of particular sensing applications and to the characteristics of the sensing devices employed by such applications was proposed in ContikiSec [191]. The characteristics of such research proposals are described in Table 4.1.

The proposals previously analyzed and summarized in Table 4.1 have shown that security can be effectively employed at higher communication layers with resource constrained smart objects, something that is in deep contrast with the classic perception of many researchers. Nevertheless, two important aspects are missing from these proposals that we believe are vital for security in the context of the IoT, and can be (at least partially) answered by network-layer security, as we proceed to discuss.

One important factor is that security mechanisms should be available that provide security for communications independently of the applications. In this respect, SSL presents the limitation of requiring explicit support from sensing applications. Other relevant aspect is that security mechanisms should be adaptable to the characteristics and security requirements of particular sensing applications. Regarding this aspect, mechanisms that work with fixed configurations in terms of parameters that control its security and resource usage may be limitative for the IoT. Aspects such as the cryptographic algorithms employed and relevant configuration parameters such as cryptographic key size and frequency of key refreshment deeply influence the lifetime of sensing applications and resource-constrained devices. Security mechanisms should therefore allow the establishment of acceptable compromises between resources required for performing security operations and the security level required for a particular sensing application. As we have discussed in Chapter 3 and consider when evaluating the security mechanisms proposed in the thesis, such aspects may be defined by appropriate security and functional application profiles.

Table 4.1 – Previous research proposals addressing end-to-end security at higher layers

Security Properties/Functionalities	SSNAIL	Sizzle	ContikiSec
Authentication	ECC (ECDSA)	ECC (ECDSA)	CMAC
Key negotiation	ECC (ECDH)	ECC (ECDH)	Not supported
Key size(s)	160 bits	160 bits	128 bits
Data encryption	RC4	RC4	AES
Hashing/Integrity	MD5, SHA1	MD5, SHA1	CMAC
Access control	Not supported	Security Gateway	Not supported
Operational layer	Transport (SSL)	Transport (SSL)	Link-layer
Gateway usage	No	Yes	No
End-to-end security	Yes, with SSL	Yes, with SSL	Not supported

It is our belief that security can be integrated at the 6LoWPAN adaptation layer with the characteristics previously identified as desirable, and thus enabling the usage of application-independent and flexible security mechanisms, which can play an important part in the integration of smart objects with the Internet. Security at the network-layer is certainly not a solution to all security requirements, and we must also expect that extremely restricted devices such as Radio-Frequency Identification (RFID) devices may require different approaches to security, as discussed in [192][193]. Such aspects can certainly be dealt with by adopting new mechanisms in the context of an appropriate security architecture for the IoT.

Taking into consideration the limitations previously discussed, we address the design of security at the network-layer for Internet-integrated WSN, by introducing security in the

6LoWPAN adaptation layer, as discussed next. As previously discussed, end-to-end security at the network-layer may provide a transparent solution to address security for communications involving 6LoWPAN WSN sensing devices and other Internet entities.

4.3 A PROPOSAL FOR SECURITY IN THE 6LOWPAN ADAPTATION LAYER

In our following discussion we begin by introducing security in the context of the header compression mechanisms of 6LoWPAN. As previously discussed, header compression is a fundamental enabling factor of functionalities designed in the context of the adaptation layer, and one that must be considered when introducing new security mechanisms.

4.3.1 SECURITY IN THE CONTEXT OF HEADER COMPRESSION

One major goal of the 6LoWPAN adaptation layer is the support of fragmentation and reassembly of IPv6 packets transmitted over LoWPAN environments. This is a necessity because IPv6 determines that any communications link may be able to support a minimum MTU of 1280 bytes, while the MTU of LoWPAN is typically lower. For example, with IEEE 802.15.4 only 102 bytes are available (without link-layer security) of payload space, as Figure 4.1 illustrates. The payload space available when using IEEE 802.15.4 depends on the overhead introduced by addressing and control information at the link layer. Security may also be enabled at the link-layer with IEEE 802.15.4 [74], which at the end also influences the final payload space available for upper layer protocols and applications.

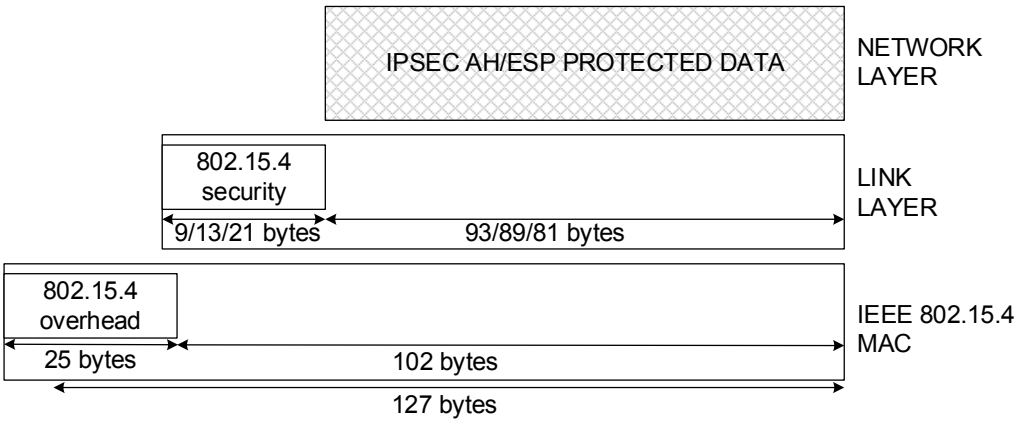

Figure 4.1 - Payload space availability at the 6LoWPAN adaptation layer

As illustrated in Figure 4.1 and previous analyzed in our state of the art study, IEEE 802.15.4 provides three link-layer security modes. Regarding its impact on the available payload space, the AES-CCM-128 security mode requires 21 bytes of payload space, AES-CCM-64 requires 13 bytes and AES-CCM-32 requires 9 bytes. In this analysis we are considering the usage of an IEEE 802.15.4 auxiliary security header occupying 5 bytes, with 1 byte being used for the security control field and 4 bytes for the frame counter field, also considering that

the cryptographic keys required for security are obtained automatically from the source and destination link-layer addresses of the frame [74]. As network-layer security can protect communications even for data transmitted between WSN devices, for the purpose of the evaluation of our proposal described later in the chapter we consider the availability of 102 bytes as the data payload for 6LoWPAN, meaning that we dispense link-layer security.

Other relevant aspect is that link-layer security may be available at the hardware in sensing devices implementing IEEE 802.15.4, as previously observed. This implies that the fact that link-layer security mechanisms are not activated doesn't mean that such efficient encryption and authentication mechanisms can't be of use. In this context, we consider the design of cross-layer security mechanisms for the 6LoWPAN adaptation layer, which allow us to benefit from the availability of such efficient cryptographic operations, as we discuss later. Figure 4.1 also illustrates the reason why header compression is so prevalent in 6LoWPAN, as even when not using link-layer security applications or upper layers communication protocols do not have that much space left to transmit data.

4.3.2 COMPRESSED SECURITY HEADERS FOR 6LOWPAN

As IEEE 802.15.4 doesn't provide any type of multiplexing information to allow a receiver to distinguish among different types of data packets, 6LoWPAN uses the first byte of the link-layer payload as a dispatch byte, which allows the identification of the transported packet and (if necessary) further information within the subtype. We need therefore to decide how new headers for security are going to be identified in 6LoWPAN using the dispatch byte. Three strategies would allow us to identify the presence of new headers in the context of the 6LoWPAN adaptation layer, as we proceed to discuss.

The first option for the identification of security is to use the ESC header type value [70], which allows the usage of an additional dispatch byte to identify the presence of new headers. Using this approach the first (original) dispatch byte remains untouched and the following (new) dispatch byte can be used to identify new security headers. This approach presents the inconvenience of requiring one additional byte for this purpose.

A second option for the identification of new security headers is to use context-based header compression as in [151], particularly using the LOWPAN_IPHC and LOWPAN_NHC headers, and to define appropriate identification values for security using the EID field of the LOWPAN_NHC header. This approach is now viable since context-based header compression has been recently adopted as a standard [71].

The third identification option is to integrate security in the context of standardized headers and identification values, by defining new dispatch type values for security using reserved values of the original payload byte. This is our approach and corresponds to a strategy identified from the start in RFC 4944 [70]. The employment of reserved dispatch values for this purpose is both accepted and encouraged in this document, which defends that with the

further development of 6LoWPAN additional functions are expected to occupy unused space.

We must note that context-based header compression was not available at the time of our proposal of new 6LoWPAN security headers [182], [184], [185], and as such was not considered for this purpose. Nevertheless, for the sake of fairness we must note that the security headers described later in this chapter are in reality independent of the approach enabling its identification at the 6LoWPAN adaptation layer, and thus this approach does not determine nor influence the effectiveness of the research solutions described in this chapter. Another (and more recent) proposal exists for security at the 6LoWPAN adaptation layer [151], using LOWPAN_IPHC compression and implemented in Contiki. In this proposal the authors don't consider the usage of security in tunnel and transport modes, and also do not consider the usage of variable-sized keys and authentication data. We find such aspects to be important for the adaptability of security to applications with different requirements in terms of security, as we consider in our experimental evaluation described later in this chapter. The same applies to a study on the impact of the proposed mechanisms on the lifetime of sensing applications, as well as the measurement of real energy consumption instead of via energy estimation as in [151]. We proceed by describing how new identification values for security are defined at the 6LoWPAN adaptation layer.

4.3.2.1 New 6LoWPAN dispatch type values for security

The 6LoWPAN adaptation layer uses the first two bits of the dispatch byte (the first byte of the IEEE 802.15.4 payload) to allow nodes to identify the presence of a 6LoWPAN packet or of other types of packets. For a 6LoWPAN packet, the remaining bits of the dispatch byte allow the identification of specific types of 6LoWPAN headers that correspond to given functionalities of the adaptation layer, namely a mesh, fragmentation or addressing header. When the first two bits identify a 6LoWPAN addressing header (value '01', please refer to Table 4.2), several dispatch values are reserved as RFC 4944 [70] describes. We use four values from the set of reserved values to identify the presence of new 6LoWPAN compressed security headers and respective usage modes, as Table 4.2 describes.

Table 4.2 – New dispatch values to identify 6LoWPAN security usage modes

Header dispatch values for 6LoWPAN security	6LoWPAN security header and usage mode
01 001xxx	AH in transport mode
01 101xxx	AH in tunnel mode
01 011xxx	ESP in transport mode
01 100xxx	ESP in tunnel mode

The values in Table 4.2 are more precisely obtained from the set of reserved values after LOWPAN_HC1, which is the value defined to identify the presence of a HC1 compressed addressing header. HC1 is the header compression format adopted in 6LoWPAN to compress addressing information, while HC2 was defined to allow the compression of transport-layer UDP header information. As we can see in Table 4.2, the first 3 of the remaining 6 bits of the dispatch byte are sufficient to identify a security header, together with its usage mode and irrespective of the value of the remaining 3 bits. The 3 remaining bits are sufficient to distinguish between different types of 6LoWPAN addressing headers. This identification strategy therefore gives us the possibility of simultaneously identifying the presence of security and addressing information on a given 6LoWPAN packet, allowing also to save payload space and easing the processing of headers in tunnel and transport modes.

4.3.2.2 Compressed ESP header for 6LoWPAN

For the design of new security headers for the 6LoWPAN adaptation layer we find it fundamental to take into consideration various aspects. The first is that the principles of simplification, compression and shared context around which other 6LoWPAN headers [70] were designed should also be considered for security. At the same time, it is desirable that the processing of such headers can be easily integrated into existing implementations of the IP Security architecture [99], as this would contribute to its evolution towards easily adopting new IPv6-enabled sensing applications. Another important aspect is that most sensing platforms currently possess or will probably adopt in the future hardware cryptographic operations.

Hardware encryption and authentication must therefore be considered together with cryptographic algorithms implemented in software. For example, IEEE 802.15.4 requires hardware cryptography and platforms such as the TelosB [194] mote support hardware security with the AES cryptographic algorithm in CCM* combined mode, using the cc2420 chip. AES/CCM provides encryption and decryption in the CTR (Counter) mode and authentication and integrity in the CBC-MAC mode. The CCM* variant of AES/CCM additionally offers encryption-only and integrity-only capabilities, a characteristic that makes it well adapted to the independent support of authentication and encryption headers. Considering that AES/CCM is part of the set of future mandatory algorithms for the IP Security architecture, we realize the importance of its consideration during the design of security headers for 6LoWPAN.

As the design of new security headers for 6LoWPAN will require header compression, Internet hosts running IPSec [99] may support 6LoWPAN security headers in the future, and adapt to the usage of the compressed security fields in communications with constrained sensing devices. Another strategy is for Internet hosts to establish IPSec associations with WSN security gateways, with such gateways translating between IPSec and 6LoWPAN IPSec for secure communications with the final sensing devices.

In Figure 4.2 we illustrate how the 6LoWPAN ESP (Encapsulating Security Payload) [24] security header is formed, and in the same Figure we also illustrate which fields are integrity protected (with an 'I') and encrypted (or confidentiality) protected (with a 'C'). The purpose of this header is to provide applications with encryption and optional authentication and integrity of 6LoWPAN packets in the context of end-to-end communications.

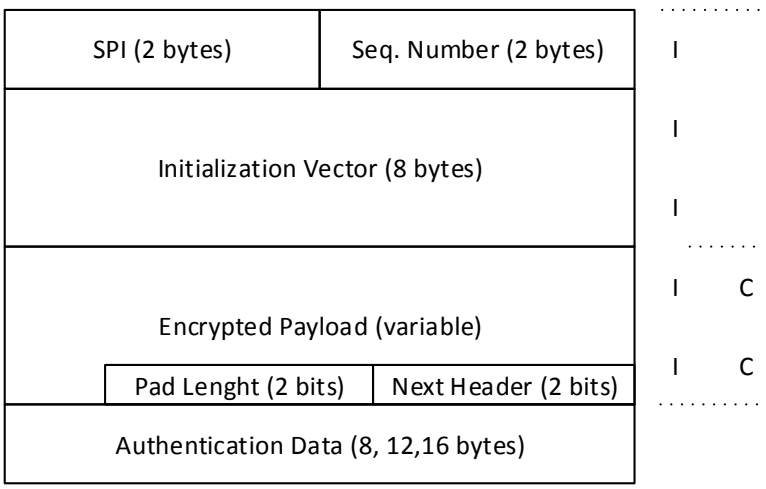

Figure 4.2 - Compressed ESP security header for 6LoWPAN

By analyzing the 6LoWPAN ESP header illustrated in Figure 4.2, we begin by identifying a 2-byte SPI (Security Parameters Index) field, whose purpose is to allow a receiving entity to relate an incoming packet to a specific security association. This allows a sensing device to obtain information such as the cryptographic algorithms and keys required to apply security operations to the packet. Given the constraints of sensing devices, a 2-byte SPI is considered appropriate. Security associations thus enable the maintenance and identification, at the both ends of end-to-end secure communications, of the relevant data to process security for the network-layer 6LoWPAN packets received and transmitted.

The next field in the 6LoWPAN ESP header stores a 2-byte sequence number, with the purpose of helping end systems in protecting against packet replay attacks. The sequence number is treated as an unsigned value and implementations must ensure that a distinct value is maintained for each different security association. Given the transmission rates of typical sensing applications, 2 bytes are considered appropriate for this field. Nevertheless, if necessary an option similar to ESN (Extended Sequence Numbers) [100] may be designed for 6LoWPAN in the future, allowing communicating parties to agree on larger sequence numbers. A 6LoWPAN ESN option would allow a device to maintain a larger sequence number for security associations requiring it, with such number being used for ICV-computation purposes, while only its lower 2 bytes being transmitted with each 6LoWPAN packet.

It is important to note in this context that other than the usage of a distinct sequence number for each security association, a key management mechanism appropriate to 6LoWPAN must be designed to allow keys to be periodically refreshed. This is due to the fact that algorithms as AES/CCM completely lose its security if a given key is reused with the same sequence number. The development of appropriate key management mechanisms for 6LoWPAN appears therefore as an important research goal. For this purpose, IKE can be simplified or in alternative completely different approaches to key management can be followed [107][105].

Next in the 6LoWPAN ESP header we encounter the IV (Initialization Vector) field, which is used to transport cryptographic synchronization data necessary for two devices to successfully apply the same cryptographic algorithm. An 8-byte IV enables the usage of all the current and future mandatory cryptographic algorithms defined for the IP Security architecture. It also reflects recommendations from RFC 4309 [195], in that it allows compatibility with current and future cryptographic suites based on AES. Synchronization data may be used as input to 3DES and AES in CBC (cypher-block chaining mode) algorithms, or together with additional data generated by end devices to produce the input required for algorithms such as AES in CTR (counter) encryption mode. The CTR mode is supported by hardware implementations of AES/CCM, and the rules currently defined for the usage of AES in CTR mode with the ESP header [195] state that 3 bytes of salt must be added to the IV data for this purpose, since AES requires an 11-byte nonce. Again, by following such rules we promote an easier integration of our new 6LoWPAN security headers in current implementations of the IP Security architecture, one of the goals of this proposal.

Next in the packet comes the encrypted data, at the end of which two fields are added that help in employing the security header with different encryption algorithms and usage modes. The first is the pad length field, which stores the number of padding bytes (from 1 to a maximum of 4) added to the original encrypted data to align up the payload and trailer, if required by the encryption algorithm employed. Next appears the next header field, which stores information on how the receiver should interpret the decrypted data by indicating the presence of a TCP, UDP or ICPMv6 packet.

At the end of the 6LoWPAN ESP header follows the ICV (Integrity Check Value) or MIC (Message Integrity Code) field, which stores the authentication data used to authenticate the origin of the 6LoWPAN packet and verify its integrity. As integrity in optional when using 6LoWPAN ESP, in practice they are only performed if required in the context of a given security association. The size of the authentication data depends on the encryption algorithm used to generate the MIC code and on the level of integrity and authentication required for the given security association. This field is of 12 bytes if generated using HMAC-SHA1-96 or AES-XCBC-MAC-96, since both algorithms produce a 96-bit MIC code. When using hardware AES/CCM, this algorithm can be used to generate 8, 12 or 16 bytes MIC codes, also in line with recommendations from RFC 4309 [195]. The MIC code is not protected by encryption, meaning that smart objects are able to verify the authenticity and

integrity of a received 6LoWPAN packet protected with ESP before being required to perform more computational demanding decryption operations.

In conclusion, we may also observe that the layout of the 6LoWPAN ESP header reflects the design principles adopted by 6LoWPAN and our goals of facilitating its future integration in the IP Security architecture and benefiting from hardware cryptography. Simplification and compression are performed whenever possible and applicable, while at the same time the relevant fields are appropriately dimensioned for software and hardware cryptographic algorithms. As we analyze later in the chapter, the cryptographic algorithms considered for 6LoWPAN security include the algorithms currently adopted as mandatory in the context of the IP Security architecture.

4.3.2.3 Compressed AH header for 6LoWPAN

The purpose of the 6LoWPAN AH (Authentication Header) is to allow end systems that do not require confidentiality to verify the integrity and origin of 6LoWPAN network-layer packets in the context of end-to-end communication sessions. 6LoWPAN AH also provides protection against replay attacks. The usage of the authentication header is of interest as security mechanisms to provide such properties are usually less demanding of the resources available on constrained smart objects, when compared against those supporting encryption and decryption (as required for 6LoWPAN ESP). We also observe that many sensing applications on the IoT will probably not require encryption, as the data transported is itself not confidential, while the most important may be to protect communications against corrupted or manipulated packets, and to authenticate its origin.

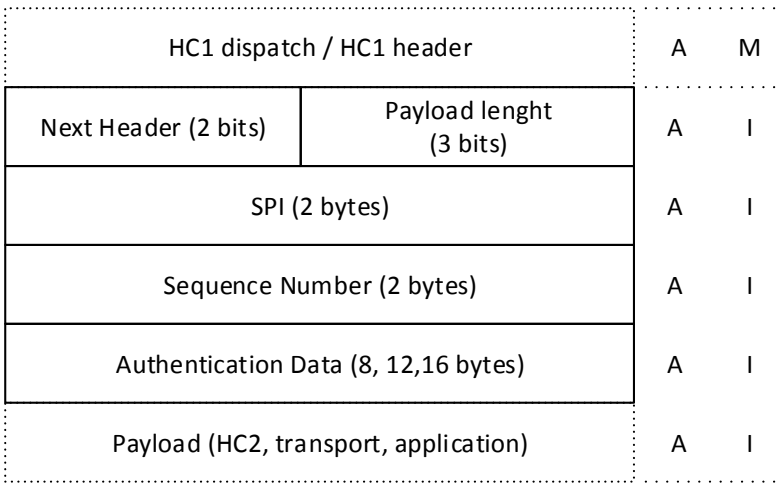

Figure 4.3 - Compressed AH security header for 6LoWPAN

In Figure 4.3 we illustrate the 6LoWPAN compressed AH header [101], and in the same figure we also indicate which parts of the header and data payload are integrity and authentication protected (as indicated by an 'A') and are considered mutable (as indicated

by an ‘M’) or immutable fields (as indicated by an ‘I’) in respect to the computation of the ICV. Mutable fields are fields for which the sending device (which must compute the ICV) is unable to calculate or predict its final value upon arrival of the packet at its destination, and thus such values are zeroed for the purpose of computing the IVC.

An added advantage of the 6LoWPAN authentication header over its ESP counterpart previously analyzed is that security (authentication and integrity) can also be applied to fields outside the security header itself. The authentication data is computed considering all the fields identified as immutable in Figure 4.3 (with the ICV field itself being zeroed for that purpose), but implementations may also decide to include immutable fields of the HC1 or HC2 addressing and transport headers. The padding required by the integrity algorithm (if any) and the high-order bits of the ESN option (if adopted for 6LoWPAN in the future, as previously discussed) are also considered during the computation of the ICV.

Analyzing the 6LoWPAN AH header illustrated in Figure 4.3, the Next Header field allows the identification of the next header as being TCP, UDP or ICMPv6. Similarly to the original authentication header [101], the payload length field stores the total length of the header in units of 32-bit words. As Figure 4.3 illustrates, 3 bits are considered to be sufficient to measure the space necessary to store the authentication data (maximum 16 bytes), sequence number, SPI, next header and payload length fields. To guarantee byte-alignment, implementations should consider that the next header and the payload length fields occupy one byte and zero out the remaining bits. Byte-alignment of the authentication header promotes higher efficiency in header processing by 6LoWPAN implementations, and this rule was followed in our TinyOS implementation evaluated later in the present chapter. The SPI field allows a device to map the 6LoWPAN packet to a particular security association, as in the 6LoWPAN ESP header, and the sequence number supports protection against packet replay attacks.

The size of the authentication field is proportional to the integrity and authentication level required for the security association and is in line with the set of cryptographic algorithms that can be used for its generation, as was previously discussed for the 6LoWPAN ESP header and utilized in our experimental evaluation study.

As a final remark concerning the 6LoWPAN security headers described, one aspect to note is that they don’t allow the maintenance of nice 32-bit or 64-bit boundaries that were a concern during the design of its counterparts [195][101] for the IP Security architecture. This is not so much of a problem for 6LoWPAN, considering that it is designed for sensing platforms typically employing low-end 8-bit or 16-bit microcontrollers.

4.3.2.4 Integration of security in the context of existing 6LoWPAN headers

As other 6LoWPAN headers are currently defined, we need to consider how the new 6LoWPAN security headers may be employed side-by-side with such headers. We need therefore to analyze the usage of security together with the mesh addressing header, the

fragmentation header and the compressed addressing header. This is important not only in respect to the implementation of a security-enabled 6LoWPAN networking stack, but also because it allows us to investigate the impact of security on the final payload space available to 6LoWPAN applications, as we consider in our evaluation study later in the chapter.

The mesh addressing header transports information for layer-two forwarding whenever a mesh routing protocol is employed for routing packets from node to node in the LoWPAN [196]. It is important to note that mesh routing is independent of 6LoWPAN, as IPv6 only cares about the source and destination addresses of the devices, independently of how the packet arrives at its destination. The fragmentation header transports information related to how the original IPv6 packet was fragmented for its transportation in the LoWPAN, and which therefore is necessary for the reassembly of the original packet at the destination node. Finally, the compressed addressing header allows the compression of IPv6 addresses and multicast addresses whenever possible.

Considering that 6LoWPAN security is inherently end-to-end, meaning that it is intended to be generated and interpreted by 6LoWPAN devices, the headers that are destined to be interpreted by each device on the path of the 6LoWPAN packet towards its final destination must not be considered for security purposes. This applies to the mesh addressing header, which therefore must appear before any 6LoWPAN security header, independently of its usage mode. The same rationale can be applied to a broadcast (identified by the LOWPAN_BC0 type) and fragmentation headers. A broadcast packet stores a sequence number intended to be interpreted at each forwarding node, allowing the implementation of the broadcast mechanism using a flooding communications algorithm. The fragmentation header transports information necessary for the reassembly of the IPv6 packet at the 6LoWPAN destination. In summary, we consider that 6LoWPAN security headers protect only end-to-end payloads, as makes sense for network-layer security, and as such appear after the mesh, broadcast and fragmentation headers.

As with the traditional IP Security architecture [99], we consider that 6LoWPAN security may be useful in two usage modes, the tunnel mode and the transport mode. Transport mode enables secure communications between two end devices (smart object or other type of 6LoWPAN or IPv6 device) and will be preferred in many usage scenarios, also considering that it requires less header space from the (already limited) link-layer payload. On the other hand, tunnel mode allows for the tunneling of secure communications via intermediate devices functioning as security gateways or as 6LoWPAN routers. The usage of 6LoWPAN security in these two usage modes is discussed in greater detail next in the chapter.

4.3.2.5 Tunnel and transport mode usage scenarios

In Figure 4.4 we illustrate the usage of 6LoWPAN security in transport mode, side-by-side with other compressed 6LoWPAN headers and data from transport protocols and applications. As previously discussed, the mesh, broadcast and fragmentation headers (if

present on a given 6LoWPAN message) appear before the security headers. Security is identified side-by-side with compressed addressing, and in transport mode acts on the payload of the original packet, which may contain an HC2 compressed UDP header and data from other transport protocols and applications. The scope of this 6LoWPAN security header in transport mode depends on it being AH or ESP, as previously discussed. When using authentication and integrity, a MIC or ICV is transmitted at the end.

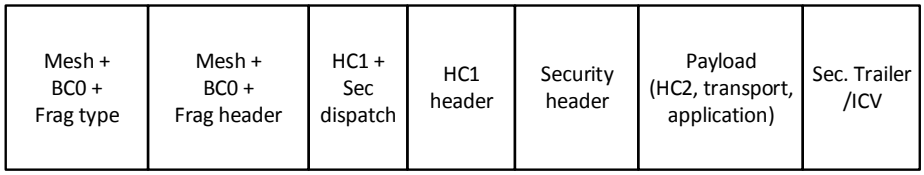

Figure 4.4 - Usage of 6LoWPAN security in transport mode

Regarding the usage of 6LoWPAN security in tunnel mode, two addressing headers are necessary and security is employed as illustrates in Figure 4.5. The inner addressing header identifies the address of the ultimate destination of the 6LoWPAN packet, which may be for example a 6LoWPAN smart object, while the outer addressing header identifies the immediate (intermediate) destination of the packet, for example a security gateway placed between the Internet and the network of smart objects supporting a given sensing application, or a 6LoWPAN router supporting secure communications between remotely deployed sensing devices.

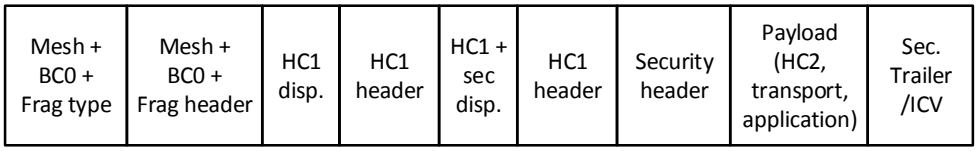

Figure 4.5 - Usage of 6LoWPAN security in tunnel mode

The usage of 6LoWPAN security in tunnel mode allows the protection of the entire inner (ultimate) addressing header and also of the original data payload. Again, which fields are considered for security depends on the usage of the 6LoWPAN AH or ESP compressed headers, and may also depend on particular implementations of 6LoWPAN security, as networking stacks may decide to include information from compressed transport headers such as HC2, for example. As with security in transport mode, authentication data follows at the end if necessary. Our discussion proceeds with an experimental evaluation of security at the 6LoWPAN adaptation layer, employing the previously described compressed security headers.

4.4 EXPERIMENTAL EVALUATION SETUP

The validation of any proposal on security for resource-constrained sensing devices is of particular relevance if performed experimentally, as in practice several unpredicted aspects related to the functioning of sensing devices and wireless communications are difficult to reproduce realistically using simulation environments. As previously discussed, such aspects also motivate the experimental evaluation of the research proposals described throughout the thesis. In our following discussion, we consider the employment of end-to-end security in the context of the reference integration architecture described in Chapter 3. The experimental evaluation and employment of 6LoWPAN security in the various usage modes also considers the framework and evaluation methodology previously discussed.

4.4.1 EXPERIMENTAL EVALUATION SCENARIO

Our proposal on security for the 6LoWPAN adaptation layer was implemented using the TinyOS operating system [186], in particular in the context of its Berkeley Low-power IP (BLIP) [197] networking stack. In the experimental evaluation we consider the employment of UDP communication sessions established between different 6LoWPAN devices, in particular between a TelosB [194] mote and a Linux host supporting both 6LoWPAN and IPv6. As previously discussed, UDP is the currently supported transport-layer protocol for Internet communications with 6LoWPAN-enabled sensing devices. In Figure 4.6 we illustrate the communications model considered for our experimental evaluation, which in practice represents a concretization of the reference model for end-to-end security illustrated in Figure 3.2 for the support of end-to-end security at the network-layer using 6LoWPAN over IEEE 802.15.4 environments.

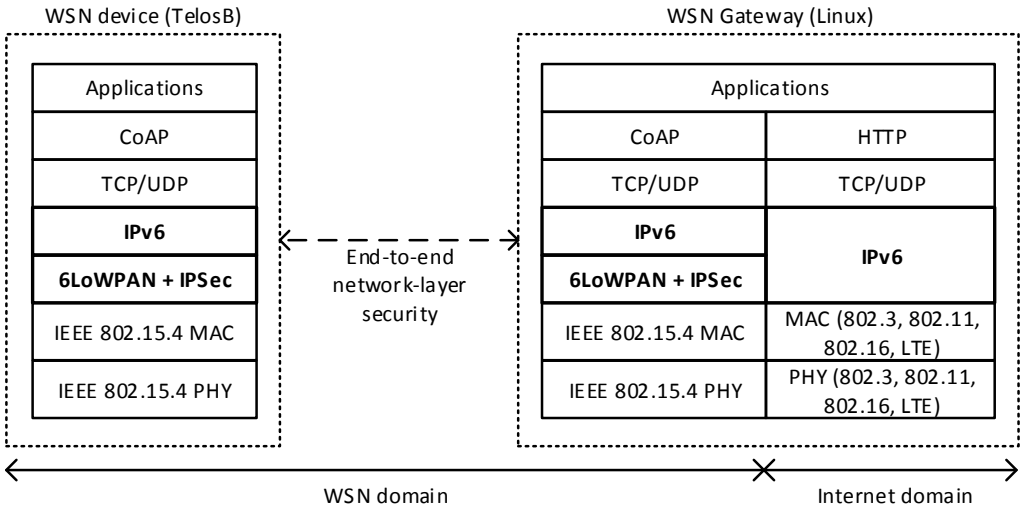

Figure 4.6 - Reference model for the evaluation of end-to-end 6LoWPAN security

As illustrated in Figure 4.6, the Linux host supports routing between an Ethernet IPv6 network and the IEEE 802.15.4 LoWPAN. Routing between the two communication environments is accomplished by employing a TelosB mote as a bridge for communications with the WSN. This Linux 6LoWPAN router also supports routing advertisements on the WSN, as required for the support of 6LoWPAN communications. As illustrated in the same figure, end-to-end communications at the network-layer are established with a WSN sensing device (TelosB). This model thus enables the evaluation of the impact of network-layer security in the presence of end-to-end communications with constrained wireless sensing devices running 6LoWPAN and the proposed compressed security headers.

Figure 4.7 – The TelosB wireless sensing platform

In Figure 4.7 we illustrate the TelosB wireless sensing platform employed in our experimental evaluation, which also provides a reference platform for the evaluation of the subsequent research proposals described in the thesis. The TelosB is a battery-powered sensing device supporting our TinyOS testing application and the 6LoWPAN security-capable BLIP networking stack.

The TelosB mote is currently considered a reference platform for the experimental evaluation of research proposals for WSN [194]. It is powered by a 16-bit RISC MSP 430 microcontroller with 10 Kbytes of RAM for program execution and 48 Kbytes of ROM for program storage. It also supports communications at 2.4 GHz and data transmissions at 250 Kbps. As it implements the IEEE 802.15.4 standard, it also provides hardware encryption and authentication using the AES/CCM cryptographic suite. This supports the cross-layer cryptographic basis, which we consider in our proposal and employ during our experimental evaluation study.

4.4.2 IDENTIFICATION OF APPROPRIATE CRYPTOGRAPHIC ALGORITHMS

The selection of the cryptographic algorithms that are appropriate to support 6LoWPAN security and to the resources of smart objects is an important requirement for our experimental study. Such algorithms or suites of algorithms enable smart objects to perform encryption, decryption, integrity verification and authentication operations, therefore allowing the processing and generation of information transported with 6LoWPAN using

security headers. For the identification of the appropriate cryptographic suites our goal is in fact twofold, as on the one side the usage of algorithms that are already accepted in the IP Security architecture would facilitate the integration of sensing applications with the Internet in a secure fashion, while on the other side we must carefully consider the effectiveness of the usage of such algorithms in resource-constrained smart objects. The selection of cryptographic algorithms regarding its impact on smart objects must nevertheless not be too conservative in this respect, as it may be expected that sensing devices will become more powerful and energy-efficient in a near future [69], and thus security mechanisms that have been showed to be unviable or marginally viable in the present may well be employed in a more mainstream fashion using future sensing platforms.

