Criptografia
em Software e Hardware

Edward David Moreno
Fábio Dacêncio Pereira
Rodolfo Barros Chiaramonte

Novatec Editora

Capítulo 1

Conceitos de Segurança de Dados e Criptografia

Neste primeiro capítulo, apresenta-se e enfatiza a necessidade da segurança de dados nos tempos modernos, focalizando o papel dos algoritmos de criptografia e as vantagens de se ter conhecimento dos conceitos, implementação e, principalmente, do seu desempenho.

1.1 Criptografia

A criptografia pode ser entendida como um conjunto de métodos e técnicas para cifrar ou codificar informações legíveis por meio de um algoritmo, convertendo um texto original em um texto ilegível, sendo possível mediante o processo inverso recuperar as informações originais (SIMON, 1999). Veja o processo na figura 1.1:

Figura 1.1 – Esquema geral para cifragem de um texto.

Pode-se criptografar informações basicamente por meio de códigos ou de cifras. Os códigos protegem as informações trocando partes destas por códigos predefinidos. Todas as pessoas autorizadas a ter acesso a uma determinada informação devem conhecer os códigos utilizados.

As cifras são técnicas nas quais a informação é cifrada por meio da transposição e/ou substituição das letras da mensagem original. Assim, as pessoas autorizadas podem ter acesso às informações originais conhecendo o processo de cifragem. As cifras incluem o conceito de chaves, que será apresentado na seção 1.2.

Os principais tipos de cifra são as de transposição, que são uma mistura dos caracteres da informação original (por exemplo, pode-se cifrar a palavra “CRIPTOGRAFIA” e escrevê-la “RPORFACITGAI”) e as cifras de substituição, que por meio de uma tabela de substituição predefinida trocam ou substituem um caractere ou caracteres de uma informação.

1.1.1 Breve História da Criptografia

A criptografia é tão antiga quanto a própria escrita, visto que já estava presente no sistema de escrita hieroglífica dos egípcios. Os romanos utilizavam códigos secretos para comunicar planos de batalha. Com as guerras mundiais e a invenção do computador, a criptografia cresceu incorporando complexos algoritmos matemáticos.

A criptologia faz parte da história humana porque sempre houve fórmulas secretas e informações confidenciais que não deveriam cair no domínio público ou na mão de inimigos.

Segundo Kahn (1967), o primeiro exemplo documentado da escrita cifrada relaciona-se aproximadamente ao ano de 1900 a.C, quando o escriba de Khnumhotep II teve a idéia de substituir algumas palavras ou trechos de texto. Caso o documento fosse roubado, o ladrão não encontraria o caminho que o levaria ao tesouro e morreria de fome perdido nas catacumbas da pirâmide.

Em 50 a.C, Júlio César usou sua famosa cifra de substituição para cifrar (criptografar) comunicações governamentais. Para compor seu texto cifrado, César alterou letras desviando-as em três posições; A se tornava D, B se tornava E etc. Às vezes, César reforçava seu método de criptografar mensagens substituindo letras latinas por gregas. O código de César é o único da Antigüidade que é usado até hoje. Atualmente qualquer cifra baseada na substituição cíclica do alfabeto denomina-se código de César. Apesar da sua simplicidade (ou exatamente por causa dela), essa cifra foi utilizada pelos oficiais sulistas na Guerra de Secessão americana e pelo exército russo em 1915.

Em 1901, iniciou-se a era da comunicação sem fio. Apesar da vantagem de uma comunicação de longa distância sem o uso de fios ou cabos, o sistema é aberto e aumenta o desafio da criptologia. Em 1921, Edward Hugh Hebern fundou a Hebern Electric Code, uma empresa produtora de máquinas de cifragem eletromecânicas baseadas em rotores que giram a cada caractere cifrado (TKOTZ, 2003).

Entre 1933 e 1945, a máquina Enigma que havia sido criada por Arthur Scherbius foi aperfeiçoada até se transformar na ferramenta criptográfica mais importante da Alemanha nazista. O sistema foi quebrado pelo matemático polonês Marian Rejewski que se baseou apenas em textos cifrados interceptados e numa lista de chaves obtidas por um espião (KAHN, 1967).

A seguir, outros acontecimentos relacionados à utilização da criptografia (TKOTZ, 2003):

Ano	Descrição
1943	Máquina Colossus projetada para quebrar códigos.
1969	James Ellis desenvolve um sistema de chaves públicas e chaves privadas separadas.
1976	Diffie-Hellman é um algoritmo baseado no problema do logaritmo discreto, é o criptossistema de chave pública mais antigo ainda em uso.

Ano	Descrição (cont.)
1976	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 (como as Caixas S e uma chave menor) e adota a cifra como padrão de criptografia de dados nos EUA (FIPS46-3, 1999), conhecido hoje como DES (Data Encryption Standard). Hoje o NBS é chamado de NIST (National Institute of Standards and Technology).
1977	Ronald L. Rivest, Adi Shamir e Leonard M. Adleman começaram a discutir como criar um sistema de chave pública prático. Ron Rivest acabou tendo uma grande idéia e a submeteu à apreciação dos amigos: era uma cifra de chave pública, tanto para confidencialidade quanto para assinaturas digitais, baseada na dificuldade da fatoração de números primos grandes. Foi batizada de RSA, de acordo com as primeiras letras dos sobrenomes dos autores (TKOTZ, 2003).
1978	O algoritmo RSA é publicado na ACM (Association for Computing Machinery), um dos melhores meios de divulgação de pesquisas científicas. Maiores detalhes desta organização podem ser obtidos no link www.acm.org.
1990	Xuejia Lai e James Massey publicam na Suíça "A Proposal for a New Block Encryption Standard" ("Uma Proposta para um Novo Padrão de Encriptação de Bloco" – LAI, 1990), o assim chamado IDEA (International Data Encryption Algorithm), para substituir o DES. O algoritmo IDEA utiliza uma chave de 128 bits e emprega operações adequadas para computadores de uso geral, tornando as implementações em software mais eficientes (SCHNEIER, 1996).
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 usuários que se comunicam por meio de uma rede de comunicação digital. PGP oferece alta segurança para o cidadão comum e, como tal, pode ser encarado como um concorrente de produtos comerciais como o Mailsafe da RSADSI. Entretanto, PGP é especialmente notável porque foi disponibilizado como freeware e, como resultado, tornou-se um padrão mundial, enquanto seus concorrentes da época continuaram absolutamente desconhecidos (BROWN et al., 2000).
1994	Novamente o professor Ronald L. Rivest, autor dos algoritmos RC2 e RC4 incluídos na biblioteca de criptografia BSAFE do RSADSI, publica a proposta do algoritmo RC5 na Internet. Esse algoritmo usa rotação dependente de dados como sua operação não linear e é parametrizado de modo que o usuário possa variar o tamanho do bloco, o número de estágios e o comprimento da chave.
1994	O algoritmo Blowfish, uma cifra de bloco de 64 bits com uma chave de até 448 bits de comprimento, é projetado por Bruce Schneier (SCHNEIER, 1994).
1997	O PGP 5.0 *Freeware* é amplamente distribuído para uso não comercial.
1997	O código DES de 56 bits é quebrado por uma rede de 14.000 computadores (CURTIN e DOLSKE, 1998).
1998	O código DES é quebrado em 56 horas por pesquisadores do Vale do Silício (DESKEY, 2001).
1999	O DES é quebrado em apenas 22 horas e 15 minutos, mediante a união da Electronic Frontier Foundation e a Distributed.Net, que reuniram em torno de 100.000 computadores pessoais ao DES Cracker pela Internet (MESERVE, 1999).

