

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE
CAMPINAS**

**CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E
DE TECNOLOGIA**

RICARDO FIALHO TAFAS JUNIOR

**CRIPTOGRAFIA POR OFUSCAMENTO DE SINAIS
EM CAMADA FÍSICA**

CAMPINAS

2016

RICARDO FIALHO TAFAS JUNIOR

**CRIPTOGRAFIA POR OFUSCAMENTO DE SINAIS EM
CAMADA FÍSICA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito parcial para obtenção do título de Mestre em Gestão de Redes de Telecomunicações.

Área de Concentração: Engenharia Elétrica.
Orientador: Prof. Dr. Eric Alberto de Mello Fagotto

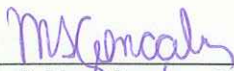
Dissertação defendida e aprovada em 16 de dezembro de 2016 pela Comissão Examinadora constituída dos seguintes professores:



Prof. Dr. Eric Alberto de Mello Fagotto
Orientador da Dissertação e Presidente da Comissão Examinadora
Pontifícia Universidade Católica de Campinas



Prof. Dr. Amilton da Costa Lamas
Pontifícia Universidade Católica de Campinas



Prof. Dr. Marcos Sérgio Gonçalves
Universidade Estadual de Campinas

Ficha Catalográfica
Elaborada pelo Sistema de Bibliotecas e
Informação - SBI - PUC-Campinas

t005.82
T124c

Tafas Junior, Ricardo Fialho.

Criptografia por ofuscamento de sinais em camada física / Ricardo Fialho Tafas Junior. - Campinas: PUC-Campinas, 2016.
67p.

Orientador: Eric Alberto de Mello Fagotto.

Dissertação (mestrado) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Gestão de Redes de Telecomunicações.

Inclui bibliografia.

1. Criptografia de dados (Computação). 2. Programação (Computadores) - Gerência . 3. Sistema de telecomunicação. 4. Tecnologia da informação - Sistemas de segurança. I. Fagotto, Eric Alberto de Mello. II. Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias. Pós-Graduação em Gestão de Redes de Telecomunicações. III. Título.

22.ed. CDD – t005.82

À minha família e àqueles que ainda vão nascer para se juntarem a ela e também àqueles que nos deixaram com muita saudade.

AGRADECIMENTOS

À Simone Stivanin,

Um agradecimento especial à minha esposa e companheira, por abdicar de mim em vários momentos para que eu pudesse completar este curso.

Aos meus pais,

Que me deram toda a base e pelo esforço para que eu tivesse o fundamental em estudo e carinho. Espero, de certa forma, recompensar o esforço deles.

Ao professor Eric,

Que antes do meu ingresso no curso, me incentivou a percorrer esta caminhada. Por sorte, talvez, também agradeço ao esforço que ele fez em me ter como aluno, quando veio a se tornar meu orientador, substituindo o professor Abbade.

Aos meus colegas de curso,

Que tanto aguentaram minhas opiniões contundentes e controversas e pelo coleguismo para enfrentar esta jornada.

À PUC-Campinas,

Pela bolsa de estudos parcial, incentivo necessário para a decisão de iniciar o curso.

“Pois é insuficiente ter o espírito bom, o mais importante é aplicá-lo bem.”

René Descartes
(1596 – 1650)

RESUMO

TAFAS JR, Ricardo Fialho. **Criptografia por Ofuscamento de Sinais em Camada Física**, 2016. Dissertação (Mestrado Profissional em Gestão de Redes de Telecomunicações) - Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Campinas, 2016.

Existe uma necessidade crescente em garantir a segurança da informação em sistemas de telecomunicações. Contudo, atualmente, a maioria das técnicas de criptografia protege apenas a mensagem, não impedindo a captura do sinal correspondente aos símbolos transmitidos, o que pode comprometer a segurança da informação. Em função disso, neste trabalho, propõe-se uma nova técnica criptográfica, que promove o ofuscamento dos sinais na camada física, dificultando ações de espionagem. Primeiramente, antes da transmissão, aplica-se uma operação de mudança de base sobre o sinal e, em seguida, uma operação de ofuscação sobre o sinal modificado. Na sequência, retorna-se à base inicial e o sinal ofuscado está pronto para transmissão. Na detecção, executa-se novamente a mudança de base, aplica-se a operação de desofuscação, seguida da operação para retornar à descrição do sinal à base original. Este trabalho estimou a taxa de erro de bit por uso de chave errada e o tempo máximo de quebra do código para avaliar a técnica. Além da aplicação em sistemas de telecomunicações, a técnica pode ser aplicada em barramentos de dados, para evitar roubo de *software* ou *firmware*, prevenindo clonagem de produtos

Palavras-Chave: Gestão de Redes e Serviços, criptografia, ofuscação, desofuscação, sistemas de telecomunicações.

ABSTRACT

TAFAS JR, Ricardo Fialho. **Criptografia por Ofuscamento de Sinais em Camada Física**, 2016. Dissertação (Mestrado Profissional em Gestão de Redes de Telecomunicações) - Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Campinas, 2016.

There is a growing need to ensure information security in telecommunications systems. However, currently, most cryptographic techniques protect only the message and do not prevent the capture of the transmitted symbols, which could compromise the security of the information. Therefore, a new cryptographic technique is proposed in this work which obfuscates the signal in the physical layer, making eavesdropping more difficult. First of all, prior to transmission, a change of basis operation is applied on the signal and, then, an obfuscation operation is performed on the base-changed signal. Afterwards, the obfuscated signal is returned to the initial base and it is ready for transmission. On the receiver side, a basis change is performed again and the de-obfuscation operation is applied, followed by an operation to return the signal to its original base. This work evaluated the bit error rate due to wrong key application and the maximum code breaking time to assess the technique. In addition, to the application in data communication and information security systems, the technique can be applied to data buses to prevent product cloning.

Keywords: Management of Networks and Services, cryptography, obfuscation, de-obfuscation, telecommunication systems.

LISTA DE GRÁFICOS

Figura 1: Ponto de falha de dentro de Equipamento.....	15
Figura 2: Barramento desprotegido entre Processador e Memória.....	15
Figura 3: Blocos operacionais para codificar a mensagem.....	25
Figura 4: Blocos operacionais para decodificar a mensagem.....	26
Figura 5: Mensagem original M (a), vetor transformado M' (b), vetor transformado e ofuscado M'' (c) e sinal transmitido T (d).	35
Figura 6: Sinal T recebido (a), sinal transformado e ofuscado T' (b), sinal transformado T'' (c) e mensagem recuperada T''' (d).	36
Figura 7: Vetor T _e '' obtido a partir de chave errada (a) e correto T'' (b).	39
Figura 8: Mensagem recuperada com chave errada T _e ''' (a) e mensagem recuperada correta T''' (b).....	40
Figura 9: Resultado da aplicação da técnica para 8 bits. Sinal original M (a) e o sinal transmitido T (b).....	41
Figura 10: Resultado da aplicação da técnica para 16 bits. Sinal original M (a) e o sinal transmitido T (b).....	42
Figura 11: Resultado da aplicação da técnica para 32 e 64 bits. Sinal original M (a e c) e o sinal transmitido T (b e d).....	43
Figura 12: Resultados com uso de chave errada para K _a (a), K _b (b), K _c (c) e K _d (d).	46
Figura 13: Gráfico acumulado do Vetor T	50

LISTA DE TABELAS

Tabela 1. Resultados Funcionais da Técnica.	34
Tabela 2. Resultados não-funcionais da técnica.....	38
Tabela 3. Comparação entre resultados funcionais e não funcionais.	38
Tabela 4. Resultados Variados de Chaves Erradas.....	45
Tabela 5. Resultados de ataque de digitalização por força bruta.....	47

LISTA DE QUADROS

Quadro 1. Declaração das variáveis.	64
Quadro 2. Declaração inicial.	65
Quadro 3. Procedimento de Ofuscação e Desofuscação.	66
Quadro 4. Procedimento de tentativa de desofuscação com chave errada	66
Quadro 5. Armazenamento dos dados.	67

LISTA DE ABREVIATURAS E SIGLAS

ASIC	=	Application Specific Integrated Circuit
ASSP	=	Application Specific Standard Product
BER	=	Bit Error Rate
CMOS	=	Complementary Metal-Oxide Semiconductor
CODEC	=	Coder Decoder
DCT	=	Discrete Cosine Transform
DES	=	Data Encryption Standard
DFT	=	Discrete Fourier Transform
FFT	=	Fast Fourier Transform
FPGA	=	Field Programmable Gate Array
HDL	=	Hardware Description Language
IDCT	=	Inverse Discrete Cosine Transform
IOT	=	Internet of Things
JPEG	=	Joint Photographic Experts Group
LVCMOS	=	Low Voltage Complementary Metal-Oxide Semiconductor
M2M	=	Machine to Machine
MPEG	=	Moving Picture Experts Group
NRZ	=	Non-Return to Zero
OSI	=	Open Systems Interconnection
PAM	=	Pulse Amplitude Modulation
PBX	=	Private Branch Exchange
RSA	=	Rivest-Shamir-Adleman encryption

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	Contextualização.....	14
1.2	Problemática.....	16
1.3	Objetivos.....	19
2	FUNDAMENTAÇÃO.....	20
2.1	A Transformada Discreta de Cosseno (DCT).....	20
3	PROPOSTA DE SOLUÇÃO.....	23
3.1	Metodologia.....	23
3.2	Sistemas de Ofuscação e Desofuscação.....	25
3.3	Criptografia por Ofuscamento.....	26
3.4	Seleção de Operação de Transformação.....	28
3.5	Aplicação da Transformada Discreta de Cosseno para Ofuscação.....	29
3.6	Operador de Ofuscação e Desofuscação.....	30
4	RESULTADOS.....	32
4.1	Definições de Parâmetros.....	32
4.2	Ferramentas Utilizadas.....	32
4.3	Testes Funcionais.....	33
4.4	Testes Não-Funcionais.....	37
4.5	Testes sobre o Tamanho do Vetor de Dados.....	40
4.6	Testes de Robustez à Espionagem.....	43
4.7	Diagrama de Olho.....	48
4.8	BER.....	51
4.9	Viabilidade de Implementação em Hardware.....	52
4.10	Observações Gerais.....	54
5	CONCLUSÃO.....	56
5.1	Trabalhos Futuros.....	57
6	REFERÊNCIAS.....	59
7	APÊNDICES.....	64
	APÊNDICE A: IMPLEMENTAÇÃO EM SCILAB.....	64

1 Introdução

1.1 Contextualização

Existe uma tendência irreversível de aumento no volume da comunicação de dados, tanto por usuários quanto por elementos de rede independentes (nós de *internet* das coisas) ou comunicação M2M, *machine to machine*.

No que tange aos elementos independentes, ainda existe muita controvérsia sobre a quantidade estimada de dispositivos conectados, mesmo assim, há consenso que estes números giram na casa dos bilhões (Evans, 2011) (Ericsson, 2016). Ao mesmo tempo, os sistemas embarcados estão cada vez mais se tornando alvo de tais ataques. (Cysco Systems, 2015).

Ao aumentar a quantidade de dados circulantes, aumentam as chances de que um ataque, em especial os ataques de força bruta, encontre a chave correta para quebrar um código. Esta situação não preocupa um único usuário, mas operadores de serviços em geral, com milhares de usuários, passam a ter maior probabilidade de encontrar um problema. Um único usuário invadido pode trazer até problemas legais aos operadores de serviços disponíveis por *internet*, como por exemplo o caso de invasão de rede da Sony em 2011 (Baker, et al., 2011).

Além disso, é um fato comum os sistemas de comunicação se protegerem apenas em suas interfaces externas, sem atenção às interfaces internas. É igualmente comum encontrar sistemas embarcados, ditos de comunicação segura, que se comunicam de maneira aberta internamente, ou seja, os módulos, circuitos integrados e dispositivos de comunicação dentro do equipamento trocam dados abertos, situação que torna possível operações de quebra de proteção de segurança de armazenamento de imagem, por exemplo, ou apenas empregam criptografia digital convencional, o que também tornam possíveis os ataques citados por (Huang, 2002).

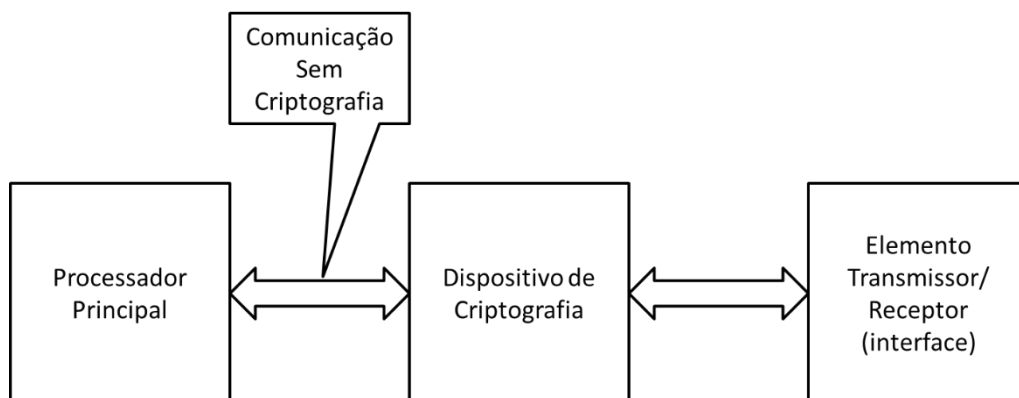


Figura 1: Ponto de falha de dentro de Equipamento

Na Figura 1 está exposta esta situação: neste caso, o elemento que realiza a criptografia de dados recebe, ainda dentro do equipamento, os dados sem nenhuma criptografia. Portanto, é possível que um invasor, com acesso físico ao equipamento em um centro de processamento de dados, por exemplo, abra a tampa do mesmo e possa conectar algum tipo de dispositivo espião entre os dispositivos do equipamento, para captação dos dados. Neste caso, então, todo o esforço de criptografia para o canal de comunicação teria sido em vão.

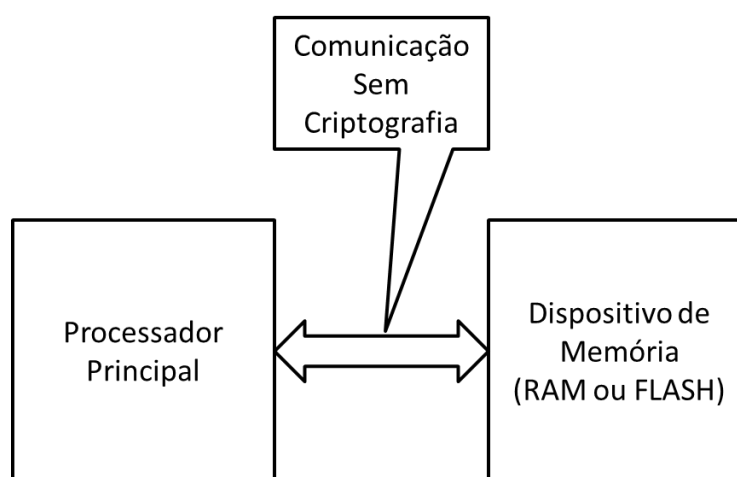


Figura 2: Barramento desprotegido entre Processador e Memória

Outra variação deste tipo de abordagem é exemplificada na Figura 2, para analisar os dados que trafegam entre um processador e a memória. Até mesmo senhas e chaves de criptografia são transferidas entre processador e memória sem nenhuma proteção. Contudo, uma das informações mais sensíveis

que podem ser espiadas são as imagens de sistema. Nelas estão todos os aplicativos, rotinas e algoritmos que o sistema embarcado possui.