As the current IP Security architecture [99] may evolve to include 6LoWPAN applications in the future, we find it useful to analyze the effectiveness of the usage of its mandatory cryptographic algorithms with smart objects. The same applies to the algorithms that will most probably be adopted as mandatory in the future. The fact that the IP Security architecture allows end systems to agree on security algorithms and related security configuration parameters at the establishment of a security association is also in line with our requirement of adaptability for 6LoWPAN security.

Adaptable security mechanisms at the network layer may allow a 6LoWPAN smart object to select a cryptographic algorithm from a pool of alternatives and to decide how to use that algorithm, and this serves our goal on providing security mechanisms that allow the establishment of acceptable compromises between security and resources required from smart objects, two aspects we consider important for the support of future IoT sensing applications. In Table 4.3 we identify the cryptographic algorithms that are either currently defined as mandatory for the IP Security architecture [99] or that will probably be adopted as such in a near future.

Table 4.3 – Current and future mandatory cryptographic algorithms for the IP Security architecture

Security Header	Cryptographic algorithm	Usage	Status
ESP	3DES-CBC	Encryption	Mandatory
	AES-CBC	Encryption	Future
	HMAC-SHA1-96	Authentication	Mandatory
	AES-XCBC-MAC-96	Authentication	Future
	AES-CCM	Combined	Future
AH	HMAC-SHA1-96	Authentication	Mandatory
	AES-XCBC-MAC-96	Authentication	Future

As we can observe in Table 4.3, a shift is expected to take place towards AES-based cryptographic solutions. This is also in line with the fact that AES is already supported by various sensing platforms, and also motivated our decision on considering the usage of AES/CCM during the design of the 6LoWPAN security headers. The AES CCM* mode available with sensing platforms such as the TelosB allows for the separate support of security operations as required for the suites that employ AES in Table 4.3. As the usage of standard security and communications mechanisms may facilitate the secure integration of sensing applications with the Internet, our experimental evaluation study considers the usage of the algorithms in Table 4.3 in obtaining network-layer 6LoWPAN security. In Table 4.4 we describe how the above algorithms are employed in support of security using the compressed ESP and AH 6LoWPAN headers.

As Table 4.4 reflects, the isolated testing of the algorithm described in Table 4.3 would not be appropriate to evaluate the effectiveness of 6LoWPAN security, as in most deployments at least two algorithms will need to be supported, one providing confidentiality (through encryption and decryption) and the other providing authentication and integrity (through creation and verification of a MIC code or secure hash). AES/CCM was tested as available at the hardware in the TelosB mote, while the other algorithms were programmed in software using code optimized for small microcontrollers with the characteristics of the MSP 430.

Table 4.4 – Usage scenarios of cryptographic algorithms and 6LoWPAN security headers

Cryptographic suites	6LoWPAN header	Security provided
3DES-CBC	ESP	Confidentiality
AES-XCBC-MAC-96		Integrity, authentication
3DES-CBC	ESP	Confidentiality
HMAC-SHA1-96		Integrity, authentication
AES-CBC	ESP	Confidentiality
AES-XCBC-MAC-96		Integrity, authentication
AES-CBC	ESP	Confidentiality
HMAC-SHA1-96		Integrity, authentication
AES/CCM (HW)	ESP	Confidentiality, integrity, authentication
AES-XCBC-MAC-96	AH	Integrity, authentication
HMAC-SHA1-96	AH	Integrity, authentication
AES/CCM (HW)	AH	Integrity, authentication

The cryptographic block size and key size used with each algorithm are the values inherent of each cryptographic algorithm itself, and are also in line with the configurations required by the IP Security architecture [99]. Such values constitute therefore the most appropriate configuration to measure the effectiveness of our proposal. In particular, 3DES-CBC uses 192-bit keys to process 64-bit blocks. AES-CBC, AES-XCBC-MAC-96 and AES/CCM (using

hardware encryption) use 128-bit keys to process 128-bit blocks. HMAC-SHA1-96 uses 160-bit keys to process 512-bit blocks, with the original 160-bit authenticator generated being truncated to 96 bits, as specified in RFC 2404 [198]. Our AES-CBC software implementation also supports the AES-XCBC-MAC-96 algorithm, with the XCBC mode modifying the classic CBC mode as documented in RFC 3566 [199].

The fact that our tests employ software and hardware based cryptographic algorithms allows us to analyze the feasibility of 6LoWPAN security for a broader set of devices. This is relevant also if we consider that the IoT will include heterogeneous sensing devices which may or may not support hardware security. In the evaluation study we describe next we consider the usage of ESP to provide confidentiality together with authentication and integrity. Although we could have considered using ESP only for confidentiality, we believe that authentication and integrity are security properties that will be required for most of the applications in the IoT. In fact, the opposite may be truer, in that many applications will probably be able to dispense confidentiality and use only AH with its authentication and integrity assurances.

4.5 EXPERIMENTAL EVALUATION OF 6LOWPAN SECURITY

Our evaluation on the feasibility of 6LoWPAN security begins by analyzing its impact on 6LoWPAN payload space. Later in the chapter we concentrate on aspects such as its energy and computational requirements, which are determinant for the achievement of acceptable transmission rates and lifetimes for sensing applications. As previously discussed, our experimental evaluation considers the framework identified in Chapter 3.

4.5.1 OVERHEAD OF SECURITY ON 6LOWPAN PAYLOAD SPACE

As the payload space available to applications is an important factor in dictating the usefulness of 6LoWPAN in real usage scenarios, we start by analyzing the packet overhead of the usage of security in both tunnel and transport modes. We start by analyzing the payload space required for 6LoWPAN in various addressing compression scenarios and also with mesh and fragmentation headers. We must also consider the payload space required for the security headers previously described. The space required for such 6LoWPAN headers is described in Table 4.5. The values illustrated in this table are used during our following analysis on the impact of security on 6LoWPAN payload space.

The first 3 lines of Table 4.5 refer to the possible address compression scenarios that 6LoWPAN allows. With link-local unicast communications between 6LoWPAN smart objects sharing the same local link address, HC1 and HC2 6LoWPAN compression allows the compression of an UDP/IPv6 header down to 7 bytes. In this scenario, the version, traffic class, flow label, payload length and next header fields, and also the link-local prefixes of the IPv6 source and destination addresses are all elided, with the correspondent IPv6 suffixes being derived from the IEEE 802.15.4 header.

The second compression scenario corresponds to communications with an object outside of the local link while on the same 6LoWPAN, and in this case the IID (Interface Identifier) suffix of the source and destination addresses is also obtained from IEEE 802.15.4 addressing information, but the source and the destination prefixes must be carried inline. At the end, an additional 16 bytes are required for addressing information.

Table 4.5 – Payload space requirements for 6LoWPAN addressing, mesh, fragmentation and security

Scenario	Payload requirement
Link-local unicast	7 bytes
Outside of link-local scope	23 bytes
Outside of local LoWPAN	31 bytes
6LoWPAN AH	37 bits
6LoWPAN ESP	96 bits
Fragmentation	4 bytes / 5 bytes
Mesh addressing	5 bytes / 17 bytes

The third scenario is also the most useful in the context of the IoT, as in this case a 6LoWPAN-enabled smart object is able to communicate directly with an Internet host or with another remote smart object. In this scenario, 6LoWPAN is only able to elide the source address IID, with the remaining part of the source address and with the full destination IPv6 address carried inline, requiring in total 31 bytes.

The remaining lines in Table 4.5 refer to the payload space required for the other 6LoWPAN headers, including the two new security headers illustrated in Figures 4.2 and 4.3. Without considering the transportation of encrypted and authentication data, the authentication header requires 37 bits and the ESP header requires 96 bits. Fragmentation requires 4 bytes for the first fragment and 5 bytes for subsequent fragments, while the mesh addressing header required 5 or 17 bytes, depending on the usage of short (16-bit) or long (EUI-64 64-bit) addresses, respectively. Such values are also represented in Table 4.5 and are considered for the following analysis on the impact of 6LoWPAN security on the available packet payload space.

4.5.1.1 Impact of 6LoWPAN security without fragmentation and mesh headers

We illustrate in Figure 4.8 the impact on the 6LoWPAN payload space of security in tunnel and transport modes, without considering the usage of fragmentation or mesh addressing headers. We consider the addressing compression scenarios previously discussed and the transportation of authentication data of 8, 12 and 16 bytes in length. Figure 4.8 illustrates the payload space available in percentage of the maximum of 102 bytes available with IEEE 802.15.4 without link-layer security. In this figure we also illustrate the payload space

available when using 6LoWPAN headers without any security-related header or data. As fragmentation is not considered, the values illustrated in Figure 4.8 correspond to the maximum payload space that applications not using mesh addressing can use without requiring fragmentation from the 6LoWPAN adaptation layer.

As we considered the values from Table 4.5 for each of the three addressing compression scenarios, the payload space required for HC1 and HC2 compressed headers is already accounted for. It is visible that transport mode security is clearly less expensive than tunnel mode security in terms of the payload space required. For link-local communications or communications with systems outside of the local link but on the same 6LoWPAN, security leaves from 51 to 82 bytes to 6LoWPAN applications using ESP or AH in transport mode. When communications with the outside of the 6LoWPAN are required, the available space also in transport mode is between 43 and 58 bytes. Security in transport mode therefore provides acceptable availability on payload space, regardless of the security header and of the integrity and authentication level.

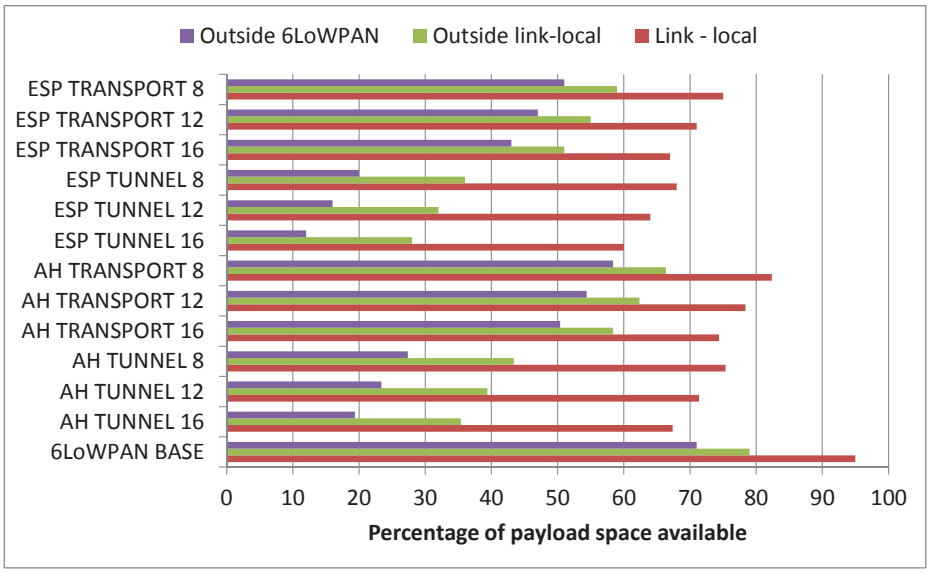

Figure 4.8 - Payload space available with 6LoWPAN security without mesh or fragmentation headers

Considering that tunnel mode in reality is not useful for link-local communications, we see that for communications outside of the local link the payload available is between 28 and 43 bytes and for communications outside of the 6LoWPAN it is between 12 and 27 bytes. Therefore, we consider that tunnel mode security for communications between devices on different LoWPANs is viable mainly for applications requiring moderate amounts of data. For communications with devices outside of the 6LoWPAN tunnel mode is viable but only for applications requiring the transportation of only a few bytes.

4.5.1.2 Impact of 6LoWPAN security with fragmentation

Our next evaluation considers the usage of a fragmentation header, and the obtained values are illustrated in Figure 4.9. As the overhead imposed from the fragmentation header is only of 5 bytes per 6LoWPAN packet, our previous conclusions remain valid regarding security in transport mode, as the payload space remains between 38 and 77 bytes.

As for tunnel mode security, it leaves between 23 and 38 bytes for communications with nodes outside the local link, and between 7 and 22 bytes for communications with other 6LoWPAN or IPv6 hosts. We are therefore able to realize that for tunnel mode the space required for the fragmentation header poses an extra pressure on the usefulness of this security mode. ESP in tunnel mode can only be considered viable if employed with an 8-byte or 12-byte MIC code.

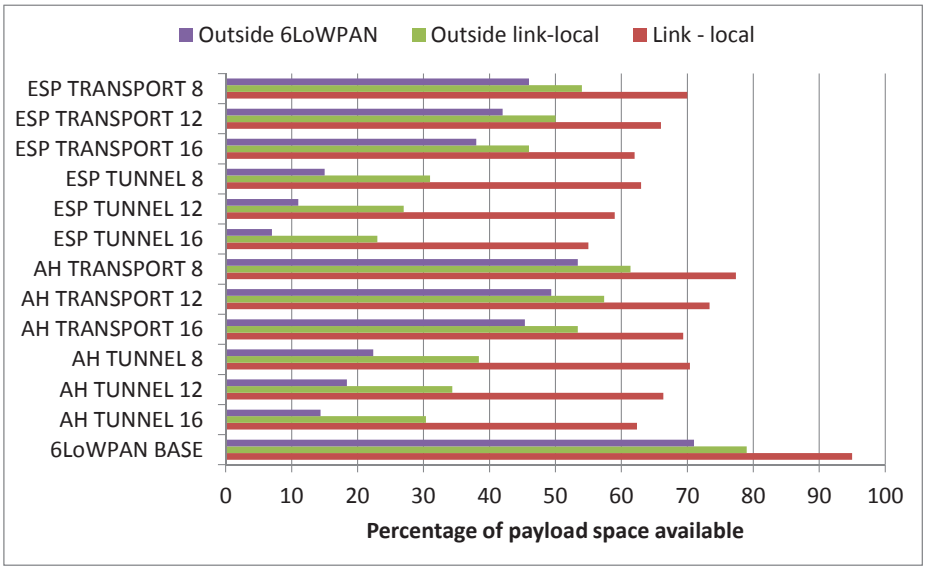

Figure 4.9 - Payload space available with 6LoWPAN security with fragmentation information

In conclusion, communications requiring fragmentation can use transport mode security viably with all addressing compression scenarios. Tunnel mode security is valid for communications with nodes outside of the local link for applications requiring the transmission of small amounts of sensing data, while for communications with Internet hosts it is viable mainly for applications that don't require confidentiality and therefore are able to use AH to protect the transportation of small amounts of data. For applications that do require confidentiality, ESP is a viable choice only if lower integrity and authentication assurances are acceptable, more precisely using ESP with a MIC code with 12 or (preferably) 8 bytes.

4.5.1.3 Impact of 6LoWPAN security with mesh addressing

Figure 4.10 illustrates the impact of 6LoWPAN security on payload space when the transportation of mesh addressing information is required. We consider the usage of a mesh-addressing header with 17 bytes, corresponding to mesh addresses obtained from the EUI-64 addresses of sensing devices.

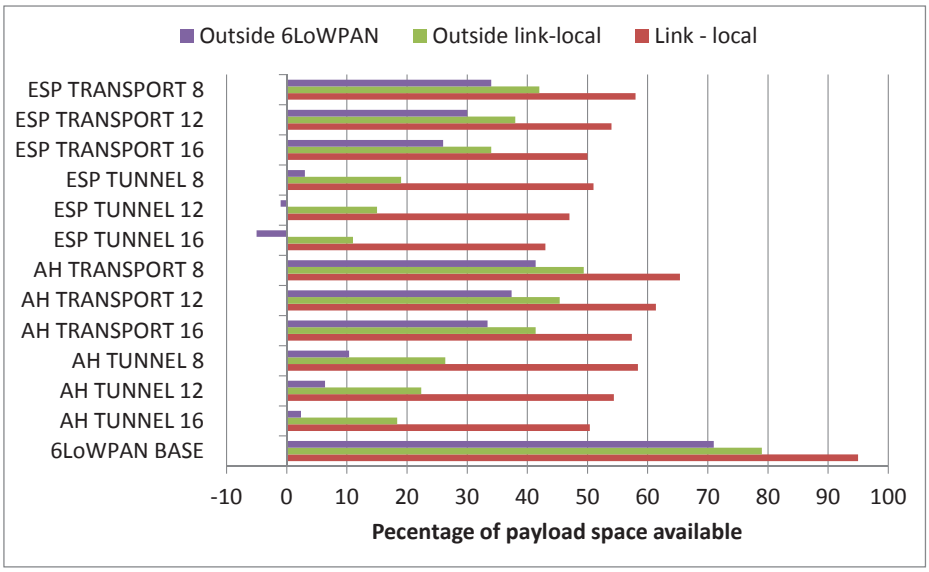

Figure 4.10 - Payload space available with 6LoWPAN security with mesh information

We can observe that, with mesh addressing and security in transport mode, the payload space available for 6LoWPAN applications drops to between 26 and 65 bytes. As for tunnel mode security, communications with nodes outside of the local link remain possible if small amount of data are transmitted, as in this case only from 11 to 26 bytes are available. For communications with nodes outside of the 6LoWPAN, tunnel mode is viable only with AH transporting MIC codes with 8 bytes, which even so only provides 10 bytes of payload space. The remaining tunnel security usage modes do not provide enough payload space, or the support of 6LoWPAN security headers would require the availability of more than 102 bytes. In conclusion, in the presence of mesh addressing information, security in transport mode remains valid but only for applications requiring the transmission of a moderate amount of data. On the other hand, tunnel mode security is viable only for applications requiring low integrity and authentication assurances or which do not need confidentiality at all.

4.5.1.4 Impact of 6LoWPAN security with fragmentation and mesh information

The worst usage scenario for 6LoWPAN security in terms of its impact on payload space corresponds to the simultaneous usage of fragmentation and mesh addressing headers, and is illustrated in Figure 4.11. The values illustrated in this figure corroborate some of our

previous conclusions. We are able to conclude that transport mode security remains a valid usage mode for small amounts of data, as in this case between 21 and 60 bytes are available to transport data from 6LoWPAN applications. Tunnel mode is clearly the most affected mode by the lack of available payload space, and in practice can be considered unviable for communications with nodes outside of the 6LoWPAN, since in this case even AH with a MIC code of 8 bytes would only leave 5 bytes of data payload space, which may be insufficient for most sensing applications on the IoT. We can see that with several configurations there is not enough space to accommodate even just the 6LoWPAN headers.

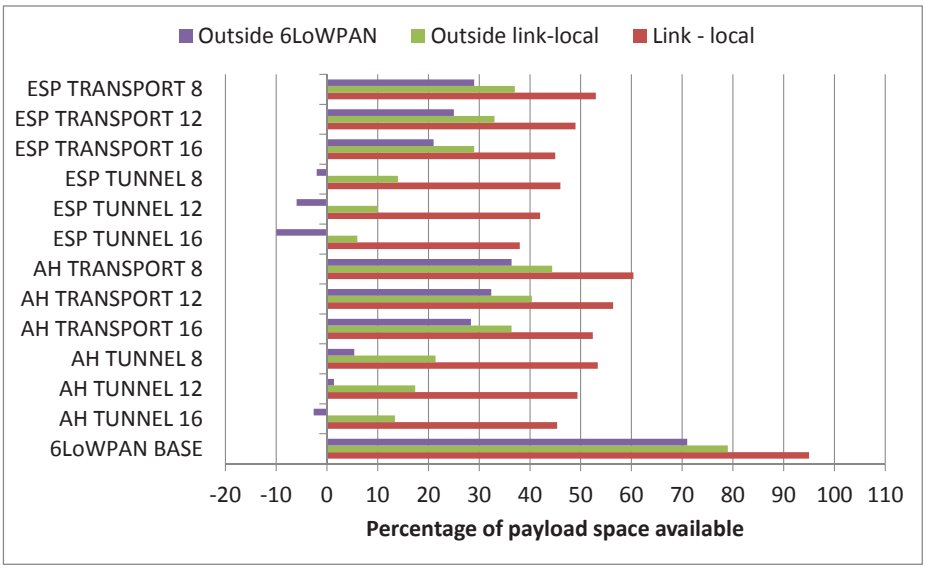

Figure 4.11 - Payload space available with 6LoWPAN security with fragmentation and mesh information

Regarding tunnel mode communications with nodes outside of the local link, it still can be considered viable for very small amounts of transmitted data, as between 6 and 21 bytes are available. This is especially true for applications that only require authentication and integrity, as with AH in tunnel mode between 13 and 21 bytes are available. The previous analysis on the impact of 6LoWPAN security on the payload space available for applications allows us to identify the viable usage modes of the proposed compressed security headers, as we discuss next.

4.5.1.5 Viable usage modes of 6LoWPAN security

Other than the identification of the viable usage modes in respect to the impact of security on 6LoWPAN payload space, we need to identify the usage modes of security that will in fact be useful in the context of IoT sensing applications. Table 4.6 identifies such modes, from the perspective of communications initiated by a 6LoWPAN-enabled WSN sensing device.

Table 4.6 – Viable usage scenarios for 6LoWPAN security in the IoT

From \ To	6LoWPAN device on same local link	6LoWPAN device outside of local link	Device outside the 6LoWPAN
6LoWPAN device	AH/ESP transport mode	AH/ESP transport mode AH/ESP tunnel mode via 6LoWPAN router	AH/ESP transport mode AH/ESP tunnel mode via security gateway

Table 4.7 - Viable usage modes of 6LoWPAN network-layer security

							Available payload space			MIC code size		
	Mesh	Frag	Transport	Tunnel	AH	ESP	High	Medium	Low	16	12	8
Link-local			✓		✓	✓	✓			✓		
		✓	✓		✓	✓	✓			✓		
	✓		✓		✓	✓		✓		✓		
Outside local link			✓		✓	✓	✓			✓		
				✓	✓	✓		✓		✓		
		✓	✓		✓	✓	✓			✓		
		✓		✓	✓	✓			✓	✓		
	✓		✓		✓	✓		✓		✓		
	✓			✓	✓	✓			✓	✓		
	✓	✓	✓		✓	✓			✓	✓		
	✓	✓		✓	✓				✓	✓		
Outside 6LoWPAN			✓		✓	✓	✓			✓		
				✓	✓	✓			✓	✓		
		✓	✓		✓	✓	✓			✓		
		✓		✓	✓				✓	✓		
		✓		✓		✓			✓		✓	
	✓		✓		✓	✓		✓		✓		
	✓			✓	✓				✓			✓

The scenarios identified in Table 4.6 consider the usage of two types of 6LoWPAN routers, one acting as a 6LoWPAN security gateway and the other as a 6LoWPAN router. A security gateway is a device without the resource constraints that are typical of smart objects, and that as such can be used to aid in the integration and interconnection of a network of smart objects with the Internet, as considered in the WSN Gateway of Figure 4.6. A security gateway may implement various security mechanisms to protect the network of smart objects from the Internet, among which the processing of network-layer security in communications with Internet devices and smart objects. On the other hand a 6LoWPAN router is a more limited device, supporting distributed sensing applications and allowing routing and enforcing security mechanisms for communications between different 6LoWPAN domains.

Our evaluation and the identification of the useful usage modes of 6LoWPAN security allow us to identify the main characteristics of the viable usage modes of the proposed security headers. In this context, viability means that enough payload space is left for applications while guaranteeing the usage of strong authentication codes. It is clear that, without employing a mesh routing protocol, 6LoWPAN network-layer security is viable in all usage modes as long as applications are able to adapt to the payload space available.

The classification in Table 4.7 reflects a qualitative evaluation for which preference is given to the usage of strong authentication and integrity codes, whenever possible. We also classify the various usage modes of 6LoWPAN security in terms of the payload space left available for applications. Other practical usage scenarios can nevertheless be identified to be viable for the IoT, for example considering that some applications may only need to use smaller authentication codes or use ESP without authentication and integrity.

4.5.1.6 Memory footprint of 6LoWPAN security

As memory is also a limited resource on smart objects, our evaluation study proceeds with the analysis of the memory footprint of our implementation of 6LoWPAN security in TinyOS and BLIP, while supporting the cryptographic suites previously identified. As discussed in Chapter 3 and illustrated in Figure 3.4, the impact of new research proposals on the limited memory space available in sensing platforms is a fundamental aspect of its viability.

For the purpose of measuring the impact of 6LoWPAN security on the memory of the TelosB, we employ different versions of a base TinyOS application in our experimental evaluation. Such versions support the security-enabled 6LoWPAN stack together with each of the cryptographic suites implemented in software or available in hardware. We separately measured the RAM and ROM memory necessary with each version of the testing application, as both types of memory are very limited on constrained sensing platforms.

In Figure 4.12 we describe the memory footprint of 6LoWPAN security with each of the cryptographic suites. The values illustrated in this figure are in percentage of the total of RAM and ROM memory available on the TelosB (10Kbytes of RAM and 48Kbytes of ROM).

For comparison purposes, we also evaluate and illustrate the memory required for a base BLIP networking stack with support for the processing of 6LoWPAN security headers, but without any cryptographic algorithm. This base application allows us to measure the impact of the different cryptographic algorithms on the memory required from a sensing device.

When compared to the baseline usage profile, we can observe that ESP using cryptographic suites based on 3DES-CBC, both together with HMAC-SHA1-96 and AES-XCBC-MAC-96, is very demanding particularly in terms of the required ROM memory, leaving almost no ROM memory left available to accommodate other mechanisms or applications. The large ROM memory footprint of 3DES-CBC is mostly due to the usage of large S-Boxes by the algorithm. We may also note that the usage of the hardware-level encryption doesn't come without a non-negligible overhead on memory, particularly in terms of ROM memory, as code is necessary to support the usage of link-layer standalone encryption using the cc2420 chip of the TelosB. Regarding the support of AES/CCM in standalone mode, we employed the standalone hardware encryption code available from the Shanghai Jiao Tong University [200].

In contrast to the inline mode, standalone encryption allows applications to perform hardware encryption and decryption without requiring the transmission or reception of a packet by the link-layer, given that such operations are controlled at a higher level in BLIP. From Figure 4.12 we can also observe that security suites based on the usage of AES-CBC with HMAC-SHA1-96 or AES-XCBC-MAC-96 broadly present a similar impact on the required ROM memory, while requiring only a few more bytes of RAM memory compared to the base 6LoWPAN security application.

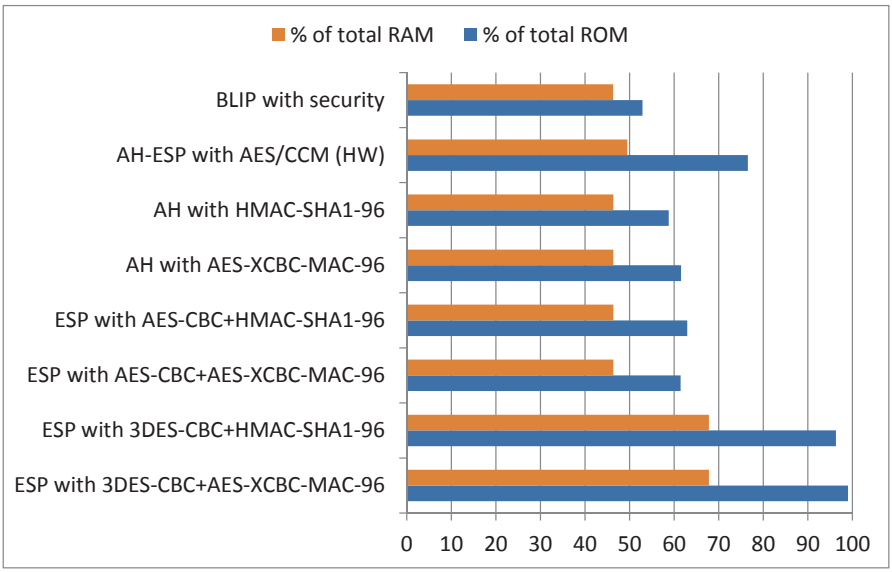

Figure 4.12 - Memory footprint of 6LoWPAN security

From this analysis we are therefore able to conclude that, regarding requirements of memory available in resource-constrained sensing devices, AES appears as a natural candidate in providing an alternative to 3DES-based security suites. AES provides good security both in the CCM and CBC modes with a lower memory footprint. Of course, the usage of AES/CCM on devices that support hardware encryption presents the advantage of freeing more memory for other mechanisms and applications, and in this case AES-CBC can probably be dispensed. Regarding the support of integrity and authentication, AES-XCBC-MAC-96 represents a good choice regarding the required memory, also because it provides superior security to HMAC-SHA1-96 with a similar memory footprint. It is interesting to note that, excluding the cryptographic suites using 3DES-CBC, security in general causes a relatively low overhead in terms of memory. Overall, we verify that the impact on the available memory of sensing devices therefore doesn't compromise the adoption of network-layer security mechanisms in the context of the 6LoWPAN adaptation layer.

4.5.2 Energy overhead of 6LoWPAN security

As many sensing applications are designed with battery-powered sensing devices in mind, the energy required from such devices to perform security operations is a critical aspect, given that it influences the expected lifetime of the device and of the overall sensing application. Energy is therefore an important evaluation criterion of the feasibility of any communications or security proposal for smart objects, and one that we evaluate for the usage of 6LoWPAN security. As previously discussed and Figure 3.4 illustrates, we may consider the energy required both for the processing of security and for the transmission of security-related data, given that such aspects at the end impact on the limited energy available in sensing platforms and influence the lifetime of the application.

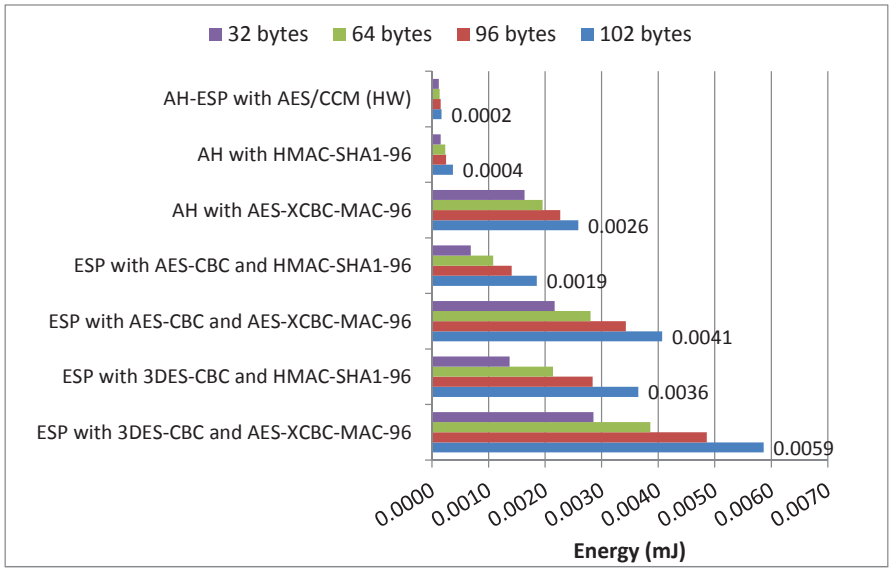

Figure 4.13 - Energy required by 6LoWPAN security

In Figure 4.13 we represent the experimentally obtained values of the energy required to process security for a 6LoWPAN packet with 32, 64, 96 or 102 bytes, using the previously identified cryptographic suites. The energy values represented are in millijoules (mJ), and the labels illustrate the energy required for the processing of security in the case of a fully sized 102 bytes 6LoWPAN packet. Energy was obtained using experimental measurements of the voltage across a current sensing resistor placed in series with the battery pack and the circuit board of the TelosB.

The values illustrated in Figure 4.13 allow us to perform a qualitative analysis of the impact of security on the energy available in smart objects, while the values obtained experimentally are later used in the context of our quantitative study on the lifetime of particular sensing applications. Please note that the values represented in this figure already include the energy required for the processing of 6LoWPAN security headers (for its interpretation and construction) in the BLIP networking stack of the TinyOS operating system. Also, note that we don't represent the energy required for the processing of a 6LoWPAN packet without any cryptographic operation, because such value is negligible when compared to the energy required for security.

The values represented are considered irrespective of the size of the MIC code generated by a specific authentication algorithm. This is due to the fact that AES-XCBC-MAC-96 and HMAC-SHA1-96 always generate 12-byte MIC codes, while for hardware AES/CCM the energy required for the generation of a 16, 12 or 8 bytes MIC using standalone hardware encryption is the same, as in this case hardware security is designed to operate on blocks of 128 bits (16 bytes).

From Figure 4.13 we again observe that cryptographic suites employing 3DES-CBC are clearly less efficient in terms of the energy required, as for example 0.0059mJ are required to encrypt a 102-byte 6LoWPAN packet and generate the correspondent MIC code using AES-XCBC-MAC-96. Regarding the support of authentication and integrity, the difference between HMAC-SHA1-96 and AES-XCBC-MAC-96 is notorious, which allows us to conclude that the bigger security provided by AES-XCBC-MAC-96 probably does not compensate its impact on energy, when compared to the alternative HMAC-SHA1-96. In fact, HMAC-SHA1-96 only requires 0.00037 mJ to encrypt a 102-byte 6LoWPAN packet, while with AES-XCBC-MAC-96 0,0026 mJ are required to process the same packet. From Figure 4.13 we can also confirm that standalone hardware encryption using the cc2420 chip of the TelosB is extremely efficient in terms of energy, and should therefore provide a superior solution to support integrity, authentication and encryption for 6LoWPAN security in devices where hardware security is available. As expected, encryption with AES/CCM is also clearly superior to AES-CBC implemented purely in software.

