CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA
ANÀLISE E DESENVOLVIMENTO DE SISTEMAS

Murilo Fujita

CRIPTOGRAFIA: O DESENVOLVIMENTO DE UMA APLICAÇÃO

Americana, SP
2015

CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Murilo Fujita

CRIPTOGRAFIA: O DESENVOLVIMENTO DE UMA APLICAÇÃO

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Análise e Desenvolvimento de Sistemas, sob a orientação da Prof.(a) Dra. Mariana Godoy Vazquez Miano.

Área de concentração: Matemática da Computação/Sistema de Computação.

Americana, S. P.
2015

Fujita, Murilo

F971c Criptografia: o desenvolvimento de uma aplicação. / Murilo Fujita. – Americana: 2015.

55f.

Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza.

Orientador: Prof. Dr. Mariana Godoy Vazquez Miano

1. Segurança em sistemas de informação I. Miano, Mariana Godoy Vazquez II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.

CDU: 681.518.5

Murilo Fujita

CRIPTOGRAFIA: O DESENVOLVIMENTO DE UMA APLICAÇÃO

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – Fatec/ Americana.
Área de concentração: Matemática da Computação/Sistema de Computação.

Americana, 25 de junho de 2015.

Banca Examinadora:

Mariana Godoy Vazquez Miano (Presidente)
Doutora
FATEC

Clerivaldo José Roccia (Membro)
Mestre
FATEC

Alberto Martins Junior (Membro)
Mestre
FATEC

AGRADECIMENTOS

Realizar este trabalho exigiu um somatório de contribuições importantes. Menciono cada parcela a seguir:

Agradeço à FATEC pela oportunidade de estudar o curso que finalmente encontrei minha realização profissional e que proporcionou oportunidades antes mesmo de concluir a graduação.

Aos professores que contribuíram para minha formação na área de desenvolvimento de sistemas e em especial minha orientadora de Trabalho de Conclusão de Curso, Dra. Mariana, que conduziu esta pesquisa com grande rigor acadêmico.

Aos funcionários da Fatec que são responsáveis por manter o bom funcionamento como a dedicada Leonice Zebiani e demais integrantes da secretaria. Cito também o coordenador de curso, professor Wladimir da Costa.

Aos amigos empenhados em aprendermos juntos que conheci nesta faculdade. Desde o início do semestre tive a boa companhia do Rafael e do Nicholas e mais tarde veio o João para acrescentar conhecimentos e uma boa amizade.

Ao Eduardo Mendes e aos membros do GECA (Grupo de Estudos de Criptografia de Americana). Participar das discussões das tardes de sábado teve uma parcela significativa deste trabalho.

À minha namorada Luciene pelo companheirismo, pelo carinho e seu admirável esforço em conciliar tantas ocupações.

DEDICATÓRIA

Aos meus pais que são os melhores pais do mundo! Sempre investiram na minha educação e me ensinaram a valorizar o conhecimento. Ambos deram total apoio para retomar os estudos.

Meu pai sempre será o exemplo que seguirei por ter me mostrado que devo acreditar com afinco no que se deseja após lutar persistentemente pela vida.

Minha mãe é um modelo de pessoa que só tem boas intenções e agradeço por tudo que ela faz por mim e minha irmã.

RESUMO

A criptografia é uma ciência antiga utilizada em tempos de guerra para enviar mensagens às tropas. A intenção continua sendo produzir textos sem sentido para o interceptador, porém as aplicações voltam-se para comércio eletrônico, validação do conteúdo, autenticação em sistemas entre várias outras. Muitas evoluções desde as primeiras cifras aconteceram principalmente pelo fato de utilizar elementos processadores capazes de realizar as mesmas tarefas várias vezes tornando a forma de misturar o texto mais sofisticada. A parte teórica desta pesquisa busca os recursos que envolvem a criptografia utilizando métodos de substituição, permutação e funções matemáticas. Outro ponto relevante é o estudo dos algoritmos de *hash* que independente se na entrada do sistema é um texto ou arquivo, sempre retorna uma sequência de tamanho fixo. Já a parte prática aborda o desenvolvimento um *software* capaz de embaralhar um texto e recuperá-lo se souber extrair a chave que está oculta em uma imagem, técnica chamada de esteganografia.

Palavras Chave: Criptografia, esteganografia, chave pública.

ABSTRACT

The cryptography is an old science employed in war times to send messages to troops. The propose remains fetch text with no sense for the interceptor, but now is used for e-commerce, content validation, authentication in systems and others. Many evolutions since first encryption happened mainly because processors are able to perform the same tasks many times becoming more sophisticated how mixes the text. The theoretical of this research look for resources that use cryptography applying replacing, permutation methods and mathematics functions. Other point is study hash algorithm that independent if in system entry is a text or file, always return a fixed length string. In practical part there is a software development to encryptation a text and recover it if you know the key hided in an image called steganography.

Keywords: Cryptography, steganography, public key.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1. Introdução

Com o advento da rede mundial de computadores nota-se um crescimento significativo de computadores interconectados. Além dos computadores, a inovação tecnológica permite que celulares, *tablets*, câmeras entre outros dispositivos ampliem as possibilidades de acesso à internet implicando em grande tráfego de dados constantemente. As características para uma troca de informações segura envolvem [1]:

- Cifragem: transformação da mensagem através de operações matemáticas;
- Assinatura digital: a certeza que o destinatário está recebendo um conteúdo enviado pelo remetente;
- Controle de acesso: direito de permitir ou negar determinadas ações;
- Integridade dos dados: ter a confiança de que não houve qualquer modificação do conteúdo digital;
- Troca de informações de autenticação: garantia do reconhecimento das partes envolvidas;
- Preenchimento de tráfego: trata-se de incluir *bits* nas lacunas para confundir tentativas de captura de dados;
- Controle de roteamento: planejamento do caminho que os dados devem seguir com o propósito de evitar rotas suspeitas;
- Certificação: envolvimento de uma terceira entidade confiável para troca de dados.

A internet é uma rede pública e por esta razão, as mensagens podem ser interceptadas por terceiros que não estão autorizados a ter acesso ao conteúdo. Assim, esta proposta discorre sobre os conceitos de criptografia justificados seus estudos em três áreas a seguir:

Acadêmico: empregar os conhecimentos adquiridos através das linguagens de programação e engenharia de *software* para desenvolver um sistema que criptografe e decriptografe.

Social: colaborar de forma ética e profissional com as empresas que necessitam de segurança quando enviam seus dados através da rede pública.

Aplicação: com a compreensão da criptografia, este conhecimento é integrado a qualquer sistema que seja feita a exigência de torná-lo seguro seja através de autenticação ou transmissão de dados.

Uma questão relevante é a segurança, pois documentos como um novo projeto, dados sobre uma pesquisa de mercado, planos financeiros como negócios envolvendo ações ou uma fusão de companhias entre muitas possibilidades jamais podem ser acessados por terceiros

[4].

Em 1994, o *Internet Architecture Board* (IAB) emitiu um relatório intitulado “Security in the internet architecture” estabelecendo o consenso geral de que a Internet precisava de mais e melhor segurança e identificava as principais áreas para mecanismos de segurança. Verificou-se a necessidade de proteger a infraestrutura da rede contra monitoração e controle não autorizados do tráfego da rede e a necessidade de proteger o tráfego de usuário final para usuário final usando mecanismos de autenticação e criptografia.

Ao longo do tempo, os ataques na Internet e em sistemas conectados à Internet se tornaram mais sofisticados, enquanto a habilidade e o conhecimento exigidos para realizar um ataque diminuíram. Os ataques se tornaram mais automatizados e podem causar mais danos.

Esse aumento nos ataques coincide com o aumento no uso da Internet e com aumentos na complexidade dos protocolos, aplicações e da própria Internet. Infraestruturas críticas contam cada vez mais com a Internet para suas operações. Os usuários individuais contam cada vez mais com a segurança da Internet, de *e-mail*, da *Web* e das aplicações baseadas na *Web*. Assim, é preciso que haja uma grande gama de tecnologias e ferramentas para agir contra a ameaça crescente. Em um nível básico, os algoritmos criptográficos para confidencialidade e autenticação assumem maior importância.

As redes de computadores foram usadas principalmente por pesquisadores universitários durante as primeiras décadas de sua existência com a finalidade de enviar mensagens de correio eletrônico e também por funcionários de empresas para compartilhar impressoras. Assim, a segurança nunca precisou de maiores cuidados. No entanto, atualmente como milhões de usuários estão usando as redes para executar operações bancárias, fazer compras e declarar seu patrimônio, a segurança das redes evidencia um problema potencial.

A segurança é um assunto abrangente, pois as variedades de vulnerabilidades são numerosas. Em sua forma mais simples, a segurança se preocupa em garantir que pessoas mal-intencionadas não leiam ou, pior ainda, modifiquem as mensagens enviadas a outros destinatários. Outra preocupação da segurança são as pessoas que tentam ter acesso a serviços remotos que elas não estão autorizadas a usar. Outra função é lidar com meios para saber se uma mensagem supostamente verdadeira é um trote. A segurança trata de situações em que mensagens legítimas são capturadas e reproduzidas, além de lidar com pessoas que tentam negar o fato de terem enviado determinadas mensagens.

A maior parte dos problemas de segurança é causada intencionalmente por pessoas maliciosas que tentam obter algum benefício, chamar a atenção ou prejudicar alguém. A seguir a classificação dos invasores mais comuns e seus objetivos:

- Estudante: divertir-se bisbilhotando as mensagens de correio eletrônico de outras pessoas;
- *Cracker*: testar o sistema de segurança de alguém ou roubar dados;
- Executivo: descobrir a estratégia de *marketing* do concorrente;
- Ex-funcionário: vingar-se por ter sido demitido;
- Contador: desviar dinheiro de uma empresa;
- Corretor de valores: negar uma promessa feita a um cliente através de uma mensagem de correio eletrônico;
- Vigarista: roubar números de cartão de crédito e vendê-los;
- Espião: descobrir segredos militares ou industriais de um inimigo;
- Terrorista: roubar segredos de armas bacteriológicas [5].

Independente se se trata de uma brincadeira de mau gosto ou um crime de alta gravidade, as técnicas de criptografia oferecem segurança pelos dois componentes:

- Uma transformação relacionada à segurança sobre a informação a ser enviada, como a criptografia da mensagem, que embaralha a mensagem de modo que fique ilegível para o oponente e o acréscimo de um código com base no conteúdo da mensagem que pode ser usado para identificar a identidade do emissor.
- Alguma informação secreta compartilhada pelos dois principais e, espera-se, desconhecida pelo oponente. Por exemplo, uma chave de criptografia usada em conjunto com a transformação para embaralhar a mensagem antes da transmissão e desembaralhá-la no recebimento.

Ainda sobre as questões de segurança, é comum o empregar de forma confusa os termos 'segurança' e 'proteção' com o mesmo significado. De um lado estão as questões técnicas, administrativas, legais e políticas enquanto de outro estão os meios específicos para que ofereçam segurança. Especificando as diferenças, segurança é o problema geral e a proteção são os mecanismos para salvaguardar o que deve ser mantido como sigiloso [4].

Uma vez que as informações passaram a ser armazenadas em formato digital, surgiu a necessidade de mantê-las seguras. Esta preocupação torna-se maior quando precisam ser compartilhadas e podem ser acessadas através da internet. Assim, tendo conhecimento que seus dados podem ser violados, muitos usuários preferem que seus dados sofram um tratamento criptográfico.

Há dúvidas sobre a eficiência da segurança e desta forma, buscam-se respostas sobre como introduzir a criptografia. A proteção dos dados é proporcionada por algumas variações

dos algoritmos criptográficos requerendo a compreensão de assuntos como números primos, aritmética modular, corpos finitos além de chaves simétricas e assimétricas. Os números primos envolvem apenas os números inteiros e suas operações de multiplicação e divisão abrangem valores de ordem muito alta para dificultar a fatoração e descobrir quais são os fatores. A aritmética modular também considera apenas os números inteiros e os reduz para um intervalo (como exemplo, validar apenas os números da tabela ASCII[1]). Corpos Finitos são axiomas que corroboram as propriedades matemáticas aplicáveis aos mecanismos de criptografia como comutação, fechamento, elemento identidade, lei distributiva entre outras. Combinando estas propriedades entre dois elementos do conjunto, geram um terceiro. Da mesma forma, as chaves são combinadas ou com o texto plano ou ilegível para embaralhar ou recuperar seu conteúdo original.