Todas as eventuais falhas e defeitos de um sistema podem ser revelados se a imagem deste sistema for esmiuçada. Tais falhas ou defeitos, em maior severidade as falhas de segurança, seriam descobertas com maior dificuldade se fosse impossível obter a imagem do sistema para observação. Enfim, o acesso interno a um equipamento é tão ou até mais comprometedor que suas interfaces externas, em termos de segurança da informação. Há ainda um agravante: em praticamente todos os sistemas internos, todos os sinais estão em banda base, o que dispensa equipamentos sofisticados para espionagem.

Portanto, é preciso proteger os dados externos e internos de um equipamento. Ao considerar que qualquer equipamento em um sistema de comunicações que realize processamento dos dados é, antes de tudo, um sistema embarcado, a comunicação entre todos os dispositivos que constituem um equipamento de telecomunicações deve também ser protegida.

1.2 Problemática

Há uma ampla variedade de estudos sobre criptografia e sobre a utilização de técnicas para se criptografar a comunicação entre sistemas. O objetivo principal destas técnicas é impedir a interpretação de conteúdo por espiões, a despeito de não se evitar a detecção de transmissão de um sinal. Isso permite ao espião capturar o sinal, porém, sem conseguir decifrar o conteúdo da mensagem.

Quando se aplica criptografia apenas na camada 3 ou camadas superiores do modelo OSI (Tanenbaum, et al., 2010), permite-se aos espiões detectar se um sinal e, conseqüentemente, uma mensagem foram transmitidos (Shiu, et al., 2011). Desta mesma forma, existe a possibilidade de que, ao saberem qual foi o algoritmo de criptografia usado, os espiões sujeitem a mensagem a um ataque de exaustão ou força bruta, o mais comum dos ataques para decifrar uma mensagem criptografada, segundo (Quisquater, et al., 2005).

Nos sistemas em que há utilização de chaves criptográficas, o ataque de força bruta para quebra de criptografia acontece mediante o teste exaustivo de chaves. Alguns autores estudam técnicas de como acelerar tal procedimento (Barbieri, et al., 2014) e, dado que o aumento da capacidade computacional dos sistemas tende a aumentar e não há indícios contrários a esta tendência, impedir este tipo de ataque torna-se cada vez mais importante.

Um sinal é transmitido com uma sequência de símbolos (Tanenbaum, et al., 2010). Em função disso, é importante evitar que um agressor consiga ter acesso aos símbolos correspondentes à mensagem em uma transmissão de dados para inviabilizar o uso de técnicas de exaustão para quebra de código.

Segundo (Priberam), ofuscar é sinônimo de desaparecer ou tornar escuro. Pode-se entender que tornar escuro significa diminuir a visibilidade por insuficiência de luz. Ofuscamento de um sinal, portanto, significa fazer o sinal desaparecer ou tornar-se imperceptível aos possíveis espiões. Atualmente, as principais técnicas para ofuscar, já em camada física, os sinais transmitidos, são as de *spread spectrum* e *frequency hopping*, amplamente estudadas em (Torrieri, 2015). Estas, porém, tem objetivo de esconder o sinal e não, necessariamente, de criptografá-lo.

Recentemente, surgiram novos estudos com técnicas para ofuscar sinais ópticos em camada física, por divisão espectral (Abbade, et al., 2013). Estas técnicas consistem em modificar as componentes espectrais do sinal transmitido, o que, resumidamente, é análogo a mudar a base de um determinado vetor (no caso, o sinal óptico) e realizar alguma operação para ofuscação, neste caso, troca de posição de faixas espectrais. Desta forma, os bits da mensagem não serão lidos por invasores que não saibam como retornar as componentes ao formato original. Contudo, todas estas técnicas ópticas são baseadas em filtros para atingir o objetivo de ofuscação e não em com uso de algoritmos ou processamento, como é o caso deste trabalho.

Além dos estudos já citados, também se verificou que existem vários trabalhos em esteganografia com uso de imagens, como (Kester, et al., 2016), os quais consistem na aplicação de operações de transformada de discreta de

cosseco (*discrete cosine transform, DCT*) para embutir e esconder alguma mensagem em uma imagem. São técnicas que utilizam um ferramental parecido, porém, com o objetivo de inserir uma mensagem em uma imagem para transmissão segura em formato de arquivo. Por ser um método de criptografia utilizado em arquivos de imagem, considera-se que acontece também nas camadas mais altas do sistema de comunicação. Neste trabalho, se trabalha na criptografia de sinais e em camada física.

Existem também outros trabalhos que utilizam as operações de transformada em processos de criptografia, principalmente, por uso de redução ou *hashing* (Lyubashevsky, et al., 2008). De um ponto de vista matemático, esta operação objetiva usar a *FFT* (*fast Fourier transform*, transformada rápida de Fourier) como ferramenta de criptografia de vetores de dados (imagens, arquivos de texto, etc), que serão transmitidos na forma digital. Portanto, a técnica apresentada por (Lyubashevsky, et al., 2008) utiliza a *FFT* no lugar de outros algoritmos de redução conhecidos ou dos algoritmos de curvas elípticas e, portanto, apresenta alternativas a outros esquemas de criptografia como *DES* (*data encryption standard*) e *RSA* (*Rivest-Shamir-Adleman encryption*), dentre vários outros. Isto difere da abordagem deste trabalho, pois o objetivo deste é atingir a transmissão do vetor resultante das operações de transformação sem nenhum tipo de encapsulamento extra, ou seja, diretamente na camada 1. O objetivo é permitir que o sinal ofuscado seja inserido no canal de transmissão após um conversor analógico digital. Portanto, pode-se resumir que, neste trabalho, as operações de transformada são usadas de acordo com sua definição e função de mudança de domínio, no caso do domínio tempo para o domínio frequência.

Por ser importante esconder, mediante ofuscação, os símbolos de um sinal, bem como impedir, mesmo quando esta detecção acontecer, ao espião interpretar o conteúdo mensagem, há necessidade de criar uma ferramenta que, simultaneamente, realize estas duas atividades. Para tal, foi concebida uma técnica para ofuscar as mensagens transmitidas logo em camada física mediante ao uso de operações de transformada.

A técnica é nova, portanto, deve-se definir quais os operadores serão utilizados para poder formalizar o algoritmo que a emprega. Com a definição dos operadores, evidencia-se que é matematicamente possível ofuscar um sinal. Também é necessário realizar testes funcionais, não-funcionais, e testes de robustez para avaliar se a técnica criada tem capacidade de ofuscar o sinal e se o mesmo tem potencial capacidade de criptografia. Cabe salientar que o uso de banda base tem uma implicação conveniente nestes testes ao dispensar a utilização de todo o ferramental necessário para o emprego de uma eventual modulação.

1.3 Objetivos

São objetivos deste trabalho:

- a) Propor e testar em simulação uma técnica de criptografia de camada física para impedir a detecção dos símbolos de uma mensagem. Este tipo de criptografia recebe o nome de ofuscação de sinal.
- b) Desenvolver uma técnica que requeira pequena capacidade processamento, o que a habilitará para a aplicação em dispositivos de *IoT*, a internet das coisas (*internet of things*) ou como periférico de dispositivos eletrônicos que compõe os sistemas embarcados.

2 Fundamentação

2.1 A Transformada Discreta de Cosseno (DCT)

As aplicações mais conhecidas da *DCT* são nas normas ou *codecs* para os formatos de imagem *JPEG* (International Telecommunication Union, 1992), áudio *MP3* e vídeo *MPEG* (International Telecommunication Union, 1995). Nestes *codecs*, a *DCT* é uma operação muito utilizada como elemento principal para compactação do arquivo contendo estas mídias.

De um modo geral, pode-se explicar a *DCT* como a versão discreta de sua contraparte contínua, no caso, a transformada cosseno, que, por sua vez, basicamente, é o resultado da parte real da transformada de Fourier. As características da *DCT* e suas propriedades estão apresentadas com detalhes em (Smith, 1999), sendo a reversibilidade, a ausência de ortogonalidade e o equacionamento simples as principais para este trabalho, pois atendem os requisitos da seção 3.4.

O resultado da aplicação da *DCT* sobre um vetor qualquer é um novo vetor finito composto por amplitudes de cosseno distribuídos em função do valor de frequência. Conforme (Smith, 1999), existem 4 tipos de *DCT*.

A *DCT* tipo *I* é a mais parecida com a *DFT* (*discrete Fourier transform*). Além disso, nota-se que equação da *DCT* tipo *I* é a mais complexa dentre os tipos de *DCT* e, embora seja o primeiro modelo estudado, não possui ampla utilização. De (Rao, et al., 1990):

$$X(j) = \frac{1}{2}(x_0 + (-1)^j \cdot x_{n-1}) + \sum_{i=1}^{n-2} (x(i) \cdot \cos\left[\frac{\pi}{n-1} \cdot i \cdot j\right]) \quad (1)$$

A *IDCT* (transformada discreta de cosseno inversa, *inverse discrete cosine transform*) tipo *I* é definida como:

$$x(i) = \frac{2}{n-1} \cdot \left(\frac{1}{2}(X_0 + (-1)^i \cdot X_{n-1}) + \sum_{j=1}^{n-2} X(j) \cdot \cos\left[\frac{\pi}{n-1} \cdot i \cdot j\right] \right) \quad (2)$$

A *DCT* tipo *II* é a mais comum e mais utilizada, principalmente, no processamento de imagens. Seu equacionamento é simples, porém, não é uma operação reversível. De (Rao, et al., 1990):

$$X(j) = \sum_{i=0}^{n-1} \left(x(i) \cdot \cos \left[\frac{\pi}{n} \cdot j \cdot \left(i - \frac{1}{2} \right) \right] \right) \quad (3)$$

A *DCT* tipo *III* e a *DCT* tipo *II* estão intimamente ligadas, pois uma é a inversa da outra, ou seja, a *IDCT* tipo *II* é igual a *DCT* tipo *III*. Esta é amplamente utilizada em compressão de imagens.

$$X(j) = \frac{1}{2} \cdot x_0 \cdot \sum_{i=1}^{n-1} \left(x(i) \cdot \cos \left[\frac{\pi}{n} \cdot i \cdot \left(j + \frac{1}{2} \right) \right] \right) \quad (4)$$

Para que a *DCT* tipo *III* seja a transformada inversa da *DCT* tipo *II*, basta multiplicá-la por $\frac{2}{n}$. O mesmo vale para transformar a *DCT-II* na transformada inversa da *DCT* tipo *III*.

Nas equações de (1) a (4), X é o resultado da transformada e corresponde a uma componente de cosseno. O vetor x é o vetor de entrada e é, normalmente, algum sinal amostrado em domínio tempo. Os índices i e j correspondem ao índice de posição do elemento nos vetores x e X , respectivamente. Por fim, n é o tamanho dos vetores trabalhados.

A *DCT* tipo *IV* é uma operação de *DCT* que possui características interessantes para este trabalho, isto porque, exceto por um fator de normalização, possui a mesma de sua transformada inversa. Utiliza-se a definição de normalização de (Rao, et al., 1990) para cálculos da *IDCT*. Para a fórmula base, será utilizada a definição de (Scilab Enterprises, 2015). Por ter sido este o tipo escolhido, ele está apresentado em detalhes e é aplicado na técnica proposta na seção 3.5.

Há uma consideração a ser feita sobre os fatores de normalização. É o fator de normalização que gera diferenças nas fórmulas de *DCT* e *IDCT*. De acordo com (Rao, et al., 1990), costuma-se multiplicar tanto a *DCT*, quanto a *IDCT*, por um fator de $\sqrt{\frac{2}{n}}$ para gerar isomorfismo entre estas operações. Ao avaliar o fator de normalização, se observa que este é apenas um fator de ajuste

de amplitude numérica e não muda a característica do vetor resultante em número de componentes e amplitude relativa entre as mesmas.

Os demais tipos de *DCT* (tipo *V* em diante) são variações dos modelos anteriores para lidar com assimetrias, condições de tamanho de tabela e etc. Estes modelos de *DCT* operam apenas sobre matrizes e conforme definido na seção 3.3, este estudo será realizado apenas sobre vetores.

3 Proposta de Solução

3.1 Metodologia

Para desenvolver uma técnica nova, primeiro é necessário desenvolver um ferramental teórico para suportar a afirmação de que é possível usar operações de transformada para o ofuscamento de um sinal. Para tal, será realizado um estudo de como um sistema genérico de criptografia por ofuscamento pode ser construído, com a definição de um diagrama de blocos genérico.

Serão avaliadas algumas operações de transformada para que, assim que uma operação seja definida, tal escolha não tenha sido realizada ao acaso. Definem-se algumas características para as operações de transformada que poderão ser verificadas ao avaliar a literatura corrente.

Para a função de ofuscação, deve-se optar por uma função que dificulte aos espões conseguir recuperar a mensagem original com facilidade. A função de ofuscação é, potencialmente, o elemento de maior impacto no tempo necessário para a quebra do segredo e, portanto, o tempo máximo para quebra de chave será calculado.

Diagramas de blocos com a solução devem ser usados e serão apresentados. Tais diagramas em conjunto com a definição das operações irão auxiliar na implementação de um algoritmo que empregue a técnica de ofuscamento de sinais. Este algoritmo será usado para execução de todos os testes, em ambiente de simulação e com uso de computador pessoal comum.

Usualmente, novas técnicas são testadas em ambiente de simulação antes da construção de qualquer tipo de *hardware* para avaliar, experimentalmente, se a técnica funciona. Isso serve também para se evitar desperdício de esforço caso a técnica, já em ambiente de simulação, não funcione.

Para as simulações, cumpriram-se as seguintes etapas, que foram:

- Definição dos parâmetros
- Testes funcionais
- Testes não funcionais
- Testes de robustez a ataques.

Foram realizados 4 testes. O primeiro teste deve provar que a técnica pode ser aplicada para ofuscar um sinal. Neste teste, o resultado esperado é que o sinal ofuscado seja diferente do sinal original e que o sinal recuperado, ao final de todo o processo, seja igual a mensagem original.

O segundo teste, o teste não-funcional deve evidenciar que a técnica pode ser aplicada para proteger o sinal, portanto, será empregada uma chave errada. Neste caso, o sinal recuperado não pode ser igual ao sinal de entrada.

Para o primeiro e segundo testes, todas as comparações de sinal será numérica e será feita avaliando o resultado em tabelas. A avaliação visual é feita em gráficos de amplitude, medida em unidades arbitrárias, em função do tempo ou frequência, ambos medidos em unidades arbitrárias. Tal avaliação visual será apenas qualitativa.

O terceiro teste avalia a da aplicação da técnica com o aumento do tamanho de símbolos no sinal. Com este teste, o impacto deste aumento no tamanho da amostra no tempo máximo de quebra de chave é avaliado. Também é evidenciada a diferença entre sinal de entrada e sinal recuperado desofuscado, de forma qualitativa.

O quarto teste realizado deve avaliar a robustez a uma tentativa de ataque. Para este teste, definido um formato de detecção de dados digitais e será avaliado quantos bits de informação errada o espião iria obter com este detector. Este dado seria taxa de erros que o espião perceberia em sua tentativa de captura.

Para poder concluir que o algoritmo que implementa esta técnica pode ser implementado em hardware, será apresentada uma avaliação dos estudos existentes, uma vez que a prototipação em *hardware* da técnica está fora do escopo deste trabalho.

