

Criptografia e Certificação Digital

Escrito por Luiz Paulo Maia (lpmaia@training.com.br) e Paulo Sergio Pagliusi (pagliusi@vento.com.br)
em: http://www.training.com.br/lpmaia/pub_seg_cripto.htm.

Este artigo apresenta os principais conceitos envolvendo criptografia e certificação digital, fundamentais para a compreensão e implementação do comércio eletrônico seguro na Internet. O artigo aborda:

- Serviços oferecidos pela criptografia
- Criptografia simétrica
- Criptografia assimétrica Assinatura digital
- Função hashing
- Certificado digital

Serviços Oferecidos

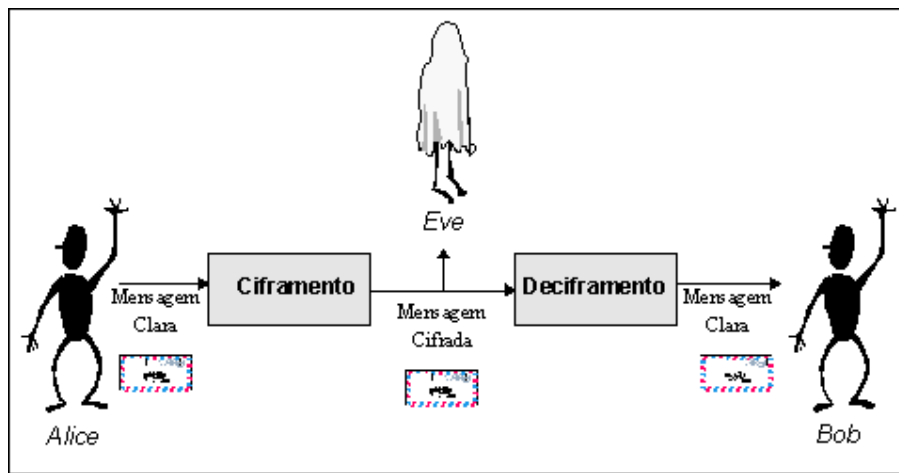
As técnicas de criptografia oferecem seis tipos de serviços básicos. Sem estes predicados não é possível realizar o comércio eletrônico seguro na Internet:

Serviços	Descrição
Disponibilidade	Garante que uma informação estará disponível para acesso no momento desejado.
Integridade	Garante que o conteúdo da mensagem não foi alterado.
Controle de acesso	Garante que o conteúdo da mensagem somente será acessado por pessoas autorizadas.
Autenticidade da origem	Garante a identidade de quem está enviando a mensagem.
Não-repudição	Previne que alguém negue o envio e/ou recebimento de uma mensagem.
Privacidade (confidencialidade ou sigilo)	Impede que pessoas não autorizadas tenham acesso ao conteúdo da mensagem, garantindo que apenas a origem e o destino tenham conhecimento.

Se imaginarmos uma compra pela Internet, podemos perceber a necessidade de todos os requisitos acima. Por exemplo, a informação que permite a transação - tais como valor e descrição do produto adquirido - precisa estar disponível no dia e na hora que o cliente deseja efetuar-la (disponibilidade), o valor da transação não pode ser alterado (integridade), somente o cliente que está comprando e o comerciante devem ter acesso à transação (controle de acesso), o cliente que está comprando deve ser realmente quem diz ser (autenticidade), o cliente tem como provar o pagamento e o comerciante não tem como negar o recebimento (não-repúdio) e o conhecimento do conteúdo da transação fica restrito aos envolvidos (privacidade).