Ano	Descrição (cont.)
2000	O NIST (National Institute of Standards and Technology) anunciou um novo padrão de uma chave secreta de cifragem, escolhido entre 15 candidatos. Esse novo padrão foi criado para substituir o algoritmo DES, cujo tamanho das chaves tornou-se insuficiente para conter ataques de força bruta (MESERVE, 1999). O algoritmo Rijndael, cujo nome é uma abreviação dos nomes dos autores Rijmen e Daemen, foi escolhido para se tornar o futuro AES (Advanced Encryption Standard) (FIPS197, 2001).
2000 - 2004	Muitos professores e profissionais da computação com vínculo em centros de pesquisa, universidades e empresas motivam-se e começam a pesquisar novas formas de implementar algoritmos e soluções de segurança. Surge, assim, uma "onda" de pesquisas e desenvolvimentos voltados a realizar otimizações dessas primeiras implementações e uma dessas tendências é a implementação em hardware. Assim, este livro destaca a importância de se implementar alguns desses algoritmos criptográficos em hardware, em especial, por meio do uso da tecnologia de circuitos programáveis (FPGAs), a qual é acessível e diminui de forma significativa os tempos e custos associados à realização de projetos e protótipos.

Os computadores são a expressão maior da era digital, marcando presença em praticamente todas as atividades humanas. Da mesma forma com que revolucionaram a informação, também influenciaram a criptologia; por um lado, ampliaram seus horizontes, por outro, tornaram a criptologia quase que indispensável. Na seção 1.1.2 se apresentará a importância da criptografia.

1.1.2 A Importância da Criptografia

Nesta seção é discutida a importância da criptografia, a segurança dos sistemas operacionais e por que se deve utilizar esse recurso contra intrusos que desejam acessar informações alheias.

A segurança eletrônica nunca foi tão amplamente discutida: casos de violação de contas bancárias, acesso a informações sigilosas, invasão e destruição de sistemas são cada vez mais comuns. Informações são transmitidas com mais eficiência e velocidade, mas como se sabe, nem sempre de forma segura.

A privacidade é importante para pessoas e empresas. Muitos problemas podem acontecer se uma pessoa não autorizada tiver acesso a dados pessoais, como: contracheque, saldo bancário, faturas do cartão de crédito, diagnósticos de saúde e senhas bancárias ou de crédito automático. No caso de empresas, os danos podem ser de maior magnitude, atingindo a organização e os próprios funcionários. Dados estratégicos da empresa, previsão de venda, detalhes técnicos de produtos, resultados de pesquisas e arquivos pessoais são informações valiosas, às quais se alguma empresa concorrente tiver acesso de forma indevida, tal fato poderá acarretar sérios problemas.

A Internet é um ambiente que viabiliza principalmente a comunicação, a divulgação, a pesquisa e o comercio eletrônico. Em 1999, havia mais de 100 milhões de usuários da Internet nos Estados Unidos. No final de 2003, esse número alcançou 177 milhões nos Estados Unidos

e 502 milhões no mundo todo (BURNETT e PAINE, 2002). O comércio eletrônico emergiu como um novo setor da economia norte-americana, sendo responsável por cerca de U$100 bilhões em vendas durante 1999. Em 2003, o comércio eletrônico excedeu U$1 trilhão. Ao mesmo tempo, o CSI (Computer Security Institute) constatou um aumento de crimes cibernéticos: 55% dos entrevistados na pesquisa informaram atividades maliciosas relacionadas com pessoas da própria organização. Ciente disso, pode-se ter certeza que as empresas em expansão precisam de produtos, mecanismos e soluções de segurança (BURNETT, 2002).

Há não muito tempo, a segurança era uma questão de se trancar uma porta ou um cofre. Atualmente as informações geralmente não estão armazenadas somente em papéis, e, sim, em banco de dados. Como proteger essas informações? O que os sistemas operacionais (SO) oferecem para essa proteção?

Os sistemas operacionais oferecem um sistema de proteção por meio de permissões (Figura 1.2), isto é, por meio do SO é possível criar usuários com diferentes níveis de acesso para as informações contidas em um computador. Tal acesso é implementado via procedimento de login. Assim quando um determinado usuário fizer login em um computador, terá acesso às pastas e aos arquivos designados pelo seu nível de permissão, isto é, se um usuário tiver restrições para o acesso de pasta, arquivos e programas, este não conseguirá acessá-los.

Os cadastros de usuários e as permissões são concedidos pelo superusuário ou administrador do sistema. Este é responsável pelo gerenciamento do sistema, até pode permitir que outros usuários alterarem algumas permissões, como de uma pasta pessoal. Independentemente das restrições impostas, com o login de superusuário pode-se ter acesso a todas as funções do sistema.

A questão é como o sistema operacional sabe que a pessoa que está acessando o sistema é realmente o superusuário? O sistema operacional concede a permissão por meio de um nome de usuário e senha, normalmente os nomes de usuários administradores são `root`, `su` ou `administrador` e, infelizmente, sabe-se que as técnicas para superar essas defesas são amplamente conhecidas (BURNETT, 2002). A seguir são descritos alguns dos principais ataques:

Figura 1.2 – Configuração da permissão de acesso a uma pasta no Windows 2000.

- **Ataque contra senhas**: Vários sistemas operacionais vêm com um nome de usuário e senhas predefinidas e, muitas vezes, o mesmo login é utilizado para realizar várias tarefas, como: criação de usuários, manutenção e backup, instalação de programas etc.

  Não é uma boa prática utilizar o login predefinido. A utilização da senha predefinida ou de uma senha derivada de uma data de aniversário ou qualquer dado pessoal pode facilitar o ataque ao sistema. Se o invasor não tiver competência para descobrir a senha, poderá utilizar aplicativos, chamados de software de cracking de senha. Esses softwares testam exaustivamente todas as possibilidades de combinações de senhas até encontrar uma válida; se a senha for fraca, em questão de minutos o invasor terá acesso ao sistema.

  Seria interessante dar preferência a diferentes senhas de superusuário para cada função do sistema, pois se o sistema for invadido, o intruso não terá acesso a todas as informações do sistema, mesmo acessando com um login de superusuário. Uma outra maneira de invadir e recuperar os dados do sistema é desviando do controle de permissões do SO, chamado de ataque de recuperação de dados.

- **Ataques de recuperação de dados**: O sistema operacional organiza as informações em arquivos e diretórios para que o usuário possa acessar rapidamente a uma determinada informação. Contudo, os dados e o controle de acesso a estes são bits eletrônicos. Assim, é possível fazer uma leitura desses bits não como arquivos de texto ou de números, e, sim, como bits, independentemente do sistema operacional. Esses ataques desviam do sistema operacional e capturam os bits brutos, reconstituindo-os em arquivos originais, burlando os controles de permissão do sistema operacional.

- **Ataque de reconstrução de memória**: Freqüentemente, o material sigiloso está armazenado na memória do computador. Quando executar um programa, este será armazenado em uma área da memória principal e o sistema operacional indicará essa área como indisponível. Quando o programa é finalizado, o sistema operacional apenas disponibiliza a área sem sobrescrevê-la. O invasor simplesmente aloca a memória liberada e examina o que sobrou (BURNETT, 2002).

Como impedir que um invasor tenha acesso a informações privadas?

Para impedir o acesso a informações privadas, pode-se utilizar a proteção por criptografia. A proteção por criptografia é uma solução prática para proteger informações sigilosas. Independentemente do algoritmo criptográfico utilizado, sempre ocorrerá transformação de um texto legível em um ilegível. Mesmo que o invasor obtenha o conteúdo de um arquivo, este será ilegível. Para ter acesso à informação original, o invasor terá que resolver um problema matemático de difícil solução. A criptografia pode adicionar também maior segurança ao processo de identificação de pessoas, criando identidades digitais fortes.