3.2 Sistemas de Ofuscação e Desofuscação

Para apresentar o funcionamento do esquema de criptografia por ofuscamento proposto, definiram-se os diagramas de blocos de transmissão, de acordo com a Figura 3, e recepção, de acordo com a Figura 4.

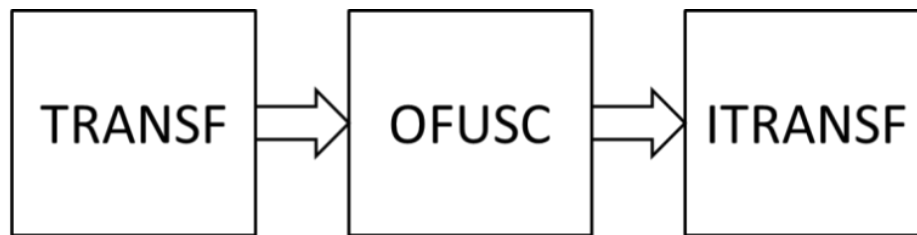


Figura 3: Blocos operacionais para codificar a mensagem.

A entrada do bloco de transformação deve ser uma mensagem genérica. A esta mensagem, será aplicada uma operação de mudança de base ou domínio, para que seja possível aplicar a etapa seguinte, que é a etapa que realizará a ofuscação do sinal transformado.

Entende-se por operação de ofuscação qualquer operação matemática ou procedimental que será aplicada sobre o vetor transformado para que o vetor a ser transmitido tenha idealmente nenhuma semelhança com o sinal original e que preferencialmente sequer se assemelhe com um sinal, mas sim, que pareça apenas ruído.

Para poder realizar a transmissão do sinal transformado, deve-se retornar o sinal à base original, com uma operação de transformação inversa. Embora o sinal possa ser transmitido apenas transformado e ofuscado, aplica-se esta etapa para dificultar ao espião detectar qual a função de ofuscação utilizada e também para aumentar a chance de que as características físicas do sinal ofuscado transmitido estejam próximas do sinal original.

Após definir a transmissão, define-se também a recepção. Como se pode observar na Figura 4, este sistema é muito semelhante ao sistema de transmissão.

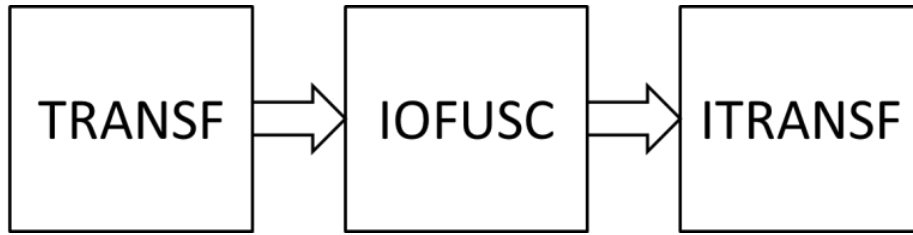


Figura 4: Blocos operacionais para decodificar a mensagem.

No diagrama de blocos para a recepção da Figura 4, observa-se que também se inicia mudando a base do sinal com a mesma operação de transformação. Aplica-se a operação de desofuscação, que deve consistir em algoritmos e funções que revertam o sinal ao seu estado transformado normal e, finalmente, aplica-se a transformada inversa.

A recepção, finalmente, deve produzir um sinal de saída igual ao sinal de entrada da transmissão.

3.3 Criptografia por Ofuscamento

Os sinais utilizados neste trabalho serão descritos mediante vetores na forma:

$$Z = [z_1, z_2, z_3, \dots, z_k, \dots, z_n] \quad (5)$$

Sendo z_k a k -ésima componente dos N elementos do vetor Z . Por exemplo, utilizando-se desta convenção, define-se uma mensagem digital M , composta por N bits, como:

$$M = [m_1, m_2, m_3, \dots, m_k, \dots, m_N] \quad (6)$$

A primeira etapa na técnica de ofuscamento envolve uma operação genérica de mudança de base, representada por $TRANSF$:

$$M' = TRANSF(M) \quad (7)$$

Sendo que M' é a mensagem transformada mediante mudança de base. A operação de ofuscamento $OFUSC$ é indicada em (8) e gerará M'' quando aplicada à M' .

$$M'' = OFUSC(M', K) \quad (8)$$

Sendo *OFUSC* o operador genérico, que pode ser uma função ou até mesmo uma sequência de operações que modifica as componentes de M' e resulta em M'' . O vetor M'' representa a mensagem transformada e ofuscada e K a chave criptográfica de tamanho igual ao da chave M . Aplica-se na sequência uma operação inversa à operação de transformação utilizada em (7) para obter novo sinal de transmissão, T , representada em (9) por *ITRANSF*.

$$T = ITRANSF(M'') \quad (9)$$

Para este trabalho se assume que não existirão erros na transmissão ou na recepção. Logo, se assume que a mensagem recebida é exatamente igual à mensagem transmitida, ou seja, o sinal recebido é também T . A partir dele, busca-se obter a mensagem original M . As operações serão, em sua maioria, as mesmas, conforme foi evidenciado na seção 3.1. Sobre este vetor T é aplicada a transformada:

$$T' = TRANSF(T) \quad (10)$$

O vetor T' representa a mensagem recebida após transformação. Ele deve ser submetido a uma operação inversa à operação de ofuscação, representada em (11) por *IOFUSC*:

$$T'' = IOFUSC(T', K) \quad (11)$$

Sendo T'' o sinal transformado e desofuscado e K o vetor de chave criptográfica. Finalmente, retorna-se a mensagem à descrição na base original, mediante a aplicação do operador *ITRANSF*.

$$T''' = ITRANSF(T'') \quad (12)$$

Este resultado é representado por T''' . Para o caso de uso correto de chave na recepção:

$$T''' = M \quad (13)$$

Já para o caso de chave errada, espera-se que T''' e M sejam diferentes.

É possível estimar o tempo máximo para acertar o segredo em testes de exaustão. Neste cenário, o tempo máximo para a quebra de chave será

proporcional ao número de combinações a serem testadas, algo que depende do suposto, por parte do espião, tamanho da chave K . Este tempo é proporcional ao tempo necessário para completar as operações descritas em (11), (12) e (13). Este tempo pode ser estimado de acordo com (14):

$$T_{quebra} = L * (T_{IOFUSC} + T_{TRANSF} + T_{ITRANSF}) \quad (14)$$

Sendo L o número de chaves possíveis, T_{IOFUSC} é o tempo de cálculo da função de ofuscação inversa, T_{TRANSF} e $T_{ITRANSF}$ são os tempos para cálculo da função de transformada e transformada inversa, respectivamente.

3.4 Seleção de Operação de Transformação

Dentre as possíveis operações de transformação de sinal, as mais utilizadas, dentro do âmbito de telecomunicações, são a *FFT* e a *DCT*. Com base na apresentação do problema na seção 1.2, pode-se definir uma lista simples de requisitos:

- Ter custo computacional e de implementação baixo, ou seja, algoritmo simples, sem recursão;
- Ser reversível e é ainda melhor se a operação de transformada for igual ou semelhante à de transformação inversa.
- Preferencialmente, apresentar resultados com todas as componentes em fase (para que seja possível transmissões em único fio ou trilha em placa de circuito impresso, por exemplo).

Para escolher uma operação de transformada que atenda ao primeiro requisito, utilizou-se o estudo comparativo realizado por (Shirado, et al., 2015) sobre operações de transformação de sinais e a implementação proposta por (Kester, et al., 2016). Embora estes estudos tenham objetivo de comparar a precisão de dados numéricos e o impacto na qualidade resultante de imagens, as informações apresentadas nos mesmos sugerem que a *DCT* atende ao requisito de ser uma operação com baixo custo de implementação, simples e sem recursão.

A *DCT* atende o segundo requisito para a aplicação na técnica, que é ser uma operação reversível. Embora não seja uma operação isomórfica, a diferença entre transformada e transformada inversa não é relevante. É possível, portanto, assumir que o segundo requisito foi preenchido.

O terceiro requisito também é atendido. Diferente da transformada de Fourier ou da transformada de Laplace, a *DCT* apresenta a característica conveniente do resultado não apresentar características ortogonais. Isto significa que o sinal transmitido pode ser mantido em banda base, sem necessidade de qualquer modulação, aumento no número de vias de transmissão ou encapsulamento para a transmissão das duas componentes.

Além da avaliação destes estudos, existe bibliografia abundante. Em (Rao, et al., 1990), a *DCT* é abordada com ênfase e bastante detalhes.

3.5 Aplicação da Transformada Discreta de Cosseno para Ofuscação

Para que a *DCT* seja corretamente empregada na técnica proposta de ofuscação, adapta-se o equacionamento do *Scilab* para se obter as equações finais dos sistemas de ofuscação e desofuscação, conforme o diagrama de blocos descrito na seção 3.1.

A *DCT-IV*, conforme (Scilab Enterprises, 2015):

$$X(j) = 2 * \sum_{i=1}^A \left(x(i) \cdot \cos \left[\frac{\pi}{n} \cdot \left(i - \frac{1}{2} \right) \cdot \left(j - \frac{1}{2} \right) \right] \right) \quad (15)$$

Na equação (15), A é o número de elementos do vetor de entrada, x é o vetor a ser transformado (vetor de entrada), X é o vetor transformado resultante, i representa o coeficiente do vetor x de entrada e j representa os coeficientes do vetor transformado X . (Scilab Enterprises, 2015) inclui um fator de normalização de 2, que deverá ser anulado na *IDCT*.

Como será necessário inverter a operação, a operação inversa da *DCT IV*, a *IDCT IV*, tem fórmula definida em (16) já multiplicada pelo fator de normalização. (Rao, et al., 1990) sugerem multiplicar por $1/n$. Multiplica-se este

fator por $1/2$ para anular o efeito da normalização da formula da *DCT IV* proposta por (Scilab Enterprises, 2015), e então se obtém a fórmula para a *IDCT IV*:

$$x(i) = \frac{1}{2 * A} \sum_{j=1}^A X(j) \cdot \cos \left[\frac{\pi}{A} \cdot \left(j - \frac{1}{2} \right) \cdot \left(i - \frac{1}{2} \right) \right] \quad (16)$$

Ao substituir *TRANSF* por (15) e *ITRANSF* por (16), os sinais M' , T , T' , T'' passam a ser definidos respectivamente por (17), (18), (19) e (20).

$$m'_b = \sum_{a=0}^{A-1} \left(m_a \cdot \cos \left[\frac{\pi}{A} \cdot \left(a - \frac{1}{2} \right) \cdot \left(b - \frac{1}{2} \right) \right] \right) \quad (17)$$

$$t_b = \frac{1}{2 * A} \sum_{a=0}^{A-1} \left(m''_a \cdot \cos \left[\frac{\pi}{A} * \left(a - \frac{1}{2} \right) \cdot \left(b - \frac{1}{2} \right) \right] \right) \quad (18)$$

$$t'_b = \sum_{a=0}^{A-1} \left(t_a \cdot \cos \left[\frac{\pi}{A} * \left(a - \frac{1}{2} \right) \cdot \left(b - \frac{1}{2} \right) \right] \right) \quad (19)$$

$$t'''_a = \frac{1}{2 * A} \sum_{a=0}^{A-1} \left(t''_a \cdot \cos \left[\frac{\pi}{A} * \left(a - \frac{1}{2} \right) \cdot \left(b - \frac{1}{2} \right) \right] \right) \quad (20)$$

Nas equações (17) a (20), a é o coeficiente de posição do vetor de entrada para os vetores a serem transformados pela *DCT* e b é, de maneira semelhante, o coeficiente dos vetores de entrada para serem transformados pela *IDCT*. Os elementos m_a , m'_b , m''_a e t_b são as componentes enumeradas dos vetores M , M' , M'' e T , respectivamente. Os elementos t_a , t'_b , t''_a e t'''_a são as componentes enumeradas dos vetores T , T' , T'' e T''' , respectivamente.

3.6 Operador de Ofuscação e Desofuscação

Várias técnicas podem ser utilizadas para ofuscar a mensagem e, desta forma, impedir que um invasor detecte os bits da mensagem. Para completar a definição da técnica de criptografia proposta por mudança de base, a inversão de componentes do vetor M' como a operação *OFUSC* e, de maneira análoga, a reversão dos mesmos para *IOFUSC*.

Este procedimento é equivalente a se executar a operação de multiplicação M' por um vetor contendo os valores a ser invertidos. De forma genérica:

$$OFUSC(X,K) = X.K \quad (21)$$

Sendo que:

$$X.K = [x_1.k_1, x_2.k_2, x_3.k_3, \dots], \quad (22)$$

Sendo X um vetor genérico cujas componentes deseja-se inverter, K é o vetor de chave criptográfica que, neste caso, deve possuir apenas valores de +1 e -1, para inverter sem modificar a amplitude. Substitui-se X em (22) por M' e utiliza-se esta nova equação para substituição de (8), de forma que:

$$M'' = M'.K \quad (23)$$

E a mesma operação deve ser aplicada na recuperação, aplicando em (11).

$$T'' = T'.K \quad (24)$$

4 Resultados

4.1 Definições de Parâmetros

Optou-se por trabalhar com 8 *bits*, pois além de facilitar o trabalho, é um tamanho usual em sistemas digitais por ser potência de 2. Neste caso:

$$N = 8 \tag{25}$$

O dado de entrada escolhido para realização dos testes iniciais é obtido pela alternância de 0 (lógico) e 1 (lógico), ou seja, uma representação de onda quadrada.

$$M = [1, 0, 1, 0, 1, 0, 1, 0] \tag{26}$$

Por ser um sinal que deva representar a maior dificuldade para ser ofuscado pela técnica por ser periódico. Outro aspecto importante: é possível estimar que este vetor, bastante comum e facilmente gerado, seja usado em eventuais ataques para tentar recuperar a chave.

As chaves de criptografia utilizadas, conforme mencionado na seção 3.6, também foram geradas ao acaso, usando as funções aleatórias do *Scilab*.

Isto não garante a aleatoriedade da chave, contudo, mesmo que esta operação possa não ser verdadeiramente aleatória, isto não interfere na qualidade dos testes realizados. O objetivo principal do trabalho, conforme a seção 1.2 e 1.3 é ofuscar o sinal, o que independe da qualidade da chave empregada. Em um cenário de criptografia real, devem-se empregar técnicas de criação de chaves realmente aleatórias e recomendações de segurança de chaves, como por exemplo (Barker, et al., 2012).

4.2 Ferramentas Utilizadas

Para avaliar se a técnica criptográfica proposta, simula-se os sistemas de ofuscação e desofuscação com o *software Scilab*.

O *Scilab* é um *software* de código fonte aberto, que se assemelha bastante com o *Matlab*, que é um *software* proprietário. O *Scilab* é executado em

sistemas operacionais Windows e Linux. Neste trabalho, se utilizou a versão 5.5.2 para Windows (Scilab Enterprises, 2016).

Um ponto importante é que o *Scilab* já possui em suas funções padrão as operações de *DCT* e *IDCT*, dos tipos I ao IV. (Scilab Enterprises, 2015).

Outras funções bastante utilizadas foram as funções *grand()*, *ones()* e *zeroes()*, para a geração de vetores aleatórios (ou, mais propriamente, pseudoaleatórios).

Como pode ser observado nas equações de (5) a (31), não há correlação entre a chave e a informação de entrada. Portanto, a correlação entre o conjunto de dados e o conjunto chaves, em qualquer arranjo, não interfere na qualidade da ofuscação do sinal.

Todos os códigos foram escritos com o próprio editor do *Scilab* e depurado dentro do próprio aplicativo.

