CENTRO UNIVERSITÁRIO NILTON LINS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

AMBIENTES COLABORATIVOS: PROPOSTA PARA TROCA SEGURA DE INFORMAÇÕES

MANAUS
2005

GLEDSON BESSA ROSA

AMBIENTES COLABORATIVOS: PROPOSTA PARA TROCA SEGURA DE INFORMAÇÕES

Monografia apresentada à disciplina Projeto Final II do Curso de Sistemas de Informação, em cumprimento às exigências para a obtenção do grau de Bacharel em Sistemas de Informação. Prof.-Orientador: Tiago Eugenio de Melo, MSc.-Co-orientadora: Rosana Noronha Gemaque, MSc.

MANAUS
2005

Resumo

Este trabalho apresenta uma proposta que busca suprir a carência do protocolo SMTP em certificar-se de que as mensagens eletrônicas são realmente provenientes da fonte da qual afirmam ser. O desenvolvimento e a implementação de um mecanismo adicional ao MTA baseado em assinatura digital e banco de dados que é proposto aqui provê este recurso necessário à crescente demanda por segurança e autenticidade nas mensagens eletrônicas.

Palavras-chave: Certificação de Emissão, Email, E-Mail, Criptografia, Assinatura Digital, Banco de Dados, Segurança da Informação, Correio Eletrônico.

Abstract

This work presents a proposal that looks for supply the lack of the SMTP protocol in certifying that the electronic messages are really coming of the source where they affirm to be. The development and implementation of an additional mechanism to the MTA based on digital signature and relational database that it is proposed here can provide this necessary resource to the crescent demand for safety and authenticity in electronic messages.

Keywords: Emission Certification, Email, E-mail, Cryptography, Digital Signature, Databases, Information Security, Electronic mail.

Sumário

Capítulo 1

Introdução

Com a popularização da *Internet* nos últimos anos, as pessoas passaram a explorar, cada vez mais, os recursos que a rede oferece. Acesso à conta bancária, compra e venda de forma eletrônica, acesso à informações corporativas, cadastro e informações comerciais são atividades que passaram a fazer parte da rotina das pessoas.

A facilidade na realização de tarefas que antes poderiam demorar horas em agências bancárias e lojas passaram a significar um conforto para as pessoas. Problemas como trânsito, filas e despesas com combustível podem ser evitadas com o uso das facilidades mencionadas providas pela *Internet*.

Devido ao uso crescente destas facilidades, que em grande parte significam transações financeiras, têm surgido problemas de segurança na mesma proporção. É possível acompanhar em noticiários e em revistas de vários segmentos da sociedade, não só da área de informática, que ataques e perdas financeiras crescem em proporções que podem ser consideradas críticas.

A formação e a descoberta de vários grupos especializados na prática de crimes baseados na exploração de falhas de arquiteturas e sistemas, com uma grande atividade no ambiente de correspondência eletrônica, demonstra a necessidade de atenção voltada para segurança deste recurso.

É possível constatar ainda que grande parte dos usuários da *Internet* possuem conhecimentos extremamente limitados, o que facilita a ação de criminosos, além disso, poucas são as pessoas que se preocupam com a configuração de seu ambiente de trabalho.

1.1 Objetivos

1.1.1 Geral

Desenvolver de um mecanismo de autenticação em que a origem das informações que chegarem ao servidor de correio eletrônico possa ser verificada, seguindo os conceitos de *software* livre apresentados pela *GNU/GPL*.

1.1.2 Específicos

- Estudar técnicas de segurança no envio e recepção de emails.

- Desenvolver e implementar um mecanismo de autenticação para correio eletrônico baseado em assinatura criptográfica.

- Implementar e avaliar o algoritmo criptográfico *MD5* no processo de resumo de mensagens eletrônicas.

- Avaliar o uso de ferramentas de *software* livre na implementação de seu caso real.

1.2 Motivação

Diferente do cenário onde as redes de computadores foram inicialmente projetadas, o real ambiente onde elas operam se tornou extremamente selvagem, principalmente devido à enorme integração provido pelos atuais meios de comunicação.

Segundo [TAN03], sob as condições nas quais o ambiente de correio eletrônico foi concebido, segurança não demandava tanta atenção. Mas agora, que milhões de cidadãos comuns usam as redes para transações bancárias, compras e pagamento de impostos, a segurança das redes se apresenta como um problema massivo em potencial.

Problemas ligados à maneira como as mensagens trafegam pela rede chamam a atenção de vários especialistas, de acordo com [JOH00], o protocolo *SMTP* é carente de recursos no que diz respeito à autenticação de envio, restando para este apenas os recursos de criptografia como *PGP* (Pretty Good Privacy) e assinaturas digitais para trazer maior

autenticidade ao seu conteúdo.

Segundo [CdE04], o número de ataques reportados mensalmente em 2004 era significativo quando levamos em consideração sua classe. Relacionado como "fraude"na tabela abaixo, temos no terceiro trimestre de 2004 um número expressivo de ataques reportados desta natureza.

Incidentes Reportados ao CERT.br -- Julho a Setembro de 2004

Tabela: Totais Mensais e Trimestral Classificados por Tipo de Ataque.

Mês	Total	worm (%)		af (%)		dos (%)		invasão (%)		aw (%)		scan (%)		fraude (%)	
jul	**6773**	4636	68	18	0	2	0	7	0	49	0	1791	26	270	3
ago	**5910**	3221	54	26	0	5	0	22	0	71	1	2194	37	371	6
set	**5167**	2997	58	56	1	4	0	13	0	52	1	1704	32	341	6
Total	**17850**	10854	60	100	0	11	0	42	0	172	0	5689	31	982	5

Figura 1.1. *Ataques registrados pelo CERT.br no Terceiro Trimestre de 2004.* [CdE04]

Porém, o mais importante segue em nova avaliação. Ainda segundo [CdE05], o número de ataques reportados sobre o mesmo item teve um aumento superior a dez vezes quando consideramos o mês de julho de 2004 em relação ao mês de abril de 2005. Tudo isso em menos de um ano.

Abaixo temos a avaliação do segundo trimestre de 2005.

Incidentes Reportados ao CERT.br -- Abril a Junho de 2005

Tabela: Totais Mensais e Trimestral Classificados por Tipo de Ataque.

Mês	Total	worm (%)		af (%)		dos (%)		invasão (%)		aw (%)		scan (%)		fraude (%)	
abr	**5253**	1432	27	17	0	0	0	20	0	25	0	1437	27	2322	44
mai	**6883**	2175	31	4	0	2	0	34	0	22	0	1489	21	3157	45
jun	**5406**	1510	27	0	0	5	0	17	0	55	1	1356	25	2463	45
Total	**17542**	5117	29	21	0	7	0	71	0	102	0	4282	24	7942	45

Figura 1.2. *Ataques registrados pelo CERT.br no Segundo Trimestre de 2005.* [CdE05]

Segurança é um tópico muito extenso, principalmente se for levado em consideração a grande variedade de problemas que poderão estar relacionados ao tema.

Nesta proposta, serão abordados os problemas de segurança relacionados à transmissão e recepção de e-mail, as mensagens de propaganda não solicitadas, os vírus, os hoaxes, os scams e a atual ausência de mecanismos padronizados para a autenticação do emissor da mensagem vinculados ao protocolo *SMTP*.

1.3 Metodologia

Será realizado um levantamento bibliográfico com o objetivo de basear esta proposta em obras que abordem adequadamente o tema.

Será estudado o algoritmo *MD5*, usado para resumo criptográfico e demais métodos ligados ao armazenamento e consulta dos resumos.

Será definido e configurado o *software* inicial que mais se adeque aos objetivos da proposta a partir de sua conformidade com os conceitos de licenciamento de software livre (GNU/GPL) e didátca de seu código fonte.

Será desenvolvido e implementado o mecanismo de autenticação deste trabalho empregando o software selecionado.

Será realizada a validação dos objetivos da proposta em relação à implementação de acordo com os objetivos deste trabalho.

1.4 Organização do trabalho

No capítulo 2, é apresentada a base teórica referente aos termos, técnicas e tecnologias usadas neste trabalho e no capítulo 3 é apresentada a estrutura do trabalho contendo o projeto em si, ferramentas escolhidas para a implementação, decisões de projeto e linguagem de programação.

No capítulo 4 são apresentados os resultados obtidos com a implementação do projeto e e no capítulo 5 são apresentadas as conclusões que puderam ser obtidas com o trabalho

e propostas para projetos futuros.

Capítulo 2

Conceitos

Criptografia de chave simétrica, armazenamento e consulta de resumos criptográficos em bancos de dados relacionais e conversa eletrônica entre servidores *SMTP* são os pilares desta pesquisa, sendo abordados detalhadamente neste e em todos os outros capítulos.

Apesar de relacionar aspectos da engenharia social nesta pesquisa, a mesma não tem por objetivo estudar ou detalhar exaustivamente tal assunto, tomando como foco sobre o tema apenas a falsidade ideológica praticada pelos indivíduos que tentam se passar por outras pessoas ou entidades.

2.1 Segurança da informação

O tema "Segurança da Informação"é extremamente abrangente, cobrindo desde a avaliação do conteúdo que deve ser objeto de proteção até os fatores ambientais, como a rede de relacionamentos do indivíduo e o conhecimento externo que as pessoas têm a seu respeito.

2.2 Engenharia social

O termo "Engenharia Social"abrange diretamente o assunto referente aos fatores externos que, superficialmente, significa o estudo dos meios pelo quais um indivíduo pode ser capaz de induzir outro a agir de determinada maneira.

2.3 Criptologia

A criptologia é a ciência das mensagens secretas. É composta pela criptografia e criptanálise. A história, estudo e desenvolvimento das técnicas de ocultação das mensagens são baseados nesta ciência.

2.3.1 Criptologia na história

Devido ao grande número de registros referentes a diversas técnicas usadas para ocultação de mensagens na história, tornou-se mais significante a exposição destas técnicas com o passar do tempo de que uma aproximação da possível história da criptologia.

Em muitos trabalhos científicos, como o de *Jaime André Sulzbach*, podemos encontrar afirmações de que a Criptografia é tão antiga quanto a própria escrita, o que ajuda a representar com maior clareza a complexidade da linha histórica deste assunto.[SUL03]

Para representar cronologicamente passagens históricas relevantes, serão relacionados 5 períodos e neles apresentados as passagens históricas marcantes no âmbito da criptologia.

2.3.1.1 História antiga - Em aproximadamente 1900 A.C., é possível identificar a primeira ocorrência histórica do emprego de técnicas para ocultação de mensagens.

Em uma vila egípcia perto do rio Nilo chamada Menet Khufu. Khnumhotep II, arquiteto do faraó Amenemhet II, construiu monumentos para o faraó os quais foram documentados em tabletes de argila substituindo algumas palavras ou trechos de texto com o intuito de confundir possíveis leitores não autorizados. Segundo o historiador David Kahn, especialista na área de inteligência e segurança, este registro é considerado como o primeiro exemplo documentado da escrita cifrada.[KAH67]

- Segundo análise química, datada de aproximadamente 1600 A.C., o disco de Phaistos, encontrado na ilha de Creta há 95 anos (ano base 2003), é o único exemplar deste tipo de escrita.

Dada a falta de outras inscrições semelhantes, os sinais presentes na peça de argila

ainda não foram decifrados. A seguir é apresentada a figura com o artefato.[MIS04]

Figura 2.1. *Disco de Phaistos.*

- Aproximadamente 1500 A.C., mercadores assírios usavam *intaglios* (peças planas de pedra com símbolos entalhados) para sua identificação. Esta técnica pode ser considerada como as primeiras assinaturas a conferir autenticidade registradas.

- Em 550 A.C., escribas hebreus usaram uma cifra de substituição simples pelo alfabeto reverso conhecida como ATBASH no processo de escrita da obra conhecida como o livro de Jeremias.

As cifras mais conhecidas da época são, além do ATBASH, o ALBAM e o ATBAH, as chamadas cifras hebraicas. Conforme as figuras a seguir temos a representação das tabelas de substituição de cada cifra.

A	B	C	D	E	F	G	H	I	J	K	L	M
Z	Y	X	W	V	U	T	S	R	Q	P	O	N

Figura 2.2. *Tabela de Substituição da cifra ATBASH.*

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Figura 2.3. *Tabela de Substituição da cifra ALBAM.*

Figura 2.4. *Tabela de Substituição da cifra ATBAH.*

As cifras hebraicas consistem em substituição simples de uma letra por outra e seus nomes derivam do método de como são feitas essas substituições.

No ATBASH, troca-se a primeira letra *(Aleph)* pela última *(Taw)*, a segunda letra *(Beth)* pela penúltima *(Shin)* e assim sucessivamente. Destas quatro letras deriva o nome da cifra: *Aleph Taw Beth SHin - ATBASH.*

No ALBAM cada letra é deslocada em 13 posições. Observe que a primeira letra do alfabeto hebreu *(Aleph)* é trocada por *(Lamed)* e que *(Beth)* é trocada por *(Mem)*. Daí a origem do nome da cifra: *Aleph Lamed Beth Mem - ALBAM.*

Já a cifra ATBAH não segue um padrão. A primeira letra do alfabeto hebreu *(Aleph)* é trocada por *(Teth)* e a segunda *(Beth)* é trocada por *(Heth)*. Por conseqüência, *Aleph Teth Beth Heth - ATBAH.*

Datada de meados de 500 A.C., uma forma de transposição que utiliza o primeiro dispositivo de criptografia militar conhecido é o *bastão de Licurgo.*

O *bastão de Licurgo* consiste num bastão no qual é enrolada uma tira de couro ou pergaminho. O remetente escreve a mensagem ao longo do bastão e depois desenrola a tira, a qual então se converteu numa sequência de letras sem sentido.

O mensageiro usa a tira como cinto, com as letras voltadas para dentro. O destinatário, ao receber o "cinto", enrola-o no seu bastão, cujo diâmetro é igual ao do bastão do remetente. Desta forma, pode ler a mensagem, conforme representação abaixo.

- Em meados de 300 A.C., ocorreram algumas concepções importantes para a criptografia que, embora não tivessem nenhuma ligação direta com as práticas da época, serviram de base para a criptografia usada na atualidade.

Euclides de Alexandria, matemático grego, viveu aproximadamente de 330 a.C. a 270 a.C. foi responsável por reunir e sistematizar a geometria e a teoria dos números da sua época em seu texto *"Elementos"*.

Figura 2.5. *Alfabeto Hebreu com as tabelas de substituição das cifras.*

Figura 2.6. *Representação do Bastão de Licurgo.*

Erastótenes de Cirene, filósofo e geômetra grego, viveu de 276 a.C. a 194 A.C. Ficou conhecido como o criador de um método para identificar números primos, o *crivo de Erastótenes*, e por ter calculado o diâmetro da Terra com grande precisão.

- Estima-se que em 60 A.C., o imperador Júlio César desenvolveu e empregou uma cifra de substituição para cifrar mensagens governamentais. Ele basicamente avançava a ordem das letras desviando-as em, por exemplo, em três posições; A se tornava D, B se

tornava E, etc. [MIS04]

O que veio então a ser conhecido como *Cifra de César* tornou-se o único da antiguidade a ser usado atualmente, apesar de representar um retrocesso em relação à criptografia existente na época.

Figura 2.7. *Representação do resultado da Cifra de César.*

2.3.1.2 História média Classicamente, a Idade Média vai de 476 D.C. (com a queda do Império Romano) até 1453 D.C. (com a queda de Constantinopla). Um pouco antes, ainda na chamada Antiguidade, alguns fatos relevantes podem ser relacionados à criptografia.

- Em 200 D.C., consta o *Papiro de Leiden*, um trabalho detalhando como fazer poções especiais ou mágicas, possui texto cifrado em codificação exclusiva nos trechos cruciais das receitas. [KAH67]

No período entre 476 a 1453, entre o fim do Império Romano e próximo ao descobrimento do Brasil, encontra-se a chamada *Idade Média*.

Na Europa, o período inicial desta época também foi conhecido como *o período das trevas* onde grande parte do conhecimento de várias ciências foi perdido, entre elas a criptologia, por serem considerados magia negra ou bruxaria.

