Fábio Ferreira Dias

SISTEMAS DE CODIFICAÇÃO PARA SEGURANÇA NA CAMADA FÍSICA BASEADOS EM TÉCNICAS DE INTERLEAVING ALEATÓRIO E JAMMING

Dissertação de Mestrado

Julho de 2014

· U C ·

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE
ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES

SISTEMAS DE CODIFICAÇÃO PARA SEGURANÇA NA CAMADA FÍSICA BASEADOS EM TÉCNICAS DE INTERLEAVING ALEATÓRIO E JAMMING

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores
- Especialização em Telecomunicações –

Autor
Fábio Ferreira Dias

Orientador
Professor Doutor Marco Alexandre Cravo Gomes
Co-orientador
Professor Doutor João Paulo da Silva Machado Garcia Vilela

Júri
Presidente Professora Doutora Teresa Martinez dos Santos Gomes

Vogal Professor Doutor Luís Alberto da Silva Cruz

Coimbra
Julho 2014

Agradecimentos

A realização desta dissertação de mestrado contou com importantes apoios e incentivos, por isso é com muita satisfação que expresso aqui o mais profundo agradecimento a todos aqueles que tornaram possível a realização da mesma.

Aos Professores Doutores Marco Alexandre Cravo Gomes e João Paulo da Silva Machado Garcia Vilela, ambos orientadores da tese, pelo apoio, pelas críticas, correções e sugestões relevantes, pelo incentivo e inesgotável disponibilidade demonstrada em todas as fases que levaram à concretização deste trabalho.

Ao Instituto de Telecomunicações, por todos os meios disponibilizados.

Aos demais docentes do DEEC (professores, alunos e funcionários), por todo o seu apoio e companheirismo.

Aos meus amigos, que estiveram ao meu lado durante esta fase, pelo companheirismo, força e apoio naqueles momentos mais difíceis.

Por último, dirijo um agradecimento especial aos meus pais, por serem um exemplo de coragem, pelo inestimável apoio, incentivo, amizade e total cooperação para superar os obstáculos que ao longo desta caminhada foram surgindo.

A todos,

Um bem-haja

Resumo

Nesta dissertação propomos dois esquemas de codificação baseados na combinação de técnicas de *jamming*, *interleaving* e códigos de bloco sistemáticos, com elevada capacidade de correção de erros, a fim de melhorar a segurança no meio sem fio, com um custo de energia reduzido e simultaneamente garantindo uma transmissão fiável dos dados. A ideia base consiste em gerar uma chave de *interleaving* aleatória que é usada para embaralhar a informação enviada pelo transmissor. A chave é enviada para o recetor legítimo durante um curto período de comunicação vantajoso sobre o *eavesdropper* (que pode ser obtido, por exemplo, através da utilização de *jammers*). Esta é obtida no lado do recetor, juntamente com o *interleaving* dos dados e, posteriormente, utilizada para o desembaralhar os mesmos. O destinatário legítimo recebe com maior fiabilidade a chave, o que lhe proporciona a necessária vantagem sobre o *eavesdropper*. A segurança em rede sem fios resulta de se usar técnicas de *interleaving* combinadas com *jamming* para fins de segurança ao invés do seu tradicional uso em garantir robustez a rajadas de erros.

Palavras-chave:

Segurança na camada física; Codificação de canal; *Jamming; Interleaving,* Códigos LDPC (*Low Density Parity Check Codes*)

Abstract

We propose two coding schemes based on the combination of jamming techniques, interleaving, and powerful channel codes for wireless secrecy at a reduced energy cost that simultaneously guarantee reliable data transmission. The basic idea lies in generating a short random interleaving key that is used to shuffle the information at the source. This key is then sent to the legitimate receiver during a brief period of advantageous communication over the eavesdropper (e.g., due to more interference from a jammer). Finally, the key is decoded at the legitimate receiver to properly deinterleave the original information. Bob receives a better quality version of the interleaving key, therefore providing the needed advantage over the eavesdropper. Wireless secrecy comes from using interleaving for secrecy purposes rather than for reliability alone.

Keywords:

Physical-layer Security; Channel Coding; *Jamming; Interleaving,* LDPC codes (*Low Density Parity Check Codes*)

Índice

GLOSSÁRIO

AWGN	Ruído Branco Aditivo Gaussiano (*Additive White Gaussian Noise*)
BEWC	*Binary Erasure Wiretap Channel*
BER	Taxa de Bits Erros (*Bit Error Rate*)
BN	Nodo de Bit (*Bit Node*)
BP	*Belief Propagation*
BSWC	*Binary Symmetric Wiretap Channel*
CN	Nodo de Teste (*Check Node*)
DVB	Transmissão de Vídeo Digital (*Digital Video Broadcast*)
FEC	Códigos corretores de erro (*Foward Error Correction*)
GWC	*Gaussian Wiretap Channel*
IRA	*Irregular Repeat and Accumulate*
LLR	Máxima Verosimilhança Logarítmica (*Log Likelihood Ratio*)
LR	Máxima Verosimilhança (*Likelihood Ratio*)
LSPA	Algoritmo Soma de Produtos no Domínio Logarítmico (*Logarithmic Sum Product Algorithm*)
MAC	Controlo de Acesso ao Meio (*Media Acess Control*)
QPSK	Modulação de Desvio de Fase em Quadratura (*Quadrature Phase Shift Keying)*
SNR	Relação Sinal Ruído (*Signal-to-Noise Ratio*)
SPA	Algoritmo Soma de Produtos (*Sum Product Algorithm*)
WC	*Wiretap Channel*

CAPÍTULO 1

INTRODUÇÃO

A obtenção de segurança na camada física, explorando as características ruidosas e dispersivas dos canais de comunicação sem fios, tem sido objeto de interesse e estudo de investigação por parte da comunidade científica com vista ao aumento do nível de segurança destas redes. Neste trabalho de dissertação é proposto a obtenção de segurança na camada física através da combinação de técnicas avançadas de *Jamming* e codificação com *interleaving.* Pretende-se com as técnicas propostas criar uma camada adicional de segurança que impeça a dispositivos maliciosos, à escuta na rede, o acesso a informação que não lhes é destinada.

Hoje em dia, os métodos de segurança mais usados recorrem a técnicas criptográficas, que requerem na sua generalidade a partilha prévia de uma chave privada comum entre dois utilizadores através de um canal seguro. No contexto de obtenção de segurança na camada física torna-se interessante o estudo de técnicas que permitam a troca sistemática de chaves secretas sem necessidade da utilização de um canal seguro adicional. Mais uma vez se realça que estas técnicas pretendem apenas complementar, e não substituir, os algoritmos de segurança usados nas camadas protocolares superiores.

A troca de chaves sobre o canal de transmissão deverá implicar sempre uma vantagem dos transmissores e recetores legítimos face a dispositivos maliciosos da rede. A utilização de *Jamming* surge como uma solução natural, isto é a introdução de ruído no sistema, sem que este interfira (ou interfira de forma limitada) com a comunicação legítima, mas que seja capaz de causar interferência suficiente a um potencial intruso. Isto coloca, no entanto, alguns desafios (objetivos) que são abordados nesta dissertação, nomeadamente: como causar interferência necessária a intrusos sem que a comunicação legítima seja comprometida; como combinar a geração de ruído com métodos de codificação robustos ao mesmo; desenvolver uma variedade de mecanismos de geração de ruído assegurando resultados vantajosos para o sistema; como reduzir o custo energético associado à geração de interferência; entre outros…

1.1 MOTIVAÇÃO E CONTEXTUALIZAÇÃO

A utilização de redes sem fio tornou-se numa parte integrante e indispensável do nosso dia-a-dia mas, as questões relacionadas com requisitos de segurança das mesmas são muito relevantes e motivo de preocupação de qualquer utilizador. Hoje em dia estamos frequentemente dependentes de uma rede sem fios para transmissão de informação privada, como por exemplo, transações de cartões de crédito ou comunicação de dados bancários. Assim, a capacidade para partilhar informação secreta em segurança, mesmo com presença de intrusos, é extremamente importante.

Em muitos casos, os intrusos focam-se em tentar lançar vários ataques para obter acesso não autorizado ou mesmo interromper fluxos de informação [Murthy 2004]. A maioria dos ataques pode ser classificada em duas categorias: passivo e ativo. Os ataques ativos podem interferir significativamente com o normal funcionamento da rede pois o intruso, na maioria das vezes, centra-se fundamentalmente na corrupção dos dados transmitidos através da rede. Já os ataques passivos não perturbam o funcionamento da rede e o objetivo dos intrusos é roubar a informação transmitida proveniente dos canais sem fios [Sheng 2011]. Neste trabalho é assumido o caso de um atacante passivo.

Os métodos de segurança mais usados recorrem a técnicas de criptografia nas camadas superiores da rede. No entanto, Wyner provou a existência de códigos, chamados *Wiretap Codes* [Wyner 1975], que providenciam na camada física simultaneamente a comunicação fiável com um recetor legítimo e segurança face a um utilizador não autorizado assumindo que este último recebe uma versão degradada dessa informação. O desenvolvimento destes códigos tem-se mostrado, no entanto, um desafio, havendo apenas algumas contribuições aplicadas a modelos de canal simplificados e de aplicação prática limitada.

No nosso trabalho iremos mostrar ser possível, de acordo com Wyner, obter simultaneamente e de forma prática, segurança e fiabilidade de transmissão na camada física, providenciando desta forma uma camada adicional de segurança na rede.

1.2 DESCRIÇÃO DO ESTADO DE ARTE

Os problemas da privacidade e segurança nas comunicações de redes sem fio assumiram um papel de relevo uma vez que estas continuam em constante crescimento. Tradicionalmente, a segurança é vista como uma característica independente, com pouca ou nenhuma relação com as tarefas de comunicação de dados, e, portanto, os algoritmos de encriptação são insensíveis à natureza física do meio sem fios. Assim, o objetivo de uma comunicação segura e fiável passa por garantir aquando da transmissão de uma mensagem que os recetores legítimos sejam capazes de recuperar a mensagem sem erros (fiabilidade), garantindo que os restantes recetores (não legítimos) não sejam capazes de adquirir qualquer informação (confidencialidade).

1.2.1 *WIRETAP CHANNEL*

O estudo de códigos ou técnicas de codificação capazes de garantir fiabilidade e confidencialidade devem ter em conta as características dos canais entre o transmissor.

Entre os modelos de canais mais utilizados neste estudo o "*wiretap channel*" (Figura 1.1) proposto por Wyner [Wyner 75] é o mais popular devido à sua simplicidade.

O modelo *wiretap channel* considera que a comunicação do transmissor (vulgarmente designado por Alice) e do recetor legítimo (normalmente designado por Bob) é imune a ruído, enquanto o eavesdropper (Eve) deveria de receber uma versão degradada do sinal transmitido [Wyner 75].

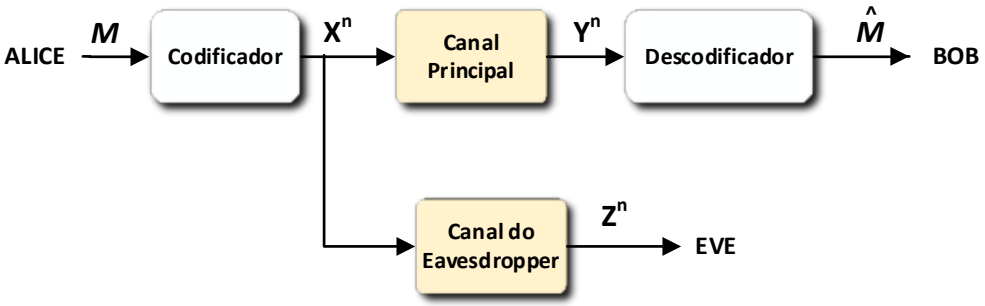


Figura 1.1 - Modelo Wyner *wiretap channel*

Este modelo pode ser generalizado, assumindo que o canal de comunicação entre Alice e o Bob pode também possuir ruído. Assim, considerando uma mensagem **M** enviada pela Alice, $\mathbf{X}^{\mathbf{n}}$ é a palavra de código obtida após a codificação, com **n** símbolos. A palavra recebida pelo Bob é $\mathbf{Y}^{\mathbf{n}}$, enquanto o Eve observa uma versão degradada da mesma, isto é, $\mathbf{Z}^{\mathbf{n}}$. Basicamente, o conceito deste modelo é garantir que tanto Alice como o Bob comunicam através de um canal principal enquanto o Eve tem acesso a uma versão degradada.

Existem diferentes variantes do *wiretap channel* usados no projeto de códigos capazes de garantir fiabilidade e confidencialidade, vulgarmente designados de *wiretap codes*. Em [Ozarow 84], Ozarow e Wyner realizam este estudo para o canal *binary erasure wiretap channel (BEWC)* que assume um canal sem ruído entre a Alice e o Bob e um canal binário erróneo (com probabilidade de erro p) entre a Alice e o Eve (Figura 1.2).

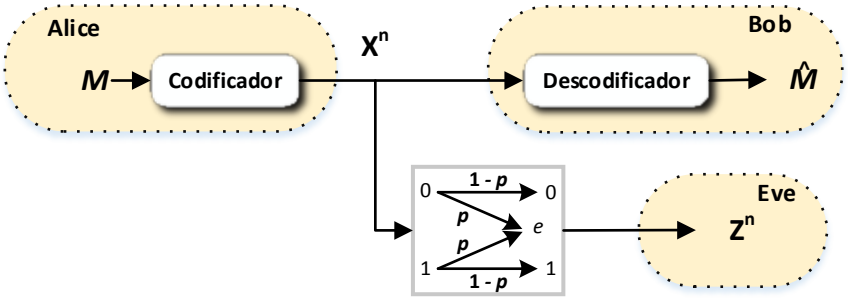


Figura 1.2 - Modelo *binary erasure wiretap channel* (BEWC)

No projeto de *wiretap codes* baseados em códigos Low Density Parity Check (LDPC) [Thangaraj2005][Thangaraj2007][Liu 2007] foi também considerado o canal *binary symmetric wiretap channel* (BSWC), em que à semelhança do BEWC assume um canal sem ruído entre a Alice e o Bob, mas em que o canal entre a Alice e o Eve é do tipo binário simétrico (Figura 1.3).

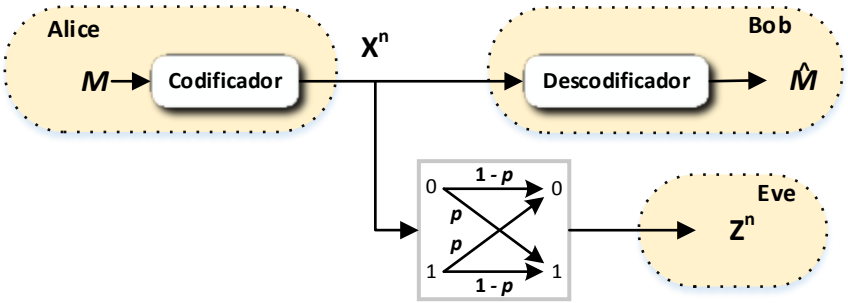


Figura 1.3 - Modelo *binary symetric wiretap channel* (BSWC)

Outro modelo comum é o *Gaussian Wiretap Channel* (GWC) usado em [Bloch 2006], [Muramatsu2006] e [Bloch 2007] no estudo da combinação de protocolos com chaves secretas baseadas em códigos LDPC poderosos. O modelo GWC (Figura 1.4) assume que os canais entre Alice e o Bob e a Alice e o Eve são ambos de ruído branco aditivo Gaussiano (AWGN), em que o primeiro tem uma densidade espetral de potência (unilateral) de ruído N_0 e o segundo, αN_0 com $\alpha > 1$.

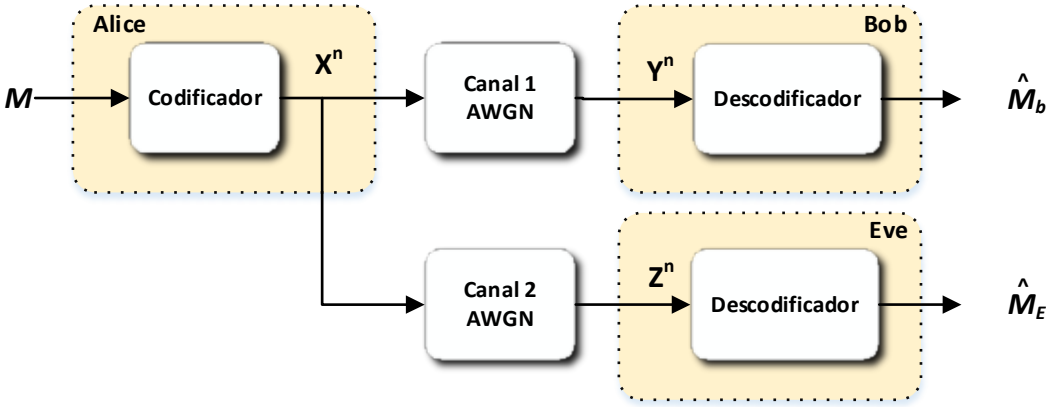


Figura 1.4 - Modelo *Gaussian wiretap channel* (GWC)

De salientar que o desenvolvimento do nosso trabalho tem por base este modelo.

1.2.2 MÉTRICAS DE SEGURANÇA

Diversas métricas podem ser usadas para medir a segurança nos canais de comunicação. Enquanto algoritmos modernos de criptografia têm fornecido uma segurança eficaz para uma serie de aplicações que usam medidas de segurança computacional, a segurança através da teoria da informação é reconhecida como a forma mais poderosa de segurança [Bloch 2011].

A primeira métrica foi introduzida por Shannon juntamente com o conceito de *perfect secrecy* [Shannon 49]. Considerando o modelo *wiretap channel* proposto por Wyner, Alice envia uma mensagem **M** que é codificada numa palavra de código $\mathbf{X^n}$, com **n** símbolos, sendo portanto, taxa de informação de envio,

$$R_A = \frac{\mathrm{H}(M)}{n}. \tag{1.1}$$

em que $\mathrm{H}(M)$ é a entropia da mensagem enviada. Os dois recetores (Bob e Eve) recebem respetivamente, $\mathbf{Y^n}$ e $\mathbf{Z^n}$, à saída dos seus canais. A quantidade de informação que o Bob e o Eve desconhecem da mensagem **M** é chamada de taxa de equivocação, definida respetivamente como:

$$R_B = \frac{1}{n}\mathrm{H}(M \mid Y^n), \tag{1.2}$$

$$R_e = \frac{1}{n}\mathrm{H}(M \mid Z^n), \tag{1.3}$$

com $0 \le R_B, R_e \le \mathrm{H}(M)/n$.

Diz-se que o código fornece *perfect secrecy* quando $R_e = R_A$, isto é quando é nula a informação mútua entre Alice e o Eve, $I(M;Z^n) = H(M) - \mathrm{H}(M \mid Z^n) = 0$.
Importa referir que esta condição não é simples de concretizar de forma prática, pelo que Wyner sugeriu o uso de um requisito mais fraco para a segurança [Wyner 75]. Em vez de se exigir que a equivocação do Eve seja exatamente igual à entropia da mensagem, pede-se que a taxa de equivocação $(1/n)\,\mathrm{H}(M \mid Z^n)$ convirja assintoticamente para a taxa de entropia da mensagem $(1/n)\,\mathrm{H}(M)$ quando $n \to \infty$, i.e.

$$\lim_{n\to\infty} \frac{1}{n} I(M;Z^n) = 0 \ . \qquad (1.4)$$

Esta condição é vulgarmente designada de *weak secrecy*. Baseado nestes conceitos, Wyner demonstrou que existem códigos que garantem fiabilidade (baixa probabilidade de erro) no Bob e confidencialidade em relação a Eve.