Os dois objetos de estudos a abordar são:

- A escolha da linguagem de programação que atenda os critérios para tornar um texto ilegível com a possibilidade recuperá-lo;
- Relacionar segurança e desempenho: analisar o custo computacional sem comprometer a tempo de espera. Em outras palavras, deve-se dominar a cifragem, pois não há serventia se o processo for complexo e não entregar a resposta no tempo certo para o destino [1].

Após a aquisição dos conceitos de criptografia, o objetivo é realizar as operações por meio de instruções computacionais, ou seja, traduzir as sequências empregadas tanto nos algoritmos de cifragem como os de decriptografia para um código-fonte executável por computadores. O desenvolvimento de um *software* reúne os conceitos estudados possibilitando cifrar um texto na origem e recuperá-lo no destino. A teoria é extensa abordando vários mecanismos como troca das posições das letras, substituição de um símbolo por outro entre outras operações matemáticas. Para obter sucesso na criação do programa é preciso planejar qual a sequência para embaralhar a informação e no momento de recuperar o texto, realizar as etapas na ordem inversa.

Para garantir a clareza dos significados, é preciso entender o funcionamento dos elementos envolvidos na ocultação da mensagem. O remetente transforma seu texto em algo incompreensível e transforma a chave em uma imagem. O arquivo contendo um texto sem sentido pode ser transmitido por correio eletrônico ou qualquer outra forma de envio enquanto

[1] American Standard Code for Information Interchange. Todos os textos são codificados entre as inúmeras opções como as variações do ISO, variações próprias do sistema operacional Windows, UTF-8, Unicode entre outras que convertem números em caracteres.

que a figura deve ser armazenada por uma terceira entidade de confiança mútua pelas partes (Verisign, RSA e Entrust, por exemplo, são empresas respeitadas pelos certificados emitidos). O destinatário combina o texto e a imagem recuperando o significado inicial.

Uma vez que os assuntos da pesquisa foram apresentados, este trabalho está organizado como descrito a seguir.

O capítulo 2 descreve os primeiros algoritmos de criptografia resgatando a evolução dos processos de tornar textos ilegíveis.

O capítulo 3 trata da fundamentação teórica reunindo alguns elementos da criptografia, conceitos matemáticos e características da linguagem de programação escolhida para desenvolver o sistema.

O capítulo 4 aborda o desenvolvimento de um *software* que permite pôr em prática os fundamentos dos algoritmos de criptografia. É descrito seu funcionamento, os recursos das telas, condições para realizar a criptografia e o processo reverso. Algumas partes do código-fonte são expostas justificando o funcionamento do processo de embaralhar e recuperar o texto. Também aborda exemplos de aplicações.

E por fim, o capítulo 5 traz as considerações finais apontando resultados sobre os conceitos e conhecimentos adquiridos.

2. PRIMEIRAS CIFRAS

As técnicas de criptografia clássicas conhecidas eram muito simples e faziam uso da substituição e transposição.

2.1. Cifras de Cesar

Julio Cesar é o autor da mais antiga cifra que se tem história e também a mais simples. Seu funcionamento consiste em substituir cada letra e avançar três posições de acordo com a sequência do alfabeto. Como a lista é circular, as últimas letras são substituídas pelas letras iniciais do alfabeto.

Seja a tabela de conversão de valores entre os caracteres abaixo:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

De acordo com a técnica de Cesar, para converter em uma sentença matemática, expressa-se:

$$C = E(e,p) = (p + 3) \bmod 26$$

Para generalizar qualquer quantidade de deslocamentos, altera-se a o valor da variável k descrita abaixo:

$$C = E(e,p) = (p + k) \bmod 26$$

Sendo que k pode assumir qualquer valor entre 1 e 25. Para decriptografar a expressão é:

$$p = D(k, C) = (C - k) \bmod 26$$

A tentativa por força bruta recupera o texto claro, pois é fácil testar as 25 chaves possíveis classificando com uma técnica fraca.

2.2. Cifras monoalfabéticas

A criptografia de Julio Cesar não é segura e se este mecanismo for modificado permutando as letras do alfabeto, existem 4 x 10^{26} chaves possíveis para encontrar usando força bruta. Esta técnica é conhecida como substituição monoalfabética que utiliza para cada mensagem um único alfabeto de cifra (transformando um texto claro em alfabeto cifrado). Pode ser entendido que para cada linha do texto é empregada uma sequência de caracteres como o texto abaixo, por exemplo.

```
UZQSOOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

A análise a ser feita é comparar a frequência de cada letra do texto acima com a de cada letra em um certo idioma. A porcentagem com que os caracteres do texto aparecem é:

P 13,33	H 8,83	F 3,33	B 1,67	C 0,00
Z 11,67	D 5,00	W 3,33	G 1,67	K 0,00
S 8,33	E 5,00	Q 2,50	Y 1,67	L 0,00
U 8,33	V 4,17	T 2,50	I 0,83	N 0,00
O 7,50	X 4,17	A 1,67	J 0,83	R 0,00
M 6,67				

A porcentagem que as letras aparecem na língua inglesa tem a seguinte taxa média:

A 8,167	B 1,492	C 2,782	D 4,253	E 12,702
F 2,228	G 2,015	H 6,094	I 6,996	J 0,153
K 0,772	L 4,025	M 2,406	N 6,749	O 7,507
P 1,929	Q 0,095	R 5,987	S 6,327	T 9,056
U 2,758	V 0,978	W 2,360	X 0,150	Y 1,974
Z 0,074				

Seguindo a linha de raciocínio, as letras mais frequentes do alfabeto são substituídas no texto cifrado:

```
UZQSOOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
 t a          e   e te   a that e e a        a
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
   e t    ta t ha e ee   a e   th     t   a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
 e   e e tat   e    the    t
```

Apenas quatro caracteres foram reconhecidos e a forma de identificar os demais seguem o mesmo princípio de comparação entre as frequências. O texto claro completo é mostrado abaixo:

```
It was disclosed yesterday that several informal but
Direct contacts have been made with political
Representatives of the viet cong in Moscow
```

Esta criptografia é fácil de ser quebrada, pois o mecanismo de tornar o texto legível é analisar a frequência dos caracteres. Uma contramedida é substituir por homófonos (palavra que se pronuncia do mesmo modo, mas com grafia diferente como “coser” e “cozer”).

2.3. Cifra de Hill

O matemático Lester Hill desenvolveu em 1929 uma cifra multiletras utilizando m equações lineares sendo m o número de letras do texto claro. Cada caracter recebe um valor numérico (a = 0, b = 1, ..., z = 25). Como exemplo, se m = 3 o sistema é [17]:

$c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$
$c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$
$c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$

O mesmo sistema pode ser expresso através de matrizes:

$$\begin{pmatrix} c_{11} \\ c_{21} \\ c_{31} \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} . \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \ mod\ 26$$

Ou ainda:

$$C = K.P \bmod 26$$

Sendo que C e P são vetores de colunas de tamanho 3 representando o texto claro e o texto cifrado. A matriz quadrada K tem dimensão 3 representando a chave de criptografa. Os resultados sofrem a operação mod 26.

Seja o exemplo de texto claro “paymoremoney”. Fazendo uso da chave de criptografia

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

e de acordo com a distribuição dos valores de caracteres, as três primeiras letras do texto claro são $\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix}$. Substituindo no sistema:

$$K.\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} mod\ 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = LNS$$

Mantendo a linha de raciocínio, o texto cifrado completo torna-se LNSHDLEWMTRW.

Para decriptografar o texto é preciso da matriz inversa de K denotada por K^{-1} que nem sempre existe, mas quando existe satisfaz a equação. A multiplicação de K por K^{-1} resulta na matriz identidade que composta pelo valor 0 exceto a diagonal principal que tem o valor 1. O inverso de K é:

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

A demonstração é feita abaixo:

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} . \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} mod\ 26 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Ao aplicar K^{-1} ao texto cifrado, então o texto claro é recuperado.

2.4. One-Time Pad

Esta técnica de criptografia consiste em elaborar uma chave aleatória tão extensa quanto o texto claro para que a chave não fosse repetida. Após a operação de criptografia ou decriptografia a chave é descartada exigindo uma nova chave para cada texto. Esta prática produz uma saída aleatória sem relacionamento estatístico com o texto claro. Como o texto cifrado não contém informação sobre o texto claro, não existem meios de quebrar o código.

Por exemplo, o texto cifrado e a chave são exibidos abaixo:

```
Texto cifrado:  ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
Chave:          pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
Texto claro:    mr mustard with the candlestick in the hall