It is also interesting to note the superior performance of HMAC-SHA1-96 even when implemented in software, as in reality it is not much more expensive than hardware-based AES/CCM. Finally, our measurements reveal a better performance in terms of energy when

compared with [151], although we must note that in this proposal energy is estimated and authors only consider link layer security and data payloads of up to 64 bytes. Our measurements are obtained experimentally and, as our overall goal is to evaluate the effectiveness of network-layer security in the context of the IP Security architecture, we also address the other (current and future) mandatory security suites.

4.5.3 COMPUTATIONAL OVERHEAD OF 6LOWPAN SECURITY

Other than the memory and energy required to process 6LoWPAN security, the computational effort required from smart objects for security operations is also a relevant aspect. As advanced mechanisms such as multi-threading are usually not supported in low-end microcontrollers such as the MSP430 of the TelosB, the computational time required to process security for a 6LoWPAN packet directly influences the maximum communications rate that a smart object can expect to achieve for a given sensing application. We have reflected such aspects in Figure 3.4 and its impact on the framework previously discussed and illustrated in Figure 3.3.

In Figure 4.14 we illustrate the computational time required for the processing of a 6LoWPAN packet of different sizes, considering the cryptographic suites previously identified. The values illustrated are in milliseconds (ms) and, as for our previous analysis, we do not represent the computational time required for the processing of 6LoWPAN security without any cryptographic operations, given that such value is negligible when compared to the effort required to process security for the same 6LoWPAN packet.

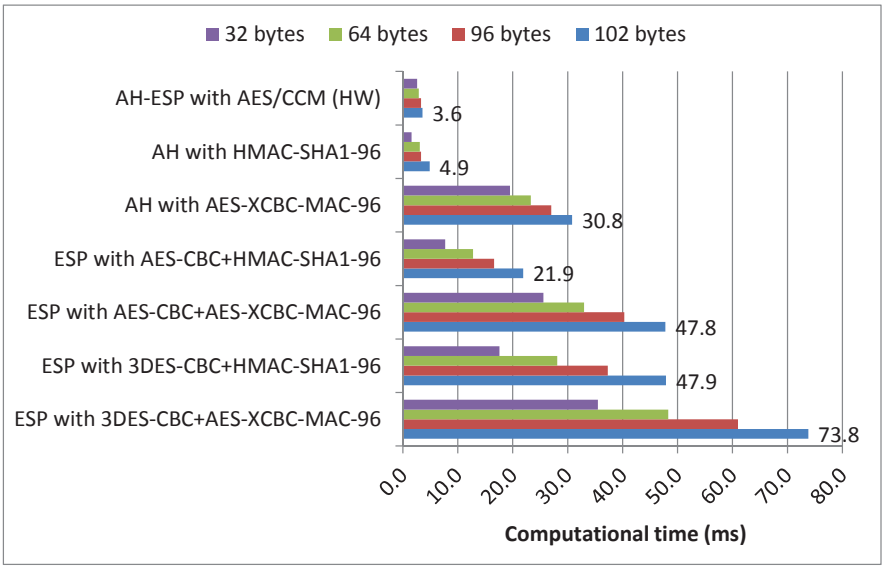

Figure 4.14 - Computational time required by 6LoWPAN security

The labels in Figure 4.14 indicate the computational time required to process security for a 102 bytes 6LoWPAN packet. Please also note that the values illustrated in this figure are

total values, measured from the reception of a 6LoWPAN packet to the time when the respective cryptographic algorithm finishes processing the packet, and therefore represents the total computational effort required to process security for a 6LoWPAN packet in the TelosB sensing platform. The values were obtained from measurements using the 32 KHz internal oscillator of the TelosB, which is accessible to TinyOS applications via the *counter* programmatic interface.

Comparing Figures 4.13 and 4.14, as expected we are able to observe a close relationship of energy consumption and computation time. The fact that the results are not directly drawn from one another may be explained by differences in the computational efficiency inherent of each algorithm and of the software implementations employed on our experimental evaluation. From Figure 4.14 we observe that the most demanding cryptographic suite appropriate to ESP is 3DES-CBC when used together with AES-XCBC-MAC-96, requiring in total approximately 74 ms for processing a 102 bytes 6LoWPAN packet.

Regarding the support of integrity and authentication, HMAC-SHA1-96 appears as the most efficient algorithm available in software. AES-XCBC-MAC-96, although providing greater security is much mode demanding, requiring approximately 31 ms to process a fully-sized 6LoWPAN packet. Standalone hardware encryption appears again as the most efficient solution, and in this case the time required to process the same 6LoWPAN packet was measured as 3.6 ms. As AES/CCM implements the CCM* combined mode, this in reality represents the time necessary to encrypt, decrypt or generate the authentication data for a 6LoWPAN packet. Regarding AES implemented in software, we observe that AES-CBC is clearly more demanding, although better than 3DES-CBC in providing confidentiality.

4.6 OVERALL EVALUATION OF 6LOWPAN SECURITY

Our experimental evaluation study on the resources required from constrained sensing devices to support 6LoWPAN security allows us to consider its impact in more concrete application scenarios. We therefore proceed to discuss the viability of our proposal regarding sensing applications with diverse requirements in terms of security, communication rates and lifetime of sensing devices. This evaluation strategy was previously discussed in Chapter 3 and is illustrated in Figure 3.4, and provides the information necessary for the employment of the framework illustrated in Figure 3.3 with particular applications.

4.6.1 IMPACT OF 6LOWPAN SECURITY ON THE COMMUNICATIONS RATE OF SENSING DEVICES

As sensing applications may be very diverse in terms of the employed communications rate, we find it appropriate to evaluate if 6LoWPAN security may represent a bottleneck in this respect. This is an important evaluation aspect since, as we have seen, security introduces a non-negligible computational overhead on constrained smart objects with the

characteristics of the TelosB, which are unable to process packets received or waiting transmission while the microcontroller is busy performing cryptographic operations.

When considering communications using IEEE 802.15.4 at 250Kbit/s, we realize that the impact of the computational time required for security on the maximum transmission rate is much larger than the impact on the time required for the transmission of a few more bytes required for the 6LoWPAN security headers and the MIC code. What we cannot exclude from consideration is the overhead introduced by IEEE 802.15.4 addressing on the bandwidth available for 6LoWPAN. This overhead represents 19.6% of the total bandwidth, since 25 bytes are required for link-layer information with each 127 bytes 6LoWPAN packet, as is illustrated in Figure 4.1.

In Figure 4.15 we illustrate the maximum transmissions rate, which can be achieved by sensing application employing 6LoWPAN security, considering the usage of the various cryptographic suites with 6LoWPAN packets with 32, 64, 96 or 102 bytes. The values obtained and illustrated in this figure are in packets per second and are valid for AH and ESP in both tunnel and transport modes, together with the transmission of the authentication data, if required. The values illustrated in Figure 4.14 consider the time required for the processing of 6LoWPAN headers (including security) on the TelosB, which we have experimentally measured as 0.09 milliseconds. We do not represent the values for the maximum transmission rate without security, but those values are fundamentally greater, in particular 252 packets per second for 102-byte 6LoWPAN packets, 268 for 64-byte packets, 402 for 96-byte packets and 803 for 32-byte packets.

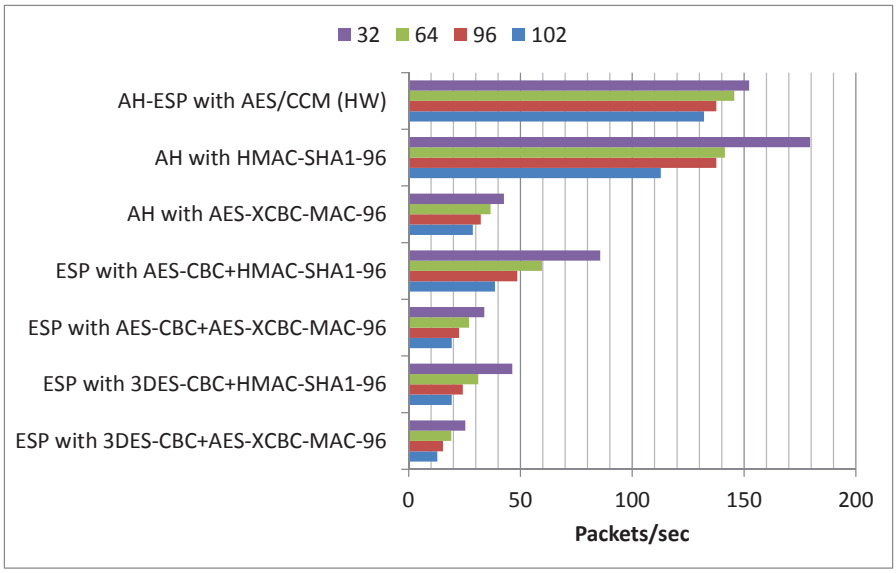

Figure 4.15 - Maximum transmission rate with 6LoWPAN security

From Figure 4.15 we can observe that the impact of 6LoWPAN security is particularly relevant when transmitting smaller packets. For larger packets (for example for packets

measuring from 64 to 102 bytes) security still allows acceptable transmission rates, particularly using cryptographic suites based on AES/CCM and SHA1. One possible design approach for 6LoWPAN applications would therefore be to employ aggregation of sensing data whenever applicable, as this allows reducing the impact of security on the communications rate.

As illustrated in Figure 4.15, security configurations employing 3DES cause a greater impact on the maximum available communications rate. For applications requiring only integrity and authentication, AH using HMAC-SHA1-96 or hardware AES/CCM are good choices. HMAC-SHA1-96 appears in fact again as a superior choice in providing such security properties using a software implemented security algorithm.

Again regarding authentication and integrity, HMAC-XCBC-MAC-96 causes a greater impact as can be observed in Figure 4.15. It can be nevertheless an appropriate choice for applications requiring lower transmission rates, as it provides security superior to HMAC-SHA1-96. In general, we observe that acceptable transmission rates are achievable using 6LoWPAN security. As applications are usually designed in order to save energy by not requiring large transmission rates, the limits identified in Figure 4.15 should not represent a limitative factor of the applicability of 6LoWPAN security.

4.6.2 IMPACT OF 6LOWPAN SECURITY ON THE LIFETIME OF SENSING APPLICATIONS

Other than the impact of 6LoWPAN security on the communication rate smart objects are able to achieve, it is also important to analyze its impact on the lifetime of such sensing devices, as it in the end may determine the lifetime of a given sensing application. The importance of this evaluation is related to the fact that most sensing applications designed for the IoT will only be viable if able to operate in unattended mode during a long period of time, as in many situations smart objects are devices for which it is difficult or impossible to replace batteries during long periods of time.

As for our previous evaluation studies, our overall goal is to analyze if acceptable compromises can be achieved between the usage of resources on smart objects and security. In Figures 4.16 to 4.19 we illustrate the lifetime that a TelosB sensing device can achieve using 6LoWPAN security to process packets with different sizes and using different communications rates. In particular, we consider the usage of lower transmission rates (from 1 to 10 transmitted packets per second) and higher transmission rates (from 20 to 200 transmitted packets per second). We also consider the processing of 32 and 102 bytes 6LoWPAN packets, as this represents two complementary scenarios in terms of the size of 6LoWPAN packets processed in such communications. The achievable lifetime are represented in days for each security and usage configurations, and due to the wide range of values we use a logarithmic scale for the representation of the obtained values.

The values illustrated in Figures 4.16 to 4.19 are derived from our experimentally obtained values using a TelosB mote powered using two new AA LR6-type batteries. As for our

previous evaluation, we also consider the energy required for the processing of 6LoWPAN headers in each packet (including security headers), which was experimentally measured as 0.007 nanojoules (nJ) per 6LoWPAN processed packet with security. This value reflects the total energy required for the processing of a 6LoWPAN packet, from the invocation of the transmission of the packet using the BLIP networking stack to the time of the completion of its transmission. For comparison purposes, Figures 4.16 to 4.19 also illustrate the expected lifetime when using 6LoWPAN communications without security.

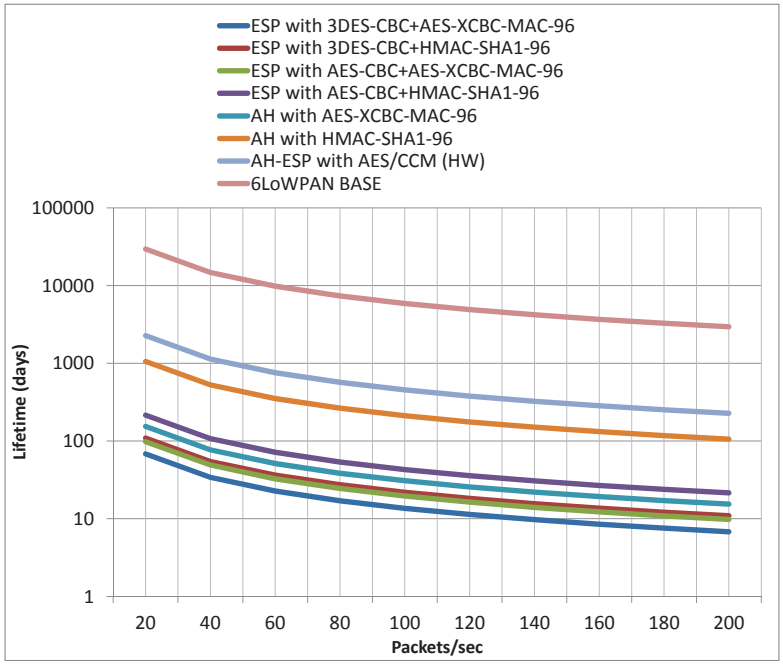

Figure 4.16 - Lifetime of a sensing device when processing security for a 102-byte 6LoWPAN packet (higher communication rates)

As with our study on the impact of security on the transmissions rate, we do not consider the extra energy required for the transmission of 6LoWPAN headers in tunnel mode versus transport mode, neither for the transmission of authentication data. This is due to the fact that the energetic cost of the transmission or reception per bit with the TelosB is very small, and consequently represents a negligible impact on the lifetime of the applications and doesn't influence our analysis and conclusions. We observe that AES-CCM and HMAC-SHA1-96 for integrity and authentication allow much higher lifetime of the sensing device, particularly for applications requiring lower transmission rates.

From the results illustrated in such figures, 3DES-CBC appears again as a bad choice independently of the transmission rate, while cryptographic suites employing AES-CBC and XCBC-MAC-96 in software are possible choices if applications requiring lower transmission rates. For the processing and transmission of smaller (32 bytes) 6LoWPAN packets, the impact of authentication and integrity using AH with HMAC-SHA1-96 is almost equal to

hardware AES/CCM. HMAC-SHA1-96 can therefore be a good alternative in providing authentication and integrity for applications transmitting smaller data payloads, in particular for the usage with smart objects that do not support hardware encryption.

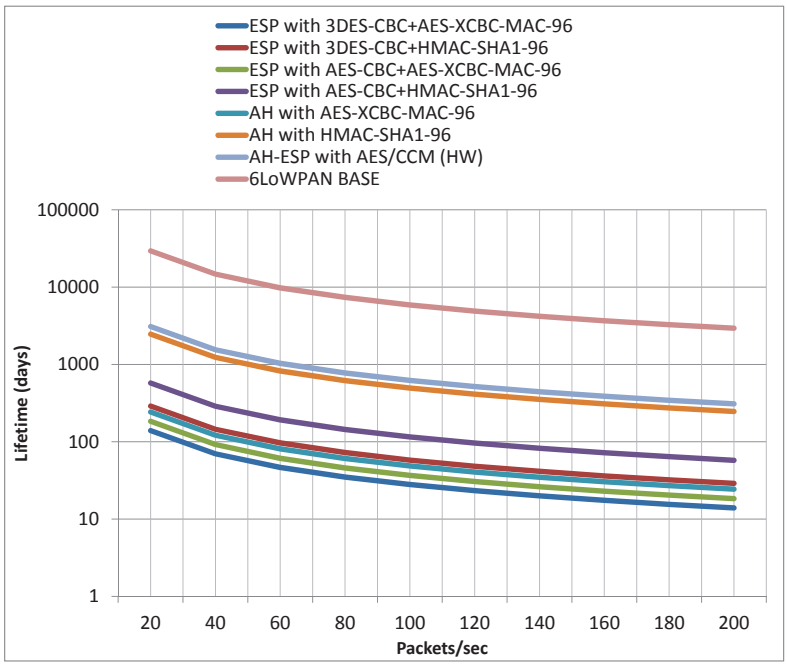

Figure 4.17 - Lifetime of a sensing device when processing security for a 32-byte 6LoWPAN packet (higher communication rates)

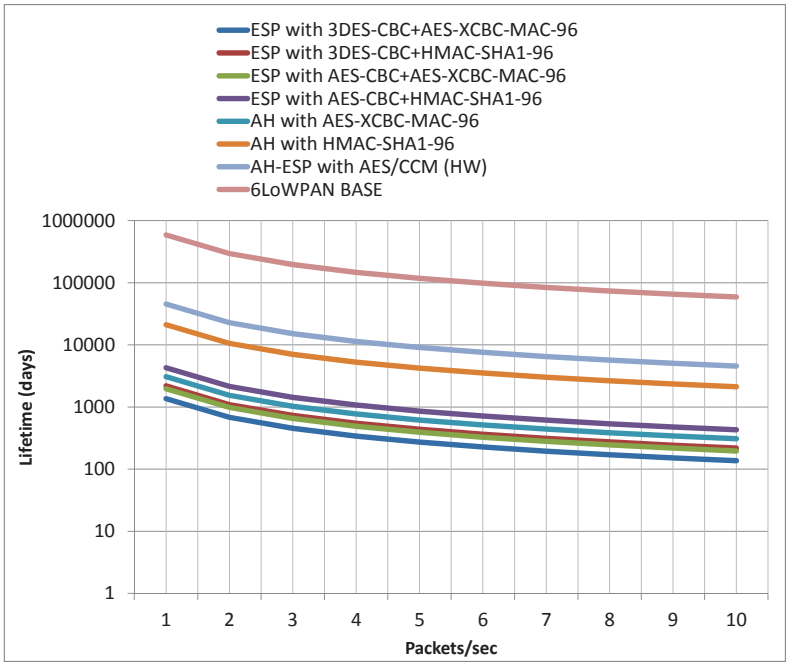

Figure 4.18 - Lifetime of a sensing device when processing security for a 102-byte 6LoWPAN packet (lower communication rates)

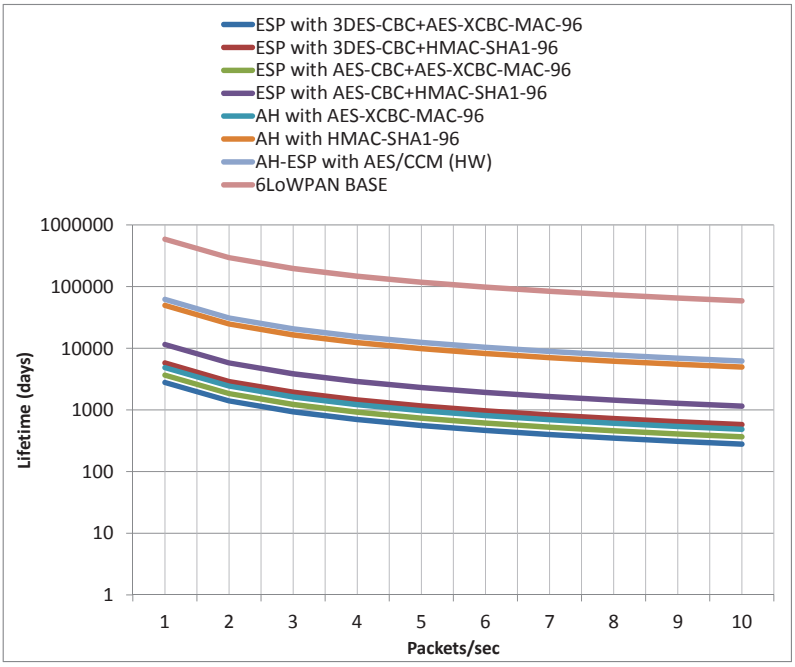

Figure 4.19 - Lifetime of a sensing device when processing security for a 32-byte 6LoWPAN packet (lower communication rates)

From the previously illustrated results we can observe that security introduces a non-negligible impact on the lifetime of applications, when compared to the baseline usage scenario without network-layer security. Nevertheless, it can also be observed that the achievable lifetime using 6LoWPAN security is in general very good, in particular for sensing applications that require lower transmission rates. Thus, for future IoT sensing applications employing lower transmission rates we are able to see that the other cryptographic suites based on the usage of software AES-CBC and of AES-XCBC-MAC-96 are also viable. It is therefore perfectly possible to employ such cryptographic suites both in software and hardware (for smart objects supporting it) with 6LoWPAN while not critically impacting the lifetime of the sensing device. This factor, together with the conclusions obtained in our previous evaluation studies, allows us to enforce our conviction on the effectiveness of the usage of 6LoWPAN security in the context of an appropriate architecture supporting security for sensing applications using Internet-integrated WSN.

4.7 SUMMARY

The IPv6 protocol and the 6LoWPAN adaptation layer can play a major role in the evolution of the Internet as we know it today. As the Internet embraces sensorial capabilities, new and exciting applications may come to life that will require and benefit from the availability of end-to-end network-layer communications between smart objects and other sensing devices or Internet hosts. Such communications can only be viably employed if appropriate security mechanisms are adopted.

In the current chapter we propose and experimentally evaluate new compressed security headers for the 6LoWPAN adaptation layer, and such headers were designed in a way such as to ease its integration with the IP Security architecture as it evolves in the future. As we have verified, 6LoWPAN security can be viably employed in various configurations by sensing applications with different requirements in terms of communications rates and payload space. We thus observe that network-layer security can be a reality for sensing applications using Internet-integrated WSN.

As the proposed mechanisms allow for the usage of different security configurations, security can be adapted to the particular requisites of each application, therefore allowing the establishment of acceptable compromises between security and the usage of resources on constrained sensing platforms. As discussed in Chapter 3, the analysis of the impact of end-to-end network-layer security as described in the present chapter enables the employment of the most appropriate security mode according to application security and functional requirements, which may be expressed by appropriate application profiles. In the same context, it also lays the ground for the design an adoption of future mechanisms enabling the dynamic reconfiguration of end-to-end security for Internet-integrated WSN applications.

This chapter provided a description of our approach to enable end-to-end security for Internet-integrated sensing applications. It started by describing the proposed compressed security headers designed for the 6LoWPAN adaptation layer. Afterwards, the proposed compressed security headers were experimentally evaluated in various usage contexts, and in consequence various usage modes were identified that may viably enable end-to-end security at the network-layer for applications employing Internet-integrated WSN.

5 END-TO-END TRANSPORT-LAYER SECURITY WITH MUTUAL AND DELEGATED PUBLIC-KEY AUTHENTICATION [4]

In this chapter we describe our research proposals to address security at the transport layer in the context of Internet-integrated WSN. Our proposal is complementary to network-layer 6LoWPAN security as discussed in the previous chapter, in the sense that we address end-to-end security at the transport-layer in a truly transparent fashion from the perspective of the communicating entities. At the same time, we seek to provide a solution to alleviate constrained sensing devices from computationally demanding security procedures.

We begin by evaluating the impact of transport-layer security as currently proposed for 6LoWPAN-based communications, a study that enables the identification of the authentication and key agreement phase to be particularly problematic in the light of the resources currently available on sensing platforms. Our research proposal also addresses security against attacks originated at external or Internet entities, and lays the ground for the support of transparent mobility from the perspective of end-to-end security. As in the previous chapter, the proposed research solutions are experimentally evaluated and employed considering the methodology and reference model discussed in Chapter 3.

5.1 INTRODUCTION

Many of the applications currently envisioned for the Internet of Things (IoT) are critical in respect to security, being it security of its users, of the processed data or of the communications taking place between devices. Despite this fact, such applications will interact with physical phenomena by employing very constrained sensing platforms and low-energy wireless communications, aspects that seriously complicate the design and adoption of appropriate security mechanisms. As wireless sensor networks (WSN) applications are starting to require interconnection with the Internet at some degree,

[4] *This chapter has supported the following publications:*

- Granjal J, Monteiro E, Silva J. *On the effectiveness of end-to-end security for Internet-integrated sensing applications* (best paper award), The IEEE International Conference on Internet of Things, iThings 2012
- Granjal J, Monteiro E, Silva J. *End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication*, IFIP Networking 2013
- Granjal J, Monteiro E, Silva J. *A framework towards adaptable and delegated end-to-end transport-layer security for Internet-integrated Wireless Sensor Networks*, Second Joint ERCIM eMobility and MobiSense Workshop - WWIC 2013
- Granjal J, Monteiro E, Silva J. *On the Feasibility of Secure Application-Layer Communications on the Web of Things*, LCN 2012

end-to-end communications between constrained sensing devices and other Internet entities will be a fundamental requirement of many sensing applications. Such communications may take place at various protocol layers, with transport-layer security promising to play also an important role in this context.

As previously discussed, the support of end-to-end security involving constrained sensing devices will represent a fundamental enabling factor of many IoT sensing applications, as it may provide security even when the underlying network infrastructure is only partially under the user's control. As with protocols such as TLS that play a fundamental role in providing security to applications in the current Internet communications infrastructure, end-to-end security at the transport-layer may provide an important contribution to the achievement of appropriate security in the context of Internet-integrated WSN.

The constraints in terms of fundamental resources such as memory, microprocessor and energy in practice determine the employment of low-energy wireless communications in WSN environments, providing low communication speeds and small packets with the goal of minimizing communication errors. The integration of LoWPAN environments, such as WSN, with the Internet brings new challenges into the design of communication and security mechanisms able to support end-to-end communications between devices that may be very different in their available resources.

Although numerous proposals exist to address security in closed LoWPAN environments [201], the integration of sensor networks with the Internet will raise challenges yet to be faced by research. As previously analyzed in the context of our SoA analysis, the current security technology adopted to protect transport-layer communications on 6LoWPAN environments is the DTLS [127] protocol, which provides confidentiality, integrity and authentication to CoAP application-layer messages. While the overhead introduced by DTLS on 6LoWPAN communications is certainly non-negligible, encryption and decryption may be facilitated by the employment of AES/CCM in IEEE 802.15.4 sensing platforms, as previously observed for network-layer security as proposed for 6LoWPAN. On the other hand, the applicability of DTLS will be mostly dependent on the viability of supporting the security modes currently proposed for CoAP security [34] that depend on Elliptic Curve Cryptography (ECC) for authentication purposes. In this context, the impact of the initial DTLS handshake providing authentication and key agreement must be evaluated.

In the present chapter we describe our research proposal to address transport-layer security in the context of Internet-integrated WSN. The proposed solution is implemented considering the reference integration architecture discussed in Chapter 3 and experimentally evaluated against its impact on the limited resources available in constrained WSN environments, considering the methodology discussed in the same chapter. Our research proposal supports mechanisms designed to contribute to the effectiveness of end-to-end transport-layer security and also to the protection of low-energy wireless communication environments against Internet-originated threats.

Ours is the first proposal focused on the previous goals, and our initial analysis on the problem of effectively supporting end-to-end security with Internet-integrated sensing applications is discussed in [202], where we compare the security mechanisms currently proposed for the CoAP [34] protocol against 6LoWPAN network-layer security as proposed in the previous chapter of the thesis, considering the algorithms defined as mandatory for the IP Security Architecture. In [203] we also present an extended experimental evaluation of the feasibility of security for application-layer communications in the context of Internet-integrated WSN. This evaluation study enables us to observe that ECC (currently adopted for CoAP) causes a major impact on the resources of constrained wireless sensing devices. Overall, these experimental evaluation works motivated the design of the research solution described later in the present chapter, which are also available in [204].

In our next discussion, we start by analyzing other proposals targeting security at the transport-layer for Internet-integrated WSN. This discussion complements our previous analysis on transport-layer security for Internet-integrated WSN in Chapter 2. Next we analyze the cost of transport-layer security using DTLS as currently adopted for CoAP, and discuss why the results of this evaluation motivate the research solutions proposed later in the chapter. As in the previous chapter, the proposed research solutions are experimentally evaluated and employed considering our discussion in Chapter 3.

5.2 ALTERNATIVE APPROACHES TO TRANSPORT-LAYER SECURITY

The following discussion complements the analysis previously performed in Chapter 2, where security in WSN environments was discussed. Although new mechanisms will be required to support security with end-to-end communications using recently standardized technologies such as 6LoWPAN and CoAP, particularly considering that such communications may take place in the context of Internet-integrated sensing applications, most of the previous approaches to security consisted in the protection of communications at link-layer for closed LoWPAN environments [3]. In such proposals sensing devices may communicate securely using individual, group or network-wide symmetric encryption keys. For example, MiniSec [205] falls on this category and supports encryption and authentication for unicast and broadcast communications at the link-layer.

As previously discussed, research proposals such as Sizzle [190] and SSNAIL provided initial approaches to end-to-end security, although with limitations which makes them incompatible with end-to-end communications for Internet-integrated WSN environments. Sizzle requires a reliable transport-layer protocol and is therefore incompatible with CoAP and 6LoWPAN, while also impacting largely on the performance of low-energy communications. Sizzle also does not support two-way authentication, thus being inappropriate to support future M2M applications on the IoT requiring mutual authentication. SSNAIL [189] supports two-way authentication using an ECC-enabled handshake, but also requires a reliable protocol at the transport-layer.

As analyzed in Chapter 2, various research proposals target the problem of end-to-end transport-layer security by modifying or optimizing DTLS to cope with the constraints of current sensing platforms. Such proposals involve either the compression of the DTLS headers [143], the introduction of mechanisms to support mapping between TLS and DTLS [140], for the support of inactive devices [144] and certificate pre-validation [147], or for the transportation of DTLS handshake messages using CoAP [148]. None of the existing proposals addresses the problem of the computational and energetic impact of the authentication and key agreement phase, while supporting two-way authentication, being completely compatible with CoAP security as currently proposed for application-layer communications and completely transparent to the end-to-end communicating entities. The proposal in [145] supports two-way authentication using RSA and trusted-platform modules (TPM) with secure storage for the private keys. This proposal doesn't support ECC public-key authentication as currently defined for CoAP [34], neither sensing devices without specialized modules to support security.

Other aspect we may note on the alternative approaches previously discussed is that such proposals do not address the support of transport-layer security in tandem with other security mechanisms designed to protect constrained sensing devices and low-energy communications from external or Internet-originated threats. We may envision this to be an important enabling factor of many sensing applications that will require the usage of constrained LoWPAN devices exposed to Internet communications. As considered with our reference model for end-to-end security, such mechanisms may be based or at least benefit from the presence of WSN security gateways, which support routing and mapping mechanisms for communications between the WSN and Internet domains.

The design of end-to-end security at the transport-layer in Internet-integrated sensing applications provides the opportunity to address the previously identified limitations, as we consider in the research solutions discussed throughout the present chapter. The proposed research solutions are developed with this goal, in the context of the reference model described in Chapter 3. This model also provides the ground for the support of transport-layer security in three complementary usage modes, as we discuss later in the chapter. Our research proposal for transport-layer security was motivated by the results of an experimental evaluation study on the impact of CoAP security as currently proposed for this protocol [34], which we discuss next.

5.3 EXPERIMENTAL EVALUATION OF THE FEASIBILITY OF COAP SECURITY

The current CoAP proposal [34] enables RESTful web communications on 6LoWPAN environments and defines bindings for the usage of DTLS at the transport layer. In our following discussion we start by analyzing how security is currently addressed to protect CoAP communications, and next what are the main limitations of the current approach according to our experimental evaluation of CoAP security. The experimental evaluation

study described next is published as the first research contribution evaluating the impact and effectiveness of transport-layer security as proposed for CoAP [203]. In [202] we also compare CoAP security against 6LoWPAN network-layer as previously proposed.

5.3.1 CoAP SECURITY MODES

As previously discussed, payload space is a scarce resource in IEEE 802.15.4 environments, and consequently header and address compression is prevalent in 6LoWPAN and CoAP specifications. In Figure 5.1 we illustrate the employment and availability of payload space in IEEE 802.15.4 low-energy communication environments using 6LoWPAN and CoAP, when supporting end-to-end communications with Internet hosts. In particular, 6LoWPAN IPHC shared-context header compression [71] enables the compression of the UDP/IPv6 header down to 10 bytes, while CoAP requires 4 bytes and DTLS a total of 13 bytes, not considering the space required for the transportation of security-related data as an Initialization Vector (IV) or authentication (HMAC) fields.

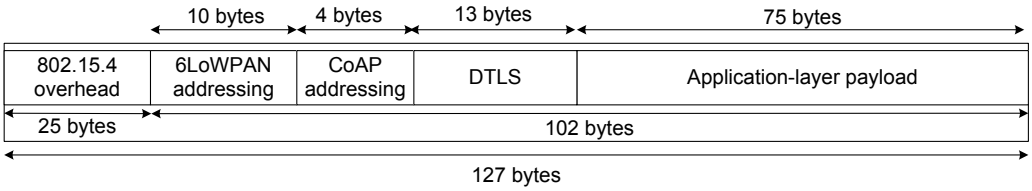

Figure 5.1 - Payload space usage for end-to-end communications in 6LoWPAN environments

Sensing platforms as the TelosB [194] implement IEEE 802.15.4 and support hardware AES/CCM encryption at the link layer. While end-to-end security may dispense link-layer security, this doesn't preclude the usage of hardware-based encryption to support security at higher layers, as we have previously observed for 6LoWPAN security and consider also for other research proposals described in the thesis. While DTLS provides confidentiality, authentication and integrity, the authentication and key agreement between communication parties may take place following different approaches.

Regarding CoAP security, two of the proposed security modes currently require the support of ECC public-key authentication [34], namely the *RawPublicKey* and *Certificates* CoAP security modes. Table 5.1 resumes the characteristics of the security modes proposed for CoAP, which we have previously identified in Chapter 3 and analyze in greater detail in our following discussion.