Nesta época, a contribuição árabe-islâmica foi significativa, documentando estudos como a criptanálise para a substituição monoalfabética. A denominação "Cifra", "Chiffre", "Ziffer", etc, como também "zero", utilizado em muitas línguas, vêm da palavra árabe "sifr", que significa "nulo".

- Em 750 D.C., al-Khalil escreve *Kitab al-mu'amma* (O Livro das Mensagens Criptográficas) sobre criptografia, em grego, para o imperador bizantino, onde decifra um

criptograma bizantino antigo.

A solução documentada baseava-se no início do texto original, que supôs corretamente como sendo "Em nome de Deus" enunciado relativamente comum na época. Este método criptoanalítico tornou-se padrão, tendo sido usado até na decifração de mensagens geradas pela máquina Enigma durante a Segunda Guerra Mundial[KAH67]

- Data de pouco mais de 830 D.C. aquele que é considerado o livro mais antigo ainda intacto sobre criptografia, *análise de freqüencia* (método eficiente de ataque às cifras daquela época), concebido pelo cientista árabe al-Kindi.[SIN00]

Apesar de não se saber com certeza quem foi o primeiro a perceber que a variação na freqüência de letras poderia ser explorada para quebrar cifras, *Risalah fi Istikhraj al Mu'amma* (Escritos sobre a decifração de mensagens criptográficas) foi a descrição mais antiga de que se tem conhecimento a descrever esta técnica.[SUL03]

- A ordem do Templo *(Templários)*, que atuou internacionalmente de 1119 até 1312, utilizava uma cifra própria. Os templários cifravam as letras de crédito que mantinham em circulação entre seus nove mil postos de comando.

Agindo desta maneira, as letras de crédito, que evitavam o transporte de riquezas, circulavam protegidas e "autenticadas". A ordem do Templo foi a primeira organização com características similares aos bancos de que se tem registro.

A cifra era de substituição simples, monoalfabética monogrâmica com a substituição de letras por símbolos. O cifrante, ao invés de ser constituído por letras era constituído por símbolos especiais.

O cifrante foi extraído da cruz chamada *Cruz das Oito Beatitudes* e que constituía o emblema da ordem. De acordo com o diagrama, cada letra do alfabeto é substituída pelas linhas indicadas em vermelho, azul e verde conforme figura abaixo.

- Por volta de 1300, na Itália, iniciou o movimento renascentista que foi responsável pelos primeiros grandes avanços. Em Veneza foi criada uma organização especializada em criptologia no ano de 1452. Eles possuíam três secretarias que solucionavam e criavam cifras para serem usadas pelo governo.

- Em 1378, devido à ordem de unificar o sistema de cifras da Itália Setentrional,

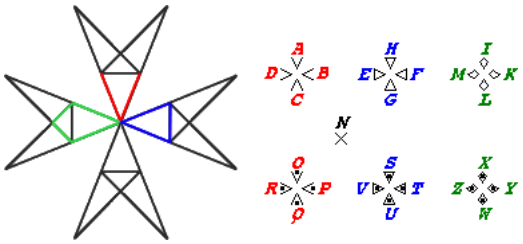

Figura 2.8. *À esquerda a Cruz das 8 Beatitudes e à direita a Cifra dos Templários.*

Gabrieli di Lavinde foi designado por Clemente VII para reunir uma coleção de cifras num manual, do qual ainda hoje o Vaticano conserva uma cópia de 1379.

Com tal alfabeto de substituição combinada (código/cifra), foi unida a cifra de substituição com um código de listas de palavras, sílabas e nomes equivalentes. Este sistema foi amplamente utilizado por diplomatas e alguns civis europeus e americanos por mais de 450 anos.[KAH67]

2.3.1.3 História moderna - Por volta de 1460, Leon Battista Alberti inventou e publicou a primeira cifra polialfabética, criando um disco de cifragem (conhecido atualmente como "Captain Midnight Decoder Badge") para simplicar o processo.

Alberti, que era amigo de Leonardo Dato, um secretário pontificial o qual, provavelmente, introduziu Alberti na criptologia, produziu também tem muitos escritos sobre o estado da arte em cifras, além da sua própria invenção.

Ao que tudo indica a classe de cifra polialfabética criada não foi quebrada até os anos de 1800. Alberti Também fez uso do seu disco para obter código cifrado. Estes sistemas eram muito mais fortes que a nomenclatura usada por diplomatas da época e continuaram sendo por muitos séculos mais.[KAH67]

O "Trattati in cifra"de Leone Battista Alberti foi publicado em Roma em 1470. Se referia "especialmente a teorias e processos de cifragem, métodos de decifração e dados estatísticos"[GAL45]

Leon Battista Alberti ficou conhecido como "O Pai da Criptologia Ocidental", em parte devido a sua nova classe de substituição.

De maneira mais detalhada, a substituição polialfabética é uma técnica que permite que diferentes símbolos cifrados possam representar o mesmo símbolo do texto claro. Isto dificulta a interpretação do texto cifrado pela aplicação da análise de frequência.

Para desenvolver esta técnica, Alberti estudou os métodos para quebrar cifras da época e elaborou uma cifra que poderia anular estes métodos.

A cifra de Alberti é uma das cifras polialfabéticas mais seguras, porém, não foi amplamente explorada devido a decisão do autor em mantê-la secreta. Seu tratado "De Componendis Cyphris"somente foi publicado em Veneza um século mais tarde como parte de outro, também de sua autoria, o "oposcoli morali", e passou quase que desapercebido.

Alberti completou esta descoberta, determinante na história da criptologia, com uma outra invenção importante: o código de recifragem. Ele compôs uma tabela com todas as combinações possíveis das cifras 1, 2, 3, 4, de 11 a 4444. Desta forma, obteve 336 grupos que utilizava como um pequeno repertório.

Segundo Alberti: "Introduzi estas palavras na minha mensagem e, pela aplicação do procedimento, elas foram substituídas pelas letras que lhe eram correspondentes". Assim, suponha que 341 signifique "Pape". Esta palavra se transformará em "iqh"numa posição do disco e em "dgb"numa outra.

Figura 2.9. *Conteúdo do Disco de Cifragem* **Captain Midnight Decoder Badge**.

- Sicco Simonetta publicou em 1474 um pequeno trabalho chamado *"Regulae ad extrahendum litteras zifferatas sine exemplo"* que ressaltava *"métodos de decifração e fornecendo dados estatísticos consideráveis"* [GAL45]

A data deste trabalho sobre cifras foi considerada importante porque se tratava de um período no qual a criptologia se tornou prática universal. Foi o momento que que as

Figura 2.10. *Disco de Cifragem* **Captain Midnight Decoder Badge**.

cifras simples evoluíram para criptogramas mais complexos. [GAL45]

- Em 1518, Johannes von Heydenberg aus Trittenheim/Mosel, ou Johannes Trithemius, escreveu o primeiro livro impresso de criptologia, a *Polygraphiae libri sex de Trithemius*, a qual incluía sua tabela de substituição *tabula recta Caesar*.

Apesar de haver dúvidas quanto à data correta. Foi reimpressa em 1550, 1564, 1571 e 1600. Uma tradução em Francês apareceu em 1561 e em 1564. Trithemius apresentou nela uma cifra esteganográfica na qual cada letra era representada como uma palavra obtida de uma sucessão de colunas.[GAL45]

A série de palavras resultantes seria uma oração legítima.

Também descreveu cifras polialfabéticas na forma de tabelas de substituição retangulares que, na época, já tinham se tornado padrão, introduzindo também a noção da troca de alfabetos a cada letra.

Escreveu, porém não publicou, sua *Steganographia*, a qual circulou como manuscrito por mais de cem anos, sendo copiada por muitas pessoas que desejavam extrair os segredos que se pensava que continha.

A *tabula recta* funcionava da seguinte forma, cifrava-se a primeira letra da mensagem clara com a primeira linha, a décima segunda letra com a décima segunda linha, e assim sucessivamente. Quando alcançava a última linha da tabela, recomeçava com a primeira linha.

Na figura que representa a *Tabula Recta*, está presente um texto em latim que descreve seu funcionamento. O texto é o seguinte:

In hac tabula literarum canonica sive recta tot ex uno et usuali nostro latinarum lite-

rarum ipsarum per mutationem seu transpositionem habes alphabeta, quot in ea per totum sunt monogrammata, videlicet quater et vigesies quatuor et viginti, quae faciunt in numero DLXXVI. ac per totidem multiplicata, paulo efficiunt minus quam quatuordecemmilia.

A tradução seria:

Sobre esta tabela regular ou carreiras de letras coloca-se, por permutação ou por transposição, o alfabeto usual das nossas letras latinas; ou então, coloca-se nesta tabela todos os monogramas, de 24 em 24, o que totaliza um número de 576 e, multiplicando por outro tanto (24) perfazem um pouco menos de 14.000.

As letras que correspondem à primeira linha não são cifradas, enquanto que as restantes são deslocadas no alfabeto as posições correspondentes ao número da linha. É como se fosse um Código de César com um deslocamento diferente para cada letra.

In hac tabula literarū canonica siue recta tot ex uno & usuali nostro latinarum literarum ipsarum per mutationem seu transpositionē habes alphabeta, quot in ea per totum sunt monogrammata, uidelicet quater & uigesies quatuor & uiginti, quæ faciunt in numero ɔ. lxxvi. ac per totidē multiplicata, paulo efficiunt minus q̄ quatuordecē milia.

o ij

Figura 2.11. *Representação da* **Tabula Recta**.

A seguir temos uma representação da cifra onde foi considerado o alfabeto latino de

26 letras.

Mensagem Clara	C	R	I	P	T	O	G	R	A	F	I	A
Deslocamento	0	1	2	3	4	5	6	7	8	9	10	11
Mensagem Cifrada	C	S	K	S	X	T	M	Y	I	O	S	L

Figura 2.12. *Aplicação da cifra na palavra* **criptografia**.

- Em 1533, Heinrich Cornelius Agrippa von Nettelsheim publica o *De occulta philosophia*, em Colônia, na Alemanha.

No livro 3, capítulo 30, descreve sua cifra de substituição monoalfabética, hoje conhecida como *Cifra de Pig Pen*, cuja tradução literal do nome é *Porco no Chiqueiro* e vem do fato de que cada uma das letras (os porcos) ser colocada numa "casa"(o chiqueiro).

Na época, a cifra aparentemente teve boa importância pois, alguns anos mais tarde, Vigenère a reproduz no seu trabalho *Traicté des chiffres* (Paris, 1586, f. 275 v). Abaixo temos a representação de tabela da cifra de *Agrippa*.

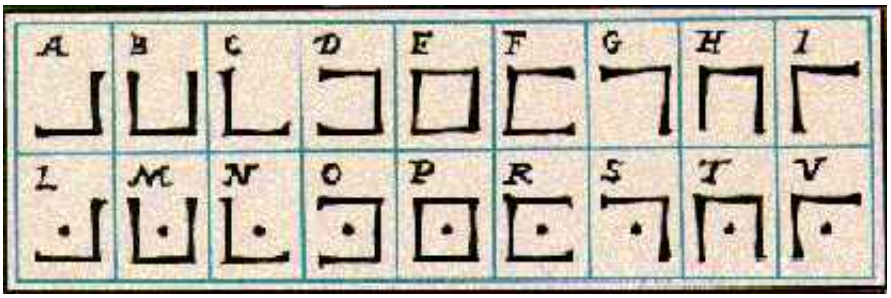

Figura 2.13. *Representação da* **Tabela de Agrippa**.

- Foi publicado em 1550 a obra *De subtilitate libri XXI* de Girolamo Cardano. Segundo Galland, tal obra de tal matemático, físico e filósofo continha uma quantidade considerável de informações a respeito de processos de cifragem. [GAL45]

A obra foi reimpressa em 1551, duas vezes em 54, em 59, outras duas vezes em 60, e em 80 e 82. Uma tradução francesa foi impressa em 1556.

Cardano inventou o primeiro procedimento com auto-chave, mas seu sistema não foi considerado perfeito.

A grelha de Cardano, como ficou conhecida, consistia em uma folha de material rígido onde se encontram, em intervalos ireegulares, pequenas aberturas retangulares da altura de uma linha de escrita e de comprimento variável.

O remetente escreve o texto nas aberturas, depois retira a folha e completa os espaços vazios com letras quaisquer. O destinatário põe a mesma grelha sobre o texto cifrado para ler a mensagem.

Abaixo temos a representação de uma mensagem cifrada com a *Grelha de Cardano*. A mensagem decifrada é *"Mensagem Secreta"*.

G	M	P	A	L	O	E	M	T	N
P	N	I	S	D	L	A	G	U	R
E	M	J	S	R	L	E	T	A	C
I	D	R	U	V	N	O	R	A	N
H	O	Q	U	E	Z	A	P	T	A

Figura 2.14. *Mensagem cifrada com a Grelha de Cardano.*

	M					E			N
			S			A	G		
E	M		S			E			C
		R							
				E				T	A

Figura 2.15. *Representação da mensagem decifrada.*

- No ano de 1553, Giovanni Battista Bellaso publicou em Veneza o livro *La Cifra del Sig.* Logo a seguir, em 1555, publica em Bréscia o *Novi et singolari modi di cifrare* e nove anos mais tarde, em 1564, também em Bréscia, publica *Il vero modo di scrivere in cifra*

Todos os trabalhos sobre criptologia, cujo grande mérito foi introduzir o conceito e o uso de palavras-chave, o antepassado direto da senha. Muito parecida com a cifra de Alberti porque usa alfabetos cifrantes invertidos e não de todo arbitrários.

A idéia de Bellaso é de utilizar diversos alfabetos desordenados de acordo com uma palavra ou frase convencionada, o que seria a senha. As letras da palavra secreta são escritas à esquerda, no início e em duas linhas. As letras do alfabeto restantes são escritas a seguir, também em duas linhas.

Utiliza-se o alfabeto latino de 20 letras, onde J=I, U=V e não há as letras K, Y e W. O exemplo original é com a palavra-chave IOVE, onde o Z também é excluído.

A imagem 2.16 representa um exemplo de como obter o primeiro alfabeto derivado onde a palavra-chave é MOLA:

M	**O**	B	C	D	E	F	G	H	I
L	**A**	N	P	Q	R	S	T	V	X

Figura 2.16. *Representação de alfabeto derivado da palavra-chave* **MOLA**.

A imagem 2.17 representa o método de obtenção do segundo alfabeto a partir do deslocamento circular da segunda linha:

M	**O**	B	C	D	E	F	G	H	I
X	**L**	**A**	N	P	Q	R	S	T	V

Figura 2.17. *Representação de segundo alfabeto.*

Utiliza-se o mesmo processo até serem obtidos 5 alfabetos derivados e cada um deles será identificado por um grupo de 4 letras, como mostrado na figura 2.18.

M E L R	**M**	O	B	C	D	**E**	F	G	H	I
	L	A	N	P	Q	**R**	S	T	V	X
O F A S	M	O	B	C	D	E	F	G	H	I
	X	L	A	N	P	Q	R	S	T	V
B G N T	M	O	B	C	D	E	F	G	H	I
	V	X	L	A	N	P	Q	R	S	T
C H P V	M	O	B	C	D	E	F	G	H	I
	T	V	X	L	A	N	P	Q	R	S
D I Q X	M	O	B	C	D	E	F	G	H	I
	S	T	V	X	L	A	N	P	Q	R

Figura 2.18. *Representação dos 5 alfabetos.*

Neste momento, deve-se usar uma segunda palavra-chave (ou frase), por exemplo *GENIAL*. Esta segunda palavra-chave serve de guia para escolher o alfabeto cifrante correspondente a cada palavra. Na figura 2.19 temos o texto "invenção da palavra-chave"cifrado:

Palavra-chave	G	E	N	I
Texto Claro	INVENCAO	DA	PALAVRA	CHAVE
Texto Cifrado	TDMPDACX	QO	ECBCMGC	XQEBA

Figura 2.19. *Representação da cifragem de texto.*

- Em 1556, Cardano publica *De rerum varietate libri XVII*, o qual continha informações criptográficas e era a continuação do seu popular *De Subtilitate*. Ambos os livros foram traduzidos por toda a Europa. *De rerum* foi reimpresso em 1557, 58, 80 e 81. [KAH67]

- No ano de 1563 foi escrita a obra *De furtivis literarum notis-vulgo de ziferis* do físico italiano Giambattista Della Porta, composto por quatro volumes que tratam, respectivamente, de cifras da antigüidade, de cifras modernas, da criptoanálise e das características lingüísticas que facilitam a decifração.[GAL45]

Della Porta classifica os procedimentos em três categorias: a mudança da ordem das letras (transposição), das suas formas (substituição por símbolos) e do seu valor (substituição por um alfabeto criptográfico).