A *secrecy capacity* é definida como a maior taxa de transmissão à qual é possível garantir transmissão com fiabilidade e confidencialidade. Wyner demonstrou ainda que atingir a *secrecy capacity* depende de Eve obter uma versão degradada (por exemplo, devido a ruído extra) da mensagem enviada por Alice.
Uma medida mais operacional de segurança consiste na *security gap* definida em [Klinc 2011]. Tendo como base o modelo GWC da figura 1.4, suponhamos que Alice quer transmitir a mensagem **M** para o Bob, e usando um código corretor de erros para codificar a mensagem **M** de *n* bits numa palavra de código $\mathbf{X^n}$**,** transmite-a ao longo de um canal AWGN para o Bob. O Eve observa a transmissão com ruído, num canal AWGN independente e tenta reconstruir a mensagem **M**. Isto conduz a uma taxa média de erro de bit (BER) sobre a estimativa do Bob $\mathbf{P}_e^B$ e a um BER sobre a estimativa de Eve $\mathbf{P}_e^E$. É então desejável que $\mathbf{P}_e^B$ seja suficientemente baixo para assegurar fiabilidade e que $\mathbf{P}_e^E$ suficientemente elevado para garantir confidencialidade. Se $\mathbf{P}_e^E$ se aproximar de 0.5, Eve não será capaz de extrair informação a partir da sequência recebida $\mathbf{Z^n}$. Assim, para qualquer ϵ > 0 e $\mathbf{P}_{e,\min}^E (\approx 0.5)$ um *wiretap code* deve garantir

a) $\mathbf{P}_e^B \leq$ ϵ(fiabilidade),
b) $\mathbf{P}_e^E \geq \mathbf{P}_{e,\min}^E$ (segurança).

Consideremos agora que $\mathrm{SNR}_{B,\min}$ é o menor SNR que assegura a) e que $\mathrm{SNR}_{\mathrm{E,max}}$ é o SNR mais elevado que garante b). Assume-se que o Bob opera a pelo menos $\mathrm{SNR}_{B,\min}$, enquanto o SNR do Eve é sempre menor do que $\mathrm{SNR}_{\mathrm{E,max}}$. A *security gap* (Figura 1.5) é

então definida como a relação entre $SNR_{B,\min} - SNR_{\mathrm{E,max}}$ (em dB), ou seja, a diferença mínima exigida das relações sinal-ruído entre Bob e Eve, para qual é possível a comunicação segura. Os códigos corretores de erros convencionais requerem uma elevada *security gap*, quando a probabilidade mínima de erro no Eve é superior a 0.4 ($P_{e,\min}^{E} > 0.4$).

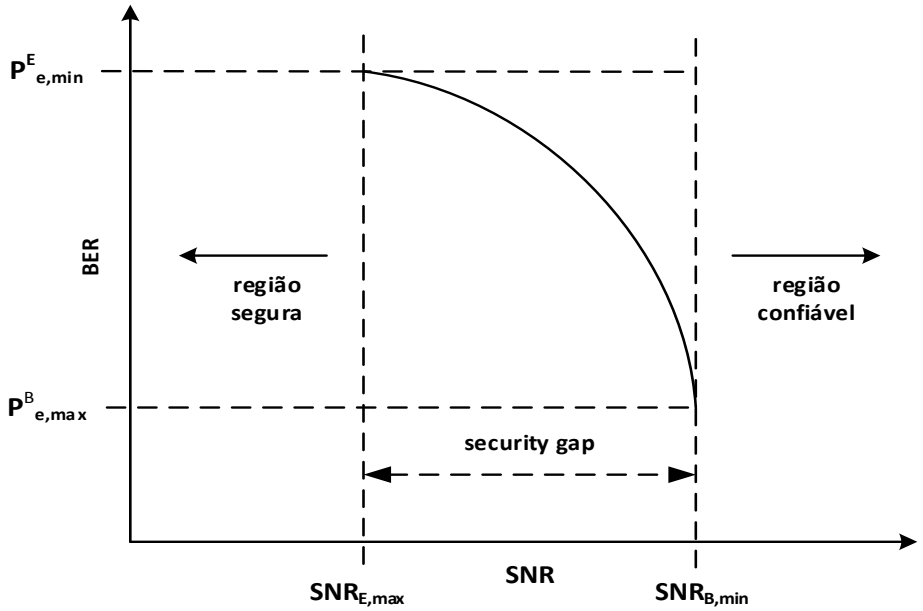


Figura 1.5 - A security gap. Esta curva mostra o BER típico vs. desempenho SNR de um código corretor de erros. $SNR_{B,min}$ define a extremidade inferior da região de confiança, enquanto que $SNR_{E,max}$ define a extremidade superior dessa região. A security gap (em dB) é a diferença de SNR entre $SNR_{B,min} - SNR_{E,max}$ que deve ser mantido entre Bob and Eve a fim de alcançar a fiabilidade e confidencialidade [Klinc 2011].

1.2.3 INGREDIENTES DE CODIFICAÇÃO PARA SEGURANÇA

Com vista ao desenvolvimento de técnicas de codificação capazes de providenciar simultaneamente fiabilidade e confidencialidade, torna-se necessário perceber quais os ingredientes que podem providenciar a desejada confidencialidade.
Considerando o modelo base do *wiretap channel* definido anteriormente que assume um canal degradado entre a Alice e o Eve, quando comparado ao canal entre a Alice e o Bob, o principal ingrediente consiste em adicionar uma componente de aleatoriedade nas palavras de código de cada mensagem enviada pela Alice. Uma possível forma para atingir este objetivo consiste em a codificação se basear em diferentes códigos, resultanto em várias palavras de código possíveis para uma determinada mensagem m. Por outras palavras, cada palavra de código é determinada pela mensagem m que a Alice pretende enviar e uma mensagem escolhida aleatoriamente m' é de tal modo que a codificação a partir da perspetiva da mensagem m seja estocástica, como é ilustrado na figura 1.6.

O conjunto de palavras de códigos que corresponde a uma dada mensagem m consiste num sub-código do código inteiro que a diferentes mensagens m faz corresponder diferentes palavras de código.

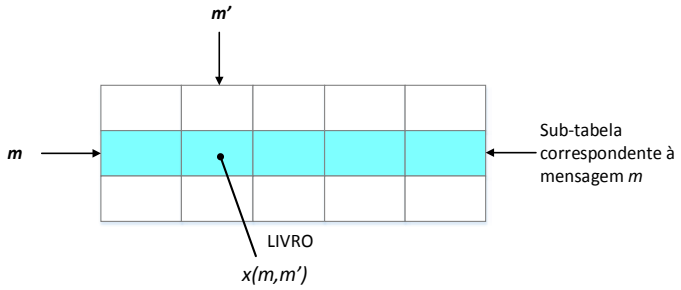


Figura 1.6 – Estrutura isolada de um código com *secrecy* (exemplo retirado de [Willie 2013])

Para melhor entendermos este conceito, vejamos o seguinte exemplo retirado de [Willie 2013].

EXEMPLO 1

[TABELA 1] Exemplo de Código para segurança				
(m,m')	0	1	2	3
0	0000	0011	1100	1111
1	0001	0010	1101	1110
2	1000	1011	0100	0111
3	1001	1010	0101	0110

Suponhamos que empregamos um código binário com taxa de ½ com $k=2$ e $n=4$ no *wiretap channel* da figura 1.1, , onde o canal principal é imune a ruido e o canal do Eve é um BEC (*Binary Erasure Channel*). A linha é escolhida tendo em conta a mensagem m, e a escolha da coluna é baseada na mensagem gerada m' aleatoriamente.

Suponha que para uma dada transmissão Eve obtém **z**=(?01?), sobre o seu canal. Podemos observar que cada linha contém uma correspondência a esta palavra de código incompleta. Desta forma, desde que o valor de m' seja escolhido aleatoriamente, Eve não obtém nenhuma informação a partir da mensagem $\mathbf{Z}^n$. No entanto o Bob não tem qualquer problema a identificar a mensagem transmitida a partir de qualquer palavra de código recebida. Já se o Eve recebesse **z**=(00??), ele seria capaz de determinar que $m \in \{0,1\}$, eliminando 2 e 3 como possíveis mensagens. O desafio é pois projetar um código ou uma estrutura de codificação que consiga garantir a confusão do Eve sempre que este possui alguma desvantagem na observação do sinal recebido relativamente ao BOB.

A estrutura *nested* do código e a aleatoriedade entre múltiplas palavras de código são dois ingredientes chaves utilizado em praticamente todos os códigos que fornecem segurança na camada física [Storn 97]. Desenvolver diretrizes claras para códigos de comprimento curto continua a ser um problema largamente aberto, mas como foi mostrado por Willie Harrison [Willie 2013], a conceção de códigos será um pouco mais fácil.

No nosso trabalho desenvolvido por nós aborda esquemas de codificação baseados na combinação de *interleaving com* códigos LDPC. Neste caso, a componente de aleatoriedade no código advém da geração de uma chave aleatória de interleaving que é usada para misturar os dados antes de serem enviados para o canal de comunicação.

1.3 CONTRIBUIÇÃO DA DISSERTAÇÃO

Nesta dissertação, propomos a obtenção de segurança na camada física através de técnicas avançadas de *Jamming* e de *interleaving com* códigos binários definidos por matrizes de teste de paridade esparsas (conhecidos por Low Density Parity-Check Codes(LDPC)) capazes de providenciar elevada fiabilidade. A componente de aleatoriedade capaz de providenciar confidencialidade é garantida pela utilização de interleaving cuja chave de baralhamento é escolhida aleatoriamente, sendo transmitida embutida na palavra de código enviada. O uso de *jamming* durante o curto espaço de transmissão da chave de *interleaving* garante a vantagem do Bob em relação ao Eve.

As principais contribuições desta dissertação são:

- Proposta de sistemas de codificação para segurança que providenciam simultaneamente fiabilidade e confidencialidade na camada física. Nomeadamente, são apresentados dois esquemas distintos de codificação baseados em códigos LDPC combinados com técnicas de *interleaving* aleatórios e *jamming;*
- Desenvolvimento de um simulador com diferentes formas de aplicação d*e jamming* no sistema;
- Submissão do artigo “Random Interleaving for Physical-layer Security” à revista internacional tipo A do IET (*The Institution of Engineering and Technology*)-Electronics Letters.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

Este documento é composto por 6 capítulos que abordam o trabalho realizado no âmbito desta dissertação de mestrado.

No capítulo 2, é feita uma descrição das características principais dos códigos LDPC, evidenciando o seu excelente desempenho assim como a importância da descodificação iterativa, mais especificamente, o algoritmo *Soma de Produtos*. Também serão descritos os dois tipos clássicos de *interleaving* (*interleaving* convolucional e *interleaving* de blocos), explicando também em que contexto as suas características são úteis ao nosso trabalho.

No capítulo 3, são apresentadas diversas técnicas de *jamming* e identificadas as que mais se adequam ao nosso projeto. A ideia fundamental deste capítulo prende-se essencialmente nas diferentes formas de aplicar ruído adicional ao *eavesdropper.* Com vista a testar o efeito do *jamming* de variadas formas no sistema por nós proposto, foi desenvolvido um simulador para diferentes tipos de *jammers* que é também apresentado neste capítulo.

No capítulo 4, é feita uma apresentação e consequente descrição de dois esquemas distintos que conjugam técnicas de *interleaving* e de codificação, com o intuito de obter uma maior segurança no meio sem fio.

No capítulo 5 são apresentados e discutidos os resultados experimentais relativos às diversas experiências efetuadas.

O capítulo 6 apresenta as principais conclusões extraídas do trabalho realizado, terminando com identificação de linhas de trabalho futuro.

CAPÍTULO 2

CÓDIGOS LDPC E TÉCNICAS DE INTERLEAVING

Este capítulo aborda questões relacionadas com a codificação de um canal, através de códigos de blocos lineares que introduzem redundância na sequência de dados, permitindo ao recetor ter a capacidade de detetar e corrigir erros causados por ruídos no canal de comunicação. Entre os códigos de blocos mais populares e com grande implantação nas normas de transmissão atuais destacam-se os códigos definidos por *Matrizes de Teste de Paridade Esparsas* (Low Density Parity Check Codes - LDPC), de que também é feito uso nos sistemas de codificação práticos para segurança e fiabilidade propostos nesta dissertação. Assim, também serão mostrados os principais conceitos dos códigos LDPC, sendo um deles a representação destes códigos por meio de gráficos bipartidos de Tanner [Tanner 81]. Além disso, também é apresentado o *Algoritmo Soma de Produtos* (SPA), que é um algoritmo iterativo utilizado na descodificação destes códigos poderosos.

Também será abordada uma técnica adicional para combater a ineficácia dos códigos referidos em corrigir rajadas de erros (*burst* de erros). De forma a contornar este inconveniente serão descritas técnicas de *interleaving* para baralhar os dados recebidos e assim espalhar as rajadas de erros no tempo.

2.1 CÓDIGOS *LOW DENSITY PARITY-CHECK*

Os códigos baseados em *Matrizes de Teste de Paridade Esparsas*, mais conhecidos como *Low Density Parity-Check Codes* (LDPC), foram originalmente criados por Robert Gallager [Gallager 63] em 1963, que os propôs na sua tese de doutoramento. Gallager propôs também um eficiente algoritmo iterativo para a descodificação dos códigos LDPC conhecido como *Algoritmo Soma de Produtos* (SPA). No entanto, estes códigos foram ignorados durante décadas, devido à elevada complexidade computacional do seu algoritmo SPA. Em 1981, R. M. Tanner [Tanner 81] generalizou o trabalho proposto por Gallager e apresentou uma representação gráfica dos códigos LDPC através de grafos bipartidos, conhecidos como *Tanner Graphs*. Apesar das suas excelentes características, os códigos LDPC, salvo raras exceções, não foram alvo de qualquer atenção por parte da comunidade científica até meados dos anos 90, altura em que foram redescobertos por Mackay e Neal [Mackay 99], tendo sido provado existirem códigos LDPC capazes de aproximar o limite de codificação de Shannon [Shannon 48]. A par dos turbo codes [Berrou 93], os códigos LDPC constituem uma das mais poderosas classes de códigos corretores de erros o que levou à sua integração recente nas principais normas de transmissão digital, e.g., DVB-S2/DVB-T2/DVB-C2 [Nick 2011], WiMAX (802.16e) [Brack 2006], WiFi (IEEE 802.11n-2009), entre outras.

2.1.1 CODIFICAÇÃO

Um código binário LDPC como qualquer código linear pode ser descrito por uma matriz binária H de teste de paridade [Shu 83] que no caso dos LPDC's é esparsa, i.e. com uma baixa densidade de 1's. A matriz H corresponde ao conjunto de restrições de paridade que as palavras de código deverão obedecer. A partir da matriz H pode ser obtida a matriz geradora *G*, que é capaz de gerar as palavras de código válidas correspondentes a cada uma das mensagens que se pretende transmitir. Ambas as matrizes descrevem perfeitamente o código.

Dado um código binário linear (n,k), ou seja, um código em que as palavras são vetores de dimensão n do tipo,

$$\mathbf{c} = [\mathrm{c}_0 \mathrm{c}_1 \cdots \mathrm{c}_{n-1}] \tag{2.1}$$

e as mensagens a codificar são vetores de dimensão k (com $n > k$) do tipo

$$\mathbf{m} = [\mathrm{m}_0 \mathrm{m}_1 \cdots \mathrm{m}_{k-1}] \tag{2.2}$$

sendo que, as palavras de código podem ser obtidas a partir da seguinte equação

$$\mathbf{c} = \mathbf{mG} \tag{2.3}$$

Devido ao facto de cada palavra de código ser gerada a partir do produto da mensagem $\mathbf{m}$ pela matriz $\mathbf{G}$, esta é designada por *Matriz Geradora*.

Como referido anteriormente, um código linear pode também ser definido por uma matriz de teste de paridade $\mathbf{H}$, que corresponde ao conjunto de restrições que as palavras de código devem obedecer, i.e, dada uma palavra $\mathbf{c}$ esta é de código se e só se:

$$\mathbf{cH}^T = \mathbf{0} \ . \tag{2.4}$$

Um código linear definido por uma matriz $\mathbf{H}$ de máxima característica é sempre sistematizável, i.e., por aplicação de operações lineares às linhas de $\mathbf{H}$, esta poderá ser escrita na forma,

$$\mathbf{H} = [\mathbf{P}^{\mathbf{T}} \mid \mathbf{I}_{n-k}], \tag{2.5}$$

com I_{n-k} a matriz identidade de dimensões $(n-k)\times(n-k)$ a partir do qual pode ser obtida a matriz $\mathbf{G}$ com

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}], \tag{2.6}$$

Sendo os códigos LDPC lineares, a forma óbvia de realizar a codificação seria partindo do conhecimento da matriz geradora $\mathbf{G}$ determinar as palavras de código de acordo com a equação (2.3), ou seja, fazendo $\mathbf{c} = \mathbf{mG}$. No entanto, os métodos utilizados na construção de códigos LDPC, assentam na obtenção da sua matriz de teste de paridade $\mathbf{H}$ ou no equivalente gráfico de Tanner. Na grande maioria das vezes a matriz $\mathbf{H}$ obtida não é sistemática, nem de máxima característica. A obtenção da matriz $\mathbf{G}$ não é pois imediata. Podem, no entanto, surgir situações em que para obter $\mathbf{H}$ na forma (2.4) se torne necessário efetuar troca de colunas, obtendo-se desta forma um código LDPC diferente mas com o mesmo desempenho do original.

Embora de fácil implementação, o método anterior é extremamente dispendioso em termo do número de operações a realizar na codificação de cada palavra de código. Devido ao facto de a sistematização da matriz $\mathbf{H}$ não conduzir, necessariamente, à obtenção de uma matriz $\mathbf{G}$ com baixa densidade de 1's, pelo que a codificação por (2.3) exige a realização de um número extremamente elevado de operações.

Uma aproximação alternativa para a solução deste problema consiste na obtenção de códigos LDPC por métodos algébricos e geométricos em que a codificação possa ser realizada por circuitos simples baseados em registos de deslocamento. É o caso dos códigos LDPC irregular repeat and accumulate (IRA) [Jin 2000] códigos sistemáticos, cuja codificação pode ser realizada diretamente a partir da matriz H.

CODIFICAÇÃO SISTEMÁTICA

Dada uma matriz geradora genérica de máxima característica e dimensões $k \times n$, cada palavra de código gerada não apresenta explicitamente a mensagem que a gerou como sendo um segmento do próprio vetor código. Isso significa que neste tipo de codificação a mensagem passa a ser conhecida somente após o processo de descodificação. Essa forma de codificação é chamada de codificação não sistemática.