4.3 Testes Funcionais

Os testes funcionais são os testes que têm como objetivo demonstrar a possibilidade de utilizar a técnica para ofuscar os dados de entrada e posteriormente recuperá-los.

O primeiro teste realizado teve objetivo de ofuscação do sinal. A fim de se realizar estes testes, foi utilizada a chave para ofuscamento definida de forma aleatória conforme (27).

$$K = [1, 1, -1, 1, -1, -1, 1, 1] \quad (27)$$

Na Tabela 1, observa-se os resultados de todas as etapas de cálculo do sistema de transmissão e recepção.

Como pode ser observado na Tabela 1, a mensagem foi corretamente recuperada, pois o conteúdo do vetor T''' é igual ao do vetor M . Também é possível observar que o vetor T possui valores diferentes do vetor M ou T''' , resultado de acordo com o esperado.

Tabela 1. Resultados Funcionais da Técnica.

N	M	M'	M''	T	T'	T''	T'''
1	1,00	5,60	5,60	0,36	5,60	5,60	1,00
2	0,00	-1,20	-1,20	0,40	-1,20	-1,20	0,00
3	1,00	1,63	-1,63	1,34	-1,63	1,63	1,00
4	0,00	-0,14	-0,14	0,00	-0,14	-0,14	0,00
5	1,00	1,43	-1,43	1,27	-1,43	1,43	1,00
6	0,00	0,49	-0,49	-0,01	-0,49	0,49	0,00
7	1,00	2,24	2,24	0,41	2,24	2,24	1,00
8	0,00	4,60	4,60	-0,36	4,60	4,60	0,00

É possível também notar que M' e M'' são valores intermediários, tal como T' e T'' , descritos anteriormente. Nota-se que a diferença entre M' e M'' se dá pela aplicação da função de ofuscação apenas dos vetores que foram invertidos pela multiplicação pela chave K e da mesma forma que T' e T'' .

É esperado também que estes valores intermediários sejam repetidos entre si, deixando muito claro a característica reversível da operação. Observa-se que:

$$M' = T'' \quad (28)$$

E, completando a análise numérica:

$$M'' = T' \quad (29)$$

A Figura 5a apresenta o vetor de entrada M , conforme (26), e também o resultado da aplicação da DCT sobre M , que se obteve M' , apresentado na Figura 5b. Após aplicação da DCT , o vetor transformado M' (Figura 5b) é submetido à operação de ofuscação, resultante no vetor transformado ofuscado M'' (Figura 5c). Nota-se que as inversões acontecem nas posições em que a chave, descrita em (27), apresenta o valor de (-1) . Na Figura 5d está representada a mensagem transmitida T . Este sinal foi obtido pela aplicação de uma transformada cosseno inversa sobre M'' (Figura 5c).

A Figura 5d contém a representação do vetor transmitido T , após a aplicação de todo o processo de transmissão. Fica evidente a diferença do mesmo para o vetor M (Figura 5a).

Outra informação importante observada na Figura 5 é que se notam as mudanças de base. A Figura 5a está no domínio do tempo, enquanto a Figura 5b está no domínio da frequência: isto acontece pela mudança de base provocada pela DCT . A Figura 5b e Figura 5c estão na mesma base, pois a operação de ofuscação não é uma mudança de base, portanto, não há mudança de domínio. Entre a Figura 5c e Figura 5d, novamente observa-se uma mudança de base, desta vez pela aplicação da $IDCT$.

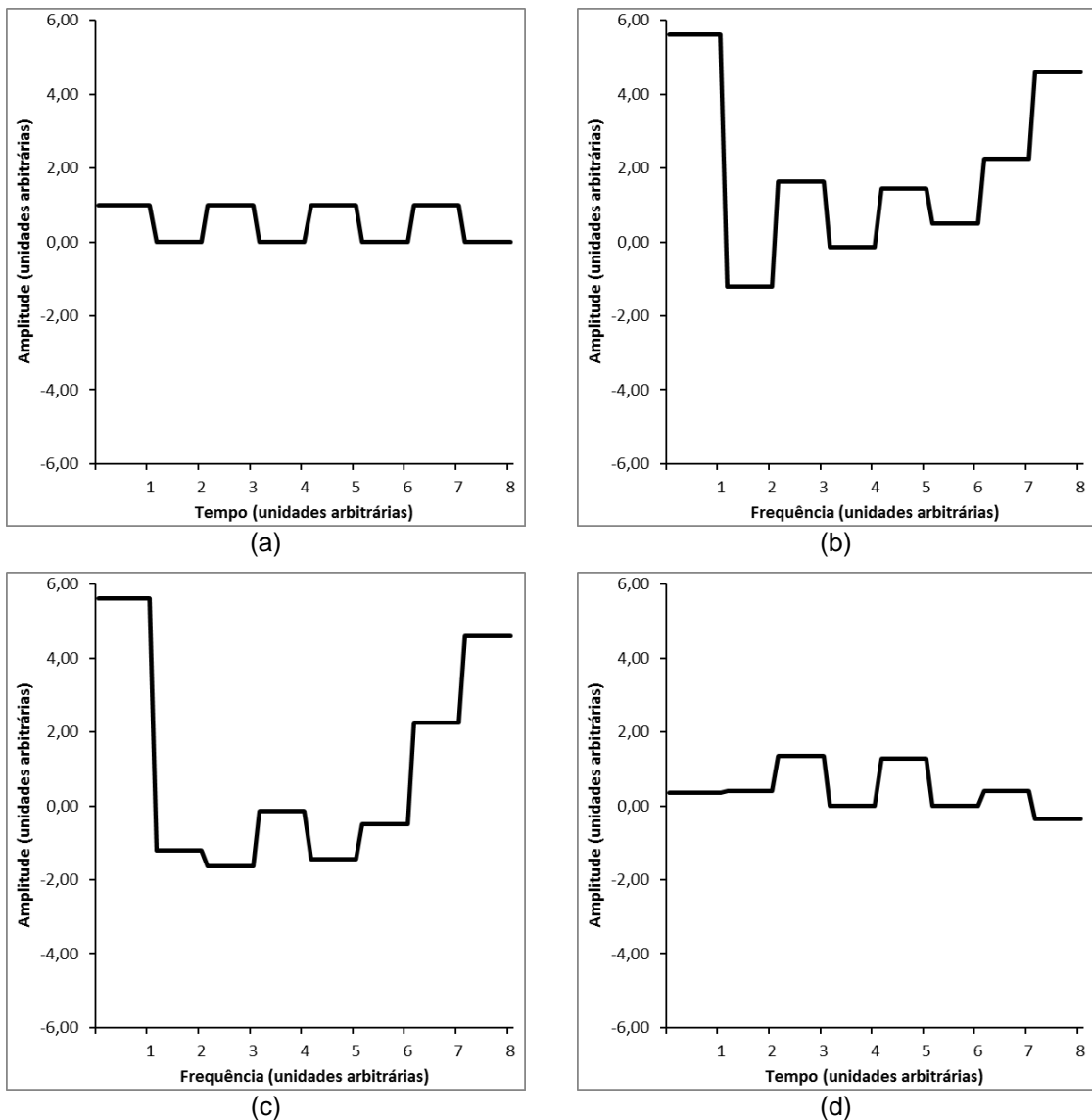


Figura 5: Mensagem original M (a), vetor transformado M' (b), vetor transformado e ofuscado M'' (c) e sinal transmitido T (d).

Com um número pequeno de vetores, ainda fica claro que um sinal está ali sendo transmitido, porém, não é possível avaliar se o dado está em banda base e é digital, ou seja, um eventual espião terá dificuldade para descobrir o formato de modulação utilizada na transmissão de dados.

A Figura 6 representa os vetores no sistema de recepção.

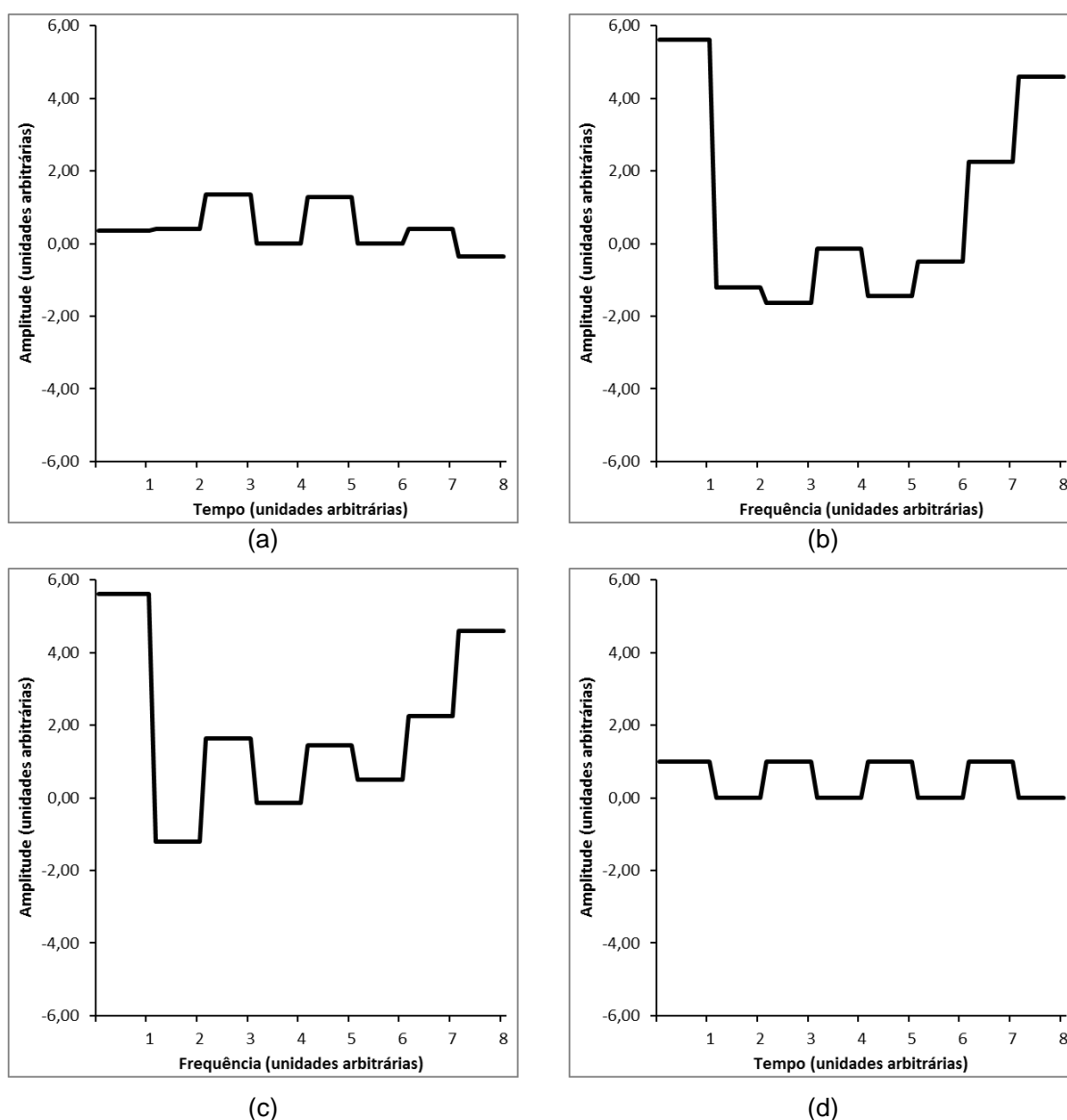


Figura 6: Sinal T recebido (a), sinal transformado e ofuscado T' (b), sinal transformado T'' (c) e mensagem recuperada T''' (d).

Após a recepção sem erros, conforme foi estabelecido na sessão 3.1, sobre o vetor T (Figura 6a) será aplicada a DCT , resultando em T' (Figura 6b). O vetor transformado embaralhado T' (Figura 6b) foi submetido à função de

desofuscação e se obteve o vetor T'' (Figura 6c). Na sequência, é aplicada a $IDCT$ sobre T'' , que resulta na mensagem recuperada T''' (Figura 6d).

Tal qual Figura 5, também se observa as mudanças de base entre domínio tempo e domínio frequência. Outro ponto importante: a Figura 6 possui as mesmas imagens da Figura 5, porém em ordem invertida.

Compara-se com a Figura 5a, que representa a mensagem M original, com a Figura 6d, para concluir que são iguais. O mesmo pode ser feito entre as figuras Figura 5b e Figura 6c, Figura 5c e Figura 5b e, por fim, Figura 5d e Figura 6a, respectivamente iguais entre os pares. Tal comparação comprova toda a fundamentação realizada no capítulo 3.

4.4 Testes Não-Funcionais

De acordo com (Dorothy Graham, 2008), testes não funcionais são os testes realizados sobre um sistema com o objetivo diferente de avaliar seu correto funcionamento. Ou seja, cria-se uma situação de teste para forçar que o sistema falhe e justamente tal falha define que houve sucesso como resultado do teste. Em criptografia, este teste é realizado de forma que, se o sistema se comportar de acordo com o que é esperado, irá produzir um resultado errôneo, ou seja, o resultado errado significa proteção do sinal.

Na seção 4.3, foi apresentado um estudo de caso em que é possível ofuscar um sinal e em seguida desofuscá-lo, para recuperação da mensagem original. No caso de técnicas de criptografia, um segundo teste seria empregar uma chave errada na recuperação do sinal. Para este teste, é importante frisar que o espião tem total conhecimento desta técnica, bem como usará um equipamento de espionagem ideal, ou seja, o espião capturaria um sinal idêntico ao da recepção, T . A única informação desconhecida pelo espião é a chave usada.

Para este teste, a mensagem de entrada será a mesma mensagem usada na seção 4.1, definida em (26). Esta mensagem será ofuscada com a chave da sessão 4.3, definida em (27). O espião desconhece a chave, portanto,

utilizaria uma chave diferente. Para representar isso, é gerada uma nova chave aleatória, K_e , de acordo com (30):

$$K_e = [-1, 1, 1, 1, 1, -1, 1, 1] \quad (30)$$

O resultado obtido ao tentar desofuscar o dado com uma chave diferente da ofuscação é apresentado na Tabela 2. Nesta tabela, estão descritos os vetores T_e' , T_e'' e T_e''' , vetores de recepção obtidos com uma chave errada, análogos aos vetores de recepção T' , T'' e T''' .

Tabela 2. Resultados não-funcionais da técnica.

n	M	M'	M''	T	T_e'	T_e''	T_e'''
1	1,00	5,60	5,60	0,36	-5,60	5,60	-0,98
2	0,00	-1,20	-1,20	0,40	-1,20	-1,20	-1,06
3	1,00	1,63	-1,63	1,34	-1,63	-1,63	0,18
4	0,00	-0,14	-0,14	0,00	-0,14	-0,14	-1,05
5	1,00	1,43	-1,43	1,27	-1,43	-1,43	0,26
6	0,00	0,49	-0,49	-0,01	0,49	-0,49	-0,58
7	1,00	2,24	2,24	0,41	2,24	2,24	0,02
8	0,00	4,60	4,60	-0,36	4,60	4,60	-0,61

A tabela 3 torna esta situação mais evidente ao comparar os vetores corretos com os vetores errados:

Tabela 3. Comparação entre resultados funcionais e não funcionais.

n	T''	T'''	T_e''	T_e'''
1	5,60	1,00	5,60	-0,98
2	-1,20	0,00	-1,20	-1,06
3	1,63	1,00	-1,63	0,18
4	-0,14	0,00	-0,14	-1,05
5	1,43	1,00	-1,43	0,26
6	0,49	0,00	-0,49	-0,58
7	2,24	1,00	2,24	0,02
8	4,60	0,00	4,60	-0,61

Como se pode observar, o sinal obtido T_e''' não repete o sinal correta T''' e tampouco repete M . Portanto, é possível dizer que houve erro na recuperação do sinal.