Texto cifrado:  ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
Chave:          mfugpmiydgaxgoufhkllmhsqdqogtewbqfgyovuhwt
Texto claro:    miss scarlet with the knife in the library
```

Com este exemplo pode-se perceber que mesmo um analista conseguisse encontrar as duas chaves e também os dois textos claros, não saberia quais entre as opções é o texto válido.

O próximo capítulo reúne teorias aplicadas à ciência de embaralhar mensagens e recuperar seu conteúdo.

3. Conceitos de criptografia

3.1. Elementos da criptografia

O texto a seguir explica os itens envolvidos na criptografia.

3.1.1. Chaves públicas e privadas

O desenvolvimento da criptografia de chave pública foi uma revolução na história da criptografia. Desde o seu início até os tempos modernos, os sistemas criptográficos utilizavam ferramentas elementares da substituição e permutação. Após milênios de trabalhos com algoritmos que basicamente podiam ser calculados manualmente, um grande avanço na criptografia simétrica ocorreu com o desenvolvimento da máquina de criptografia/decriptografia de rotor. O rotor eletromecânico permitiu o desenvolvimento de sistemas de cifra incrivelmente complexos.

Com a disponibilidade dos computadores, sistemas ainda mais complexos foram criados, sendo que o mais importante foi o Lucifer na IBM que culminou com o *Data Encryption Standard* (DES). Mas tanto as máquinas de rotor quanto o DES, embora representando avanços significativos, ainda contavam com ferramentas básicas de substituição e permutação.

A criptografia de chave pública oferece uma mudança expressiva em relação a tudo o que havia sido feito. Os algoritmos de chave pública são baseados em funções matemáticas em vez de na substituição e permutação. Mais importante, a criptografia de chave pública é assimétrica, envolvendo o uso de duas chaves separadas, diferentemente da criptografia simétrica (utiliza apenas uma chave). O uso de duas chaves tem profundas consequências nas áreas de confidencialidade, distribuição de chaves e autenticação. É relevante saber que os algoritmos simétricos são mais rápidos do que os assimétricos.

O conceito de criptografia de chave pública evoluiu de uma tentativa de atacar dois dos problemas mais difíceis associados à criptografia simétrica: a distribuição de chaves e as assinaturas digitais [10][11].

A distribuição de chaves sob a criptografia simétrica requer:

- que dois comunicantes compartilhem uma chave por algum meio distribuída a eles;
- o uso de um centro de distribuição de chaves (CDC).

Whitfield Diffie, um dos inventores da criptografia de chave pública (juntamente com Martin Hellman, ambos da Stanford University, na época), descobriu que o segundo requisito

anulava a própria essência da criptografia: a capacidade de manter sigilo total sobre sua própria comunicação. Diffie afirmou: “afinal, qual é a vantagem de desenvolver criptossistemas impenetráveis, se seus usuários forem forçados a compartilhar suas chaves com um CDC que pode estar sujeito a roubo ou suborno?”.

O segundo problema sobre o qual Diffie ponderou e que não estava aparentemente relacionado com o primeiro, foi o das “assinaturas digitais”. Se o uso da criptografia tivesse de se tornar comum, não apenas nas situações militares, mas para fins comerciais e particulares, então as mensagens e documentos eletrônicos precisariam do equivalente das assinaturas usadas nos documentos em papel. Ou seja, poderia ser criado um método para garantir, de modo que todas as partes ficassem convencidas, que uma mensagem digital foi enviada por determinada pessoa? Diffie e Hellman obtiveram um avanço em 1976 ao apresentarem um método que resolvia os dois problemas e que era radicalmente diferente de todas as técnicas anteriores de criptografia – a criptografia de chave pública.

A Tabela 1 compara a criptografia convencional e a de chave pública sob aspectos de funcionalidade e segurança:

Tabela 1- Comparação entre criptografia convencional e de chave pública

Criptografia Convencional	**Criptografia de Chave Pública**
Necessário para funcionar: 1. O mesmo algoritmo com a mesma chave é usado para criptografia e decriptografia; 2. O emissor e o receptor precisam compartilhar o algoritmo e a chave.	Necessário para funcionar: 1. Um algoritmo é usado para e criptografia e decriptografia com um par de chaves, uma para criptografia e outra para decriptografia; 2. O emissor e o receptor precisam ter uma das chaves do par casado de chaves (não a mesma chave).
Necessário para a segurança: 1. A chave precisa permanecer secreta; 2. Deverá ser impossível ou pelo menos impraticável decifrar uma mensagem se nenhuma outra informação estiver disponível; 3. O conhecimento do algoritmo mais amostras do texto cifrado precisam ser insuficientes para determinar a chave.	Necessário para a segurança: 1. Uma das duas chaves precisa permanecer secreta; 2. Deverá ser impossível ou pelo menos impraticável decifrar uma mensagem se nenhuma outra informação estiver disponível; 3. O conhecimento do algoritmo, uma das chaves e as amostras do texto cifrado precisam ser insuficientes para determinar outra chave.

A criptografia de chave pública pode ser modelada de seguinte forma: em uma situação na qual Bob tem a intenção de enviar uma mensagem à Alice, primeiramente, Alice gera uma chave pública, que é publicada em seu nome em um diretório público acessível a qualquer usuário, e também gera uma chave privada, a qual é função matemática da chave pública, mas está acessível somente a ela. Para mandar uma mensagem secreta à Alice, Bob procura a chave de Alice no diretório público, Bob então cifra a mensagem usando a chave pública.

O texto cifrado resultante será enviado à Alice através de um canal público, por exemplo, Internet. Finalmente, ao receber o texto cifrado, Alice poderá decifrar e recuperar a mensagem usando sua chave secreta. O mecanismo de cifragem de chave pública é baseado na dificuldade de resolver certos tipos de problemas matemáticos, por exemplo, fatoração de inteiros e logaritmo discreto garantindo que é necessário um tempo muito grande para que se pudesse encontrar a chave privada a partir da chave pública somente.

Até o momento, diversas técnicas de criptografia de chave pública foram desenvolvidas de forma intensa, o que resultou em diferentes níveis de eficiência e segurança. Um dos progressos mais importantes é o relacionado ao tópico de segurança demonstrável: trata-se da segurança que pode ser rigorosamente provada a impossibilidade de um adversário quebrar um criptossistema, enquanto este adversário não for capaz de resolver um problema matematicamente difícil.

Antes, aproximações heurísticas eram constantemente utilizadas para analisar a segurança de esquemas criptográficos, isto é, se ninguém fosse capaz de quebrar o esquema após muitos anos, então sua segurança era amplamente aceita. Contudo, essa abordagem está sendo substituída pela abordagem de segurança demonstrável.

Segurança demonstrável vem se tornando uma condição indispensável para criptossistemas padrão, e assim, a maioria dos criptossistemas atuais atendem a essa condição. Observa-se que uma das maiores vantagens da criptografia de chave pública é o estabelecimento de comunicação segura utilizando apenas dados publicamente acessíveis. Ainda, o número de chaves correspondente a cada parceiro de comunicação que um usuário precisa administrar em esquemas convencionais é reduzido de forma significante ao utilizar criptografia de chave pública e o usuário precisa guardar apenas sua própria chave privada.

À primeira vista, parece que em todos os aspectos a criptografia de chave pública é superior à criptografia de chave simétrica. Contudo, isso não é sempre verdade. Umas das desvantagens da (atual) criptografia de chave pública é sua velocidade: esses tipos decriptossistemas demandam muitos recursos computacionais e o tamanho da mensagem é

significativamente limitado.

A melhor solução é, portanto, uma combinação das vantagens da criptografia convencional e da criptografia de chave pública. Isto é, não é prático cifrar a mensagem inteira usando apenas criptografia de chave pública. É mais sensato usar criptografia de chave pública para cifrar a chave de uma sessão de trabalho, de natureza temporária, a qual então será utilizada como uma chave simétrica comum entre o remetente e o destinatário, enquanto eles estabelecem a sessão de trabalho.

Os sistemas por chave secreta são eficientes, pois a quantidade de computação necessária para criptografar ou decriptar uma mensagem é controlável, porém há uma grande desvantagem: o emissor e o receptor devem possuir a chave secreta compartilhada. Podem até mesmo obtê-la fisicamente, um dando-a ao outro. Para contornar esse problema, é usada a criptografia por chave pública segundo Diffie e Hellman apud STALLINGS, 1976. Esse sistema apresenta a seguinte propriedade: chaves distintas são usadas para criptografia e decriptação e dada uma chave criptográfica bem conhecida, é praticamente impossível descobrir a chave correspondente de decriptação. Sob essas circunstâncias, a chave criptográfica pode ser pública e somente a chave de decriptação privada é mantida em segredo.

Para ter uma noção da criptografia por chave pública, considere estas duas questões:

Questão 1: Quanto é 374159265358979 x 3141592658979?

Questão 2: Qual é a raiz quadrada de 391257150641938709059482850841?

Para responder a responder à questão “1” estima-se uma ou duas horas. Já a questão “2” não dispensa o uso de uma calculadora, um computador ou alguma ajuda externa. Embora as operações de elevar ao quadrado e de extração da raiz quadrada sejam opostas, suas complexidades computacionais são distintas. Esse tipo de assimetria forma a base da criptografia por chave pública. A criptografia utiliza a operação fácil, mas a decriptação sem a chave requer a realização da operação difícil.

Um sistema de chave pública, chamado RSA, explora o fato de a multiplicação de grandes números serem muito mais fácil para um computador que a fatoração de grandes números, especialmente quando toda a aritmética é implementada com base na aritmética de módulo e todos os números envolvidos têm centenas de dígitos (Rivest et aL apud STALLINGS, 1978). Esse sistema é amplamente usado no mundo criptográfico, assim como sistemas baseados em logaritmos discretos (El Gamal apud STALLINGS, 1985). O principal problema da criptografia por chave pública é que ela é milhares de vezes mais lenta que a criptografia simétrica.

A criptografia por chave pública funciona com todos escolhendo um par de chaves (pública, privada) e tornando pública a chave pública. A chave pública é a chave criptográfica; a chave privada é a de decriptação. Em geral, a criação da chave é automatizada, possivelmente com uma senha escolhida pelo usuário alimentada em um algoritmo como uma semente. Para enviar uma mensagem secreta para um usuário, o emissor criptografa a mensagem usando a chave pública do receptor. Como somente o receptor tem a chave privada, apenas ele pode decriptar a mensagem. Estas operações podem ser descritas da seguinte forma: seja E a função de criptografia e D a função de decriptografia. Assim, são verdadeiras as sentenças:

$$E(D(x)) = x \text{ e } D(E(x)) = x$$

ou seja, não importa a ordem de aplicação, as funções são comutativas e recuperam o valor inicial.

Para que somente os envolvidos consigam ler a mensagem, a questão é como transmitir a chave pública exclusivamente para o destinatário. Alguns publicam suas chaves públicas nas páginas da *web*, mas não é um procedimento recomendado porque um invasor pode alterar secretamente seu conteúdo. A forma alternativa consiste em apresentar um certificado assinado digitalmente por um terceiro que seja de confiança de ambos [4]. A mesma prática é utilizada para distribuir a chave privada, porém este tem um nome específico: PKG (*Private Key Generator* – Gerador de chave privada) [12].

3.1.2. Números primos

Os números primos envolvem o conjunto dos números inteiros, ou seja, os números negativos e positivos desconsiderando os números fracionários. Um inteiro “b” divide outro inteiro “a” se existe um terceiro número inteiro “c” tal que a = bc. Neste caso, também dizemos que “b” é um divisor ou fator de “a”, ou ainda que a é múltiplo de “b”. Todas estas expressões significam a mesma coisa. Quando $1 < b < a$, dizemos que b é um fator ou divisor próprio de a. Naturalmente só há dois divisores que não são próprios, 1 e o próprio “a”. O número “c”, na definição acima é chamado de cofator de “b” em “a”. Por exemplo, 5 divide 20 porque 20 = 5 x 4. Neste exemplo, 4 é o cofator de 5 em 20. Na prática, determina-se que “b” divide “a” efetuando a divisão e verificando que o resto é zero. O cofator é o quociente da