Apesar de resumido, este é o primeiro exemplo da divisão dos procedimentos, atualmente clássica, em dois princípios: transposição e substituição.

Della Porta também foi o inventor do primeiro sistema literal de chave dupla, ou seja, a primeira cifra na qual o alfabeto cifrante muda a cada letra. Este sistema polialfabético era extremamente robusto para a época, de modo que muitos consideram Della Porta como o *"Pai da Criptografia Moderna"* [SIN00]

Della Porta empregou 11 alfabetos diferentes e reversíveis que ele designou por AB, CD, EF, etc. que podem ser vistos na figura 2.20. O princípio é o mesmo do deslocamento

circular visto na *Cifra de Bellaso*.

Para cifrar uma letra, escolhe-se um alfabeto, digamos o alfabeto MN. Neste caso, o A será trocado por U e a letra U será trocada pela letra A; o B será substituído por V e a letra V pela letra B, e assim por diante.

Percebe-se que esta é uma cifra reversível: se cifrarmos um texto cifrado com a mesma chave, obtemos novamente o texto claro. Para não ter que usar todos os alfabetos (originalmente, onze), Della Porta sugere o uso de quatro, cinco ou seis alfabetos.

Além do que foi mencionado, Della Porta propõe o uso de uma palavra chave cujas letras indicam os alfabetos que devem ser utilizados sucessivamente. Esta palavra-chave constitui a chave do criptograma.

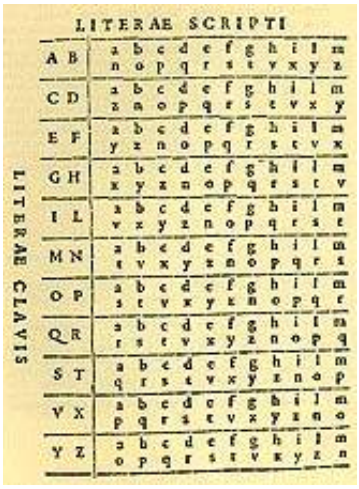

Figura 2.20. *Cifra original de Della Porta com 11 Alfabetos.*

Temos na figura 2.21 uma tabela baseada na cifra de Della Porta com 13 alfabetos ao invés dos 11 originais, afim de poder cifrar todas as letras do nosso alfabeto ocidental atual

- No ano de 1586, Blaise de Vigenère publicou o *Traicté des Chiffres où secrètes manières d'escrire*, onde descreve os primeiros sistemas autenticos de texto claro e texto cifrado com auto-chave nos quais letras prévias do texto claro ou cifrado são usadas para a letra da chave atual.

Neste mesmo trabalho, Vigenère publicou sua tabela de substituição, conhecida atu-

AB	A N	B O	C P	D Q	E R	F S	G T	H U	I V	J W	K X	L Y	M Z											
CD	A Z	B N	C O	D P	E Q	F R	G S	H T	I U	J V	K W	L X	M Y											
EF	A Y	B Z	C N	D O	E P	F Q	G R	H S	I T	J U	K V	L W	M X											
GH	A X	B Y	C Z	D N	E O	F P	G Q	H R	I S	J T	K U	L V	M W											
IJ	A W	B X	C Y	D Z	E N	F O	G P	H Q	I R	J S	K T	L U	M V											
KL	A V	B W	C X	D Y	E Z	F N	G O	H P	I Q	J R	K S	L T	M U											
MN	A U	B V	C W	D X	E Y	F Z	G N	H O	I P	J Q	K R	L S	M T											
OP	A T	B U	C V	D W	E X	F Y	G Z	H N	I O	J P	K Q	L R	M S											
QR	A S	B T	C U	D V	E W	F X	G Y	H Z	I N	J O	K P	L Q	M R											
ST	A R	B S	C T	D U	E V	F W	G X	H Y	I Z	J N	K O	L P	M Q											
UV	A Q	B R	C S	D T	E U	F V	G W	H X	I Y	J Z	K N	L O	M P											
WX	A P	B Q	C R	D S	E T	F U	G V	H W	I X	J Y	K Z	L N	M O											
YZ	A O	B P	C Q	D R	E S	F T	G U	H V	I W	J X	K Y	L Z	M N											

Figura 2.21. *Cifra baseada na de Della Porta, ajustada para o alfabeto ocidental atual contendo 13 Alfabetos.*

almente por *Carreiras de Vigenère.*[SIN00]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
26	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Figura 2.22. *Cifra de Vignère, conhecida também como* **Vignère Tableau** *e* **Carreiras de Vignere**.

- Em 1918, Arthur Scherbius idealizou a máquina *Enigma*, cuja estrutura e funcionamento a descreveu como uma versão elétrica do *Disco de Alberti*.

Consistia de um teclado com 26 letras, um painel com 26 letras individualmente iluminadas por uma pequena lâmpada onde era lido ou gerado o código e um disposi-

tivo denominado *"scrambler"*. Este dispositivo era constituído, em uma das versões da máquina, de três rotores e um refletor.

O painel de conexões consistia de 26 contatos, um para cada letra, 18 onde conectava-se 6 cabos diferentes, cada um deles com um plug em cada uma das extremidades, permitindo a permutação de até 12 letras.

Os três rotores giravam num mesmo eixo, avançando uma posição a cada letra digitada, sendo que cada um possuía dois conjuntos de 26 contatos, um do lado esquerdo e outro do lado direito, conectados internamente entre si por condutores elétricos de tal modo que não existia correspondência direta entre eles. [KAH67]

Temos a seguir a ilustração da máquina alemã *Enigma*.

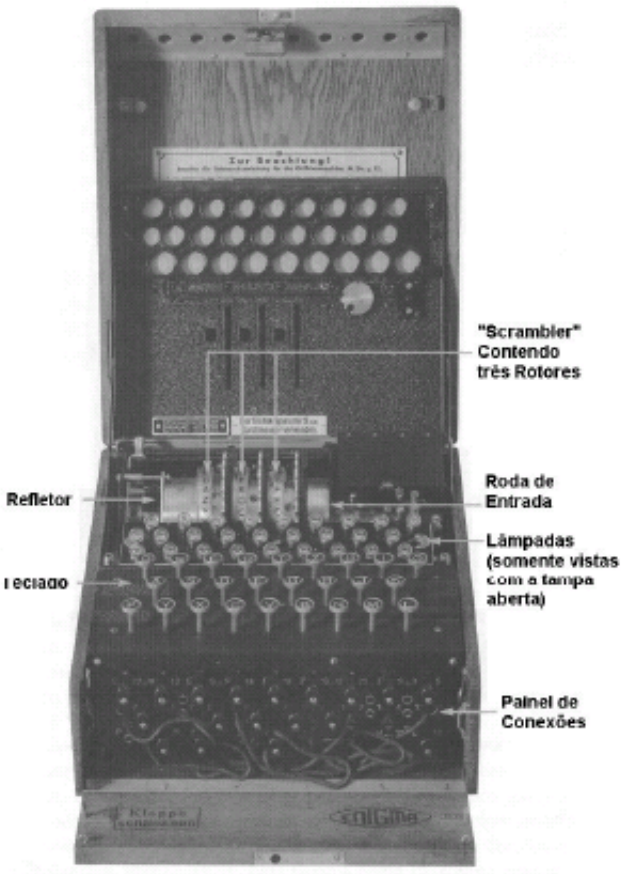

Figura 2.23. *A máquina* **Enigma** *aberta.*

O refletor fazia com que a corrente elétrica retornasse aos rotores seguindo o caminho

inverso através dos circuitos internos para então acender a letra correspondente no painel. O uso do refletor evitava que uma letra pudesse ser cifrada por ela própria.

Na ilustração a seguir é descrito o seu funcionamento funcionamento. Quando pressionada a letra *"B"*, o painel de conexões inicialmente efetua uma substituição, trocando a letra *"B"* por *"A"*, conduzindo a corrente elétrica pela nova saída por ele estabelecida.

A corrente elétrica será conduzida pelos contatos através dos rotores até alcançar o refletor que a encaminhará por outro contato de volta para os rotores seguindo até acender a letra correspondente no painel luminoso.[SIN00]

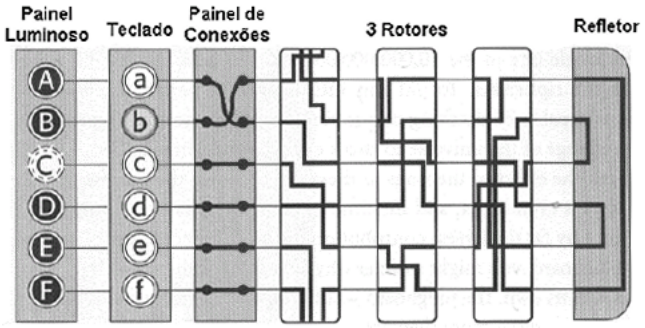

Figura 2.24. *Descrição do funcionamento da máquina* **Enigma**.
[SUL03]

O Emissor então anotava a mensagem cifrada que era enviada ao destinatário por rádio, telégrafo, telefone ou correio. O Receptor utilizando uma Enigma com configuração idêntica digitava a mensagem cifrada e, tomando nota das letras iluminadas no painel, obtinha a mensagem decifrada.

Como a máquina não tinha nenhum dispositivo de saída, sua utilização não era prática, uma vez que os operadores precisavam digitar as mensagens e anotar as cifras geradas.[SUL03]

2.3.1.4 História recente No período considerado *Recente*, entre 1790 a 1900, o detalhe marcante foram os sistemas de comunicação à distância, por serem sistemas abertos, deram um novo impulso à criptografia.

Da mesma maneira que garantiam as grandes vantagens de uma comunicação rápida e eficiente a distância, por outro lado, as mensagens trafegariam basicamente desprotegidas. Este aspecto viria a ser tratado por técnicas da criptologia.

- Em meados de 1790, Thomas Jefferson, possivelmente com a ajuda do Dr. Robert Patterson, um matemático da Universidade da Pensilvânia, inventa sua *Roda Criptográfica* (também conhecido como *cilindro cifrante*, *cifra de roda do Jefferson* ou do inglês *Jefferson Cipher Weel*).

Apesar da engenhosidade do dispositivo composto por 26 discos, nunca chegou a ser utilizado em massa. Segundo Misagui, Jefferson e James Monroe cifravam as suas cartas para manter em sigilo as suas discussões políticas.[MIS04]

O cilindro de Jefferson é um dispositivo que permite realizar com rapidez e segurança uma substituição polialfabética.

Os cilindros cifrantes são, por assim dizer, uma invenção do século XIX. Foram reinventados por diversas vezes e utilizados pelos militares no século XX, até a Segunda Guerra Mundial.

Na figura a seguir, numa das linhas é possível ler *"THOMAS JEFFERSON WHEEL CIPHER"*. Esta seria a mensagem plana. O remetente, deve então escolher qualquer outra linha e a enviar ao destinatário.

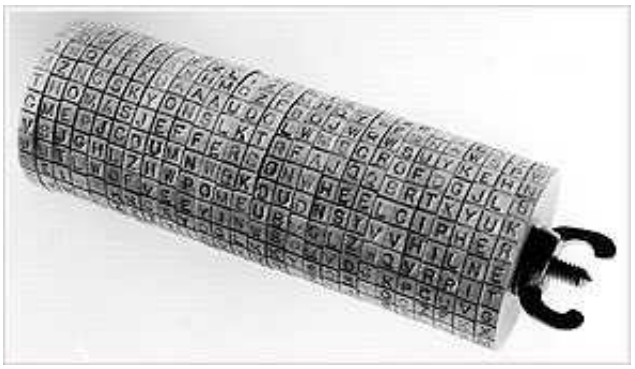

Figura 2.25. *Roda Cifrante de Jefferson.*

Se usarmos como base a linha imediatamente acima da mensagem que identificamos teríamos uma mensagem cifrada para envio com o seguinte conteúdo:

MZNCSK YONSLKTRF AJQQB RTXYUK

O destinatário, que possui um cilindro com a mesma sequência de discos exatamente iguais é então capaz de transferir a mensagem recebida para o seu cilindro e fazer a busca por uma linha com texto legível.

- EM 1840, Samuel Morse desenvolve o código que recebeu o seu nome. Na verdade não é um código, mas sim um alfabeto cifrado em sons curtos e longos.

Morse, que era artista e não cientista, também foi o inventor de um dispositivo que chamou de telégrafo e, em 1844, enviou sua primeira mensagem com os dizeres *"What hath God wrought"*.

- Em 1937, Alan Turing (importante personalidade da história não só da criptografia como também da computação) publica seu mais importante artigo intitulado *"On Computable Numbers"* onde descreve uma máquina que poderia efetuar de forma automática os processos geralmente desenvolvidos por um matemático.

De acordo com a proposta, deveria haver uma máquina para cada processo (soma, divisão, calculo de integrais e assim por diante). Tais máquinas ficaram conhecidas como *Máquinas de Turing*.

Posteriormente, Turing chegou a conclusão de que poderia, em vez de ter uma máquina específica para cada processo matemático, desenhar um modelo universal que tivesse condições de realizar todas as operações, desde que fossem programadas para tal.[SUL03]

Operando desta forma, foi então criada a *Máquina Universal de Turing*, cujos computadores atuais ainda possuem sua fundamentação conceitual para processamento.

Em 1939, Turing recebeu o convite para fazer parte da equipe de criptoanalistas da *Escola de Códigos e Cifras do Governo da Inglaterra*. O objetivo principal do Instituto era quebrar as chaves da máquina alemã *Enigma*.[SIN00]

- No ano de 1799 o espanhol Francisco de Paula Martí escreveu *Poligrafía, ó Arte de Escribir en Cifra de Diferenter Modos*, baseando-se nas obras de Trithemius e Kircher.

Martí expõe então uma variedade de cifras, entre elas cifras de substituição numéricas e alfabéticas. A segunda parte da obra trata da escrita invisível, onde Martí descreve métodos para tornar textos legíveis em mensagens onde a tinta tenha desbotado ou que tenham sido escritas com tinta invisível.

- Em 1934, Louis Braille, educador francês que ficou cego aos 3 anos de idade, interessou-se por um sistema de escrita, apresentado na escola Charles Barbier, no qual uma mensagem codificada em pontos era cunhada em papel-cartão.

Braille então aos seus 15 anos de idade trabalhou numa adaptação, escrita com um instrumento simples. O Código Braille consiste de 63 caracteres, cada um deles constituído por 1 a 6 pontos dispostos numa matriz ou célula de seis posições.

Mais tarde adaptou este sistema para a notação musical. Publicou tratados sobre seu sistema em 1829 e 1837. O Sistema Braille é universalmente aceito e utilizado até os dias de hoje.

- Em 1854, o matemático inglês Charles Babbage, hoje chamado de "o pai do computador", quebra a cifra de Vigenère, considerada inquebrável desde o século XVII, e projeta as primeiras máquinas de cáculo sofisticadas, precursoras do computador. A *"Máquina das Diferenças"* e a *"Máquina Analítica"*.

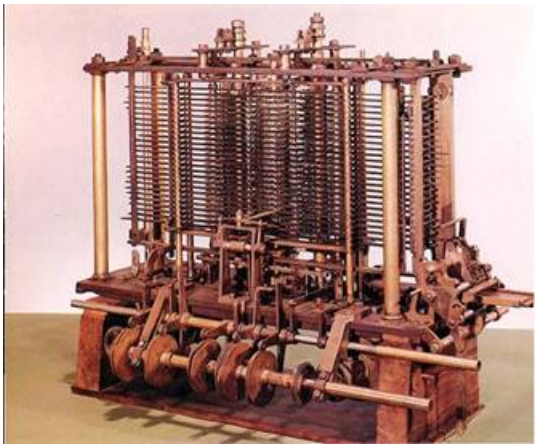

Figura 2.26. Máquina Analítica *de Babbage.*

2.3.1.5 História atual O período a partir do ano de 1900 é considerado como atual. Os computadores são o diferencial do século XX revolucionando a informação e também exercendo grande impacto na criptologia.