Uma característica desejável num processo de codificação para um código de bloco linear é aquela que permite que o vetor código seja composto por dois segmentos: um segmento composto pelos $(n-k)$ bits de redundância que permitem a verificação da validade do vetor e outro segmento correspondente aos k bits da mensagem que gerou os bits de redundância. A disposição do segmento redundância e do segmento mensagem é uma questão de normalização, sendo um exemplo a normalização na Figura 2.1. A forma de codificação que permite a obtenção do vetor código nesse formato é chamada de *codificação sistemática.*

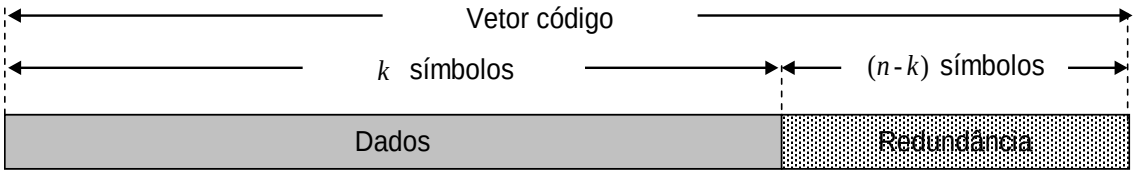


Figura 2.1 – Exemplo de codificação sistemática

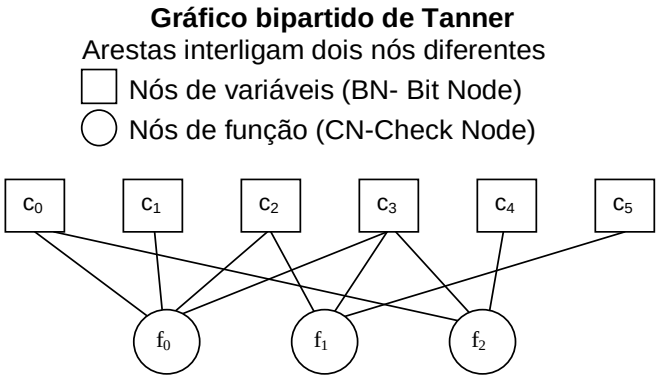
Vetores códigos com o conceito mostrado na Figura 2.1 podem ser obtidos a partir de matrizes geradoras com um formato específico com a matriz geradora a ser formada por duas sub-matrizes: uma matriz de paridade com dimensões $k \times (n-k)$ e outra matriz identidade de dimensões $k \times k$. No caso do exemplo da Fig. 2.1 a matriz geradora possui o formato apresentado em (2.5). Desta forma, a matriz de paridade permite que o segmento paridade seja obtido pela soma linear dos bits da mensagem, enquanto a matriz identidade permite que o segmento mensagem seja replicado de seguida.

Uma vez que uma matriz geradora é um arranjo de vetores linearmente independentes, uma matriz geradora de um código de bloco linear na forma sistemática pode ser obtida pela conveniente combinação linear dos vetores geradores e/ou permutação de colunas ou linhas da matriz geradora na forma não sistemática para a obtenção de outro arranjo de novos vetores geradores, linearmente independentes, no formato desejado.

2.1.2 REPRESENTAÇÃO POR MEIO DE GRÁFICOS BIPARTIDOS

O avanço na teoria dos códigos LDPC teve a contribuição de Tanner [Tanner 81], visto que este usou gráficos bipartidos para representar graficamente a matriz de verificação de paridade.

Um gráfico bipartido (também conhecido por *Gráfico de Tanner*) é dividido em duas regiões distintas, cada uma com o seu nó, ligadas por linhas ou ramos que unem nós de diferente tipo. Supondo que se quer representar um código de blocos, num dos lados do gráfico são colocados n nós de variáveis (*Bit Nodes* - BN's), geralmente representados por quadrados, e no outro são colocados $n-k$ nós de paridade (nós de função ou *Check Node* - CN's), habitualmente representados por círculos. Ao número de ramos que convergem num dado nó chama-se *grau* do nó e o número total de ramos do gráfico é igual ao número de 1's da matriz **H** do código. Para melhor compreensão deste conceito, observamos a fig. 2.2 que apresenta um exemplo de gráfico de Tanner.



Matriz H

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Ex.:A primeira linha da matriz corresponde a f_0: 111100

Figura 2.2 - Exemplo de um Gráfico bipartido simples de um código LDPC e a respetiva matriz de paridade

Nos gráficos de Tanner, cada BN está associado um bit de código (coluna da matriz H) e cada CN está associado uma equação de teste de paridade (linha da matriz H), podendo-se encarar cada CN como uma função com dois possíveis valores: se a respetiva equação de paridade for satisfeita vale um, caso contrário toma valor zero. Assim, estes gráficos são de grande utilidade para o processo de descodificação dos códigos, pois esta estrutura é usada na troca iterativa de mensagens (probabilidades) entre os nós com vista à determinação da mensagem. Quanto mais linhas o gráfico possuir mais mensagens são transferidas e consequentemente mais cálculos serão necessários. Os códigos LDPC, devido ao seu reduzido número de 1's na matriz H, podem ser descodificados de forma muito eficiente com algoritmos iterativos que operam sobre o grafo de Tanner do código.

2.1.3 DESCODIFICAÇÃO ITERATIVA

Na descodificação de um código, existem duas abordagens possíveis e que variam de acordo com o modelo do canal de transmissão adoptado. Quando um conjunto de símbolos à entrada do descodificador é assumido como sendo finito, o que, no caso de uma transmissão binária significa que o descodificador toma uma decisão acerca do valor "0" ou "1" de cada bit recebido, diz-se que a descodificação é do tipo *hard*.

Por outro lado, quando o alfabeto à entrada do descodificador é considerado contínuo (e.g., caso de um canal Gaussiano) e a descodificação baseia-se na distribuição probabilística de cada símbolo, sendo associado a cada bit da palavra recebida a descodificar uma probabilidade de ser "0" ou "1", a descodificação diz-se do tipo *soft*.

No contexto dos códigos LDPC ambas as abordagens de descodificação têm vindo a ser objeto de estudo. Os algoritmos iterativos *hard decoding,* normalmente designados por algoritmos de troca de bits (BF - *bit flipping algorithms*) [Richardson 2003] são menos complexos e exigem menos processamento computacional. Já os algoritmos *soft decoding* são baseados no conceito de troca de mensagens (*message passing algorithms*) e apresentam melhor desempenho ao nível de uma baixa taxa de erro de bit (*BER*), assumindo-se como os algoritmos mais vantajosos para aplicações práticas.

De seguida, será apresentado o algoritmo de descodificação iterativa *Soma-Produto* do tipo *soft decoding,* dada a sua popularidade e visto que uma parte do nosso estudo incidiu neste algoritmo bem como as suas variantes, que foram usados nos esquemas de descodificação desenhados para as novastécnicas de Jamming e codificação com interleaving propostas nesta dissertação.

ALGORITMO *SOMA-PRODUTO*

O sucesso e o reconhecimento dos códigos LDPC deve-se em parte ao algoritmo de descodificação iterativo implementado por Gallager [Gallager 62], [Gallager 63]. O algoritmo de descodificação iterativo é do tipo *soft decoding*, e tem em consideração a distribuição probabilística dos símbolos recebidos pelo descodificador. Este algoritmo é vulgarmente conhecido por Algoritmo Soma de Produtos (SPA), sendo também designado por Belief Propagation (BP).

O algoritmo *Soma-Produto* opera sobre o gráfico de Tanner de um código de bloco binário linear, e caracteriza-se como um algoritmo de passagem de mensagens entre nós. Cada nó pode ser visto como um processador de mensagens recebidas dos seus vizinhos (nós ligados), aos quais devolve posteriormente mensagens atualizadas. As mensagens recebidas por um CN, dos BN's que a ele se encontram ligados, ou as enviadas desse CN para esses BN's, não são mais do que "opiniões" sobre o valor lógico desse nó de variável. Essas "opiniões" são expressas em termos da distribuição probabilística dos símbolos recebidos pelo descodificador.

O algoritmo SPA é o que apresenta melhor desempenho na descodificação de códigos LDPC, tendo Chung, Forney, Richardson e Urbanke [Chung 2001] demonstrado ser possível aproximar-se (a menos de 0.0045dB) da capacidade de um canal AWGN considerando, no limite, um código de comprimento infinito. Nesta medida, em toda a literatura científica em que é abordada a descodificação *soft decoding* de códigos LDPC, o algoritmo SPA é descrito na forma original, e no domínio logarítmico [Frey 2001], [Ksch. 2003], [Leung 2001], [Eleftheriou 2001] e [Hu 2002].

Considerando códigos de blocos lineares, a descodificação do algoritmo SPA baseia-se na troca de informação entre nós, no âmbito de um gráfico de Tanner do código dado. Pode-se descrever o processo de descodificação em três etapas:

- **Comunicação Ascendente**: Transmissão de informação disponível em cada BN para cada um dos CN_s a ele ligados. A informação (probabilidade) enviada de BN_n para cada CN_m , denotada por $q_{nm}(b)$, com b∈{0,1}, é baseada na informação recebida do canal (amostra y_n) e na "informação extrínseca" recebida de todos os outros CN's ligados a BN_n, exceto CN_m.

- **Comunicação Descendente**: mensagem que é enviada do CN_m para BN_n, indicando a probabilidade, denotada por $r_{mn}(b)$, com b∈{0,1}, com base em todos os BN's ligados ao CN_m , com exceção do BN_n.

- **Estimação da Palavra Recebida**: Com base na informação recebida do canal, y_n, e na totalidade da informação extrínseca disponibilizada pela descodificação, calcula-se uma estimativa da razão de verossimilhança à *posteriori,* $Q_n(b)$, com b∈{0,1}, referente a cada bit c_n. Consoante a estimativa é superior ou inferior a 1, pode decidir-se que $\hat{c}_n = 0$ ou 1, respetivamente.

A Figura 2.3 representa um exemplo de gráfico de Tanner baseado num código de *Hamming* (7,4).

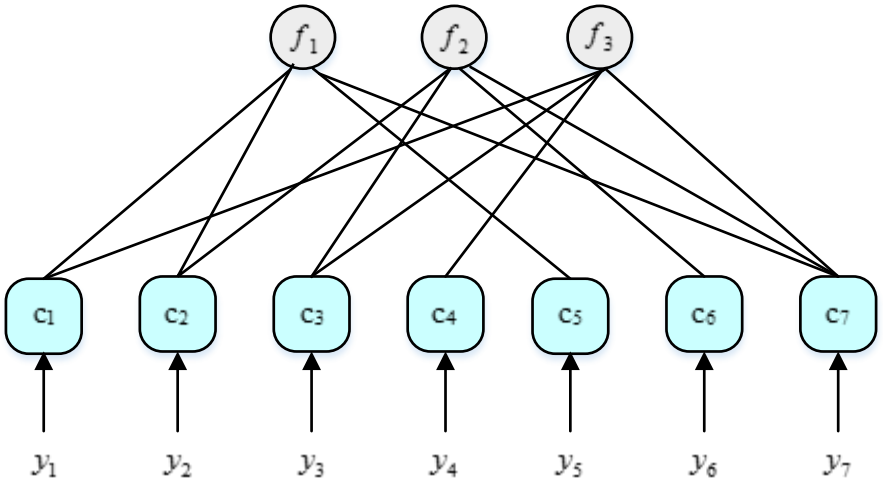


Figura 2.3 – Gráfico de Tanner associado ao código de Hamming (7,4)

Com o algoritmo SPA a descodificação processa-se de forma iterativa, envolvendo alternadamente troca de mensagens entre BN's e CN's e mensagens entre CN's e BN's no âmbito do gráfico de Tanner, as quais permitem atualizar as razões de verossimilhança à *posteriori* (LR's - *Likelihood Ratio*) estimadas de iteração para iteração, no que diz respeito aos diversos bits c_n .

Nas figuras 2.4 e 2.5 encontram-se representados os gráficos parciais, obtidos do gráfico de Tanner da figura 2.3, relativos ao processo de atualização de um CN e de um BN.

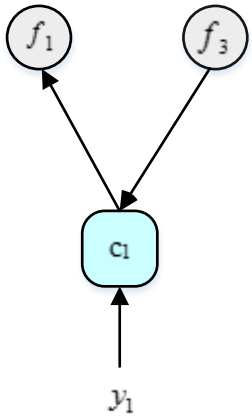


Figura 2.4

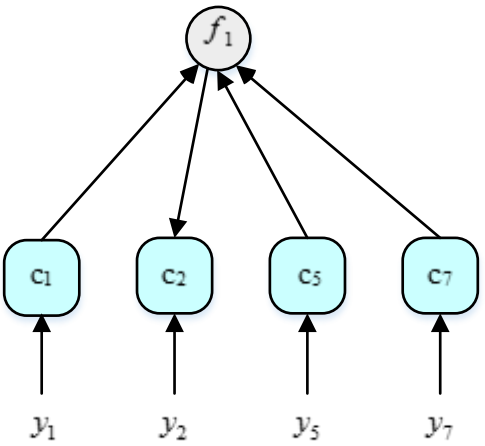


Figura 2.5

Na figura 2.4, a mensagem que o BN c_1 envia para o CN f_1 é formada pela informação y_1 recebida do canal e pela informação extrínseca recebida do CN f_3. Igualmente, na figura 2.5, a informação que o CN f_1 envia para o BN c_2 é formada pela informação extrínseca recebida dos nodos c_1, c_5 e c_7.

No âmbito do nosso trabalho, consideramos o algoritmo SPA no domínio logarítmico (LSPA - *Logarithm Sum Product Algorithm*). Como o algoritmo SPA implementado no domínio linear [Gallager 63] requer um número elevado de multiplicações, que a nível computacional são mais dispendiosas que adições, é conveniente transferir os cálculos para o domínio logarítmico, ou seja, converter as multiplicações em adições.

O algoritmo LSPA corresponde diretamente ao algoritmo SPA, substituindo as multiplicações por adições e, igualmente, os LR's por LLR's (*Log-Likelihood Ratio*). A máxima verosimilhança logarítmica de uma variável aleatória binária *b* é definida como

$$L(b) = \log \frac{\Pr(b=0)}{\Pr(b=1)}, \tag{2.7}$$

Assim, o algoritmo LSPA pode ser descrito do seguinte modo:

ALGORITMO 1 – Algoritmo LSPA

Para todos os pares (BN_n, CN_m), ou seja, todos os pares (m,n) para os quais na matriz de verificação de paridade $\mathbf{H}$ se tem $h_{mn}=1$.

0. Inicialização:

$$L(q_{nm}) = L(\mathrm{c}_n) = \frac{2y_n}{\sigma^2}$$

1. Calcular a máxima verosimilhança logarítmica (LLR's) da mensagem que o CN_m envia para o BN_n:

$$L(\mathrm{r}_{mn}) = \Big(\prod_{n'\in N(m)\setminus n} \alpha_{n'm}\Big)\Phi\Big(\sum_{n'\in N(m)\setminus n} \Phi(\beta_{n'm})\Big)$$

com

$$\alpha_{nm} = \mathrm{sgn}(\mathrm{L}(\mathrm{q}_{nm})),$$
$$\beta_{nm} = \big|L(\mathrm{q}_{nm})\big|,$$
$$\Phi(\mathrm{x}) = -\log\tanh\Big(\frac{1}{2}\mathrm{x}\Big) = \log\Big(\frac{e^x+1}{e^x-1}\Big).$$

2. Calcular a máxima verosimilhança logarítmica (LLR's) da mensagem que o BN_m envia para o CN_n:

$$L(\mathrm{q}_{nm}) = \mathrm{L}(\mathrm{c}_n) + \sum_{m'\in M(n)\setminus m} L(\mathrm{r}_{m'n}).$$

3. Calcular a máxima verosimilhança logarítmica (LLR's) das pseudo-probabilidades *à posteriori*:

$$L(\mathrm{Q}_n) = \mathrm{L}(\mathrm{c}_n) + \sum_{m'\in M(n)} L(\mathrm{r}_{m'n}).$$

4. $\forall_n$, fazer

$$\hat{c}_n = \begin{cases} 1 \Leftarrow \mathrm{L}(\mathrm{Q}_n) < 0 \\ 0 \Leftarrow \mathrm{L}(\mathrm{Q}_n) > 0 \end{cases}$$

Se a palavra descodificada $\hat{c}$ verificar as equações de verificação de paridade ($\hat{\mathbf{c}}\mathbf{H}^T = \mathbf{0}$) ou o número máximo de iterações tiver sido atingido, paramos. Caso contrário, voltar ao passo (1).

No caso de paragem, devolver:

- Palavra descodificada $\hat{\mathbf{c}}$ se $\hat{\mathbf{c}}\mathbf{H}^{\mathrm{T}} = \mathbf{0}$;
- Erro se (número de iterações=max_iterações).

2.2 INTERLEAVING

Os códigos LDPC foram desenvolvidos para a correção de erros, sendo que a ocorrência destes num canal é completamente aleatória ao longo do tempo e de certa forma reduzem o desempenho dos descodificadores, assim como a qualidade do serviço oferecido pelos sistemas de comunicação. Quando no canal é adicionado ruído, uma grande quantidade de bits contíguos pode ser afetada e assim os descodificadores, de uma maneira em geral, não têm capacidade de corrigir essas longas sequências de erros em rajada (ou *burst* de erros). De forma a contornar este inconveniente, foi proposto o espalhamento de bits através da técnica de *interleaving* (Figura 2.6). Assim, bits adjacentes não são afetados, o que facilita o processo de deteção e correção de erros. Basicamente, o *interleaving* não altera os bits (ou símbolos) de dados, nem acrescenta qualquer tipo de redundância, altera unicamente a sequência temporal dos bits (ou símbolos) de dados, facilitando assim o processo de deteção e correção de erros.

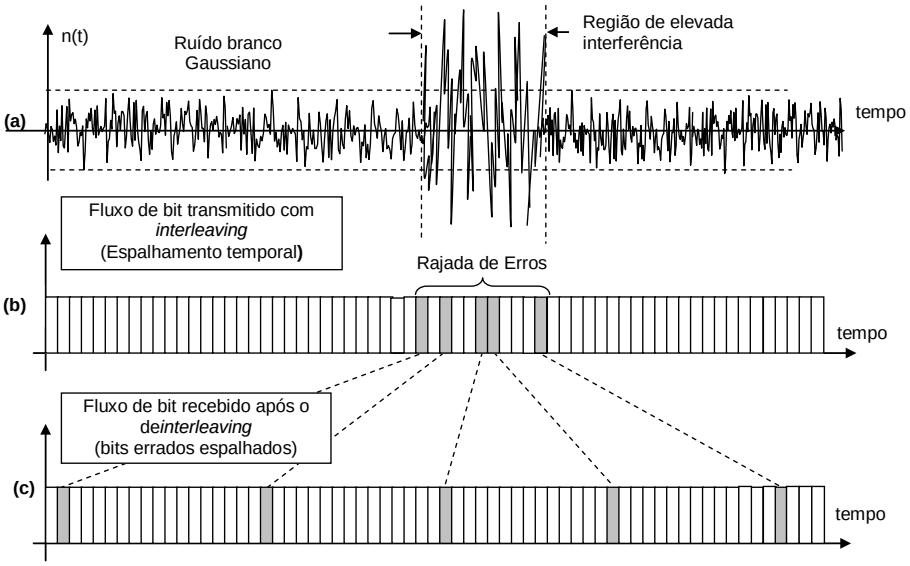


Figura 2.6 – Fluxo de bits (ou símbolos) baralhados (figura 2.6(b)), com erros em rajada, na saída do canal e fluxo de bits (ou símbolos) na saída do descodificador com de*interleaving* (figura 2.6(c)).

2.2.1 TIPOS DE INTERLEAVING

Atualmente existem dois tipos clássicos de *interleaving*: *interleaving* de blocos e o *interleaving* convolucional. No *interleaving* de blocos, os dados são escritos nas suas linhas, originando uma matriz, que posteriormente serão lidos coluna a coluna. Já no *interleaving* convolucional os bits são baralhados através de uma sequência pseudo-aleatória.

INTERLEAVING DE BLOCOS

Nesta abordagem, os bits (ou símbolos) começam por ser escritos linha a linha numa matriz com dimensão $L=T \times N$, em que a dimensão vertical T é o número de linhas, e a dimensão horizontal N é o número de colunas e, portanto, L é a capacidade total. A dimensão vertical T também é chamada de profundidade do código de *interleaving*. Assim, o embaralhamento de bits é obtido fazendo-se simplesmente a leitura desta matriz, agora coluna por coluna, como é mostrado na Figura 2.7. O desembaralhamento no recetor, tecnicamente conhecido como *deinterleaving*, é feito de modo inverso, inserindo-se agora a informação na matriz $L= T \times N$, coluna por coluna, e seguidamente será feita uma leitura da matriz linha por linha.