divisão. O primeiro resultado é uma lista das propriedades dos múltiplos.

Dois inteiros quaisquer sempre têm pelo menos 1 como fator comum; afinal, um divide qualquer inteiro. Se 1 for o único fator comum a dois números, é dito que não têm fator próprio comum ou que são primos entre si. Note que um par de números primos distintos não têm fator próprio comum. No entanto, há muitos números compostos sem fator próprio comum, como é o caso de 6 e 35, por exemplo.

Para decompor inteiros em primos, é conveniente recordar a definição de número primo. Um número inteiro "p" é primo se "p" = ±1 e os únicos divisores de "p" são ±1 e ±p. Portanto 2, 3, 5 e −7 são primos, mas 45 = 5 x 9 não é primo. Um número inteiro, diferente de ±1, que não é primo é chamado de composto. Logo 45 é composto [6].

3.1.3. Aritmética modular

Para compreender esta definição, é preciso entender a periodicidade dos fenômenos, ou seja, a frequência de repetição. Como exemplos cita-se a semana que a duração é de sete dias, um jogo de tabuleiro com uma quantidade de posições que retorna para o início e o relógio. Assim, o valor do módulo é o número da repetição (no caso da semana, o módulo é 7). Assim, se n é o módulo e a e b são números inteiros, então é dito que “a” é congruente a “b” módulo n se a − b é um múltiplo de n.

Se dois inteiros “a” e “b” são congruentes módulo “n”, se expressa:

$$a \equiv b \pmod{n};$$

se não são congruentes, afirma-se

$$a \not\equiv b \pmod{n}.$$

Como exemplos,

$3 \equiv 8 \pmod{5}$, ao passo que 3 não congruente 8 (mod 7).

Por outro lado,

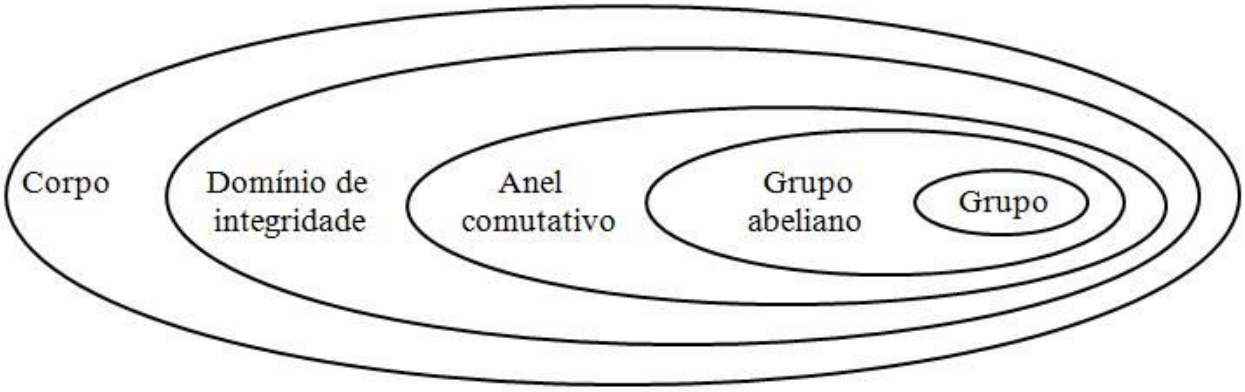
$3 \equiv -25 \pmod{7}$, embora 3 não congruente −25 (mod 5).

Em outras palavras, a aritmética modular é uma manipulação de inteiros que reduz todos os números a um em um conjunto fixo [0 … n-1] para um número n. Qualquer inteiro fora desta faixa é reduzido a um nessa faixa, tomando-se o resto após a divisão por n.

3.1.4. Corpos finitos

Corpos finitos reúnem conceitos de conjuntos dispostos da seguinte forma: “Grupo” é o conjunto elementar que está contido pelo “Grupo abeliano”. Este, por sua vez, está contido no “Anel comutativo” que está dentro do “Domínio de integridade” e por fim, pelo “Corpo” como exibido na Figura 1 [13].

Figura 1- Conjuntos do grupo, anel e corpo



Grupos, anéis e corpos são os elementos fundamentais da álgebra abstrata ou álgebra moderna. Os conjuntos dos elementos podem operar algebricamente, ou seja, dois elementos do conjunto podem ser combinados de várias maneiras para obter um terceiro elemento do conjunto. Essas operações estão sujeitas a regras específicas que definem a natureza do conjunto. Por convenção, a notação para as duas classes principais de operação sobre elementos do conjunto normalmente é a mesma notação para adição e multiplicação em números comuns. Porém, é importante observar que a álgebra abstrata não se limita a operações aritméticas comuns.

Grupos

Um grupo G, às vezes indicado por {G, ●}, é um conjunto de elementos com uma operação binária, indicada por ● que é um operador genérico e pode se referir a adição, a multiplicação ou a alguma operação matemática. Os seguintes axiomas são obedecidos para cada par ordenado (a,b) de elementos em G:

(A1) Fechamento: Se a e b pertencem a G, então a ● b também está em G.

(A2) Associativo: a ● (b ● c) = (a ● b) ● c para todo a, b, c em G.

(A3) Elemento identidade: Existe um elemento e em G de modo que a ● e = e ● a para todo a em G.

(A4) Elemento inverso: Para cada a em G existe um elemento a' em G de modo que a ● a' = a' ● a = e.

Se um grupo tem um número finito de elementos, diz-se um grupo finito e a ordem do grupo é igual ao número de elementos no grupo. Caso contrário, o grupo é um grupo infinito.

Um grupo é considerado abeliano se satisfizer à seguinte condição adicional:

(A5) Comutativo: a ● b = b ● a para todo a,b em G.

Quando a operação em grupo for a adição, o elemento de identidade é 0. O elemento inverso de "a" é "-a" definindo que a regra de subtração é dada por: a – b = a + (– b).

Um grupo cíclico é definido pela exponenciação dentro de um grupo como aplicação repetida do operador de grupos de modo que a^3 = a ● a ● a. Além disso, define-se a^0 = e, o elemento de identidade e $a^{-n} = (a')^n$. Um grupo G é cíclico se cada elemento de G for uma potência a^k (k é um inteiro) de um elemento fixo a ∈ G. O elemento a gera o grupo G ou que é gerado de G. Um grupo cíclico é sempre abeliano e pode ser finito ou infinito.

O grupo aditivo de inteiros é um grupo cíclico infinito gerado pelo elemento 1. Neste caso, as potências são interpretadas aditivamente, de modo que n é a n-ésima potência de 1.

Anéis

Um anel R, às vezes indicado por {R, +, ×}, é um conjunto de elementos com duas operações binárias chamadas adição e multiplicação de forma que para a, b, c em R, os seguintes axiomas são obedecidos:

(A1-A5) R é um grupo abeliano com relação à adição, ou seja, R satisfaz os axiomas de A1 a A5. Para o caso de um grupo aditivo, indicamos o elemento de identidade como zero e o inverso de a como –a.

(M1) Fechamento sob multiplicação: Se a e b pertencem a R, então ab também está em R.

(M2) Associatividade da multiplicação: a(bc) = (ab)c para todo a,b,c em R.

(M3) Leis distributivas: a (b + c) = ab + ac para todo a, b, c em R.
(a + b) c = ac + ab para todo a, b, c em R.

Basicamente, um anel é um conjunto que realiza adição, subtração [a – b = a + (–b)] e multiplicação sem sair do conjunto.

Com relação à adição e à multiplicação, o conjunto de todas as matrizes quadradas em n sobre os números reais é um anel.

Um anel é considerado comutativo e satisfizer a seguinte condição adicional:

(M4) Comutatividade da multiplicação: ab = ba para todo a, b em R.

Define-se domínio de integridade o anel comutativo que obedece o seguinte axioma:

(M5) Identidade multiplicativa: Existe um elemento 1 em R de modo que a1 = 1a = a para todo a em R.

(M6) Sem divisores zero: Se a, b em R e ab =0, então a = 0 ou b = 0.

Considere que S seja o conjunto de inteiros, positivos, negativos e 0 sob as operações normais de adição e multiplicação. S é um domínio de integridade.

Corpos

Um corpo F, às vezes indicado por {F, +, ×}, é um conjunto de elementos com duas operações binárias chamadas de adição e multiplicação, de modo que para todo a, b, c em F, os seguintes axiomas são obedecidos:

(A1-M6) F é um domínio de integridade, ou seja, F satisfaz os axiomas de A1 a A5 e de M1 a M6.

(M7) Inverso multiplicativo: Para cada a em F, exceto 0, existe um elemento a^{-1} em F tal que $aa^{-1} = (a^{-1})a = 1$

Basicamente, um corpo é um conjunto que realiza adição, multiplicação e divisão sem sair do conjunto. A divisão é definida pela regra $a/b = a(b)^{-1}$. Exemplos conhecidos de corpos são os números racionais e os números complexos. O conjunto de todos os inteiros não é um corpo, pois nem todo elemento do conjunto tem um inverso multiplicativo. As exceções são

os elementos “1” e “-1” que possuem inversos multiplicativos nos inteiros.

Um corpo finito é um corpo com um número finito de elementos.

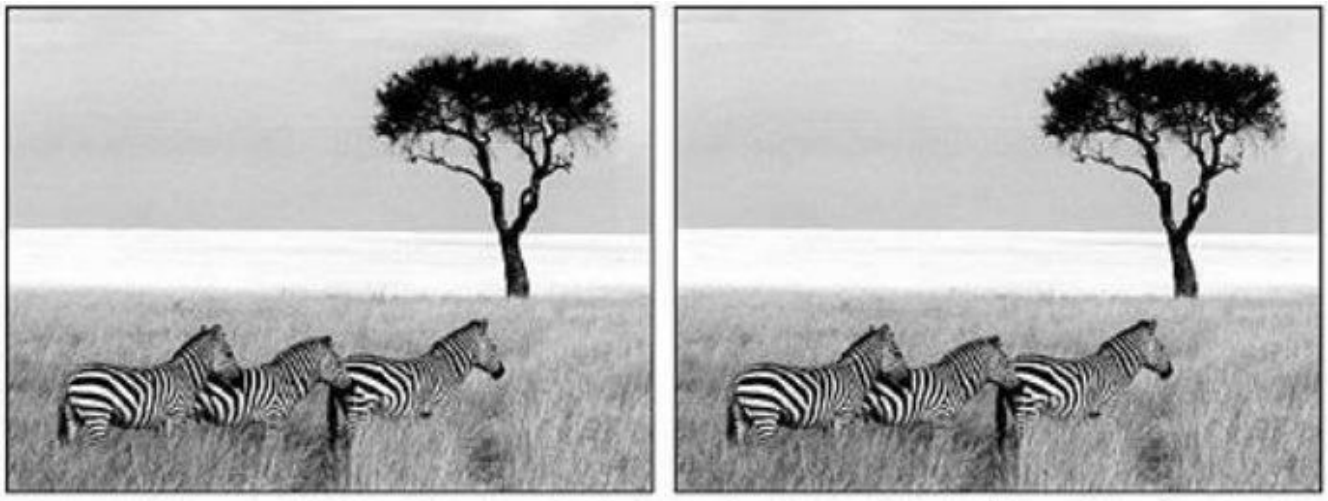
3.1.5. Esteganografia

Os métodos de criptografia têm o propósito de tornar um texto ilegível para não descobrir seu conteúdo original. O objetivo da esteganografia é negar a existência de uma mensagem secreta. A palavra tem origem grega sendo: “estegano” significa esconder ou cobrir e “grafia”, escrita. Esta técnica esconde um texto ou uma imagem em um arquivo digital (por exemplo, uma fotografia). Em outras palavras, é a arte de esconder informações para que não suspeitem como decriptar ou investigar o arquivo. O texto ocultado conserva as impressões como se a fotografia não sofresse alguma alteração evitando despertar a atenção. A criptografia não é a melhor solução para comunicações seguras e sim, parte da solução. Desta forma há um bom desempenho da esteganografia combinada com a criptografia.

Uma aplicação é desta técnica é a de conseguir enviar mensagens que não sejam barradas pelos governantes em países onde a censura é rígida. A criptografia permite que mensagens secretas sejam enviadas (embora talvez isso não seja legal), pois é trabalho em vão se for exigido uma licença do governo para exportar o conteúdo.

Através da internet é possível enviar mensagens ocultas como no exemplo da Figura 2.

Figura 2- (a) Imagem original. (b) Imagem modificada contendo textos.



O item (a) é uma foto tirada no Quênia contendo três zebras contemplando uma árvore. O item (b) parece ter exatamente as mesmas três zebras e a árvore mas, além disso, há uma atração a mais. A segunda fotografia contém o texto completo de cinco peças de Shakespeare incorporado a ela: Hamlet, Rei Lear, Macbeth, O Mercador de Veneza e Júlio César. Juntas, essas peças totalizam mais de 700 KB de texto. A imagem em cores original tem 1024x768 pixels. Cada pixel consiste em três números de 8 bits, cada um representando a intensidade de uma das cores, vermelha, verde e azul, desse *pixel*. A cor do pixel é formada pela superposição linear das três cores. O método de codificação esteganográfico utiliza o LSB (*Less Significant Bit* – Bit Menos Significativo) de cada valor de cor RGB como um canal oculto. Desse modo, cada pixel tem espaço para três bits de informações secretas, um no valor vermelho, um no valor verde e um no valor azul. Com uma imagem desse tamanho, podem ser armazenados até 1024x768x3 *bits* ou 294.912 *bytes* de informações secretas. A fotografia original (a) tem 24 *bits* de cores enquanto que a versão (b) tem 21 *bits* e a diferença é imperceptível.

Técnicas recentes adotam dois domínios de transformada sendo a DCT (Transformada de coseno discreta) e a DWT (Transformada Wavelet discreta). O LSB pode ser aplicado a este domínio e depois substituído o LSB dos coeficientes transformados com a imagem escondida. Uma transformada inversa é aplicada aos coeficientes que contém a mensagem para reconstruir a imagem original. É um método mais robusto a ataques [24].

Esteganografia baseada em DCT

Quando o LSB de um dos coeficientes DCT é alterado, seu efeito é disperso aos *pixels* no espaço reconstruído. Em outras palavras, esta mudança é localmente aplicada à frequência de domínio, mas quando a DCT inversa é usada, todos os *pixels* são contaminados, mas esta pequena mudança não é percebida pelos olhos humanos.

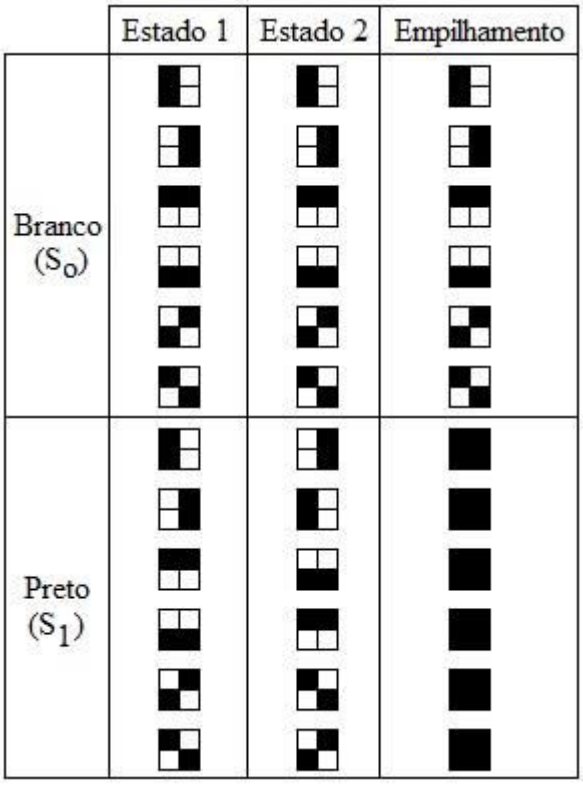
Esteganografia baseada em DWT

A informação é substituída pelo LSB do coeficiente *wavelet* para cobrir a imagem. A vantagem desta aproximação é que a informação armazenada nos coeficientes *wavelet* a mudança nas intensidades da reconstrução da imagem será mais imperceptível do que a DCT.

A criptografia visual de Naor e Shamir propõe o esquema (2,2) sendo dois estados e

duas cores. A imagem secreta é dividida em duas que contém pontos brancos e pretos aleatórios. Para decriptar a imagem, as duas imagens são empilhadas resultando em um quadro completamente preenchido de cor preta ou não. Este esquema consiste de duas matrizes S_0 e S_1 chamadas de matrizes básicas associando o valor 0 para branco e 1 para preto. A Figura 3 mostra a combinação dos dois estados.

Figura 3- Resultado ao empilhar as matrizes básicas



Uma outra aplicação é permitir que os proprietários de imagens codifiquem mensagens secretas nas suas imagens, declarando seus direitos de propriedade. Se tal imagem for roubada e colocada em um *web site*, o dono legal poderá revelar a mensagem esteganográfica no tribunal para provar a quem pertence a imagem. Essa técnica é conhecida como marca d'água.

3.1.6. A operação lógica XOR

A operação XOR (também chamada de OU Exclusivo) é simbolizada por $\oplus$ e possui as seguintes aplicações [16]:

- Apontar quando dois bits são diferentes;
- Ser usado como uma lógica de negação;

- Utilizado em funções de criptografia.

Como citado anteriormente, um texto é representado sob várias codificações e cada caractere possui seu valor binário. Então somente após esta conversão é que possível aplicar as operações XOR entre os valores 0 (zero) e 1 (um). A Tabela 2 exibe os resultados de acordo com as entradas:

Tabela 2- Tabela XOR

Entrada 1	Entrada 2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Para elucidar seu funcionamento *bit* a *bit*, sejam as palavras “FATEC” e “ADS15”. Seus valores decimais de acordo com a tabela ASCII são: F (70), A (65), T(84), E (69), C (67), D (68), S (83), 1 (49) e 5 (83). Convertendo para binário e realizando a operação XOR:

```
Texto Claro  FATEC   01000110  01000001   01010100  01000101  01000011
XOR          ADS15   01000001  01000100   01010011  00110001  00110101
Resultado            00000111  00000101   00000111  01110100  01110110
```

Na linha “Resultado”, os valores decimais são respectivamente: 7, 5, 7, 116 e 118. O valor 116 corresponde a letra “t” (minúscula) e 118 equivale a “v” (minúsculo). Porém, o valor 7 tem o valor *bell* (campainha) e 5 equivale a *enquiry* (caractere de controle que é disparado para confirmar o fim da transmissão). Estes dois valores são caracteres não-imprimíveis e se forem ignorados ao realizar as operações de decriptografia não retornam ao texto original.

O próximo item descreve as características da linguagem de programação escolhida para desenvolver uma aplicação desta pesquisa.

3.2. Linguagem de programação

Para atender às diversas operações de criptografia, a opção requer uma linguagem de

programação orientada a objetos que permita a reutilização do código, a organização das funcionalidades distribuídas em arquivos e deve possuir bibliotecas com funções que auxiliem no desenvolvimento. Diante dos requisitos levantados, a escolha é a linguagem C# (lê-se C *sharp*), plataforma Microsoft .NET executável nas variações do sistema Windows [20]. São características da plataforma .NET: tem forte conceitos das linguagens C e C++ combinando a produtividade do Visual Basic com o poder bruto do C++. A licença é *freeware* (gratuita), o projeto está ativo e a versão atual é a 5.0.

Todas as variáveis do programa devem ser declaradas no escopo da classe com exceção das *structs* e enumerações. A tipagem segura define tipos primitivos:

- lógico incompatível com inteiros;
- inteiros com variações de tamanhos (8, 16, 32 e 64 bits considerando ou não o sinal);
- ponto flututante com variações de 4 e 8 bits;
- *strings* e *char* que armazenam caracteres Unicode (16 bits por caractere);
- decimal com 28 dígitos evitando erros de arredondamento (128 bits para representar o número).

Também é relevante abordar os seguintes aspectos da linguagem:

- Os objetos e *arrays* (vetores) são alocados dinamicamente no *heap* através do operador *new*. Em outras palavras, não basta fazer a declaração para manipulá-los. É preciso utilizar o comando *new* para que efetivamente exista na memória do computador e realize o processamento;
- O índice dos *arrays* inicia com zero e durante a compilação é feita uma verificação do intervalo;
- O *cast* (conversões de tipos) é realizado em tempo de exceção. Esta sintaxe pode ser implícita ou explícita;
- Da mesma forma com na linguagem C++ é utilizado “.”, “::” e “->” para as operações de referências, no C# usa-se “.”;
- Além dos laços originários do C (*for*, *while*, *do..while*), existe o *foreach* que percorre todos os elementos de um *array* ou *collection* (coleção);
- O comando switch permite processar casos mutuamente exclusivos sendo que ao fim de cada opção é obrigatório o uso da palavra reservada *break*;
- O tratamento de erros é o *exception* e é único mecanismo;
- Ao invés de macros, existe a compilação condicional (#ifdef). As diretivas de pré-processamento podem ser usadas para incluir ou excluir uma parte do código-fonte,

ajudam a sinalizar em linha encontra-se o erro ou gerar um relatório com erros fatais e/ou alertas. As diretivas são: #if, #else, #elif, #endif, #define, #undef, #warning, #error, # line, #region e #endregion [21];

- Os *Templates* não são suportados, porém utiliza-se *refletions* (responsável por descrever os *assemblies*, módulos e tipos) que trabalha com metadados derivados da classe *System.Attribute* [22];
- Há suporte de sobrecarga de funções e de operadores;
- Admite a herança simples sendo o ancestral comum a todos os objetos é chamado de *System.Object* que tem as propriedades de criar, comparar, converter *strings* e informações em tempo de execução. No lugar de herança múltipla, as classes podem implementar várias *interfaces* que funciona como uma classe abstrata contendo os protótipos de métodos sem um corpo de instruções;
- Como todos os objetos têm como ancestral a classe *Object*, permite a propriedade chamada *Boxing* e *Unboxing* que converte qualquer informação em objeto podendo recuperá-la no seu formato original. Em outras palavras, um método encapsula, por exemplo, um tipo *string* ou *double* em um objeto (*boxing*) e retorna ao seu formato inicial (*unboxing*) facilitando a compatibilidade de assinatura dos métodos.
- A palavra reservada *virtual* especifica uma reimplementação de um método virtual, ou seja, sobreescreve um método existente;
- A palavra reservada *delegate* é um ponteiro para o método que contém o endereço de uma função. Os *delegates* permitem que uma classe chame métodos em outras sem exigir que esta outra classe seja derivada de uma classe ancestral conhecido.

Para o desenvolvimento foi utilizado o IDE (*Integrated Development Enviroment* – Ambiente Integrado de Programação) Visual Studio 2010 que suporta Visual Basic, Visual C++ e as linguagens de scripting VBScript e Jscript). Todas as linguagens citadas fornecem acesso à plataforma Microsoft .NET que possuem um motor comum de execução e biblioteca de classes completas chamado CLS (*Common Language Subset* – Subconjunto de Linguagem Comum) compatibilizando as linguagens e as bibliotecas[2].

3.3. Algoritmo de Hash

Hash é uma função de mão única que ao aplicar seus algoritmos não é possível retornar ao original. A Figura 4 mostra a mensagem m que é a entrada para a função de *hash* H e que retorna o valor do *hash* h (também conhecido como *digest*).

Figura 4- Processo unidirecional da função de Hash

Matematicamente: h = H(m).