- Em 1901, Guglielmo Marconi dá início a era da comunicação sem fio. Apesar da vantagem de uma comunicação de longa distância sem o uso de fios ou cabos, o sistema

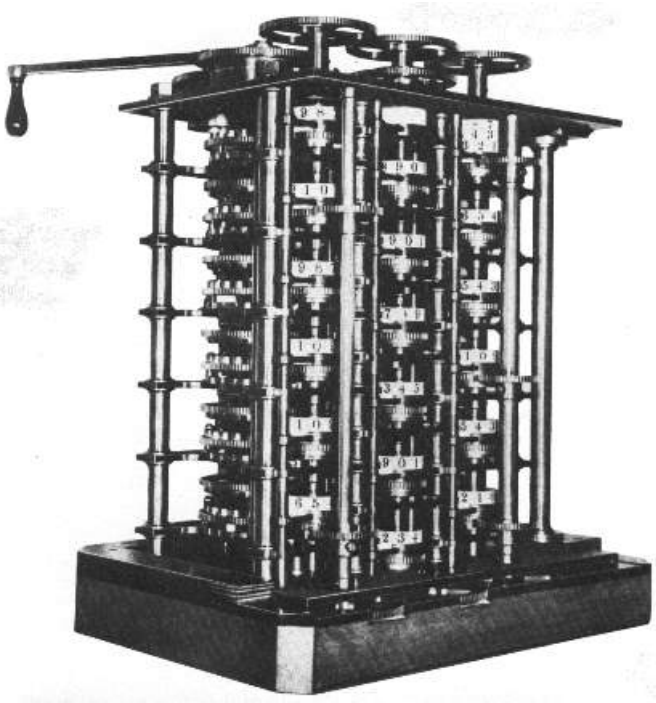

Figura 2.27. Máquina Diferencial *de Babbage.*

é aberto e aumenta o desafio da criptologia.

Inicialmente a telegrafia sem fio utilizava apenas o código *Morse*, acessível a todos que captassem os sinais. Impunha-se a necessidade de codificações que garantissem o sigilo das mensagens.

- Em 1917, William Frederick Friedman, o homem que introduziu o termo *"criptoanálise"* e que posteriormente será chamado de "pai da criptoanálise dos EUA", começa a trabalhar como criptoanalista civil no Riverbank Laboratories, que também presta serviços ao governo dos EUA.

Mais tarde, Friedman cria uma escola de criptoanálise militar, inicialmente no Riverbank e depois em Washington.

- Entre os anos de 1913 e 1930 são colocadas em uso várias máquinas de cifrar, incluíndo a alemã *Enigma*, a inglesa *TypeX* e a norte-americana SIGABA (M-134-C). Todas com propósitos militares.

- No ano de 1937 foi criada a máquina cifradora japonesa *Purple Machine* a partir das revelações feitas por Herbert O. Yardley. Seu sistema criptográfico foi quebrado por

uma equipe liderada por William Frederick Friedman.

A Máquina Púrpura usava relês telefônicos escalonados ao invés de rotores, apresentando, portanto, permutações totalmente diferentes a cada passo ao invés das permutações relacionadas de um rotor em diferentes posições (consideradas convencionais).

- Em 1960, o Dr. Horst Feistel, liderando um projeto de pesquisa no IBM Watson Research Lab, desenvolveu a cifra *Lucifer*. Alguns anos depois, esta cifra servirá de base para o *DES* e outros produtos cifrantes, criando uma família conhecida como *"cifras Feistel"*.

- Em 1969, James Ellis desenvolve um sistema de chaves públicas e chaves privadas separadas.

- Em 1974 a IBM apresenta a cifra *Lucifer* ao *NBS* (National Bureau of Standards) o qual, após avaliar o algoritmo com a ajuda da *NSA* (National Security Agency), introduz algumas modificações.

Caixas S e uma chave menor são as principais características alteradas, sendo adotada em seguida como a cifra padrão de encriptação de dados para os EUA, *FIPS PUB-46*, conhecido hoje como *DES* (Data Encryption Standard).

Hoje o *NBS* é chamado de National Institute of Standards and Technology, NIST. Conheça e entenda o algoritmo DES, a cifra de bloco mais conhecida do mundo.[oST05]

- Em 1978 é publicado o algoritmo RSA.[oST05]

- A Criptografia de curva elíptica é sugerida por Miller em 1986.

Durante a década de 90, trabalhos com computadores quânticos, criptografia quântica, biométrica para autenticação (impressões digitais, a íris, etc) foram os destaques essencias.

- Em 1991, Phil Zimmermann torna pública sua primeira versão de *PGP* (Pretty Good Privacy) como resposta ao *FBI*, o qual invoca o direito de acessar qualquer texto claro da comunicações entre cidadãos.

O *PGP* oferece uma segurança alta para o cidadão comum e, como tal, pode ser encarado como um concorrente de produtos comerciais como o *Mailsafe* da *RSADSI*.[GAR95]

A característica determinante no processo de adoção do *PGP* foi o fato do mesmo ter sido disponibilizado como freeware e, como resultado, tornou-se um padrão mundial

enquanto que seus concorrentes da época continuaram absolutamente desconhecidos.

- Em 1997 o código DES de 56 bits é quebrado por uma rede de 14.000 computadores. Em 1998 o código DES é quebrado novamente em apenas 56 horas por pesquisadores do Vale do Silício. Em 1999 o código DES é quebrado em apenas 22 horas e 15 minutos.[oST05]

2.4 Criptografia

A criptografia é um tema pertencente ao domínio da criptologia e aborda maneiras para processamento (computacional ou não) de documentos com o objetivo de esconder seu conteúdo, validar sua integridade e proteger seu uso contra pessoas não autorizadas.

Segundo *Bruce Schneier*, a criptografia deve prover outros serviços além de confidencialidade. São eles:

- *Capacidade de autenticação.* Deve ser possível ao receptor de uma mensagem certificar-se da veracidade de sua origem, onde um intruso não deve ser capaz de mascarar-se como outro indivíduo.

- *Verificação de integridade.* Deve ser possível ao receptor de uma mensagem verificar se o seu conteúdo foi modificado em trânsito, onde um intruso não deve ser capaz de substituir uma mensagem legítima por uma falsificada.

- *A não-retratação.* Um emissor não deve ser capaz de falsamente negar mais tarde que enviou uma mensagem. [SCH96]

Ainda segundo *Bruce Schneier*, os métodos criptográficos estão divididos em 4 tipos conhecidos como:

- Sistemas de cifras.

- Sistemas de códigos.

- Esteganografia.

- Criptografia atual.

2.4.1 Sistemas de cifras

Basicamente, a diferença fundamental dos sistemas de cifras está nos métodos que são empregados nela. Transposição e Substituição.

A diferença entre eles é que, na substituição, o valor normal ou convencional das letras do texto original é mudado, sem que sua posição seja mudada; na transposição, apenas a posição das letras do texto original é mudada, sem qualquer alteração no seu valor normal ou convencional.

Como os métodos de encriptação são radicalmente diferentes, os princípios envolvidos na criptoanálise dos dois métodos também são fundamentalmente diferentes.

2.4.1.1 Cifras de substituição monoalfabéticas Na substituição monoalfabética, ou substituição simples, substitui-se cada um dos caracteres do texto original por outro, de acordo com uma tabela pré-estabelecida.[SUL03]

É possível também realizar este tipo de substituição de acordo com uma chave, que nesse caso será um número que indica quantas posições deve-se avançar no alfabeto, para se obter o texto cifrado.

É possível identificar que a freqüência de ocorrência dos símbolos que resultam na mensagem cifrada é a mesma das letras da língua usada na mensagem original. Sendo então mais fácil para o criptoanalista identificar a cifra original a partir do padrão de comportamento identificado.

Este tipo de substituição pode ser *monográfica* ou *poligráfica*.

Quando usada a substituição *monográfica*, verifica-se que cada letra da mensagem original deve ser substituída por apenas um símbolo. Esta regra resultará consequentemente em uma mensagem cifrada de comprimento igual ao da mensagem original.

Quando usada a substituição *poligráfica*, cada letra da mensagem original deve ser substituída por mais de umsimbolo, resultando em uma mensagem cifrada de comprimento diferente da mensagem original.

São exemplos deste tipo de técnica as *Cifras Hebraicas, Cifra de Cesar, Cifra dos Templários*, dentre outras.

2.4.1.2 Cifras de substituição polialfabéticas As substituições polialfabéticas consistem em substituir um conjunto de letras por outro conjunto de letras. Os conjuntos de letras (alfabeto) não precisam necessariamente ser de origens diferentes, como o alfabeto romano e o cirílico, por exemplo.

O simples fato de alterar a ordem na seqüência das letras já caracteriza um *"novo alfabeto"*.

Por exemplo, z-y-x-w-v-u... é um alfabeto de substituição; b-a-d-c-f-e... é um outro alfabeto de substituição diferente. Se ambos forem utilizados para cifrar uma mesma mensagem, substituindo as letras originais, então trata-se de uma substituição polialfabética.

A substituição polialfabética pode ser com palavra-chave ou com auto-chave. Num sistema de palavra-chave é essa que indica os alfabetos cifrantes que devem ser usados.

Já com auto-chave há uma chave que indica a escolha inicial do alfabeto cifrante e depois a própria mensagem determina os alfabetos subsequentes.[SUL03]

São exemplos de cifras de substituição pilialfabéticas o *Disco de Alberti, A Tabula Recta de Trithemius, O Cilindro de Jefferson*, dentre outros.

2.4.1.3 Cifras de transposição A transposição consiste em embaralhar as letras das mensagens de acordo com um padrão definido, sem alterá-las[SUL03]

É um criptograma no qual os caracteres do texto original, que podem ser tratados um a um ou em grupo de comprimento constante, são deslocados de acordo com um sistema predefinido e uma chave.

É importante salientar que os caracteres (letras, números ou símbolos) não são modificados, eles apenas mudam de posição conforme a cifra.

Um exemplo histórico do uso do método de cifragem por transposição é o *Bastão de Licurgo*.

2.4.2 Sistemas de códigos

Um código é essencialmente uma linguagem secreta inventada para esconder o sentido de uma mensagem. Nesta técnica é comum a simples troca de uma palavra por outra ou a contextualização de cenários com personagens adversos.

Na frase, *"Garça mãe, aqui é a garça filha, identifiquei o corvo na árvore."*, é possível perceber que um agente *(Garça filha)* usa palavras-código para avisar o agente *(Garça mãe)* que o alvo *(Corvo)* está no local esperado *(Árvore)*.

Na prática, códigos de espionagem ou militares são denominados comumente de números-código ao invés de palavras-código. São utilizados livros de código que fornecem um dicionário de números e suas respectivas palavras. Por exemplo, a mensagem acima poderia ser codificada como:

91258 91588 00178 36778

Onde "91258"seria *(Garça mãe)*, "91588"seria *(Garça filha)*, "00178"seria *(Corvo)* e "36778"seria *(Árvore)*.

São exemplos do uso dos sistemas de códigos em mensagens o *Código Braille e o Código Morse*.

2.4.3 Esteganografia

Dentre as técnicas para proteção do conteúdo de documentos, existe também a *esteganografia*, palavra que vem do grego *steganos* (coberto) e *graphein* (escrever). Esta técnica realiza a comunicação secreta por ocultação da mensagem.

Contrariamente à criptografia, que cifra as mensagens de modo a torná-las incompreensíveis, a esteganografia esconde as mensagens através de artifícios, por exemplo imagens ou um texto que tenha sentido mas que sirva apenas de suporte.

São exemplos históricos do uso da esteganografia o como o alfabeto biliteral de Francis Bacon ou as famosas cartas de George Sand.

A idéia é a mesma das grelhas de Cardano e o "barn code": mesclar a mensagem numa outra e onde apenas determinadas palavras devem ser lidas para descobrir o texto

camuflado.

2.4.4 Criptografia atual

Na obra de Schneier, encontramos a definição de algoritmo criptográfico, também chamado de cifrador *(em inglês "Cipher")* como uma função matematica usada para encriptar e decriptar (Genéricamente, existem duas funções relacionadas. Uma para encriptação e outra para decriptação).

Se a segurança de um algoritmo é baseada no segredo de seu funcionamento, então ele é chamado de *algoritmo restrito*, também conhecido como algoritmo seguro por obscuridade.

Algoritmos restritos tem um grande interesse histórico, mas são considerados inadequados atualmente, principalmente quando consideradas as condições de segurança que esta classificação é capaz de oferecer.

Por exemplo, se tivermos um grupo de usuários muito grande ou que ofereça rotatividade acentuada de membros, não será adequado seu uso, pois cada vez que um usuário deixar o grupo todos os outros deverão migrar para um algoritmo diferente.

Se algum dos membros acidentalmente revelar o segredo, todos os outros deverão substituir o algoritmo também.

Mesmo com todos estes motivos para não se usar algoritmos restritos, eles são utilizados em grande escala, principalmente em aplicações de baixo nível de segurança

O usuários simplesmente desconhecem ou não se importam com os cuidados inerentes à segurança destes sistemas.

A criptografia moderna resolve este problema fazendo uso de *chave (Key)*, na função representada pela letra K. Esta chave deve poder ser qualquer valor de qualquer tamanho. O número de valores possíveis de uma chave é chamado *KeySpace*. Tanto o método de encriptação como de decriptação usam esta chave (eles são dependentes desta chave, fato representado pela letra k no exemplo), onde as funções seriam:

$\mathrm{E}K(\mathrm{M}) = \mathrm{C}$

$\mathrm{D}K(\mathrm{C}) = \mathrm{M}$

(Estas funções teriam a propriedade verdadeira de:

DK(EK(M)) = M

Alguns algoritmos podem usar chaves diferentes para encriptação e decriptação, cuja chave de encriptação seria *K1* diferente de sua chave de decriptação correspondente *K2*. Neste caso:

E*K1*(M) = C

D*K2*(C) = M

D*K2*(E*K1* (M)) = M

Toda a segurança nestes algoritmos é baseada em sua chave (ou chaves), nenhum é baseado em detalhes do algoritmo. Isto significa que o algoritmo pode ser publicado e analizado livremente.

Produtos que fizerem uso do algoritmo podem ser produzidos em massa pois não importará mais o fato do avaliador conhecer o algoritmo. Caso ele não conheça a chave correta usada no processo criptográfico, não será capaz de conhecer seu conteúdo.

2.4.4.1 Chaves simétricas No *Modelo Simétrico* a chave que embaralha a mensagem é a mesma que desembaralha. A segurança desse modelo restringe-se essencialmente ao segredo da chave, uma vez que os algoritmos criptográficos são conhecidos.

O tamanho da chave apenas dificulta a criptoanálise, não garantindo a segurança do modelo. O modelo é utilizado principalmente em aplicações onde existe somente um emissor e um receptor, como por exemplo, um terminal bancário ATM conectado com sua central de dados.

O motivo de sua aplicação restrita deve-se ao seu principal problema: a logística de distribuição de chaves.

Como a segurança do modelo é garantida somente pela segurança da chave, faz-se necessário a utilização de um canal de comunicação efetivamente seguro entre emissor e receptor, imune a qualquer intervenção de terceiros.

Como não havia até então uma forma de estabelecer um canal de comunicação realmente seguro, o modelo torna-se inviável para um grupo maior de indivíduos. Transmitir

a chave por e-mail, telefone, correio convencional ou mensageiro, por exemplo, tem seu significativo grau de interceptação.

A forma definitivamente mais segura seria o próprio interessado conduzir a chave até o destinatário.[SUL03]

2.4.4.2 Chaves assimétricas O Modelo Assimétrico (também conhecido como modelo de chave pública) utiliza duas chaves distintas para criptografar e decriptografar uma mensagem. Uma delas é de domínio público e a outra é secreta, somente de conhecimento de seu detentor.

Apesar de saber que a chave secreta é a função inversa f -1 (x) da chave pública, a sua determinação compreende um cálculo árduo. Os algoritmos assimétricos utilizam-se de problemas computacionalmente intratáveis ditos NP-Completos para garantir a segurança do modelo.