Criptografia Simétrica

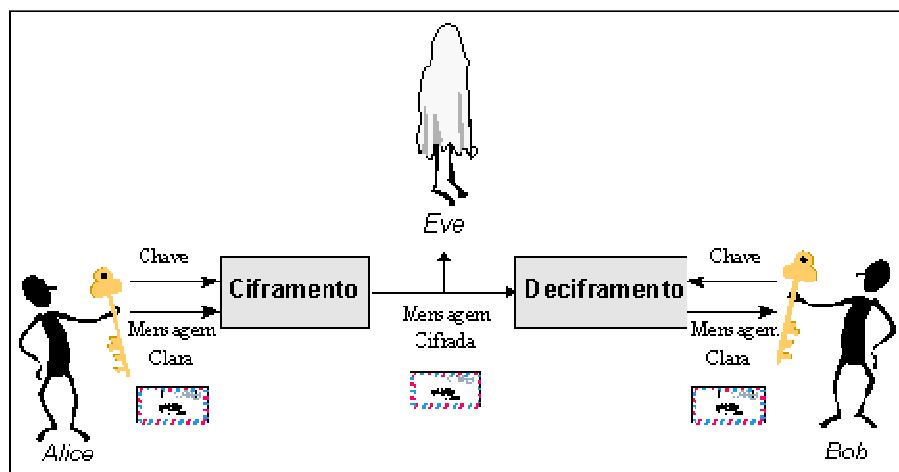
O ciframento de uma mensagem baseia-se em dois componentes: um *algoritmo* e uma *chave*. Um *algoritmo* é uma transformação matemática. Ele converte uma mensagem em claro em uma mensagem cifrada e vice-versa. Quando *Alice* (origem) cifra uma mensagem, ela utiliza um algoritmo de ciframento para transformar o conteúdo em claro da mensagem em texto cifrado. Quando *Bob* (destinatário) decifra uma mensagem, ele utiliza o algoritmo de deciframento correspondente para converter o texto cifrado de novo em uma mensagem clara.



Antigamente, a segurança do ciframento estava baseada somente no sigilo do algoritmo criptográfico. Se *Eve* (um intruso) conhecesse o algoritmo sem chave, poderia decifrar uma mensagem cifrada tão facilmente quanto *Bob*. Pode-se contornar o problema apresentado utilizando o segundo componente básico da criptografia de mensagens: a chave. Uma chave é uma cadeia aleatória de bits utilizada em conjunto com um algoritmo. Cada chave distinta faz com que o algoritmo trabalhe de forma ligeiramente diferente.

Embora existam algoritmos que dispensem o uso de chaves, sua utilização oferece duas importantes vantagens. A primeira é permitir a utilização do mesmo algoritmo criptográfico para a comunicação com diferentes receptores, apenas trocando a chave. A segunda vantagem é permitir trocar facilmente a chave no caso de uma violação, mantendo o mesmo algoritmo.

O número de chaves possíveis depende do tamanho (número de bits) da chave. Por exemplo, uma chave de 8 bits permite uma combinação de no máximo 256 chaves (2^8). Quanto maior o tamanho da chave, mais difícil quebra-la, pois estamos aumentando o número de combinações.



Quando *Alice* cifra uma mensagem, ela utiliza um algoritmo de ciframento e uma chave secreta para transformar uma mensagem clara em um texto cifrado. *Bob*, por sua vez, ao decifrar uma mensagem, utiliza o algoritmo de deciframento correspondente e a mesma chave para transformar o texto cifrado em uma mensagem em claro. *Eve*, por não possuir a chave secreta, mesmo conhecendo o algoritmo, não conseguirá decifrar a mensagem. A segurança do sistema passa a residir não mais no algoritmo e sim na chave empregada. É ela que agora, no lugar do algoritmo, deverá ser mantida em segredo por *Alice* e *Bob*.

Quando a chave de ciframento é a mesma utilizada para deciframento ou esta última pode facilmente ser obtida a partir do conhecimento da primeira, ambas precisam ser compartilhadas previamente entre origem e destinatário, antes de se estabelecer o canal criptográfico desejado, utilizando-se um canal seguro e independente do destinado à comunicação sigilosa. Este tipo de ciframento emprega a criptografia conhecida como *simétrica* ou de chave secreta.