De modo algum a criptografia é a única ferramenta necessária para assegurar a segurança de dados, nem resolverá todos os problemas de segurança. É um instrumento entre vários outros. Além disso, a criptografia não é à prova de falhas. Toda criptografia pode ser quebrada

e, sobretudo, se for implementada incorretamente, não agregará nenhuma segurança real (BURNETT e PAINE, 2002).

1.1.3 Alguns Termos Utilizados na Criptografia

Com o conceito de criptografia, têm-se alguns termos oficiais comumente utilizados, que são conceituados nesta seção.

O ato de transformar um texto legível (texto claro, texto original, texto simples) em algo ilegível (cifra, texto cifrado, texto código) é chamado de "encriptar" (codificar, criptografar, cifrar). A transformação inversa é chamada de "decriptar" (decodificar, decriptografar, decifrar).

O algoritmo de criptografia é uma seqüência de procedimentos que envolvem uma matemática capaz de cifrar e decifrar dados sigilosos. O algoritmo pode ser executado por um computador, por um hardware dedicado e por um humano. Em todas as situações, o que diferencia um de outro é a velocidade de execução e a probabilidade de erros. Existem vários algoritmos de criptografia. Neste livro, apresentam-se especificamente o DES, AES, RC5, IDEA e RSA, MD5 e SHA-1, pois são os mais utilizados atualmente, além do ALPOS (um algoritmo didático criado pelos autores).

Além do algoritmo, utiliza-se uma chave. A chave na criptografia computadorizada é um número ou um conjunto de números. A chave protege a informação cifrada. Para decifrar o texto cifrado, o algoritmo deve ser alimentado com a chave correta, que é única. Na figura 1.3, tem-se a ilustração do esquema geral de cifragem utilizando chave:

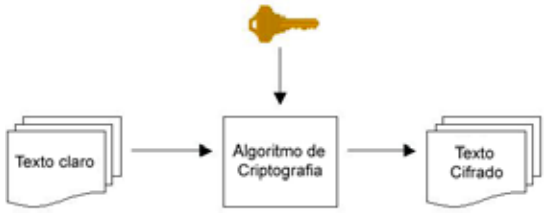

Figura 1.3 – Esquema geral de cifragem com chave.

Na história da criptografia, sempre ficou evidente que não existe um algoritmo que não possa ser quebrado (descoberto ou solucionado). Com a criptografia computadorizada, atualmente os algoritmos são divulgados à comunidade e o sigilo das informações é garantido apenas pela chave, o que significa que se alguém descobrir a chave para decifrar uma determinada informação, todas as outras informações cifradas com esse algoritmo ainda estarão protegidas, por terem chaves diferentes.

Já um algoritmo criptográfico que não utiliza o recurso de chaves para cifrar as informações pode levar a um efeito cascata perigoso. Se esse algoritmo for quebrado, todas as informações cifradas com ele estarão desprotegidas, pois o que garante o sigilo das informações seria o próprio algoritmo. Assim, toda criptografia moderna e computadorizada opera com chaves.

A criptografia desde seu início foi desenvolvida para impedir que um invasor ou intruso, alguém que está tentando acessar informações sigilosas, tivesse êxito. Há pouco tempo a criptografia era amplamente utilizada para proteger informações militares. Atualmente protege uma gama maior de diferentes informações. Os invasores não necessariamente só querem acessar informações sigilosas, mas também desativar sites, excluir informações de uma pessoa ou empresa, danificar sistemas em geral.

O estudo sobre a quebra de sistemas criptográficos é conhecido como análise criptográfica. Semelhantemente ao invasor, o criptoanalista procura as fraquezas dos algoritmos. O criptógrafo desenvolve sistemas de criptografia. É importante que a comunidade de criptografia conheça as fraquezas, pois os invasores também as estão procurando. É quase certo que os invasores não irão publicar suas descobertas para o mundo (BURNETT, 2002).

1.1.4 Algoritmos de Bloco e Fluxo

Pode-se classificar os algoritmos de criptografia por meio do tratamento dado às informações que serão processadas; assim, têm-se os algoritmos de bloco e os algoritmos de fluxo.

A cifra de blocos opera sobre blocos de dados. O texto antes de ser cifrado é dividido em blocos que variam normalmente de 8 a 16 bytes que serão cifrados ou decifrados. Quando o texto não completa o número de bytes de um bloco, este é preenchido com dados conhecidos (geralmente valor zero “0”) até completar o número de bytes do bloco, cujo tamanho já é predefinido pelo algoritmo que está sendo usado.

A forma mais comum de preenchimento é determinar o número de bytes que deve ser preenchido e utilizar esse valor para preencher o bloco. Por exemplo, suponha que o tamanho do bloco em um determinado algoritmo seja de 16 bytes e foram utilizados apenas 9. Deve-se preencher os 7 bytes restantes com o valor 07.

Um problema na cifra de bloco é que se o mesmo bloco de texto simples aparecer mais de uma vez, a cifra gerada será a mesma, facilitando o ataque ao texto cifrado. Para resolver esse problema são utilizados os modos de realimentação.

O modo mais comum de realimentação é a cifragem de blocos por encadeamento CBC (Cipher Block Chaining). Neste modo é realizada uma operação de XOR do bloco atual de texto simples com o bloco anterior de texto cifrado. Para o primeiro bloco, não há bloco anterior de texto cifrado; assim, faz-se uma XOR com um vetor de inicialização. Este modo não adiciona nenhuma segurança extra. Apenas evita o problema citado da cifra de bloco.

Portanto, os algoritmos de blocos processam os dados como um conjunto de bits, sendo os mais rápidos e seguros para a comunicação digital. Ainda, como vantagem, que os blocos podem ser codificados fora de ordem, o que é bom para acesso aleatório, além de ser resistente a erros, uma vez que um bloco não depende de outro. Entretanto, como desvantagem, se a mensagem possuir padrões repetitivos nos blocos, o texto cifrado também o apresentará, o que facilitará o serviço do criptoanalista. Outra desvantagem é que um bloco pode ser substituído por outro modificando a mensagem original.

Os algoritmos de fluxo criptografam (cifram) a mensagem bit a bit, em um fluxo contínuo, sem esperar que se tenha um bloco completo de bits. É também chamado de criptografia em stream de dados, onde a criptografia se dá mediante uma operação XOR entre o bit de dados e o bit gerado pela chave.

A tabela 1.1 apresenta algumas recomendações para selecionar o tipo de algoritmo que deverá ser usado em uma determinada aplicação:

Tabela 1.1 – Escolhendo um algoritmo por aplicação (BURNETT, 2002)

Aplicação	Cifragem recomendada	Comentários
Banco de dados	Bloco	A interoperabilidade com um outro software não é relevante, mas é necessário reutilizar as chaves.
E-mail	AES	Ganha-se interoperabilidade em todos os pacotes de e-mail utilizando o AES-padrão.
SSL	RC4	A velocidade é extremamente importante, cada conexão pode ter uma nova chave. Assim, na prática, a maioria dos navegadores e servidores possui o RC4.
Criptografia de arquivos	Bloco	A interoperabilidade não é relevante, porém cada arquivo pode ser cifrado com a mesma chave e, então, protegê-la.

1.1.5 Vírus ou Informação Cifrada

Fernando de la Cuadra (2003), editor técnico internacional da empresa Panda Software, empresa de software de segurança e antivírus, aponta em seu artigo vantagens e problemas da criptografia atual.