Todos os problemas computacionais pertencem a uma classe de complexidade NP (Nondeterministic Polinomial Time). São problemas que possuem pelo menos uma solução algoritmica capaz de verificar, dada uma solução para o problema, se esta efetivamente é uma solução para o problema.

Se o problema possui uma solução cujo tempo de resposta cresce polinomialmente em função do tamanho da entrada., então esse problema pertence a subclasse P (Polinomial Time). Na verdade, a terminologia subclasse não existe.

Foi utilizada apenas para esclarecer que os problemas Classe P estão contidos no conjunto de problemas Classe NP. Ou seja, todo problema Classe P também é, por definição, Classe NP. Os problemas NP ainda podem ser da subclasse NP-Completos.

Assim, todo problema NP-Completo também é um problema da Classe NP. Os NP-Completos são problemas que podem ser reduzidos polinomialmente a um Problema de Decisão. Em vez de procurar pelo tamanho máximo de um grafo, por exemplo, questiona-se a existência de um valor maior que x.[SUL03]

Resolvendo-se o problema de decisão, pode-se resolver o problema original. Porém, os Problemas de Decisão são capazes de responder a questão, mas para determinados casos,

como a fatoração por exemplo, calcular esses valores é uma tarefa computacionalmente árdua. [SHA97].

Assim, o problema NP-Completo da fatoração de inteiros também conhecido como *Função One-Way* não possui uma solução polinomial. Todavia, embora essa solução nunca tenha sido encontrada, também não foi possível provar que ela definitivamente não exista.

2.4.4.3 Funções *hash* de via única Este tipo de função é aplicada para a geração de resumo, com significado considerado similar a uma *impressão digital* (também chamada de *fingerprint*), de determinado documento para posterior conferência da autenticidade de seu conteúdo.

Seria basicamente a caracterização do recurso de *não-repúdio* que deve ser provido pela criptografia segundo *Schneider*.

De maneira objetiva, funções *hash* de via única são algoritmos, *H(M)*, que operam em mensagens de tamanho arbitrário, *M*, retornando um resumo criptográfico (*Hash*) de tamanho fixo *h*.[SCH96]

Considera-se que um documento originalmente escrito e gerado seu resumo *r*, desde que não seja alterado em nenhum momento, gerará um resumo idêntico ao original em uma posterior conferência, caso contrário, caracterizará que o documento foi alterado.

Ainda segundo *Schneider*, muitas funções podem manipular uma entrada de tamanho arbitrário e retornar uma saída de tamanho fixo, mas apenas as funções hash de via única possuem características adicionais que as fazem efetivamente de única via.[SCH96]

Fornecida a mensagem *M*, é fácil computar o resumo *h*.

Fornecido o resumo *h*, é dificil computar a mensagem *M* de tal forma que *H(M)= h*.

Fornecida a mensagem *M*, é dificil encontrar outra mensagem *M'*, de tal forma que *H(M) = H(M')*.

Em algumas aplicações, o fato de um algoritmo ter caminho único não é suficiente. Existe a necessidade de um requisito especial chamado *Resistência à Colisão*.

Esta resistência caracteriza-se pela dificuldade em se encontrar duas mensagens ran-

domicas, *M* e *M'*, de tal maneira que *H(M) = H(M)*.

Na obra de *Schneier* é demonstrado um tipo de ataque chamado *Birthday Attack* (Ataque de Aniversário), que é baseado na busca por dois indivíduos randomicos em uma amostra que possuam data de aniversário idênticas.

Na criptanalise de funções hash de via única, o resultado de resumos idênticos para testes em mensagens randomicas distintas é chamado de *Colisão*.

As funções *hash* são consideradas seguras enquanto não for identificada uma maneira de *forjar* sua *fingerprint* (encontrar uma *colisão* em mensagens distintas de maneira mais rápida de que em ataques de força-bruta.

2.4.4.4 Códigos de autenticação de mensagens *(MAC)* Um código de autenticação de mensagem, ou *MAC* (do inglês *Message Authentication Code*) é uma função *hash* de via única dependente de uma chave. Um *MAC* tem as mesmas propriedades de uma função *hash* de via única e além disso faz também uso de uma chave.

Apenas alguem com uma chave idêntica pode verificar o *fingerprint* gerado.

Um exemplo de aplicação clássica para uma função *MAC* seria onde um simples usuário aplicaria a mesma para determinar se seus arquivos foram ou não alterados, talves por um vírus ou outro agente qualquer.

O usuário poderia computar a *MAC* de seus arquivos e armazená-los em uma tabela.

Deve-se considerar neste exemplo então que, caso o usuário tivesse feito uso de uma função *hash* de via única, então o vírus poderia computar o mesmo valor *hash* após a infecção dos arquivos e substituir seus registros na tabela.

Considerando então uma função *MAC*, um vírus não seria capaz de realizar a mesma ação mencionada simplesmente porque ele, conceitualmente, não saberia a chave.

Avaliando as mudanças implementadas que caracterizam *os Códigos de Autenticação de Mensagens*, é possível afirmar que a diferença fundamental entre ambos é a segurança do processo de geração do resumo e sua posterior avaliação apenas por indivíduos que possuam sua chave.

Uma maneira simples de tornar uma função *hash* de via única em uma *MAC* seria

encriptar o valor *hash* com um algoritmo criptográfico de chave simétrica. Qualquer *MAC* pode se tornar uma função *hash* de via única simplesmente publicando sua chave. [SCH96]

2.4.4.5 Algoritmos *MD* e *SHA* Na obra *Criptografia Aplicada* de *Bruce Schneier*, os algoritmos escolhidos pelo autor como bons candidatos para a geração de *fingerprints* foram o *MD5* e o *SHA*, além das construções baseadas em cifras de blocos.

Os algoritmos da série *MD*(do inglês *Message Digest*) são funções *hash* de via única desenvolvidos por *Ron Rivest.*[SCH96]

Avaliando algoritmos mais recentes da série, o *MD4* por exemplo, produz um valor *hash* de 128-bits de uma dada mensagem qualquer de tamanho variável. Na apresentação desta concepção, Rivest descrevia como os objetivos de sua pesquisa o seguinte:

- *Segurança.* Onde deveria ser computacionalmente dificil encontrar duas mensagens diferentes que resultassem em um *hash* de mesmo valor. Nenhum ataque poderia ser mais eficiente que o de força bruta.

- *Segurança Direta.* A segurança do algoritmo *MD4* não é baseada em nenhuma suposição, assim como dificuldade de fatoração.

- *Velocidade.* O *MD4* é recomendado à implementações de *software* de alta velocidade. Ele é baseado em um conjunto simples de manipulação de bits com operadores de 32-bits.

- *Simples e Compacto.* O *MD4* é tão simples quanto o possível, sem largas estruturas ou programas complexos.

- *Favorecimento à Arquiteturas Menos Robustas.* O *MD4* foi otimizado para arquitetura de microprocessadores (especificamente à arquitetura *Intel* - grande fabricante de processadores atualmente); computadores mais robustos e rápidos podem fazer uso de qualquer tradução conforme necessidade.

Após a primeira introdução do algoritmo, *Bert den Boer* e *Antoon Bosselaers* conseguiram efetuar a criptanálise com sucesso dos últimas duas das três rodadas do algoritmo.

Em um resultado de criptanalise não relacionado, *Ralph Merkle* conseguiu realizar um ataque com sucesso das primeiras duas rodadas.

Eli Biham discutiu um ataque diferencial de criptanálise às primeiras duas rodadas do *MD4*.

Mesmo estes ataques não tendo abrangido toda a extensão do algoritmo, *Rivest* fortaleceu o algoritmo. O resultado foi o *MD5*.

Mesmo que mais complexo que o *MD4*, o *MD5* é similar em design e também produz resumos de 128-bits.

A descrição do *MD5*, após algum processamento inicial, processa a entrada de texto em blocos de 512-bits divididos em 16 sub-blocos de 32-bits cada.

A saída gerada do algoritmo é uma série de quatro blocos de 32-bits, que são concatenados para formar um simples valor *hash* de 128-bits.

NIST, em conjunto com o *NSA*, desenvolveram o *Secure Hash Algorithm (SHA)* para uso como padrão nacional para assinatura digital.

O conjunto é definido como padrões para *hash* seguro (do inglês *Secure Hash Standard - SHS*), *SHA* é o algoritmo usado por padrão neste esquema.[SCH96]

De acordo com o registro federal americano, o padrão para processamento de uma informações ferderais (do inglês *Federal Information Processing Standard - FIPS*) para (SHS) foi proposto de maneira a usar o *SHA* como padrão para assinaturas digitais.[SCH96]

Quando uma mensagem de qualquer tamanho menor que 264 bits for inserida, o *SHA* produzirá uma saída de 160-bits chamada *message digest*. O *message digest* será então inserido no *DSA* (algoritmo criptográfico), que computará a assinatura para a mensagem.[SCH96]

Assinar o *message digest* ao invez da mensagem aumenta a eficiencia do processo, porque o *message digest* é normalmente muito menor que a mensagem.

O mesmo *message digest* deve ser obtido pelo avaliador da assinatura quando a versão recebida da assinatura for entrada no *SHA*.

O *SHA* é considerado seguro por ter sido desenvolvido de maneira a ser computacionalmente difícil para se recuperar uma mensagem a partir de seu *message digest*,ou para se encontrar duas mensagens diferentes que produzam o mesmo *message digest*.

Qualquer mensagem que seja modificada em transito irá, com probabilidade extrema-

mente grande, resultar em *message digest* defirentes, e a assinatura falhará no momento da verificação

O *SHA* é baseado em princípios similares aqueles usados pelo Professor *Ronald L. Rivest* do *MIT* quando desenvolveu o algoritmo *MD4*.

A diferença substancial do *SHA* em relação ao *MD4* e *MD5* é o uso de chaves, caracterizando como *MAC*.

Em conclusão na mesma obra, o autor vota pelo uso do algoritmo *SHA*. O algoritmo tem um valor *hash* mais longo que o valor gerado pelo *MD5*, é mais rápido que a grande variedade de cifras de bloco da época e foi desenvolvido pela *NSA*.

Segundo *Schneier*, é possível confiar nas habilidades em criptanalise da *NSA*, mesmo considerando o fato da entidade não tornar publico seus resultados.

A tabela a seguir demonstra resultados de velocidade para algumas funções *hash* existentes. Estes resultados são demonstrados apenas para propósitos comparativos.

2.5 Criptoanálise

A criptoanálise é um conjunto de métodos para analisar mensagens cifradas com o objetivo de decifrá-las. Seria o lado contrário da criptografia onde a mesma é concebida como a ciência de quebrar códigos, desvendar segredos, violar esquemas de autenticação e efetuar a quebra de protocolos criptográficos.

De maneira a desenvolver algoritmos de criptografia robustos ou protocolos criptográficos, o interessado deverá usar um criptoanalista para encontrar quaisquer brechas em seu estudo.

Esta é exatamente a razão pela qual os melhores (mais confiáveis) algoritmos criptográficos são aqueles disponibilizados ao público. Por exemplo, o algoritmo *DES* foi exposto ao público por anos e se manteve confiável, enquanto o *Skipjack* ainda é um algoritmo secreto e considerado de pouca confiança.

Esta é uma premissa básica da criptologia onde a segurança de um algoritmo não deve ser baseado no segredo de seu funcionamento. Inevitavelmente, o algotítmo secreto será

Speeds of Some Hash Functions on a 33 MHz 486SX

Algorithm	Hash Length	Encryption Speed (kilobytes/second)
Abreast Davies-Meyer (with IDEA)	128	22
Davies-Meyer (with DES)	64	9
GOST Hash	256	11
HAVAL (3 passes)	variable	168
HAVAL (4 passes)	variable	118
HAVAL (5 passes)	variable	95
MD2	128	23
MD4	128	236
MD5	128	174
N-HASH (12 rounds)	128	29
N-HASH (15 rounds)	128	24
RIPE-MD	128	182
SHA	160	75
SNEFRU (4 passes)	128	48
SNEFRU (8 passes)	128	23

Figura 2.28. *Resultados de algumas funções* **hash** *em computador 486SX de 33Mhz.* [SCH96]

descoberto e as brechas (se houver) serão exploradas.

As várias técnicas em criptoanálise que buscam comprometer os criptosistemas são denomidadas *staques*. Alguns ataques são genéricos, outros são voltados para criptosistemas específicos, mas todos buscam sua quebra.[Lab98]

2.5.1 Ataques de força-bruta

Ataques de força-bruta são metodos para testar um grande número de combinações possíveis dentro de um determinado esquema criptográfico.

Nesta técnica, é realizado um processamento de maneira automática para realizar tentativas sequenciais de busca por todas as chaves ou resultados possíveis que venham, finalmente, decriptar uma mensagem.

Na maioria dos esquemas considerados seguros, a possibilidade teórica de um ataque de força-bruta é conhecida, mas sua realização é considerada computacionalmente inviável.

A inviabilidade computacional mencionada é normalmente pela quantidade de tempo total necessário para que tal tipo de ataque possa ser considerado bem sucedido, ou seja, o tempo total para que o ataque realize todas as tentativas possíveis.

2.5.2 Quebra de algoritmos

Segundo *Schneier*, é considerada a definição de *quebra* de um esquema criptográfico quando é encontrada uma maneira de ataque com resultados mais rápidos de que o ataque por *força-bruta.*[SCH96]

2.5.3 Segurança por obscuridade

Método de segurança que se baseia na falta de conhecimento por parte do avaliador da mensagem. Uma mensagem é considerada segura enquanto não for de conhecimento público a maneira correta para decodificá-la, não oferecendo nenhuma dificuldade de visualização posterior ou decorrente do emprego da técnica.

2.6 Princípios do correio eletrônico

A maneira como o correio eletrônico funciona atualmente envolve programas, protocolos e equipamentos que vão muito além do computador usado para composição, envio e recebimento das mensagens.

Os agentes de recebimento, agentes de transporte, servidores de armazenamento, servidores de repasse, protocolos *SMTP* e *POP3* são figuras permanentemente presentes na arquitetura de correio eletrônico e serão apresentados nesta seção.

2.6.1 Agente do usuário de correspondências - *MUA*

Os programas usados para enviar e receber mensagens de correio eletrônico (algumas vezes chamados apenas de *mailers*) são formalmente chamados de Agentes do Usuário de Correspondências (do inglês *Mail User Agent - MUA*).

Eles são desenvolvidos de maneira a prover interfaces convenientes para acesso às correspondências eletrônicas por parte de usuários comuns.

Exibem a chegada de mensagens que estejam armazenadas nas caixas privadas, ajudam o usuário a compor mensagens para envio e provêem facilidades para gerenciamento de pastas de mensagens recebidas gravadas. São considerados os *"front end"* (interfaces de usuário final) do sistema de correio eletrônico.[HAZ01]

Vários *MUAs* diferentes podem estar instalados, e podem ser simultaneamente operacionais em um simples computador provendo assim oportunidade de escolha por diferentes interfaces de usuário.[HAZ01]

Contudo, quando um *MUA* envia uma mensagem, ele não assume todo o trabalho de entrega efetiva da mensagem ao destinatário. Ao invéz disso, envia a mensagem ao *MTA* que pode estar executando na mesma maquina ou em um servidor qualquer.[HAZ01]

A escolha por qual *MUA* estará operando será sempre do usuário.[HAZ01]

2.6.2 Agente de transporte de correspondências - *MTA*

O serviço de um agente de transporte de correspondências (do inglês *Mail Transport Agent - MTA*) é receber mensagens de deferentes fontes e entregá-las à seus destinatários, potencialmente em um grande número de maneiras diferentes.[HAZ01]

Estes agentes realizam o trabalho de transferir as mensagens entre os servidores, e, após terem atingido seus servidores de destino, de entregá-las em caixas de correspondências privadas ou a processos de gerenciamento destas caixas. [HAZ01]

Este serviço é complicado e não pode ser dependente do *MUA* que contenha todo o aparato necessário. O fluxo dos dados de uma mensagem de seu emissor ao destinatário é demonstrada na figura a seguir.