Novamente, para avaliar melhor este comportamento, adota-se o mesmo tipo de gráficos realizados na seção 4.4. Todos os gráficos de transmissão são novamente idênticos com os gráficos da Figura 5 e não serão repetidos. O uso de uma chave errada começa a se manifestar quando se aplica a operação de desofuscação, ou seja, para tentar reproduzir T'' conforme (24).

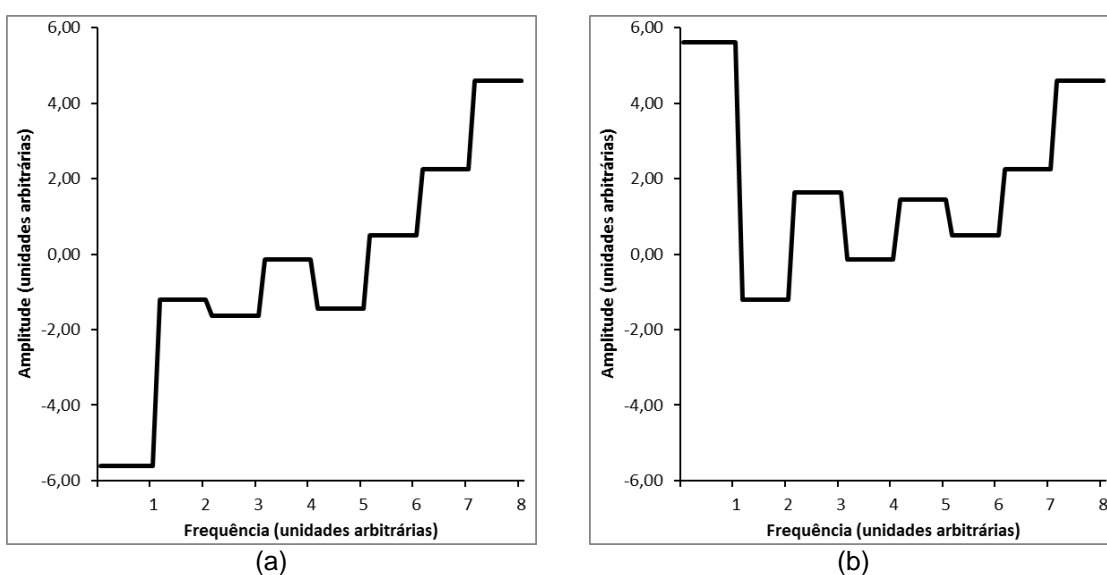


Figura 7: Vetor T_e'' obtido a partir de chave errada (a) e correto T'' (b).

Na Figura 7a está o sinal transformado e com as componentes modificadas com o uso de chave errada, ou seja, equivalente a T_e'' . Ao comparar com o gráfico da Figura 7b, que representa T'' nota-se que as componentes que deveriam ter sido invertidas ou mantidas para recuperação da mensagem M não foram corretamente decifradas pelo espião, ou seja, uma chave errada gerou um sinal equivalente a T'' errado.

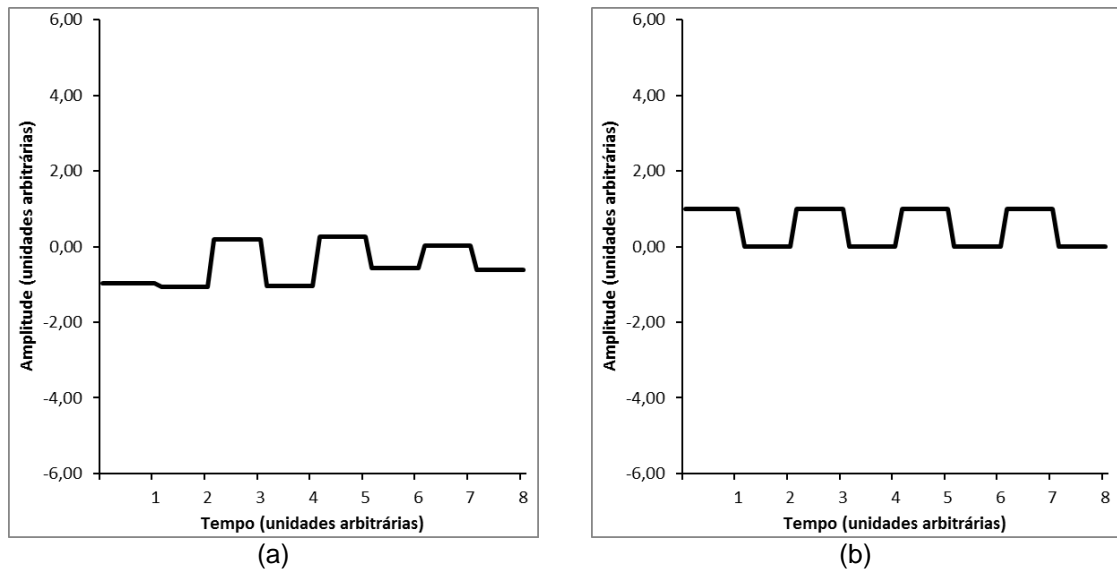


Figura 8: Mensagem recuperada com chave errada T_e''' (a) e mensagem recuperada correta T''' (b).

Este sinal errado resulta em uma mensagem errada. Na Figura 8a, o sinal recuperado final do espião pouco se parece o sinal original recuperado usando a chave correta, na Figura 8b. É difícil, portanto, o espião saber quais bits foram transmitidos em 0 e 1, ou seja: sem tal informação, reduz-se muito a oportunidade de ataques de força bruta.

4.5 Testes sobre o Tamanho do Vetor de Dados

Como a chave tem o mesmo tamanho do vetor na operação de ofuscação proposta, há implicação que outro fator determinante na robustez da chave é o seu tamanho de palavra.

Em todos os casos apresentados até então, utilizou-se apenas 1 *byte*. Neste caso, é possível se obter até 256 chaves diferentes. Contudo, ao se aumentar a palavra para 256 *bits*, por exemplo, obtêm-se 2^{256} chaves distintas, algo que dificultará em muito a recuperação da mensagem por um espião. De modo a ilustrar, ainda que qualitativamente, os efeitos deste aumento dos tamanhos de palavra e de chave, nas Figura 9, Figura 10 e Figura 11, mostra-se as mensagens transmitidas para 8, 16, 32 e 64 *bits*.

Nestes resultados, em que se utilizam vetores de maior dimensão, considera-se que será utilizado um vetor com alternância de *bits*, semelhante ao vetor M definido em (26).

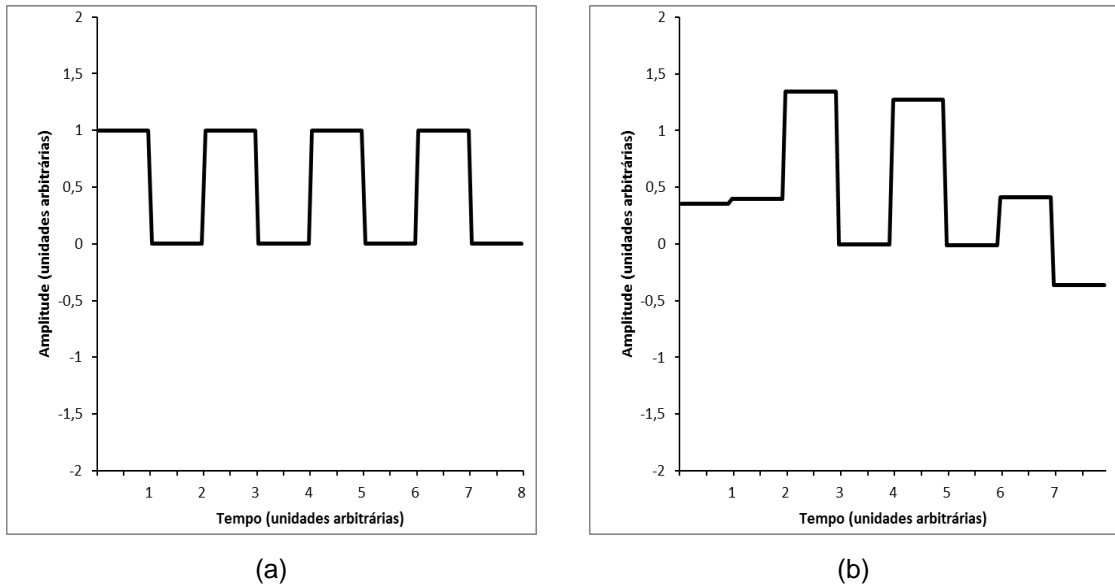


Figura 9: Resultado da aplicação da técnica para 8 bits. Sinal original M (a) e o sinal transmitido T (b).

Na Figura 9a, está o vetor de entrada M para 8 *bits* e na Figura 9b está o vetor de transmissão T após aplicação da técnica.

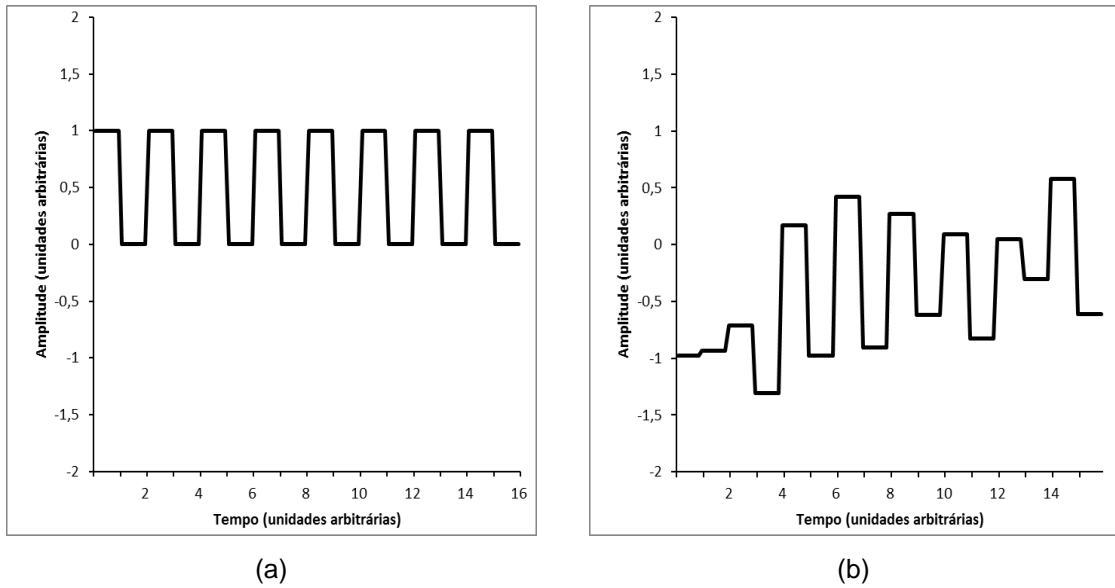
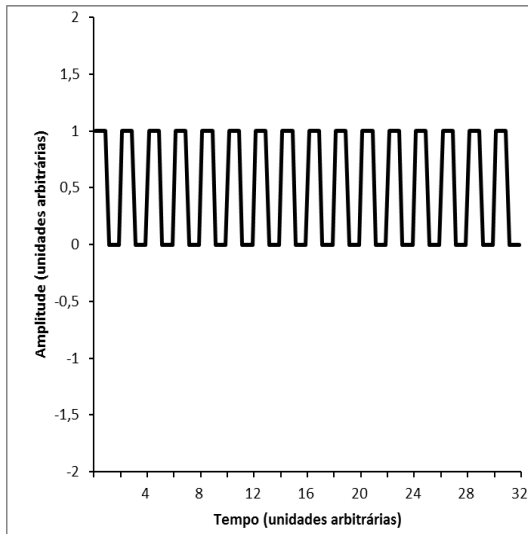


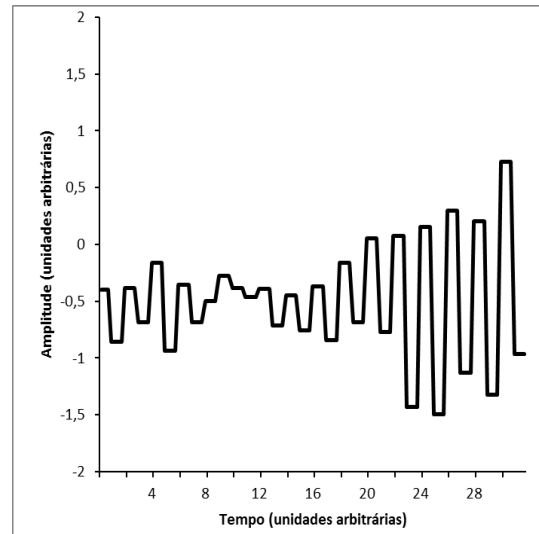
Figura 10: Resultado da aplicação da técnica para 16 bits. Sinal original M (a) e o sinal transmitido T (b).

Na Figura 10a, está representado o vetor M para 16 bits, semelhante ao vetor de 8 bits da Figura 9a, porém com largura de 16 bits. Já na Figura 10b está representado o sinal transformado para transmissão após a técnica ter sido aplicado sobre este vetor.

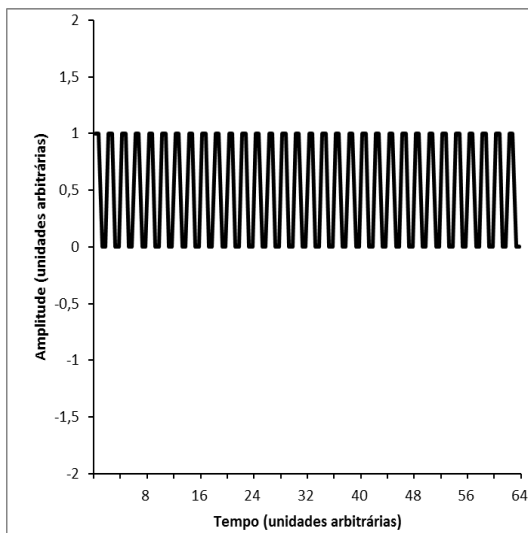
De maneira análoga, observando-se a Figura 11, os vetores M para 32 e 64 bits (a) e (c), bem como os sinais de transmissão T respectivos após aplicação da técnica em (b) e (d). Nota-se um efeito de degradação na aparência do sinal (fazendo o mesmo parecer ruído). Isto, por consequência, aumenta a qualidade de ofuscação do sinal transmitido.



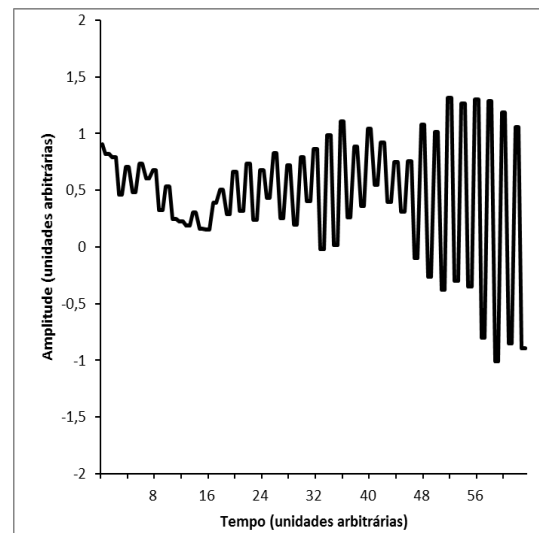
(a)



(b)



(c)



(d)

Figura 11: Resultado da aplicação da técnica para 32 e 64 bits. Sinal original M (a e c) e o sinal transmitido T (b e d).