Algoritmo Simétrico	Bits	Descrição
DES	56	<p>O Data Encryption Standard (DES) é o algoritmo simétrico mais disseminado no mundo. Foi criado pela IBM em 1977 e, apesar de permitir cerca de 72 quadrilhões de combinações (2^{56}), seu tamanho de chave (56 bits) é considerado pequeno, tendo sido quebrado por "força bruta" em 1997 em um desafio lançado na Internet.</p> <p>O NIST (National Institute of Standards and Technology), que lançou o desafio mencionado, recertificou o DES pela última vez em 1993 e desde então está recomendando o 3DES. O NIST está também propondo um substituto ao DES que deve aceitar chaves de 128, 192 e 256 bits, operar com blocos de 128 bits, ser eficiente, flexível e estar livre de "royalties".</p> <p>O novo padrão, denominado AES (Advanced Encryption Standard), está sendo estudado desde 1997 a partir de vários algoritmos apresentados pela comunidade. Os finalistas são: Serpent, Mars, RC6, Twofish e Rijndael, e o resultado deverá ser divulgado no final de 2000.</p>
Triple DES	112 ou 168	O 3DES é uma simples variação do DES, utilizando-o em três ciframentos sucessivos, podendo empregar um versão com duas ou com três chaves diferentes. É seguro, porém muito lento para ser um algoritmo padrão.
IDEA	128	O International Data Encryption Algorithm foi criado em 1991 por James Massey e Xuejia Lai e possui patente da suíça ASCOM Systec. O algoritmo é estruturado seguindo as mesmas linhas gerais do DES. Mas na maioria dos microprocessadores, uma implementação por <i>software</i> do IDEA é mais rápida do que uma implementação por <i>software</i> do DES. O IDEA é utilizado principalmente no mercado financeiro e no PGP, o programa para criptografia de e-mail pessoal mais disseminado no mundo.
Blowfish	32 a 448	Algoritmo desenvolvido por Bruce Schneier, que oferece a escolha entre maior segurança ou desempenho através de chaves de tamanho variável. O autor aperfeiçoou-o no Twofish, concorrente ao AES.
RC2	8 a 1024	Projetado por Ron Rivest (o R da empresa RSA Data Security Inc.) e utilizado no protocolo S/MIME, voltado para criptografia de e-mail corporativo. Também possui chave de tamanho variável. Rivest também é o autor do RC4, RC5 e RC6, este último concorrente ao AES.

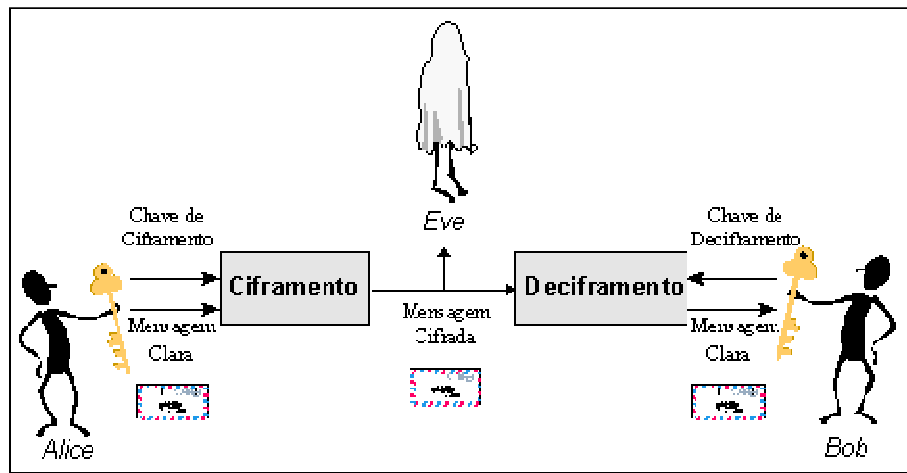
Apesar de sua simplicidade, existem alguns problemas na criptografia simétrica:

- Como cada par necessita de uma chave para se comunicar de forma segura, para um uma rede de n usuários precisaríamos de algo da ordem de n^2 chaves, quantidade esta que dificulta a gerência das chaves;

- A chave deve ser trocada entre as partes e armazenada de forma segura, o que nem sempre é fácil de ser garantido;
- A criptografia simétrica não garante a identidade de quem enviou ou recebeu a mensagem (autenticidade e não-repudição).

Criptografia Assimétrica

A maneira de contornar os problemas da criptografia simétrica é a utilização da criptografia assimétrica ou de chave pública. A criptografia assimétrica está baseada no conceito de par de chaves: uma chave privada e uma chave pública. Qualquer uma das chaves é utilizada para cifrar uma mensagem e a outra para decifrá-la. As mensagens cifradas com uma das chaves do par só podem ser decifradas com a outra chave correspondente. A chave privada deve ser mantida secreta, enquanto a chave pública disponível livremente para qualquer interessado.