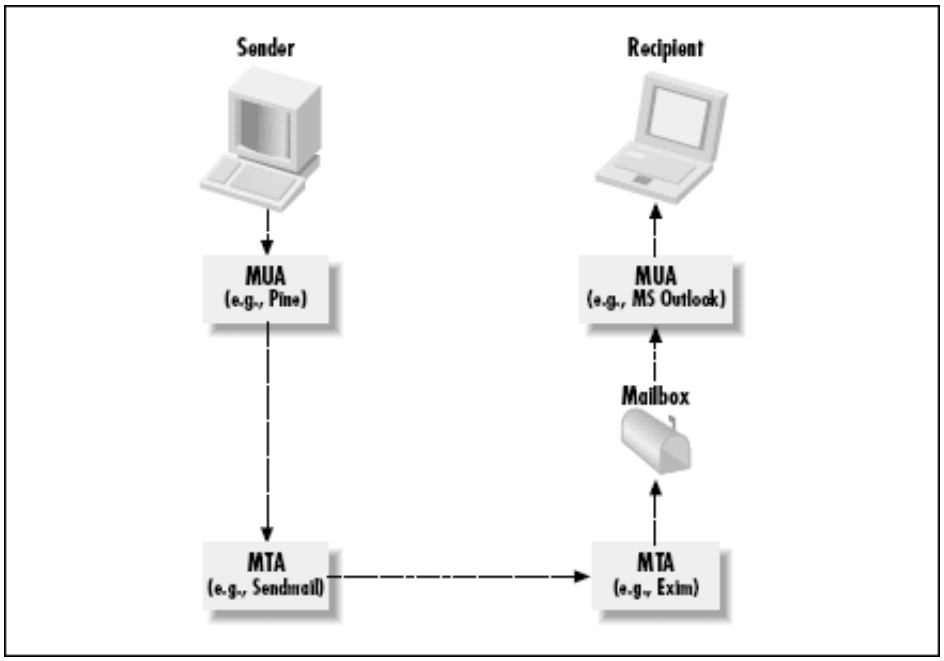

Figura 2.29. *Fluxo de dados de uma correspondência eletrônica.* [HAZ01]

De qualquer maneira, quando uma aplicação ou script precisa enviar uma mensagem como parte de alguma atividade automática, ele normalmente faz uma chamada direta ao *MTA* sem envolver um *MUA*.[HAZ01]

Apenas um *MTA* pode estar completamente operacional em um servidor ao mesmo tempo, devido ao fato de apenas um programa poder ser designado a receber mensagens que chegam de outros servidores.[HAZ01]

Deve ser um programa privilegiado de maneira a estar *ouvindo* por conexões *TCP/IP* de entrada na porta *SMTP* e hábil a escrever em caixas privadas de usuários. A escolha por qual *MTA* estará operando será sempre do administrador do sistema.[HAZ01]

2.6.3 Sumário do protocolo - *SMTP*

O chamado Protocolo Simples para Transferência de Correspondências (do inglês *Simple Mail Transfer Protocol - SMTP*) é o recurso padrão da pilha TCP/IP para transmissão de mensagens eletrônicas.[For01]

O *SMTP* é um protocolo simples de comando-resposta. A máquina cliente envia o comando ao servidor e então aguarda por sua resposta antes de enviar o próximo comando. As respostas sempre começam com um número decimal de três dígitos.[HAZ01]

Um exemplo seria:

250 Message accepted

O texto que segue a seqüência numérica é normalmente uma informação desenvolvida para interpretação humana, porém, existem algumas excessões. Existem séries de numeros que codificam o tipo de resposta. O primeiro dígito é o mais importante e segue conforme a tabela a seguir.

Code	Meaning
2*xx*	The command was successful
3*xx*	Additional data is required for the command
4*xx*	The command suffered a temporary error
5*xx*	The command suffered a permanent error

Figura 2.30. *Códigos de Resposta* **SMTP**.
[HAZ01]

O segundo dos três digitos fornece informações adicionais sobre a resposta, mas um *MTA* não precisa necessáriamente ter atenção a elas. O *MTA Exim*, por exemplo, opera totalmente baseado no primeiro digito da sequencia dos códigos de resposta *SMTP*.

Respostas podem ser compostas por várias linhas de texto. Para todos ou apenas para o último deles, o código é seguido é um hífem; na última linha é seguido por um espaço em branco. Por exemplo:

550-Host is not on relay list

550 Relaying prohibited by administrator

Quando um cliente conecta a porta de um servidor *SMTP* (porta 25), ele deve aguardar por uma resposta inicial antes de dar sequencia ao transito de comandos.

Alguns servidores incluem a identificação do *software SMTP* que estão executando (e até outras informações) na resposta, mas nada disto é requisito do protocolo. Outros servidores enviam uma resposta extremamente compacta, tal como:[HAZ01]

220 ESMTP Ready

O cliente então inicializa a seção enviando um comando *EHLO* (uma chamada equivalente à um "olá"extendido, do inglês *extended hello*), onde fornece seu próprio nome, por exemplo:

EHLO client.example.com

O protocolo *SMTP* original usava o comando *HELO* para a inicialização, e os servidores compatíveis com o protocolo atual ainda são obrigados a reconhecer este comando.[For01]

A diferença é que a resposta ao novo comando *EHLO* inclui uma lista de extensões *SMTP* que o servidor suporta.[For01]

Infelizmente, existem muitos *MTAs* em uso que estão potencialmente mal configurados, tanto acidentalmente quanto intencionalmente, de tal maneira que não enviam corretamente seu nome no comando *EHLO*.

Isto significa que os dados obtidos deste comando perdem completamente sua utilidade.

A resposta do servidor ao comando *EHLO* fornece o nome do servidor na primeira

linha, seguido opcionalmente por outro texto de informações, listando ao final as funções extendidas que o servidor *SMTP* suporta, por exemplo:[HAZ01]

250-server.example.com Hello client.example.com

250 SIZE 10485760

A resposta acima indica que o servidor em questão suporta a opção *SIZE* (do inglês "tamanho"), indicando um tamanho maximo para mensagem em 10,485,760 caracteres.[HAZ01]

Uma vez que o comando *EHLO* tenha sido aceito, o cliente pode dar início ao processo de envio de qualquer número de mensagens. Cada mensagem deve ser inicializada a partir do comando *MAIL*, que contem o envelope com o endereço do remetente.[HAZ01]

Se a opção *SIZE* for suportada pelo servidor, o tamanho da mensagem também pode ser fornecido neste momento, por exemplo:

MAIL FROM:<remetente@servidor.exemplo> SIZE=12345

Após ter sido aceito, cada endereço de destinatário deve ser transmitido em comandos *RCPT* separados da seguinte maneira:

RCPT TO: <destinatario@servidor.exemplo>

O cliente então aguardará por uma resposta para cada um antes de proceder com a sequencia.

O servidor pode aceitar alguns destinatários e rejeitar outros, tanto de maneira permanente quanto temporária. Após um erro permanente, o cliente não deve tentar enviar reenviar a mensagem para aquele endereço.[HAZ01]

Os motívos mais comuns para a rejeição são os seguintes:

- O endereço contém um domínio que é reconhecido como local para o servidor, porém, a parte local não pode ser identificada.[HAZ01]

- O endereço contem um domínio que não é local para o servidor, porém o cliente não é autorizado a fazer *relay* (usar o servidor para intercambio de mensagens com domínios externos) através daquele servidor para aquele domínio específico.[HAZ01]

Erros temporáios são causados por problemas que devem ser resolvidos em curto espaço de tempo, tais como a não possibilidade de verificação de endereços de chegada

devido a uma instabilidade do banco de dados ou falta de espaço em disco.[HAZ01]

Após um erro temporário, é esperado que o cliente tente se comunicar novamente dentro de alguns instantes com aquele endereço de destino a partir de uma nova conexão *SMTP*. Este tempo pode variar entre pelo menos 10 a 15 minutos após a primeira falha.[HAZ01]

Se a condição de erro temporário persistir, o período entre as novas tentivas de envio deve ser aumentado.

Assumindo que, ao menos, um destinatário tenha sido aceito, o cliente envia:

DATA

Para este comando o servidor responde com o código *354*, requisitando os possíveis dados, normalmente significa a mensagem a ser remetida em sí.[HAZ01]

O cliente transmite a mensagem completa sem aguardar por quaisquer respostas durante o procedimento, terminando com uma linha contendo apenas um simples caracter ponto (.).[HAZ01]

Se a mensagem contem quaisquer linhas que comecem com um simples ponto, deve ser inserido um ponto extra pelo cliente com a finalidade de evitar um encerramento prematuro da mensagem.

O servidor retira automáticamente um eventual primeiro caracter ponto de cada linha que contiver mais texto além dele.

Se o servidor retornar uma mensagem de sucesso após os dados serem enviados, ele assume a responsabilidade pelo tramite subsequente da mensagem e o cliente pode discartar a cópia de envio da mesma.

Uma vez enviadas todas as mensagens, o cliente encerra a sessão *SMTP* enviando um comando *QUIT* (significa "saír"em português).

Devido ao fato do protocolo *SMTP* transmitir os endereços separados da mensagem em sí, os servidores podem rejeitar alguns endereços individualmente, antes que muitos dados sejam transmitidos.

Porém, se um servidor não aceitar o conteúdo de uma mensagem devido a eventuais violações de regras de segurança e/ou privacidade em seu conteúdo, ele não poderá enviar

uma negação deste envio enquanto toda a mensagem não for transmitida.[HAZ01]

Infelizmente alguns *softwares* clientes (em violação ao *RFC 821* e consequentemente ao *RFC 2821*) trata qualquer resposta de erro ao comando *DATA* como temporários, e continua a tentar enviar aquela mesma mensagem em intervalos sucessivos.[HAZ01]

2.6.4 Falsificação de origem - *Forgery*

Segundo *Hazel*, é comum a forja (falsificação) de mensagens eletrônicas não criptografadas.[HAZ01]

Em geral, os *MTAs* são "estranhos"uns para os outros, desta forma, não existe uma maneira de a um *MTA* poder autenticar o conteúdo de uma mensagem em envelope que está recebendo.

Tudo que se pode fazer é registrar (do inglês *log* o endereço *IP* de um servidor que está enviando a correspondencia e então incluí-lo na linha de recebimento (do inglês *Received:*) que é adicionada à mensagem.[HAZ01]

Mensagem de propaganda não solicitada (*SPAM*) normalmente contém algumas linhas de cabeçalho forjadas. É necessário que o avaliador esteja consciente deste tipo de atitude se tiver que investigar a origem de tal correspondência.

Se uma mensagem contiver uma linha de cabeçalho como:

Received: from teste.com.exemplo ([10.3.2.1])
by luluzinha.edu.exemplo (8.9.1/8.9.1) with SMTP id DAA00447;
Tue, 6 Mar 2001 03:21:43 -0500 (EST)

Isto não quer dizer necessáriamente que a empresa *Teste* ou a entidade educacional *Luluzinha* estão envolvidos naquele processo de comunicação necessariamente.

Na realidade eles provavelmente apenas tiveram seus nomes inseridos na correspondência, intencionalmente, pelo emissor do *SPAM* com a finalidade de confundir seu rastreamento.

As únicas linhas de recebimento (*Received:*) que o avaliador poderá confiar serão aquelas emitidas no topo da mensagem que forem adicionadas pelos *MTAs* que são gerenciados por administradores de confiança.[HAZ01]

Após estas linhas, quaisquer outras poderão estar forjadas, mesmo se elas parece-

rem autenticas e/ou relacionadas à alguma empresa ou provedor de serviços da *Internet.*[HAZ01]

2.6.5 Autenticação e criptografia

O protocolo *SMTP* original não provia recursos de autenticação para os clientes do serviço ou para criptografia das mensagens que eram transmitidas entre os servidores.[HAZ01]

Na medida que a *Internet* foi crescendo, se tornou claro que estes recursos eram necessários e que o protocolho deveria ser expandido para permití-los.[HAZ01]

Porém, a grande maioria do fluxo de correspondência na *Internet* ocorre entre servidores não autenticados, sobre conexões não criptografadas.[HAZ01]

2.6.6 Visão geral do *DNS*

O serviço de nomes de domínio (do inglês *Domain Naming System - DNS*) é um banco de dados mundial distribuído que armazena vários tipos de dados indexados por chaves chamados nomes de domínio.[HAZ01]

Aqui temos um breve sumário dos recursos fornecidos que são relevantes para a manipulação de correspondencia eletrônica.

As informações são mantidas em unidades chamadas registros (do inglês *records*), cada um contendo um número de ítens.[HAZ01]

Os recursos mantidos pelo *DNS* são muitos conforme relacionados no *RFC3658*, os seguintes ítens são relevantes para aplicações que usam seus recursos.[For03]

<domain name> <record type> <type-specific data>

Por exemplo, o registro:

www.endereco.exemplo. A 10.3.2.1

O nome de domínio seria www.endereco.exemplo, o tipo do registro seria "A"(de endereço, do inglês *"address"*), e os dados específicos seriam 10.3.2.1.

Registros de endereços como estes são usados para encontrar o endereço IP dos

servidores a partir de seus nomes e são provavelmente o tipo mais comum de registro *DNS*.[HAZ01]

No uso do *DNS*,um nome de domínio completamente qualificado é sempre representado terminando com um ponto (.), como no exemplo anterior.

Nomes de domínio incompletos são representados sem o ponto ao final, isto representa que eles são então relativos à um domínio superior.[HAZ01]

Infelizmente, existe uma confusão devido a algumas aplicações que interagem com o *DNS* que não exibem ou exigem o ponto ao final.

Em particular, nomes de domínio em endereços de email não devem incluir este ponto, devido a isto ser contrário à referência de sintaxe originalmente definida nos *RFC 821/822*.[For82]

O esquema atual de endereçamento da *Internet* que usa endereçamentos de 32-bits é conhecido como *IPv4*, será substituído gradualmete por um novo esquema conhecido como *IPv6*, que utilizará endereçamentos de 128-bits.[HAZ01]

Dois tipos de registros *DNS* são usados atualmente para armazenar endereços *IPv6*, que são normalmente escritos com valores hexadecimais usando o simbolo (:) como separador.[For00]

O registro$AAAA$, que tem uma direção analoga ao registro A, foi definido primeiro. Por exemplo:[For00]

ipv6.exemplo. AAAA 5f03:1200:836f:0a00:000a:0800:200a:c031

Porém, considera-se um esquema mais flexível quando as porções de prefixos dos endereços IPv6 possam ser armazenadas separadamente. É considerado desta forma pois permite uma agregação e realocação mais fácil.[HAZ01]

Por esta razão, outro tipo de registro *DNS* foi criado, o *A6*. Espera-se que com o tempo ele supere em uso o tipo $AAAA$. O exemplo anterior pode ser convertido para um simples registro *A6* como este:[HAZ01]

ipv6.exemplo. A6 0 5f03:1200:836f:0a00:000a:0800:200a:c031

O valor zero indica que nenhum prefixo adicional é necessário. Modificando de forma nativa, o endereço poderá ter seu prefixo armazenado em um registro separado, como

este:[HAZ01]

ipv6.exemplo. A6 64 ::000a:0800:200a:c031 prefixo.exemplo.

prefixo.exemplo. A6 0 5f03:1200:836f:0a00::

O valor *64* significa que um prefixo adicional de 64 bits é necessário e que o nome de domínio *prefixo.exemplo* identifica outro registro *A6* onde este prefixo pode ser encontrado. Muitos níveis de prefixos são permitidos.[HAZ01]

Se um servidor tem mais de uma interface *IP*, cada uma aparecerá separadamente em registros de endereços distintos com o mesmo nome de domínio.