4.6 Testes de Robustez à Espionagem

Para violar os dados protegidos com a utilização desta técnica, inicialmente, um espião deve conhecer três fatores:

- Saber qual sistema de criptografia foi aplicado, ou seja, qual operação de transformada;
- Conhecer a função ou o procedimento de ofuscação utilizado;
- Conhecer qual a chave ou o procedimento reverso usado para ofuscar o sinal.

Para avaliar como um espião observaria o sinal recebido, é necessário estimar algumas tentativas de ataque. Na seção 4.4, foi proposto um ataque em que o espião tinha total conhecimento da técnica e apenas não possuía a chave de criptografia.

A definição das funções OFUSC e IOFUSC na seção 3.6 permitiu modificar e adaptar a estimativa do tempo máximo para quebra de criptografia, definida em (14).

$$T_{quebra} = 2^N * (T_{IOFUSC} + T_{TRANSF} + T_{ITRANSF}) \quad (31)$$

Além disso, é possível comparar o tempo de quebra de chave para estes diferentes tamanhos. Conforme foi definido em (31), a utilização de métodos de exaustão para quebra de chave levará a um tempo proporcional ao número de chaves possíveis. Supondo que o tempo total para os cálculos seja:

$$T_{IOFUSC} + T_{TRANSF} + T_{ITRANSF} = 1ns, \quad (32)$$

Para uma chave de 8 *bits*, aplica-se (25) e (32) em (31) e obtém-se o tempo máximo para quebra de uma mensagem criptografada com este tamanho de chave, em (33).

$$T_{quebra} = 256 ns \quad (33)$$

De forma semelhante, para uma chave de tamanho de 256 *bits*, também é possível obter o tempo máximo para quebra de chave:

$$T_{quebra} = 3,67 \cdot 10^{68} s \quad (34)$$

Portanto, conclui-se que o aumento do tamanho do número de bits da mensagem e, conseqüentemente, no tamanho da chave, contribui significativamente para a proteção e segurança da mensagem.

Para se testar a robustez da técnica criptografia do sinal, foram realizados mais alguns testes com diferentes chaves erradas. Considerando-se o mesmo vetor *M* utilizado anteriormente e as chaves incorretas definidas de acordo com:

$$K_a = [1, -1, -1, -1, -1, -1, -1, -1] \quad (35)$$

$$K_b = [1, -1, -1, 1, 1, 1, 1, -1] \quad (36)$$

$$K_c = [-1, 1, 1, 1, 1, -1, 1, 1] \quad (37)$$

$$K_d = [-1, 1, 1, 1, 1, -1, 1, 1] \quad (38)$$

Aplicando-as a recuperação do vetor determinado em (26), usa-se o mesmo método definido na seção 4.4, na qual se mantém a mesma chave e mensagem dos testes funcionais e realizam-se novos testes não funcionais com as chaves erradas obtidas acima. Em seguida, obtém-se como resultado as mensagens exibidas na Tabela 4.

Nesta tabela, observam-se os diferentes resultados obtidos para o vetor de mensagem recuperada T''' em função das chaves erradas definidas em (35), (36), (37) e (38). Ao comparar este resultado com a mensagem M ou T''' corretas, observa-se que a escolha de diferentes chaves leva a resultados errados, portanto, fortalecendo a hipótese de que a técnica funciona e realmente protege os dados.

Tabela 4. Resultados Variados de Chaves Erradas.

n	$T'''(Ka)$	$T'''(Kb)$	$T'''(Kc)$	$T'''(Kd)$
1	1,04	0,89	0,38	-0,56
2	0,94	0,96	0,38	-0,91
3	-0,10	0,51	1,31	-0,79
4	1,09	0,20	0,00	-0,59
5	-0,38	0,00	1,31	-0,15
6	0,67	0,96	0,00	-0,72
7	-0,01	-0,51	0,38	0,95
8	0,50	0,89	-0,38	-0,67

E em forma gráfica na Figura 12. Como se pode observar, fica ainda mais claro que um eventual espião terá dificuldade na interpretação dos resultados, visto que estes são bem diferentes da mensagem original M (Figura 5a).

Contudo, é importante salientar que é possível que o espião, de fato, conheça o padrão usado. Imagina-se que o espião, com conhecimento da técnica, uma vez que a mesma seja pública, tentará, a partir do resultado que se obteve, recuperar a mensagem de qualquer jeito. Sendo assim, no caso da aplicação em

banda base, após a aplicação da técnica e a utilização de alguma chave (correta ou não), o espião esperará encontrar um sinal digital.

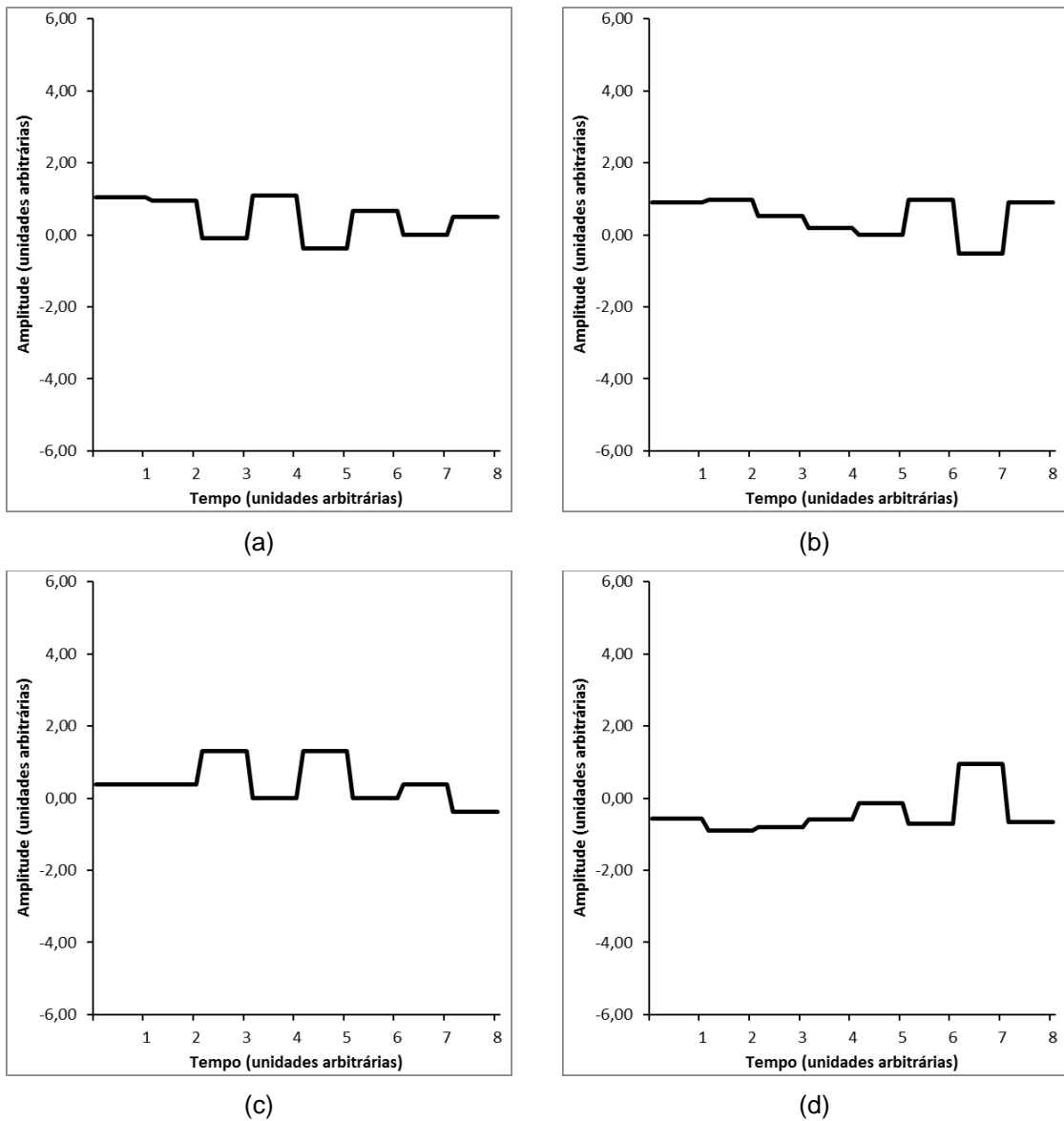


Figura 12: Resultados com uso de chave errada para K_a (a), K_b (b), K_c (c) e K_d (d).

Neste caso, o que espião perceberá é que o sinal não segue os padrões convencionais de barramentos de linha presentes na maioria dos dispositivos eletrônicos digitais atuais. Por isso, o espião precisaria escolher como referência algum formato de ataque e algum formato de recuperação de dados.

Faz-se necessário avaliar, então, que o espião poderá tentar forçar a detecção dos bits da mensagem aplicando um detector digital no mesmo padrão esperado para a mensagem. Para tal, testou-se a técnica com um detector digital

baseado na definição usada para o padrão LVCMOS (JEDEC Solid State Technology Association, 2007).

Propõe-se o seguinte modelo de sistema detector de *bit*:

- Se o resultado de determinado componente vetor T''' for maior ou igual que 0,5, o bit recuperado deve ser 1.
- Se o resultado de determinado componente vetor T''' for menor que 0,5, o bit recuperado deve ser 0.

Há ainda mais uma importante condição. Desconsidera-se o fato que o vetor recuperado com chave errada T''' contém valores intermediários entre o zero elétrico e o um elétrico, algo que, para padrões LVCMOS ou semelhantes, acarretaria não só uma detecção potencialmente errada bem como propagaria erros ao longo do sistema, conforme mencionado por (Carro, 2001). Ou seja, o detector ideal do espião, neste experimento, é melhor que o detector descrito em (JEDEC Solid State Technology Association, 2007). Formalmente:

$$Detec(T''') = \begin{cases} 0, & T''' < 0,5 \\ 1, & T''' \geq 0,5 \end{cases} \quad (39)$$

Em (39), a função $Detect(x)$ representa o detector empregado pelo espião. O espião aplicaria este detector sobre cada componente dos vetores obtidos (Tabela 4), ou seja, forçar níveis digitais após a tentativa de quebrar a chave com uma chave errada. Os resultados obtidos em número de bits errados para as chaves utilizadas nos experimentos realizados foram:

Tabela 5. Resultados de ataque de digitalização por força bruta.

Chave	Detectado	Número de erros
K _a (35)	[1, 1, 0, 1, 0, 1, 0, 1]	7
K _b (36)	[1, 1, 1, 0, 0, 1, 0, 1]	4
K _c (37)	[0, 0, 1, 0, 1, 0, 0, 0]	2
K _d (38)	[0, 0, 0, 0, 0, 0, 1, 0]	3

Observa-se, na coluna 2 da Tabela 5, os vetores digitais resultantes muito diferentes do vetor de entrada, definido em (26). Nesta situação, é possível

que um espião considere o equipamento ou sistema defeituoso e desista do ataque.

4.7 Diagrama de Olho

Diagramas de olho são recursos eficientes para se avaliar a qualidade da recepção ou condição de um sinal. Mediante a avaliação do diagrama de olho, é possível medir *jitter*, *wander*, impedância, prever efeitos não-lineares, dentre outros (On Semiconductor, 2015).

É esperado, então, que uma técnica que trabalhe em banda base e em camada física venha a ser submetida a um teste de diagrama de olho. Neste trabalho, optou-se por utilizar a transmissão e detecção ideais, ou seja, conversão de dados analógicos para digital ideais, ou ainda, no limite da precisão numérica de um computador pessoal convencional (o mesmo vale para a situação oposta, conversão digital para analógico).

A técnica de ofuscamento, por outro lado, transforma o sinal digital em um sinal analógico, portanto, sem diagrama de olho. É possível demonstrar esta ausência de diagrama de olho neste trabalho com o uso de avaliação dos conjuntos numéricos dos vetores. Ao avaliar a equação da *DCT*, descrita em (17), é possível definir os intervalos de resultado de amplitude. Primeiro, avaliando o vetor de entrada x , sabemos que pode obter valores entre 0 e 1:

$$\{m_n \in \mathbb{N} \mid 0 \leq x \leq 1\} \quad (40)$$

Da mesma forma, o termo em cosseno de (17) pode possuir valores conforme (41).

$$\left\{ \cos \frac{\pi}{A} \cdot \left(a - \frac{1}{2}\right) \cdot \left(b - \frac{1}{2}\right) \in \mathbb{R} \mid -1 \leq \cos \frac{\pi}{A} \cdot \left(a - \frac{1}{2}\right) \cdot \left(b - \frac{1}{2}\right) \leq 1 \right\} \quad (41)$$

Portanto, ao avaliar (40) e (41), é viável definir os dois extremos das componentes de M' , conforme (43). Ao retornar o resultado limite do termo cosseno para a equação da *DCT*, considerando o valor de N conforme (25), obtém-se:

$$2. \sum_{a=0}^{N-1} m'_a * (\pm 1) \leq \pm 8 \quad (42)$$

Portanto, o resultado final para as componentes do vetor M' :

$$\{m'_n \in \mathbb{R} \mid -8 \leq x \leq 8\} \quad (43)$$

Em seguida, utiliza-se o mesmo procedimento para determinar o intervalo de T . Utilizando o resultado de (41) e o resultado de (43), tendo como base a equação da IDCT definida em (16), com cuidado de considerar o termo normalização, obtém-se o intervalo:

$$\{t_n \in \mathbb{R} \mid -4 \leq x \leq 4\} \quad (44)$$

Ao avaliar este resultado, embora estes sejam os limites absolutos, pode-se perceber que a mensagem T pode assumir qualquer valor dentro do intervalo definido, ou seja, não fica definida em níveis discretos. Ao sobrepor o resultado das transformadas para 8 bits , obtém-se um gráfico conforme a Figura 13.

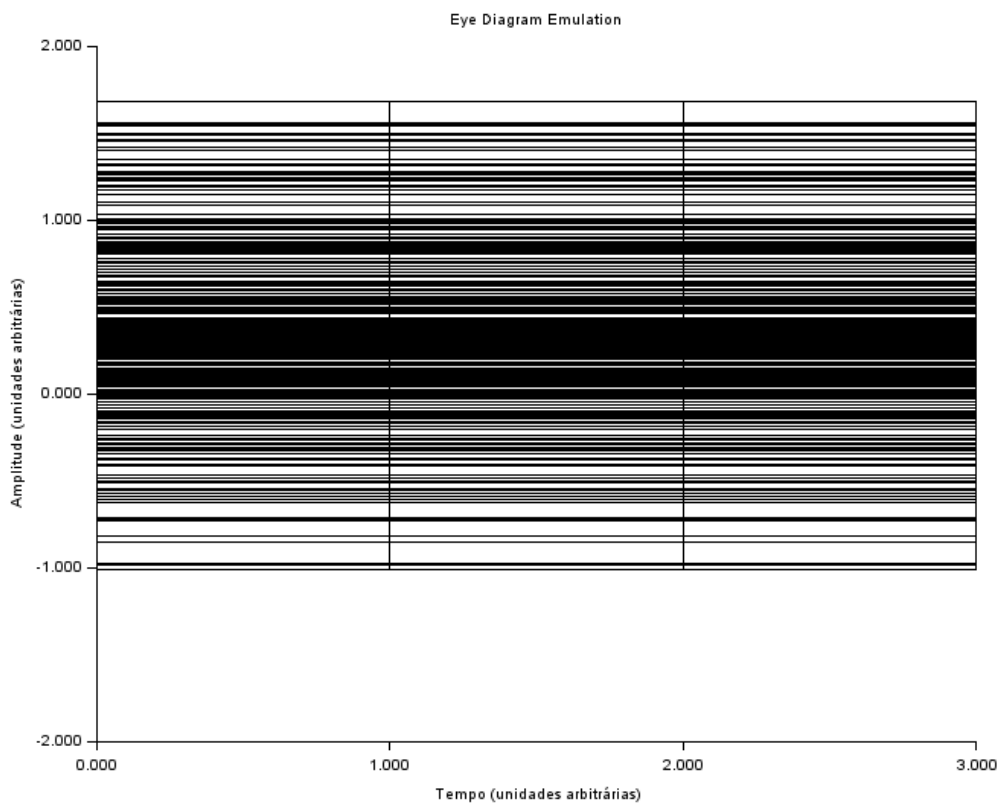


Figura 13: Gráfico acumulado do Vetor T

Nesta Figura 13, acumulou-se o vetor resultante T para as 256 possíveis entradas de 8 *bits*. Observa-se que o resultado em amplitude não permite a formação do olho, pois sem uma amplitude definida com valores discretos, conforme já era esperado pelo resultado de (44), é inexistente o diagrama de olho.