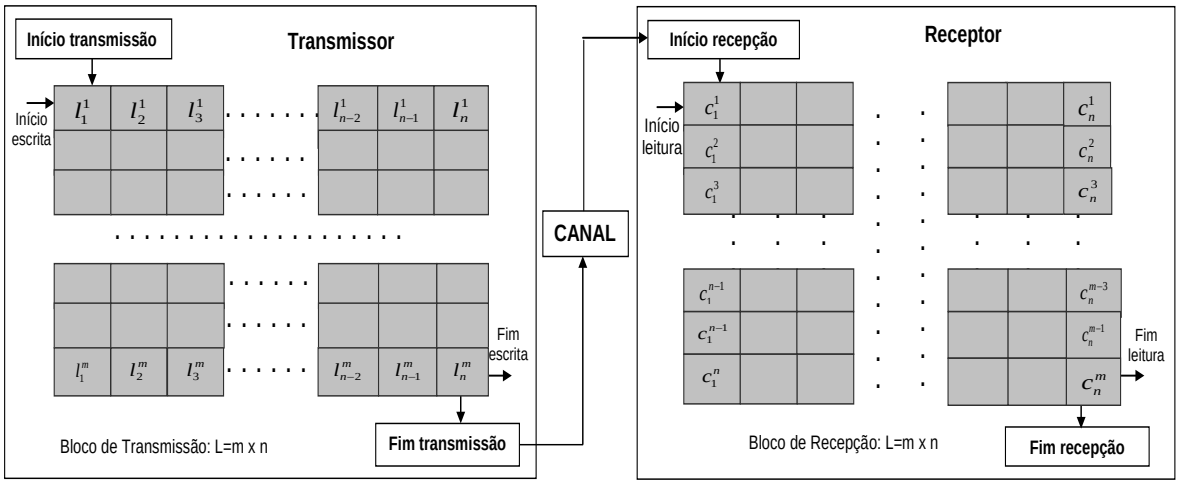


Figura 2.7 – *Interleaving* em bloco L=m x n bits (ou símbolos), através de escrita sequencial nas linhas horizontais e posterior leitura coluna por coluna.

Uma das grandes desvantagens dos códigos com *interleaving* de bloco é o atraso que introduzem. O atraso *A* pode ser calculada pela expressão:

$$A = 2TN = 2L \tag{2.8}$$

O atraso é diretamente proporcional a duas vezes o tamanho do bloco de memória do *interleaver*. Quanto maior *L* maior a rajada que pode ser admitida e, consequentemente, o atraso introduzido no sistema será maior.

INTERLEAVING CONVOLUCIONAL

No *interleaving* convolucional (Figura 2.8), os símbolos são deslocados sequencialmente para dentro de um banco de *T* registos de deslocamento, que têm a capacidade de atrasar a sequência de bits. Cada um desses registos sucessivos, tem mais *M* símbolos de capacidade que o precedente. Como pode ser visto através da Figura 2.8, o registo de deslocamento de ordem zero não tem capacidade de armazenamento, sendo então o símbolo transmitido de imediato.

Com a chegada de um novo símbolo M, o comutador muda para outro registo e, após percorrer todos os registos (*T*-1 no total), o comutador regressa ao registo de ordem zero e recomeça um novo processo.

No *deinterleaving* realiza-se a operação inversa e os comutadores de entrada e saída (tanto no *interleaving* como no *deinterleaving*) devem estar sincronizados.

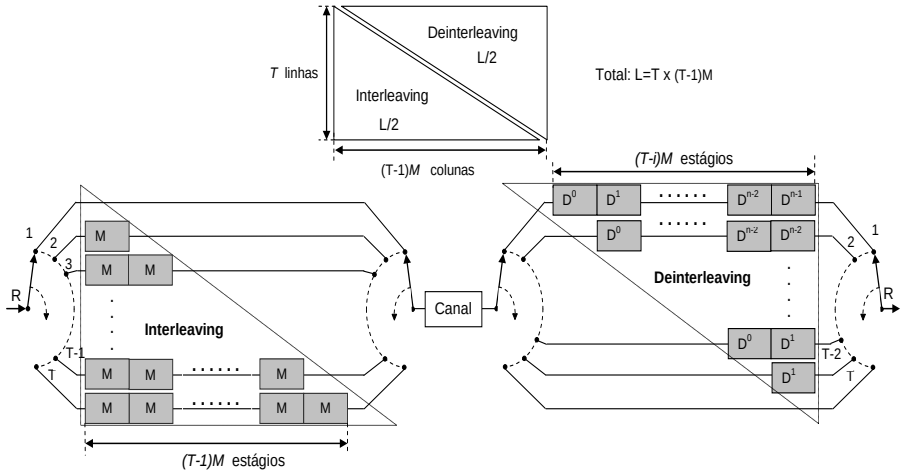


Figura 2.8 **–** Codificador e descodificador com *interleaving* do tipo convolucional, conforme [Hanna 93].

O i'ésimo deslocador do *interleaver* possui então (i-1)M estágios de deslocamento sendo que 1<i≤T. Da mesma forma, o i'ésimo deslocador do *deinterleaver* será composto de (T-i)M estágios. Um qualquer símbolo até chegar à saída, passará, portanto por um total de estágios dado por:

$$(i-1)M + (T-i)M = M(T-1) \tag{2.9}$$

Posto isto, o desempenho do *interleaving* convolucional é muito semelhante ao de *interleaving* de blocos, mas o surgimento destes veio combater as principais desvantagens dos códigos de *interleaving* de blocos, ou seja, contornar a questão do atraso e do tamanho da memória. Assim, o atraso total *A* é dado por:

$$A = T \times [(i-1)M + (T-i)M] = N(T-1) \text{ , em que } N = TM \tag{2.10}$$

Assim, o valor $N(T-1)$ representa o atraso total de um símbolo, medido pelo número total de tempos de símbolo, mas também representa o número total de elementos de memória do interleaver/deinterleaver. Observa-se então, que há uma redução para cerca de metade no atraso e nas exigências de memória relativamente ao *interleaving* de blocos.

2.2.2 CONTEXTUALIZAÇÃO

No nosso trabalho, a utilização do *interleaving* (é apenas utilizado um *interleaving* convolucional) tem um objetivo diferente do original. Propomos um esquema de codificação baseado na combinação de mecanismos de *interleaving* com códigos LDPC poderosos, a fim de melhorar segurança com um custo de energia reduzido.

Será gerada uma chave de *interleaving* aleatória que é usada para baralhar a informação na fonte (Alice). Esta chave é partilhada com o Bob (recetor legítimo), podendo o Eve receber uma chave que se espera degrada devido à interferência por parte do *jammer*, conforme descrito no capítulo seguinte. A mensagem original é então baralhada, sendo depois codificada por Alice e enviada para o Bob juntamente com a chave *de interleaving.* O Bob, inicia o processo de descodificação e *deinterleaving,* com auxílio da respetiva chave de desembaralhamento, obtendo então a mensagem com a sequência original de bits transmitidos. Por outro lado, como no canal do Eve há uma adição extra de ruído, este recebe uma versão degradada da chave que limita a capacidade do *deinterleaving* extrair corretamente a informação original.

CAPÍTULO 3

INTRODUÇÃO DE RUÍDO NO SISTEMA ATRAVÉS DE TÉCNICAS AVANÇADAS DE *JAMMING*

3.1 INTRODUÇÃO

Neste capítulo será abordado o conceito de *jamming* e de que forma este se apresenta vantajoso para o nosso projeto. Serão também apresentados alguns modelos típicos de ataques *jamming* que serviram de base para o desenvolvimento de um simulador que implementa as mesmas. Este simulador é por nós utilizado na aplicação de ruído adicional (sob diversas formas) ao canal do *eavesdropper*, garantindo desta forma que o Eve possui um canal mais degradado quando comparado com o canal do Bob, assegurando que o Bob possui a vantagem necessária para os esquemas de segurança através da camada física que serão descritos.

3.2 DIFERENTES ESTRATÉGIAS PARA APLICAR *JAMMING*

A natureza de transmissão do meio da rede sem fio leva a uma série de questões ao nível da segurança. Em particular, torna-se difícil limitar o acesso a estas redes e há facilidade por parte do Eve para aceder a toda essa informação. Adicionalmente, se o Eve for passivo, em geral, a sua localização e até mesmo a sua presença, é desconhecida.

Através dos resultados obtidos por Csiszar [Csiszar 78], mostrou-se que a comunicação segura pode ser possível se o canal do Eve for mais degradado que o canal do recetor. Contudo, no geral, não há garantias que o Bob vá ter um canal melhor que o do Eve.

No âmbito desta dissertação, introduz-se ruído no sistema com vista a garantir que as condições de segurança na comunicação sejam satisfeitas. O ruído produzido provocará degradação no canal do Eve, afetando o mínimo possível o canal do Bob. A ideia básica consiste no envio de sinais de *jamming* adequados de forma a provocar interferência no Eve. Para tal, diferentes políticas para a seleção dos *jammers* que estão ativos podem ser escolhidos [Vilela 2012], [Vilela 2011]. Nas comunicações sem fio tradicionais, a interferência é tipicamente indesejável e deve ser evitada. No nosso caso, o ruído é utilizado de forma construtiva, na medida em que aumenta o grau de segurança, causando interferência a *eavesdroppers*.

Existem diferentes estratégias de ataques que um *jammer* pode empregar para interferir nas comunicações sem fio. É impraticável descrever todos os possíveis modelos de ataques que podem existir mas, segundo Xu [Xu 2005], propõe-se os seguintes tipos de *jammers*:

- *Constant jammer* (ou jammer ativo);
- *Random jammer*;
- Jammer reativo.

Importa referir que os *jammers* propostos por Xu aplicam-se a uma área distinta daquela que o nosso trabalho abrange. O interesse nesta área serviu única e exclusivamente para adquirirmos conhecimentos acerca das variadas formas de aplicar *jamming* e assim garantirmos uma comunicação segura, enquanto o conceito apresentado por Xu é baseado nas diferentes formas de um intruso (neste caso é denominado por *jammer*) atacar uma rede. Segue-se então a descrição dos vários tipos de jammers:

1. ***Jammer* constante*:*** emite continuamente um sinal de rádio, enviando bits aleatórios para o canal sem seguir padrões da camada MAC. O *jammer* constante não espera que o canal entre num período de inatividade para começar a transmitir. Este ataque tem a capacidade de introduzir interferência nas comunicações que estejam no canal. O consumo de energia para execução do mesmo é elevado.

2. ***Jammer* aleatório*:*** este *jammer* alterna entre dois estados, *repouso* e *jamming*. Mais detalhadamente, após fazer *jamming* por t_i unidades de tempo, o *jammer* aleatório desliga a emissão de ruído e entra em modo *repouso*. Retornará o *jamming* após ficar *repouso* por t_j unidades de tempo. De realçar que as varíaveis t_i e t_j podem tomar valores fixos ou aleatórios. Durante o estado de *jamming*, o atacante pode-se comportar tanto como *jammer* constante. A distinção entre este modelo de *jamming* e o referido anteriormente, reside no facto de o *jammer* aleatório levar em consideração a conservação da energia, especialmente para os *jammers* que não têm fonte de energia ilimitada.

3. ***Jammer reativo:*** os dois modelos acima descritos são *jammers* ativos na medida em que tentam bloquear o canal independentemente do padrão de tráfego que está implícito no canal. *Jammers* ativos normalmente apresentam maior eficácia porque mantém o canal ocupado durante todo o tempo. Mas há uma contrariedade nestes métodos, são facilmente detetáveis. Uma

alternativa de *jamming* nas comunicações sem fios reside em empregar uma estratégia reativa. No *jammer reativo* não é necessário interferir no canal quando não existe qualquer tipo de comunicação. Posto isto, o *jammer* mantém-se hibernado enquanto o canal está inativo mas assim que qualquer tipo de atividade seja detetada sobre o sinal, este inicia de imediato a transmissão de ruído. Assim, um *jammer reativo* espera primeiro pela receção de uma mensagem e só depois é que atua. Este modelo apresenta algumas vantagens quando comparado com os modelos mencionados anteriormente. Para além de serem de difícil deteção, também são capazes de reduzir a vazão e a taxa de entrega dos nós, e ao mesmo tempo consumir menos energia, aumentando o seu tempo de vida na rede.

3.3 SIMULADOR COM DIFERENTES TIPOS DE *JAMMERS*

A criação do simulador teve por base os conceitos dos modelos de ataque de *jamming* acima referidos. Contextualizando para o nosso trabalho, o conceito de *jamming* ganha preponderância na medida em que é utilizado para introduzir ruído no sistema e com isso obter segurança e confidencialidade.

O desenvolvimento do protótipo seguiu aspetos a ter em consideração: diferentes estratégias de gerar ruído Gaussiano (AWGN) conjugadas com formas distintas de o introduzir no sistema. Com este simulador foram implementados diferentes formas de aplicar *jamming* ao canal do Eve, por forma a garantir uma degradação do mesmo quando comparado com o do Bob.

De seguida, será apresentado mais em detalhe os diferentes tipos de *jammers* desenvolvidos, assim como as suas principais características:

- **Jammer Aleatório:** Ao invés de enviar continuamente um sinal, um random jammer alterna entre dois estados: *repouso* e *jamming*. Ou seja, após a aplicar *jamming* em *t* bits, este desliga-se e deixa de emitir ruído, entrando no modo *repouso*. Retomará o *jamming* após ficar em *repouso* por *k* bits. Do conceito deste jammer convergem duas estratégias diferentes de introdução de ruído, uma com *t* fixo e outra com *t* variável. Para *t* fixo, temos o jammer **Aleatório de tempo fixo** que se vai ativando aleatoriamente, com duração constante de *t* bits, até perfazer o tempo total de *jamming* (em %) imposto por nós. No caso do jammer **Aleatório de tempo variável,** também se ativa aleatoriamente mas, para cada período de emissão, *t* tem uma duração variável aleatória entre t_{min} e $t_{máx}$.

- **Jammer Periódico:** Este jammer também vai alternar entre os dois estados, *repouso* e *jamming*. Ativa-se periodicamente de *t* em *t* bits e após a sua ativação vai aplicar *jamming* a *k* bits consecutivos.

- **Burst Jammer:** Ativa-se de forma aleatória e mantém-se ativo ao longo de um tempo *t*. Após a introdução de ruído, entra no estado *repouso* durante os restantes *k* bits da palavra de código. Este *jammer* tem um comportamento idêntico ao do Aleatório de tempo fixo, só que em vez de se ir ativando várias vezes ao longo tempo, ativa-se uma única vez com uma duração de *t* bits. De salientar, que esta duração é o tempo total de *jamming* (em %).

- ***Constant* Jammer:** Também conhecido por *jammer* ativo, este emite sinal continuamente ao longo de todo o canal.

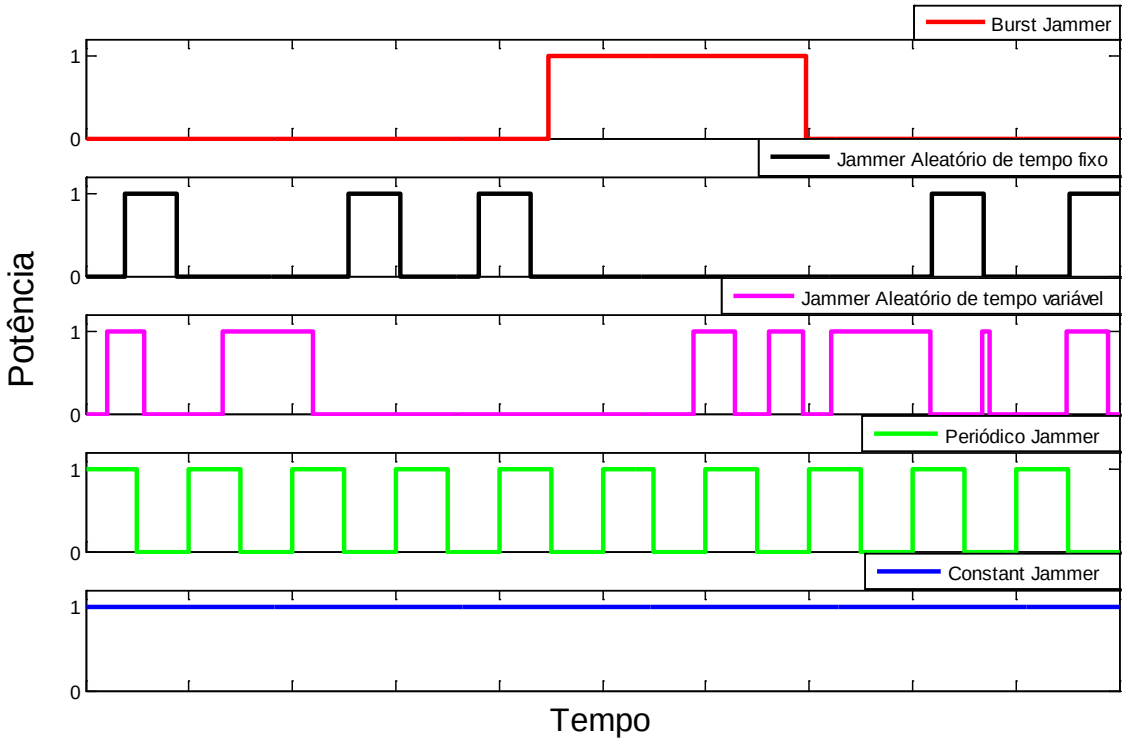


Figura 3.1 – Esquematização do comportamento de cada *jammer*

Com base no simulador desenvolvido, procedeu-se a alguns testes para sua validação e para compreendermos o comportamento de cada *jammer* (Figura 3.1). Para esta experiência, considerou-se a potência de cada *jammer* unitária e um *bit stream* de 10^4 bits.

Para o Burst Jammer, considerou-se um tempo total de *jamming* de 25%. Assim que o *jammer* se ativa (de forma aleatória), introduz-se ruído em 25% dos bits, ou seja,

afeta os 2500 bits consecutivos. Após este processo, o *jammer* entra no estado *repouso* para os restantes bits.

No Aleatório de tempo fixo**,** também se considerou um tempo total de *jamming* de 25% (2500 bits). Para cada período de emissão de *jamming,* tem-se uma duração fixa de 500 bits e o jammer vai-se ativando aleatoriamente, com essa duração constante de bits, até completar o tempo total de *jamming*. Ou seja, o *jammer* continuará ativar-se (aleatoriamente) enquanto não forem afetados com ruído 25% dos bits.

No caso do Aleatório de tempo variável temos os seguintes parâmetros:

- ✓ $t_{\min} = 50$ bits;
- ✓ $t_{\max} = 1000$ bits;
- ✓ tempo total de *jamming* = 50%;

Este *jammer* também se ativa aleatoriamente mas, para cada período de emissão de ruído, *o jammer* atua durante um período de tempo, variável e aleatório, compreendido entre 50 e 1000 bits. O *jammer* só entrará em modo *repouso* definitivo assim que 50% dos bits sejam afetados.

O *jammer* Periódico ativa-se periodicamente de 1000 em 1000 bits e, após a sua ativação, aplica *jamming* nos 500 bits que se encontram imediatamente a seguir.

Para o *Constant jammer*, como o próprio nome indica, mantém-se ativo de forma constante ao longo do tempo.