O caso de letras nos nomes de domínio *DNS* não são significantes, e os componentes individuais de um nome poderão conter uma grande faixa de caracteres. Por exemplo, um registro no *DNS* pode conter o nome abc$_x$*yz*2.*exemplo.com*.[HAZ01]

Porém, os caracteres usados para nomes de servidor são restritos conforme descrito no *RFC 952* para letras, digitos e hífens. Os domínios que são usados para endereçamento de correspondência eletrônica no protocolo *SMTP* são restritos de maneira similar.[For85]

2.6.7 Registros *DNS* usados para roteamento de *emails*

O nome de domínio em um endereço de email não precisa necessáriamente ser correspondente a um nome de servidor. Por exemplo, uma organização pode usar o domínio *abc.exemplo.com* para todos os seus emails, mas por outro lado pode gerenciar estes emails com servidores chamados *mail-1.abc.exemplo.com* e *mail-2.abc.exemplo.com*. Este tipo de flexibilidade é possível a partir do uso de registros *DNS* do tipo *MX* (do inglês *"Mail Exchange"*). Um registro *MX* mapeia um domínio para um servidor registrado como gerenciador de *emails* para aquele domínio, incluíndo valores de preferência. Podem existir qualquer número de registros *MX* para um domínio, e no momento em que um servidor *DNS* for pesquisado sobre tal retornará todos eles. Por exemplo:

abc.exemplo.com. MX 5 mail.abc.exemplo.com.

abc.exemplo.com. MX 7 mail-3.abc.exemplo.com.

abc.exemplo.com. MX 7 mail-5.abc.exemplo.com.

Representa três servidores para gerenciamento de email para *abc.exemplo.com*.

Os valores de preferência podem ser interpretados pela distância do destino, ou seja, quanto menor o valo, maior é sua preferência, como no exemplo temos *mail.abc.exemplo.com* como o registro de maior preferência a partir do valor menor de todos, *5*.[HAZ01]

Um *MTA* que esteja entregando correspondência para este domínio deverá tentar efetuar a entrega inicialmente para *mail.abc.exemplo.com*, se falhar, então tentará para os outros servidores presentes nos registros conforme o valor de sua ordem de preferência. [HAZ01]

É apenas usada uma ordem numérica de preferência. O número absoluto não importa. Quando existirem registros *MX* com valores de preferência idênticos (como no exemplo anterior), eles serão sorteados em ordem randomica antes de serem utilizados.[HAZ01]

Antes de um *MTA* fazer uso de uma lista de servidores que ele tenha obtido a partir de registros *MX* ele primeiro tenta encontrar o endereço *IP* dos servidores.[HAZ01]

Eles realizam esta tarefa a partir dos registros de endereçamento correspondentes (*A* para *IPv4*, e *AAAA* ou A6 para *IPv6*). Para o exemplo anterior, deverão existir os seguintes registros de endereços.[HAZ01]

mail.abc.exemplo.com. A 192.168.8.8

mail-3.abc.exemplo.com A 192.168.8.88

mail-5.abc.exemplo.com A 192.168.8.188

Na prática, se um nome de servidor já possui um registro de endereçamento para qualquer servidor em uma lista de *MX* que esteja retornando, ele envia o registro de endereço junto com os registros *MX*.

Em muitos casos isto ajuda a evitar pesquisas adicionais no *DNS*.[HAZ01]

Antigamente os *DNS* não possuiam registros *MX* e os endereços de *email* correspondiam aos nomes dos próprios servidores.[HAZ01]

Para manter a compatibilidade, se não houver nenhum registro *MX* para um domínio,

o *MTA* deverá pelo registro de endereçamento do domínio e interpretá-lo como se fosse um registro *MX* com um valor de preferência igual a xero (maior preferência).

Porém, se não for possível identificar se existem ou não registros do tipo *MX* para o domínio (porque, por exemplo, não foi possível entrar em contato com os servidores de nome *DNS*), o *MTA* não deverá agir desta forma.

Registros *MX* foram originalmente desenvolvidos para uso em *gateways* (equipamento de intercâmbio e roteamento de informações) para outros sistemas de correspondência.

Atualmente os registros *MX* são largamente utilizados para implementação de domínios corporativos que não especificamente signifiquem nomes de servidores específicos.[HAZ01]

Capítulo 3

Projeto

A estrutura do projeto está baseada na escolha, customização e utilização de um *MTA*, de maneira que o mesmo seja capaz de prover métodos de autenticação de mensagens eletrônicas com o intuito de confirmar ou não o envio de tais mensagens por parte do *MTA*.

A codificação será efetuada dentro do algoritmo do próprio *MTA*, caracterizando-se como alterações ao longo do código-fonte a fim de capacitar a aplicação a realizar tarefas de resumo das mensagens, armazenamento e pesquisa destes resumos.

Aplicativos para gerenciamento de bancos de dados serão incorporados ao conjunto de recursos da proposta para viabilizar o armazenamento e a pesquisa dos resumos e diminuir a quantidade de codificação e criação de métodos para tais finalidades.

3.1 Ferramentas

3.1.1 *Hardware*

Foram utilizados 2 (dois) equipamentos PC de configuração idêntica. Processador AMD Duron 1,6 Ghz, 256 Mb de memória usando um módulo de 256Mb DDR-400Mhz (PC-3200) da marca Elixir, 80Gb de espaço em disco rígido Seagate padrão IDE 7200RPM ATA 133.

Placa mãe MSI modelo KT4 Ultra-SR usando chipset Via KT400 com 400Mhz de FSB, unidade de disquete Sony com capacidade para 1,44Mb.

Unidade LG de leitura e gravação de mídias dos tipos CD/CD-R/CD-RW com velocidade nominal de 52x32x52x e leitura de mídias DVD com velocidade nominal de 16x, 6 interfaces de comunicação USB padrão 2.0 controladas pelo chipset.

Interface ethernet Realtek integrada à placa-mãe com velocidade nominal de 10/100Mbps, placa de vídeo Pine padrão AGP com 64Mb DDR-266Mhz (PC-2100) com memória própria e GPU Geforce 2 / MX400.

Controlador de áudio C-Media CMI8738/C3DX integrado à placa-mãe com capacidade de reprodução de áudio no padrão Dolby Surround 5.1.

Os equipamentos aqui descritos foram adquiridos em fevereiro de 2003.

3.1.2 *Software*

3.1.2.1 Sistema operacional Para a implementação e testes dos recursos presentes nesta pesquisa foi utilizado o sistema operacional *Linux*, distribuição *Debian* na versão *Sarge 3.1*, lançado oficialmente em 06/06/2005.

3.1.2.2 Aplicação *MTA* A aplicação para serviço de coleta e entrega de correspondencia eletrônica escolhida foi o *MTA Exim* Versão 4.

3.1.2.3 Compilador Para a geração de produto binário executável foi utilizado nesta proposta o *GNU Compiler Collection (GCC)*, versão 3.3.5.

3.1.2.4 Serviços de *DNS* O *software* para fornecimento de respostas a nomes de domínio, *DNS*, foi o *BIND* versão 9.

3.1.2.5 Gerenciador de bancos de dados Para o gerenciamento, armazenamento e pesquisa dos resumos das mensagens enviadas e recebidas pelo *MTA* foi utilizado o servidor de banco de dados *MySQL* versão estável 4.0.24.

3.1.2.6 Interpretador gráfico Para a interpretação de aplicações em ambiente gráfico foi empregado o *software XFree86* Versão 4.3.0.

3.1.2.7 Ambiente *desktop* Para o gerenciamento de recursos da área de trabalho gráfica foi utilizado o *GNOME desktop environment* versão 2.8.

3.1.2.8 Agente de emails O *software Mozilla Thunderbird*, versão 1.0.2, foi utilizado para a avaliação do funcionamento de envio, registro do resumo criptográfico, entrega e autenticação das mensagens eletrônicas dentro do ambiente da pesquisa.

3.1.2.9 Editor de textos Para auxílio à codificação e pesquisa de código fonte foi utilizado o *VIM*, versão 6.3.

3.1.3 Linguagens, algoritmos e protocolos

3.1.3.1 Linguagem de programação Foi escolhida para codificação e modificação do código fonte do *MTA Exim* a linguagem de programação *C*.

3.1.3.2 Algoritmo para resumo criptográfico Para a geração de resumos criptográficos foi escolhido o algoritmo *MD5*.

3.1.3.3 Protocolo de comunicação e pesquisas O protocolo escolhido para comunicação foi o *SMTP*, o qual recebeu as adições para a etapa de consulta e resposta às pesquisas de autenticidade de envio.

3.2 Decisões de projeto

3.2.1 *Hardware*

Para a escolha do *hardware* foi definido como ideal qualquer configuração considerada suficiente para a completa elaboração da pesquisa, de ampla oferta no mercado e custo médio entre os demais equipamentos similares disponíveis.

O objetivo destas características para escolha buscam demonstrar a viabilidade econômica de acesso aos equipamentos envolvidos, facilitando assim a futura implementação da pesquisa em cenário real, não apenas em laboratório.

3.2.2 *Software*

3.2.2.1 Sistema operacional A escolha do sistema operacional levou em consideração características como estabilidade, suporte e filosofia.

O conceito de *software* livre (livre como em liberdade) difundido pela *FSF (Free Software Foundation)*, a partir do projeto *GNU*, foi de extrema importância na escolha do sistema operacional e de todos os componentes de *software* presentes nesta pesquisa.

O suporte à liberdade de expressão, mídia, recursos da *Internet* e o correto uso de *software* para criptografia voltado à comunicação privada da maneira como é feito pelo projeto *GNU* teve significado essencial para a viabilização desta pesquisa.

3.2.2.2 Aplicação *MTA* O *MTA Exim* foi escolhido para esta pesquisa por prover código fonte disponibilizado sob a *GNU/GPL*. Sua estruturação simples, boa documentação e abordagem direta por obras da área científica facilitaram sua compreesnão e adequação às necessidades.

3.2.2.3 Compilador O compilador *GCC* foi escolhido para esta pesquisa por prover código fonte disponível sob a *GNU/GPL* e por possuir inúmeras fontes de pesquisa para seu estudo e compreensão.

3.2.2.4 Serviços de *DNS* O *BIND* versão 9 foi escolhido para serviços *DNS* por ter ampla aplicação no mercado, alta disponibilidade, boa estabilidade e ampla gama de informações disponíveis para estudo em sua documentação básica e da *Internet*.

3.2.2.5 Gerenciador de bancos de dados O *MySQL* foi escolhido para esta pesquisa por disponibilizar seu código fonte sob a *GNU/GPL* e por possuir grande base instalada, o que foi considerado como sinônimo de praticidade, desempenho e estabilidade.

3.2.2.6 Interpretador gráfico O interpretador gráfico *XFree86* foi escolhido por possuir ampla distrbuição, grande número de aplicativos e ambientes *desktops* e por possuir código fonte sob a *GNU/GPL*.

3.2.2.7 Ambiente *desktop* O gerenciador *GNOME desktop environment* foi escolhido por ser o desktop padrão de várias distribuições *Linux*, incluindo a *Debian*, por possuir boa documentação e por possuir código fonte disponibilizado sob a *GNU/GPL*.

3.2.2.8 Agente de emails O *software Mozilla Thunderbird* foi escolhido por posuir bom gerenciamento de correspondência eletrônica, interface gráfica considerada amigável e possuir seu código fonte disponível sob a *GNU/GPL*.

3.2.2.9 Editor de textos Os recursos para edição de texto *VIM* foram escolhidas para esta proposta por apresentarem seu desenvolvimento voltado para o uso como ferramenta de programação, interface amigável e por possuir seu código fonte disponível sob a *GNU/GPL*.

3.2.3 Linguagens, algoritmos e protocolos

3.2.3.1 Linguagem de programação A linguagem de programação *C* foi escolhida por ser a linguagem padrão para desenvolvimento de aplicativos no ambiente da pesquisa e por ser também a linguagem usada no desenvolvimento do *MTA Exim*.

3.2.3.2 Algoritmo para resumo criptográfico A escolha do algoritmo para geração de resumos das mensagens *MD5* foi feita buscando dois objetivos:

Primeiro por se tratar de um algoritmo comum que não faz uso de chaves que podem ou devam ser trocadas entre os sistemas que venham a fazer uso da técnica pesquisada. Ocorrendo desta maneira, não existirá a necessidade da troca de chaves entre os equipamentos envolvidos.

Segundo por ser eficiente na geração de resumos únicos e rápidos para mensagens.

Foi avaliado, durante esta pesquisa, a existência de vários registros vindos de fontes notórias que atestam a existência de algoritmos capazes de provocar *colisões* no *MD5*, com maior eficiência que a força bruta.

Levando este fato em consideração, foi adicionado um recurso para provocar uma *expiração* dos resumos gerados a fim de eliminar registros da base de dados em períodos

menores do que os atingidos pelas informações obtidas, conforme [Sch05].

3.2.3.3 Protocolo de comunicação e pesquisas O protocolo para comunicação *SMTP* foi escolhido por ser o protocolo padrão para transferência de correio eletrônico. Seu uso e customização, para adicionar novos recursos, facilitou a elaboração dos componentes necessários à edificação deste trabalho.

3.3 Estrutura

3.3.1 Customizações no *MTA*

O *MTA Exim* teve que sofrer algumas modificações em seu código fonte para ser capaz de operar, de maneira adequada, à esta pesquisa. As modificações se caracterizaram como adições ao longo de sua codificação como segue.

3.3.1.1 Métodos pós envio pelo *MUA* A etapa de recepção inicial das mensagens pelo *MTA* foi a primeira a ser adaptada.

No momento em que o *MUA* conclui o envio de seu envelope para o *MTA*, após todas as validações usuais do *Exim*, o mesmo faz uma chamada para o novo método que criará um resumo *MD5* para a mensagem recebida, contendo os campos considerados chave.

Os campos considerados chave são: remetente, destinatário e o corpo da mensagem propriamente dita.

Após esta chamada, o *MTA* fará chamada a outro novo método que acionará o *MySQL* com a finalidade de adicionar o resumo obtido em um novo registro com data e hora de criação.

Neste momento, a mensagem estará pronta para envio, passando apenas pelas ava-

liações finais sobre destino e recursos disponíveis naquele momento por parte do *MTA*, como decisões sobre enfileiramento ou envio imediato.

3.3.1.2 Métodos pós recepção pelo *MTA* do destinatário

A etapa de recepção das mensagens pelo *MTA*, quando vinda de remetentes externos, foi a próxima a receber adaptadações.

No momento em que o *MTA* receber a mensagem pelas rotinas que tratam correspondência externa ao domínio, o mesmo efetuará a etapa para geração de resumo análoga à realizada pelo *MTA* emissor.

Esta etapa é importante por ser capaz de gerar o resumo com os campos de interesse e conceitualmente assume ser capaz de gerar exatamente o mesmo resumo gerado pelo *software* responsável pela recepção da mensagem original.

Caso a mensagem sofra qualquer tipo de alteração nos campos considerados aqui como chave ao longo de seu percurso, esta etapa gerará potencialmente um resumo diferente do gerado pelo *MTA* emissor, o que provocará uma falha de autenticação na etapa a seguir.

3.3.1.3 Métodos de pesquisa pelo *MTA* do destinatário

A etapa de recepção das mensagens pelo *MTA* sofreu mais alterações no sentido de, após a geração de resumo da mensagem recebida, ser capaz de questionar os servidores, via registros *MX* oficiais do domínio remetente, com recursos adicionais ao protocolo *SMTP*.

Fazendo uso da porta 25, o *MTA* do destinatário estabelecerá conexão com um ou mais servidores *MX* oficiais do domínio do remetente seqüêncialmente e enviará o comando *HASHAUTH*.

3.3.1.4 O comando *HASHAUTH - (Hash Authentication)*

O *MTA* foi preparado com a adição deste comanto ao conjunto já padronizado do protocolo *SMTP*.

A sintaxe deste comando é ***HASHAUTH ¡md5 hash¿***, como no exemplo:

textbfHASHAUTH 4c42fa4886bc0fff9c28ee1ed8255ba7

O resumo *MD5* é usado como chave no banco de dados.

Sendo esta a etapa mais importante da interação entre os servidores, este comando foi adicionado com o intuito de viabilizar a confirmação de autenticidade da mensagem questionada a partir de seu resumo.

Com uma busca bem sucedida no banco de dados, o *MTA* que recebeu o comando é capaz de responder com o código *250*, significando o sucesso desta pesquisa, ou seja, a mensagem do qual o resumo foi obtido realmente foi emitida por aquele servidor.

Caso a mensagem não seja encontrada, o *MTA* questionado responderá com o código *550*.

A pesquisa foi implementada assumindo como negação qualquer código de resposta diferente de *250*. Tal implementação teve a intenção de prover suporte aos *MTAs* que não tiverem o recurso implementado, assumindo como mensagem não autenticada.