De uma forma simplificada, o sistema funciona assim: *Bob* e todos os que desejam comunicar-se de modo seguro geram uma chave de ciframento e sua correspondente chave de deciframento. Ele mantém secreta a chave de deciframento; esta é chamada de sua *chave privada*. Ele torna pública a chave de ciframento: esta é chamada de sua *chave pública*.

A chave pública realmente condiz com seu nome. Qualquer pessoa pode obter uma cópia dela. *Bob* inclusive encoraja isto, enviando-a para seus amigos ou publicando-a em boletins. Assim, *Eve* não tem nenhuma dificuldade em obtê-la. Quando *Alice* deseja enviar uma mensagem a *Bob*, precisa primeiro encontrar a chave pública dele. Feito isto, ela cifra sua mensagem utilizando a chave pública de *Bob*, despachando-a em seguida. Quando *Bob* recebe a mensagem, ele a decifra facilmente com sua chave privada. *Eve*, que interceptou a mensagem em trânsito, não conhece a chave privada de *Bob*, embora conheça sua chave pública. Mas este conhecimento não o ajuda a decifrar a mensagem. Mesmo *Alice*, que foi quem cifrou a mensagem com a chave pública de *Bob*, não pode decifrá-la agora.

A grande vantagem deste sistema é permitir que qualquer um possa enviar uma mensagem secreta, apenas utilizando a chave pública de quem irá recebê-la. Como a chave pública está amplamente disponível, não há necessidade do envio de chaves como é feito no modelo simétrico. A confidencialidade da mensagem é garantida, enquanto a chave privada estiver segura. Caso contrário, quem possuir acesso à chave privada terá acesso às mensagens.

Seguem descrições de alguns algoritmos assimétricos:

Algoritmo	Descrição
RSA	O RSA é um algoritmo assimétrico que possui este nome devido a