CAPÍTULO 4

SEGURANÇA EM REDES SEM FIOS ATRAVÉS DE TÉCNICAS AVANÇADAS DE JAMMING E CODIFICAÇÃO

Neste capítulo, propomos dois esquemas de codificação baseados na combinação de *interleaving* com códigos de bloco sistemáticos e poderosos (LDPC) a fim de melhorar a segurança no meio sem fio e com um custo de energia reduzido. Nestes esquemas, o *interleaving* é usado também para fins de segurança. Estes esquemas são baseados na troca de uma chave de *interleaving* aleatória durante um curto período de comunicação vantajoso sobre o *eavesdropper* (que pode ser obtido, por exemplo, através da utilização de *jammers*, conforme descrito anteriormente). Esta chave é utilizada na fonte para embaralhar os dados originais antes de serem enviados para o canal. A chave é obtida no lado do recetor, juntamente com o *interleaving* dos dados e, posteriormente, utilizada para o desembaralhar os dados transmitidos e descodificados. Desde que o *eavesdropper* receba uma versão degradada da chave, esta limita a sua capacidade de desembaralhar os dados originais.

4.1 *INTERLEAVING* ALEATÓRIO PARA SEGURANÇA

Considere uma variante do modelo *Wiretap Channel*, como o ilustrado na Figura 4.1. O transmissor Alice quer enviar uma mensagem M para o recetor legítimo Bob enquanto um intruso Eve está a captar informações. Ao contrário da configuração de *Wiretap Channel* típica, uma chave de *interleaving* K é passada ao codificador e concatenada com a mensagem original M antes de ser enviada para o canal pela Alice, com potência de transmissão P_a. Adicionalmente considera-se a presença de um *jammer* que provoca interferências (ruído Gaussiano branco extra aditivo) com potência de transmissão $P_j = \alpha P_\alpha$ (uma fração α da potência de transmissão de Alice). O *jammer* está ativo apenas durante a transmissão da chave de *interleaving*, com o objetivo de induzir um canal degradado para o *eavesdropper*, mas também pode causar alguma interferência para a chave no seu caminho para Bob (daí a existência de um canal degradado para ambos).

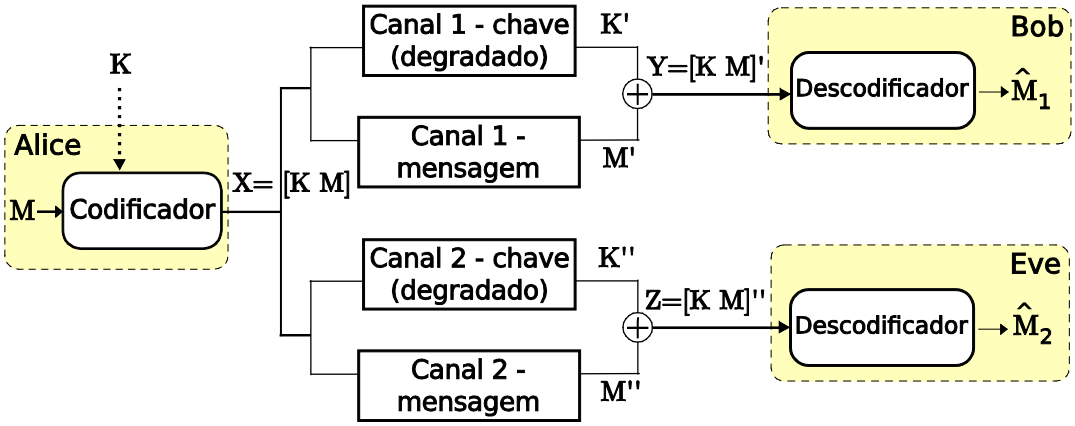


Figura 4.1 - Variante do *Wiretap Channel* onde uma chave de *interleaving* K é enviada com a mensagem original M. A chave é enviada durante um período em que um jammer está ativo e, portanto, os canais são degradados durante a transmissão da chave.

Consideremos que $\dot{X}$, $\ddot{X}$ e $\dddot{X}$ que representam um bloco de dados X , que foi descodificado uma e duas vezes, respetivamente. X 'representa um bloco de dados, que pode ter sido alterado durante a passagem através de um canal, enquanto $\hat{X}$ corresponde a uma aproximação da informação original X obtida no destino. Finalmente, consideramos as funções de *interleaving* e *deinterleaving*, denotadas por inter() e deinter() respectivamente, que realizam uma permutação aleatória das informações recebidas de acordo com um procedimento de *interleaving* de blocos convencional, onde o conjunto de símbolos/bits da mensagem são reorganizados sem repetir ou omitir qualquer um dos símbolos no conjunto, de acordo com uma tabela definida pela permutação da chave K.

Consideramos o Eve como um adversário passivo com as mesmas capacidades que Bob. Em particular, o Eve tem conhecimento dos processos de codificação e descodificação e é capaz de descodificar a informação original se os dados são recebidos com níveis de erro suficientemente baixa.

Com base na variante do modelo *wiretap channel* (Figura 4.1) e da conjugação de códigos LDPC, *interleaving* e *jamming*, apresentamos agora dois esquemas de codificação/descodificação:

- o primeiro baseado na utilização de um código exterior (para fiabilidade na transmissão da chave) conjugado com um código interior (para fiabilidade na transmissão dos dados);
- o segundo baseado em códigos concatenados, que permite melhorar a confidencialidade da informação transmitida mas que tem a particularidade de requerer a inversão da ordem de descodificação com o código exterior (chave) a ser descodificado antes do código interior (dados).

4.1.1 PRIMEIRA ABORDAGEM

Na Figura 4.2 são apresentados os processos de codificação e descodificação para a primeira abordagem. O codificador e o descodificador são os mesmos tanto para o Bob como para o Eve, a única diferença reside na qualidade dos canais. O esquema tem como objetivo garantir fiabilidade para Bob e confidencialidade contra o Eve, explorando estas diferenças de canal.

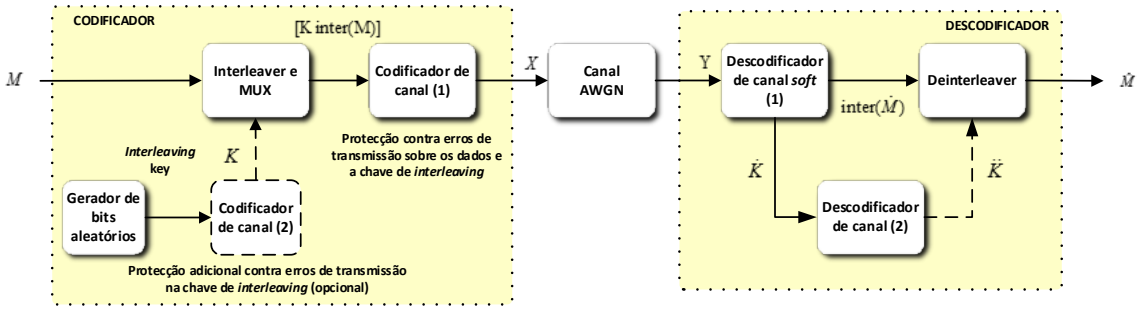


Figura 4.2 – Processos de Codificação e Descodificação referentes à primeira abordagem

A transmissão fiável da mensagem M é assegurada através do uso de um código poderoso sistemático de bloco linear C1, com dimensões (η_1, k_1). Para cada mensagem, é gerada uma chave diferente de *interleaving* aleatória K, com tamanho η_K, que é usada para embaralhar o conteúdo de M. O bloco concatenado [K inter(M)] é depois codificado por C1 para produzir a palavra de código X que é enviada através de um canal com ruído AWGN.

Por fim o descodificador, Bob e Eve realizam descodificação iterativa, com base em *soft decoding*, da palavra recebida Y (Z para o Eve), produzindo uma estimativa da mensagem que sofreu *interleaving*, inter($\dot{M}$). A chave também é descodificada ($\dot{K}$) e usada para desembaralhar inter($\dot{M}$), resultando em $\hat{M}$.

A determinação correta da chave de *interleaving* é crítica para o desempenho deste sistema, pois um único erro em $\dot{K}$ pode produzir uma aproximação $\hat{M}$ completamente diferente da informação da mensagem original M. Para combater este problema, um grau adicional de fiabilidade pode ser obtido através do uso de um código de correção de erro curto C2 sobre a chave K, como pode ser visto na Figura 4.2.

O uso de codificação sistemática permite a um *jammer* causar interferência somente durante um curto período de transmissão da chave de *interleaving* K. Para isso, o jammer pode coordenar com Alice usando um esquema de sinalização tal como proposto em [Vilela2 2011]. A segurança deste esquema é conseguida à custa de um

decréscimo na taxa de informação sendo a taxa de código útil $R_u = \frac{k_1 - \eta_k}{\eta_1}$ (onde $\eta_K < k_1$ é o comprimento de K).

Há também um custo de energia associado ao *jamming*, que pode ser medido como a energia do *jammer* por bit de informação, E_{Jb} , normalizada para a energia de Alice por bit de informação, E_b , como se segue

$$\frac{E_{Jb}}{E_b} = \frac{\frac{\eta_K P_j}{k_i - \eta_K}}{\frac{\eta_i P_a}{k_i - \eta_K}} = \frac{\alpha \eta_K}{n_i} \qquad (4.1)$$

4.1.2 SEGUNDA ABORDAGEM

A Figura 4.3 refere-se aos processos de codificação e descodificação para o segundo esquema. De forma idêntica ao primeiro esquema, tanto o codificador como o descodificador são os mesmos para o Bob e Eve. Com este esquema pretende-se também garantir fiabilidade para o Bob e confidencialidade em relação a Eve, assumindo-se uma vantagem no canal para o Bob durante a transmissão de uma chave de *interleaving.* A mensagem M é transmitida de forma segura através de um código sistemático de bloco linear C1, de dimensões (η_1, k_1). Para cada mensagem M é gerada uma diferente chave de *interleaving* K, com tamanho η_K, (que é binária e aleatória, tal e qual como no primeiro esquema) que é concatenada à informação M. O bloco concatenado [K M] é depois codificado por C1 para produzir um outro bloco [K M p], onde p corresponde aos bits de paridade de [K M]. De seguida a chave, K é usada para embaralhar uma parte da palavra de código produzida correspondente aos dados e aos bits de paridade, i.e. [M p], sendo desta forma produzindo a palavra de código Y ([K X=inter(M,p)]) que é enviada através do canal AWGN.

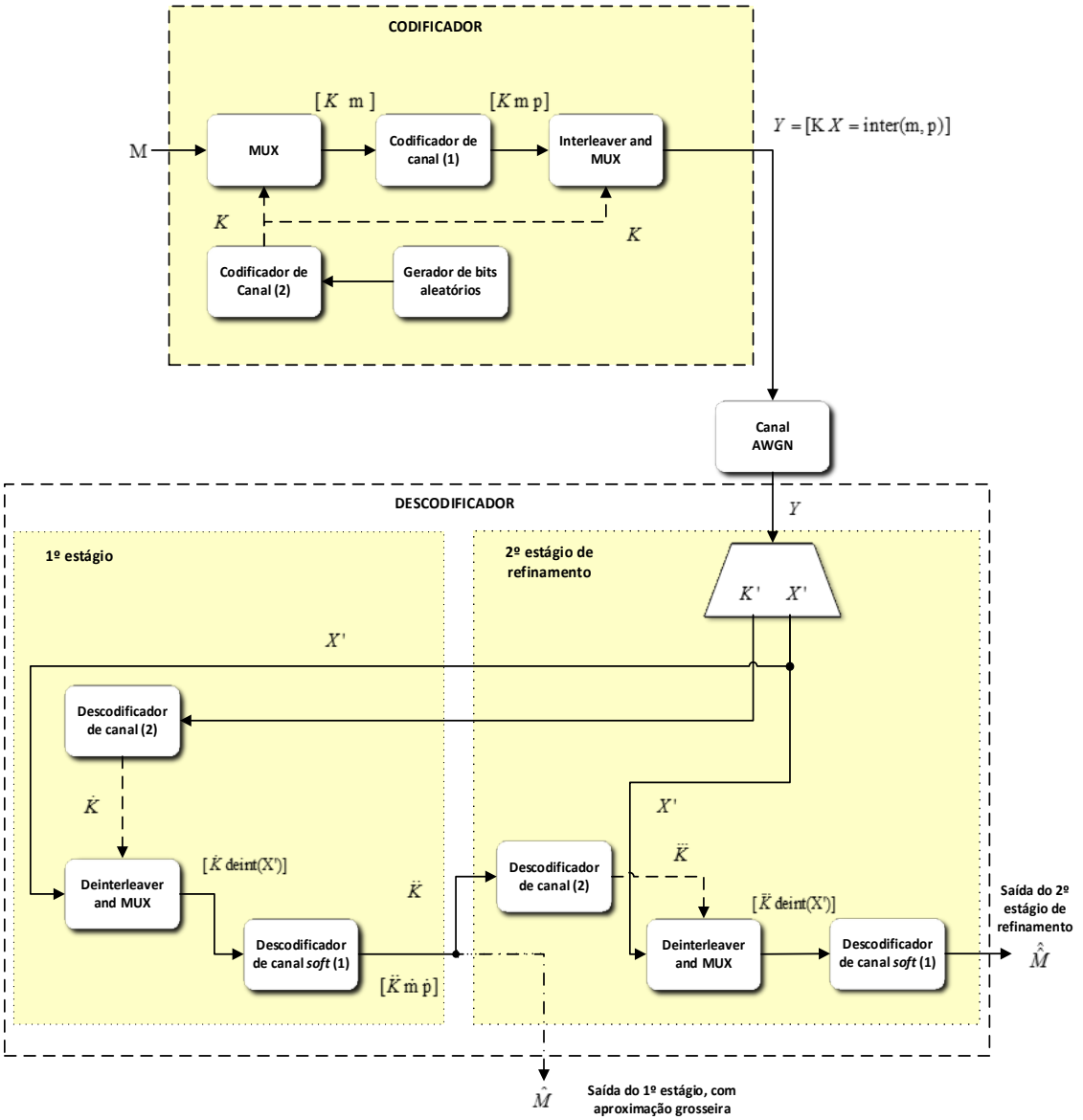


Figura 4.3 – Processos de Codificação e Descodificação referentes à segunda abordagem

Descodificação

A descodificação é iterativa e do tipo *soft decoding*, mas além disso este esquema apresenta uma particularidade. Não é possível ao descodificador C1 descodificar previamente Y sem que tenha conhecimento da chave. No entanto sendo o código C1 sistemático e sendo a chave K enviada de forma inalterada pelo sistema de codificação é possível descodificar a chave de *interleaving* através do código curto C2 antes de descodificar os dados através do código longo C1. Esta estratégia permite obter uma melhor aproximação da chave de *interleaving* antes de se proceder ao desembaralhamento e descodificação dos dados. Mais ainda, este processo pode ser

melhorado ao ser efetuado em dois estágios: um primeiro estágio que permite obter uma aproximação grosseira, que é posteriormente refinada num segundo estágio.

Para melhor compreensão do sistema descodificador representado na Figura 4.3, apresenta-se na Figura 4.4 o fluxo de dados gerados nos dois estágios do processo de descodificação. Visto que estamos perante uma descodificação iterativa do tipo *soft decoding*, as variáveis à entrada de cada descodificador estão sob a forma de LLR's, ou seja, máxima verosimilhança logarítmica. Para tal definiram-se algumas variáveis:

- L_K são os LLR's correspondentes à chave de *interleaving* K ;
- L_X são os LLR's referentes à estimativa dos bits que sofreram *interleaving*;
- L_m correspondem aos LLR's da estimativa da mensagem original M ;
- L_p refere-se aos LLR's da estimativa dos bits de paridade.

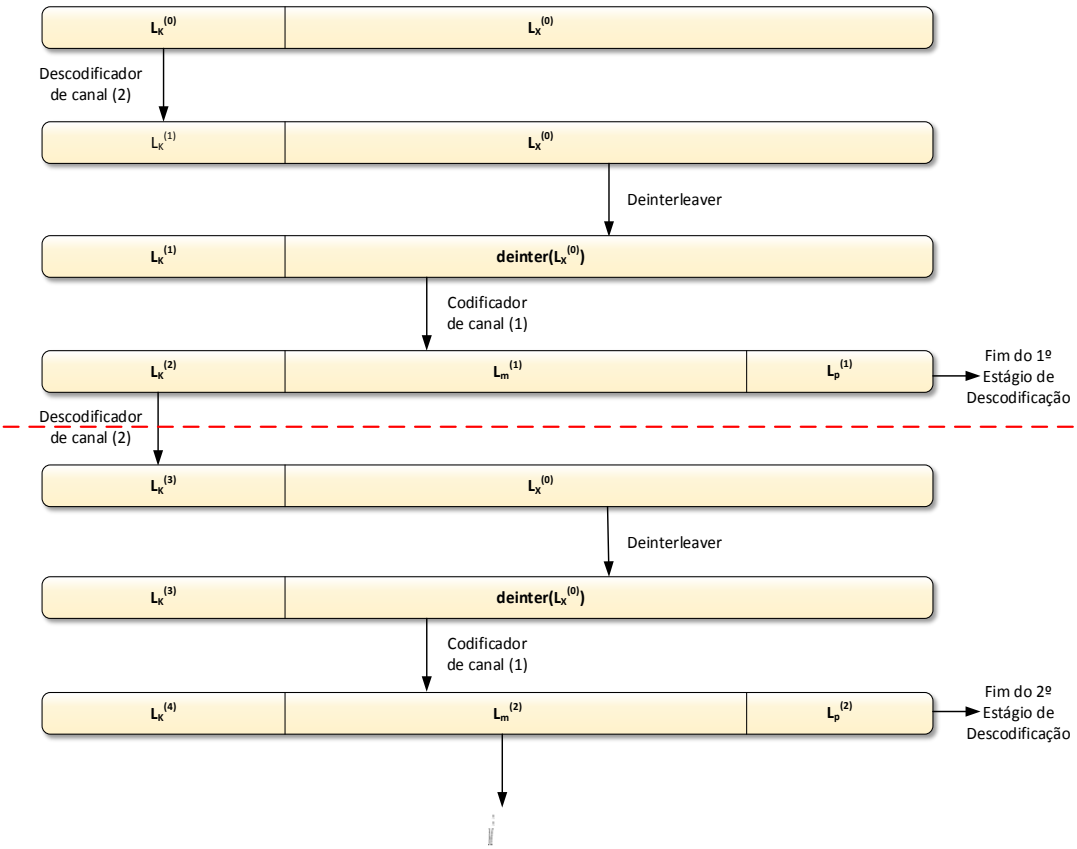


Figura 4.4 – Diagrama de fluxo de dados gerados nos dois estágios do processo de descodificação da segunda abordagem

No primeiro estágio, a partir da palavra recebida Y , vinda do canal AWGN, são geradas duas estimativas: uma estimativa da mensagem que sofreu *interleaving*, X ', e outra corresponde à chave de *interleaving* K '. A chave é descodificada ($\dot{K}$) através de C2, e usada para desembaralhar inter($\dot{X}$). Após o bloco concatenado [$\dot{K}$ deinter($\dot{X}$)] ser descodificado por C1, resulta um novo bloco [$\ddot{K}$ $\dot{M}$ $\dot{p}$] e , portanto, o estágio de aproximação grosseira está completo, com uma primeira estimativa da mensagem original $\hat{M}$.

Para finalizar o último processo de descodificação, a estimativa da chave de *interleaving* $\ddot{K}$ que está à saída do primeiro estágio, é descodificada e será novamente usada para desembaralhar a mensagem e bits de paridade originalmente recebidos, i.e. inter($\dot{X}$). O bloco concatenado [$\ddot{K}$ deinter($\dot{X}$)] é descodificado por C1 e dá origem à estimativa final da mensagem original $\hat{\hat{M}}$. Importa referir que todas estas etapas de codificação e descodificação referidas se aplicam de igual forma tanto para o Bob como para o Eve.