As illustrated in this table, the *PreSharedKey*, *RawPublicKey* and *Certificates* modes support security with different configurations. In particular, Elliptic Curve Cryptography (ECC) operations are employed in the *RawPublicKey* and *Certificates* security modes using the Elliptic Curve Digital Signature Algorithm (ECDSA) [133], and key agreement using the Elliptic Curve Diffie-Hellman with Ephemeral Keying Algorithm (ECDHE) [133]. Encryption employs

AES in CCM (at the hardware when available) or CBC modes. After authentication, both parties share a pre-master shared secret from which they derive a shared master secret, and from this master secret they obtain the keying material required for encryption and authentication [127].

Table 5.1 – Security modes defined for CoAP communications

Security mode	Security usage
NoSec	• *Encryption*: None; • *Authentication*: None;
PreSharedKey	• *Encryption*: Using DTLS; • *Authentication*: None; • *Keys obtained:* Predefined keys (key for a node, key for a group of nodes); • *Mandatory cypher suites:* TLS_PSK_WITH_AES_128_CCM_8
RawPublicKey	• *Encryption*: Using DTLS; • *Authentication*: Mutual using public keys; • *Keys obtained:* Device has one or various public keys; Device identification(s) derived from public key(s); Device stores identification of authorized nodes; • *Mandatory cypher suites:* TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
Certificates	• *Encryption*: Using DTLS; • *Authentication*: Mutual using X.509 certificates; • *Keys obtained:* Device has one or more certificates binding public keys to authority names of the device; Public key and device id of other nodes obtained from certificates; Device has root trust anchors for certificate validation; • *Mandatory cypher suites:* TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 TLS_RSA_PSK_WITH_AES_128_CBC_SHA

In the *PreSharedKey* security mode a device stores predefined keys used to communicate securely with another sensing device or with a group of sensing devices. This may be useful in situations where public-key cryptography requires too many resources, or when the management of pre-shared keying is convenient, as in closed environments where pre-configuration of devices is necessary due to other reasons. This mode uses the TLS_PSK_WITH_AES_128_CCM suite, which uses the Authenticated Encryption with Associated Data (AEAD) [206] operational mode AEAD_AES_128_CCM [134]. AES is used in the Counter and CBC-MAC (CCM) mode, providing confidentiality and data origin

authentication. The usage of an AEAD mode is a requirement from DTLS, and it adds the ability to verify the integrity and authenticity of data other then that which is encrypted. A 128-bit authentication tag is used with the CCM mode, and a unique 12-byte nonce is used for each packet using the same key. Integrity is supported using the Pseudorandom Function (PRF) defined for TLS 1.2 [134] with HMAC with SHA-256.

In the *RawPublicKey* mode a sensing device accesses one or more public keys, from where it may extract its identification. ECC public-key cryptography is employed to perform authentication of peer devices (DTLS client and server) and the device must also store the identity of the nodes it communicates with, given that a certification chain is not available. The cipher suite proposed for this mode is TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, which employs Elliptic Curve Diffie-Hellman with Ephemeral Keying (ECDHE) and ECDSA ECC cryptographic algorithms. As with the *PreSharedKey* mode, this cipher suite uses the AEAD_AES_128_CCM operational mode. Ephemeral keys used with ECDHE enable perfect forward secrecy. During the authentication process the server generates and sends an ephemeral ECDH ECDSA-capable public key to the client, together with the indication of the corresponding ECC curve to be used for ECC encryption. The client then generates an ECDH key pair, sends its public component to server in a message signed with ECDSA. The authentication process allows for the client and the server to agree on a shared premaster secret for DTLS.

Finally, the *Certificates* security mode adds to the operational modes of the previous mode the usage of certificates for authentication purposes. A Certification Authority (CA) should be available so that a device is able to use root trust anchors for certificate validation purposes. For compatibility with devices not supporting ECC encryption, authentication with RSA is available to perform authentication and pre-shared key agreement. The RSA_PSK [207] key exchange algorithm is used in this fall back usage scenario, employing the TLS_RSA_PSK_WITH_AES_128_CBC_SHA security suite. Confidentiality if guaranteed using AES in CBC mode and SHA provides integrity.

In terms of security and also of the availability of critical resources, a chain is only as strong as the weakest link. CoAP encryption and authentication using DTLS may be efficiently supported by AES/CCM at the hardware in any of the previously described security modes, but authentication and key agreement may provide the largest impact on the limited resources of low-energy devices and communications. Authentication and key agreement are performed in the initial DTLS handshake, which end-to-end devices are required to support for DTLS to be a viable solution in supporting end-to-end communications at the transport layer in WSN environments. This aspect motivated our preliminary experimental evaluation of the impact of CoAP security as currently proposed for 6LoWPAN environments [203], which we discuss next.

5.3.2 IDENTIFICATION OF CRYPTOGRAPHIC ALGORITHMS FOR COAP SECURITY

Our preliminary evaluation study was performed with the goal of evaluating experimentally the usage of security for communications using CoAP, as a fundamental requirement for the successful integration of smart objects with the Internet at the application layer. The main goal of this evaluation is to investigate the impact of CoAP security as currently proposed [34], which depends on DTLS and ECC-based authentication and key agreement, as previously discussed. For this evaluation, we again consider the TelosB [194] reference sensing platform.

Our experimental evaluation study employs 6LoWPAN and CoAP communication sessions established between different 6LoWPAN devices, in particular between a TelosB sensing device and a Linux host supporting 6LoWPAN, CoAP and IPv6. The TelosB mote supports the latest version of the TinyOS [56] operating system with 6LoWPAN and CoAP, together with different configurations in terms of the security algorithms employed and mechanisms supported.

The Linux host is a router between an Ethernet IPv6 network and the IEEE 802.15.4 LoWPAN, employing a second TelosB mote as a bridge supporting communications with the network of smart objects. The TelosB mote used for measurement of the experimental parameters is a battery-powered device supporting our TinyOS testing application and the 6LoWPAN/CoAP security enabled stack. As in our previous evaluation of 6LoWPAN security, we consider the TelosB mote to provide a good reference for the validation of new mechanisms for constrained sensing platforms.

Table 5.2 - Cryptographic algorithms and suites for CoAP security

	TLS_PSK_ WITH_AES_128_CCM_8	TLS_ECDHE_ECDSA_ WITH_AES_128_CCM_8	TLS_RSA_PSK_ WITH_AES_128_CBC_SHA
Encryption	AES/CCM	AES/CCM	AES/CBC
Authentication	PSK key exchange	ECDHE and ECDSA	RSA
Integrity	TLS 1.2 PRF (SHA-256)	TLS 1.2 PRF (SHA-256)	TLS 1.2 PRF (SHA-1)
Security mode(s)	PreSharedKey	RawPublicKey or Certificates	Certificates

The adoption of ECC cryptography in supporting security in LoWPAN communication environments is motivated by its ability of supporting similar security than classic public-key cryptography with significantly smaller key sizes. Smaller key sizes and more efficient computations result in savings of critical resources on constrained sensing devices. In Table 5.2 we describe the cryptographic algorithms required for the previously discussed CoAP security modes. Our implementation considers that, for the configurations described in Table 5.2, AES/CCM encryption is always performed at the hardware on the TelosB, which provides AES in the CCM* combined mode with the cc2420 chip. This mode uses 128-bit keys to process 128-bit blocks of data and supports integrity, authentication and

confidentiality as required for cipher suites employing AES. We also observe that all security modes use SHA-256 for integrity verification operations. AES is also used in the CBC mode, which in our implementation we support in software.

It is important to note that the individual evaluation of the usage of each of the algorithms identified in Table 5.2 would in practice not provide a clear picture of the viability security for CoAP. Our goal is therefore to evaluate each CoAP security mode globally, by considering all the necessary cryptographic algorithms and related operations. For this purpose, we employ software implementations of the various suites optimized for small microcontrollers with the characteristics of the MSP 430 available on the TelosB, side-by-side with AES/CCM hardware-based encryption. The fact that software and hardware-based cryptographic algorithms are employed simultaneously contributes to the conclusions of our experimental evaluation study, as it traduces the heterogeneity of the characteristics typically available on constrained sensing platforms.

5.3.3 OVERHEAD ON NETWORK-LAYER PAYLOAD SPACE

As LoWPANs have very low throughputs and may present significant packet error rates, packets are small and designed to transport only limited amount of data. This implies that payload space is a scarce resource in 6LoWPAN environments, and consequently one that we must evaluate the impact of CoAP security against. Other important evaluation aspects are related to the constraints of sensing devices, particularly ROM and RAM memory and energy (required to compute and transmit security-related information). The evaluation of such resources allows an overall analysis on the impact of CoAP security, considering the physical characteristics of real constrained sensing devices and requirements from particular applications.

The payload space available to applications may greatly influence the usefulness of protocols such as CoAP. Although the main goal of CoAP is to guarantee a low message overhead, the protocol [34] itself only provides an upper bound of the message size, particularly that a CoAP message, appropriately encapsulated, should fit within a single IP packet. Nevertheless, in constrained environments based on the 6LoWPAN adaptation layer, only 102 bytes are available to upper layer protocols and applications in the best case. This does not preclude the usage of larger IPv6 messages but implies costly fragmentation operations at the adaptation layer that should be avoided. Given such aspects, we consider that applications for the WoT must in practice be frugal in payload space usage and our study concentrates on analysing how much space CoAP security modes leaves to applications without requiring fragmentation from the 6LoWPAN adaptation layer. Our analysis also considers the absence of mesh and fragmentation 6LoWPAN headers, as is appropriate to support secure end-to-end communications with smart objects.

For the analysis of the impact of CoAP security on payload space we must also consider the space required for 6LoWPAN addressing at the adaptation layer. As we are interested in

end-to-end communications with sensing devices, we consider a 6LoWPAN address compression scenario where compression is only able to elide the IID (Interface Identifier) of the source device, with the remaining part of the source address and the full destination IPv6 address being carried inline, requiring a total of 31 bytes for 6LoWPAN addressing. Figure 5.2 illustrates the payload space available using the three security modes proposed for CoAP, in percentage of the 102 bytes of a fully sized 6LoWPAN packet.

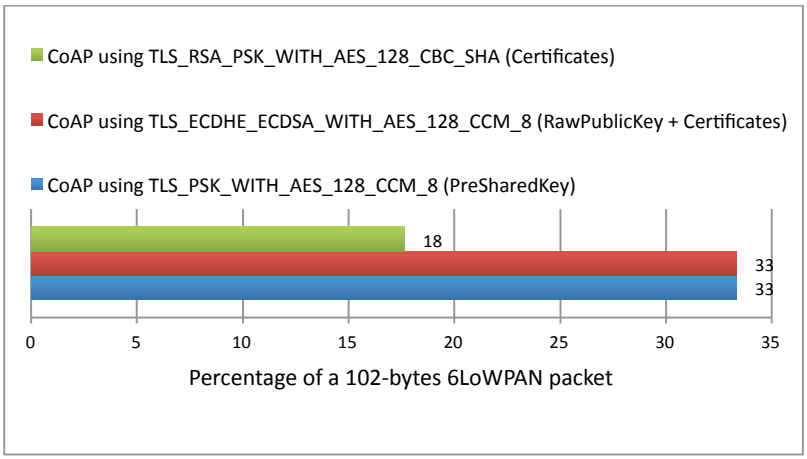

Figure 5.2 - Payload space available to applications using CoAP security

For the values illustrated in Figure 5.2 we consider the necessity of transporting a fixed CoAP header occupying 4 bytes [34] plus a DTLS header requiring 13 bytes [131]. Information as URI (Universal Resource Identifiers) or confirmation messages is transported in CoAP binary options or on the payload after the fixed header, and therefore is considered to be part of the payload in our evaluation. This analysis also considers the transportation of security-related information, namely of a MIC (authentication) code and a nonce value used for symmetric encryption. The MIC code requires 8 bytes for security modes based on the AEAD_AES_128_CCM_8 AEAD mode. For such modes the nonce value requires 12 bytes. When using TLS_RSA_PSK_WITH_AES_128_CBC_SHA, a 16-byte nonce value required for AES_128_CBC and a 160-bit authentication code resulting from the usage of SHA-1 are transported.

As Figure 5.2 illustrates, the usage of CoAP security for end-to-end communications with Internet hosts significantly impacts on the payload space available to applications. The worst scenario is with TLS_RSA_PSK_WITH_AES_128_CBC_SHA, which frees only 18% of the total payload space to applications. The two other security modes are best in this respect, leaving 33% of the total payload space available. The impact is the same because both security suites are based on the AEAD_AES_128_CCM_8 mode. Although impacting significantly on payload space, CoAP security can be viable as long as applications are frugal in respect to payload space requirements, if one wants to avoid fragmentation at the 6LoWPAN adaptation layer. Applications requiring larger application-layer payloads will be unable to

avoid fragmentation or in alternative employ other security mechanisms, for example network-layer security as previously proposed.

5.3.4 OVERHEAD ON MEMORY

As the memory available on constrained sensing devices is a scarce resource, our next evaluation is on the memory (RAM and ROM) necessary with each version of a testing application using TinyOS with 6LoWPAN and CoAP, together with the software required for each security configuration. In Figure 5.3 we illustrate the values of memory required for each CoAP security mode, in percentage of the total of RAM and ROM memory available on the TelosB sensing platform. The TelosB supports 48 Kbytes of ROM for program storage and 10 Kbytes of RAM for program execution. For comparison purposes, in Figure 5.3 we also illustrate the memory required for a TinyOS application with BLIP and CoAP support without security, which therefore provides a basis reference in terms of memory requirements.

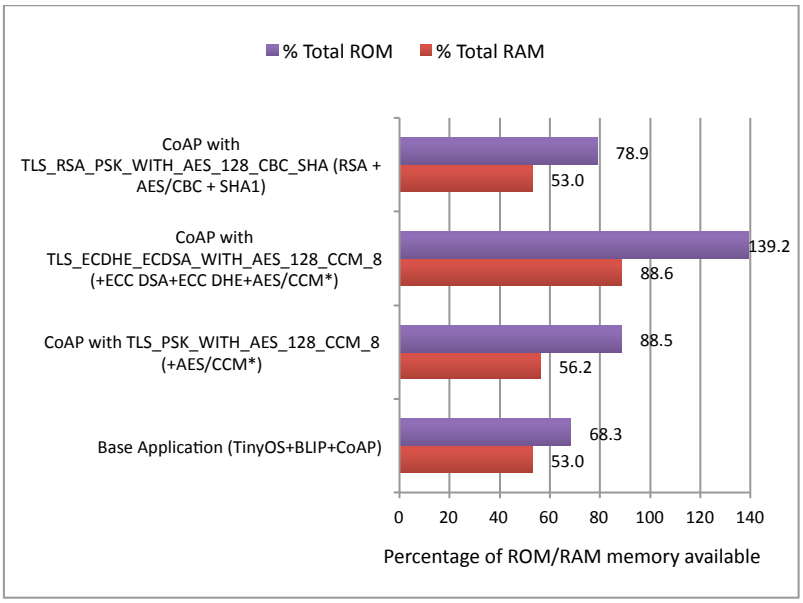

Figure 5.3 – Memory footprint of CoAP security

Although AES_128_CCM_8-based modes make use of AES/CCM hardware encryption with the cc2420 chip available on the TelosB, hardware encryption in the standalone mode still requires software support [200]. In contrast to the inline mode, standalone encryption allows applications to perform hardware encryption and decryption without requiring the transmission or reception of a packet by the link-layer. ECC cryptographic operations are supported using TinyECC [208]. Remaining algorithms such as RSA, AES/CBC and SHA-1 are evaluated using code optimized for 8-bit architectures.

As we are able to observe in Figure 5.3, hardware-level encryption doesn't come without a non-negligible overhead on memory, particularly in terms of ROM memory. This is clearly

visible on the impact of the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 and TLS_PSK_WITH_AES_128_CCM_8 security suites. We are also able to identify one major limitation of the TelosB, in particular that not enough ROM memory is available in this platform to support all cryptographic operations required with the usage of TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8. This mode requires the simultaneously support of ECCDH, ECCDSA and AES/CCM*. RAM memory is also potentially a problem in this case, since 88.6% of memory usage during processing of security for CoAP packets may prove to be a problem in many usage scenarios where other applications are required to run on the sensing device. Sensing platforms with more memory available than the TelosB, both for storage and running security and applications, will be required in the future to effectively use ECC-based security with CoAP. The other security modes are valid in respect to their requirements on the memory available on the sensing device.

5.3.5 COMPUTATIONAL AND ENERGY OVERHEAD OF COAP SECURITY

As advanced mechanisms such as multi-threading are usually not supported in low-end microcontrollers such as the MSP430 of the TelosB, the computational time required to process security for a CoAP packet directly influences the maximum communications rate that a smart object can expect to achieve for a given sensing application. Energy is also a very scarce resource on sensing platforms, and as such many sensing applications are designed with battery-powered sensing devices in mind. Our evaluation therefore measures the computational and energy impact of CoAP security, as they represent two important evaluation criteria of the feasibility of any communications or security proposal for smart objects. Table 5.3 describes the experimentally obtained values for these two resources using CoAP with the proposed CoAP security modes. As for the previous evaluation, we consider the application of security to a fully sized 102-byte 6LoWPAN packet.

Table 5.3 - Computational and energy impact of CoAP security

Cipher suite	Processing overhead (ms)	Energy overhead (mJ)
TLS_PSK_WITH_AES_128_CCM_8	3.6	0.0002
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	2019.6	10.89
TLS_RSA_PSK_WITH_AES_128_CBC_SHA	21.9	0.0019

We can clearly observe that ECC public-key cryptography represents a much larger impact in terms of both the computational time and the energy required from the sensing device, certainly two interrelated aspects. This is mostly due to the fact that the TLS_ECDHE_ECDSA_WITH_AES_CCM_8 security suite requires processing of each CoAP packet with both AES/CCM* and ECDSA. In addition, ECDH is used for key establishment during the establishment phase of a DTLS session. Although ECC-based cryptography is an interesting alternative to classical public cryptography, we are able to verify that it still

represents a non-neglectable impact on the performance and energy of current sensing platforms. This inevitably influences the lifetime of sensing applications or its maximum achievable transmission rate. This also points to the fact that sensing platforms can evolve to support efficient hardware-based operations to aid in the processing of ECC cryptography, similarly to what is nowadays possible using AES/CCM in the standalone mode with sensing platforms implementing IEEE 802.15.4.

Based on the results our experimental evaluation illustrated in Table 5.3, we may also observe that the alternative modes TLS_PSK_WITH_AES_128_CCM_8 and TLS_RSA_PSK_WITH_AES_128_CBC_SHA are much more efficient, since the former only requires AES/CCM*, while the later requires AES/CBC plus SHA-1 and RSA. Given the availability of AES/CCM at the hardware with the TelosB, TLS_PSK_WITH_AES_128_CCM_8 is clearly the most efficient security mode, therefore an excellent choice when pre-deployment and configuration of security-related parameters on sensing devices is desired. If public-key cryptography is necessary, we verify that TLS_RSA_PSK_WITH_AES_128_SHA appears as an acceptable alternative to ECC using the TelosB. Public-key cryptography will be required in deployment scenarios where CoAP security must coexist with existing certification infrastructures, either at the Internet or at backend networks.

5.3.6 IMPACT OF COAP SECURITY ON THE COMMUNICATION RATE OF SENSING DEVICES

The previously described experimental evaluation study enabled the identification of the impact of CoAP security on constrained resources of sensing devices, namely the memory, energy, and computational time required for the processing of CoAP security. These results are next used to perform an overall analysis of the impact of CoAP security on sensing applications. This analysis also sustains our conclusions on the high impact of ECC-based security as employed with DTLS to protect CoAP messages.

As the WoT will enable sensing applications with very diverse requirements in terms of communications, it is important to analyse if the usage of CoAP security may represent a bottleneck in this respect. Security operations may introduce a non-negligible computational overhead on constrained smart objects, meaning that such devices are unable to process packets received or waiting transmission while the microcontroller is busy performing cryptographic operations.

When considering communications using IEEE 802.15.4 at 250Kbit/s, we realize that the impact of the computational time required for security on the maximum transmission rate is much larger than the impact on the time required for the transmission of a few more bytes required for the 6LoWPAN or CoAP addressing and auxiliary security data. What we cannot exclude from consideration is the overhead introduced by IEEE 802.15.4 on the bandwidth available for 6LoWPAN and application data. As previously discussed, this overhead represents 19.6% of the total bandwidth, as 25 bytes are required for link-layer information with each 127 bytes 6LoWPAN packet.

In Table 5.4 we identify the maximum transmission rate achievable by an application using the previously described CoAP security modes, in packets per second. The values in this table are obtained also considering the time required for the processing of 6LoPWAN, CoAP and DTLS headers on the TelosB, which we have experimentally measured as 0.09 milliseconds. We do not represent the maximum transmission rate achievable using CoAP without security, which was determined as 246 packets per second when processing 102-byte packets. We again consider the transmission of fully sized 102-byte 6LoWPAN packets. While CoAP applications may require smaller packets, in practice we must also consider the transportation of 6LoWPAN, CoAP and DTLS headers, together with data required for security operations. 6LoWPAN requires 31 bytes when communications occur between sensing devices and Internet hosts. CoAP uses a 4-byte fixed header and DTLS requires 13 bytes for header information. These values are therefore valid for applications requiring the transmission of at most 54 bytes.

Table 5.4 – Maximum transmission rates with CoAP security

Cipher suite	Maximum transmission rate (packets/sec)
TLS_PSK_WITH_AES_128_CCM_8	132.12
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	0.494
TLS_RSA_PSK_WITH_AES_128_CBC_SHA	38.65

We can observe that ECC based security will only be viable for sensing applications requiring very low transmission rates. Nevertheless, as many applications will probably be designed to save energy by not requiring large transmission rates, ECC may still be viable in many deployment scenarios. This may be particularly true if we consider the advantages of ECC, namely its superior security, the savings provided in terms of memory (particularly in comparison with RSA) and the its easy integration with public-key management and certification infrastructures, important factors for the secure integration of CoAP communications with the Internet.

If on the other hand pre-configuration of devices is desired and can also be applied to predefine security information for each device prior to its usage in a given sensing application, AES/CCM provides the best choice and enables security for applications requiring higher transmission rates with the TLS_PSK_WITH_AES_128_CCM_8 security mode. Security based on RSA, AES/CBC and SHA appears as a good alternative, and can be used to enable security for applications requiring moderate transmission rates. In general, we observe that as long as the appropriate security mode is selected accordingly to the requirements of each sensing application, CoAP security is viable in respect to its overhead on communications.

5.3.7 IMPACT OF COAP SECURITY ON THE EXPECTED LIFETIME OF SENSING DEVICES

Our next evaluation is on the impact of CoAP security on the lifetime of sensing devices, as it in the end may determine the lifetime of a particular sensing application. Most sensing applications designed for the WoT will probably be viable only if able to operate in unattended mode during an acceptable period of time. In many deployments smart objects will require the usage of batteries, and its replacement may be difficult or even impossible for long periods of time. In Figures 5.4 and 5.5 we illustrate the impact of CoAP security on the lifetime of sensing applications. These values were obtained from our experimental evaluation study using a TelosB and different testing applications using TinyOS with 6LoWPAN, CoAP and security.

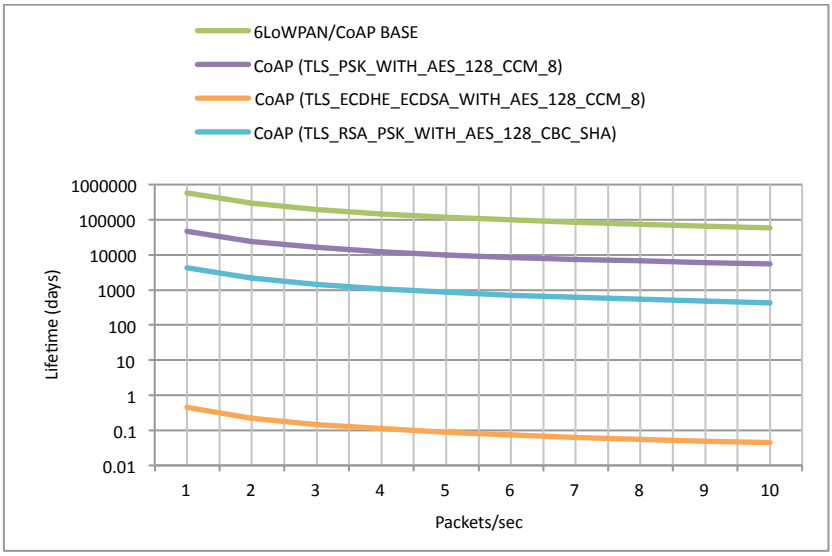

Figure 5.4 - Lifetime of sensing applications with CoAP security (higher communication rates)

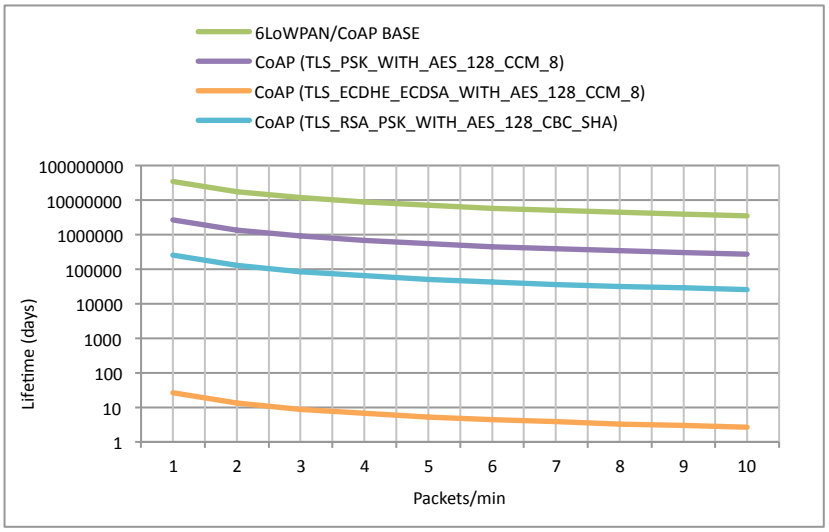

Figure 5.5 - Lifetime of sensing applications with CoAP security (lower communication rates)

In Figure 5.4 we illustrate the lifetime of sensing applications (in days) for higher transmission rates (from 1 to 10 packets transmitted per second). This may correspond to applications requiring frequent transmissions of sensing data, for example of high-priority information from sensors in an industrial environment being sent to a server in a control room. In Figure 5.5 we illustrate an alternative usage scenario where lower transmission rates are required (from 1 to 10 packets transmitted per minute). This could correspond for example to a home surveillance application requiring periodic readings from security sensors to be sent to a central (backend) server.

Due to the wide range of values, in Figures 5.4 and 5.5 we employ a logarithmic scale. For comparison purposes, in both figures we also illustrate the lifetime expectancy when using communications without security. We may again observe the impact of ECC based security on the lifetime of sensing applications. The TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 security mode applied to CoAP can be considered enviable for higher communication rates. The protection of CoAP communications using ECC will only be viable for applications requiring very low transmission rates or on the other hand for applications intended for short-term deployments.

Although more acceptable, the remaining security modes also cause a noticeable impact of the lifetime of sensing applications. The expected lifetime using TLS_PSK_WITH_AES_128_CCM_8 is 7.7% of the expected lifetime for the same application using CoAP communications without security, while this value is of only 0.7% for TLS_RSA_PSK_WITH_AES_128_CBC_SHA. Albeit the efficiency of AES/CCM at the hardware level, this cryptographic algorithm does not prevent a large impact on the expected lifetime of a sensing device. Nevertheless, we again observe that the impact of security does not prevent acceptable compromises, as it should not be limitative of the applicability of CoAP security, as long as applications for the WoT are designed to depend on moderate or low transmission rates and small data payloads.

In conclusion, our experimental evaluation of the impact of CoAP security as currently proposed enables us to identify its effectiveness in various usage scenarios employing sensing devices with the characteristics of the TelosB, but also the large impact of the security modes requiring the employment of ECC security for authentication and key agreement purposes. Our evaluation study allowed the identification of relevant limitations of current sensing platforms that should deserve attention in future designs. Particularly relevant is the lack of ROM and RAM space to accommodate all the require security mechanisms and algorithms for CoAP security, and the lack of support for hardware-based efficient ECC cryptography. The lack of efficient support for ECC-based security is particularly important for the support of CoAP security in the context of Internet-integrated WSN, since it will require or at least may benefit from compatibility with public-key certification infrastructures.

Our conclusions from the previous experimental evaluation study are also supported by our comparison between 6LoWPAN security (as proposed in Chapter 4) and CoAP security as previously evaluated, with the goal of protecting end-to-end communications according to predefined application security usage profiles [202]. In this experimental comparison study the most appropriate security modes at the network and application layers are identified for a set of representative applications, characterized according to its security requirements in terms of confidentiality, authentication and integrity, as well as the support of web services and public-key infrastructures.

Taking into consideration the previously identified limitations of transport-layer security, our research proposals constitute a solution for the efficient support of end-to-end transport-layer security in the context of Internet-integrated WSN. As discussed next, our approach is completely transparent from the point of view of the communicating entities, and consequently no modifications are required for the support of transport-later security using DTLS on such devices. We also consider the support of mutual public-key authentication and of protection against attacks at the transport-layer, as well as the support of transparent mobility from the perspective of transport-layer security.

5.4 A PROPOSAL FOR END-TO-END TRANSPORT-LAYER SECURITY WITH MUTUAL AND DELEGATED PUBLIC-KEY AUTHENTICATION

In the context of the reference model described in Chapter 3, we consider that transport-layer end-to-end security may be achieved with three complementary usage modes. One of such security usage modes consists in the full delegation of end-to-end security to the 6LBR, which may be appropriate to devices unable to support neither authentication nor the normal application of security to transport-layer as required for DTLS. The second usage mode may consist in the employment of DTLS in a truly end-to-end fashion, with sensing devices being required to support all the required security-related procedures. Finally, DTLS encryption may be supported in an end-to-end fashion, with the DTLS authentication and key agreement being supported by the gateway through delegation, as we consider in the research solutions presented later in the chapter.

The main motivation for our approach is that delegation presents a solution to the problem of having to support ECC cryptography on constrained sensing devices, which may impact greatly on its limited resources. Other important aspect is that DTLS security, after the handshake, may be efficiently supported using AES/CCM available efficiently at the hardware in IEEE 802.15.4 sensing platforms. AES/CCM hardware encryption thus appears again as an important cross-layer mechanism for the enabling of end-to-end security mechanisms at higher layers of the stack.

As previously discussed, CoAP security [34] considers the usage of ECC cryptography, and as such ECC public-key authentication and key negotiation in the context of DTLS is a requirement. We must also note that sensing platforms may not be ready to viably support

ECC at this stage, as we have discussed. A related while also important limitation is that it may be costly to store and interpret certificates and ECC public-keys in constrained sensing devices with very limited amounts of RAM and ROM memory, as is the case of the TelosB reference sensing platform.

Other goal we may address is to leverage security by designing and supporting mechanisms, to be employed side-by-side with end-to-end transport-layer security. For example, mechanisms may be required to support control of accesses to resources available on CoAP constrained sensing devices. Related mechanisms may also be necessary supporting operations such as authentication and trust management between devices on the LoWPAN. We may thus consider that the employment of such mechanisms in parallel with transport-layer security may provide an opportunity to promote security as an enabling factor of Internet-integrated sensing applications.

Although CoAP adopts ECC cryptography in supporting authentication and key negotiation, ECC still represents a non-negligible impact on current sensing platforms, as we have previously discussed. This limitation is also expressed in the adoption of RSA in alternative research proposals such as [145]. Even though sensing devices may be expected to evolve to support more memory space and increased computational capability in the future, the integration of sensor networks with the Internet must be supported in the near future by mechanisms designed in a realistic fashion, accordingly to the limitations and characteristics of current sensing platforms. Such aspects motivate our research approach to the design of mechanisms that may intervene in the effective support of DTLS for communications in the context of Internet-integrated WSN.

Considering the impact of the initial DTLS authentication and key agreement handshake, one major goal of our research efforts is to target alternative approaches for the support of ECC-based public-key authentication and key agreement using “off-the-shelf” sensing platforms, as mechanisms found to be viable for such platforms may be appropriate to a wide range of sensing platforms likely to support future IoT applications. Of particular importance is the overhead of the DTLS handshake and the security of Internet-integrated LoWPAN from Internet-originated threats, two issues that are not addressed in the current 6LoWPAN and CoAP specifications.

Regarding the overhead of the DTLS authentication and key agreement handshake, we verify that, other that the payload space required for the DTLS header (around 11% of the available space using 6LoWPAN and CoAP), end-to-end authentication using ECC public-key cryptography requires the exchange of various large messages and certificates. Large handshake messages such as those transporting certificates may require fragmentation at the 6LoWPAN adaptation layer. In fact, the most computationally expensive part of a DTLS session is the handshake and it requires more effort from the server than from the client. It is also important to note that many sensing applications are likely to require that sensing devices support CoAP servers. Adding to the time required exchanging handshake messages

in low-energy wireless networks at low speeds, sensing devices are required to support ECC public-key authentication and key negotiation. The memory required to store ECC certificates and public-keys might also be a problem, depending on the sensing device and on the application at hand.

As previously discussed, other relevant aspect is the protection of end-to-end communications and of WSN sensing devices against external or Internet-originated threats, and in this respect the WSN security gateway in the reference integration architecture described in Chapter 3 serves as a strategic point for the enabling of appropriate security mechanisms. Regarding DTLS, we verify that it supports limited protection against Denial of Service (DoS) attacks by requiring that a connecting client answer a challenge from the server with a particular stateless cookie. Although this is a desirable mechanism, it may also impact on the resources available on constrained sensing devices. It is also fair to consider that a plethora of similar threats are likely to appear from the minute we start integrating LoWPANs with the Internet.