Observa-se, ainda na Figura 13, que tal sobreposição gerou linhas verticais através das linhas horizontais de amplitude. Tais linhas indicam a transição dos símbolos. Isto é equivalente às rampas de subida e descida de um diagrama de olho convencional. Mesmo assim, não é possível determinar níveis de amplitude discretos e, portanto, avaliar um diagrama de olho.

Estimar que, em um caso de conversão digital analógica real, a possibilidade de comparar a técnica apresentada neste trabalho com as técnicas

de modulação *PAM NRZ* (*pulse amplitude modulation, non return to zero*), descritas em (Oppenheim, et al., 2015).

Neste cenário, haverá presença de diagrama de olho. A quantidade de olhos do diagrama será igual ao número de níveis permitidos pelo conversor digital analógico, portanto, este diagrama de olho será equivalente ao de uma modulação *PAM* com o mesmo número de níveis. Como exemplo, se usado um conversor com 12 bits de resolução, pode-se dizer que o diagrama de olho da técnica se assemelharia a uma modulação *PAM1024*.

4.8 BER

Testes de *BER* (*bit error rate*) são testes que avaliam a quantidade de erros em um canal de largura de banda finita ao se dividir o número de bits errados recebido pelo número total de bits recebidos (Mazda, 1993). Conforme a seção 3.1, se adotou um canal ideal, livre de erros e, de maneira implícita, de largura de banda infinita. A técnica não penaliza a comunicação e o canal foi definido como ideal. Estas situações impossibilitam análises de *BER* em função de erros no introduzidos pelo canal de transmissão.

Apesar disso, estimaram-se os erros encontrados por um invasor na seção 4.4. Observa-se que, nestes casos, o erro não acontece por erros no canal de transmissão, mas, sim, por uso de chave errada de criptografia. Esta mesma condição pode ser observada na sessão 4.6, que, além dos erros de utilização de chave errada, há ainda erros por utilização de um detector errado. Nenhum destes pode ser considerado, portanto, *BER* conforme a definição de (Mazda, 1993).

Além de desconsiderar efeitos do canal de transmissão também se desconsidera erros de quantização ou imprecisão numérica em função de conversão entre analógico digital. A adição destes efeitos corresponderia à adição de ruído ao canal de comunicação.

Para poder realizar estimativas de *BER*, é importante definir alguns parâmetros:

- O Sistema físico transmissor (sensibilidade, níveis de tensão e etc);
- Os parâmetros do meio;
- O sistema físico receptor (sensibilidade, níveis de tensão e etc).

Conforme discutido na seção 4.7, uma vez que os parâmetros acima estejam definidos, a taxa de ocupação de canal, estimativa de *BER* e demais parâmetros podem ser calculados usando as mesmas definições usadas para sistemas em modulação *PAM*. Para tal, basta novamente considerar equivaler o número de níveis definidos pelo conversor digital analógico com a modulação *PAM*. No exemplo também discutido na seção 4.7, um conversor digital analógico de 12 bits equivaleria a uma modulação *PAM 1024*.

Este elevado número de níveis pode ser um problema. Novamente, (Oppenheim, et al., 2015) avalia algumas condições necessárias para o funcionamento de modulações *PAM*, tais como efeitos do ruído e interferência entre símbolos, que podem ser aplicados à técnica proposta. Encontra-se uma comparação entre *PAM* e outras modulações em (Elganimi, 2013).

Tais avaliações permitem afirmar que a taxa de erro medida não corresponde ao que normalmente entende-se por *BER*, mas corresponde ao erro provocado por uso incorreto da técnica por parte de um eventual espião. Dependendo do sistema de recepção utilizado pelo invasor, ele pode obter diferentes taxas de erro.

4.9 Viabilidade de Implementação em Hardware

A principal aplicação prevista para esta técnica é em sistemas embarcados, nos pinos de dispositivos de microeletrônica como memórias e processadores, por exemplo, e dispositivos de *IOT*. Contudo, o escopo deste trabalho não contempla a experimentação da técnica. Para poder concluir que é viável implementar a técnica, é possível fazer uma avaliação de estudos e técnicas semelhantes ou que tenham foco na *DCT*.

Algumas técnicas citadas na seção 3.5 como semelhantes à técnica apresentada neste trabalho possuem implementações em hardware. Por

exemplo, (Györ, et al., 2012) descreve uma implementação em *FPGA* para a técnica descrita por (Lyubashevsky, et al., 2008). Ao analisar este trabalho de (Györ, et al., 2012), observa-se que os autores obtiveram sucesso na implementação de em *hardware* da *FFT*. Isto é um indicativo positivo para este trabalho, pois conforme a seção 2.1, a *DCT* pode ser entendida como um caso especial e reduzido da *FFT*.

O principal elemento da técnica é o elemento de transformação, ou seja, a *DCT*. É válido afirmar, em função disso, que o principal bloco para implementação em *hardware* da técnica seja o bloco que executará a operação de transformada cosseno. Isto é outro ponto favorável à esta atividade: a *DCT* é uma operação de ampla utilização, de tal forma que existem vários estudos acadêmicos voltados para concepção de novos e melhores algoritmos, até obras inteiras especializadas no assunto, como por exemplo (Paul, et al., 2013).

Assim como estudos sobre novos algoritmos, existem também muitas discussões sobre implementações de *DCT* e *IDCT* em *hardware*, especialmente, em dispositivo de matriz de portas lógicas programáveis em campo, os *FPGAs* (Imam, et al., 2016). Também é possível encontrar implementações em circuitos integrados dedicados (*asic*) ou em circuitos específicos de aplicação padrão (*assp*) (Lee, et al., 2004). Usualmente estas discussões fazem uso de alguma linguagem de descrição de *hardware* (*HDL*) (Murthy, 2004).

As discussões sobre a *DCT* tendem a priorizar soluções e em técnicas para desempenho, como, por exemplo (Yu, et al., 2012), (Tumeo, et al., 2007) e (Lee, et al., 2009). Porém, há também trabalhos com foco em eficiência energética (Livramento, et al., 2011), (He, et al., 2013), e (Schlachter, et al., 2016). Por fim, há também vários estudos para redução de custos¹, como por exemplo (Gong, et al., 2004), (Hsia, et al., 2007) e (Belkouch, et al., 2010).

¹ Neste caso, reduzir custo de algo que vai dentro de um chip significa reduzir a área em silício ocupada.

4.10 Observações Gerais

Nos estudos sobre esteganografia, por exemplo, utilizam-se padrões alternados de cores. Analogamente, neste trabalho, foi usado, como amostra, um padrão alternado de *bit*. Para gerar as chaves, utilizam-se funções pseudoaleatórias.

Conforme observado na seção 4.2, na Tabela 1 e na Figura 5, observa-se que a técnica pode ser utilizada potencialmente para ofuscar um sinal e recuperá-lo. Da mesma forma, ao avaliar o teste realizado na seção 4.4, tanto na Tabela 2 como da Figura 7 e Figura 8, um espião com total conhecimento da técnica não conseguirá recuperar o sinal original.

Além disso, como seria de se esperar, observa-se que o tamanho da amostra e da chave tem papel fundamental na proteção à ataques de força. Quanto maior for o vetor e, por consequência, o tamanho da chave, maior será o tempo necessário para quebrar a codificação.

Testou-se, ainda, um detector digital dentro assemelhado ao padrão *LVC MOS*, obtendo taxas elevadas de erro, acima de 1 *bit* errado a cada 8 *bits* transmitidos.

Aplicou-se esta técnica sobre sinais digitais, contudo, a mesma pode ser utilizada para ofuscar sinais totalmente analógicos. Isso permite expandir a aplicação da técnica para comunicação segura de voz, por exemplo, ao permitir que a comunicação de voz seja protegida no trecho entre o microfone e o elemento transmissor, em um sistema embarcado como um celular, ou ainda entre uma central privada (*PBX, private branch exchange*) e o ramal de usuário.

Outro resultado importante é que além da *DCT*, a técnica pode ser aplicada com outras operações de transformada. Em sinais modulados transmitidos em quadratura, por exemplo, *FM* estéreo, em que há transmissão em portadora com mais de uma fase representada por sinais ortogonais, pode-se considerar a utilização de transformadas de Fourier sem prejuízo à simplicidade ou aumento no número de canais.

Finalmente, os resultados de simulação indicam que a técnica funciona para ofuscar sinais digitais em banda base. Os mesmos resultados sugerem ser possível aplicar a técnica para sinais analógicos e também que é possível estimar sua aplicação para sinais modulados.

5 Conclusão

Neste trabalho, apresentou-se uma técnica de criptografia baseada na ofuscação dos dados em banda base para aumentar a segurança de sistemas de comunicação e sistemas embarcados. O objetivo da técnica é proteger tanto os sinais transmitidos como os sinais internos dos equipamentos.

A técnica apresentada emprega operações de mudança de base e funções de ofuscação. Especificamente, utilizaram-se operações de transformada cosseno como a operação de mudança de base e inversão de amplitude como operação de ofuscação. Para avaliar esta proposta testou-se o sistema por meio de simulações.

Os resultados da simulação comprovaram que, ao aplicar a técnica sobre um sinal, este acabou sendo ofuscado e não apresenta semelhança com a mensagem original. Tal situação pode ser comprovado em valores numéricos (Tabela 1) e também por comparação visual (Figura 5).

A hipótese de que a técnica protege o sinal de um eventual espião também foi testada. Os resultados obtidos indicaram que um invasor não conseguiria recuperar o sinal original. Tal situação pode ser verificada numericamente (Tabela 2 e Tabela 3) ou por comparação visual (Figura 7).

Ainda para testar a hipótese de que a técnica proposta apresenta robustez a espionagem, foi calculado o tempo de quebra de chave para uma situação hipotética, sendo que, dependendo do tamanho de chave usada, o tempo máximo para quebra do segredo chega a vários anos. Para o caso de 256 *bits*, por exemplo, tem-se 10^{61} anos.

Simulou-se a tentativa de ataque com outras chaves e com a proposta de um detector de *bits* hipotético. Nestes testes, o resultado de *bits* errados percebido por um invasor, a *BER*, fica no mínimo, em $1,25 \times 10^{-1}$. Esta taxa de erros torna difícilimo recuperar alguma informação útil do sinal captado pelo espião e concluiu-se que é mais provável que o mesmo entenda o equipamento como defeituoso.

Para testes de diagrama de olho, foi demonstrado que em sistemas em que há conversão ideal de dados digitais (vetores de amplitude), com possibilidade de infinitos níveis, a ideia de diagrama de olho perde o significado. Contudo, também se avaliou que esta técnica, em ambientes reais, é assemelhada às modulações tipo *PAM NRZ*. Esta é uma comparação que permite utilizar, na eventualidade de se realizar testes físicos sobre a técnica, os procedimentos e ideias existentes para esta modulação.

Devido à adoção de uma condição de canal ideal com banda infinita para facilitar a concepção e avaliação inicial da técnica, não foram possíveis testes de *BER*. É importante ainda salientar que a *BER* poderá ser avaliada em um sistema real fim-a-fim. Nesta situação em que os efeitos do canal de comunicação (ruído, não linearidades e etc) estiverem presentes, estima-se que o comportamento do sistema em teste seja semelhante ao de sistemas que empreguem modulações *PAM*.

Também foi possível concluir, ao avaliar estudos em trabalhos correlatos, independente dos mesmos terem atingido objetivos diferentes dos deste trabalho, que é viável uma futura implementação em *hardware*.

Este trabalho, portanto, contribui com a definição e exploração de uma técnica de criptografia por ofuscação de sinal, em camada física e banda base. Além disso, é apresentada a perspectiva da implementação da mesma em protótipos. Com estes resultados, aproxima-se da aplicação da técnica proposta para fins de aumentar a segurança nos sistemas de comunicação e sistemas embarcados, em geral.

5.1 Trabalhos Futuros

Os efeitos do ruído e da amostragem poderão ser estudados para se avaliar o impacto dos mesmos sobre a técnica apresentada.

No que se refere à amostragem, será importante investigar se será necessário o desenvolvimento de alguma estratégia para a recuperação de clock.

Diferentes técnicas de ofuscação poderão ser investigadas, isso sob vários aspectos, por exemplo, como estas podem dificultar a detecção da chave criptográfica. Dentre estas técnicas, em substituição da função de ofuscação definida na 3.6, listam-se:

- Troca de componentes (*swap*): neste caso, no lugar de inverter a amplitude de alguns componentes, opta-se por trocá-las de lugar, gerando um sinal diferente do original.
- Adição de valor (soma vetorial): adicionam-se artificialmente valores constantes ao vetor transformado para modificar o sinal.
- Sobreamostragem: amostra-se cada bit mais de uma vez, fazendo com que o resultado pareça possuir mais informação do que a mensagem original².

No presente estudo sobre a técnica, utilizou-se apenas uma chave constante para recuperar a mensagem. Como esta operação é aplicada termo a termo, é possível criptografar cada vetor de dados da mensagem com uma chave única e circular estas chaves de maneira sequencial. Isto garantiria que mesmo que o invasor quebre a chave de uma das palavras, ele teria acesso a apenas uma palavra. Esta variação de chave implica na necessidade de avaliação de qualidade de chave, modo real de geração da chave, aleatoriedade etc.

² Neste caso, é necessário aplicar uma das já citadas técnicas de ofuscação, já que, por si só, esta técnica não irá ofuscar o sinal.

6 Referências

A Cryptographic, Discrete Cosine Transform and Frequency Domain

Watermarking Approach for Securing Digital Images [Conferência] / A. Kester Quist-Aphetsi [et al.] // IPCV'15 - The 19th International Conference on Image Processing, Computer Vision, & Pattern Recognition. - Las Vegas : CSREA Press, 2016.

A Pipelined Fast 2D-DCT Accelerator for FPGA-based SoCs [Conferência] / A.

Tumeo Antonino [et al.] // ISVLSI '07 - IEEE Computer Society Annual Symposium on VLSI. - [s.l.] : IEEE, 2007.

All-Optical Cryptography through Spectral Amplitude and Delay Encoding

[Artigo] / A. Abbade M. L. F. [et al.] // Journal of Microwaves, Optoelectronics and Electromagnetic Applications. - Dezembro de 2013. - Vol. 12.