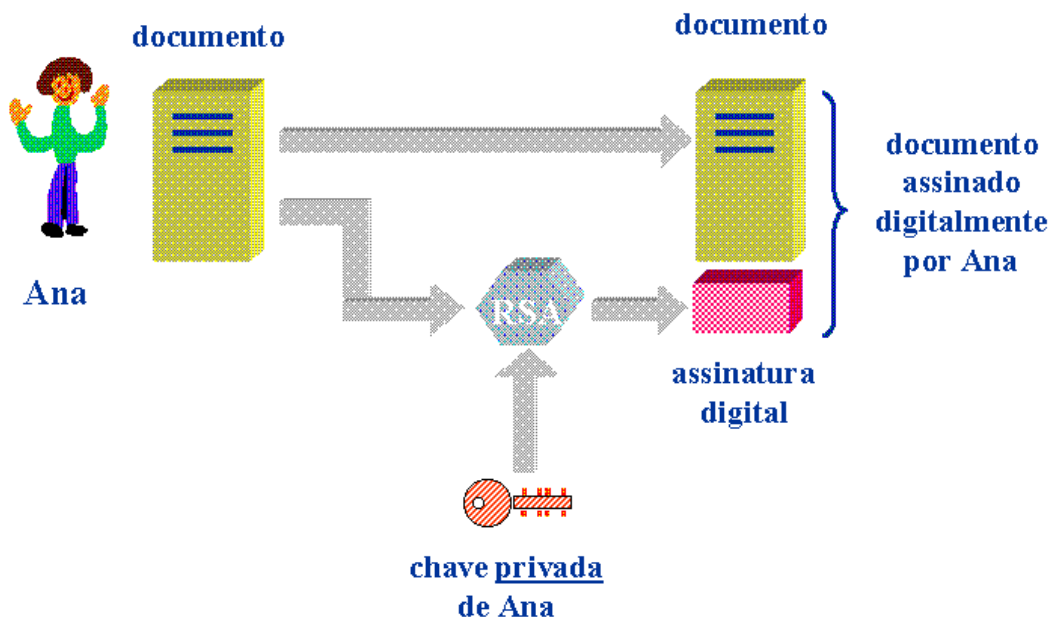
	<p>seus inventores: Ron Rivest, Adi Shamir e Len Adleman, que o criaram em 1977 no MIT. É, atualmente, o algoritmo de chave pública mais amplamente utilizado, além de ser uma das mais poderosas formas de criptografia de chave pública conhecidas até o momento. O RSA utiliza números primos.</p> <p>A premissa por trás do RSA é que é fácil multiplicar dois números primos para obter um terceiro número, mas muito difícil recuperar os dois primos a partir daquele terceiro número. Isto é conhecido como <i>fatoração</i>. Por exemplo, os fatores primos de 3.337 são 47 e 71. Gerar a chave pública envolve multiplicar dois primos grandes; qualquer um pode fazer isto. Derivar a chave privada a partir da chave pública envolve fatorar um grande número. Se o número for grande o suficiente e bem escolhido, então ninguém pode fazer isto em uma quantidade de tempo razoável. Assim, a segurança do RSA baseia-se na dificuldade de fatoração de números grandes. Deste modo, a fatoração representa um limite superior do tempo necessário para quebrar o algoritmo.</p> <p>Uma chave RSA de 512 bits foi quebrada em 1999 pelo Instituto Nacional de Pesquisa da Holanda, com o apoio de cientistas de mais 6 países. Levou cerca de 7 meses e foram utilizadas 300 estações de trabalho para a quebra. Um fato preocupante: cerca de 95% dos sites de comércio eletrônico utilizam chaves RSA de 512 bits.</p>
ElGamal	<p>O ElGamal é outro algoritmo de chave pública utilizado para gerenciamento de chaves. Sua matemática difere da utilizada no RSA, mas também é um sistema comutativo. O algoritmo envolve a manipulação matemática de grandes quantidades numéricas. Sua segurança advém de algo denominado problema do logaritmo discreto. Assim, o ElGamal obtém sua segurança da dificuldade de se calcular logaritmos discretos em um corpo finito, o que lembra bastante o problema da fatoração.</p>
Diffie-Hellman	<p>Também baseado no problema do logaritmo discreto, e o criptosistema de chave pública mais antigo ainda em uso. O conceito de chave pública aliás foi introduzido pelos autores deste criptosistema em 1976. Contudo, ele não permite nem ciframento nem assinatura digital. O sistema foi projetado para permitir a dois indivíduos entrarem em um acordo ao compartilharem um segredo tal como uma chave, muito embora eles somente troquem mensagens em público.</p>
Curvas Elípticas	<p>Em 1985, Neal Koblitz e V. S. Miller propuseram de forma independente a utilização de curvas elípticas para sistemas criptográficos de chave pública. Eles não chegaram a inventar um novo algoritmo criptográfico com curvas elípticas sobre corpos finitos, mas implementaram algoritmos de chave pública já existentes, como o algoritmo de Diffie e Hellman, usando curvas elípticas. Assim, os sistemas criptográficos de curvas elípticas consistem em modificações de outros sistemas (o ElGamal, por exemplo), que passam a trabalhar no domínio das curvas elípticas, em vez de trabalharem no domínio dos corpos finitos. Eles possuem o potencial de proverem sistemas criptográficos de chave pública mais seguros, com chaves de menor tamanho.</p> <p>Muitos algoritmos de chave pública, como o Diffie - Hellman, o ElGamal e o Schnorr podem ser implementados em curvas elípticas sobre corpos finitos. Assim, fica resolvido um dos maiores problemas dos algoritmos de chave pública: o grande tamanho de suas chaves. Porém, os algoritmos de curvas elípticas atuais, embora possuam o potencial de serem rápidos, são em geral mais demorados do que o RSA.</p>

Assinatura Digital

Outro benefício da criptografia com chave pública é a assinatura digital, que permite garantir a autenticidade de quem envia a mensagem, associada à integridade do seu conteúdo. Por exemplo, suponha que Alice (origem) queira comunicar o nascimento de sua filha para todos os seus amigos (destinatários = Bob), mas queira garantir aos mesmos que a mensagem foi enviada realmente por ela. E, embora não se importe com o sigilo da mensagem, deseja que a mesma chegue íntegra aos destinatários, sem alterações como, por exemplo, do sexo da criança.