Seu propósito é criar uma impressão digital da informação que associa uma cadeia alfanumérica ao texto, imagem, formato multimídia ou qualquer tipo de arquivo. Assim, quando o destinatário recebe o conteúdo que foi transmitido pela rede pública, deve conferir com se mesma sequência alfanumérica combina com a do remetente para garantir que não houve modificações. Todas as vezes que executar este algoritmo para o mesmo conteúdo, este deve resultar nos mesmos caracteres [7].

Os algoritmos mais comuns são:

- SHA-1 que retorna um comprimento fixo de 20 bytes (160 bits);
- MD5 que apresenta uma sequência fixa de 16 bytes (128 bits).

As exigências para uma função *hash* devem satisfazer algumas condições para não enfraquecer a segurança dos algoritmos de criptografia no qual elas são usadas. Será necessário então que uma função *hash* h satisfaça algumas propriedades para prevenir falsificações [14]:

1. Se x é uma mensagem, deve ser computacionalmente inviável encontrar uma mensagem $x' \neq x$ tal que $h(x') = h(x)$.
2. Deve ser computacionalmente inviável encontrar duas mensagens x e x' tais que $x \neq x'$ e $h(x') = h(x)$.
3. Dado algo de tamanho menor e fixo z, deve ser computacionalmente inviável encontrar uma mensagem x tal que h(x) = z.

Existem diversas funções *hash*, algumas atualmente não são mais seguras (não satisfazem as três propriedades acima), como a função *hash* MD4 (mais rápido do que o MD5, porém menos segura) [18]. Salienta-se que funções *hash* que satisfazem as três propriedades acima citadas podem não mais satisfazer tais propriedades no futuro, pois possíveis ataques podem ser descobertos, sendo necessárias melhorias no algoritmo ou até mesmo um algoritmo totalmente novo.

Outra característica é a resistência às colisões. Se, ao calcular o *hash* de duas mensagens diferentes com uma determinada função de *hash*, os valores de *hash* resultantes forem iguais, resulta em colisão. Uma boa função criptográfica de *hash* deve ser resistente a

essas colisões. Por exemplo, sejam duas mensagens diferentes m_a e m_b, então [15]:

O *hash* da mensagem m_a seria: $h_a = H(m_a)$

O *hash* da mensagem m_b seria: $h_b = H(m_b)$

Se h_a e h_b forem iguais evidencia uma colisão.

Para provar que uma colisão é possível, analisa-se entrada e saída da função de *hash*. Como a entrada é uma mensagem com qualquer formato e qualquer tamanho, considera-se que o número de entradas possíveis é infinito (∞). Com relação a saída, se a função de *hash* for de 128 *bits*, o número de saídas possíveis é 2^{128}. Já para uma função de *hash* de 160 bits, o número de saídas possíveis seriam 2^{160}. São números expressivos, mas finitos que podem resultar em uma colisão.

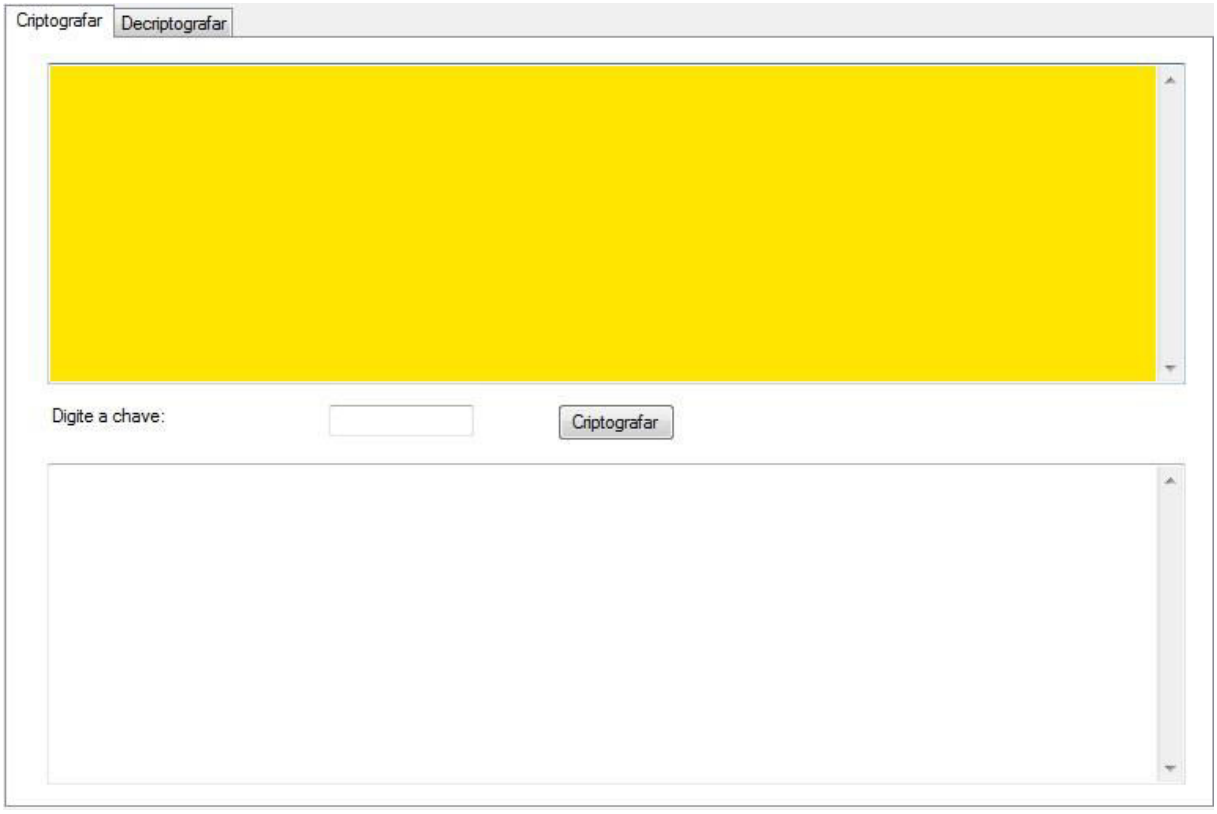
Após os estudos dos conceitos essenciais para compreensão do que é pertinente ao tema de criptografia, o próximo capítulo descreve as etapas do *software* criado especificamente para esta pesquisa.

4. Desenvolvimento do software

Este capítulo descreve o desenvolvimento do *software* que cumpre com os objetivos propostos. Os métodos empregados envolvem substituição de caracteres, utilização de chave pública e várias sequências de operações lógicas Ou Exclusivo. Além disso, a obtenção da chave pública emprega a esteganografia que dificulta conhecer seu valor já que é um texto ocultado em uma imagem. O algoritmo de *hash* pode contribuir com a garantia de que a mensagem não foi alterada durante seu trajeto, porém, o algoritmo não foi utilizado devido ao fato de que os envolvidos têm em confiança uma terceira entidade, que dispensa a verificação.

A Figura 5 mostra a tela inicial assim que o programa é executado e composto de duas abas: a função de criptografar, ou seja, transformar um texto plano em um formato ilegível e a função decriptografar, que realiza a operação inversa, a de tornar o texto compreensível novamente. As etapas devem ser seguidas da seguinte forma: o usuário digita seu texto podendo conter acentos, mudança de linha e demais símbolos na parte superior (fundo amarelo). Em seguida escolhe uma senha que é a chave criptografadora com extensão entre quatro e oito caracteres. O programa emite um aviso caso esta exigência não seja obedecida.

Figura 5- Funções da operação de criptografia



No momento em que o botão “Criptografar” é pressionado, duas tarefas serão realizadas:

- Cifragem:
    - Composta de operações de transformar caracteres em números segundo a tabela ASCII;
    - Substituição de números por outro;
    - Converter da base decimal para binária;
    - Realizar as operações lógicas XOR diversas vezes entre os números binários e a chave criptografadora;
    - Transformar o novo número binário em caracteres.

Processo de esteganografia:

- Esconder uma sequência de caracteres dentro de uma imagem. O valor da chave é convertido em números da base dez segundo a tabela ASCII. Em seguida mudam para o sistema binário transformam-se nos pixels de cor branca se o valor for “0” e preto para “1”. Para dificultar em que região encontra-se o texto equivalente da chave, os 64 *bits* são transformados em uma matriz quadrada de ordem 8 ao invés de se disporem linearmente. A localização da chave é sempre na mesma posição e o restante da imagem é gerado aleatoriamente até que atinja uma dimensão quadrada 256 como a Figura 7.

As operações de cifragem e esteganografia realizadas pelo *software* são mostradas na Figura 6.

Figura 6- Transformação do texto claro em cifrado

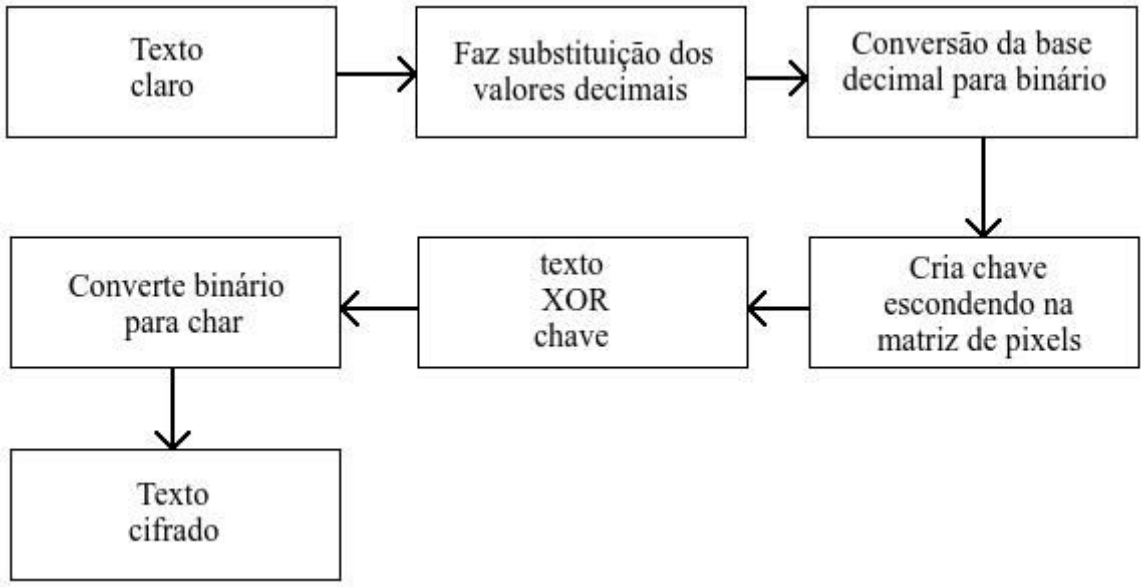


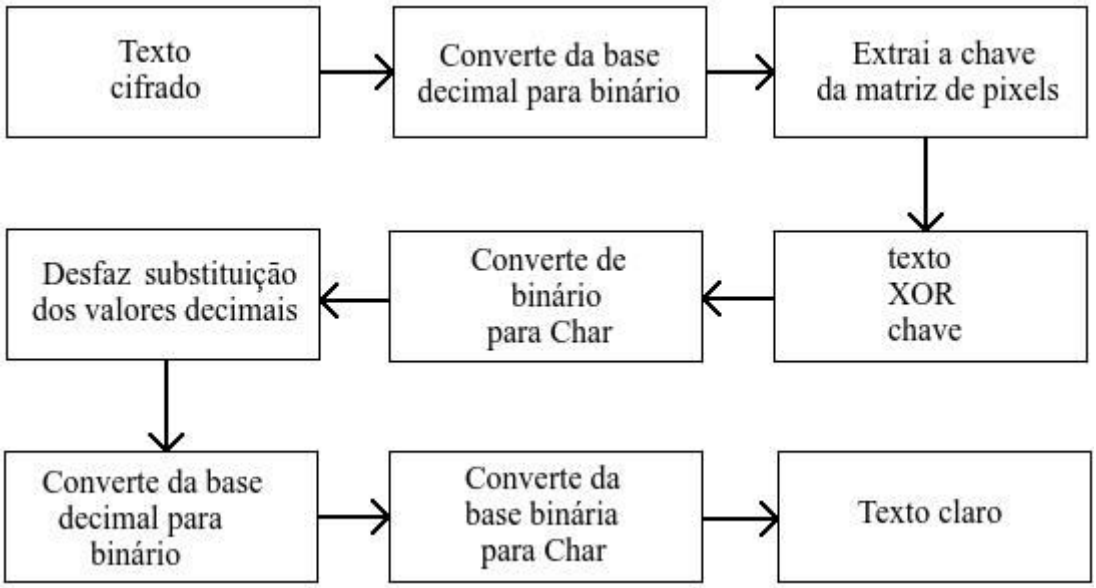
Figura 7- Matriz de pixel contendo a chave de criptografia (detalhe ampliado 3 vezes)

A criação da chave dentro da imagem é realizada pelo método codificado a seguir:

```
public bool ConverteTextoEmPixels()
{
	int i, x, y, posicao = 0, tamanho;
	string cadeia = "";
	bool verifica = false;

	byte[] ansi = Unicode2ANSI(chave);

	for (i = 0; i < ansi.Length; i++)
	{
		cadeia += Convert.ToString(ansi[i], 2).PadLeft(8, '0');
	}

	tamanho = cadeia.Length; // Completa com 0 até obter 64 bits
	for (i = tamanho + 1; i <= 64; i++)
	{
		cadeia += "0";
	}

	for (y = 40; y < 48; y++)
	{
		for (x = 150; x < 158; x++)
		{
			string zeroOuUm = cadeia.Substring(posicao, 1);
			posicao++;
			if (zeroOuUm.Equals("0"))
				imagem.SetPixel(x, y, Color.White);
			else
				imagem.SetPixel(x, y, Color.Black);
		}
	}

	imagem.Save("esteganografia.bmp");
	if (File.Exists("esteganografia.bmp"))
		verifica = true;

	return verifica;
}
```

Para recuperar o texto original, o processo de decriptografia é mostrado na Figura 8.

Figura 8- Transformação do texto cifrado para o conteúdo original



A Figura 9 mostra a tela com a aba "Decriptografar" selecionada para o destinatário recuperar o sentido original quando está de posse do arquivo contendo texto criptografado e a imagem com a palavra estenografada. Para aumentar a proteção das palavras originais, o botão está desabilitado.

O texto recebido deve ser copiado para a área de transferência e colado na parte superior (fundo amarelo). Em seguida, o usuário pressiona o botão "Selecionar arquivo" e é exibida uma API[2] do Windows para localizar o arquivo de nome "estenografia.bmp" na caixa de diálogo. Somente se a designação do arquivo for "estenografia.bmp" o botão habilita. Por fim, a senha a ser digitada no local ao lado da inscrição "Chave decriptadora" deve ser igual ao conteúdo ocultado na imagem.

Com o campo da chave preenchido e o botão habilitado, as operações reversas são disparadas resultando em duas situações possíveis:

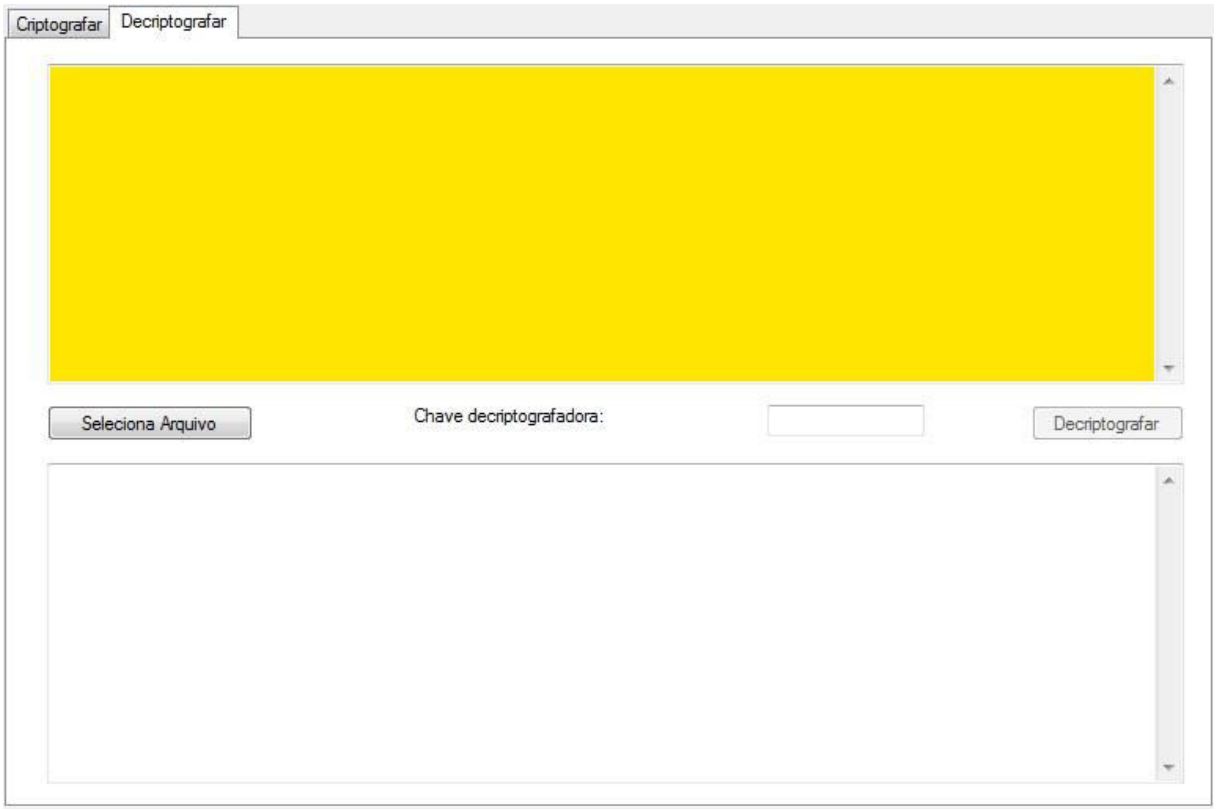
1. Se a chave for a correta, o texto com sentido inicial é exibido na outra caixa de texto;
2. Se a chave for incorreta, as combinações das operações lógicas XOR, a permutação e substituição resultarão em caracteres que continuarão incompreensíveis.

Aplicar a operação XOR entre sequências de binários várias vezes torna o algoritmo mais forte. Por esta razão, este método é executado dezesseis vezes reforçando o

[2] *Application Programming* Interface – Interface de Programação de Aplicativos é um conjunto de recursos disponível pelo sistema operacional bastando apenas fazer a sua chamada.

embaralhamento do texto.

Figura 9 - Tela com a função de recuperar o texto inicial



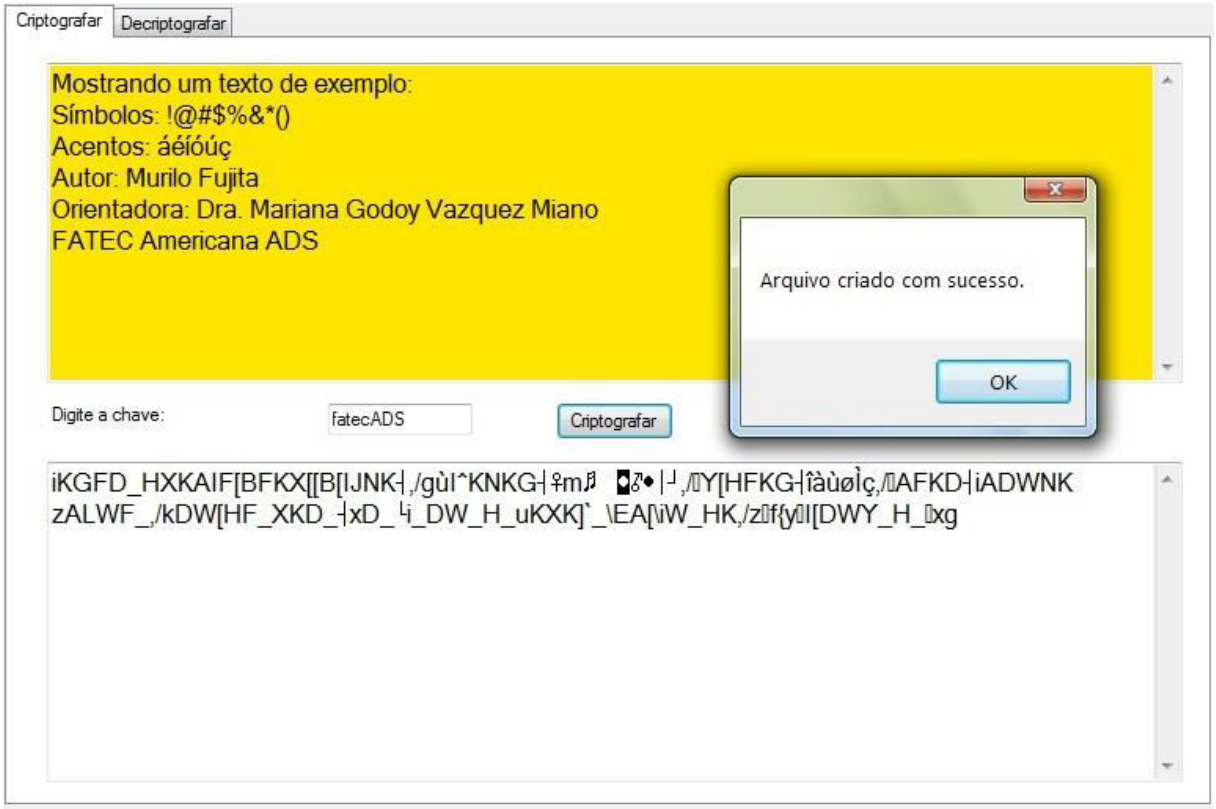
A Figura 10 mostra um texto com várias linhas, letras acentuadas e símbolos. Após digitar uma chave com extensão entre quatro e oito caracteres e pressionar o botão "Criptografar", o texto é exibido na caixa de texto inferior e uma caixa de diálogo informa que o arquivo foi criado referindo-se ao arquivo "esteganografia.bmp" que se encontra no mesmo diretório no qual está o *software*. A chave utilizada foi "fatecADS".

O trecho do código-fonte que combina os bits da chave com os bits de cada caractere do texto é mostrado abaixo:

```
public string PermutaComChave(string binario, string chave)
{
       int i, pos8 = 0;
       char[] bits = new char[8];
       string textoPermutado = "";


       string[] binarioArrayString = new string[binario.Length / 8]; // declaração do
vetor com valores binários contendo 8 dígitos
```

```
        for (i = 0; i < binario.Length / 8; i++)
        {
                binarioArrayString[i] = binario.Substring(pos8, 8);
                pos8 += 8; // de 8 em 8 caracteres para cada índice do array conter um
valor binário


                for (int j = 0; j < 8; j++)
                {
                        bits = binarioArrayString[i].ToCharArray();
                        textoPermutado += LogicaXOR(Convert.ToChar(bits[j]), chave[j]);
                }
        }