Esta implementação apresenta algumas particularidades que merecem ser discutidas, visto que de certa forma caracterizam o desempenho deste esquema. Uma delas está relacionada com a implementação de dois estágios de descodificação, um de aproximação grosseira e outro de refinamento. Este último não é mais nem menos que uma repetição do primeiro estágio. Assim, permite-nos obter uma versão melhorada da chave de *interleaving*, visto que esta é submetida a uma descodificação adicional partindo de uma estimativa de maior fiabilidade $\ddot{K}$, que resultou de duas etapas de descodificação anteriores por C2 e C1 . Desta forma, com a introdução de duas etapas de descodificação pretende-se obter aumento do grau de fiabilidade quando comparado com o primeiro esquema.

Outro factor a ter em consideração, está associada aos bits que sofrem *interleaving.* Para além da mensagem original M, também os bits de paridade à saída de C1 sofrem *interleaving*, o que permite aumentar o grau de confidencialidade do sistema. É sabido que bom desempenho de descodificação dos códigos LDPC depende da correta estimação da fiabilidade (i.e. probabilidades ou LLR) da informação recebida. O facto de no Eve haver uma interferência adicional sobre a chave de *interleaving* K, e de a descodificação ter de ser iniciada pelo código C2 (um código mais fraco), é de esperar que dê origem a uma taxa de erros na descodificação de K muito superior à observada pelo BOB, levando à divergência da etapa seguinte de descodificação C2 e como tal à obtenção de uma estimativa errónea de $L_k^{(2)}$. Esta estimativa é essencial para o sucesso do segundo estágio de descodificação, pelo que operando sobre um $L_k^{(2)}$ erróneo o Eve apenas sofre mais confusão, o que providencia o desejado aumento de confidencialidade ao sistema.

CAPÍTULO 5

RESULTADOS EXPERIMENTAIS

Neste capítulo apresentamos resultados experimentais obtidos para os esquemas descritos no capítulo 4. Começamos por fazer referência a medidas de desempenho de códigos, seguindo-se a descrição das condições de simulação e, por fim, a análise e interpretação dos resultados.

5.1 ANÁLISE DE DESEMPENHO E COMPARAÇÃO DE CÓDIGOS

A comparação de códigos com diferentes dimensões e taxas de informação, torna necessário a medida da taxa de erros de transmissão função de uma métrica comum característica dos sistemas de comunicação. A medida mais comum é a razão entre a energia gasta por bit de informação enviado (denotada por E_b) e a densidade espectral de potência de ruído AWGN (denotada por N_0). Esta relaciona-se com relação sinal ruído (SNR) do canal, um importante parâmetro a especificar aquando da realização de simulações, estando relacionada com as características de diferentes componentes do sistema de comunicação (e.g. modelador e código).

Assim nesta secção descrever-se-á de forma sucinta os componentes bases de um sistema de comunicação digital, e de que forma o desempenho do mesmo pode ser avaliado.

5.1.1 SISTEMAS DE COMUNICAÇÃO DIGITAIS

A codificação de canal assume um papel fundamental nos sistemas de comunicação digitais atuais e, neste contexto, a forma como se relacionam diversos blocos torna claro o método como é avaliado o desempenho de um código de correção de erros e como podem ser comparados códigos com diferentes taxas de informação. O diagrama de blocos típico de um sistema deste tipo encontra-se representado na Figura 5.1.

A fonte de informação discreta emite símbolos que são codificados pelo codificador de fonte, de acordo com um dado alfabeto código. O codificador de fonte procura remover toda a redundância estatística presente nos símbolos emitidos pela fonte reduzindo, desta forma, o débito simbólico (de informação útil) $\mathbf{r_s}$ imposto ao sistema transmissor.

O codificador de canal, por sua vez, introduz redundância (de forma controlada) nas mensagens a transmitir por forma a aumentar a sua imunidade ao ruído do canal. Assim, admitindo um alfabeto binário e considerando o caso de um código de bloco (η, k) às mensagens, $\mathbf{m}$, de comprimento k, este faz corresponder palavras de código, $\mathbf{c}$, de comprimento n, pelo que o débito simbólico à saída passa a ser, $r_c = r_s / R$, com $R = k/n$ a taxa de informação do código.

Os símbolos produzidos pelo codificador de canal são convertidos pelo modulador em sinais que possam ser transmitidos de forma eficiente através do canal de comunicação. A escolha do esquema de modulação é, normalmente, baseada num conjunto de restrições, como sejam, a potência máxima possível de transmissão, a largura de banda disponível, entre outros. Estes sinais são transmitidos através do canal sofrendo, regra geral, diversos efeitos, como sejam, distorção, atenuação e desvanecimento [Carlson 2002].

O desmodulador opera sobre o sinal recebido, y, realizando a operação inversa ao do modulador, fornecendo uma palavra, r, ao descodificador de canal. No caso de um sistema do tipo FEC (*Forward Error Correction*), sistema que tem como principal função detetar e corrigir erros, este tenta detetar e corrigir os erros da palavra recebida, fornecendo à sua saída uma estimativa, m', da mensagem transmitida.

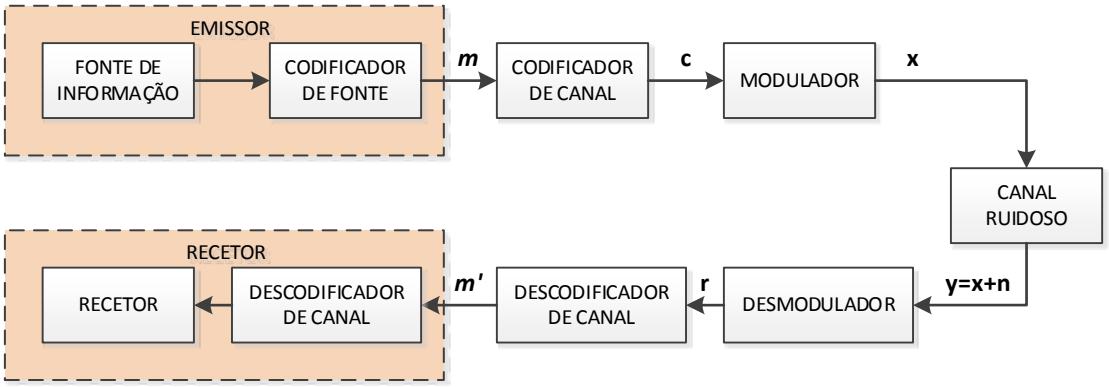


Figura 5.1 – Componentes de um sistema de comunicação digital

5.1.2 MEDIDAS DE DESEMPENHO E COMPARAÇÃO ENTRE CÓDIGOS

São várias as medidas usadas para exprimir o desempenho de um sistema de transmissão digital, sendo a mais comum a taxa de erros (BER) à saída do descodificador de canal. Esta taxa é baseada na comparação entre os dados originalmente transmitidos com aqueles que efetivamente são recebidos, após a ação de todos os processos que minimizam ou eliminam a ocorrência de erros, causados pelo ruído presente na comunicação. O BER é então rácio do número de erros de bit pelo número total de bits transmitidos da informação obtida após descodificação com eventual correção de erros.

Definida a medida de desempenho por nós utilizada, o problema que se coloca é de como comparar de forma justa o desempenho de códigos com diferentes dimensões e taxas de informação.

O uso de um código de canal com taxa de informação, $R = k/n$, tem duas consequências [Wick 95], [Bos 99]. Em primeiro lugar, o débito simbólico de transmissão aumenta passando a ser, $r_c = r_s/R$, com $\mathrm{r_s}$ e $\mathrm{r_c}$ o débito à entrada e saída do codificador de canal, respetivamente. Em segundo lugar, a energia de transmissão aumenta para $\mathrm{n\,E_s}$ em vez de $\mathrm{k\,E_s}$, sendo E_s a energia transmitida por símbolo. De forma a comparar o desempenho de diferentes esquemas de codificação, a energia do sinal passa a ser expressa em termos da energia enviada por bit de informação, $\mathrm{E_b}$,com

$$\mathrm{E_b} = \frac{\mathrm{E_s}}{\mathrm{R}} \tag{5.1}$$

Assim, considerando o caso de um canal Gaussiano, os gráficos de desempenho passam a ser expressos em termos da figura de mérito, E_b / N_o, diretamente relacionada com o SNR do canal em que:

$$\frac{E_b}{N_o} = \frac{E_s}{2R\sigma^2} = \frac{SNR}{R} \tag{5.2}$$

onde σ^2 representa a variância do ruído AWGN que é adicionado a cada símbolo da palavra de código transmitida, N_o/2 a densidade espectral de potência do ruído.

5.2 SIMULADOR E CONFIGURAÇÕES

Nesta secção são descritas de forma sumárias as principais características do simulador desenvolvido, nomeadamente o modelo de sistema e de atacante, e as opções de configuração tomadas.

5.2.1 MODELO DE SISTEMA E ATACANTE

Os esquemas desenvolvidos têm por base o modelo do *wiretap channel* apresentado na Fig. 4.1, onde existem dois canais de comunicação, um do transmissor Alice que envia a mensagem *M* com potência P_a para Bob e outro entre Alice e Eve, que se encontra à "escuta". Adicionalmente, consideramos a presença de um *jammer* que causa interferência (ruído extra) com uma potência de transmissão $P_j = \alpha P_\alpha$. O *jammer* é ativado apenas durante a transmissão da chave de *interleaving* com o objetivo de induzir um canal degradado para o Eve, mas também pode interferir na chave no caminho para o Bob. A mensagem *M* assume-se como transmitida em ambos os canais, Bob e Eve, sem interferência, desde que o *jammer* esteja inativo.

Relativamente à modulação, esta é efetuada recorrendo ao modulador QPSK (*Quadrature Phase Shift Keying*). Baseia-se no conceito de desvio de fase, em que se altera a fase da portadora num de quatro pontos igualmente espaçados no intervalo 0 e 2π. A constelação QPSK considerada e o respectivo mapeamento Gray [Carlson 2002] encontra-se representada na Fig. 5.2. O sinal modulado é pois composto por duas componentes de sinal em quadratura, sendo transmitido dois bits simultaneamente, resultando uma velocidade de transmissão igual ao dobro da velocidade de modulação, apresentando melhor imunidade a ruídos.

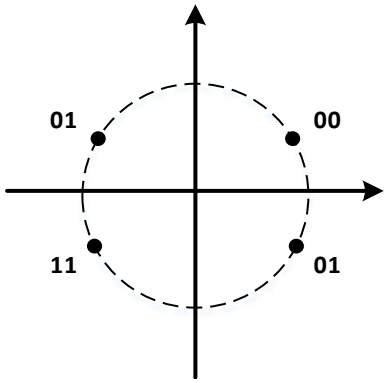


Figura 5.2 – Exemplo de constelação QPSK.

Os sistemas de codificação propostos no capítulo 4 foram implementados de acordo com os diagramas de blocos das Figuras 4.2 e 4.3, tendo sido utilizado o algoritmo SPA na descodificação dos códigos LDPC

Ambos os esquemas foram estudados nas mesmas condições de simulação (para cada código em particular), palavras de código testadas com igual comprimento (na ordem dos 10^6), corrompidas pelo mesmo nível de ruído e, ainda, o mesmo número máximo de iterações de descodificação do algoritmo SPA (50 iterações). Assim, o número de palavras testadas para cada um dos esquemas desenvolvidos foi escolhido de forma a garantir, após descodificação, um número de palavras erradas superior a 100, por forma a obter valores de BER entre 10^{-1} e 10^{-6}.

No que diz respeito ao adversário, considerámos um *eavesdropper* passivo com as mesmas capacidades do Bob. Nomeadamente, o *eavesdropper* tendo conhecimento dos processos de codificação e descodificação e sendo capaz de descodificar a informação original caso a obtivesse com níveis de erro suficientemente baixos.

Eis o conjunto de parâmetros que caracterizam as simulações em ambas as abordagens:

- Código LDPC longo (C1) com dimensões $(\eta_1, k_1) = (1056, 880)$;
- Código LDPC curto (C2) com dimensões $(\eta_K, k_K) = (288, 144)$;
- Taxa de código útil, $\mathrm{R}_u = (k_1 - \eta_K)/\eta_1 \approx 0.56$;
- Potência de transmissão do *jammer* $P_j = \alpha P_\alpha$, com $10\% \le \alpha \le 100\%$.

5.2.2 CÓDIGO DESENVOLVIDO

O nosso código foi desenvolvido exclusivamente em MATLAB®. A imensa variedade de funções predefinidas por este *software* levou-nos aproveitar esse fator para utilizar algumas delas no desenvolvimento do código. Apresentemos algumas das funções de maior relevância e que se relacionam diretamente com o tema principal do nosso trabalho:

- Para gerar ruído AWGN (tanto para os canais de comunicação como para o simulador de *jamming*) utilizamos a função *randn()*, que devolve uma matriz (dimensões ao critério do utilizador) que contém valores pseudo-aleatórios obtidos a partir de um padrão de uma distribuição normal;
- Com o objetivo de gerar palavras com uma sequência de bits aleatórios, usou-se o *randint()*. Esta gera uma matriz de inteiros aleatórios uniformemente distribuídos, ou seja, gera um "0" ou "1" com igual probabilidade;
- O *randintrlv()* foi a função que adaptamos para a aplicação de *interleaving* aos variados fluxos de dados. Tem a capacidade de reorganizar os elementos de dados usando uma permutação aleatória dos bits, com auxílio de uma chave que é um valor escalar inteiro compreendido entre 0 a 2^{32-1} e que vai determinar a permuta específica. O desembaralhamento dos dados é assegurado de forma idêntica, sendo este processo assegurado pela função *randdeintrlv().*

- A classe LDPC da *toolbox* de comunicações que possui funções para codificação e descodificação de códigos LDPC sistemáticos.

5.3 RESULTADOS EXPERIMENTAIS

De seguida apresentamos os resultados de simulação para as duas abordagens descritas no capítulo 4. Importa referir que o tamanho das palavras testadas para cada um dos códigos referidos foi escolhido deforma a garantir, após descodificação, um número de palavras erradas superior a 100. Assim, para se obter um BER na ordem dos 10^{-4}, testou-se palavras com uma ordem de grandeza a rondar os 10^{6}.

5.3.1 PRIMEIRA ABORDAGEM

Nesta abordagem, conforme descrito no capítulo 4 e apresentado na Fig. 4.2, a chave de *interleaving* K após ser codificada pelo código curto C2, é usada no embaralhamento da mensagem original *M*, e concatenada com a mesma. A palavra resultante [K inter(M)] é codificada por C1 (código longo) antes de ser enviado para o canal.

A Figura 5.3 mostra resultados correspondentes à taxa de erro de bit (BER) para diferentes valores de E_b / N_o, cujo cálculo por (5.2) no ajuste do SNR do canal, levou em conta a taxa de código útil R_u dada por (4.1). As diferentes curvas a azul mostram a degradação de BER na mensagem *M* com o aumento da potência de transmissão do *jammer* ($P_j = \alpha P_\alpha$) que é ativado apenas durante a partilha da chave de *interleaving*.

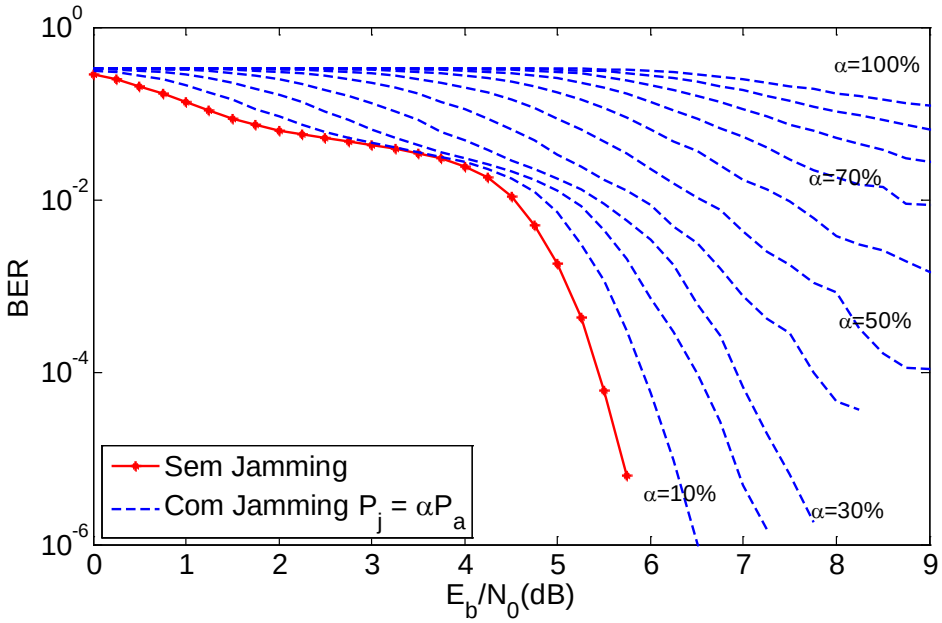


Figura 5.3 –BER da mensagem $\hat{M}$ função de E_b/N_0 para diferentes níveis de potência de jamming $(P_j = \alpha P_\alpha)$ aplicados à chave de *interleaving* K para o esquema de codificação 1.

Verifica-se que a utilização de *jamming* nesta abordagem, só é vantajosa quando este afeta ligeiramente o Bob (e.g., menos que 30% de P_a) e em contrapartida interfere de forma significativa na recepção da chave do Eve, i.e. com uma potência de transmissão acima 70% de P_a, proporcionando assim uma vantagem significativa para o recetor legítimo sobre o intruso em termos de taxas de erro. No entanto é de referir que apesar de ser necessário operar o *Jammer* a uma potência elevada, o curto intervalo de tempo durante o qual emite correspondente ao tempo de transmissão de K implica um consumo adicional de energia do sistema muito baixo (como será demonstrado mais à frente) e como tal a técnica apresentada é capaz de providenciar de forma eficiente uma transmissão robusta e uma significativa confidencialidade.

Para avaliar a importância do código LDPC curto C2 sobre a chave de *interleaving* nesta implementação, removemo-lo do sistema e os resultados correspondentes são apresentados na Figura 5.4.

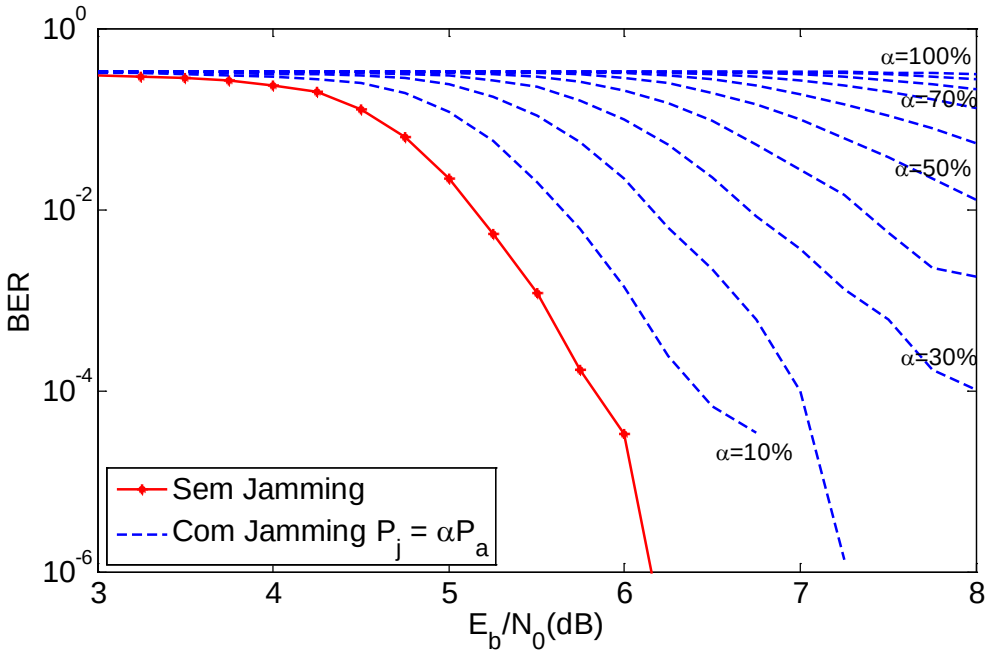


Figura 5.4 –BER da mensagem $\hat{M}$ função de E_b/N_0 para diferentes níveis de potência de *jamming* $(P_j = \alpha P_\alpha)$ aplicados à chave de *interleaving* K para o esquema 1, na ausência do código LDPC curto (C2) no esquema.