Enviar uma mensagem cifrada por correio eletrônico traz vantagens tanto para o emissor como para o receptor. A confidencialidade está praticamente assegurada. Ninguém que não conheça a chave utilizada na cifragem poderá entender as informações da mensagem. Assim, pode-se enviar todo tipo de informação com um bom nível de segurança, estando-se praticamente a salvo de teóricas interceptações na comunicação.

Mas quem intercepta uma comunicação? Em princípio, imagina-se que seja um hacker, um espião ou qualquer outro usuário que queira acessar as informações da comunicação, mas também pode ser um antivírus fazendo uma verificação.

Um antivírus sempre tentará impedir o ataque de um vírus, para isso examinará a mensagem enviada pelo correio eletrônico. O que pode acontecer é que dentro do conteúdo de uma mensagem cifrada exista um vírus e o antivírus não o identifique. Ou, ainda, o antivírus pode identificar uma mensagem cifrada como sendo um vírus e essa mensagem poderá ser excluída.

Atualmente 90% dos vírus estão espalhados pela Internet. Sendo assim, o mais lógico é a instalação de antivírus nos firewalls, proxys etc. Mas se o vírus estiver em uma mensagem de correio eletrônico cifrada, até o melhor antivírus poderá falhar em seu objetivo de proteger as informações. E os usuários que receberem a mensagem cifrada, inconscientemente, serão infectados (CUANDRA, 2003).

Em definitivo, ninguém dúvida de que os sistemas de criptografia sejam uma ferramenta que fornece segurança às comunicações, mas podem ter uma deficiência: esconder vírus. Uma solução para evitar que o vírus cifrado entre em uma determinada companhia ou organização deveria ser uma proteção efetiva que bloqueia as mensagens cifradas não autorizadas, antes de chegar aos usuários finais (CUANDRA, 2003).

1.2 Importância da Chave ou "Senha"

O termo "chave" vem do fato de que o número secreto, a famosa senha utilizada nos sistemas computacionais, funciona da mesma maneira que uma chave convencional usada nas portas e entradas a lugares fechados de residências, empresas etc., de modo a proteger o patrimônio de um determinado usuário.

Assim, algo similar ocorre com a criptografia, em que para proteger a informação de um determinado usuário (armazenada em arquivos de computador), deve-se instalar uma fechadura (algoritmo de criptografia). Para operar a fechadura, precisa-se da famosa chave ou senha (número secreto), a qual permite cifrar ou decifrar a informação desejada.

O algoritmo realiza seus passos utilizando a chave para alterar o texto simples (mensagem original) e convertê-lo em texto cifrado. Para recuperar a informação em forma legível, é necessário inserir a mesma chave ou outra que esteja relacionada com aquela que foi usada no processo anterior e executar a operação inversa. O algoritmo inverte os passos e converte o texto cifrado de novo ao texto simples original.

Assim como apenas a chave correta de um determinado prédio pode abrir a entrada deste, apenas a chave correta usada em criptografia pode cifrar ou decifrar os dados. Na criptografia de chave simétrica, a chave que é utilizada para criptografar os dados é a mesma chave utilizada para decifrá-los. Na criptografia assimétrica, usa-se outra chave que possui um valor relacionado com essa primeira chave. Na seção 1.3, apresentam-se mais detalhes sobre criptografia simétrica e assimétrica.

Toda criptografia moderna e computadorizada opera com chaves. Por que uma chave é necessária? Por que não criar um algoritmo que não necessite de uma chave? Se os invasores podem entender o algoritmo, podem recuperar os dados secretos simplesmente executando o algoritmo.

Uma primeira solução seria manter o algoritmo em segredo, mas essa abordagem apresenta vários problemas. Supondo que não seja possível manter o algoritmo em segredo, os invasores sempre quebrarão o algoritmo. Tal fato não seria possível se houvesse especialistas em criptografia que desenvolvessem seus próprios algoritmos, mas neste caso também se deveria acreditar que a empresa ou usuário que escreveu o algoritmo nunca o revelaria.

Neste ponto, há um aspecto relevante em criptografia: o que é mais importante, um algoritmo que deve ser mantido em segredo ou um algoritmo que possa cifrar informações mesmo que os usuários de sistemas computacionais saibam como ele funciona. Neste segundo aspecto, destacam-se a importância dos algoritmos com chave e, principalmente, a relevância da chave.

As chaves minimizam a preocupação com o algoritmo utilizado no esquema de criptografia. Em termos computacionais, para proteger os dados com uma chave, é necessário proteger apenas a chave, algo que é mais fácil do que proteger um algoritmo. Além disso, se utilizar chaves para proteger os segredos, é possível utilizar diferentes chaves para proteger diversos segredos. Assim se alguém descobrir ("quebrar") uma das chaves, os outros segredos ainda poderão estar seguros. Se alguma informação depender somente de um algoritmo secreto, qualquer invasor que quebre esse segredo terá acesso a todas as informações contidas em um determinado sistema computacional.

As chaves são muito importantes em criptografia. Segundo o princípio de Kerckhoffs, que menciona a relevância do espaço de chaves, é muito mais seguro um sistema onde se conhece o algoritmo de criptografia que o espaço de chaves. Ao se manter em segredo as chaves, supondo-se conhecer o algoritmo usado, gera-se um sistema incondicional e computacionalmente seguro.

Esse princípio ainda é válido por várias razões (KERCKHOFFS, 1983). Os invasores podem deduzir um algoritmo sem nenhuma ajuda. Na história da criptografia, nunca alguém foi capaz de manter um algoritmo criptográfico em segredo. Este é sempre descoberto.

Um exemplo disso é que durantes as guerras, os espiões sempre encontraram maneiras de descobrir o algoritmo, seja este originado de uma operação matemática, seja de uma máquina. Eles o roubavam ou faziam com que alguém o revelasse (por meio de chantagem, extorsão ou uso de técnicas de análise criptográfica). Agentes sempre descobriam o algoritmo ou obtinham uma cópia da máquina.

Por exemplo, na Segunda Guerra Mundial, os soldados poloneses capturaram a máquina alemã Enigma, logo no início da guerra. Enigma era uma máquina de criptografia utilizada pelo exército alemão. Os aliados (isto é, os britânicos) foram capazes de quebrar o código mais facilmente porque tinham a posse dessa máquina.

De forma alternativa, os analistas criptográficos podem descobrir o algoritmo. Na Segunda Guerra Mundial, decifradores de código dos Estados Unidos foram capazes de determinar o funcionamento interno das máquinas codificadas japonesas sem ter a posse de uma dessas máquinas.

Um caso mais recente refere-se ao RC4, um algoritmo inventado em 1987, mas nunca publicado. Os analistas de criptografia e outros especialistas o estudaram e determinaram que o RC4 era uma boa maneira de manter os dados em segredo. Atualmente, o RC4 é utilizado como parte do Secure Sockets Layer (SSL), o protocolo de comunicação segura da WEB (World Wide Web).

Mas a empresa que o criou, a RSA Data Security, nunca tornou público o funcionamento interno do algoritmo RC4. Esse segredo era mantido por interesses financeiros e não de segurança. A empresa esperava que mantendo-o em sigilo ninguém mais o implementaria e comercializaria. Em 1994, hackers anônimos divulgaram o algoritmo na Internet. Acredita-se que eles provavelmente o tenham descoberto usando um depurador de linguagem assembly, após terem acesso a uma cópia do código-objeto.

Se um sistema criptográfico estiver baseado no hardware, os engenheiros podem abri-lo e examinar as partes internas dele. Em 1998, David Wagner e Ian Goldberg, nessa época alunos graduados da Universidade da Califórnia, em Berkeley, abriram um telefone celular digital, supostamente seguro, e quebraram seu código.