        return textoPermutado;
}
```

Figura 10- Exemplo de um texto a ser criptografado



Um determinado método da classe pode ser chamado diversas vezes realizando repetidamente a mesma tarefa. Então, para aumentar o sigilo da informação, a combinação entre os bits que representam os caracteres e a os bits da chave sofrem 16 rodadas.

Supondo que este conteúdo foi transmitido para outra pessoa que tem o mesmo *software* e conhecimento dos procedimentos a executar, o conteúdo do arquivo é colado na

caixa de texto situada na parte superior, indicada a localização do arquivo "esteganografia.bmp", digitada a chave correta ("fatecADS") e o resultado é visto na Figura 11.

Figura 11- Exemplo do texto recuperado

Para recuperar o texto original, o método que combina *bit* a *bit* entre a chave e o texto criptografado deve executar a mesma quantidade que tornou o texto ilegível, ou seja, 16 vezes.

Mesmo que um terceiro tente decifrar a mensagem conhecendo os procedimentos, mas errando o valor da chave decriptografadora (a tentativa desta vez foi a chave "software"), o resultado é exibido na Figura 12.

Figura 12- Resultado da decriptografia ao errar a chave

4.1. Aplicações

Nos primeiros sistemas de computação existentes, as senhas dos usuários eram armazenadas de forma clara, não cifrada, em um arquivo de senhas. A identificação era simples: o usuário digitava seu nome de usuário e senha, o sistema então verificava a identidade do usuário comparando a senha digitada com a senha armazenada no arquivo. A proteção das senhas ficava por conta do sistema operacional fazendo que somente alguns usuários e programas pudessem acessar o arquivo. Este tipo de proteção não era eficiente porque algumas falhas de segurança dos programas que acessavam o arquivo de senhas eram exploradas comprometendo as senhas. Para solucionar este problema, o sistema de controle de acesso foi modificado. As senhas não são mais armazenadas de forma clara, mas sim o valor *hash* das senhas. A identificação do usuário continua simples: é comparado o valor *hash* da senha digitada com o valor *hash* armazenado.

Com o aumento de usuários conectados e a expansão da internet que possibilitou criar a área de *e-commerce* (comércio eletrônico), percebeu-se a necessidade de garantir a

segurança das transações para conquistar a confiança dos compradores. Dados sigilosos como a sequência numérica do cartão de crédito, prazo de validade, código de verificação do cartão e senha devem ser transmitidos de forma incompreensível se forem capturados por terceiros já que esta situação é possível porque a internet é uma rede pública [19].

Existem empresas envolvidas em grandes projetos que não se sentem seguras enviando seus documentos através da internet. Quando a matriz e sua filial estão geograficamente distantes, preferem deslocar um profissional de confiança para transportar tais dados gerando um alto custo e uma espera significativa.

Exemplos como os que foram descritos são resolvidos através da criptografia que tornam seus conteúdos compreensíveis apenas para quem conhece o segredo do sistema. Qualquer situação que se queria fazer uso de segurança como criar uma partição criptografada no disco rígido manterão os dados protegidos.

A utilização de *hashing* gera números pseudoaleatórios para os protocolos criptográficos. Devido ao seu comportamento aleatório, as funções de *hashing* são úteis na construção de funções geradoras de números pseudoaleatórios. Outra aplicação é o controle de acesso que utiliza *hashing* unidirecional permitindo que as senhas sejam armazenadas de maneira segura, protegidas até dos administradores do sistema e possibilitem uma verificação rápida.

A criptografia é essencial para geração de certificados possibilitando:

- Privacidade: a entidade autenticadora não fica conhecendo o conteúdo do documento;
- Confiança distribuída: a entidade autenticadora não consegue modificar a data e hora que um certificado foi feito.

4.2. Quando a criptografia não resolve

Uma assinatura em uma carta, uma impressão digital em um documento, um lacre de cera em um envelope, um cofre de aço cheio de cédulas também terão que ser substituídos por seus equivalentes eletrônicos. A criptografia moderna tem resposta para a maioria destes desafios. Entretanto, a cada dia surgem novos desafios que também precisam ser respondidos.

Organizações com grande poder de processamento já conseguiram quebrar o DES tradicional. Mesmo que um adversário não tenha muito poder de processamento é possível quebrar a segurança do controle de acesso utilizando ataques de dicionário. O ataque de dicionário consiste em armazenar em um arquivo (chamado de dicionário) as palavras mais

prováveis de serem utilizadas como senhas: a própria palavra senha, nomes próprios, atletas, artistas, nomes de cidades, etc. Com o dicionário pronto, basta combinar um algoritmo disponível na internet e compará-lo com o arquivo de senhas. Existem vários programas que automatizam este processo, como o Crack, CrackerJack e Qcrack [19].

A criptografia é um reforço no fator de segurança para os carros de luxo, mas há publicações que mostram como burlar este sistema. Segundo o autor, um professor universitário de ciências da computação da *University of Birmingham* (Inglaterra), sua intenção não é apoiar quadrilhas especializadas a usar as determinadas ferramentas para roubar carros e sim conscientizar o público sobre a realidade. O responsável analisou o *chip* que troca informações entre o carro e a chave através de um microscópio e deduziu o funcionamento (o processo custa cerca de £ 50.000) [23].

A utilização da criptografia é empregada em várias situações quando a confidencialidade é uma necessidade, mas há outros fatores relevantes. Não se pode ignorar o fato que os profissionais cumprem com as suas tarefas enquanto estão em harmonia com a empresa, mas as suas intenções mudam quando desavenças acontecem. A partir deste ponto, ou deixam de executar corretamente suas obrigações ou criam formas de vulnerabilidade nos sistemas. Infelizmente, nem mesmo o melhor algoritmo de criptografia pode manter os dados completamente a salvo.

Esta é uma questão que deve ser discutida pelas políticas das empresas buscando acompanhar o desempenho do funcionário, monitorar os recursos dos sistemas, conhecer o nível de privilégio e acesso para realizar determinada tarefa entre outros cuidados. A criptografia assegura que o conteúdo seja interpretado somente a quem interessa, mas é preciso que as pessoas comprometidas nas operações não se corrompam.

O último capítulo descreve as considerações finais após levantar os pontos relevantes da teoria e a elaboração do software de criptografia.

5. Considerações finais

Desenvolver um sistema capaz de tornar um texto sem sentido é fácil, porém recuperá-lo é o desafio da criptografia. Requer um planejamento detalhado das operações para criar condições de reverter o processo tornando legível novamente.

Unindo os conceitos e a criação do *software* teve como resultado satisfatório alcançando os propósitos da pesquisa: a exibição de um texto incompreensível e sua forma recuperada.

O desenvolvimento do programa permitiu estudar a força dos algoritmos: aos poucos novas funcionalidades foram sendo introduzidas e pôde-se perceber o distanciamento do texto original. Conclui-se que apesar de ser um protótipo, alguns fatores contribuem para manter o conteúdo inteligível:

1. O atacante desconhece a existência do arquivo que contém uma matriz de *pixels*. Caso tenha uma cópia do arquivo "esteganografia.bmp" não saberá como extrair a chave da imagem;
2. Se a mensagem for capturada, este não saberá quais as operações envolvidas para reverter o passo original.
3. O formato como o texto é apresentado difere da quantidade de parágrafos entre o texto original e o criptografado impedindo de analisar possíveis cabeçalhos, extensão do corpo de texto ou qualquer outro indício que suspeite seu caractere correspondente.
4. O tamanho do texto criptografado é o mesmo do original, mas devido aos caracteres não imprimíveis, torna o resultado visualmente diferente contribuindo para dificultar qualquer associação entre o conteúdo original e o embaralhado.
5. A programação orientada a objeto permite executar tarefas apenas fazendo a chamada do método. Portanto, para reforçar a forma como a criptografia torna um texto ilegível, os conceitos de programação são fundamentais para executar certas tarefas várias vezes. Simplificando, o atacante não tem conhecimento da sequência e nem da frequência do processo de criptografia (por exemplo, a quantidade de operações XOR).

Se a criptografia dependesse exclusivamente dos algoritmos e os códigos fossem publicados, por exemplo, em alguma página da internet, passaria a ser inútil. Um programador com bons conhecimentos poderia fazer o programa de acordo com as instruções encontradas e desvendar qualquer texto. Por esta razão, a força da chave para realizar os dois processos (embaralhar ou tornar legível) é vital para a segurança. A ideia é a mesma para um ladrão que roubou centenas de milhares de chaves e agora tenta roubar uma casa: é preciso testar as

chaves até encontrar a que funciona com o segredo da fechadura.

6. Referências bibliográficas

[1] STALLINGS, William. **Criptografia e segurança de redes**. Princípios e práticas. São Paulo: 4ª ed., Person, 2007.

[2] MICROSOFT Corporation®, C#: **Segredos da linguagem**. Trad. Kátia Roque. Rio de Janeiro: Campus, 2001. 403p.

[3] COMER, Douglas E. **Redes de Computadores e Internet**. Abrange transmissão de dados, ligação inter-redes e web. Porto Alegre: 2ª ed., Bookman, 2007.

[4] TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. São Paulo: 2ª ed., Person. 1995.

[5] TANENBAUM, Andrew S. **Redes de computadores**, São Paulo: 5 ª ed., 2011. p. 543.

[6] COUTINHO, S. C. **Números inteiros e criptografia RSA**. 2ª ed , IMPA - Instituto De Matemática Pura E Aplicada. 2014.

[7] COBB, Chey. **Cryptography For Dummies**. Indiana. Wiley Publishing, Inc. 2004.

[10] FONTES, E. L. G. **Praticando a segurança da informação**. Rio de Janeiro, Brasport, 2008.

[11] LYRA, M. R. **Segurança e Auditoria em Sistemas de Informação**. Rio de Janeiro, Edit. Ciência Moderna, 2008.

[12] ANAND, Darpan; KHEMCHANDANI, Vineeta; SHARMA, Rajendra K. **Identity-Based Cryptography Techniques and Applications (A Review)**. 2013. Disponível em <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6658013&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D6658013>. Acessado em 03/03/2015.

[13] SILVA, R. A. da. **Análise de seleção de parâmetros em criptografia baseada em curvas elípticas**. 2006. 114 f. Dissertação (Mestrado em Ciência da Computação). Universidade Estadual de Campinas. 2006.

[14] Encontro de Iniciação Científica e Pós-Graduação do ITA, 12., 2006, São José dos Campos. Anais. Instituto Tecnológico de Aeronáutica. Disponível em < http://www.bibl.ita.br/xiiencita/FUND%203.pdf>. Acessado em 03/06/2015.

[15] SERAFIM, Vinicius da Silveira. **Introdução à Criptografia: Funções Criptográficas de Hash**. Disponível em <http://www.serafim.eti.br/academia/recursos/Roteiro_08-Funcoes_de_ Hash.pdf>. Acessado em 03/06/2015.

[16] GOUVÊA, C. P. L. **Implementação em software de criptografia baseada em emparelhamentos para redes de sensores usando o microcontrolador**. 2010. 117f.

Dissertação (Mestrado em Ciência da Computação). Universidade Estadual de Campinas. 2010.

[17] BARBOSA, L. D. A. **Cifras de Hill** – Uma aplicação ao estudo de matrizes. 2015. 17f. Dissertação (Mestrado Profissional em Matemática). Universidade Federal de São João Del Rei. 2015.

[18] MIRANDA, R. A. **Criptossistemas baseados em curvas elípticas**. 2002. 98f. Dissertação (Mestrado em Ciência da Computação). Universidade Estadual de Campinas. 2002.

[19] SENDIN, I. da S. **Funções de hashing criptográficas**. 1999. 97 f. Dissertação (Mestrado em Ciência da Computação). Universidade Estadual de Campinas. 1999.

[20] SANT´ANNA, Mauro. **C#, uma linguagem para o novo milênio**. Disponível em <https:// msdn.microsoft.com/pt-br/library/cc518016.aspx>. Acessado em 04/04/2015.

[22] ELIAS, Marcio. **Diretivas de pré-processamento em C#**. Disponível em <https:// msdn.microsoft.com/pt-br/library/cc564864.aspx>. Acessado em 11/04/2015.

[22] Biblioteca MSDN. **Reflection (C# and Visual Basic)**. Disponível em <https://msdn.microsoft.com/pt-br/library/ms173183.aspx>. Acessado em 11/04/2015.

[23] The guardian. Scientist banned from revealing codes used to start luxury cars. Disponível em <http://www.theguardian.com/technology/2013/jul/26/scientist-banned-revealing-codes-cars>. Acessado em 07/06/2015.

[24] MOSTAGHIM, Melika; BOOSTANI, Reza. CVC: Chaotic Visual Cryptography to Enhance Stenography. 2014. Disponível em <http://ieeexplore.ieee.org/xpl/ articleDetails.jsp?arnumber=6994020&sortType%3Dasc_p_Sequence%26filter%3DAND%28p_IS_Number%3A6994006%29>. Acessado em 29/07/2015.

APÊNDICE

A tabela a seguir são os valores ASCII que foram referências para o trabalho de criptografia. Fonte: http://www.ascii-code.com/

Dec	Descrição
0	Null char
1	Start of Heading
2	Start of Text
3	End of Text
4	End of Transmission
5	Enquiry
6	Acknowledgment
7	Bell
8	Back Space
9	Horizontal Tab
10	Line Feed
11	Vertical Tab
12	Form Feed
13	Carriage Return
14	Shift Out / X-On
15	Shift In / X-Off
16	Data Line Escape
17	Device Control 1 (oft. XON)
18	Device Control 2
19	Device Control 3 (oft. XOFF)
20	Device Control 4
21	Negative Acknowledgement
22	Synchronous Idle
23	End of Transmit Block
24	Cancel
25	End of Medium
26	Substitute
27	Escape
28	File Separator
29	Group Separator
30	Record Separator
31	Unit Separator
32	(Space)
33	! (Exclamation mark)
34	“ (Double quotes)
35	#
36	$
37	%

Dec	Descrição
38	&
39	‘ (Single quote)
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K

Dec	Descrição
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[
92	\
93	]
94	^ (Caret – circumflex)
95	_ (Underscore)
96	` (Grave accent)
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w

Dec	Descrição
120	x
121	y
122	z
123	{
124	|
125	}
126	~
127	(Delete)
128	€
129	
130	‚
131	*ƒ*
132	„
133	…
134	†
135	‡
136	ˆ
137	‰
138	Š
139	‹
140	Œ
141	
142	Ž
143	
144	
145	‘
146	’
147	“
148	”
149	•
150	–
151	—
152	˜
153	™
154	š
155	›
156	œ
157	
158	ž
159	Ÿ
160	
161	¡
162	¢
163	£

Dec	Descrição
164	¤
165	¥
166	¦
167	§
168	¨
169	©
170	ª
171	«
172	¬
173	
174	®
175	¯
176	°
177	±
178	²
179	³
180	´
181	µ
182	¶
183	·
184	¸
185	¹
186	º
187	»
188	¼
189	½
190	¾
191	¿
192	À
193	Á
194	Â
195	Ã
196	Ä
197	Å
198	Æ
199	Ç
200	È
201	É
202	Ê
203	Ë
204	Ì
205	Í
206	Î
207	Ï

Dec	Descrição
208	Ð
209	Ñ
210	Ò
211	Ó
212	Ô
213	Õ
214	Ö
215	×
216	Ø
217	Ù
218	Ú
219	Û
220	Ü
221	Ý
222	Þ
223	ß
224	à
225	á
226	â
227	ã
228	ä
229	å
230	æ
231	ç
232	è
233	é
234	ê
235	ë
236	ì
237	í
238	î
239	ï
240	ð
241	ñ
242	ò
243	ó
244	ô
245	õ
246	ö
247	÷
248	ø
249	ù
250	ú
251	û

Dec	Descrição
252	ü
253	ý

Dec	Descrição
254	þ
255	ÿ