Constata-se que nestas circunstâncias as curvas de BER só começam afundar com um SNR com cerca de 4dB. Adicionalmente, isto conduz a que para os mesmos valores de BER, seja necessário um maior gasto de energia (E_b/N_0), desta forma ficando clara a utilidade do código LDPC curto para a chave de *interleaving*. Da análise conjunta das Fig. 5.3 e 5.4, conclui-se que a utilização do código C2 é especialmente importante no que respeita ao sinal recebido pelo Bob, na medida em sendo de esperar que o mesmo seja

afectado de forma reduzida por Jamming, eventuais erros que existam sobre a chave após a descodificação por C1, podem ser facilmente corrigidos pelo código curto C2.

IMPORTÂNCIA DO TAMANHO DA CHAVE DE *INTERLEAVING*

Com o intuito de analisar o efeito do tamanho da chave de *interleaving* K, utilizamos o primeiro esquema implementado e removemos o LDPC curto (C2) do sistema. Desta forma, o sistema funciona apenas com o código LDPC longo (C1) e variámos então o tamanho da chave de *interleaving*, que toma valores entre 200 e 300 bits. Para esta análise, consideramos que o Eve sofre interferência com potência de transmissão igual à da Alice ($\alpha = 100\%$), enquanto Bob está livre de interferência $(\alpha = 0)$. A Figura 5.5 mostra-nos o desempenho obtido nestas condições.

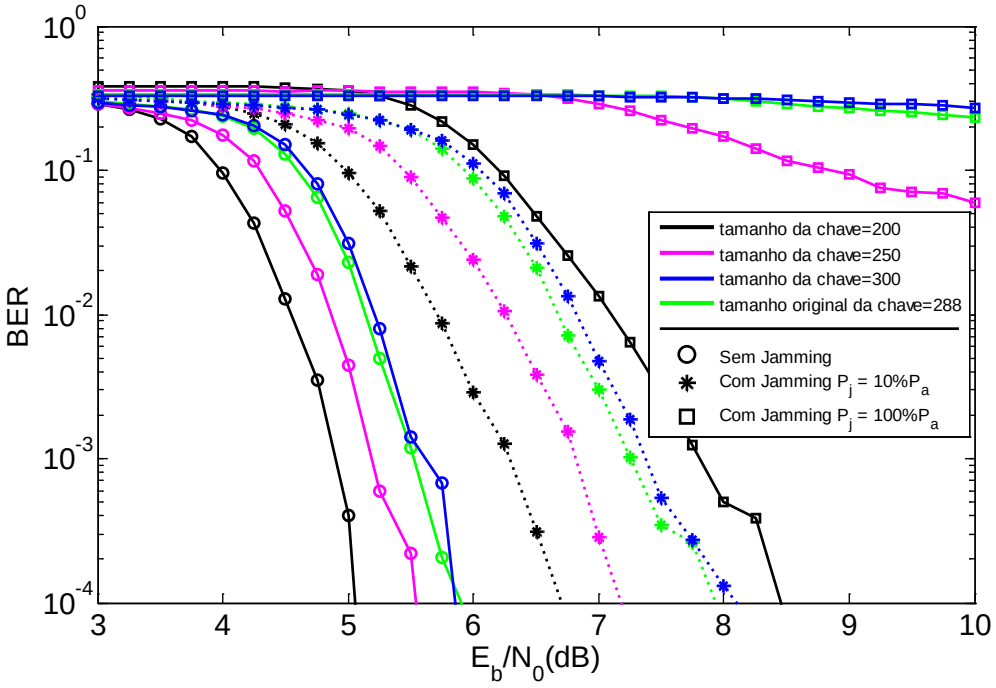


Figura 5.5 – BER da mensagem $\hat{M}$ função de E_b/N_0 e do tamanho da chave para $P_j = \alpha P_\alpha$ esquema 1 na ausência do código LDPC curto (C2).

Com o *jammer a* atuar com uma potência igual à potência de transmissão, o Eve apresenta três cenários diferentes: para os tamanhos da chave de *interleaving* de 288[1] e 300, o Eve degrada-se acentuadamente; com a chave de 250, este apresenta alguma degradação, mas menos acentuada que no caso anterior; por último, para a chave de 200 o Eve apresenta um comportamento em nada favorável ao nosso trabalho, visto que acompanha a curva de BER do Bob e a sua degradação não é muito evidente,

[1] Este comprimento foi testado por corresponder ao comprimento da chave usada na obtenção dos resultados das Fig.(s) 5.3 e 5.4.

conseguindo portanto aceder a grande parte da informação do Bob. No âmbito do nosso estudo, este comprimento deve ser completamente descartado por apresentar um mau desempenho a nível da confidencialidade desejada para o sistema.

No caso do Bob, conclui-se de imediato, que o comportamento do sistema para o tamanho das chaves em questão, apresenta poucas alterações. O decréscimo de desempenho em termos de BER com o aumento do tamanho da chave, resulta de uma penalização de potência devido à transmissão de menor número de bits de mensagem por palavra de código.

A conclusão mais óbvia é que a segurança na transmissão da mensagem $\hat{M}$ é garantida ao longo dos períodos de comunicação do canal degradado para o Eve. Mas um fator com muita preponderância, está relacionado com um tamanho da chave de *interleaving, q*ue deverá ser correctamente dimensionado. Desta forma, os desempenhos dos sistemas por nós implementados (como se pode observar através das Figuras 5.3 e 5.6) mostram que há uma vantagem durante um curto período utlizado para a partilha de chaves de *interleaving*, garantindo que a comunicação é fiável e confidencial, mesmo que o Eve tenha um canal com a mesma qualidade que o Bob ao longo da transmissão de dados.

5.3.2 SEGUNDA ABORDAGEM

Nas Fig.(s) 5.6 e 5.7 apresentam-se os resultados de BER para os dois estágios de descodificação do segundo esquema proposto no capítulo 4 e resumido pela Fig. 4.3. Nesta abordagem, a chave de *interleaving* K é usada no embaralhamento não só da mensagem M mas também dos bits de paridade que resultam da codificação pelo código longo C1, o que obriga a inverter o tradicional processo de descodificação conforme descrito anteriormente, tendo sido proposto uma descodificação em dois estágios.

Os resultados são apresentados de forma idêntica aos do primeiro esquema, ou seja, assumindo que é a transmissão da chave de *interleaving* K é sujeita a Jamming com a potência $P_j = \alpha P_\alpha$.

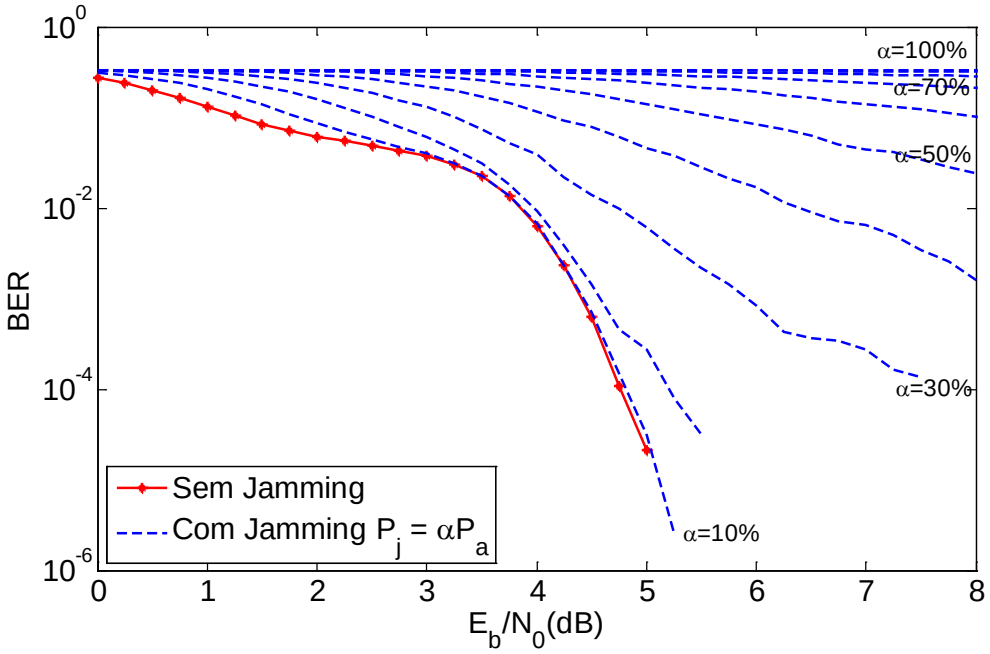


Figura 5.6 –BER da mensagem $\hat{\hat{M}}$ função de E_b/N_0 para diferentes níveis de potência de jamming $(P_j = \alpha P_\alpha)$ aplicados à chave de *interleaving* K para o esquema de codificação 2.

Observando a Figura 5.6 podemos concluir que o esquema 2 apresenta uma melhor performance do que aquela apresentada pela primeira abordagem. Há efetivamente uma diminuição da potência consumida para obtenção de baixas taxas de erros. Para um BER bit de 10^{-5}, este esquema apresenta face ao anterior uma vantagem em termos de E_b/N_0 de sensivelmente 1dB (o primeiro necessita de um E_b/N_0 de cerca de 6dB enquanto a segunda abordagem precisa apenas 5dB). Além disso, nesta abordagem se o Eve sofrer interferência de *jamming* com potência de transmissão acima 30% de P_a o BER degrada-se muito rapidamente, verificando-se que para valores acima de 50% de P_a, o BER apresenta valores muito próximos do máximo.

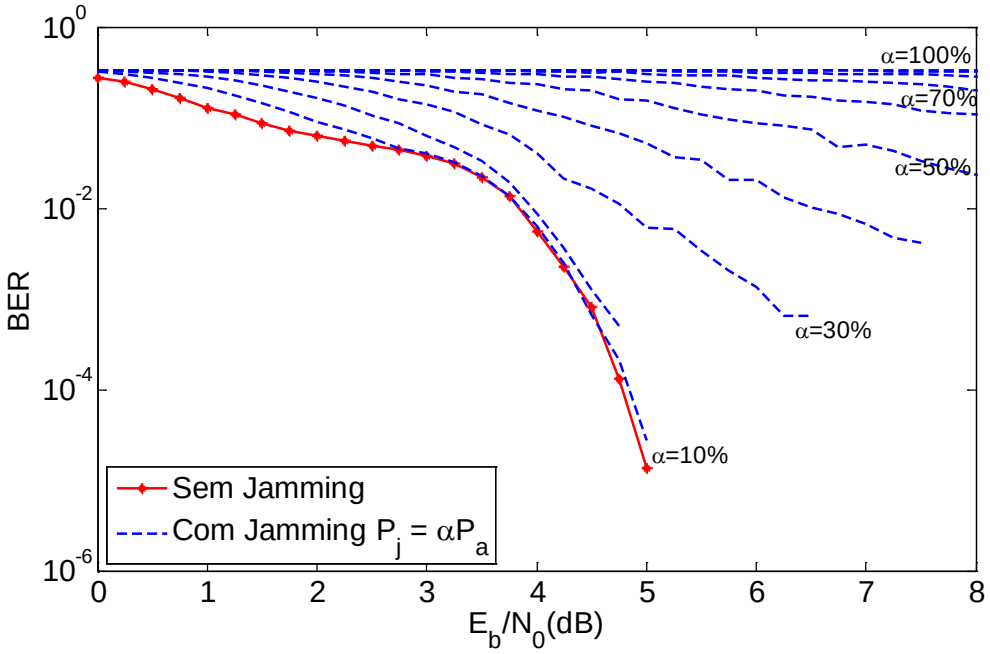


Figura 5.7 –BER da mensagem $\hat{M}$ em função de E_b/N_0 para diferentes níveis de potência de jamming $(P_j = \alpha P_\alpha)$ aplicados à chave de *interleaving* K para o esquema de codificação 2.

Esta abordagem apresenta a particularidade de ter dois estágios de descodificação: um primeiro estágio que conduz a uma aproximação grosseira e outro estágio que permite o refinamento dos resultados. No entanto quando comparadas as figuras 5.6 e 5.7 verifica-se neste caso não haver qualquer vantagem em proceder a um segundo estágio de descodificação, visto as melhorias observadas serem residuais. Tal foi atribuído ao facto de para a gama de valores de E_b/N_o, em que C1 fornece um bom desempenho o código C2 ser capaz de corrigir a maioria dos erros sobre a chave, evitando a necessidade de um segundo estágio de descodificação.

5.3.3 DESEMPENHOS DE AMBAS AS ABORDAGENS PARA CÓDIGOS LONGOS

Com vista analisar o desempenho de ambos esquemas aquando da utilização de um código C1 mais longo, garantindo desta forma um R_u mais elevado. O primeiro esquema foi testado comum código longo DVB-S2 com dimensões $(\eta_1, k_1) = (16200, 10800)$, mantendo o código LDPC curto (C2) utilizado nas circunstâncias anteriores. A taxa de código útil nestas condições melhora, passando a ter o valor

$$R_u = \frac{k_1 - \eta_k}{\eta_1} = \frac{10800 - 288}{16200} \approx 0.65. \qquad (5.3)$$

Através da Figura 5.8, podemos verificar que com a utilização de um código longo a primeira implementação apresenta um mau desempenho. As diferentes curvas com *jamming* seguem a curva sem *jamming*, *mesmo* com o aumento da potência de transmissão do *jammer.* Estes resultados indiciam que a primeira abordagem tem um desempenho precário para códigos longos, que permitiriam melhorar o code-rate obtido, i.e. com vista à obtenção de segurança a chave de *interleaving* K deve ocupar uma parte considerável da palavra [K inter(M)] codificada por C1.

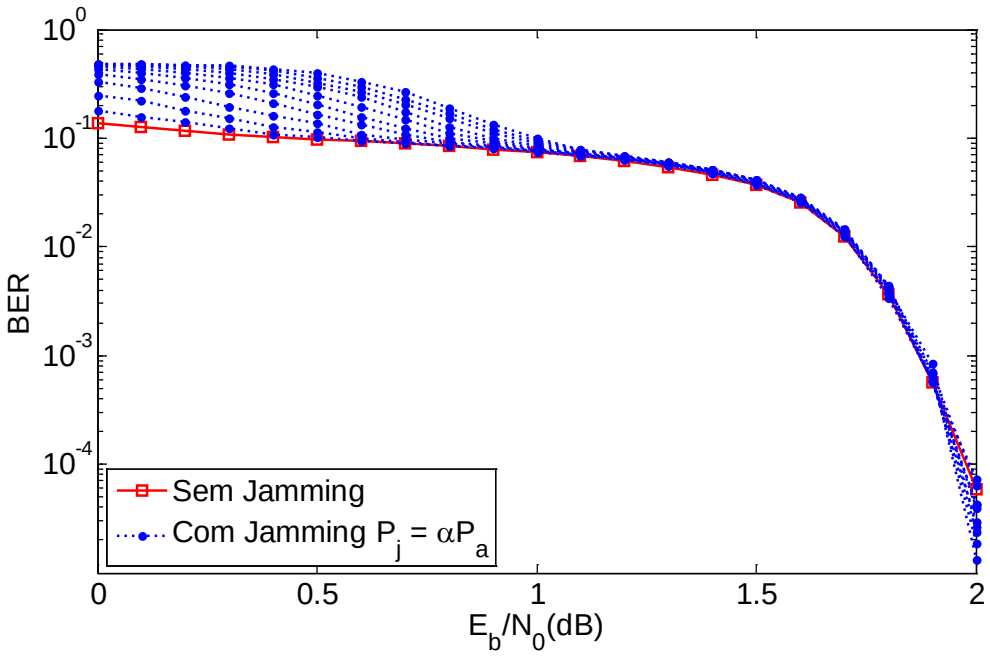


Figura 5.8 –BER da mensagem $\hat{M}$, de um código longo, em função de E_b/N_0 para diferentes níveis de potência de jamming ($P_j = \alpha P_\alpha$) aplicados à chave de *interleaving* K para o esquema de codificação 1.

No caso do segundo esquema foi usado um código de dimensões $(\eta_2, k_2) = (16200, 14400)$, logo com uma maior taxa de código útil.

$$R_u = \frac{k_1 - \eta_k}{\eta_1} = \frac{14400 - 288}{16200} \approx 0.871. \tag{5.4}$$

Nas Figuras 5.9 e 5.10 encontram-se representados o seu desempenho. Comecemos por realçar o papel preponderante do 2º estágio de descodificação na melhoria significativa do desempenho de BER, justificando a arquitetura proposta para o descodificador, quando o sistema é esperado operar para SNR mais baixos por utilização de um código C1 mais longo.

Comparando agora os resultados da Figura 5.9 com os da 5.8, observa-se que a segunda abordagem permite a obtenção de um melhor BER com um consumo de potência reduzido na ausência de *Jamming*. Para além disso, verifica-se que para potências de transmissão do *jammer* acima 30% de P_a conduzem a uma considerável

degradação do BER. Estes resultados mostram que a segunda abordagem possibilita a utilização de códigos mais longos, permitindo aumentar R_u do sistema codificador, sem prejuízo dos níveis de segurança desejados.

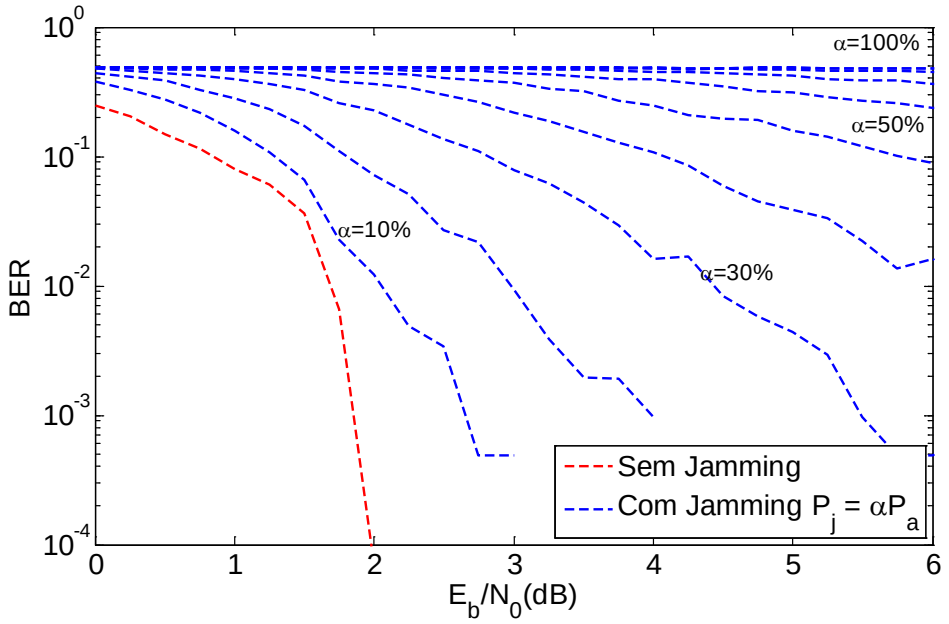


Figura 5.9 –BER da mensagem $\hat{\hat{M}}$, de um código longo, em função de E_b/N_0 para diferentes níveis de potência de jamming ($P_j = \alpha P_\alpha$) aplicados à chave de *interleaving* K para o esquema de codificação 2.