Às vezes é possível manter um algoritmo em segredo por um período suficientemente longo, para que seja eficaz, mas após certo tempo alguém acabará descobrindo-o.

A segunda razão de relevância da senha, mais do que o algoritmo usado, refere-se a assuntos comerciais. Empresários e usuários sempre desejam saber como um determinado software pelo qual se interessam foi criado. Caso a empresa não revele os segredos de implementação e construção, por meio de técnicas de reengenharia reversa sobre um software podem-se conhecer detalhes da sua criação e respectiva implementação.

Soma-se a isso o fato de que somente pessoas que adquiriram o produto podem se comunicar entre si. Tal fato inviabiliza a comunicação com pessoas que não adquiriram ou compraram o algoritmo de um mesmo fornecedor. Dessa maneira, como resultado, os algoritmos devem ser padronizados, o que significa que devem ser públicos.

Se alguém quiser utilizar a criptografia, é necessário empregar um dispositivo de hardware ou um programa de software. Portanto, torna-se necessário adquirir o produto em algum estabelecimento especializado. Da mesma forma como os usuários podem ter acesso a esse dispositivo, os invasores também o têm. Desse modo, possíveis invasores podem ir à mesma fonte e obter suas próprias cópias. Tal fato faz com que os algoritmos devam ser disponibilizados, ou melhor, deve-se cuidar da chave.

Uma outra razão para cuidar da chave, e não do algoritmo, refere-se ao fato de que é possível construir sistemas criptográficos nos quais o algoritmo é completamente conhecido e seguro, pois os possíveis invasores precisam conhecer a chave para descobrir as informações. Esses sistemas são mais seguros que aqueles que não têm chave, visto que somente confiam no segredo de não se conhecer o algoritmo.

Quando os algoritmos se tornam públicos, os analistas criptográficos e os profissionais da área de computação têm uma chance de examinar as fraquezas destes. Se um algoritmo for vulnerável, pode-se optar por não utilizá-lo. Caso contrário, pode-se ter certeza de que os dados estão seguros. Por outro lado, se um algoritmo for mantido em segredo, os analistas não serão capazes de encontrar nenhuma fraqueza que este possa apresentar, e, assim, não saberão se é ou não vulnerável.

1.2.1 Como Gerar a Chave

Em um sistema criptográfico de chave simétrica, a chave é formada por um conjunto de caracteres, podendo ser um número qualquer ou uma seqüência de caracteres alfanuméricos (letras e símbolos especiais), contanto que tenha um tamanho correto e adequado àquele que o algoritmo criptográfico selecionado permitir.

Geralmente, aconselha-se que a chave seja formada por números ou caracteres alfanuméricos sem coerência nenhuma entre si, pois assim será mais difícil descobri-la (ou, em termos criptográficos, quebrá-la). A chave deve ser, dentro do possível, selecionada de forma aleatória. Para os criptógrafos, valores aleatórios são simplesmente conjuntos de números que passam em testes estatísticos de aleatoriedade e não são repetíveis.

Com o intuito de gerar chaves, há várias técnicas utilizadas. Entre elas, destaca-se o uso de sistemas geradores de números aleatórios (GNAs), ou random number generator (RNGs). Esses dispositivos funcionam agrupando números de diferentes tipos de entradas imprevisíveis, como a medição da desintegração espontânea de radioatividade, o exame das condições atmosféricas ou o cálculo de minúsculas variâncias na corrente elétrica. Esses números passam por testes de aleatoriedade. Se solicitar um segundo grupo de números, a nova seqüência será completamente diferente, isto é, nunca receberá a mesma seqüência novamente. Tal situação ocorre porque a saída é baseada em uma entrada que sempre está mudando.

Esses números podem ser obtidos utilizando-se algoritmos chamados de geradores de números pseudo-aleatórios (GNPAs), ou pseudo-random number generators (PRNGs). Se um desses algoritmos for utilizado para gerar alguns milhares de números e aplicar testes estatísticos, os números passariam no teste de aleatoriedade. O que torna esses números pseudo-aleatórios, e não aleatórios, é o fato de ser repetíveis. Se for instalado o mesmo GNPA em um outro computador, os mesmos resultados poderão ser obtidos. Se o programa for executado um certo tempo depois, esses mesmos resultados também poderão ser obtidos.

Para evitar esse problema, costuma-se usar GNPA com uma entrada diferente em cada evento de utilização (chamada de semente). Assim, gerarão dados diferentes, pois alterando a entrada, a saída também o será. A geração aleatória dessa chave está associada a dois parâmetros importantes: velocidade de geração e entropia, que estão intimamente relacionadas com o tamanho da chave.

1.2.2 Importância do Tamanho da Chave

Se os invasores puderem descobrir qual é a chave usada na cifragem dos dados, poderão decifrar a mensagem enviada e obter os dados contidos nela.

Um método conhecido como ataque de força bruta consiste em tentar todas as possíveis chaves até que a correta seja identificada. Suponha que a chave seja um número entre 0 e 1.000.000 (um milhão). O invasor pega o texto cifrado e alimenta o algoritmo de criptografia com a "suposta chave" de valor "0". O algoritmo realiza seu trabalho e produz um resultado.

Se os dados resultantes parecerem razoáveis, “0” provavelmente é a chave correta. Se for um texto sem sentido, “0” não é a verdadeira chave. Nesse caso, ele tentará outro valor, por exemplo, “1” e, em seguida, “2”, “3”, “4”, e assim por diante.

Um algoritmo simplesmente realiza os passos, independentemente da entrada. Não há nenhuma maneira de saber se o resultado que ele produz é o correto. Mesmo se o valor estiver próximo da chave, talvez errado em apenas “1”, o resultado será um texto sem sentido.

Assim, é necessário examinar o resultado, compará-lo para identificar algum sentido e, assim, informar se o valor usado como chave pode ser a chave realmente usada para cifrar as mensagens.

Como tal procedimento depende de uma seqüência de entrada e saída, com valores supostos de chaves, um método consiste em criar programas que sigam esses passos até descobrirem alguma informação.

Normalmente, esse processo requer pouco tempo para testar uma chave. Assim, pode-se escrever um programa que verifique várias chaves por segundo. No âmbito da computação, tal operação torna possível descobrir qualquer chave, somente se necessita de tempo.

É interessante perceber que esse tempo de procura está muito associado ao tamanho da chave. Chaves criptográficas são medidas em bits. O intervalo de possíveis respostas para identificar uma chave está em correspondência ao número 2^{TC}, em que “TC” é o tamanho da chave em bits.

Assim, uma chave de 2 bits significa que o intervalo de possíveis valores é de 0 até $2^2 = 4$. Uma chave de 40 bits significa que o intervalo dos possíveis valores é de 0 até aproximadamente 1 trilhão (2^{40}). Uma chave de 56 bits é de 0 até aproximadamente 72 quatrilhões (2^{56}). O intervalo de uma chave de 128 bits é tão grande que é mais fácil apenas dizer que se trata de uma chave de 128 bits (número de possibilidades igual a 2^{128}).

Cada bit adicionado ao tamanho da chave dobrará o tempo requerido para um ataque de força bruta. Se uma chave de 40 bits levasse 3 horas para ser quebrada, uma chave de 41 bits levaria 6 horas, uma chave de 42 bits, 12 horas, e assim por diante. Essa situação ocorre visto que cada bit adicional da chave dobra o número de chaves possíveis (lembre-se que esse número está em função de 2^{TC}). Assim, ao adicionar um bit, o número de chaves possíveis é dobrado. Dobrando o número de chaves possíveis, o tempo médio que um ataque de força bruta leva para encontrar a chave correta também é dobrado.