Alice então cifra a mensagem com sua chave privada e a envia, em um processo denominado de assinatura digital. Cada um que receber a mensagem deverá decifrá-la, ou seja, verificar a validade da assinatura digital, utilizando para isso a chave pública de Alice. Como a chave pública de Alice apenas decifra (ou seja, verifica a validade de) mensagens cifradas com sua chave privada, fica garantida assim a autenticidade, integridade e não-repudição da mensagem. Pois se alguém modificar um bit do conteúdo da mensagem ou se outra pessoa assiná-la ao invés de Alice, o sistema de verificação não irá reconhecer a assinatura digital de Alice como sendo válida.

É importante perceber que a assinatura digital, como descrita no exemplo anterior, não garante a confidencialidade da mensagem. Qualquer um poderá acessá-la e verificá-la, mesmo um intruso (Eve), apenas utilizando a chave pública de Alice. Para obter confidencialidade com assinatura digital, basta combinar os dois métodos. Alice primeiro assina a mensagem, utilizando sua chave privada. Em seguida, ela criptografa a mensagem novamente, junto com sua assinatura, utilizando a chave pública de Bob. Este, ao receber a mensagem, deve, primeiramente, decifrá-la com sua chave privada, o que garante sua privacidade. Em seguida, "decifrá-la" novamente, ou seja, verificar sua assinatura utilizando a chave pública de Alice, garantindo assim sua autenticidade.



Segue a descrição de alguns algoritmos utilizados para assinatura digital:

Algoritmo	Descrição
RSA	Como já mencionado, o RSA também é comutativo e pode ser utilizado para a geração de assinatura digital. A matemática é a mesma: há uma chave pública e uma chave privada, e a segurança

	do sistema baseia-se na dificuldade da fatoração de números grandes.
ElGamal	Como o RSA, o ElGamal também é comutativo, podendo ser utilizado tanto para assinatura digital quanto para gerenciamento de chaves; assim, ele obtém sua segurança da dificuldade do cálculo de logaritmos discretos em um corpo finito.
DSA	O Digital Signature Algorithm, unicamente destinado a assinaturas digitais, foi proposto pelo NIST em agosto de 1991, para utilização no seu padrão DSS (Digital Signature Standard). Adotado como padrão final em dezembro de 1994, trata-se de uma variação dos algoritmos de assinatura ElGamal e Schnorr. Foi inventado pela NSA e patentado pelo governo americano.