An energy-efficient 8x8 2-D DCT VLSI architecture for battery-powered

portable devices [Conferência] / A. Livramento Vinícius S. [et al.] // IEEE International Symposium on Circuits and Systems (ISCAS), 2011. - Rio de Janeiro : IEEE, 2011.

Circuit-Level Timing-Error Acceptance for Design of Energy-Efficient

DCT/IDCT-Based Systems [Artigo] / A. He Ku, Gerstlauer Andreas e Orshansky

Michael // IEEE Transactions on Circuits and Systems for Video Technology. - [s.l.] : IEEE, 2013. - 6 : Vol. 23.

Cisco - The Internet of Things - How the Next Evolution of the Internet Is

Changing Everything [Online] / A. Evans Dave // Cisco Systems - The Internet of Things. - Cisco Systems, 04 de 2011. - 27 de 09 de 2016. -

https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

Computer Networks [Livro] / A. Tanenbaum Andrew S. e Wetherall David J.. -

New Jersey : Prentice Hall, 2010. - 5ª.

Computing with Memory for Energy-Efficient Robust Systems [Livro] / A. Paul

Somnath e Bhunia Swarup. - New York : Springer, 2013. - Vol. 1.

Cryptography algorithm using DCT transform for colour images [Artigo] / A.

Tripathi Tribodh, Chourasia Bharti e Jain Anshuj // International Journal of Science, Engineering and Technology Research. - Janeiro de 2016. - 1. - Vol. 5.

Design and implementation of discrete cosine transform algorithm on FPGA device [Conferência] / A. Imam Elmubarak, Ahmed Mohamed Elhafiz Mohamed e Abdalla Ghassan // 2016 Conference of Basic Sciences and Engineering Studies (SGCAC). - Khartoum : IEEE, 2016. - Vol. 1.

Design of energy-efficient discrete cosine transform using pruned arithmetic circuits [Conferência] / A. Schlachter Jeremy, Camus Vincent e Enz Christian // 2016 IEEE International Symposium on Circuits and Systems (ISCAS). - Montreal : IEEE, 2016.

Dicionário Priberam de Língua Portuguesa [Online] / A. Priberam // Dicionário. - Priberam. - 26 de Dezembro de 16. - <https://www.priberam.pt/dlpo/ofuscar>.

Discrete Cosine Transform: Algorithms, Advantages, Applications [Livro] / A. Rao K. Ramamohan e Yip P.. - San Diego : Academic Press, Inc, 1990.

Efficient VLSI implementation of inverse discrete cosine transform [image coding applications] [Conferência] / A. Lee J., Vijaykrishnan N. e Irwin M.J. // IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). - Montreal : IEEE, 2004.

Ericsson Mobility Report 2016 [Online] / auth. Ericsson // Ericsson Mobility Report. - Ericsson, 06 2016. - 09 27, 2016. - <https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf>.

Estudo comparativo entre algoritmos das transformadas discretas de Fourier e Wavelet [Artigo] / A. Shirado Wilson Hissamu [et al.] // Revista Brasileira de Computação Aplicada. - Passo Fundo : [s.n.], 2015. - 3 : Vol. 7.

Exhaustive Key Search of the DES: Updates and Refinements [Conferência] / A. Quisquater Jean-Jacques e Standaert François-Xavier // Proceedings of SHARCS 2005, Special-purpose Hardware for Attacking Cryptographic Systems. - Paris : [s.n.], 2005.

Exhaustive Key Search on Clusters of GPUs [Conferência] / A. Barbieri Davide, Cardellini Valeria e Filippone Salvatore // Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International. - Phoenix : IEEE, 2014.

Foundations of Software Testing: ISTQB Certification [Livro] / A. Dorothy Graham Erik Van Veenendaal, Isabel Evans. - Londres : Cengage Learning EMEA, 2008.

H262 - Information Technology – Generic Coding of Moving Picture and Associated Audio Information: Video [Online] / A. International

Telecommunication Union // ITU-T. - ITU-T, Julho de 1995. - 15 de Novembro de 2016. - http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.262-199507-S!!PDF-E&type=items.

Hardware Implementation of Windowed Discrete Cosine and Discrete Sine Transforms in VHDL [Livro] / A. Murthy Smitha. - Charlotte : University of North Carolina at Charlotte, 2004. - Vol. 1.

High-Throughput Hardware Architecture for the SWIFFT / SWIFFTX Hash Functions [Conferência] / A. Györ Tamás [et al.] // Progress in Cryptology – LATINCRYPT 2012. - Santiago : Springer, 2012.

Improved implementation of a modified Discrete Cosine Transform on low-cost FPGA [Conferência] / A. Belkouch S. [et al.] // 5th International Symposium on I/V Communications and Mobile Network (ISVC). - Rabat : IEEE, 2010.

Internet of Things (IOT) [Online] / A. Cisco Systems // IoT Threat Environment. - Cisco Systems, 10 de 2015. - 24 de 7 de 2016. - <https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/C11-735871.pdf>.

JEDEC Standard JESD8C.01, Interface Standard for Nominal 3 V/3.3 V Supply Digital Integrated Circuits [Online] / A. JEDEC Solid State Technology Association. - Setembro de 2007. - 14 de Junho de 2016. - <http://www.jedec.org/sites/default/files/docs/jesd8c-01.pdf>.

Keeping Secrets in Hardware: the Microsoft XBox(TM) Case Study [Online] / A. Huang Andrew. - 26 de 05 de 2002. - 10 de 05 de 2016. - <ftp://publications.ai.mit.edu/ai-publications/2002/AIM-2002-008.pdf>.

New cost-effective VLSI implementation of a 2-D discrete cosine transform and its inverse [Artigo] / A. Gong Danian, He Yun e Cao Zhigang // IEEE Transactions on Circuits and Systems for Video Technology. - [s.l.] : IEEE, 2004. - 4 : Vol. 14.

New fast full search algorithms using DCT coefficients [Artigo] / A. Lee Hyuk, Jin Soonjong e Jeong Jechang // IEEE Transactions on Consumer Electronics. - [s.l.] : IEEE, 2009. - 2 : Vol. 55.

Performance Comparison between OOK, PPM and PAM Modulation Schemes for Free Space Optical (FSO) Communication Systems: Analytical Study [Artigo] / A. Elganimi Taissir Youssef // International Journal of Computer Applications. - New York : IJCA, 2013. - 11 : Vol. 79.

Physical Layer Security in Wireless Networks: a Tutorial [Artigo] / A. Shiu Yi-Sheng [et al.] // IEEE Wireless Communications. - [s.l.] : IEEE, Abril de 2011.

Principles of Spread-Spectrum Communication Systems 3rd ed. 2015 Edition [Livro] / A. Torrieri Don. - New York : Springer, 2015.

Projeto e Prototipação de Sistemas Digitais [Livro] / A. Carro Luigi. - Porto Alegre : Editora da Universidade (UFRGS), 2001.

Recommendation for Cryptographic Key Generation / A. Barker Elaine e Roginsky Allen. - Gaithersburg : National Institute of Standards and Technology, 2012.

Research on Fast 1-D DCT Algorithm Based on Parallel Computing [Conferência] / A. Yu Guang [et al.] // 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE). - Hangzhou : IEEE, 2012. - Vol. 3.

Scilab Help, Signal Processing, DCT/IDCT [Online] / A. Scilab Enterprises // Scilab Help. - Scilab Enterprises, 01 de 04 de 2015. - 27 de 09 de 2016. - https://help.scilab.org/docs/5.5.2/ru_RU/dct.html.

Scilab: Free and Open Source software [Online] / A. Scilab Enterprises. - 2016. - 5.5.2 (64 bit). - <http://www.scilab.org>.

Secure Data Transmission using Encrypted Secret Message [Artigo] / A. Singh Jaishree e Sodhi Dr. J.S. // International Journal of Computer Science and Information Technologies. - Chennai : AIRCC, 2013. - 3 : Vol. 4.

Shift-Register-Based Data Transposition for Cost-Effective Discrete Cosine Transform [Artigo] / A. Hsia Shih-Chang e Wang Szu-Hong // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. - [s.l.] : IEEE, 2007. - 6 : Vol. 15.

Signals, Systems and Inference [Livro] / A. Oppenheim Alan V. e Verghese George C.. - New York City : Pearson, 2015. - Vol. 1.

Sony PlayStation suffers massive data breach [Online] / A. Baker Liana B. e Finkle Jim // Reuters. - Reuters, 26 de Abril de 2011. - 08 de Novembro de 2016. - <http://www.reuters.com/article/us-sony-stoldendata-idUSTRE73P6WB20110426>.

SWIFFT: A Modest Proposal for FFT Hashing [Conferência] / A. Lyubashevsky Vadim, Micciancio Daniele e Peikert Chris // Fast Software Encryption - 15th International Workshop 2008. - Lausanne : Springer, 2008. - Vol. 5086.

T.81 - Information Technology Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Guidelines [Online] / A. International Telecommunication Union // ITU-T. - International Telecommunication Union, 09 de 1992. - 15 de Novembro de 2016. - <http://www.itu.int/rec/recommendation.asp?lang=en&parent=T-REC-T.81-199209-I>.

Telecommunications Engineer's Reference Book [Livro] / A. Mazda Fraidoun. - Londres : Butterworth Heinemann, 1993.

The Scientist & Engineer's Guide to Digital Signal Processing [Livro] / A. Smith Steven W.. - San Diego : California Technical Publishing, 1999. - 2ª.

Understanding Data Eye Diagram Methodology for Analyzing High Speed Digital Signals [Online] / A. On Semiconductor // On Semiconductor. - On Semiconductor, 23 de 06 de 2015. - 13 de Novembro de 2016. - http://www.onsemi.com/pub_link/Collateral/AND9075-D.PDF.

7 Apêndices

Apêndice A: Implementação em *Scilab*

A implementação em *Scilab* do programa para testar a operação de criptografia em camada física por ofuscação reflete, de maneira direta, a mesma orientação do equacionamento definido na seção 3.1. Iniciou-se por declarar e assumir um valor constante para todos os vetores, para fins de inicialização.

Quadro 1. Declaração das variáveis.

```
M      = 1:N
MI     = 1:N
MII    = 1:N
T      = 1:N
TI     = 1:N
TII    = 1:N
TIII   = 1:N
K      = 1:N
Ke     = 1:N
Ka     = 1:N
Kb     = 1:N
Kc     = 1:N
Kd     = 1:N
TIIa   = 1:N
TIIb   = 1:N
TIIc   = 1:N
TIIId  = 1:N
TIIe   = 1:N
TIIIa  = 1:N
TIIIb  = 1:N
TIIIc  = 1:N
TIIId  = 1:N
TIIIe  = 1:N
```

Fonte: o autor, 2016.

Outro ponto a se observar é a modificação necessária na notação: o *Scilab* não aceita o caractere apóstrofe (') no nome dos vetores. O mesmo foi substituído por "I", notação adequada de números romanos.

No Quadro 1, também estão declaradas as chaves de *Ka* a *Ke*, que são chaves aleatórias diferentes de *K*, portanto, são chaves erradas. A partir delas, calcula-se também o equivalente a *T''* (no quadro, *TII*), representados por *TIIa* a *TIIe*. Por fim, usando *TIIa* a *TIIe* obtém-se respectivamente os vetores *TIIIIa* a *TIIIIe*, análogos a *TIII* (no equacionamento *T''*), que são o resultado da aplicação da técnica com uso de chave errada. Todos estes vetores estão novamente definidos na seção 4.4 e 4.6.

Quadro 2. Declaração inicial.

```
N = 8

dct_type = "dct4"

norm_factor = 1 / ( 2 * N )

//gera a chave com 0 e 1 alternado
M = ((-1).^([0:N-1]) + ones(N))/2;

//gera as chaves aleatórias entre 0 e -1
K = ones(N) - 2*grand(1, N, "uin", 0, 1)

Ke = ones(N) - 2*grand(1, N, "uin", 0, 1)

Ka = ones(N) - 2*grand(1, N, "uin", 0, 1)

Kb = ones(N) - 2*grand(1, N, "uin", 0, 1)

Kc = ones(N) - 2*grand(1, N, "uin", 0, 1)

Kd = ones(N) - 2*grand(1, N, "uin", 0, 1)
```

Feita a inicialização, o segundo passo é definir o valor de N , conforme o tamanho da amostra desejada para teste. Este valor pode ser modificado anterior à execução do programa, com outros valores para outros tamanhos de mensagem, para executar, por exemplo, os testes descritos conforme a seção 4.5.

Neste trecho é também declarado o tipo de DCT e também é calculado o fator de normalização, única diferença entre a DCT e a $IDCT$. Gera-se também o vetor de trabalho, que contém zeros e uns alternados, conforme discutido na seção 4.1. Por fim, salienta-se neste trecho o uso da função $grand()$ para as chaves. Neste caso, é gerado um vetor de "1" do qual subtrai-se um vetor com valores entre zero e dois, resultando em um vetor de valores entre um e um negativo.

Em seguida, conforme o Quadro 3, são executadas as operações de ofuscação e desofuscação, conforme estão descritas na seção 3.1, ou seja, as equações (17), (23), (18), (19), (24) e (20), nesta ordem.

Quadro 3. Procedimento de Ofuscação e Desofuscação.

```
//M'  
MI = dct(uf,1,dct_type)  
  
//M''  
MII = (MI . *K)  
  
//T  
MIII = norm_factor*dct(MII,-1,dct_type)  
  
//T'  
T = dct(ef,1,dct_type)  
  
//T''  
TII = TI . *K  
  
//T'''  
TIII = norm_factor*dct(tmp2b,-1,dct_type)
```

Fonte: o autor, 2016.

Já no Quadro 4, está o trecho de código que implementa os testes a serem descritos na seção 4.4 e 4.6, em que se gera os resultados para as chaves erradas. Observa-se, ainda, que as duas primeiras etapas da desofuscação não são afetadas pela chave, portanto, não foram repetidas. Ou seja, aplica-se para cada uma das chaves erradas as equações (24) e, que resultarão em um vetor T''' igualmente errado, conforme explicação anterior.

Quadro 4. Procedimento de tentativa de desofuscação com chave errada

```
//T'' errados: a, b, c, d, e  
TIIa = TI . *Ka  
  
TIIb = TI . *Kb  
  
TIIc = TI . *Kc  
  
TIIId = TI . *Kd  
  
TIIe = TI . *Ke  
  
//T''' errados: a, b, c, d, e  
TIIIa = norm_factor*dct(TIIa,-1,dct_type)  
  
TIIIb = norm_factor*dct(TIIb,-1,dct_type)  
  
TIIIc = norm_factor*dct(TIIc,-1,dct_type)  
  
TIIId = norm_factor*dct(TIIId,-1,dct_type)  
  
TIIIe = norm_factor*dct(TIIe,-1,dct_type)
```

Por fim, conforme apresentado no Quadro 5, para finalizar o programa e armazenar todos os dados gerados, utiliza-se a função `csvWrite()` para posterior leitura do arquivo em softwares de edição de planilha ou aplicativos para a impressão de gráficos, por exemplo.

Quadro 5. Armazenamento dos dados.

```
//Gravação em Planilha de dados, formato CSV  
A = [  
    N;  
    K; Ka; Kb; Kc; Kd; Ke;  
    M; MI; MII; T; TI; TII; TIII;  
    TIIa; TIIb; TIIc; TIIId; TIIe;  
    TIIIa; TIIIb; TIIIc; TIIIId; TIIIe;  
    ];  
csvWrite (A, 'output.csv', ';',',',')
```