Portanto, para se ter maior segurança, isto é, tornar o trabalho de um determinado invasor mais difícil, deve-se escolher uma chave maior. Chaves mais longas significam maior segurança. A tabela 1.2 mostra o impacto de se aumentar o tamanho da chave e o respectivo tempo de quebrar a chave usando ataque por força bruta, assim como a respectiva estimativa de custo (em dólares) da tecnologia necessária para encontrar a chave. Pode-se constatar que tendo mais recursos disponíveis (em tecnologia e, portanto, maior custo em dólares), é possível diminuir

o tempo para encontrar uma determinada chave. Não obstante seja possível achar a chave, há tamanhos de chave que inviabilizam essa ação, dado que demandaria muito tempo.

Nessa tabela, o termo 2s significa 2 segundos; 200 ms significa que o tempo é dado em milissegundos (10^{-3} segundos); 200 us significa que o tempo é dado em microssegundos (isto é, 10^{-6} segundos):

Tabela 1.2 – Tempo gasto para quebra de chaves por força bruta (SCHNEIER, 1996)

	Tamanho da chave (bits)					
Custo U$	**40**	**56**	**64**	**80**	**112**	**128**
100 mil	2 s	35 horas	1 ano	70.000 anos	10^{14} anos	10^{19} anos
1 milhão	200 ms	3,5 h	37 dias	7.000 anos	10^{13} anos	10^{18} anos
10 milhões	20 s	21 m	4 dias	700 anos	10^{12} anos	10^{17} anos
100 milhões	2 ms	2 m	9 h	70 anos	10^{11} anos	10^{16} anos
1 bilhão	200 us	13 s	1 h	7 anos	10^{10} anos	10^{15} anos
10 bilhões	20 us	1 s	5,4 m	245 dias	10^{9} anos	10^{14} anos
100 bilhões	2 us	100 ms	32 s	24 dias	10^{8} anos	10^{13} anos
1 trilhão	0,2 us	10 ms	3 s	2,4 dias	10^{7} anos	10^{12} anos
10 trilhões	0,02 us	1 ms	300 ms	6 horas	10^{6} anos	10^{11} anos

É importante lembrar que o poder da computação dobra a cada 1,5 ano e que a estimativa do tempo de existência do universo, segundo os últimos estudos científicos, está em torno de 10^{10} anos. Assim, chega-se à conclusão de que sempre é possível decifrar uma determinada mensagem, pois sempre será possível descobrir a chave: basta testar todas as chaves possíveis; é somente uma questão de tempo. Entretanto, essa ação pode demorar mais que o tempo de duração do universo (SCHNEIER, 1996).

Então, qual o tamanho máximo que uma chave deve ter? Com o passar dos anos, o RSA Laboratories propôs alguns desafios. A primeira pessoa ou empresa a quebrar uma mensagem em particular ganharia um prêmio em dinheiro. Alguns desafios foram testes do tempo de um ataque de força bruta. Em 1997, uma chave de 40 bits foi quebrada em 3 horas e uma chave de 48 bits durou 280 horas. Em 1999, a Electronic Frontier Foundation encontrou uma chave de 56 bits em 24 horas. Em cada um dos casos, pouco mais de 50% do espaço de chave foi pesquisado antes de a chave ser encontrada.

Em todas essas situações, centenas ou até milhares de computadores operavam conjuntamente para quebrar as chaves. Na realidade, com o desafio de 56 bits de DES que a Electronic Frontier Foundation quebrou em 24 horas, um dos computadores era um cracker especializado em DES. Esse tipo de computador realiza apenas uma tarefa: verifica as chaves de DES.

Um invasor que trabalhe secretamente, provavelmente não seria capaz de reunir a força de centenas de computadores e talvez não possua uma máquina especificamente construída para quebrar um algoritmo em particular. Para a maioria dos invasores, essa é a razão pela qual o tempo que despendem para quebrar a chave quase certamente seria significativamente maior. Por outro lado, se o invasor fosse uma agência governamental de inteligência com grandes recursos, a situação seria diferente.

Pode-se pensar em situações ainda mais críticas. Suponha que examinar 1% do espaço de chave de uma chave de 56 bits leva 1 segundo e examinar 50% , 1 minuto. Todas as vezes que se adicionar um bit ao tamanho de chave, dobra-se, então, o tempo de pesquisa. Os resultados são mostrados na tabela 1.3, onde se percebe que o impacto e observações realizadas para os dados da tabela 1.2 são também aplicáveis.

Tabela 1.3 – Tempo gasto para quebra de chaves (BURNETT, 2002)

Bits	1% do espaço da chave	50% do espaço da chave
56	1 segundo	1 minuto
57	2 segundos	2 minutos
58	4 segundos	4 minutos
64	4,2 minutos	4,2 horas
72	17,9 horas	44,8 dias
80	190,9 dias	31,4 anos
90	535 anos	321 séculos
108	140 mil milênios	8 milhões de milênios
128	146 bilhões de milênios	8 trilhões de milênios

Atualmente, 128 bits é o tamanho de chave simétrica mais comumente utilizado. Se a tecnologia avançar e os invasores de força bruta puderem melhorar esses números (talvez possam reduzir para alguns anos o tempo das chaves de 128 bits), então serão necessárias chaves de 256 bits ou ainda maiores.

Assim, considerando que a tecnologia está avançando sempre, teremos de aumentar repetidas vezes o tamanho das chaves. Com o passar do tempo, será necessário aumentar o tamanho das chaves. É bom lembrar que aumentar a chave significa também aumentar o tempo cifragem e decifragem. Por isso, recomenda-se atenção em relação à escolha do tamanho da chave. Assim, é possível pensar que poderá chegar um momento em que precisaremos de uma chave tão grande que com esta será muito difícil de lidar. No momento atual, considerando-se os conhecimentos adquiridos na área de segurança, quase certamente nunca precisaremos de uma chave mais longa que 512 bits (64 bytes). Supondo que cada átomo no universo conhecido (há aproximadamente 2^{300}) fosse um computador e que cada um desses computadores pudesse verificar 2^{300} chaves por segundo, essa tarefa levaria cerca de 2^{162} milênios para pesquisar 1% do espaço de chave de uma chave de 512 bits. De acordo com a teoria do `big-bang`, o tempo decorrido desde a criação do universo é menor que 2^{24} milênios. Em outras palavras, é altamente improvável que a tecnologia vá tão longe para forçar a utilizar chaves que sejam "muito grandes" (BURNETT e PAINE, 2002).

Apesar de o ataque de força bruta precisar de muito tempo, principalmente quando a chave possui um tamanho razoável, há outros ataques que exploram as fraquezas nos algoritmos criptográficos e tendem a diminuir o tempo do ataque. Por esse motivo, no momento se recomenda usar chaves relativamente longas, superiores a 128 bits. Já para aplicações comerciais e financeiras (por exemplo, transações bancárias), é necessário que a chave seja maior que 128 bits. Tal fato significa que um usuário comum que tente fazer um ataque de força bruta, precisará de muito tempo para ter uma invasão considerada bem-sucedida.

1.3 Criptografia de Chaves Simétrica e Assimétrica

Na criptografia de chave simétrica, os processos de cifragem e decifragem são feitos com uma única chave, ou seja, tanto o remetente quanto o destinatário usam a mesma chave. Em algoritmos simétricos, como, por exemplo, o DES (Data Encription Standard), ocorre o chamado "problema de distribuição de chaves". A chave tem de ser enviada para todos os usuários autorizados antes que as mensagens possam ser trocadas. Essa ação resulta num atraso de tempo e possibilita que a chave chegue a pessoas não autorizadas.