Another aspect motivating our approach to transport-layer end-to-end security is that mainstream sensing platforms such as the TelosB [194] currently are unable of efficiently supporting ECC encryption, as our previous experimental evaluation study has demonstrated. Regarding the support of DTLS, this implies that the energy and the computational time required supporting ECC public-key authentication and key agreement in the context of the initial handshake may undesirably impact on the lifetime of sensing applications or on its maximum achievable communications rate, two enabling aspects in the context of the framework illustrated in Figure 3.3. Despite such limitations, the support of ECC cryptography in a fashion compatible with the current CoAP proposal is fundamental, and as such transparency of new security mechanisms from the point of view of the communicating entities is a desired property.

5.4.1 DELEGATED MUTUAL AUTHENTICATION AND KEY NEGOTIATION

The system model illustrated in Figure 5.6 materializes the reference integration model previously discussed in Chapter 3 and illustrated in Figure 3.1. For the purpose of supporting end-to-end communications at the transport layer, messages in the context of such communications are transparently intercepted by the WSN security gateway, which is also capable of mediating the initial DTLS handshake and key negotiation phase. The WSN gateway assumes also the role of 6LBR, and the CA and AC components fulfill the roles previously discussed in Chapter 3. For the purpose of application-layer communications, we consider that a constrained sensing device and an Internet host may both assume the role of the CoAP client or server.

The architecture supports end-to-end security at the transport layer for communications between constrained sensing devices and Internet host, with the DTLS handshake being transparently intercepted and mediated by the 6LBR. The 6LBR thus intercepts and forwards

packets at the transport-layer, an operation that is feasible in the context of its usage as a router supporting communications between the WSN (LoWPAN) and the Internet domains. A major motivation of this model is that it enables us to delegate the computational load related with ECC public-key authentication and key negotiation from constrained sensing devices to the 6LBR, a device that we assume to be without the resource limitations of the WSN sensing devices.

The roles of other components in our architecture are important to support authentication and key negotiation, and have been previously introduced in Chapter 3. The Certification Authority (CA) server supports ECC public-key certification of communicating entities with X.509 certificates. The Access Control (AC) server supports authentication and trust operations between the 6LBR and sensing devices, as required for the delegation of authentication and key agreement in a secure fashion. This server also provides access control and authorization of secure accesses to CoAP resources, either residing on a CoAP sensing device or on the outside of the LoWPAN (in particular on the Internet).

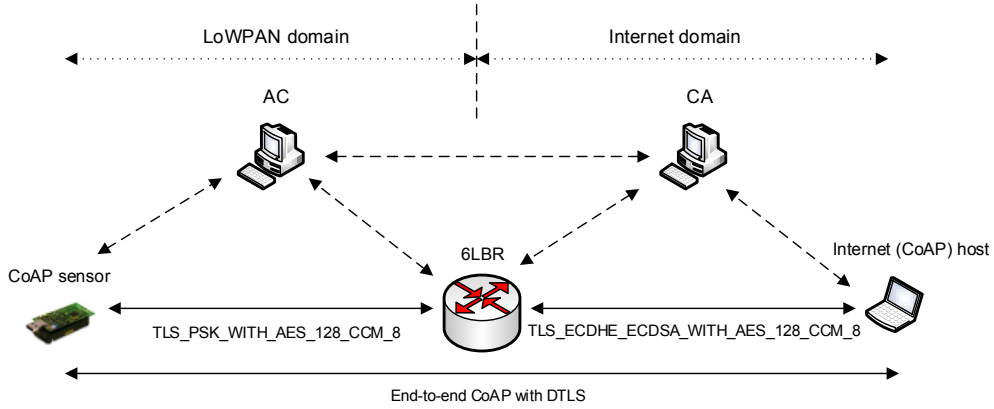

Figure 5.6 - System model for end-to-end security via 6LBR

While guaranteeing end-to-end security, we employ two separate cipher suites for authentication and key negotiation purposes with the two ends of communications, as illustrated in Figure 5.6. This strategy enables the 6LBR to mediate authentication and key negotiation between both ends while guaranteeing that they end up using the same keying material for end-to-end DTLS encryption and integrity, after the initial authentication phase. As Figure 5.6 illustrates, from the point of view of an Internet host the 6LBR supports negotiation via the *Certificates* CoAP security mode using the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 cipher suite.

The participation of the 6LBR in the authentication and key negotiation phase is transparent to the Internet CoAP device, which is unaware of its presence. On the LoWPAN domain, the session is negotiated with the CoAP sensing device using the *PreSharedKey* security mode and the corresponding TLS_PSK_WITH_AES_128_CCM_8 cipher suite. This is also transparent to the CoAP sensing device, which is unaware of the fact that it is authenticating

via a 6LBR. Thus, while end-to-end security may be achieved supporting the most secure CoAP security mode, on the LoWPAN we make use of a security mode more in line with the capabilities of current sensing platforms, according to the conclusions of our previous experimental evaluation study. The TLS_PSK_WITH_AES_128_CCM_8 cipher suite may be considered to be the most appropriate for LoWPAN environments using devices with the characteristics of the TelosB [194], as authentication and initial key agreement may be performed based on pre-shared secret keys.

End-to-end encryption and integrity using DTLS is supported by AES/CCM after the handshake, and as such our architecture must guarantee that both ends of the communications session use the same keying material. Other goal of the architecture is to support mutual authentication between CoAP endpoints. Contrary to proposals such as [205][190][145], our approach supports mutual authentication over standard 6LoWPAN communications and without requiring the usage of special purpose hardware. We proceed by describing how the DTLS handshake is transparently mediated by the 6LBR.

5.4.2 TWO-PHASE MUTUAL DTLS HANDSHAKE

One major mechanism of the proposed end-to-end security model implements a mediated DTLS handshake supporting delegated ECC public-key authentication. As previously observed, the main goal in this context is for the DTLS handshake messages to be transparently intercepted by the 6LBR. As illustrated in Figure 5.7, the handshake is implemented in two phases, with the 6LBR controlling the handshake and supporting ECC cryptographic operations on behalf of CoAP constrained sensing devices.

As illustrated in Figure 5.7, a CoAP Internet client establishes a secure communication session with a CoAP server residing in a sensing device, although the opposite scenario is also supported by the proposed handshake. Thus, the CoAP client may also reside on the WSN and contact a CoAP server in a different WSN domain, or in alternative on an external network or the Internet. Figure 5.3 also illustrates the role of the AC server in the handshake in supporting authentication of LoWPAN devices, using the LoWPAN authentication protocol described later in the chapter.

The initial request transported by a *ClientHello* message is transparently intercepted by the 6LBR, which responds with a *ClientHelloVerify* message. This message enables security against DoS attacks at the transport-layer and contains a cookie generated by the 6LBR [131]. The client is required to respond with the same cookie, thus proving its willingness to communicate and establish a communication session. The delegation of this mechanism to the 6LBR enables the saving of resources and the protection of the CoAP sensing device against requiring the processing of fake requests (cookies).

A secure DTLS session requires the two parties to agree on the cipher suite and encryption keys to be employed. The handshake previously illustrated supports the transportation of the information required to obtain such secret material. According to the rules of the DTLS

protocol, the encryption keys are obtained from a master key that the client and server must share [131]. This master key may, on the other hand, be obtained by both parties using a pair of client and server random values, together with a pre-master secret key. The client and server random values are exchanged during the handshake, while the pre-master shared key is used or obtained depending on the authentication procedure, which fundamentally depends on the cipher suite employed, as we proceed to discuss.

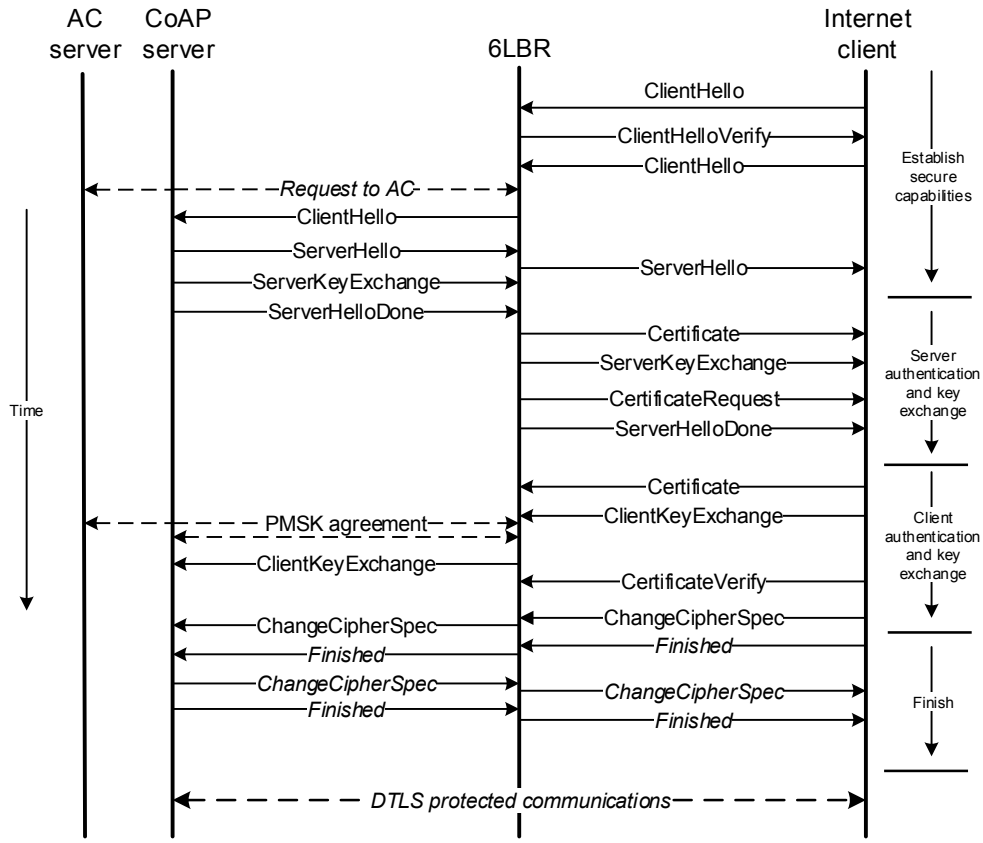

Figure 5.7 - Transparently mediated end-to-end DTLS handshake

When using cipher suites employing public-key authentication the client is allowed to generate the pre-master shared key and send it to the server encrypted with the server's public-key. This is true when using the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 security suite with the *Certificates* CoAP security mode. Pre-shared key security suites as TLS_PSK_WITH_AES_128_CCM_8 [207] don't support this, because at an initial stage the two entities are unable to support the secure transmission of the pre-shared secret. As this limitation would prevent end-to-end agreement of the pre-master secret key in the context of our proposed mediated DTLS authentication, we consider the modification of DTLS pre-shared key authentication using TLS_PSK_WITH_AES_128_CCM_8 to enable the 6LBR to transmit the pre-master secret to the CoAP server running on the sensing device. Thus, the pre-master secret key received from the Internet client is forwarded to the CoAP server and

stored at the 6LBR if required for additional security mechanisms, as we discuss later. In order to guarantee appropriate security for the transmission of this secret in the WSN domain, we also introduce a LoWPAN authentication protocol supported by the CA, which is described later in the present chapter.

Returning to our analysis of the message exchange illustrated in Figure 5.7, the *ClientHello* message confirming the initial request also transports the client random value, the protocol version and the list of cipher suites supported by the client. After reception of this message, the 6LBR requests from the AC server security-related information concerning the destination CoAP sensing device, in particular its supported cipher suites and its X.509 certificate. This information is obtained in the context of the LoWPAN authentication protocol, and includes the supported cypher suites, the X.509 certificate of the device and a description of its capabilities. This information may also be used as input in the process of computing the most appropriate end-to-end security mode for the communication session, according to the framework discussed in Chapter 3 and illustrated in Figure 3.3.

The *ClientHello* message received form the Internet host includes a request for public-key authentication and is forwarded by the 6LBR to the CoAP server with a request for pre-shared key-based authentication, as appropriate for the usage of the TLS_PSK_WITH_AES_128_CCM_8 security suite. We may note that, although this is the currently evaluated cipher suite in the considered integration model, other ciphers may be adopted in the future.

The *ServerHello* message containing the server's response is also forwarded back to the CoAP Internet client, with an acknowledgement for public-key authentication included in the message. The following *ServerKeyExchange* message contains the server random value and is also forwarded to the CoAP client, the same applying to the *ServerHelloDone* message terminating this message flight. In the following message flight the 6LBR authenticates the CoAP server on its behalf by sending the appropriate X.509 certificate previously received from the AC server. The 6LBR also requests that the client authenticates itself with its own certificate. This message flight finishes with the *ServerHelloDone* message. Next the client sends its certificate and a *ClientKeyExchange* message containing the client's random value and pre-master secret key generated by the client, which the 6LBR forwards to the sensing device supporting the CoAP server.

As we illustrate in Figure 5.7, pre-master secret key agreement is preceded by mutual authentication between the 6LBR and the CoAP server via the AC server, using mechanisms detailed later in the chapter, in the context of the proposed LoWPAN authentication protocol. After reception of the *ClientKeyExchange* message, both CoAP entities are in possession of the same pair of random values and pre-master secret key required to compute the DTLS master key, and from this key the secret material for DTLS security may be derived.

It is important to note that this approach also enables the employment of other cipher suites and delegation approaches, as appropriate for different types of sensing devices, as long as compatibility is guaranteed for the pair of ciphers employed. Very-constrained sensing devices may require the full delegation of all DTLS security functionalities to the 6LBR, while on the other hand more capable devices may fully support DTLS. In all situations, it is important to note that the 6LBR is able to learn the pre-master secret key and random values for a given DTLS security session, thus enabling the computation of the final master key and the subsequent derivation of the keying material. This may provide the ground for the employment of other security mechanisms, for example those involving the interpreting and filtering of encrypted CoAP message exchanges, namely for the support of intrusion detection mechanisms detecting and recognizing attacks at the CoAP application-layer.

5.4.3 AUTHENTICATION AND PMSK EXCHANGE ON THE LOWPAN

As previously discussed, in the considered integration model we modify the TLS_PSK_WITH_AES_128_CCM_8 security suite to support pre-master secret key exchange in the context of the DTLS handshake, more precisely by propagating this value towards the CoAP sensing device using the initial *ClientKeyExchange* message. In this context, one important goal is not to compromise end-to-end security by accepting inappropriate low security message exchanges on the LoWPAN. With this in mind, we introduce an authentication protocol supported by the AC server with the goal of guaranteeing appropriate security for communications between the 6LBR and CoAP sensing devices. This authentication protocol is integrated with the two-phase DTLS handshake controlled by the 6LBR, as illustrated in Figure 5.7, and fulfills the important goal of guaranteeing a high-degree of security for end-to-end communications at all stages of an end-to-end DTLS session between a WSN sensing device and an external or Internet device.

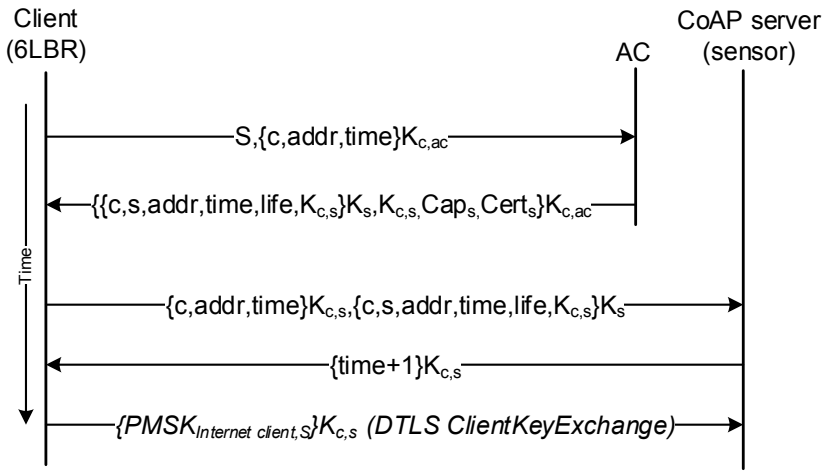

Figure 5.8 - LoWPAN support authentication protocol

In Figure 5.8 we illustrate the functioning of the proposed LoWPAN authentication protocol, more precisely the messages exchanged between the participating entities, the CoAP sensing device, WSN gateway and AC server. The LoWPAN authentication protocol supports confidentiality of the messages exchanged during the handshake and mutual authentication between the 6LBR and CoAP device, while assuming the AC server to be a trusted entity on the network. The proposed LoWPAN authentication protocol inherits characteristics from the Kerberos [209] protocol, and also introduces others required to support our two-phase delegated DTLS handshake and the transportation of the pre-master secret key.

In the context of the system architecture illustrated in Figure 5.6, the AC server is responsible for maintaining security-related information for each registered CoAP sensing device. In particular, for each device the AC stores its client ID, its X.509 ECC certificate and the list of supported ciphers and compression methods. The current mandatory security suite is TLS_PSK_WITH_AES_128_CCM_8, although further ciphers may be adopted in the future, as long as compatibility is maintained with the cipher employed for communications on the Internet domain. The certificate may be preconfigured for a sensing device or in alternative directly obtained from the CA server whenever required, as illustrated in Figure 5.6. Compression negotiation is supported by the DTLS handshake and also with the proposed mediated DTLS handshake. The client ID for a CoAP device is its LoWPAN IPv6 link-local address, and we assume that communications between the AC and 6LBR run over a communications medium without the limitations of the LoWPAN.

The 6LBR and AC server share a secret key (illustrated as $K_{c,ac}$ in Figure 5.8) employed to encrypt messages exchanged between the two devices. The goal of the first message flight in the authentication protocol is to enable the 6LBR to obtain security-related information for the destination CoAP sensing device. This information consists of its certificate, the list of supported encryption and compression methods and an access token for subsequent authentication of the 6LBR to the CoAP device. The request in the first message identifies the CoAP server device and the address of the 6LBR, while also including a timestamp. The AC server builds an authentication token with the previous information plus a lifetime value and the secret session key (illustrated as $K_{c,s}$ in Figure 5.8) to be used by the 6LBR and the CoAP server. The authentication token is encrypted with a secret key that the AC server shares with the CoAP device (K_s in Figure 5.8) and is forwarded unmodified by the 6LBR to the CoAP device. In this reply the 6LBR also receives the secret session key, a list of ciphers and compression methods supported by the CoAP device, and its public-key certificate. Depending on the ciphers supported by the CoAP device, the 6LBR may also decide to terminate the two-phase handshake at this stage, and in consequence the DTLS handshake illustrated in Figure 5.7 would terminate by returning a *Finished* message to the Internet CoAP client.

The second message flight supports mutual authentication between the 6LBR and CoAP sensing device and the secure pre-master secret key exchange. The 6LBR transmits the authentication token previously obtained from the AC server together with a similar token

containing its identification and address plus a timestamp. The CoAP server compares the information contained in the two tokens received in order to authenticate the 6LBR, while also analyzing the timestamp and lifetime values. Such values offer protection against message replay attacks. In the case of successful authentication, the CoAP server is now in the possession of the secret session key (illustrated as $K_{c,s}$ in Figure 5.8). The next reply message is encrypted with this key and authenticates the CoAP server to the 6LBR, by having the server transmit the received timestamp plus one.

The final message is the *ClientKeyExchange* message sent in the context of the two-phase mutual DTLS handshake. This message transports the pre-master secret key and modifies the TLS_PSK_WITH_AES_128_CCM_8 security suite as previously discussed. After this last message the DTLS handshake proceeds, as previously illustrated in Figure 5.7. After the computation of the master secret and the keying material on the CoAP client and server, end-to-end DTLS security may be enabled employing AES/CCM. AES/CCM may be supported in software on the Internet CoAP entity and (when available) by hardware cryptography on the sensing device. The support of further ciphers for the encryption of communications between the 6LBR and sensing devices on the WSN makes this authentication protocol extensible and adaptable to applications using other security suites to protect communications on the LoWPAN domain, while the current support of the TLS_PSK_WITH_AES_128_CCM_8 security suite guarantees compatibility with end-to-end transport-layer security as currently defined for the CoAP [34] protocol.

5.5 EXPERIMENTAL EVALUATION OF MEDIATED DTLS TRANSPORT-LAYER SECURITY

The mechanisms previously discussed in the context of the proposed integration model may contribute to the security of Internet-integrated LoWPANs and to the intelligent allocation, to security, of limited resources available on constrained CoAP sensing platforms. ECC public-key authentication and key negotiation as proposed for CoAP may be supported for Internet-integrated sensing applications using devices unable to otherwise support it directly, due to the impact of ECC encryption on such platforms. On the other hand, attacks and threats motivated by the integration of LoWPAN communications and devices with the Internet may be efficiently circumvented using mechanisms deployed on a non-constrained WSN Gateway.

We next describe the experimental evaluation of the previously described mechanisms supporting end-to-end security at the transport-layer. As in our previous evaluation of network-layer 6LoWPAN security, for the purpose of the evaluation of the research proposals we consider the methodology discussed in Chapter 3 and illustrated in Figure 3.4. The following evaluation also enables the employment of end-to-end transport-layer security in the context of the framework illustrated in Figure 3.3.

5.5.1 EXPERIMENTAL EVALUATION SETUP

In Figure 5.9 we illustrate the communications model considered for our experimental evaluation of end-to-end transport-layer security with the mediated authentication and key agreement handshake. As previously discussed, the proposed solution involves also the employment of an authentication protocol supporting appropriate security on the LoWPAN. Regarding the reference model for end-to-end security discussed in Chapter 3, we also consider that the Security Manager and Node Manager components on the 6LBR and CoAP sensing device provide the support required for the proposed DTLS mediated handshake and LoWPAN authentication protocol. We also consider the employment of access control information via an AC server, as previously discussed, and of the Certification Authority.

As in our previous evaluation of end-to-end 6LoWPAN security, the WSN gateway (6LBR) supporting the mediated DTLS handshake also supports routing between an Ethernet IPv6 network and the IEEE 802.15.4 LoWPAN, by employing a second TelosB mote as a bridge. This device also supports routing advertisements on the WSN domain, as required for the support of 6LoWPAN.

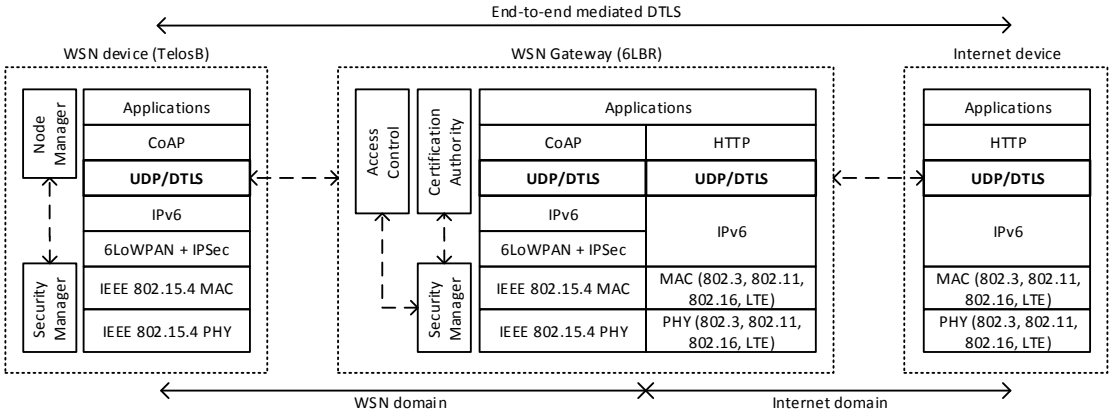

Figure 5.9 – Reference model for the evaluation of end-to-end mediated transport-layer security

The main goal of the following experimental evaluation is to determine the impact of end-to-end security using the mediated DTLS handshake in comparison with the current proposal of supporting pure end-to-end DTLS security [34]. The evaluation focuses, not only on the potential benefits of delegating the authentication and key agreement operations to the 6LBR, but also on the impact of the proposed LoWPAN authentication protocol previously described on the resources of WSN sensing devices.

The reference model illustrated in Figure 5.9 employs a TelosB [194] sensing platform and Linux hosts, with the TelosB supporting the TinyOS [53] operating system with the Berkeley Low-IP (BLIP) 6LoWPAN stack, plus CoAP support and the two different DTLS configurations. As in our previous evaluation of network-layer security in Chapter 4, we consider that,

although the experimental results are specific to the TelosB, they may provide an acceptable reference, considering the representativeness of this sensing platform. We also support standalone AES/CCM encryption available in the TelosB using the encryption code from the Shanghai Jiao Tong University [200], while ECC is supported using code based on TinyECC [208]. The 6LBR, CA server, AC server and the Internet CoAP entities are supported using Linux, according to the evaluation model illustrated in Figure 5.9. As in our previous evaluation of 6LoWPAN security, the 6LBR supports routing between an Ethernet IPv6 network and the IEEE 802.15.4 LoWPAN and employs a second TelosB device in bridge mode for communications with the WSN. The external Internet CoAP client employs *libcoap* [210] integrated with DTLS support. The TelosB sensing device and the AC server support the proposed LoWPAN authentication protocol.

5.5.2 IMPACT ON THE RESOURCES OF CONSTRAINED SENSING DEVICES

Our initial evaluation is on the RAM and ROM memory required to support end-to-end security at the transport-layer using the proposed research solutions, given its scarcity on sensing platforms such as the TelosB. We again consider memory to be a fundamental aspect of the effectiveness of new research solutions addressing end-to-end security in the context of Internet-integrated WSN, as previously discussed in Chapter 3.

5.5.2.1 Memory footprint of end-to-end security

In our following analysis, for illustration purposes the proposed end-to-end CoAP security mode is identified as ME2ECoAP, thus mediated end-to-end CoAP security using the delegated handshake with mutual authentication. On the other hand, the original end-to-end CoAP security mode is identified as E2ECoAP, or end-to-end CoAP.

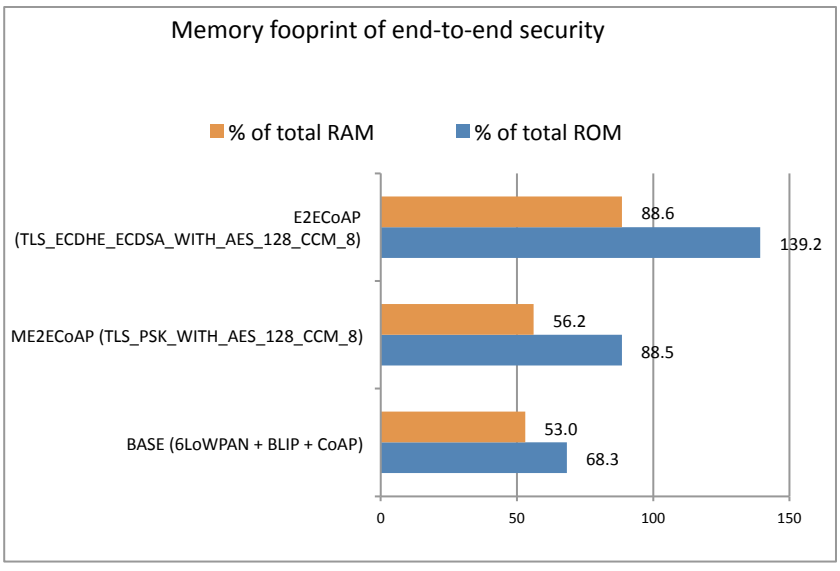

Figure 5.10 - Memory footprint of transport-layer end-to-end security

In Figure 5.10 we illustrate the impact of the support of the two end-to-end security modes in respect to its usage of memory on a TelosB sensing platform, and also a base usage scenario without end-to-end security, which provides a basis for comparison. We must note that the values illustrated in Figure 5.10 are derived directly from our previous experimental evaluation study of CoAP security, considering the usage of the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 security suite for the support of security modes using ECC-based authentication and key agreement, and of the TLS_PSK_WITH_AES_128_CCM_8 suite for the usage scenario where ECC-based computations are delegated to our WSN gateway. For the obtainment of the illustrated values we also consider the support of the nesC code required for BLIP and CoAP in TinyOS, and also of the code required to support the appropriate cipher and the DTLS security protocol. We also consider the support of TLS 1.2 PRF using SHA-256, as required by CoAP to support integrity. For the measurement of the impact of ME2ECoAP, we also account for the code required to support the LoWPAN authentication protocol.

From Figure 5.10 we may observe that hardware-level encryption doesn't come without a non-negligible overhead on memory, particularly ROM. The limitations of the TelosB sensing platform in terms of memory are again visible, as more ROM memory is required to fully support end-to-end security using the original CoAP *Certificates* security mode. RAM may also be a problem in usage scenarios where larger applications require more available memory from the sensing device, the same also applying to the storage and processing of X.509 certificates and related public-keys. In general, we may observe the superior performance of ME2ECoAP in terms of memory usage and availability using the TelosB to support the CoAP server.

5.5.2.2 Impact of security on the lifetime of CoAP sensing applications

Energy is certainly another scarce resource in constrained sensing platforms, and many sensing applications must be designed with battery-powered sensing devices in mind and to run for acceptably long periods of time. In order to obtain the expected lifetime of IoT sensing applications employing end-to-end security, we start by experimentally measuring the impact of packet processing, security and communications on the energy available on the TelosB. As previously discussed, such measurements enable the calculation of the impact of security on the lifetime of sensing applications employing Internet-integrated WSN, according to the experimental evaluation methodology of Figure 3.4.

In our experimental evaluation study energy was obtained using experimental measurements of the voltage across a current resistor placed in series with the battery pack of the TelosB. In particular, we measure the energy required to support the DTLS handshake (handshake processing plus handshake communications energy) and the energy required to support DTLS encryption using AES/CCM (DTLS encryption plus communications energy). For all measurements we consider the usage of 6LoWPAN 102-byte packets as previously discussed in the context of Figure 5.1.

Regarding the handshake, the original DTLS handshake requires a total of 39 6LoWPAN 102-byte messages and a total of 54.4 mJ (Millijoules), according to our measurements. This is in contrast with our delegated two-way handshake, which involves 15 LoWPAN messages and 0.001 mJ, also accounting in the messages required for the support of the proposed LoWPAN authentication protocol. Regarding DTLS encryption, 0.0002 mJ are required to process security for a packet using AES/CCM encryption and 10.89 mJ for digital signing the same packet using ECC, as required for the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 security suite. As previously observed, we again note that ECC-based cryptography represents a bottleneck using the TelosB sensing platform. The previous values are total, measured from the reception of a 6LoWPAN packet to the time when cryptography finished processing the packet, and thus represents the total energetic effort to process end-to-end security for a transport-layer packet. Finally, the energy required for the processing of a packet and related security headers was measured as 0.007 nJ (Nanojoules) and is accounted for in our following evaluation. From the experimental values previously discussed we derive the expected lifetime for a sensing application, which we illustrate in Figures 5.11 and 5.12.

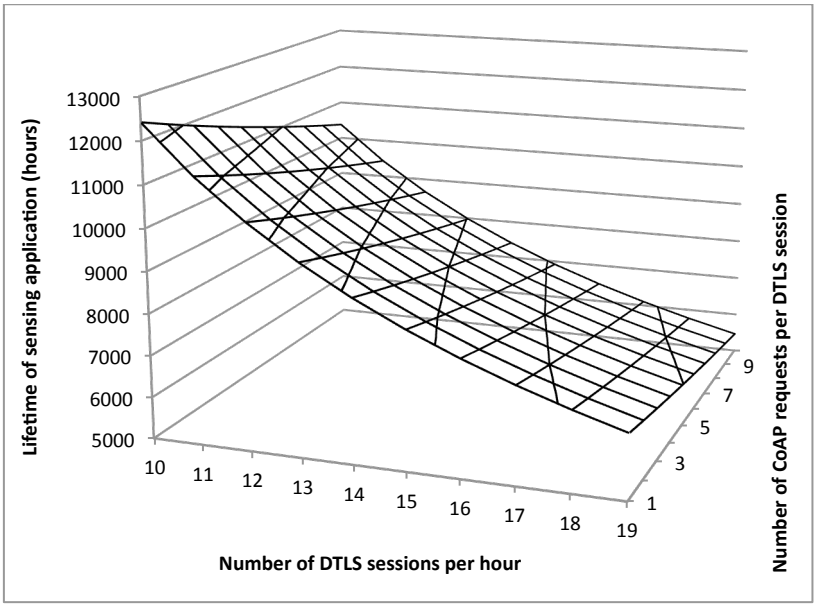

Figure 5.11 - Impact of end-to-end security on the lifetime of applications (E2ECoAP)

The expected lifetime values illustrated in Figures 5.11 and 5.12 considers the usage of the TelosB sensing device powered using two new AA LR-6 batteries and applications with different requirements in terms of the number of DTLS sessions established per hour and the number of CoAP requests served per DTLS session. We count a CoAP request as two 102-byte 6LoWPAN packets, one transporting a confirmable request and the other its corresponding reply.

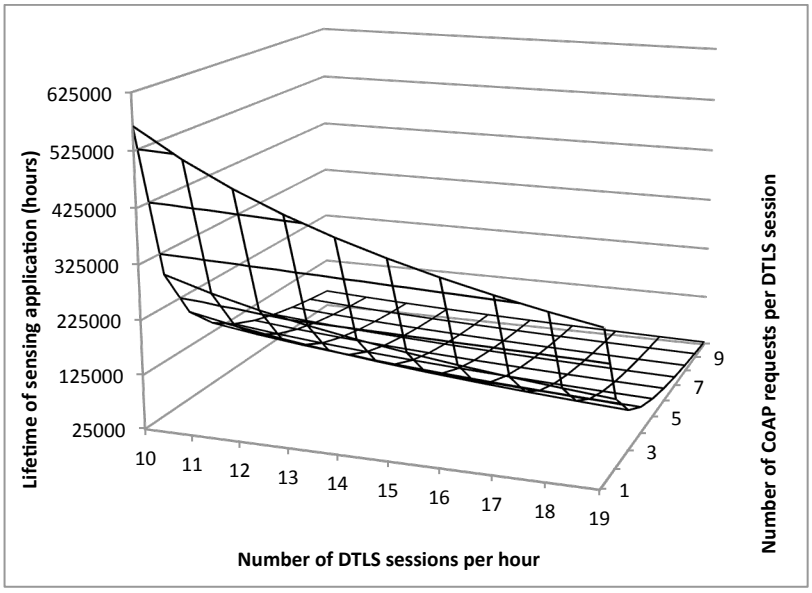

Figure 5.12 - Impact of end-to-end security on the lifetime of sensing applications (ME2ECoAP)

In Figure 5.12 we observe again the superior performance of ME2ECoAP, given that in the worst scenario (corresponding as illustrated to 19 DTLS sessions per hour with 10 CoAP requests per session) the expected lifetime is about 29900 hours, approximately 5 times the corresponding value for E2ECoAP (5461 hours). We may also observe a more expressive decline for ME2ECoAP in respect with the expected lifetime when the number of CoAP requests per session increases. This is due to the larger impact of AES/CCM security in comparison with the impact of the DTLS handshake, in contrast with E2ECoAP in Figure 5.6 for which the lifetime is dominated by the much larger impact of the DTLS handshake. Despite this, in all usage scenarios ME2ECoAP is superior in respect to the expected lifetime.