3.3.1.5 Marcação da mensagem pelo *MTA* do destinatário

Após o recebimento do resultado da pesquisa, o *MTA* do destinatário adiciona uma nova linha especial ao cabeçalho da mensagem para posterior avaliação por regras a serem adicionadas pelo *MUA* do destinatário.

Caso o resultado da pesquisa seja *250* (positivo), o *MTA* adicionará a linha ***Confirmation-Result:Positive*** ao cabeçalho da mensagem.

Caso o resultado da pesquisa seja qualquer outro diferente de *250*, o *MTA* considerará a mensagem como não autenticada e adicionará a linha ***Confirmation-Result:Negative*** ao cabeçalho da mensagem.

Capítulo 4

Implementação

O conteúdo deste trabalho foi implementado usando as ferramentas e métodos conforme descrito no capítulo do projeto. Será apresentado a seguir a representação do funcionamento do trabalho considerando dois cenários, um considerado clássico (onde o remetente faz o envio e a mensagem é transferida via *SMTP* até o seu destinatário) e outro considerado malicioso (onde um remetente tenta se passar por outro forjando seu endereço de email).

4.1 Representações topologicas

4.1.1 Representação da topologia em cenário padrão

É demonstrado na figura a seguir a representação do cenário de atuação padrão do algoritmo.

Neste cenário a pesquisa será responsável pela criação e gerenciamento das assinaturas das mensagens enviadas pelo servidor do remetente, disponibilizando resposta para a pesquisa do servidor do destinatário a partir do resumo *MD5*.

1 - Representa o momento em que o remetente envia sua mensagem para o seu *MTA*, via protocolo *SMTP*.

2 - Representa o momento em que o *MTA* do remetente pesquisa os servidores *DNS* para identificar o(s) servidore(s) / rota para o envio da mensagem recebida de seu remetente.

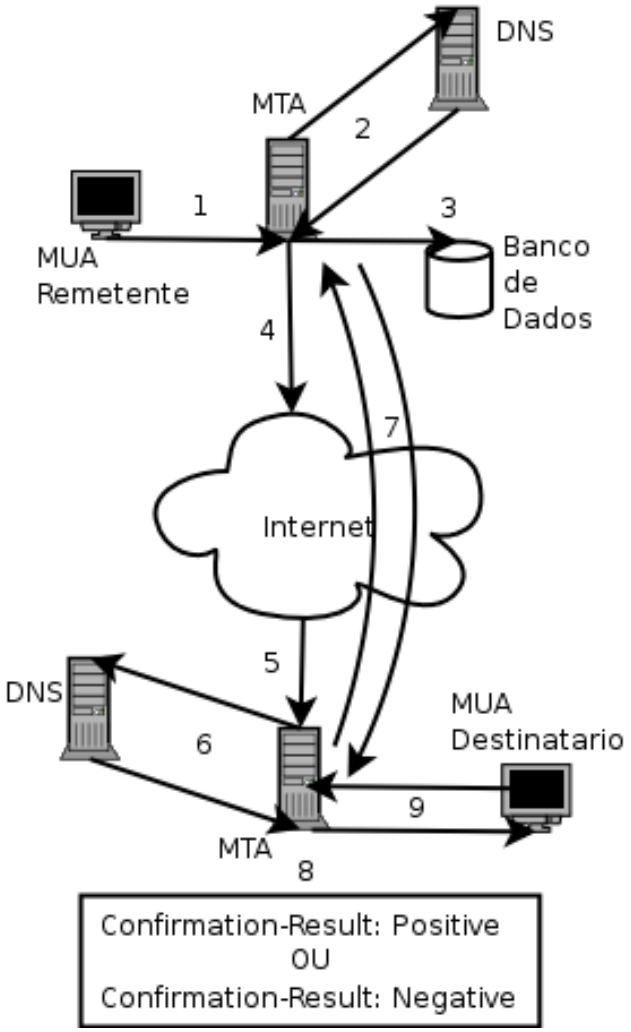

Figura 4.1. *Cenário Padrão.*

3 - Representa a primeira característica exclusiva desta pesquisa sendo aplicada.

Com identificação de ação tracejada em vermelho, este é o momento em que o algoritmo prepara o resumo *MD5* obtido a partir da combinação dos campos chave da mensagem, gravando o mesmo no banco de dados MySQL local.

O uso de um servidor de banco de dados externo é importante quando se levar em consideração o uso de vários *MTAs* para envio, onde os mesmos podem fazer uso compartilhado dos registros armazenados neste servidor.

4 - Representa o momento em que o *MTA* do remetente finalmente envia a mensagem para o servidor do destinatário.

5 - Representa o momento em que o *MTA* do destinatário recebe efetivamente a mensagem de correio eletrônico.

6 - Representa o momento em que o *MTA* do destinatário pesquisa os servidores *DNS* para identificar qual ou quais são os servidores *MX* oficiais do domínio da qual a mensagem diz ter sido enviada.

7 - Representa a segunda característica exclusiva desta proposta, sendo colocada em prática.

Também com a identificação de ação tracejada em vermelho, representa o momento em que o *MTA* de destino, já de posse dos *MXs* oficiais do domínio do qual a mensagem diz ter sido originada e busca pela resposta referente à confirmação de envio da mensagem por qualquer um desses *MXs* oficiais.

O traço liga diretamente o *MTA* de destino com o *MTA* de origem apenas para representar que a busca já tem endereço certo, contudo, a comunicação ocorre respeitando todas as formas de tráfego, roteamento e exceções que podem ocorrer no ambiente da *Internet*.

A seta com dois sentidos representa tanto a pesquisa do MTA destinatário com o MTA remetente, quanto a resposta, positiva ou negativa, do MTA remetente para o MTA destinatário.

8 - Representa a terceira característica exclusiva desta proposta.

Este é o momento em que o *MTA* do destinatário recebe a resposta do *MTA* de origem, o mesmo deve anexar um cabeçalho no envelope que indicará o resultado do processo de confirmação do remetente.

Este cabeçalho tem a finalidade de servir como um recurso confiável a ser usado pelo administrador do *MTA* ou até mesmo pelo *MUA* do destinatário quanto à classificação ou eventual descarte da correspondência.

9 - Representa o momento em que o *MUA* do destinatário recolhe a mensagem.

4.1.2 Representação da topologia em cenário malicioso

É demonstrado na figura abaixo a representação do cenário de atuação maliciosa do algoritmo, onde o mesmo é responsável pela pesquisa das assinaturas das mensagens recebidas pelo servidor do destinatário a partir do resumo *MD5*.

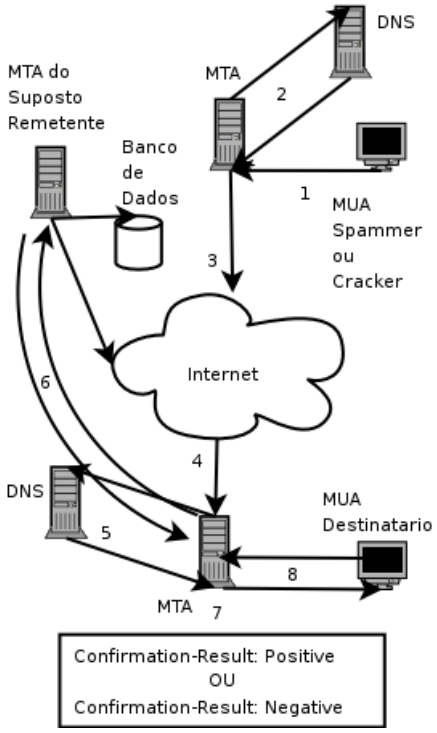

Figura 4.2. *Cenário Malicioso.*

1 - Representa o momento em que o remetente malicioso *(spammer, cracker ou outro)* envia sua mensagem com origem forjada (se fazendo passar por outra pessoa em outro domínio) para o seu *MTA* via *SMTP*.

2 - Representa o momento em que o *MTA* do remetente pesquisa os servidores *DNS* para identificar o(s) servidore(s) / rota para o envio da mensagem recebida de seu remetente.

3 - Representa o momento em que o *MTA* do remetente envia a mensagem com origem forjada para o servidor do destinatário.

4 - Representa o momento em que *MTA* do destinatário recebe efetivamente a mensagem de correio eletrônico.

5 - Representa o momento em que o *MTA* do destinatário pesquisa os servidores *DNS* para identificar qual ou quais são os servidores *MX* oficiais do domínio da qual a mensagem diz ter sido enviada.

6 - Representa com identificação de ação tracejada em vermelho, o momento em que o *MTA* de destino, já de posse dos MXs oficiais do domínio do qual a mensagem diz ter sido originada, busca pela resposta referente à confirmação de envio da mensagem por qualquer um desses *MXs* oficiais.

O traço liga diretamente o *MTA* de destino com o *MTA* oficial da suposta origem apenas para representar que a busca já tem endereço certo, contudo, a comunicação ocorre respeitando todas as formas de tráfego, roteamento e exceções que podem ocorrer no ambiente da *Internet*

A seta com dois sentidos representa tanto a pesquisa do *MTA* destinatário com o *MTA* remetente, quanto a resposta, positiva ou negativa, do *MTA* remetente para o *MTA* destinatário.

7 - Representa o momento em que o *MTA* do destinatário recebe a resposta do *MTA* de origem, o mesmo deve anexar um cabeçalho no envelope que indicará o resultado do processo de confirmação do remetente.

Este cabeçalho tem a finalidade de servir como um recurso confiável a ser usado pelo administrador do MTA ou até mesmo pelo MUA do destinatário quanto à classificação ou eventual descarte da correspondência.

Neste cenário, o algoritmo deve retornar um cabeçalho com o resultado negativo, pois o resumo *MD5* que ele busca não deverá estar presente no *MX* oficial, uma vez que a mensagem não foi emitida pelo mesmo.

8 - Representa o momento em que o *MUA* do destinatário pode recolher a mensagem.

4.2 Resultados

A adequação dos requisitos da pesquisa, assim como sua devida implementação diretamente no código-fonte do *MTA* escolhido apresentaram acréscimo do tamanho em Kb inferior a 1% em relação ao binário executável oficial.

O desempenho foi afetado de maneira um pouco mais significante, apresentando capacidade de manipulação de mensagens 10% inferior em relação à compilação oficial sem tais recursos.

Estima-se que o acréscimo de tamanho e a pequena perda de desempenho apresentada seja suportável para o tráfego de informações em cenário real, devido ao fato de outras soluções também resultar em situações similares.

O retorno em termos de segurança que a pesquisa busca prover também é considerada um fator positivo em relação ao desempenho atingido.

O uso do algoritmo *MD5* apresentou desempenho dentro do esperado, contribuíndo para os resultados apresentados neste capítulo.

O uso de outros algoritmos de maior complexidade podem ser facilmente adaptados à proposta, contudo, é importante que o custo computacional seja avaliado levando-se em consideração a demanda de entrega de mensagens em cenário real.

A perda de desempenho da solução é diretamente proporcional ao custo computacional da geração de resumos para as mensagens, dentre outros fatores.

Capítulo 5

Conclusões

Devido aos resultados apurados durante a etapa de implementação, estima-se que a pesquisa seja viável para implementação em cenário real.

Sua transparência e potencial aplicabilidade é estimada, principalmente, devido ao fato de ter sido desenvolvida dentro dos padrões usados atualmente e também por estar preparada para interação com outras soluções que não estejam prontas para os recursos desenvolvidos.

5.1 A pesquisa como solução

Este trabalho buscou contribuir para o processo de desenvolvimento das tecnologias voltadas para a segurança da informação, servindo como um acréscimo às combinações e esforços empreendidos para que os resultados positivos neste aspecto sejam cada vez maiores.

5.2 Trabalhos futuros

A avaliação dos resultados e recursos estudados nesta pesquisa aplicados em cenário real, assim como a comparação de seu desempenho em relação a outras pesquisas que busquem o mesmo objetivo de autenticação de emissão das mensagens eletrônicas dariam seqüência a este trabalho.

Uma comparação neste nível ajudaria a definir soluções mais adequadas ao crescente

volume de informações que trafegam diariamente na *Internet*, assim como agilizar o processo de autenticação e padronização dos serviços neste sentido.

O estudo e a implementação de novos algoritmos para elaboração de resumos criptográficos seria igualmente importante devido ao rápido avanço das técnicas de criptoanálise.

O emprego de algoritmos para autenticação de mensagens *(MAC)* com complexidade superior ao *SHA-1* (como o *SHA-256*) e o estudo de técnicas para troca segura de suas chaves seria importante para a confiabilidade e segurança dos resultados.

Referências Bibliográficas

[CdE04] Resposta e Tratamento de Incidentes de Segurança no Brasil CERT Centro de Estudos. Incidentes reportados ao cert.br – julho a setembro de 2004, 2004. http://www.cert.br/stats/incidentes/2004-jul-sep/total.html. Acesso em: 23 set 2005.

[CdE05] Resposta e Tratamento de Incidentes de Segurança no Brasil CERT Centro de Estudos. Incidentes reportados ao cert.br – abril a junho de 2005, 2005. http://www.cert.br/stats/incidentes/2005-apr-jun/total.html. Acesso em: 23 set 2005.

[For82] Internet Engineering Task Force. Rfc 822 - standard for arpa internet text messages, 1982. Disponível em: http://www.ietf.org/rfc/rfc822.txt. Acesso em: 24 set 2005.

[For85] Internet Engineering Task Force. Rfc 952 - dod internet host table specfication, 1985. Disponível em: http://www.ietf.org/rfc/rfc952.txt. Acesso em: 25 set 2005.

[For00] Internet Engineering Task Force. Rfc 2874 - dns extensions to support ipv6 address aggregation and renumbering, 2000. Disponível em: http://www.ietf.org/rfc/rfc2874.txt. Acesso em: 25 set 2005.

[For01] Internet Engineering Task Force. Rfc 2821 - simple mail transfer protocol, 2001. Disponível em: http://www.ietf.org/rfc/rfc2821.txt. Acesso em: 24 set 2005.

[For03] Internet Engineering Task Force. Rfc 3658 - delegation signer (ds) resource record (rr), 2003. Disponível em: http://www.ietf.org/rfc/rfc3658.txt. Acesso em: 25 set 2005.

[GAL45] Joseph S. GALLAND. *An Historical and Analytical Bibliography of the Literature of Cryptology.* Northwestern University Press, Evanston, IL, 1945.

[GAR95] Simson GARFINKEL. *PGP: Pretty Good Privacy.* O'Reilly Associates, Inc., Evanston, IL, 1995.

[HAZ01] Phillip HAZEL. *Exim, The Mail Transfer Agent.* OReilly Associates, Inc., Sebastopol, CA, 2001.

[JOH00] Kevin JOHNSON. *Internet Email Protocols: A Developer's Guide.* Addison-Wesley Professional, Boston, 2000.

[KAH67] David KAHN. *The Codebreakers.* Macmillan, New York, 1967.

[Lab98] RSA Laboratories. *RSA Laboratories Frequently Asked Questions About Todays Cryptography, v4.0.* RSA Data Security. Inc., California, 1998.

[MIS04] Mehran MISAGHI. Segurança e auditoria em informática, 2004. Disponível em: http://www.vision.ime.usp.br/ mehran/ensino/20042.html. Acesso em: 20 Set 2005.

[oST05] NIST National Institute of Standards and Technology. Information for researchers., 2005. Disponível em: http://www.nist.gov/publicaffairs/researchers.htm. Acesso em: 23 Set 2005.

[SCH96] Bruce SCHNEIER. *Applied Cryptography.* John Wiley Sons, Inc., New Jersey, 1996.

[Sch05] Bruce Schneier. The md5 defense, 2005. http://www.schneier.com/blog/archives/2005/08/the-md5-defense.html. Acesso em: 10 out 2005.

[SHA97] Clifford A. SHAFFER. *A practical introduction to data structures and algorithm analysis.* Prentice-Hall, Upper Saddle River, 1997.

[SIN00] Simon SINGH. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography.* Anchor Books, New York, 2000.

[SUL03] Jaime André SULZBACH. Análise de viabilidade da criptografia quântica, 2003. Disponível em: http://galileo.unisinos.br/alunos/arquivos/TCJaimeSulzbach.pdf. Acesso em: 21 Set 2005.

[TAN03] Andrew S TANENBAUM. *Redes de Computadores.* Campus, Rio de Janeiro, 2003.