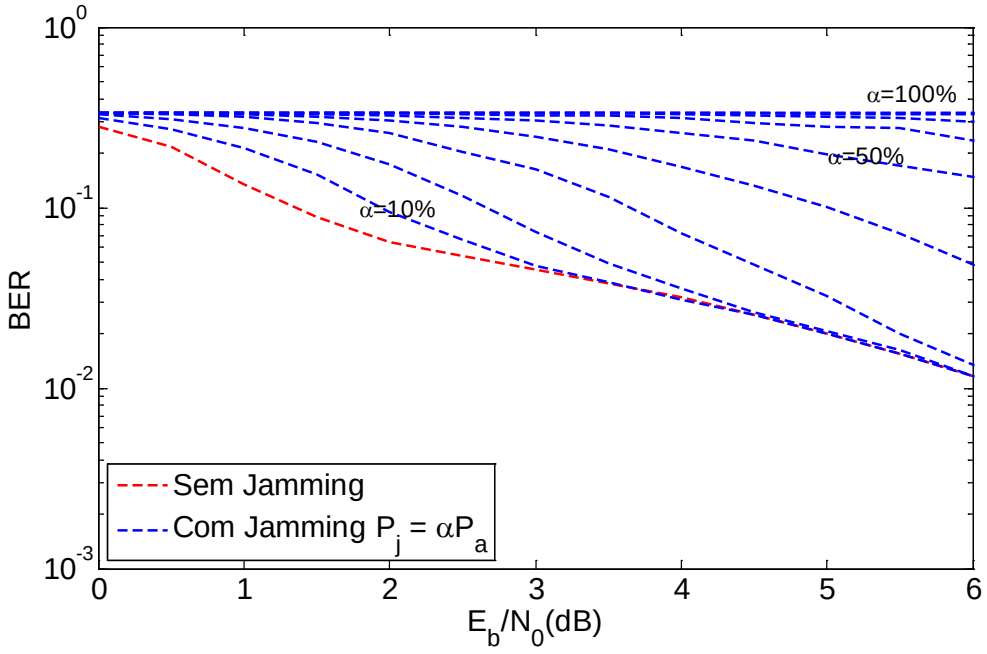


Figura 5.10 –BER da mensagem $\hat{M}$, de um código longo, em função de E_b/N_0 para diferentes níveis de potência de jamming ($P_j = \alpha P_\alpha$) aplicados à chave de *interleaving* K para o esquema de codificação 2.

5.3.4 CUSTO ENERGÉTICO DA UTILIZAÇÃO DE JAMMING

Para analisarmos a fração de energia gasta pelo *jammer* aquando da introdução de ruído, consideramos dois códigos diferentes (um curto e outro longo) e um tamanho fixo da chave de *interleaving* K. Eis então os dados utilizados:

- ✓ Código curto com dimensões $(\eta_1, k_1) = (1056, 880)$;
- ✓ Código longo (DVB-S2) com dimensões $(\eta_2, k_2) = (16200, 14400)$;
- ✓ Chave de *interleaving* K com tamanho $\eta_K = 288$.

Para tal, consideramos o custo de energia associado ao *jamming,* que pode ser medido como a razão entre E_{Jb}, definida como a energia de *jamming* gasta por transmissão de um bit de informação, normalizada à energia gasta pela Alice por cada bit de informação transmitido, i.e. E_b. Esta razão é dada por:

$$\frac{E_{Jb}}{E_b} = \frac{\alpha \eta_K}{\eta_i}, \tag{5.4}$$

onde se E_{Jb}/E_b considera a presença de um *jammer* que provoca interferências (ruído AWGN extra) com potência de transmissão $P_j = \alpha P_\alpha$ (uma fração α da potência de transmissão de Alice).

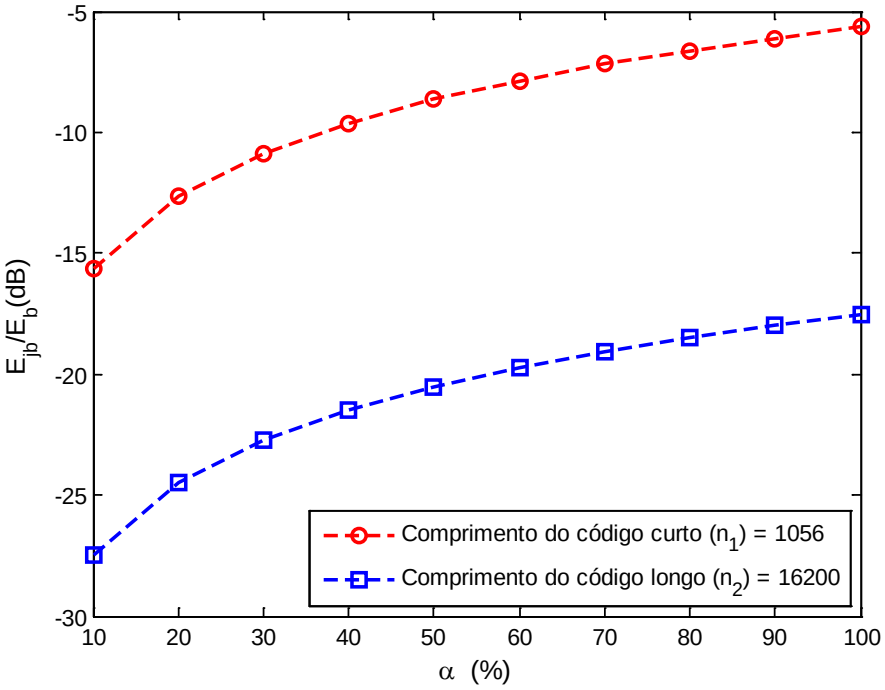


Figura 5.11 – Percentagens do custo de energia de *jamming* em função das frações da potência de transmissão de Alice

Através da análise dos resultados obtidos no gráfico 5.11, verifica-se que o *jammer* consome apenas uma pequena fração de potência comparado com a de transmissão gasta pela Alice. Por exemplo, para o caso do código curto, quando se aplica *jamming* com 10 % da potência de transmissão de Alice, o *jammer* vai apenas consumir, aproximadamente, -15 dB dessa mesma fração, uma vez que está ativo apenas durante a transmissão da chave de *interleaving*. Também se pode observar que o custo de energia de *jamming* reduz-se significativamente quando utilizado em códigos mais longos. Concluímos assim que este sistema garante uma redução do custo energético associado à geração de interferência.

A segunda abordagem destaca-se então pela sua versatilidade em lidar tanto com códigos curtos como longos, garantindo níveis de segurança desejados.

CAPÍTULO 6

CONCLUSÕES E TRABALHO FUTURO

6.1 CONCLUSÕES

Neste trabalho propusemos uma codificação para esquemas de segurança em que em que a mesma resulta da transmissão de uma chave pequena de *interleaving* durante um período de comunicação vantajoso sobre o Eve. Esta chave é usada para embaralhar aleatoriamente os dados originais antes de ser enviada juntamente com os dados através da rede protegido por um poderoso código de canal que garante a fiabilidade da transmissão, i.e. um baixo BER. Os esquemas propostos nesta dissertação fornecem confidencialidade, evitando que a descodificação pelo Eve seja bem-sucedida, i.e. experimentando BER elevados quando comparado com o Bob.

Este trabalho pretendeu romper com o paradigma dominante em que a segurança de transmissão de dados é assegurada apenas durante períodos de comunicação do canal degradado para o Eve. Na verdade os nossos resultados mostram que ter uma vantagem durante um pequeno período utilizado para a distribuição de chaves de *interleaving*, é suficiente para garantir uma comunicação confiável e confidencial, mesmo que o Eve experimente um canal com a mesma qualidade que o recetor legítimo durante a transmissão de dados.

De todos os testes exaustivos realizados, a segunda abordagem foi aquela que obteve resultados mais satisfatórios a todos os níveis. Evidenciou-se por conseguir uma comunicação segura e fiável, garantindo, na transmissão da mensagem *M,* que o Bob é capaz de recuperar a mensagem sem erros (fiabilidade), sem que o Eve seja capaz de adquirir qualquer informação. Tal facto deve-se à adição de ruído extra, através da aplicação de *jamming* à chave K, que degrada substancialmente este canal (confidencialidade).

Assim, nesta dissertação foram propostos dois esquemas similares baseados na combinação do *interleaving* com códigos poderosos de blocos sistemáticos (LDPC) para melhorar a segurança da rede sem fios com um custo reduzido de energia aquando da geração de interferência no sistema.

Dos dois esquemas desenvolvidos, a segunda abordagem apresentou uma performance superior a todos os níveis. Na comparação das curvas de BER, e considerando a utilização dos mesmos códigos LDPC em ambas as abordagens, para o segundo esquema, o Bob apresenta claramente um consumo de potência mais reduzido para obtenção de um BER mais baixo.

Quanto à confusão que se pretende gerar ao Eve, pode-se concluir que o segundo esquema apresenta-se novamente mais vantajoso na medida em que a potência de *jamming* necessária para degradar o canal do Eve é mais baixa. Por exemplo, no segundo esquema, com uma potência de *jamming* de 30% de P_a o Eve já apresenta alguma

degradação, verificando-se que para valores acima de 50% de P_a, o Bob já apresenta taxas de erro muito mais vantajosas. Já para o primeiro esquema, só a partir de valores de 40% de P_a é que o Eve começa apresentar alguma degradação, ainda que pouca e, por outro lado, só com uma potência de transmissão acima 70% de P_a é que o Eve começa a ter um comportamento esperado e atinge níveis de degradação mais elevados e, consequentemente, a perda quase total de informação sobre o Bob. Ainda assim, a maior simplicidade do primeiro esquema, nomeadamente a semelhança do descodificador a um descodificador de canal clássico (código interior $\rightarrow$ interleaving $\rightarrow$ código exterior) tornam-no interessante para transmissões com taxas de código útil R_u mais humildes.

6.2 TRABALHO FUTURO

A realização deste trabalho abre portas a novas linhas de investigação, tais como:

- Desenvolvimento de um esquema de *interleaving,* adaptado à estrutura dos códigos de canal, por forma a maximizar os ganhos de confidencialidade;

- A avaliação destes esquemas com outras gamas de códigos, nomeadamente códigos mais curtos BCH sobre a chave, que garantem correção perfeita de um dado número de erros, bem como com diferentes modelos de canal, por exemplo com desvanecimento (*fading*);

- Estudo da determinação acerca do tamanho ideal da chave de *interleaving* que garanta fiabilidade e confidencialidade, reduzindo o peso sobre a taxa de código útil.

BIBLIOGRAFIA

[Murthy 2004] C. S. R. Murphy and B.S. Manoj, Ad Hoc Wireless Net-works: Architectures and Protocols, Prentice Hall PTR, 2004.

[Sheng 2011] Yi-Sheng Shiu ; Nat. Tsing Hua Univ., Hsinchu, Taiwan ; Shih Yu Chang ; Hsiao-Chun Wu ; Huang, S.C.-H., "Physical layer security in wireless networks: a tutorial",IEEE Wireless Communications, vol. 18, pp.66-74, April 2011

[Wyner 75] A.D. Wyner, "The Wiretap Channel", Bell System Tech. J., vol. 54, 1975, pp. 1355-87.

[Ozarow 84] L. H. Ozarow, A. D. Wyner,"Wire-Tap Channel II", AT&T Bell Laboratories Technical Journal, Volume 63**,** Issue 10**,** pages 2135–2157**,** December 1984

[Thangaraj2005] Thangaraj,A. ,Dihidar, S. ; Calderbank, A.R. ; McLaughlin, S.W. ; Merolla, J.-M., "Applications of LDPC Codes to the Wiretap Channel", Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on, pp. 1498 – 1502, 4-9 Sept. 2005

[Thangaraj2007] Thangaraj,A. ,Dihidar, S. ; Calderbank, A.R. ; McLaughlin, S.W. ; Merolla, J.-M., "Applications of LDPC Codes to the Wiretap C hannel**",** Information Theory, IEEE Transactions on**,** Volume-53, pp. 2933-2945, Aug. 2007

[Liu 2007] Ruoheng Liu; Princeton Univ. Princeton, Princeton; Yingbin Liang; Poor, H.V.; Spasojevic, P., "Secure Nested Codes for Type II Wiretap Channels"

[BLOCH 2006] Bloch, M; GTL-CNRS Telecom, Metz; Thangaraj, A.; McLaughlin, S.W.; Merolla, J.M., "LDPC-based secret key agreement over the Gaussian wiretap channel", Information Theory, 2006 IEEE International Symposium on, pp. 1179-1183, 9-14 July 2006

[Muramatsu2006] Jun Muramatsu, “Secret Key Agreement from Correlated Source Outputs Using Low Density Parity Check Matrices”, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences , Vol.E89-A No.7 pp.2036-2046, 2006/07/01

[Bloch 2007] Matthieu Bloch,, João Barros , Miguel R. D. Rodrigues and Steven W. McLaughlin, “Information Theoretic Security for Wireless Channels - Theory and Practice” *in* Information Theory and Applications Workshop 2007, UCSD, pp.-, 2007

[Bloch 2011] Matthieu Bloch, Joao Barros, "Physical-Layer Security: From Information Theory to Security Engineering", Cambridge University Press 2011

[Shannon 49] C.E.Shannon, "Communication theory of secrecy systems", Bell System Technical Journal, vol. 28, no.4, pp.656-715, April 1949

[Klinc 2011] D. Klinc, J. Ha, S. W. McLaughlin, J. Barros, and B.-J. Kwak, “LDPC codes for the Gaussian wiretap channel,” IEEE Trans. Inform. Forensics Sec., vol. 6, no. 3, pp. 532–540, Sept. 2011.

[Willie 2013] Willie K Harrison, Joao Almeida, Matthieu R Bloch, Steven W McLaughlin, Joao Barros, Coding for Secrecy An Overview of Error Control Coding Techniques for Physical Layer Security, IEEE Signal Processing Magazine, Vol. 30, no. 5, pp. 41-50, September 2013

[Storn 97] R. Storn and K. Price, ”Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”, Journal of Global Optimization, Vol. 11, pp. 341-359, 1997.

[Tanner 81] R.M. Tanner, "A recursive approach to low complexity codes", IEEE Transactions on Information Theory, September, 1981

[Gallager 63] R. G. Gallager, "Low-Density Parity-Check Codes", MIT Press, Cambridge, MA, 1963

[Mackay 99] D. Mackay, " Good Error correcting codes based on very sparse matrices", IEEE Transactions on Information Theory,vol. 45, pp. 399-431, March, 1999

[Shannon 48] C.E. Shannon, "A Mathematical Theory of Communication", Bell Systems Technical Journal, vol.27, 1948

[Berrou 93] C. Berrou, A. Glavieux e P. Thitimajshimi, “Near Shannon Limit Error-Correcting Coding and Decoding”, Proceedings ICC’93, Genéve, Suíça, pp. 1064-70, Maio 1993.

[Nick 2011] Nick Wells, "DVB-T2 in relation to the DVB-x2 Family of Standards",Nick Wells (Chairman of DVB TM-T2 working group), BBC R&D, London, England, UK.ATSC Symposium on Next Generation Broadcast Television, February 15, 2011.

[Brack 2006] Brack, T.;Alles, M. ; Kienle, F. ; Wehn, N. , "A Synthesizable IP Core for WIMAX 802.16E LDPC Code Decoding", Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on, pp. 1-5, 11-14 Sept. 2006

[Shu 83] Shu Lin, Daniel J. Costello;" Error Control Coding: Fundamentals and Applications", Prentice-Hall, 1983

[Jin 2000] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," 2nd International Symposium on Turbo Codes & Related Topics, Brest, France, 2000.

[Richardson2003] T. Richardsom, “The Renaissance of Gallager’s Low-Density Parity-Check Codes”, IEEE Communications Magazine, pp. 126-130, Agosto de 2003.

[Gallager 62] R. G. Gallager, “Low-Density Parity-Check Codes”, IRE Transactions Information Theory, vol. IT-8, pp. 21-28, Janeiro 1962.

[Chung 2001] Sae-Young Chung, G David Forney Jr, Thomas J Richardson, Rüdiger Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit", IEEE Communications Letters, vol. 5, no.2, pp. 58-60, Fevereiro 2001.

[Frey 2001] F. R. Kschischang, B. J. Frey e H. Loeliger, “Factor Graphs and the Sum-Product Algorithm”, IEEE Transactions on Information Theory, vol. 47, nº 2, pp. 498-519, Fevereiro 2001.

[Ksch. 2003] F. R. Kschischang, “Codes Defined on Graphs”, IEEE Communications Magazine, pp. 118-125, Agosto 2003.

[Leung 2001] W. K. Leung, W. L. Lee, A. Wu e L.Ping, “Efficient implementation technique of LDPC decoder”, Electronic Letters, vol. 37, nº 20, pp. 1231-1232, 27 de Setembro 2001.

[Eleftheriou 2001] E. Eleftheriou, T. Mittelholzere A. Dholakia, “Reduced-Complexity Decoding Algorithm for Low-Density Parity-Check Codes”, E lectronics Letters, vol. 37, nº 2, pp. 102-104, Janeiro 2001.

[Hu 2002] X.-Y. Hu e T. Mittelholzer, “An Ordered-Statistics-Based Approximation of the Sum-Product Algorithm”, ITS 2002, InternacionalTelecommunications Symposium, pp. 205-210, Setembro 2002.

[Hanna 93] Hanna, S.A,"Convolutional interleaving for digital radio communications", Universal Personal Communications, vol. 1, pp. 443-447, Oct 1993

[Csiszar 78] Csiszar, I., Korner, J., "Broadcast channels with confidential messages",IEEE Transactions on Information Theory, vol.24, pp. 339-348, May 1978

[Vilela 2011] J.P. Vilela, P.C. Pinto, J. Barros, "Jammer Selection Policies for Secure Wireless Networks",IEEE ICC 2011 Workshop on Physical-Layer Security, Kyoto, Japan, June 2011.

[Vilela 2012] J.P. Vilela, J. Barros, "A Cooperative Protocol for Jamming Eavesdroppers in Wireless Networks",IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 2012.

[Xu 2005] Wenyuan Xu , Yanyong Zhang , Timothy Wood, "The feasibility of launching and detecting jamming attacks in wireless networks", ACM international symposium on Mobile ad hoc networking and computing (USA), pp. 46-57, 2005

[Vilela2 2011] J. P. Vilela, P. C. Pinto, and J. Barros, “Position-based jamming for enhanced wireless secrecy,” IEEE Transactions on Information Forensicsand Security, vol. 6, no. 3, pp. 616–627, September 2011.

[Carlson 2002] A. Bruce Carlson, P.B. Crilly, and J.C. Rutledge, Communications Systems, 4th edition, New York; McGraw-Hill, 2002

[Wick 95] Stephen B. Wicker, Error Control Systems for Digital Communication and Storage, Prentice Hall, 1995.

[Bos 99] Martin Bossert, Channel Coding for Telecommunications, John Wiley & Sons, 1999.