Overall, ME2ECoAP would be the best choice for sensing applications designed to operate in a closed fashion, where CoAP devices are able to maintain security sessions with a closed set of Internet devices for long time periods, but also for open applications where CoAP devices accept requests from any Internet client.

5.5.2.3 Impact of security on the communications rate of CoAP sensing applications

As advanced mechanisms such as multi-threading are usually absent from low-end microcontrollers such as the MSP430, the computational time required to support security directly influences the maximum communications rate that a sensing device may support. This is also an important evaluation strategy to conclude on the effectiveness of end-to-end transport-layer security in the context of Internet-integrated WSN applications, as discussed in Chapter 3.

We experimentally measure the computational time required to support the DTLS handshake (handshake processing plus handshake communications delay). The DTLS handshake employing the original CoAP security proposal requires 10.09 s, in major contrast

with the DTLS delegated handshake, which requires 15.39 ms. Such values include the time required for communications in the context of the handshake, and for the later also the time required for the LoWPAN authentication protocol. This clear difference is again due to the large impact of ECC cryptography on the TelosB, giving that ECC digital signing is required to process a few of the messages of the handshake. ECC encryption for digital signing requires a total of 2019.6 ms, while with ME2ECoAP this is not an issue since ECC computation is delegated to the 6LBR proxy. We again include the overhead of AES/CCM, which was measured as 3.6ms per packet.

Based on the experimentally obtained values previously discussed, we are able to derive the maximum number of CoAP requests that a CoAP sensing device may support with end-to-end security, which we illustrate in Figure 5.13. The illustrated values reflect the weight of the DTLS handshake in the overall CoAP communications rate. We may observe that, although the difference in the performance of the two end-to-end security modes may be of less significance for applications requiring a smaller number of DTLS sessions per hour, for others ME2ECoAP is clearly the best choice.

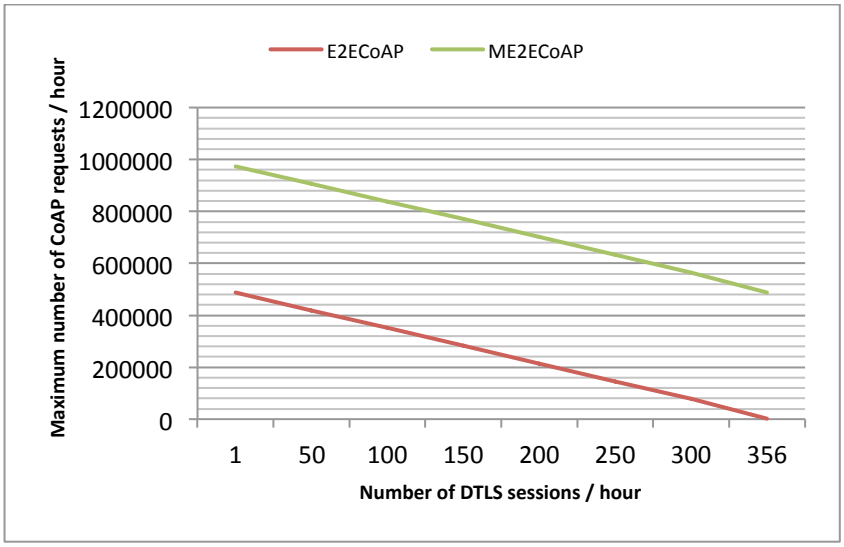

Figure 5.13 – Impact of end-to-end transport-layer security on the maximum transmission rate of CoAP applications

We may observe that the original DTLS handshake is only viable up to around 356 DTLS sessions per hour (roughly one new session each 10 seconds), due to the computational weight of ECC encryption in the context of the DTLS handshake. For applications requiring a larger numbers of secure sessions per hour the current proposal for CoAP security is completely unviable. Regardless of the number of DTLS sessions per hour required by a particular sensing application, ME2ECoAP is clearly the most appropriate choice.

5.5.3 APPLICATION SECURITY AND FUNCTIONAL PROFILES

Our following discussion considers the employment of security and functional profiles designed as appropriate for specific applications, and which enable a more focused evaluation of transport-layer security complementing our previous analysis. As discussed in Chapter 3, application security and functional profiles play an important part in the framework for reconfigurable end-to-end security in the context of Internet-integrated WSN, which is illustrated in Figure 3.3.

Regarding the definition of appropriate functional and security profiles, we consider two types of applications, as we proceed to discuss. One is that of applications requiring a moderate number of DTLS sessions per hour, also with a moderate number of CoAP requests per DTLS session. For experimental evaluation purposes we consider from 1 to 400 DTLS sessions per hour, and 2 CoAP requests per DTLS session. The other is that of applications requiring a higher number of DTLS sessions per hour, also with a higher number of CoAP requests per DTLS session. For experimental evaluation purposes we consider from 14 to 7200 DTLS sessions per hour with 10 CoAP requests per DTLS session. We must also note that a CoAP request involves two messages, one containing the request sent to the server and (at least) other containing the corresponding reply. We are also interested in evaluating two end-to-end security modes, one with full end-to-end DTLS security supported by the sensing device, and the other with the proposed DTLS handshake plus the LoWPAN authentication protocol.

As we proposed in [202], end-to-end communications with support for ECC-based public-key infrastructures may serve sensing applications in areas such as healthcare or vehicular applications, and the proposed mediated handshake may support DTLS security for such applications, with added advantages in terms of the lifetime of sensing devices as well as the protection of WSN domains against external attacks. From the previously discussed experimental measurements we may derive expected lifetime values for sensing applications described by the two profiles discussed, which we illustrate in Figures 5.14 and 5.15. As in our previous evaluation, for the calculation of the estimated lifetime we consider the usage of a TelosB powered using two new AA LR-6 batteries.

For both usage scenarios we may again observe a clear advantage of the proposed delegated DTLS handshake, particularly for a lower number of DTLS sessions per hour. The illustrated values also consider the energy required to support the LoWPAN authentication protocol. This advantage in less expressive for a higher number of DTLS sessions per hour, mostly due to the higher impact of AES/CCM encryption in comparison with the impact of the DTLS handshake. If we consider that many IoT applications will probably require low or moderate transmission rates, the proposed mechanisms prove to be effective for devices with the characteristics of our TelosB reference sensing platform.

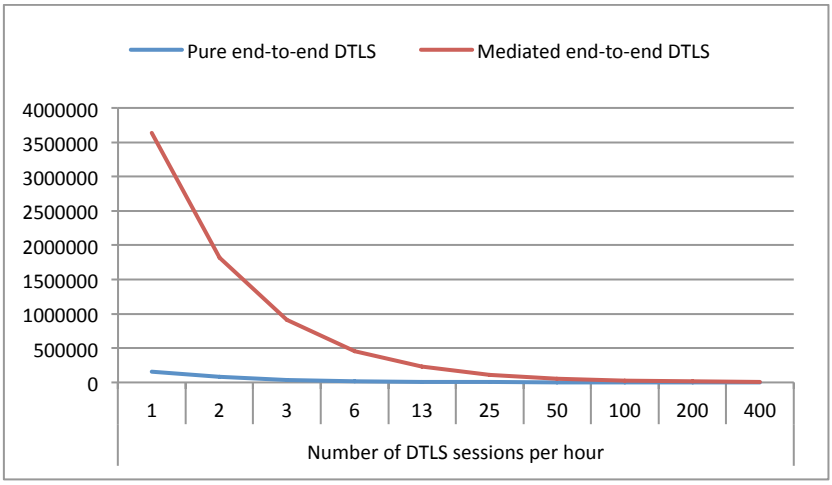

Figure 5.14 - Impact of end-to-end security on the lifetime of sensing applications (moderate usage profile).

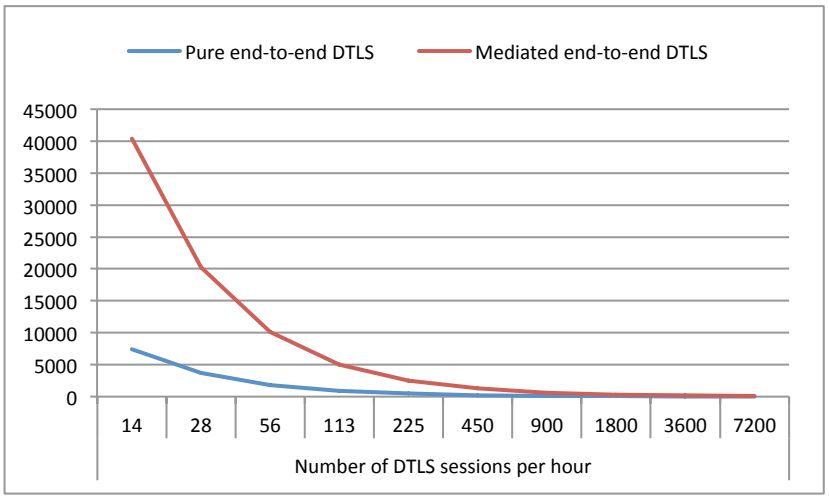

Figure 5.15 - Impact of end-to-end security on the lifetime of sensing applications (higher usage profile).

As previously discussed, we may also consider other advantages of the proposed delegated and mediated handshake, particularly in what respects the protection of LoWPAN communication domains against external (Internet) threats, and the availability of a cryptographic basis for the support of other security mechanisms which depend on the analysis of encrypted application-layer communications.

5.6 SUMMARY

Many of the currently envisioned IoT sensing applications may require, or at least benefit from, the usage of end-to-end standard Internet communications between constrained

sensing devices and Internet hosts or external backend servers. In the present chapter we discuss research proposals to support measurable and controllable end-to-end security at the transport-layer, in the sense that authentication and key agreement delegation to a more powerful WSN security gateway provides an effective solution for the support of CoAP security on more constrained sensing devices, while other end-to-end security modes are also supported. The mechanisms proposed in this chapter support such functionalities, while being completely compatible with transport-layer security as currently proposed for CoAP, and also transparent from the point of view of the two CoAP communication entities.

The research solutions described in the present chapter provide benefits in respect to the efficient support of ECC authentication and key agreement, and also contribute to promote security of LoWPAN devices and communications. As verified with our experimental evaluation, when employing current sensing platforms the delegation of costly ECC computations to a more powerful device clearly pays off, even with the additional overhead of supporting the LoWPAN authentication protocol required by our proposal. Other challenges remain to be addressed in the context of the proposed mechanisms, for example the design of different approaches to end-to-end security or new techniques to decide on the security mode in the presence of particular sensing platforms or application profiles.

As we consider in the reference integration model described in Chapter 3, mechanisms may also be designed to support end-to-end security in the presence of mobile (roaming) devices. If different IPv6 prefixes are employed in the origin and destination WSN domains, a change of address may take place. In this context, mechanisms may be designed to guarantee the transparency of mobility from the point of view of end-to-end transport-layer security, so that a device moving between different LoWPAN domains is able to continue using previously negotiated security sessions and its associated keying material. The support of transparent mobility from the perspective of end-to-end security may be supported via trust relationships established between AC servers on different LoWPAN domains, as well as by the security gateways serving communications with such domains.

The proposed LoWPAN authentication protocol is a fundamental component of the proposed delegation model, as it enables appropriate security in the WSN part of end-to-end transport-layer communications. The challenge here is to provide appropriate LoWPAN security in the context of end-to-end transport-layer communications, with a minimal or acceptable impact on the resources of constrained WSN sensing devices. As we have verified in the experimental evaluation, the impact of the proposed LoWPAN authentication protocol does not compromise the lifetime of sensing applications nor the achievement of acceptable compromised between security and the resources required from the WSN.

Other challenge guiding future work in the context of the proposed research solutions may be to design different end-to-end security approaches or new techniques to decide on the most appropriate security mode in the presence of particular sensing platforms and applications. The LoWPAN authentication protocol may also provide the ground for the

employment of different security approaches, for example by employing AES/CCM to support integrity only (by using CBC-MAC) or encryption mechanisms better appropriate to sensing platforms that do not support AES/CCM at the hardware, for the purpose of securing communications on the LoWPAN domain between the 6LBR and CoAP sensing devices supporting DTLS.

The proposed approach enables the employment of other cipher suites and delegation approaches, as appropriate for different types of sensing devices, and as long as compatibility is guaranteed for the pair of ciphers employed. Very-constrained devices may require the full delegation of all DTLS security operations to the 6LBR, while on the other hand more powerful devices may fully support DTLS. The adaptation of security to the requirements and characteristics of applications and devices has been previously discussed in Chapter 3.

6 END-TO-END COAP APPLICATION-LAYER MESSAGE SECURITY[5]

In the previous chapters we focused on how end-to-end security at the network and transport layers may be implemented with different strategies and impact on the resources of constrained WSN devices. In the case of network-layer security, we proposed and evaluated the addition of new compressed security headers to the 6LoWPAN adaptation layer, while for transport-layer security we address the support of DTLS authentication and key agreement by delegating costly security operations to a security gateway in a transparent fashion, while supporting other security mechanisms and functionalities. The two research solutions thus represent complementary approaches to the problem of end-to-end security in the context of Internet-integrated WSN.

In the present chapter we focus on how security may be supported for end-to-end communications at the application-layer, again targeting an approach that may complement the previous research proposals. We consider the design of security mechanisms to operate in the context of the communication protocol itself, with various benefits related to how applications may employ security. In our following discussion we start by discussing the general goals of application-layer security in the context of Internet-integrated WSN, and next we describe our research solution to address end-to-end security in the context of the CoAP application-layer protocol. As in the previous proposals, later in the chapter we also discuss the experimental evaluation of our research proposal.

6.1 INTRODUCTION

Although many of the applications currently envisioned for the Web of Things (WoT) are critical in respect to security, the fact that they are envisioned to employ very constrained sensing platforms and wireless communications complicates the design of appropriate security solutions. As already discussed, in practice many applications are required to accept

[5] *This chapter has supported the following publications*:

- Granjal J, Monteiro E, Silva J. *On the effectiveness of end-to-end security for Internet-integrated sensing applications* (*best paper award*), The IEEE International Conference on Internet of Things, iThings 2012
- Granjal J, Monteiro E, Silva J. *Application-layer security for the WoT: Extending CoAP to support end-to-end message security for Internet-integrated sensing applications*, The 11th International Conference on Wired/Wireless Internet Communications WWIC 2013
- Granjal J, Monteiro E, Silva J. *On the Feasibility of Secure Application-Layer Communications on the Web of Things*, The 37th IEEE Conference on Local Computer Networks, LCN 2012

compromises between security and the usage of resources available on constrained sensing platforms. Energy is a scarce resource in typical wireless sensing devices, and in consequence WSN environments are required to employ link-layer LoWPAN communication technologies such as IEEE 802.15.4 [74]. WSN environments thus employ low-energy wireless communications at low transmission rates using small packets, in order to minimize transmission errors. These limitations deeply influence mechanisms designed at upper layers of the stack, as is the case of 6LoWPAN-based communication technologies designed for constrained sensing platforms.

As addressed in previous chapters, technologies are being designed to support the integration of LoWPAN environments such as WSN with the Internet, and which are expected to play an important role in the fulfillment of the vision of the WoT. Various communications and security technologies for the WoT are currently in the design phase, and consequently a communications and security architecture for the WoT is currently not completely defined. This aspect also motivates the identification and usage of the reference integration architecture previously described in Chapter 3 to support end-to-end security at the various layers, including at the application-layer.

Focusing on how CoAP [34] adopts security, we observe that the current choice to support end-to-end security is to adopt the Datagram Transport Layer Security (DTLS) Protocol [127]. This design choice implies that security is not integrated with the application-layer protocol itself, but rather transparently applied at the transport layer to all CoAP messages. The adoption of transport-layer security makes sense from the point of view of the current Internet architecture, where TLS [128] as the transport-layer is used to protect HTTP web communications. Since 6LoWPAN environments currently support only UDP, DTLS appears as a logical choice in protecting communications at higher layers. Despite this, in this chapter we argue that this approach misses various advantages of addressing security at the application layer, which we explore to propose new security mechanisms for CoAP. With such advantages in mind, in this chapter we propose the addition of appropriate options to the protocol, which extend CoAP to support application-layer security.

6.2 LIMITATIONS OF THE TRANSPORT-LAYER SECURITY APPROACH

The current Internet architecture illustrates the importance of employing complementary security mechanisms at the various protocol layers, as such mechanisms may better support applications with different security and functional requirements. The design of new security mechanisms for Internet-integrated WSN may also take this aspect in consideration, and in particular in what respects the protection of end-to-end communications, also because of the need to support constrained sensing devices. End-to-end transport-layer security using 6LoWPAN/IPSec or DTLS may be appropriate to applications requiring the transparent encryption of all network-layer or transport-layer communications, as we have previously addressed. On the other side, applications may also benefit from a more granular and

flexible approach to security, which end-to-end transparent security at lower layers is unable to support.

The current CoAP specification defines bindings to the DTLS (Datagram Transport-Layer Security) Protocol in order to enable security at the transport-layer. DTLS may apply security to all messages in a given security session, thus supporting confidentiality, authentication and integrity for all CoAP communications. While DTLS is a good choice in respect to its support of efficient AES/CCM cryptography as available at the hardware in IEEE 802.15.4 sensing platforms, we may identify a few aspects motivating our alternative approach to CoAP security, as we discuss next. Please also note that another particularly important aspect of the usage of DTLS to protect CoAP communications is the computational and energetic cost of its initial authentication and key agreement handshake, which we targeted in the previous chapter of the thesis.

One important aspect to consider when employing DTLS to protect CoAP communications is that security is transparently applied to all CoAP messages of a given communication session, irrespective of its type or contents. Applications are thus unable to define how security is applied according to the type of messages exchanged, the contents of the message or the semantics of the CoAP protocol. This limitation prevents applications from applying granular security policies to its communications and from saving critical resources on constrained sensing platforms, in particular energy. DTLS thus difficult the definition and application of granular security policies, and this implies that security may be more costly than what would be required by particular applications. The fact that a secure session must exist between the client and server may also be limitative for many applications, which may require that devices are able to communicate securely without the predefinition of security-related parameters and configuration.

Other aspect to consider is that with DTLS security applications are required to employ a static security configuration, since after the DTLS handshake all messages are protected using a particular cipher suite and the corresponding cryptographic algorithms and keys. Applications are thus unable to employ different security algorithms and keys to protect different types of CoAP messages, for example according to its role in the protocol or its contents. The maintenance of predefined or fixed end-to-end security associations may thus be limitative and difficult the usage of security to protect application-layer using CoAP.

Another aspect that may complicate the employment of DTLS to protect application-layer communications in the context of Internet-integrated WSN is that the CoAP protocol is being designed to support intermediaries (proxies) in both forward and reverse modes [34]. CoAP proxies will be in fact very useful in supporting accesses to resources available on WSN devices in a controlled and energy-efficient way, with the help of WSN mechanisms as subscription and push protocols. In this context, the problem is that end-to-end security is incompatible with the employment of CoAP intermediaries. Although end-to-end communications are at the hearth of IPv6, CoAP intermediaries may in fact break DTLS

security. Alternative approaches may thus be required also to support security when using CoAP with intermediaries, and new mechanisms designed with this purpose may support the secure transversal of multiple LoWPAN domains, as well as of flexible authentication mechanisms, requirements that may promote the usefulness of application-layer security for applications encompassing WSN in different administrative domains.

The previously discussed limitations of the employment of DTLS to protect CoAP communications motivate our alterative approach of designing and evaluating security at the application-layer, as we discuss in this chapter. As with the research solutions proposed in the previous chapters, it is our goal that CoAP security as we propose next may complement other end-to-end security approaches, and consequently enrich the set of security mechanisms available to protect end-to-end communications in the context of Internet-integrated WSN. As discussed in Chapter 3, the most appropriate mechanism for a given sensing application may be then determined statically or dynamically, according to various functional and security aspects described by appropriate application profiles.

We may consider that different approaches to end-to-end security may not only enrich the set of solutions available for Internet communications in the context of Internet-integrated WSN, but also contribute to a more intelligent allocation of resources to security, particularly considering its computational and energetic impact. The support of granular security policies is only one of the advantages of application-layer security, as we discuss later in the chapter.

Although security for WSN is a prolific research area, investigation concerning the integration of LoWPAN environments with the Internet is recent, and few research proposals address security for communications at the application-layer in such environments. As discussed in Chapter 2, previous proposals addressing the integration of security at the application-layer with CoAP consisted in the definition of options for the activation and deactivation of security contexts [149], and the addition of options to support intermediaries and link-layer security [150]. None of such proposals address the support of granular security, of flexible authentication and the secure transverse of multiple trust domains, aspects motivating our research efforts. Ours was thus the first proposal with such goals in mind [211] mind, and is detailed next.

6.3 A PROPOSAL FOR CoAP APPLICATION-LAYER MESSAGE SECURITY

As previously discussed, payload space is a scarce resource in LoWPAN IEEE 802.15.4 communication environments, and as a consequence 6LoWPAN and CoAP incorporate header and address compression whenever viable. At the 6LoWPAN adaptation layer, 102 bytes of payload space are available for protocols such as DTLS and CoAP at upper layers, and for applications. 6LoWPAN IPHC shared-context header compression [71] enables the compression of the UDP and IPv6 headers down to 10 bytes, while CoAP employs a 4-byte fixed header and DTLS a 13-byte header. Without transport-layer security, 88 bytes are available for applications using CoAP without incurring in costly 6LoWPAN fragmentations.

An important concept of CoAP is that, other than a basic set of information, most of the information is transported by options. Options thus extend the functionalities of the protocol, and therefore despite security being absent from the current CoAP specification, new options may be adopted that extend CoAP to support application-layer security. CoAP options may be designed to be critical, elective, safe or unsafe. In short, a critical option is one that an endpoint must understand, while an elective option may be ignored by an endpoint not recognizing it. Safe and unsafe options determine how an option may be processed by an intermediary entity. An unsafe option needs to be understood by the proxy in order to safely forward it, while a safe option may be forwarded even if the proxy is unable to process it. As discussed in our SoA study in Chapter 2 and Figure 2.10 illustrates, each option instance in a CoAP message specifies the Option Number of the CoAP option, the length of the Option Value and the Option Value itself.

Our proposed mechanisms to integrate security at the application-layer with the CoAP Protocol targets the issues previously discussed and may provide various benefits, which we also address in the context of the experimental evaluation of our research proposal. Packet payload space usage is one aspect to address, as security-related information at the application-layer may be transported in the same context as headers and control information of the CoAP protocol itself. As in our previous research proposals, the overhead in terms of the required energy and computational time on constrained sensing devices is also worth investigating, given the significance of such aspects on the lifetime and the communications rate of wireless sensing applications.

In our following discussion we describe the format and usage of new CoAP options designed to support application-layer security according to various goals and envisioned deployment scenarios. All the discussed CoAP options are critical, unsafe and non-mandatory. The non-mandatory status of such options results from the fact that applications may opt for security mechanisms at different layers, particularly at the network and transport layers as per our approaches in previous chapters.

6.3.1 THE *SECURITYON* COAP SECURITY OPTION

The first option designed to support application-layer security with the CoAP protocol is the *SecurityOn* option, which we illustrate in Figure 6.1. This option identifies the protection of a given CoAP message by application-layer security, and transports the information necessary to process security for the message. In particular, this option identifies how security is applied to the message, what entity should process or verify security for the message, the security context that the message belongs to, and temporal information relevant to ascertain about the validity of the message. All CoAP options are formatted in the TLV (Type, Length, Value) format, and therefore the length of the *Destination Entity* field of the option illustrated in Figure 6.1 may be obtained from the total length of the option.

The *SecurityApplied* field in Figure 6.1 identifies if the CoAP message is encrypted, signed, or both encrypted and signed. A given application may thus protect its CoAP messages differently, according to criteria such as the contents of the messages or its type. The *DestinationEntity* field identifies the entity that should process or verify security for the message, and this may be either the final CoAP destination device or an intermediary in the path towards the final destination of the CoAP message. The actor URI identifies this entity, and this enables the usage of CoAP secure communications that are managed by an intermediary. This field thus states that the secured CoAP message is meant for any endpoint acting in the capacity indicated by the actor URI. This option may be employed more than once in a given CoAP message, providing the support for the secure transversal of multiple trust domains, since the various intermediaries may use different encryption keys.

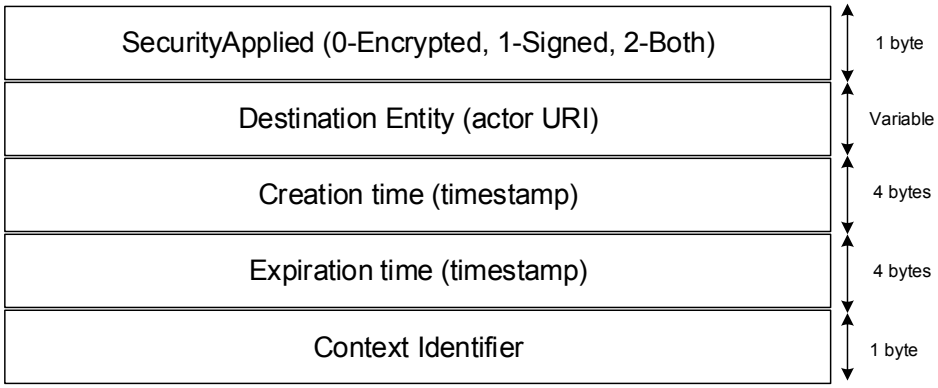

Figure 6.1 - SecurityOn CoAP security option

As illustrated in Figure 6.1, the *SecurityOn* option also transports temporal values that support the verification of the legitimacy of the message. In particular, the *Creation time* and *Expiration time* field of the message are inserted by its creator, and enable an intermediary or the final CoAP destination to ascertain about the validity of the message. Finally, the *Context Identifier* field enables the client, server and/or intermediaries to contextualize the message in terms of security, in particular in determining the appropriate ciphers and keys. Various contexts may be active for a given CoAP sensing application, given that different types of CoAP messages may be secured differently in the context of a single application, as previously discussed.

6.3.2 THE *SECURITYTOKEN* COAP SECURITY OPTION

The support of identification and authorization in the context of application-layer security motivates the design of the *SecurityToken* option, which is illustrated in Figure 6.2. This option enables the usage of identity and authorization mechanisms at the application-layer, on a per message basis. Using this option, a CoAP client (a requester) may state his identify and include authentication information, in order to obtain access to a given CoAP resource

on the server. By employing granular security, applications may provide accesses to CoAP resources with different criteria, according to the identity of the client and to the criticality of the sensing data requested. Thus, applications may employ various security contexts and also require different authorization mechanisms in the context of a single or multiple security contexts, providing support for granular and flexible security policies.

A CoAP message only transports data related with one particular authorization mechanism at a time, and thus the length of the corresponding field in Figure 6.2 is obtained from the total length of the option. A CoAP destination or intermediary entity along the path of the message may enforce the usage of a *SecurityToken* option in order to authorize CoAP requests.

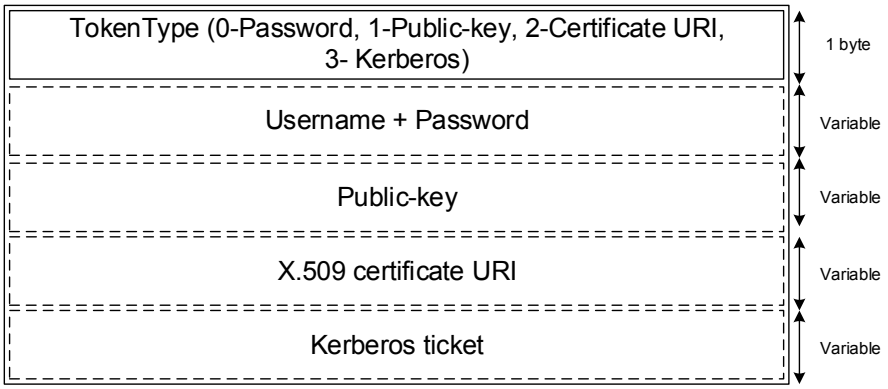

Figure 6.2 - SecurityToken CoAP security option

As illustrated in Figure 6.2, the currently defined format for this options enables a client to authenticate itself using a simple username and password scheme, using an identifying public-key, a X.509 certificate referred by a URI (using a NULL-terminated string) or a Kerberos [209] ticket previously obtained form a domain server (in binary format). Further authorization mechanisms may be designed or adopted in the future by defining appropriate identification values and the format of the authorization data to be transported.

A CoAP requestor may be authorized at a destination or intermediary using its public-key or X.509 certificate to validate an encrypted MAC (Message Authentication Code) transported by a *SecurityEncap* option that we discuss later. An URI to the certificate is transported rather then the certificate itself, given the payload restrictions already discussed. When authenticating requestors using public-keys or certificates, the *SecurityToken* option must be sent in a CoAP message also transporting an encrypted MAC (signature). In order to support Kerberos-based authentication domains, a Kerberos ticket may identify and authorize CoAP requests. The support of Kerberos promotes compatibility with the AS server as employed to support LoWPAN authentication in the context of mediated transport-layer security, as discussed in the previous chapter. As with the *SecurityOn* option, a CoAP message may

transport more than one *SecurityToken* option, thus supporting multiple trust domains and intermediaries.

6.3.3 THE *SECURITYENCAP* COAP SECURITY OPTION

The final CoAP security option is the *SecurityEncap* option, which is illustrated in Figure 6.3. As previously discussed, this option transports the security-related data required for the processing of security for a given CoAP message, according to the contents of the *SecurityOn* and *Securitytoken* options. As for the previous option, only one of the variable-length fields in required for a given CoAP message. The length of this field is thus derived from the length of the option itself.

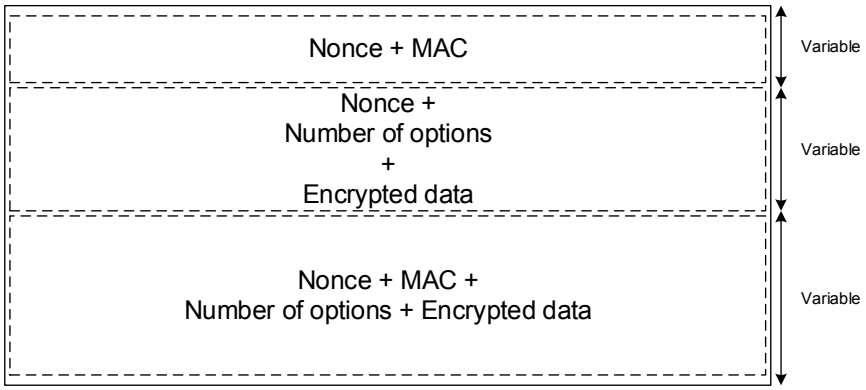

Figure 6.3 - SecurityEncap CoAP security option

When supporting sender authentication, replay protection and integrity for a CoAP message (in the *SecurityOn* option the *SecurityApplied* field value is 1) this option may be used to transport an encrypted MAC plus a *Nonce* value for freshness. If only encryption is required (the *SecurityApplied* value is 0 in the *SecurityOn* option) this option transports a *Nonce* value plus the number of options following in the encrypted part of the payload.

As all other options plus the CoAP packet payload are encrypted, the *Number of Options* field is transported as information helping in the processing of the message by a CoAP intermediary or final entity. In the last scenario, the CoAP message is fully protected and all security-related data is transported. The MAC value is computed using the hash or keyed hash algorithm associated with the security context negotiated by the communicating entities and identified in the *SecurityOn* option. The MAC value is computed considering the complete CoAP message plus the options, considering also the *SecurityEncap* option itself with the MAC value field set to all zeros.

6.3.4 DEFAULT COAP SECURITY USING AES/CCM

As we have previously analyzed in the context of network-layer and transport-layer security, the current proposals addressing security for 6LoWPAN environments are strongly based on the usage of AES/CCM, given its availability at the hardware in wireless sensing platforms supporting IEEE 802.15.4 [74] as the TelosB. As in our research solutions at the network and transport layers, AES/CCM in such platforms may also be employed to support security solutions at higher layers of the stack, via its employment in the standalone mode. Default CoAP security thus promotes the efficient support of application-layer security in existing IEEE 802.15.4 sensing platforms.

For the purpose of supporting CoAP security by default using AES/CCM in existing sensing platforms, we identify this mode with the value 1 and consider its employment when no specific security context has been negotiated by the CoAP communication entities. The usage of a default CoAP security mode may be of interest to simple applications employing key pre-configuration, or for the initial secure bootstrap of applications employing more complex context negotiation and key management mechanisms.

In the default security context, AES/CCM is employed with a 12-byte *nonce* value and an 8-byte MAC. This is also in line with the capabilities of current sensing platforms and with the usage of AES/CCM with TLS [128][134], thus promoting the design of cross-layer security mechanisms in the future, for example to support authentication and key management mechanisms for the transport and application layers simultaneously. We also consider that applications using the default security context may omit the *Destination Entity* identification on the *SecurityOn* option. This may be appropriate for applications where devices only answer for a default actor URI, while we must note that the final CoAP address is always part of the CoAP request.