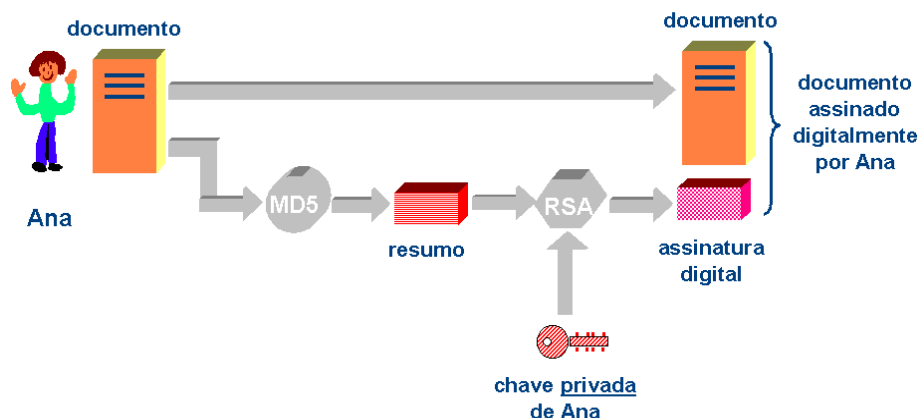
Função Hashing

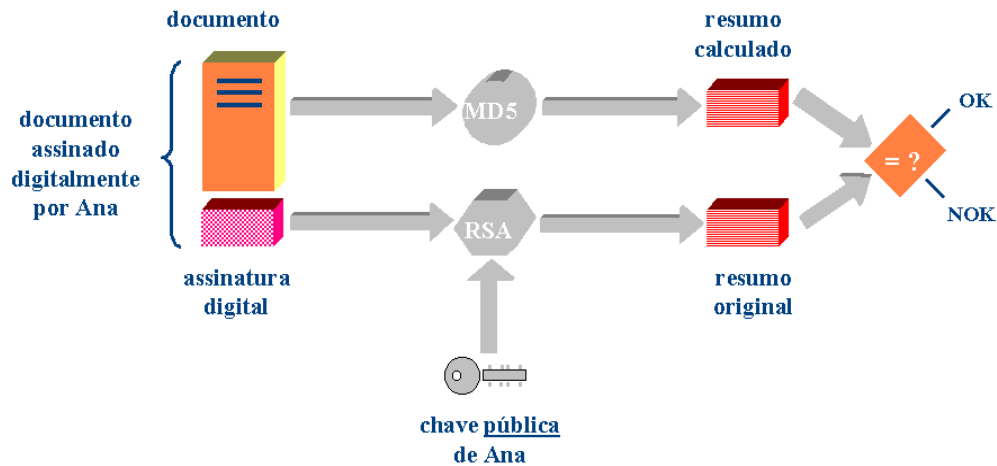
A assinatura digital obtida através do uso da criptografia assimétrica ou de chave pública infelizmente não pode ser empregada, na prática, de forma isolada, do modo como foi didaticamente descrito no item anterior. Está faltando, portanto, descrever um mecanismo fundamental para o adequado emprego da assinatura digital. Este mecanismo é a função Hashing. Sua utilização como componente de assinaturas digitais se faz necessário devido à lentidão dos algoritmos assimétricos, em geral cerca de 1.000 vezes mais lentos do que os simétricos.

Assim, na prática é inviável e contraproducente utilizar puramente algoritmos de chave pública para assinaturas digitais, principalmente quando se deseja assinar grandes mensagens, que podem levar preciosos minutos ou mesmo horas para serem integralmente "cifradas" com a chave privada de alguém. Ao invés disso, é empregada uma função Hashing, que gera um valor pequeno, de tamanho fixo, derivado da mensagem que se pretende assinar, de qualquer tamanho. Assim, a função Hashing oferece agilidade nas assinaturas digitais, além de integridade confiável, conforme descrito a seguir.

Também denominada Message Digest, One-Way Hash Function, Função de Condensação ou Função de Espalhamento Unidirecional, a função Hashing funciona como uma impressão digital de uma mensagem gerando, a partir de uma entrada de tamanho variável, um valor fixo pequeno: o digest ou valor hash.

Este valor está para o conteúdo da mensagem assim como o dígito verificador de uma conta-corrente está para o número da conta ou o check sum está para os valores que valida. Serve, portanto, para garantir a integridade do conteúdo da mensagem que representa. Assim, após o valor hash de uma mensagem ter sido calculado através do emprego de uma função hashing, qualquer modificação em seu conteúdo -mesmo em apenas um bit da mensagem - será detectada, pois um novo cálculo do valor hash sobre o conteúdo modificado resultará em um valor hash bastante distinto.





Segue a descrição de algumas funções Hashing empregadas em produtos e protocolos criptográficos:

Funções	Descrição
MD5	É uma função de espalhamento unidirecional inventada por Ron Rivest, do MIT, que também trabalha para a RSA Data Security. A sigla MD significa Message Digest. Este algoritmo produz um valor hash de 128 bits, para uma mensagem de entrada de tamanho arbitrário. Foi inicialmente proposto em 1991, após alguns ataques de criptoanálise terem sido descobertos contra a função Hashing prévia de Rivest: a MD4. O algoritmo foi projetado para ser rápido, simples e seguro. Seus detalhes são públicos, e têm sido analisados pela comunidade de criptografia. Foi descoberta uma fraqueza em parte do MD5, mas até agora ela não afetou a segurança global do algoritmo. Entretanto, o fato dele produzir um valor hash de somente 128 bits é o que causa maior preocupação; é preferível uma função Hashing que produza um valor maior.
SHA-1	O Secure Hash Algorithm, uma função de espalhamento unidirecional inventada pela NSA, gera um valor hash de 160 bits, a partir de um tamanho arbitrário de mensagem. O funcionamento interno do SHA-1 é muito parecido com o observado no MD4, indicando que os estudiosos da NSA basearam-se no MD4 e fizeram melhorias em sua segurança. De fato, a fraqueza existente em parte do MD5, citada anteriormente, descoberta após o SHA-1 ter sido proposto, não ocorre no SHA-1. Atualmente, não há nenhum ataque de criptoanálise conhecido contra o SHA-1. Mesmo o ataque da força bruta torna-se impraticável, devido ao seu valor hash de 160 bits. Porém, não há provas de que, no futuro, alguém não possa descobrir como quebrar o SHA-1.
MD2 e MD4	O MD4 é o precursor do MD5, tendo sido inventado por Ron Rivest. Após terem sido descobertas algumas fraquezas no MD4, Rivest escreveu o MD5. O MD4 não é mais utilizado. O MD2 é uma função de espalhamento unidirecional simplificada, e produz um hash de 128 bits. A segurança do MD2 é dependente de uma permutação aleatória de bytes. Não é recomendável sua utilização, pois, em geral, é mais lento do que as outras funções hash citadas e acredita-se que seja menos seguro.