A criptografia assimétrica contorna o problema da distribuição de chaves mediante o uso de chaves públicas. A criptografia de chaves públicas foi inventada em 1976 por Whitfield Diffie e Martin Hellman, a fim de resolver o problema da distribuição de chaves. Neste novo sistema, cada pessoa tem um par de chaves denominado chave pública e chave privada. A chave pública é divulgada, enquanto a chave privada é mantida em segredo. Para mandar uma mensagem privada, o transmissor cifra a mensagem usando a chave pública do destinatário pretendido, que deverá usar a sua respectiva chave privada para conseguir recuperar a mensagem original.

Atualmente, um dos mecanismos de segurança mais utilizados é a assinatura digital, que precisa dos conceitos de criptografia assimétrica. A assinatura digital é uma mensagem que só uma pessoa poderia produzir, mas que todos possam verificar. Normalmente autenticação se refere ao uso de assinaturas digitais: a assinatura é um conjunto inforjável de dados assegurando o nome do autor que funciona como uma assinatura de documentos, ou seja, que determinada pessoa concordou com o que estava escrito. Tal procedimento também evita que a pessoa que assinou a mensagem depois possa se livrar de responsabilidades, alegando que a mensagem foi forjada. Um exemplo de criptossistema de chave pública é o RSA (Rivest-Shamir-Adleman), cuja maior desvantagem é a sua capacidade de canal limitada, ou seja, o número de bits de mensagem que pode transmitir por segundo (BURNETT, 2002).

A figura 1.4 ilustra o funcionamento das criptografias simétrica e assimétrica. Observe que existem duas chaves na criptografia assimétrica:

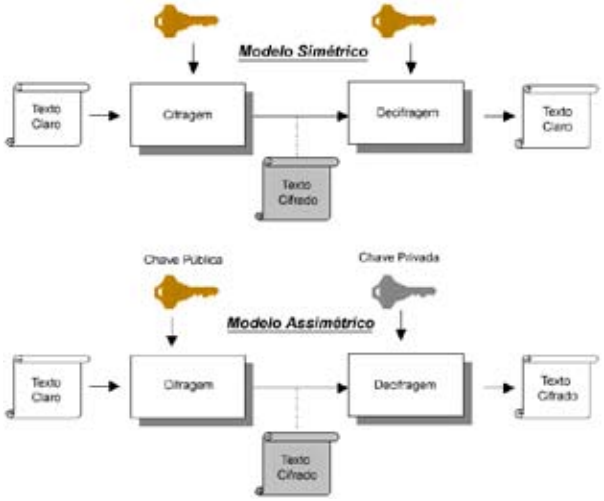

Figura 1.4 – Modelos simétrico e assimétrico de criptografia.

Após analisar o conteúdo exposto anteriormente, pode-se questionar qual modelo utilizar: simétrico ou assimétrico. Pois bem, em virtude dessa escolha foi desenvolvido um modelo híbrido, ou seja, que aproveitasse as vantagens de cada tipo de algoritmo. O algoritmo simétrico, por ser muito mais rápido, é utilizado no ciframento da mensagem em si, enquanto o assimétrico, embora lento, permite implementar a distribuição de chaves e é utilizado em aplicações de assinatura digital.

Então, os algoritmos criptográficos podem ser combinados para a implementação dos três mecanismos criptográficos básicos: o ciframento, a assinatura digital e o hashing, e estes últimos dois conceitos ainda serão descritos e analisados em capítulos posteriores, em especial, no capítulo 9. Esses mecanismos são componentes de protocolos criptográficos, embutidos na arquitetura de segurança dos produtos destinados ao comércio eletrônico. Esses protocolos criptográficos, portanto, provêm os serviços associados à criptografia que viabilizam o comércio eletrônico.

Descrevem-se, a seguir, alguns exemplos de protocolos que empregam sistemas criptográficos híbridos.

O IPSec é o padrão de protocolos criptográficos desenvolvidos para o IPv6. Realiza também o tunelamento de IP sobre IP. Tenciona-se adotá-lo como futuro padrão para todas as formas de VPN – Virtual Private Network (MAIA, 1999).

O SSL e TLS oferecem suporte de segurança criptográfica para os protocolos NTTP, HTTP, SMTP e Telnet. Permitem utilizar diferentes algoritmos simétricos, message digest e métodos de autenticação e gerência de chaves para algoritmos assimétricos (MAIA e PAGLIUSI, 1999). SSL (Security Socket Layer) é o protocolo mais conhecido em transações via WEB e hoje domina esse mercado, estando presente principalmente em vendas on-line envolvendo cartão de crédito. Foi criado pela Netscape, sendo o padrão livre para uso pessoal e empresarial. Ressalte-se novamente o nível de segurança desse protocolo, o qual assegura a inviolabilidade das vendas on-line com cartão de crédito (GEEK, 2002).

Pode-se citar também como exemplo o PGP, descrito anteriormente, lembrando-se de que se trata de um programa de criptografia famoso e bastante difundido na Internet, destinado à criptografia de e-mail. Suporta os algoritmos hashing MD5 e SHA-1.

O S/MIME (Secure Multipurpose Internet Mail Extensions) consiste em um esforço de um consórcio de empresas, tendo como um dos líderes a Microsoft, para adicionar segurança a mensagens eletrônicas no formato MIME. Acredita-se que o S/MIME deverá estabelecer-se no mercado corporativo, enquanto o PGP atuará no mundo do e-mail pessoal (MAIA, 1999).

Já o SET é um conjunto de padrões e protocolos para realizar transações financeiras seguras, como as realizadas com cartões de crédito na Internet. Oferece um canal seguro entre todos os envolvidos na transação, bem como autenticidade e privacidade entre as partes envolvidas.

A especificação X.509 define o relacionamento entre as autoridades de certificação, com base em criptografia de chave pública e assinatura digital (MAIA e PAGLIUSI, 1999).

Pode-se, então, comparar os modelos vistos (criptografias simétrica e assimétrica), conforme se mostra na tabela 1.4:

Tabela 1.4 – Comparação entre os tipos de algoritmos de criptografia

Criptografia simétrica	**Criptografia assimétrica**
Rápida	Lenta
Gerência e distribuição das chaves são complexas	Gerência e distribuição são simples
Não oferece assinatura digital	Oferece assinatura digital

A tabela 1.5 apresenta detalhes dos algoritmos simétricos mais conhecidos pela comunidade da área de segurança da informação e criptografia:

Tabela 1.5 – Características dos algoritmos simétricos mais conhecidos

Algoritmo	**Tipo**	**Tamanho da chave**	**Tamanho do bloco**
DES	Bloco	56	64
Triple DES (2 chaves)	Bloco	112	64
Triple DES (3 chaves)	Bloco	168	64
IDEA	Bloco	128	64
Blowfish	Bloco	32 a 448	64
RC5	Bloco	0 a 2.040	32, 64, 128
CAST-128	Bloco	40 a 128	64
RC2	Bloco	0 a 1.024	64
RC4	Stream (fluxo)	0 a 256	--
Rijndael (AES)	Bloco	128, 192, 256	128, 192, 256
MARS	Bloco	Variável	128
RC6	Bloco	Variável	128
Serpent	Bloco	Variável	128
Twofish	Bloco	128, 192, 256	128

Nessa tabela é possível observar que a maioria dos algoritmos criptográficos modernos, os mais usados na atualidade, baseia-se no sistema de cifrar as informações por bloco, cujo tamanho de bloco mais utilizado é de 64 bits e possui chaves de tamanho relativamente grande, superior a 56 bits. Outro aspecto que merece destaque é alguns algoritmos possuírem tamanhos de chave e/ou de bloco variáveis (ver situação dos algoritmos blowfish, RC5, CAST, RC2, RC4, Rijndael - AES, MARS, RC6, Serpent e Twofish, que casualmente foram os algoritmos que fizeram parte do projeto AES) (BIHAM, 1999).