6.4 EVALUATION OF COAP APPLICATION-LAYER MESSAGE SECURITY

As for the research proposals described in the previous chapters, we consider the experimental evaluation to be of particularly interest to investigate the impact of application-layer as previously proposed. Our experimental evaluation allows us to measure the energetic and computational impact of end-to-end security using CoAP security and DTLS. As our goal is to evaluate end-to-end security in the context of Internet-integrated sensing applications, we consider the usage of a CoAP client residing on an external Internet host and requesting resources from a CoAP server on a LoWPAN wireless sensing device, as illustrated in Figure 6.4. This integration model is also in line with the reference integration model previously discussed in Chapter 3 and illustrated in Figure 3.1.

As illustrated in Figure 6.4, we consider that end-to-end security may be achieved in a pure fashion either using DTLS at the transport-layer or in alternative with the proposed CoAP security options at the application-layer. In line with our reference integration model, we

also consider the usage of a CoAP intermediary (a forward proxy) in the processing of security. The security intermediary provides authorization of CoAP clients and control of accesses to resources on the LoWPAN via the *SecurityToken* CoAP option. We also consider the usage of AES/CCM cipher in the default CoAP security context, due on the one side to the availability of this cipher in the TelosB [194] experimental sensing platform, and on the other to guarantee a fair comparison of CoAP security against DTLS as currently proposed for the CoAP Protocol [34].

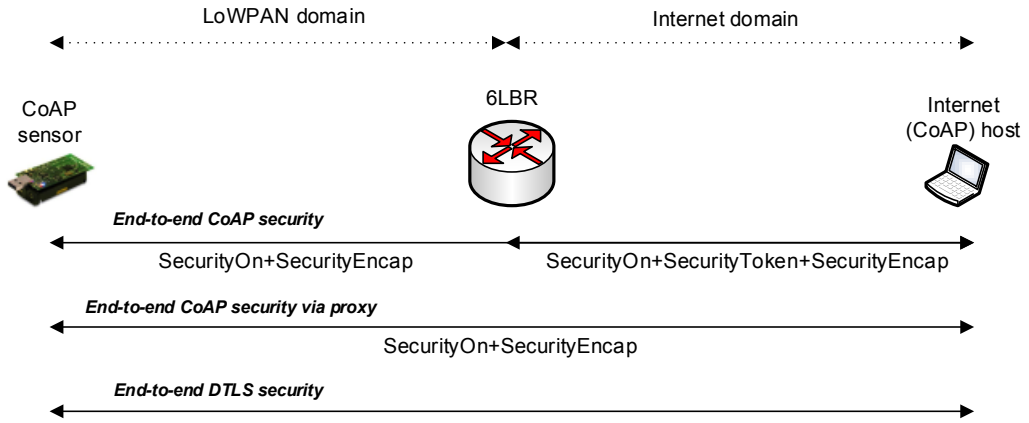

Figure 6.4 - CoAP and DTLS security end-to-end usage scenarios

In Figure 6.5 we illustrate the communications model considered for our experimental evaluation of security for end-to-end communications at the application-layer using the proposed CoAP security options. We also consider that the Access Control (AC) and Security Manager (SM) components of the reference model for end-to-end security are employed in the WSN gateway to support the management and selection of the application-layer security mode to employ for particular CoAP requests, as defined by appropriate access control rules in the context of particular application security profiles.

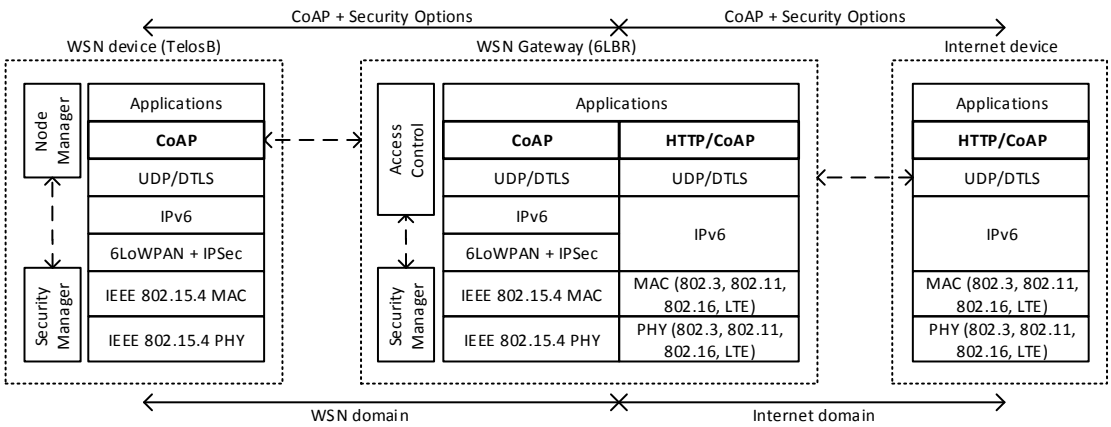

Figure 6.5 - Reference model for the evaluation of application-layer CoAP security

Also considering the model illustrated in Figure 6.5, on the WSN sensing device we consider that the SM supports the processing of application-layer security as appropriate to the sensing application and security context at hand. In our experimental evaluation we analyze the impact and feasibility of CoAP application-layer security as proposed in this chapter, in comparison with end-to-end DTLS security as currently proposed for the CoAP protocol [34]. We also investigate the feasibility of our proposal in respect to its impact on the payload space, since this is a critical aspect from the point of view of CoAP applications. The following evaluation also adopts the experimental evaluation framework discussed in Chapter 3, which we also employ in the previous chapters to evaluate the network and transport-layer security mechanisms.

6.4.1 IMPACT OF END-TO-END SECURITY ON COAP PACKET PAYLOAD SPACE

As in the evaluation of previous research solutions, we also consider the impact of application-layer CoAP security on the packet payload space available to applications. Our goal is to analyze if application-layer security leaves enough payload space to transport data from CoAP sensing applications without requiring costly fragmentations at the 6LoWPAN adaptation layer, since the proposed security options require costly space from the limited application-layer payload in a 6LoWPAN and CoAP packet. We thus must compare our proposal against the alternative usage of DTLS as currently adopted for CoAP in this respect.

In Figure 6.6 we illustrate the impact of security on the payload space available for CoAP applications in the presence of end-to-end security. The values illustrated are in percentage of the maximum available payload without security and correspond to the usage scenarios previously illustrated in Figure 6.4. The illustrated values enable us to compare CoAP application-layer security as proposed against the usage of end-to-end security using DTLS.

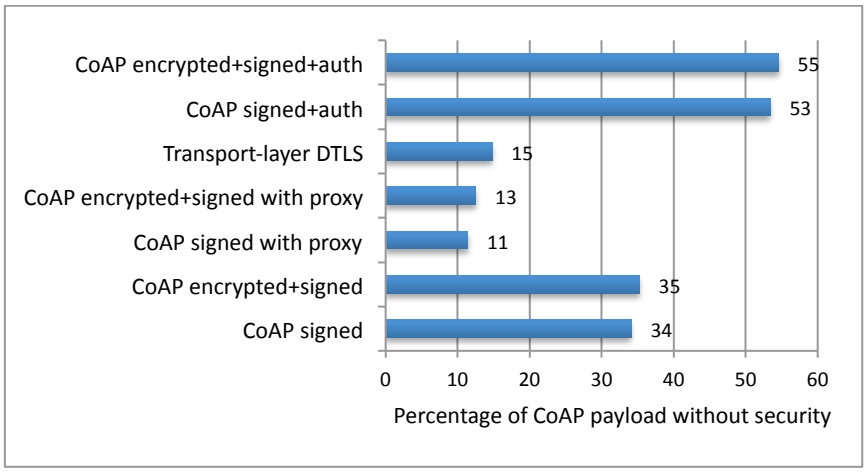

Figure 6.6 - Impact of end-to-end security on packet payload space available to CoAP

As we may observe in Figure 6.7, end-to-end security usage scenarios involving the participation of a CoAP security intermediary (or proxy) performs better than DTLS. The usage of a security intermediate thus provides the benefit of permitting the offloading of computationally-heavy computations to a more specialized entity, as considered with our transport-layer security proposal discussed in the previous chapter, while guaranteeing a very small impact on CoAP payload space.

The impact of end-to-end security without a proxy on CoAP packet payload space is greater, mostly due to the usage of the *Destination Entity* field in the *SecurityOn* option. We consider that this field requires an average of 20 bytes to transport the URI. Although the impact in this usage scenario is greater, in the worst case 65% of the original 6LoWPAN payload of 88 bytes is still available. Thus, we may consider that CoAP security is a viable approach for end-to-end security from the point of view of its impact on packet payload space.

6.4.2 IMPACT OF END-TO-END SECURITY ON THE LIFETIME OF SENSING APPLICATIONS

As energy is a critical resource on LoWPAN environments, it directly dictates the lifetime of wireless sensing applications and, in consequence, security mechanisms must be tested against its impact on energy. As in the experimental evaluation of the research solutions discussed in the previous chapters, this motivates our measurement of the impact of application-layer security on the energy of sensing platforms. In our experimental measurements, we obtained the energy consumption for security using experimental measurements of the voltage across a current sensing resistor placed in series with the battery pack and the circuit board of the TelosB experimental sensing platform.

From our experimental measurements, the energy required for the processing of a 102-byte 6LoWPAN message and related headers (including DTLS and CoAP security headers plus options) was measured as 0.007 nJ (Nano joules). The energy required for the processing of security using AES/CCM in standalone mode for a similar message was measured as 0.2 mJ (Micro joules), while the energy required for the transmission of a packet has been measured as 0.004 nJ (Nano joules) per bit. These experimentally obtained measurements enable us to predict the impact of end-to-end security on the lifetime of CoAP sensing applications.

From the values illustrated in Figure 6.6 we are able to obtain the maximum payload space that CoAP applications may employ without causing fragmentation at the 6LoWPAN adaptation layer. This corresponds to the usage scenario where end-to-end CoAP security performs encryption, integrity and authentication without the usage of a proxy, for which 45% of the original 6LoWPAN payload (or 40 bytes) is available to transport CoAP data. From this value we subtract 20 bytes required for the transportation of the security-related data (nonce and MAC values) for AES/CCM encryption.

Taking into account such considerations and the analytically determined values previously discussed we obtain the expectable lifetime for wireless sensing applications in the context

of Internet-integrated sensing applications, which we illustrate in Figure 6.7. For the obtainance of the illustrated values we assume the processing and transmission of two messages for each CoAP request, one containing a confirmable request and the other the corresponding reply carrying a piggybacked acknowledgment as defined in the protocol [34]. We also assume the usage of two new AA LR6-type batteries on the TelosB sensing platform, which provides a total of 6912 joules of energy from the start of our experimental measurements.

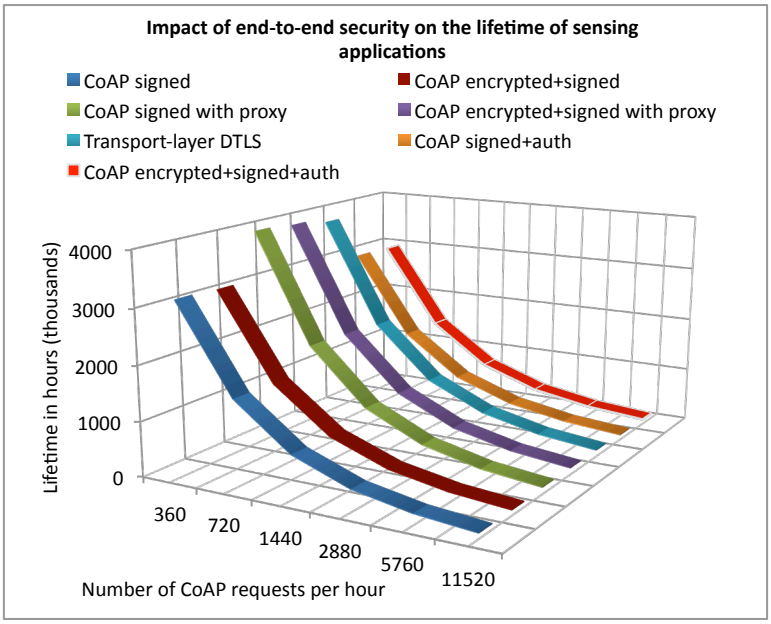

Figure 6.7 - Impact of end-to-end security on the lifetime of sensing applications

As in our previous analysis, the results illustrated in Figure 6.7 enables us to observe that end-to-end CoAP security performs better that DTLS when employing a security proxy providing support for the processing of the *SecurityToken* option. On the other hand, pure end-to-end CoAP security without a security intermediate causes a greater impact on the expected lifetime of sensing applications, particularly for lower communication rates where the cumulative impact of AES/CCM encryption is lower when compared to the impact of the energy required to process and transmit the proposed CoAP security options. Despite this observation, we may consider that CoAP security provides acceptable lifetime values in all usage scenarios, particularly considering WoT applications designed to require low or moderate wireless communications rates.

As previously discussed, one major motivation of the design of application-layer message security for CoAP is in the support of granular security policies. Security policies may define how each message must be protected, according to the semantics of the CoAP protocol, the type of message, its contents or particular requirements of the application. We also note that the definition of such aspects belongs in the context of application functional and

security profiles, which we consider as previously discussed in Chapter 3. For the purpose of the evaluation of granular security, we consider four usage scenarios of end-to-end security at the application layer, which in practice correspond to the four usage profiles considered in the experimental evaluation later in the chapter. Table 6.1 resumes the characteristics of the various security usage profiles considering in our following evaluation.

As Table 6.1 describes, the first security usage mode is for applications that require security only for the CoAP replies transporting sensorial data, and particularly the verification of the integrity of such messages. For such applications the confidentiality of the transported data is not a requirement, as long as it is protected against tampering or communication errors. As previously discussed, the transportation of an encrypted MAC plus a *Nonce* value in the context of the *SecurityEncap* CoAP security option supports integrity, replay protection and sender authentication.

Table 6.1 – Security usage modes for the evaluation of CoAP security

Security mode	Security properties provided	CoAP security options
CoAP signed (replies)	Integrity, replay protection, sender authentication	SecurityOn, SecurityEncap
CoAP encrypted and signed (replies)	Confidentiality, integrity, replay protection, sender authentication	SecurityOn, SecurityEncap
CoAP encrypted and signed with authentication (requests)	Confidentiality of identity and authorization data, integrity, replay protection, sender authentication	SecurityOn, SecurityToken, SecurityEncap
CoAP encrypted and signed (requests and replies)	Confidentiality, integrity, replay protection, sender authentication	SecurityOn, SecurityEncap

The second security usage profile described in Table 6.1 applies to applications requiring security also for CoAP reply messages, but in this case also protection against disclosure attacks. Thus, the CoAP reply messages must be also confidentiality protected using the *SecurityOn* and *SecurityEncap* security options. This mode may apply to CoAP messages transporting data that is of sensitive-nature, in the context of a given CoAP application.

A third usage scenario is that of applications requiring confidentiality and integrity, but in this case only for CoAP requests transporting authentication-related information. Such applications are thus concerned with the protection of identity and authorization data against disclosure or tampering attacks. In this case, we thus consider also the employment of the *SecurityToken* CoAP security option previously discussed.

The final security usage mode described in the previous table is for applications requiring confidentiality and integrity for all CoAP messages, irrespective of its type or contents. In this scenario, applications consider all messages to be of sensitive nature. This may be case for example of applications supporting critical environments, for example in industrial sensing and control environments.

In Figure 6.8 we illustrate the impact of end-to-end security according to the usage profiles previously described. For comparison purposes, we illustrate also the impact of pure end-to-end DTLS as currently proposed for CoAP. From the values illustrated in Figure 6.9 we are able to clearly observe the advantage of the granular security approach in what respects the lifetime of sensing applications, in comparison with transparent transport-layer security using DTLS, where this approach is unavailable. From Figure 6.8 we may also observe that the only security profile performing worst than DTLS is when CoAP security is employed to encrypt and sign all messages. This is due to the difference in terms of the payload space required to accommodate security. Despite this, in this scenario the expectable lifetime for applications is still acceptably large, even considering that applications require the application of security to many CoAP messages per hour.

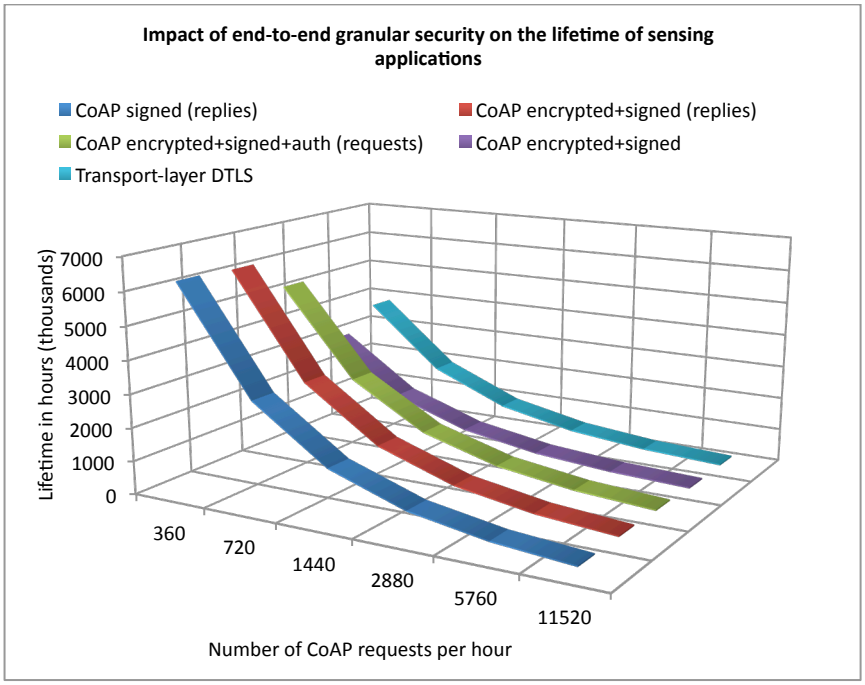

Figure 6.8 - Impact of (granular) end-to-end security on the lifetime of sensing applications

In conclusion, we are able to observe that our comparative analysis clearly illustrates the advantages of application-layer message security in protecting CoAP communications. When compared with DTLS, the proposed research solution introduces flexibility while providing security functionalities not possible when employing the transport-layer security approach, as previously discussed. The usage of security intermediaries participating in security also benefits energy and in consequence the lifetime of sensing applications. We also observe that even when CoAP security is employed to protect all messages as with DTLS, it still provides comparable performance.

6.4.3 IMPACT OF END-TO-END SECURITY ON THE COMMUNICATIONS RATE OF APPLICATIONS

Similarly to the experimental evaluation of our previous research proposals on network-layer and transport-layer security, we find it important to also analyze how CoAP security impacts on the maximum communications rate achievable by sensing applications. This analysis is also relevant at the application-layer, as it enables us to determine if security is a bottleneck in terms of communications, considering its impact on the payload space available to applications, which we have previously evaluated.

As addressed in our previous experimental evaluations, when considering wireless communications using IEEE 802.15.4 at 250Kbit/s we need to consider the overhead introduced by IEEE 802.15.4 on the bandwidth available for 6LoWPAN and upper protocols, which is of 19.6% of the total bandwidth, given that 25 bytes are required for link-layer information with each 127-byte 6LoWPAN packet. In Figure 6.9 we illustrate the maximum transmission rate achievable using DTLS versus the previously described CoAP security profiles. The values illustrated in this figure are obtained considering our experimental evaluation results and that CoAP transports an average of 20 bytes of payload data per message. We also consider the time required for the application of AES/CCM cryptography to CoAP messages, according to the security usage profiles.

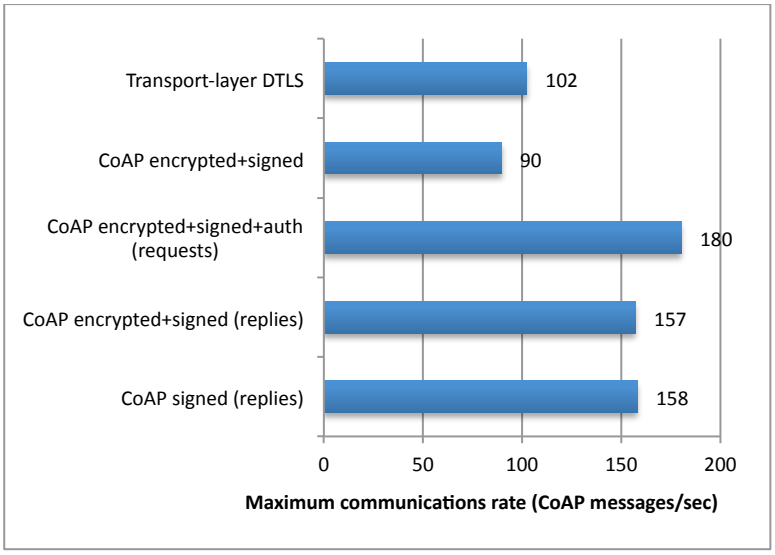

Figure 6.9 - Impact of end-to-end security in the communications rate of sensing applications

From the values illustrated in Figure 6.9, we may again observe the superior performance of the security profiles requiring the usage of granular application-layer security. CoAP signing and encryption of all messages (as with transport-layer security using DTLS) provides inferior performance, but despite this it still allows for 90 CoAP protected messages per second, a limit we may consider to be appropriate for many CoAP wireless sensing applications.

6.5 SUMMARY

As previously discussed, one major motivation of the research solutions described throughout the thesis is that secure end-to-end communications in the context of Internet-integrated WSN may provide an important contribution to enable future sensing applications on the IoT, as many of such applications may benefit from the availability of direct and secure end-to-end communications between Internet hosts and constrained sensing devices. The research proposal described and evaluated in the present chapter provides a contribution in this context, particularly by addressing the design of security at the CoAP application-layer communications protocol, rather than considering security to be a transparent end-to-end approach, as in our previous approaches to network-layer and transport-layer security.

With this goal in mind, we propose the introduction of new options to support security in the CoAP Protocol. Such options are designed with the goal of supporting granular security and the transversal of multiple WSN (trust and administrative) domains, by providing support for the usage of multiple security contexts and security mechanisms by a single CoAP application. We note that this approach is unachievable with DTLS, in the context of which a security session is negotiated between the two CoAP communicating entities in order to transparently and uniformly protect all CoAP communications.

Our experimental evaluation of CoAP security allowed us to observe that CoAP application-layer security may perform similarly or better than transport-layer security as currently proposed for CoAP. We may also note that the integration of security in the context of the application-layer communications protocol may provide added benefits and motivate further research efforts on the subject, as we discuss in Chapter 7. It is our purpose that application-layer security as proposed may enrich the set of solutions available to Internet-integrated WSN, by supporting applications requiring granular security policies, extensible authentication mechanisms and the secure transversal of different trust domains.

7 CONCLUSIONS AND FUTURE RESEARCH CHALLENGES

In this chapter we conclude our discussion by summing up the research proposals previously analyzed and evaluated, as well as by identifying research opportunities and challenges. As previously discussed, our research proposals offer solutions to the problem of enabling end-to-end security for Internet communications involving 6LoWPAN-based sensing devices, according to different strategies. We target end-to-end security at the network, transport and application layers with different and complementary goals. Many challenges remain certainly to be addressed in order to fully support the Integration of WSN with the Internet with complete security, which also motivate future research efforts.

7.1 CONCLUSIONS

In this thesis we have proposed research solutions targeting the problem of supporting effective and robust security for end-to-end communications between constrained sensing devices and external (Internet) entities, thus in the context of sensing applications employing Internet-integrated WSN. As we have observed throughout the various chapters of the thesis, end-to-end security can be effectively supported according to different strategies. Security can also be applied according to the specific security and functional requirements of applications, which may be appropriately formalized in application profiles. Such requirements may on the other hand map to particular security mechanisms at different layers of the stack, with different approaches to how end-to-end security may be achieved, also to adapt end-to-end security mechanisms to the capabilities of the sensing devices employed by the application.

Regarding end-to-end security at the network-layer, our approach was to design new compressed security headers for the 6LoWPAN adaptation layer [183]–[185][202]. One aspect considered throughout the design of the various research proposals in the thesis was the support of default security using AES/CCM encryption, in line with its availability at the hardware in sensing platforms supporting the IEEE 802.15.4 MAC, which also provides the support for the various 6LoWPAN-based communication technologies forming the reference integration architecture discussed in Chapter 3.

Regarding transport-layer security, in Chapter 5 we have addressed the problem of supporting ECC-based security procedures in an efficient manner using constrained sensing devices. We propose and evaluate mechanisms for the transparent interception and mediation of the DTLS authentication and key agreement handshake. As we have observed, this phase of end-to-end transport-layer security is problematic to support using constrained sensing devices with the characteristics of the TelosB. The mediation of the handshake by an intermediate entity also supports the delegation of ECC-based security from constrained sensing devices to more powerful entities. Ours was the first research proposal with such goals in mind [202][203][204], and provides an effective and practical alternative to the

employment of end-to-end DTLS as currently adopted for CoAP. The proposed mechanisms are able to offer effective end-to-end security using DTLS in a totally transparent fashion from the perspective of the CoAP communicating entities, while laying the ground for the future adoption of other security mechanisms based on a security gateway. This is the case of mechanisms involving the examination of encrypted CoAP communications, which would otherwise be impossible to support in the presence of full end-to-end encrypted communications using DTLS.

Our approach to security at the application-layer complements the previous research solutions by enabling security in the context of the application-layer communications protocol itself. Our proposal is discussed in Chapter 6 [203][211] and focus on the introduction of security in the context of the CoAP application-layer protocol, rather than by modifying or optimizing the DTLS protocol as in alternative approaches. As discussed in the previous chapter, this approach provides various advantages, among which is the support for the transversal of trust and administrative domains using different identification and authentication mechanisms, as well as the support of granular security policies by CoAP sensing applications.

As previously discussed, the diversity of the approaches and solutions for end-to-end security considered in the proposals analyzed throughout the thesis seeks to contribute to the effective and secure integration of WSN with the Internet, as mechanisms with such characteristics may be part of a future Internet security architecture encompassing communications in which Internet-integrated sensing devices participate. As previously discussed, 6LoWPAN-based communication technologies offer the promise of extending Internet communications to WSN domains, and in consequence appropriate security mechanisms will be required to secure Internet communications with devices in such domains. The security mechanisms proposed and experimentally evaluated throughout the thesis are designed with this in mind, and considering that most of the sensing applications currently envisioned for the IoT will require appropriate security mechanisms as a fundamental enabling factor.

7.2 RESEARCH CHALLENGES AND FUTURE WORK

In the light of the research solutions proposed in the previous chapters of the thesis, and also considering the reference model for end-to-end security discussed in Chapter 3, we are able to identify various research challenges that may motivate future research efforts. As end-to-end Internet communications with sensing devices are enabled by 6LoWPAN and related communication technologies, other approaches can be developed offering new solutions to address end-to-end security, and also targeting security aspects that may be addressed with cross-layer approaches, as we proceed to discuss.

Regarding security in the context of 6LoWPAN, other mechanisms will be required in addition to the proposed compressed security headers, in order to fully support

network-layer end-to-end secure communications with 6LoWPAN-enabled sensing devices. This is the case of mechanisms enabling the integration of 6LoWPAN security with existing implementations of the Internet security architecture, which may involve the design of mechanisms to manage security associations and security policies, as well as the related security associations and security policies databases, which may be properly adapted to support end-to-end secure communications with 6LoWPAN devices.

As for transport-layer security, other than the proposed mechanisms for the transparent interception and mediation of the DTLS handshake, research and standardization is also targeting the optimization or profiling of DTLS to appropriately support constrained sensing platforms, as previously discussed. Our strategy of delegating costly operations to more powerful entities may also support further security solutions and mechanisms, for example to support security for inactive devices or secure communications with groups of devices, as will be required for many applications employing multicast addressing and communication.

As discussed in Chapter 5, our reference integration model also supports the future design of mechanisms supporting transparent end-to-end security for sensing devices moving between different LoWPAN domains. The proposed LoWPAN authentication protocol may also support further security approaches to protect communications between the 6LBR and CoAP sensing devices in the context of end-to-end transport-layer security. A challenge motivating further work, not only at the transport-layer but also at the other layers of the stack, is the design of techniques to decide on the most appropriate end-to-end security mode in the presence of particular sensing platforms and applications described by appropriate functional or security profiles.

One particularly interesting advantage of the proposed mediated DTLS handshake is that the 6LBR is also able to learn the pre-master secret key and random values for a given DTLS security session. This enables the security gateway to also compute the final master key and to subsequently derive the required keying material. The knowledge of this security material provides the ground for the employment of other security mechanisms based on traffic filtering and analysis, particularly to examine and filter the contents of application-layer CoAP communications. This functionality may for example support new intrusion detection mechanisms to detect and filter application-layer attacks against CoAP sensing devices on the LoWPAN domain.

Regarding CoAP security, our proposal addresses a complementary approach to how security is current being considered to protect application-layer communications with this protocol, as discussed in the previous chapter. Our approach also leaves the door open to the adoption of further security options in the future, in order to support other security requirements, new authentication mechanisms and particular security policies.

Other than the mechanisms designed in the context of the various communication protocols, many cross-layer security aspects may be targeted by future mechanisms not necessarily related with a particular protocol layer, as is the case on intrusion detection,

traffic control and key management. Regarding intrusion detection, we observe that the proposed mechanisms can provide useful support for the design of solutions appropriate to Internet-integrated WSN. As previously observed, the mediation of the DTLS handshake provides the security gateway with the keying material required to interpret encrypted communications after the completion of the DTLS handshake. This implies that intrusion detection can benefit from the analysis of DTLS communications, enabling for example the detection of internal attackers sending CoAP messages to victim nodes. A similar reasoning can be applied to security at the application layer using CoAP, as previously discussed. The compatibility of CoAP security with the employment of intermediaries in reverse and forward modes provides the opportunity to employ such intermediaries also as strategic security-enforcement devices, in what regards the support of security at the application-layer.

Key management is also certainly a fundamental aspect of security, and one that will require immediate attention in order to promote the usefulness of the research solutions proposed in the thesis, as any proposal based on the employment of cryptographic protocols requires the periodic refreshment of the keys supporting security. One approach we intend to pursuit in future research efforts is the design of a cross-layer security solution supporting key negotiation and refreshment for end-to-end security, which may provide support for the various research solutions at the network, transport and application-layers. Despite the existence of the previously analyzed proposals addressing the simplification of IKE for constrained communication environments, key management may be addressed in a cross-layer fashion by designing mechanisms that support communication technologies at different protocol layers.

REFERENCES

[1] T. O'Donovan, J. Brown, U. Roedig, C. J. Sreenan, J. do Ó, A. Dunkels, A. Klein, J. S. Silva, V. Vassiliou, and L. Wolf, "GINSENG: Performance Control in Wireless Sensor Networks," in *2010 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, 2010, pp. 1–3.

[2] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Communications Surveys Tutorials*, vol. 10, no. 3, pp. 6–28, 2008.

[3] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.

[4] E. De Cristofaro, J.-M. Bohli, and D. Westhoff, "FAIR: fuzzy-based aggregation providing in-network resilience for real-time wireless sensor networks," in *Proceedings of the second ACM conference on Wireless network security*, New York, NY, USA, 2009, pp. 253–260.

[5] H. Chan, A. Perrig, B. Przydatek, and D. Song, "SIA: Secure information aggregation in sensor networks," *J. Comput. Secur.*, vol. 15, no. 1, pp. 69–102, Jan. 2007.

[6] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, Dec. 2002.

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[8] A. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[9] G. Gu and G. Peng, "The survey of GSM wireless communication system," in *2010 International Conference on Computer and Information Application (ICCIA)*, 2010, pp. 121–124.

[10] A. Wood, J. A. Stankovic, and S. H. Son, "JAM: a jammed-area mapping service for sensor networks," in *24th IEEE Real-Time Systems Symposium, 2003. RTSS 2003*, 2003, pp. 286–297.

[11] M. Anand, E. Cronin, M. Sherr, M. Blaze, Z. Ives, and I. Lee, "Sensor network security: more interesting than you think," in *Proceedings of the 1st USENIX Workshop on Hot Topics in Security*, Berkeley, CA, USA, 2006, pp. 5–5.

[12] M. Shaneck, K. Mahadevan, V. Kher, and Y. Kim, "Remote Software-Based Attestation for Wireless Sensors," in *Security and Privacy in Ad-hoc and Sensor Networks*, R. Molva, G. Tsudik, and D. Westhoff, Eds. Springer Berlin Heidelberg, 2005, pp. 27–41.

[13] “Trusted Computing Group,” *Trusted Computing Group - Home*. [Online]. Available: http://www.trustedcomputinggroup.org. [Accessed: 15-Apr-2014].

[14] P. England, B. Lampson, J. Manferdelli, and B. Willman, “A trusted open platform,” *Computer*, vol. 36, no. 7, pp. 55–62, 2003.

[15] T. Roosta, “Probabilistic geographic routing protocol for ad hoc and sensor networks,” in *in International Workshop on Wireless Ad-hoc Networks (IWWAN*, 2005.

[16] J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[17] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” in *2003 IEEE International Workshop on Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE*, 2003, pp. 113–127.

[18] Y. Hu, A. Perrig, and D. B. Johnson, “Wormhole attacks in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 370–380, 2006.

[19] R. Poovendran and L. Lazos, “A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks,” *Wirel. Netw.*, vol. 13, no. 1, pp. 27–59, Jan. 2007.

[20] R. M. Needham and M. D. Schroeder, “Using encryption for authentication in large networks of computers,” *Commun. ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978.

[21] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: a secure on-demand routing protocol for ad hoc networks,” *Wirel. Netw.*, vol. 11, no. 1–2, pp. 21–38, Jan. 2005.

[22] Y. G. Iyer, S. Gandham, and S. Venkatesan, “STCP: a generic transport layer protocol for wireless sensor networks,” in *14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005. Proceedings*, 2005, pp. 449–454.

[23] J. Newsome, E. Shi, D. Song, and A. Perrig, “The sybil attack in sensor networks: analysis & defenses,” in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, New York, NY, USA, 2004, pp. 259–268.

[24] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” in *Proceedings of the 10th ACM conference on Computer and communications security*, New York, NY, USA, 2003, pp. 42–51.

[25] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proceedings of the 9th ACM conference on Computer and communications security*, New York, NY, USA, 2002, pp. 41–47.

[26] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *Proceedings of the 10th ACM conference on Computer and communications security*, New York, NY, USA, 2003, pp. 52–61.

[27] S. Zhu, S. Setia, and S. Jajodia, “LEAP+: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 4, pp. 500–528, 2006.

[28] R. Di Pietro, L. V. Mancini, and A. Mei, “Efficient and resilient key discovery based on pseudo-random key pre-deployment,” in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, 2004.

[29] E. Shi and A. Perrig, “Designing secure sensor networks,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[30] B. Arazi, I. Elhanany, O. Arazi, and H. Qi, “Revisiting public-key cryptography for wireless sensor networks,” *Computer*, vol. 38, no. 11, pp. 103–105, 2005.

[31] G. Gaubatz, J. Kaps, and B. Sunar, “Public key cryptography in sensor networks - revisited,” in *In 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, 2004, pp. 2–18.

[32] D. Liu and P. Ning, “Multilevel µTESLA: Broadcast authentication for distributed sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 4, pp. 800–836, Nov. 2004.

[33] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, “SPINS: security protocols for sensor networks,” *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[34] Z. Shelby, K. Hartke, and C. Bormann, “Constrained Application Protocol (CoAP),” *draft-ietf-core-coap-18 (work in progress)*. [Online]. Available: https://tools.ietf.org/html/draft-ietf-core-coap-18. [Accessed: 15-Apr-2014].

[35] P. Michiardi and R. Molva, “Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks,” in *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, Deventer, The Netherlands, The Netherlands, 2002, pp. 107–121.

[36] S. Buchegger and J.-Y. Le Boudec, “Performance analysis of the CONFIDANT protocol,” in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, New York, NY, USA, 2002, pp. 226–236.

[37] S. Ganeriwal and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” in *In SASN ’04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, 2004, pp. 66–77.

[38] S. Oh, S. Russell, and S. Sastry, “Markov chain Monte Carlo data association for general multiple-target tracking problems,” in *43rd IEEE Conference on Decision and Control, 2004. CDC*, 2004, vol. 1, pp. 735–742 Vol.1.

[39] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proceedings of the 6th*

annual international conference on Mobile computing and networking, New York, NY, USA, 2000, pp. 56–67.

[40] D. Wagner, "Resilient aggregation in sensor networks," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, New York, NY, USA, 2004, pp. 78–87.

[41] T. Dimitriou and D. Foteinakis, "Secure and efficient in-network processing for sensor networks," in *In First Workshop on Broadband Advanced Sensor Networks (BaseNets)*, 2004.

[42] J. Deng, R. Han, and S. Mishra, "Security support for in-network processing in Wireless Sensor Networks," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, New York, NY, USA, 2003, pp. 83–93.

[43] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, New York, NY, USA, 2003, pp. 138–149.

[44] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA, 2004, pp. 39–49.

[45] H. Vogt, M. Ringwald, and M. Strasser, "Intrusion Detection and Failure Recovery in Sensor Nodes," in *In Tagungsband INFORMATIK 2005, Workshop Proceedings, LNCS*, 2005.

[46] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *3rd IEEE Consumer Communications and Networking Conference, 2006. CCNC 2006*, 2006, vol. 1, pp. 640–644.

[47] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized Intrusion Detection in Wireless Sensor Networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks (Q2SWINET'05)*, 2005, pp. 16–23.

[48] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *(WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005*, 2005, vol. 3, pp. 253–259 Vol. 3.

[49] I. Demirkol, F. Alagoz, H. Deliç, and C. Ersoy, "Wireless sensor networks for intrusion detection: packet traffic modeling," *IEEE Communications Letters*, vol. 10, no. 1, pp. 22–24, 2006.

[50] "ZigBee Alliance, Specifications," *ZigBee Specifications*. [Online]. Available: http://www.zigbee.org/Specifications.aspx. [Accessed: 15-Apr-2014].

[51] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu, “Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards,” *Computer Communications*, vol. 30, no. 7, pp. 1655–1695, May 2007.

[52] D. Sturek, “ZigBee IP stack overview,” *ZigBee IP stack overview*, 2009. [Online]. Available: https://docs.zigbee.org/zigbee-docs/dcn/09-5375.pdf. [Accessed: 15-Apr-2014].

[53] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An Operating System for Sensor Networks,” in *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer Berlin Heidelberg, 2005, pp. 115–148.

[54] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, *The TESLA Broadcast Authentication Protocol*. 2002.

[55] C. Karlof, N. Sastry, and D. Wagner, “TinySec: a link layer security architecture for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA, 2004, pp. 162–175.

[56] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An Operating System for Sensor Networks,” in *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer Berlin Heidelberg, 2005, pp. 115–148.

[57] “GainSpan Unveils Industry First Wi-Fi® and ZigBee® IP Single Chip,” *GainSpan - Low Power Wi-Fi Modules and Embedded Software*, Feb-2013. [Online]. Available: http://www.gainspan.com/news/news_20130226_gs2000. [Accessed: 15-Apr-2014].

[58] “IEEE 802.3 Ethernet Working Group,” *IEEE 802.3 Ethernet*. [Online]. Available: http://www.ieee802.org/3/. [Accessed: 15-Apr-2014].

[59] “OTN - SDH & SONET Related Recommendations and Standards,” *OTN - SDH & SONET Related Recommendations and Standards*. [Online]. Available: http://www.itu.int/ITU-T/2001-2004/com15/otn/SDH-rec.html. [Accessed: 15-Apr-2014].

[60] “ETSI, the European Telecommunications Standards Institute,” *ETSI, the European Telecommunications Standards Institute*. [Online]. Available: http://www.etsi.org/. [Accessed: 15-Apr-2014].

[61] S. Chia, “The Universal Mobile Telecommunication System,” *IEEE Communications Magazine*, vol. 30, no. 12, pp. 54–62, 1992.

[62] “The 3rd Generation Partnership Project (3GPP),” *3GPP*. [Online]. Available: http://www.3gpp.org/. [Accessed: 15-Apr-2014].

[63] “3GPP - LTE Advanced,” *LTE-Advanced*. [Online]. Available: http://www.3gpp.org/lte-advanced. [Accessed: 15-Apr-2014].

[64] "IEEE Standards Association: IEEE 802.11™: Wireless LANs," *IEEE-SA -IEEE Get 802 Program - 802.11: Wireless LANs*. [Online]. Available: http://standards.ieee.org/about/get/802/802.11.html. [Accessed: 15-Apr-2014].

[65] "IEEE Standards Association: IEEE 802.16™: Broadband Wireless Metropolitan Area Networks (MANs)," *IEEE-SA -IEEE Get 802 Program - 802.16: Broadband Wireless MANs*. [Online]. Available: http://standards.ieee.org/about/get/802/802.16.html. [Accessed: 15-Apr-2014].

[66] S. Aust, R. V. Prasad, and I. G. M. M. Niemegeers, "IEEE 802.11ah: Advantages in standards and further challenges for sub 1 GHz Wi-Fi," in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 6885–6889.

[67] "IEEE Standards Association: IEEE 802.15™: Wireless Personal Area Networks (PANs)," *IEEE-SA -IEEE Get 802 Program - 802.15: Wireless PANs*. [Online]. Available: http://standards.ieee.org/about/get/802/802.15.html. [Accessed: 15-Apr-2014].

[68] K.-S. Kwak, S. Ullah, and N. Ullah, "An overview of IEEE 802.15.6 standard," in *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, 2010, pp. 1–6.

[69] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," *RFC4919*, Aug. 2007.

[70] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15. 4 networks," *RFC 4944*, Sep. 2007.

[71] J. Hui and P. Thubert, "Compression format for IPv6 datagrams over IEEE 802.15. 4-based networks," *RFC 6282*, Set 2011.

[72] "ITU Telecommunication Standardization Sector," *ITU: Commited to connecting the world*. [Online]. Available: http://www.itu.int/en/Pages/default.aspx. [Accessed: 15-Apr-2014].

[73] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[74] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011.

[75] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, 2012.

[76] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kesley, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: Ipv6 routing protocol for low-power and lossy networks," *RFC 6550*, Mar. 2012.

[77] "Sensinode," *Sensinode Home - Sensinode Ltd*. [Online]. Available: http://www.sensinode.com/. [Accessed: 15-Apr-2014].

[78] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: enabling sustainable and scalable outdoor wireless sensor network deployments," in *Proceedings of the 5th international conference on Information processing in sensor networks*, New York, NY, USA, 2006, pp. 407–415.

[79] W. Youssef and M. Younis, "Intelligent Gateways Placement for Reduced Data Latency in Wireless Sensor Networks," in *IEEE International Conference on Communications, 2007. ICC '07*, 2007, pp. 3805–3810.

[80] S. Duquennoy, G. Grimaud, and J.-J. Vandewalle, "The Web of Things: Interconnecting Devices with High Usability and Performance," in *International Conference on Embedded Software and Systems, 2009. ICESS '09*, 2009, pp. 323–330.

[81] D. Yazar and A. Dunkels, "Efficient application integration in IP-based sensor networks," in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, New York, NY, USA, 2009, pp. 43–48.

[82] R. Dickerson, J. Lu, J. Lu, and K. Whitehouse, "Stream Feeds - An Abstraction for the World Wide Sensor Web," in *The Internet of Things*, C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern, and S. E. Sarma, Eds. Springer Berlin Heidelberg, 2008, pp. 360–375.

[83] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, New York, NY, USA, 2006, pp. 307–320.

[84] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)*, pp. 1–203, 2007.

[85] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 3: Alternative Physical Layer Extension to support the Japanese 950 MHz bands," *IEEE Std 802.15.4d-2009 (Amendment to IEEE Std 802.15.4-2006)*, pp. c1–27, 2009.

[86] I. ISA, “100.11 a-2009: Wireless Systems for Industrial Automation: Process Control and Related Applications,” *International Society of Automation: Research Triangle Park, NC, USA*, 2009.

[87] A. N. Kim, F. Hekland, S. Petersen, and P. Doyle, “When HART goes wireless: Understanding and implementing the WirelessHART standard,” in *IEEE International Conference on Emerging Technologies and Factory Automation, 2008. ETFA 2008*, 2008, pp. 899–907.

[88] K. S. J. Pister and L. Doherty, “TSMP: Time Synchronized Mesh Protocol,” presented at the IASTED Distributed Sensor Networks, Orlando, Florida, USA, 2008, pp. 391–398.

[89] N. Aes, “Advanced encryption standard,” *Federal Information Processing Standard, FIPS-197*, p. 12, 2001.

[90] T. Instruments, “2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver,” 2007. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf. [Accessed: 15-Apr-2014].

[91] E. Kim and D. Kaspar, “Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs),” *RFC 6568 - Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks*, 2012. [Online]. Available: http://tools.ietf.org/html/rfc6568.txt. [Accessed: 15-Apr-2014].

[92] C. Gomez, E. Kim, D. Kaspar, and C. Bormann, “Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing,” *RFC 6606 - Problema Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing*, 2012. [Online]. Available: http://tools.ietf.org/html/rfc6606. [Accessed: 15-Apr-2014].

[93] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, “Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs),” *RFC 6775*, Nov. 2012.

[94] J. Hui and D. Culler, “Stateless IPv6 Header Compression for Globally Routable Packets in 6LoWPAN Subnetworks,” *draft-hui-6lowpan-hc1g-00*, Jun-2007. .

[95] J. Nieminen, B. Patil, T. Savolainen, M. Isomaki, Z. Shelby, and C. Gomez, “Transmission of IPv6 packets over bluetooth low energy,” *draftietf-6lowpan-btle-12 (work-in-progress)*, Nov-2013. .

[96] P. Mariager and J. Petersen, “Transmission of IPv6 Packets over DECT Ultra Low Energy,” 2012.

[97] A. Brandt and J. Buron, “Transmission of IPv6 packets over ITU-T G.9959 Networks,” *draft-brandt-6man-lowpanz-02*. [Online]. Available: http://tools.ietf.org/html/draft-brandt-6man-lowpanz-02. [Accessed: 15-Apr-2014].

[98] D. Trček, “Lightweight protocols and privacy for all-in-silicon objects,” *Ad Hoc Networks*, vol. 11, no. 5, pp. 1619–1628, Jul. 2013.

[99] S. Kent and K. Seo, “Security architecture for the internet protocol 2005,” *RFC 4301*, Dec. 2005.

[100] S. Kent, “IP Encapsulating Security Payload (ESP),” *RFC 4303*, Dec. 2005.

[101] S. Kent, “IP Authentication Header (AH),” *RFC 3202*, Dec. 2005.

[102] T. Narten, E. Nordmark, and W. Simpson, “Neighbor Discovery for IP version 6 (IPv6),” *RFC 4861*, Sep. 2007.

[103] J. Arkko, J. Kempf, B. Zill, and P. Nikander, “SEcure Neighbor Discovery (SEND),” *RFC 3971*, Mar. 2005.

[104] T. Aura, “Cryptographically Generated Addresses (CGA),” *RFC 3972*.

[105] T. K. <kivinen@iki.fi>, “Minimal IKEv2,” *draft-kivinen-ipsecme-ikev2-minimal-01*. [Online]. Available: http://tools.ietf.org/html/draft-kivinen-ipsecme-ikev2-minimal-01. [Accessed: 15-Apr-2014].

[106] S. Raza, T. Voigt, and V. Jutvik, “Lightweight IKEv2: A Key Management Solution for both the Compressed IPsec and the IEEE 802.15. 4 Security,” in *Proceedings of the IETF Workshop on Smart Object Security*, 2012.

[107] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, “Key management systems for sensor networks in the context of the Internet of Things,” *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 147–159, Mar. 2011.

[108] K.-H. Kim, W. Haddad, J. Laganier, S. Park, and S. Chakrabarti, “IPv6 over Low Power WPAN Security Analysis,” *draft-daniel-6lowpan-security-analysis-05*. [Online]. Available: http://tools.ietf.org/html/draft-daniel-6lowpan-security-analysis-05. [Accessed: 15-Apr-2014].

[109] R. Riaz, K.-H. Kim, and H. F. Ahmed, “Security analysis survey and framework design for IP connected LoWPANs,” in *International Symposium on Autonomous Decentralized Systems, 2009. ISADS ’09*, 2009, pp. 1–6.

[110] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. John Wiley & Sons, 2011.

[111] H. Kim, “Protection Against Packet Fragmentation Attacks at 6LoWPAN Adaptation Layer,” in *International Conference on Convergence and Hybrid Information Technology, 2008. ICHIT ’08*, 2008, pp. 796–801.

[112] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, “6LoWPAN fragmentation attacks and mitigation mechanisms,” in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, New York, NY, USA, 2013, pp. 55–66.

[113] T. Winter, A. B. P. Thubert, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, “RPL: IPv6 Routing Protocol for Low Power and Lossy Networks,(RFC 6550),” *IETF ROLL WG, Tech. Rep*, 2012.

[114] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, “Routing Requirements for Urban Low-Power and Lossy Networks,” *RFC 5548*, May 2009.

[115] K. Pister, P. Thubert, S. Dwars, and T. Phinney, “Industrial Routing Requirements in Low-Power and Lossy Networks,” *RFC 5673*, Oct. 2009.

[116] A. Brandt, J. Buron, and G. Porcu, “Home Automation Routing Requirements in Low-Power and Lossy Networks,” *RFC 5826*, Apr. 2010.

[117] J. Martocci, P. Mil, N. Riou, and W. Vermeylen, “Building Automation Routing Requirements in Low-Power and Lossy Networks,” *RFC 5867*.

[118] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, “Routing metrics used for path calculation in low power and lossy networks,” *RFC 6551*, Mar. 2011.

[119] M. Richardson, A. Lozano, T. Tsao, V. Daza, R. Alexander, and M. Dohler, “A Security Threat Analysis for Routing over Low-Power and Lossy Networks,” *draft-ietf-roll-security-threats-06 (work in progress)*. [Online]. Available: http://tools.ietf.org/html/draft-ietf-roll-security-threats-06. [Accessed: 15-Apr-2014].

[120] I. ISO, “7498-2. Information Processing Systems Open Systems Interconnection Basic Reference Model-Part 2: Security Architecture,” *ISO Geneva, Switzerland*, 1989.

[121] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, “The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3685–3692, 2013.

[122] A. Dvir, T. Holczer, and L. Buttyan, “VeRA - Version Number and Rank Authentication in RPL,” in *2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2011, pp. 709–714.

[123] L. Dora, A. Dvir, L. Buttyan, and T. Holczer, “Version Number and Rank Authentication,” *draft-dvir-roll-security-authentication-01*. [Online]. Available: http://tools.ietf.org/html/draft-dvir-roll-security-authentication-01. [Accessed: 15-Apr-2014].

[124] K. Weekly and K. Pister, “Evaluating sinkhole defense techniques in RPL networks,” in *2012 20th IEEE International Conference on Network Protocols (ICNP)*, 2012, pp. 1–6.

[125] J. Postei, “User Datagram Protocol,” *RFC 768*, Aug. 1980.

[126] J. Postel, “Transmission control protocol,” *RFC 793*, Sep. 1981.

[127] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security,” *RFC 4347*, Apr. 2006.

[128] T. Dierks and E. Rescorla, “The transport layer security (TLS) protocol,” *RFC 5246*, Aug. 2008.

[129] T. Zheng, A. Ayadi, and X. Jiang, “TCP over 6LoWPAN for Industrial Applications: An Experimental Study,” in *2011 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2011, pp. 1–4.

[130] T. Braun, T. Voigt, and A. Dunkels, “TCP support for sensor networks,” in *Fourth Annual Conference on Wireless on Demand Network Systems and Services, 2007. WONS ’07*, 2007, pp. 162–169.

[131] K. Hartke and O. Bergmann, “Datagram Transport Layer Security in Constrained Environments,” *raft-hartke-core-codtls-02*. [Online]. Available: http://tools.ietf.org/html/draft-hartke-core-codtls-02. [Accessed: 15-Apr-2014].

[132] P. Wouters, H. Tschofenig, J. Gilmore, and T. Kivinen, “Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS),” *draft-ietf-tls-oob-pubkey-11 (work in progress)*. [Online]. Available: http://tools.ietf.org/html/draft-ietf-tls-oob-pubkey-11. [Accessed: 15-Apr-2014].

[133] G. Seroussi, “Elliptic curve cryptography,” in *Information Theory and Networking Workshop, 1999*, 1999, p. 41.

[134] D. Bailey and D. McGrew, “AES-CCM Cipher Suites for Transport Layer Security (TLS),” *RFC 6655*, Jul. 2012.

[135] S. Blake-Wilson, B. Moeller, V. Gupta, C. Hawk, and N. Bolyard, “Elliptic curve cryptography (ECC) cipher suites for transport layer security (TLS),” *RFC 4492*, May 2006.

[136] “ETSI IoT CoAP Plugtests,” *IoT CoAP Plugtests*. [Online]. Available: http://www.etsi.org/plugtests/coap/coap.htm. [Accessed: 14-Apr-2014].

[137] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,” *RFC 6960*, Jun. 2013.

[138] D. Eastlake, “Transport Layer Security (TLS) Extensions: Extension Definitions,” *RFC 6066*, Jan. 2011.

[139] O. Garcia-Morchon, S. Kumar, R. Struik, S. Keoh, and R. Hummen, “Security Considerations in the IP-based Internet of Things,” *draft-garcia-core-security-06*. [Online]. Available: http://tools.ietf.org/html/draft-garcia-core-security-06. [Accessed: 15-Apr-2014].

[140] M. Brachmann, S. L. Keoh, O. G. Morchon, and S. S. Kumar, “End-to-End Transport Security in the IP-Based Internet of Things,” in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, 2012, pp. 1–5.

[141] S. Keoh, O. Garcia-Morchon, S. Kumar, and S. Dijk, “DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs),” *draft-keoh-tls-multicast-security-00 (work in progress)*, Oct-2012. [Online]. Available: http://tools.ietf.org/html/draft-keoh-tls-multicast-security-00. [Accessed: 15-Apr-2014].

[142] K. H. <hartke@tzi.org>, “Practical Issues with Datagram Transport Layer Security in Constrained Environments,” *draft-hartke-dice-practical-issues-01 (work in progress)*. [Online]. Available: http://tools.ietf.org/html/draft-hartke-dice-practical-issues-01. [Accessed: 15-Apr-2014].

[143] S. Raza, D. Trabalza, and T. Voigt, “6LoWPAN Compressed DTLS for CoAP,” in *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2012, pp. 287–289.

[144] M. Sethi, J. Arkko, and A. Keranen, “End-to-end security for sleepy smart object networks,” in *2012 IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops)*, 2012, pp. 964–972.

[145] T. Kothmayr, C. Schmitt, W. Hu, M. Brunig, and G. Carle, “A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication,” in *2012 IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops)*, 2012, pp. 956–963.

[146] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, “DTLS based security and two-way authentication for the Internet of Things,” *Ad Hoc Networks*.

[147] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, “Towards viable certificate-based authentication for the internet of things,” in *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy*, New York, NY, USA, 2013, pp. 37–42.

[148] S. Kumar, Z. Shelby, and S. Keoh, “Profiling of DTLS for CoAP-based IoT Applications,” *draft-keoh-dtls-profile-iot-00*. [Online]. Available: http://tools.ietf.org/html/draft-keoh-dtls-profile-iot-00. [Accessed: 15-Apr-2014].

[149] A. Yegin and Z. Shelby, “CoAP Security Options.” [Online]. Available: http://tools.ietf.org/html/draft-yegin-coap-security-options-00. [Accessed: 17-Sep-2013].

[150] W. Wang, L. Zhu, L. Wang, and F. Yu, “CoAP Option Extensions: Profile and Sec-flag,” *raft-wang-core-profile-secflag-options-02*. [Online]. Available: http://tools.ietf.org/html/draft-wang-core-profile-secflag-options-02. [Accessed: 15-Apr-2014].

[151] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, “Securing communication in 6LoWPAN with compressed IPsec,” in *2011 International*

Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011, pp. 1–8.

[152] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the Internet of Things—a comparison of link-layer security and IPsec for 6LoWPAN," *Security and Communication Networks*, 2012.

[153] M. A. Simplício Jr., P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed Wireless Sensor Networks," *Computer Networks*, vol. 54, no. 15, pp. 2591–2612, Oct. 2010.

[154] "GS1 Identification Keys (ID Keys)," *GS1 Identification Keys (ID Keys) | Techical | BarCodes & Identification | Products & Solutions | GS1 - The global languages of business*. [Online]. Available: http://www.gs1.org/barcodes/technical/id_keys. [Accessed: 15-Apr-2014].

[155] "Learning about ucode," *Learning about ucode - Ubiquitous ID Center*. [Online]. Available: http://www.uidcenter.org/learning-about-ucode. [Accessed: 15-Apr-2014].

[156] S. O. Amin, Y. jig Young, M. S. Siddiqui, and C. Hong, "A novel Intrusion Detection Framework for IP-based sensor networks," in *International Conference on Information Networking, 2009. ICOIN 2009*, 2009, pp. 1–3.

[157] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[158] R. Roman and J. Lopez, "Integrating wireless sensor networks and the internet: a security analysis," *Internet Research*, vol. 19, no. 2, pp. 246–259, Apr. 2009.

[159] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[160] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A Survey on Sensor-Cloud: Architecture, Applications, and Approaches," *International Journal of Distributed Sensor Networks*, vol. 2013, Feb. 2013.

[161] "Xively by LogMeIn - Business Solutions for the Internet of Things," *Xively by LogMeIn - Business Solutions for the Internet of Things*. [Online]. Available: http://xively.com. [Accessed: 15-Apr-2014].

[162] "SensorCloud - powered by LORD MicroStrain." [Online]. Available: http://www.sensorcloud.com/. [Accessed: 15-Apr-2014].

[163] "SensaTrack," *SensaTrack M2M Wireless Sensor Monitoring Service*. [Online]. Available: http://www.sensatrack.com. [Accessed: 15-Apr-2014].

[164] "NimBits - The Open Source Internet of Things on a Distributed Cloud," *Nimbits*. [Online]. Available: http://www.nimbits.com. [Accessed: 15-Apr-2014].

[165] “ThingSpeak,” *Internet of Things - ThingSpeak*. [Online]. Available: http://www.thingspeak.com. [Accessed: 15-Apr-2014].

[166] D. Guinard, V. Trifa, T. Pham, and O. Liechti, “Towards physical mashups in the Web of Things,” in *2009 Sixth International Conference on Networked Sensing Systems (INSS)*, 2009, pp. 1–4.

[167] “Towards the web of things: Web mashups for embedded devices,” presented at the Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2008), Madrid, Spain, 2009.

[168] V. Trifa, “Design and implementation of a gateway for web-based interaction and management of embedded devices,” in *Proceedings of the 2nd International Workshop on Sensor Network Engineering (IWSNE 2009)*, Marina del Rey, California, USA, 2009.

[169] W. I. Grosky, A. Kansal, S. Nath, J. Liu, and F. Zhao, “SenseWeb: An Infrastructure for Shared Sensing,” *IEEE MultiMedia*, vol. 14, no. 4, pp. 8–13, 2007.

[170] S. Nath, J. Liu, and F. Zhao, “SensorMap for Wide-Area Sensor Webs,” *Computer*, vol. 40, no. 7, pp. 90–93, 2007.

[171] D. Guinard, V. Trifa, and E. Wilde, “A resource oriented architecture for the Web of Things,” in *Internet of Things (IOT), 2010*, 2010, pp. 1–8.

[172] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, “IOT Gateway: BridgingWireless Sensor Networks into Internet of Things,” in *2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC)*, 2010, pp. 347–352.

[173] “SENSEI (Integrating the Physical with the Digital World of the Network of the Future),” *SENSEI - Home*. [Online]. Available: http://www.sensei-project.eu/. [Accessed: 15-Apr-2014].

[174] Y. W. Law, Y. Zhang, J. Jin, M. Palaniswami, and P. Havinga, “Secure rateless deluge: pollution-resistant reprogramming and data dissemination for wireless sensor networks,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, pp. 5:1–5:22, Jan. 2011.

[175] N. Bui, O. Ugus, M. Dissegna, M. Rossi, and M. Zorzi, “An integrated system for secure code distribution in Wireless Sensor Networks,” in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010, pp. 575–581.

[176] J.-M. Bohli, A. Hessler, O. Ugus, and D. Westhoff, “Security enhanced multi-hop over the air reprogramming with Fountain Codes,” in *IEEE 34th Conference on Local Computer Networks, 2009. LCN 2009*, 2009, pp. 850–857.

[177] C. Mauro, D. P. Roberto, M. Luigi V, and M. Alessandro, “Mobility and cooperation to thwart node capture attacks in manets,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, 2009.

[178] L. Sanchez, J. A. Galache, V. Gutierrez, J. M. Hernandez, J. Bernat, A. Gluhak, and T. Garcia, “SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities,” in *Future Network Mobile Summit (FutureNetw), 2011*, 2011, pp. 1–8.

[179] H. Hellbruck, M. Pagel, A. Kroller, D. Bimschas, D. Pfisterer, and S. Fischer, “Using and operating wireless sensor network testbeds with WISEBED,” in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, 2011, pp. 171–178.

[180] “Internet of Things - Architecture,” *Internet of Things - Architecture - IOT-A: Internet of Things Architecture*. [Online]. Available: http://www.iot-a.eu/public. [Accessed: 15-Apr-2014].

[181] P. Suriyachai, J. Brown, and U. Roedig, “Time-Critical Data Delivery in Wireless Sensor Networks,” in *Distributed Computing in Sensor Systems*, R. Rajaraman, T. Moscibroda, A. Dunkels, and A. Scaglione, Eds. Springer Berlin Heidelberg, 2010, pp. 216–229.

[182] J. Granjal, E. Monteiro, and J. Sa Silva, “A secure interconnection model for IPv6 enabled wireless sensor networks,” in *Wireless Days (WD), 2010 IFIP*, 2010, pp. 1–6.

[183] J. Granjal, J. Sa Silva, E. Monteiro, J. Sa Silva, and F. Boavida, “Why is IPSec a viable option for wireless sensor networks,” in *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008*, 2008, pp. 802–807.

[184] J. Granjal, E. Monteiro, and J. Sa Silva, “Enabling Network-Layer Security on IPv6 Wireless Sensor Networks,” in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, 2010, pp. 1–6.

[185] J. Granjal, E. Monteiro, and J. S. Silva, “Network-layer security for the Internet of Things using TinyOS and BLIP,” *International Journal of Communication Systems*, p. n/a–n/a, 2012.

[186] J. J. P. C. Rodrigues and P. A. C. S. Neves, “A survey on IP-based wireless sensor network solutions,” *International Journal of Communication Systems*, vol. 23, no. 8, pp. 963–981, 2010.

[187] J. Ko, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, and P. Levis, “Connecting low-power and lossy networks to the internet,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 96–101, 2011.

[188] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, “6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach,” *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1189–1212, 2012.

[189] W. Jung, S. Hong, M. Ha, Y.-J. Kim, and D. Kim, “SSL-Based Lightweight Security of IP-Based Wireless Sensor Networks,” in *International Conference on Advanced*

Information Networking and Applications Workshops, 2009. WAINA '09, 2009, pp. 1112–1117.

[190] V. Gupta, M. Wurm, Y. Zhu, M. Millard, S. Fung, N. Gura, H. Eberle, and S. Chang Shantz, "Sizzle: A standards-based end-to-end security architecture for the embedded Internet," *Pervasive and Mobile Computing*, vol. 1, no. 4, pp. 425–445, Dec. 2005.

[191] L. Casado and P. Tsigas, "ContikiSec: A Secure Network Layer for Wireless Sensor Networks under the Contiki Operating System," in *Identity and Privacy in the Internet Age*, A. Jøsang, T. Maseng, and S. J. Knapskog, Eds. Springer Berlin Heidelberg, 2009, pp. 133–147.

[192] A. Juels, "RFID security and privacy: a research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[193] C.-F. Lee, H.-Y. Chien, and C.-S. Laih, "Server-less RFID authentication and searching protocol with enhanced security," *International Journal of Communication Systems*, vol. 25, no. 3, pp. 376–385, 2012.

[194] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Fourth International Symposium on Information Processing in Sensor Networks, 2005. IPSN 2005*, 2005, pp. 364–369.

[195] R. Housley, *RFC 4309–Using Advanced Encryption Standard CCM Mode with IPsec Encapsulating Security Payload (ESP)*. Dec, 2005.

[196] L. M. L. Oliveira, A. F. de Sousa, and J. J. P. C. Rodrigues, "Routing and mobility approaches in IPv6 over LoWPAN mesh networks," *International Journal of Communication Systems*, vol. 24, no. 11, pp. 1445–1466, Nov. 2011.

[197] J. W. Hui and D. E. Culler, "IP is dead, long live IP for wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, New York, NY, USA, 2008, pp. 15–28.

[198] C. Madson and R. Glenn, "RFC 2404 The Use of HMAC-SHA—1—96 within ESP and AH," *IETF, November*, 1998.

[199] S. Frankel and H. Herbert, "RFC 3566—The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec," *NIST-National Institute of Standards and Technology*, vol. 820.

[200] "The Standalone AES Encryption of CC2420 (TinyOS 2.10 and MICAz)," *The Standalone AES Encryption of CC2420 (TinyOS 2.10 and MICAz) - CIS Lab, SJTU, Chine*. [Online]. Available: http://cis.sjtu.edu.cn/index.php/The_Standalone_AES_Encryption_of_CC2420_(TinyOS_2.10_and_MICAz). [Accessed: 15-Apr-2014].

[201] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.

[202] J. Granjal, E. Monteiro, and J. S. Silva, “On the Effectiveness of End-to-End Security for Internet-Integrated Sensing Applications,” in *2012 IEEE International Conference on Green Computing and Communications (GreenCom)*, 2012, pp. 87–93.

[203] J. Granjal, E. Monteiro, and J. Sa Silva, “On the feasibility of secure application-layer communications on the Web of Things,” in *2012 IEEE 37th Conference on Local Computer Networks (LCN)*, 2012, pp. 228–231.

[204] J. Granjal, E. Monteiro, and J. Sa Silva, “End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication,” in *IFIP Networking Conference, 2013*, 2013, pp. 1–9.

[205] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, “MiniSec: A Secure Sensor Network Communication Architecture,” in *6th International Symposium on Information Processing in Sensor Networks, 2007. IPSN 2007*, 2007, pp. 479–488.

[206] D. M. <mcgrew@cisco.com>, “An Interface and Algorithms for Authenticated Encryption,” *RFC 5116*, Jan. 2008.

[207] P. Eronen and H. Tschofenig, “Pre-Shared Key Ciphersuites for Transport Layer Security (TLS),” *RFC 4279*, Dec. 2005.

[208] A. Liu and P. Ning, “TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks,” in *International Conference on Information Processing in Sensor Networks, 2008. IPSN ’08*, 2008, pp. 245–256.

[209] B. C. Neuman and T. Ts’o, “Kerberos: an authentication service for computer networks,” *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, 1994.

[210] O. Bergman, “libcoap: C-implementation of CoAP.” [Online]. Available: http://sourceforge.net/projects/libcoap/. [Accessed: 15-Apr-2014].

[211] J. Granjal, E. Monteiro, and J. S. Silva, “Application-Layer Security for the WoT: Extending CoAP to Support End-to-End Message Security for Internet-Integrated Sensing Applications,” in *Wired/Wireless Internet Communication*, V. Tsaoussidis, A. J. Kassler, Y. Koucheryavy, and A. Mellouk, Eds. Springer Berlin Heidelberg, 2013, pp. 140–153.