1.4 Assinatura Digital

Alguns algoritmos criptográficos de chave-pública podem ser utilizados para gerar o que se denomina de assinaturas digitais. O algoritmo RSA é um desses algoritmos, assim, além da operação normal de cifrar com a chave-pública e decifrar com a chave-privada, permite que, cifrando-se com a chave-privada, o processo de decifrar com a chave-pública resulte na recuperação da mensagem (BUCHMANN, 2001).

Obviamente essa forma de uso não assegura o sigilo da mensagem, uma vez que qualquer um pode decifrá-la, dado que a chave-pública é de conhecimento público. Entretanto, se essa operação resulta na mensagem esperada, pode-se ter certeza de que somente o detentor da correspondente chave-privada poderia ter realizado a operação de cifragem.

Assim, uma assinatura digital é o criptograma resultante da cifração de um determinado bloco de dados (documento) pela utilização da chave-privada de quem assina em um algoritmo assimétrico. A verificação da assinatura é feita decifrando-se o criptograma (assinatura) com a suposta chave-pública correspondente. Se o resultado for válido, a assinatura é considerada válida, ou seja, autêntica, uma vez que apenas o detentor da chave privada, par da chave pública utilizada, poderia ter gerado esse criptograma. Na figura 1.5, ilustra-se esse procedimento:

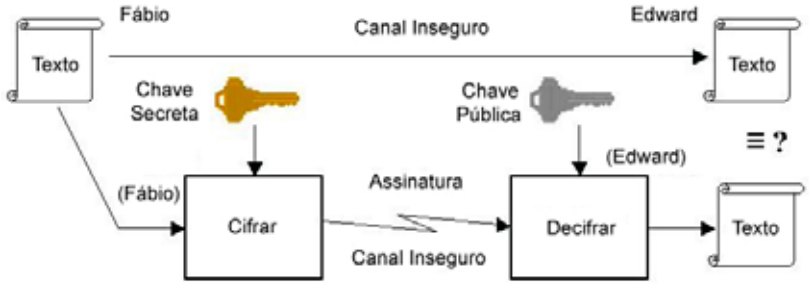

Figura 1.5 – Geração de assinatura digital de um documento (BURNETT, 2002).

Nesta figura, um usuário de nome Fábio assina um documento, cifrando-o com sua chave-privada e enviando tanto o documento original quanto a assinatura para um outro usuário chamado Edward. Este usuário verifica a assinatura decifrando-a com a chave-pública de Fábio (de conhecimento público) e comparando o resultado com o documento recebido. Se estiverem de acordo, a assinatura conferirá caso contrário será considerada inválida, significando que ou não foi Fábio quem assinou ou o documento foi adulterado após a assinatura. É interessante observar que esse procedimento é capaz de garantir tanto a origem (autenticação do emissor), tendo em vista que supostamente somente Fábio conhece sua chave privada e, portanto, somente ele é capaz de gerar uma assinatura que possa ser verificada com sua chave-pública, como também a integridade do documento, já que se este for alterado, a verificação da assinatura irá indicar a adulteração caso tenha vindo efetivamente do pretenso emissor.

Em geral, diante da ineficiência dos algoritmos assimétricos no computador, os métodos para assinatura digital empregados na prática não assinam o documento que se deseja autenticar em si, mas uma súmula deste, obtida pelo seu processamento por meio do que se denomina função de hashing, que é uma função criptográfica que gera uma saída de tamanho fixo (geralmente 128 a 256 bits) independentemente do tamanho da entrada. Essa saída se denomina de hash da mensagem (ou documento ou o que quer que seja a entrada). Segundo Burnett e Paine (2002), para ter utilidade criptográfica, a função de hashing deve ser:

- simples (eficiente, rápida), se computar o hash de dada mensagem;
- impraticável, se determinar a entrada a partir de seu hash.
- impraticável, se determinar uma outra entrada que resulte no mesmo hash de uma dada entrada.

Valores de hash possíveis são estatisticamente eqüiprováveis.

A tabela 1.6 apresenta as características dos algoritmos de hashing mais conhecidos pela comunidade da área de segurança de dados.

Tabela 1.6 – Características dos algoritmos de hashing mais conhecidos (GUELFI, 2002)

Algoritmo de hash	Tamanho hash	Kbytes/s
Abreast Bavies-Meyer (c/IDEA)	128	22
Davies-Meyer (c/DES)	64	9
GOST-Hash	256	11
NAVAL (3 passos)	Variável	168
NAVAL (4 passos)	Variável	118
NAVAL (5 passos)	Variável	95
MD4 – Message Digest 4	128	236
MD5 – Message Digest 5	128	174
N-HASH (12 rounds)	128	29
N-HASH (15 rounds)	128	24
RIPE-MD	128	182
RIPE-MD-160	160	--
SHA – Secure Hash Algorithm	160	75
SNEFRU (4 passos)	128	48
SNEFRU (8 passos)	128	23

Nessa tabela é possível observar que a maioria dos algoritmos de hashing utiliza um tamanho de hash fixo, com destaque para 128 e 160 bits, que são os tamanhos mais usados pela comunidade. Os algoritmos mais conhecidos pela comunidade acadêmica e com utilidade comercial são SHA, MD4 e MD5. Já os outros, como o nome indicado na tabela 1.6 sugere, pertencem a aplicações governamentais e de uso privado, como, por exemplo, aplicações militares e navais.

1.5 Considerações Finais

Ainda no contexto da visão atual de criptografia, pode-se elencar outros fatores que impulsionam seu estudo e evolução:

- A tecnologia de protocolos de rede tornará a Internet uma rota vulnerável às invasões e ao furto de informações.

- Crackers de todas as partes estão dispostos a destruir até mesmo os mais sólidos negócios on-line, possuindo armas intelectuais para isso.
- E-commerce começou a perder adeptos, por ser suscetível a fraudes em operações envolvendo transferência de dinheiro e cartões de crédito.
- Correio eletrônico também não é seguro.
- Diversos serviços on-line aproveitam-se da ingenuidade do internauta para invadir um microcomputador e obter dados pessoais.
- Pelo computador é muito mais fácil e provável a aplicação de métodos de monitoramento.
- Ainda há um certo sentimento de insegurança quando se realiza alguma transação comercial via Internet, fazendo com que não se aproveite todo o potencial desse tipo de comércio.
- A invasão de máquinas/sistemas e roubo de informação é uma realidade, deixando de lado a imagem fictícia e cinematográficas.
- Os algoritmos de criptografia ainda são um conceito distante da grande massa de usuários, apresentando-se como algo abstrato e intangível.

Por esses motivos, este livro além de apresentar de forma clara os algoritmos criptográficos mais utilizados no momento, preocupa-se também com dados de desempenho: tempos de execução do processo de cifragem e decifragem, de processamento de alguns algoritmos simétricos (DES, AES, IDEA, RC5) e assimétricos, como o RSA e funções de hashing (MD5 e SHA-1), quando implementados em software (usando-se a linguagem C e alguns deles em Java) e hardware (em circuitos programáveis – FPGAs por meio da linguagem especial para descrição de sistemas digitais, conhecida como VHDL).

É fundamental destacar ao leitor que ao aumentar o tamanho da chave, incrementam-se o nível de segurança e também o tempo de processamento do processo de cifragem e decifragem. Este livro mostra qual é o real impacto desse processo, como os impactos do algoritmo no tempo de processamento, do sistema operacional (Windows, Linux), do processador etc.