Criptografia Simétrica x Assimétrica: Protocolos Criptográficos

Qual o modelo de criptografia que devemos utilizar? Simétrico ou assimétrico? A resposta é simples: devemos utilizar os dois, em um modelo denominado híbrido. 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 a assinatura digital. Além disso, como já exposto no item anterior, deve-se utilizar também o mecanismo de Hashing para complemento da assinatura digital.

Criptografia Simétrica.	Criptografia Assimétrica.
Rápida.	Lenta.
Gerência e distribuição das chaves é complexa.	Gerência e distribuição simples.
Não oferece assinatura digital	Oferece assinatura digital.

Em resumo, os algoritmos criptográficos podem ser combinados para a implementação dos três mecanismos criptográficos básicos: o ciframento, a assinatura e o Hashing. Estes mecanismos são componentes dos protocolos criptográficos, embutidos na arquitetura de segurança dos produtos destinados ao comércio eletrônico. Estes protocolos criptográficos, portanto, provêm os serviços associados à criptografia que viabilizam o comércio eletrônico: disponibilidade, sigilo, controle de acesso, autenticidade, integridade e não-repúdio.

Seguem exemplos de protocolos que empregam sistemas criptográficos híbridos:

Protocolo	Descrição
IPSec	Padrão de protocolos criptográficos desenvolvidos para o IPv6. Realiza também o tunelamento de IP sobre IP. É composto de três mecanismos criptográficos: Authentication Header (define a função Hashing para assinatura digital), Encapsulation Security Payload (define o algoritmo simétrico para ciframento) e ISAKMP (define o algoritmo assimétrico para Gerência e troca de chaves de criptografia). Criptografia e tunelamento são independentes. Permite Virtual Private Network fim-a-fim. Futuro padrão para todas as formas de VPN.
SSL e TLS	Oferecem suporte de segurança criptográfica para os protocolos NTTP, HTTP, SMTP e Telnet. Permitem utilizar diferentes algoritmos simétricos, message digest (hashing) e métodos de autenticação e gerência de chaves (assimétricos).
PGP	Inventado por Phil Zimmermann em 1991, é um programa criptográfico famoso e bastante difundido na Internet, destinado a criptografia de e-mail pessoal. Algoritmos suportados: hashing: MD5, SHA-1, simétricos: CAST-128, IDEA e 3DES, assimétricos: RSA, Diffie-Hellman/DSS. Versão mais recente: 6.5.3.
S/MIME	O S/MIME (Secure Multipurpose Internet Mail Extensions) consiste em um esforço de um consórcio de empresas, liderado pela RSADSI e pela Microsoft, para adicionar segurança a mensagens eletrônicas no formato MIME. Apesar do S/MIME e PGP serem ambos padrões Internet, o S/MIME deverá se estabelecer no mercado corporativo, enquanto o PGP no mundo do mail pessoal.
SET	O SET é um conjunto de padrões e protocolos, para realizar transações financeira seguras, como as realizadas com cartão de crédito na Internet. Oferece um canal de comunicação seguro entre todos os envolvidos na transação. Garante autenticidade X.509v3 e privacidade entre as partes.
X.509	Recomendação ITU-T, a especificação X.509 define o relacionamento entre as autoridades de certificação. Faz parte das séries X.500 de recomendações para uma estrutura de diretório global, baseada em

nomes distintos para localização. Utilizado pelo S/MIME, IPSec, SSL/TLS e SET. Baseado em criptografia com chave pública (RSA) e assinatura digital (com hashing).
--

Certificado Digital

Com um sistema de chave pública, o gerenciamento de chaves passa a ter dois novos aspectos: primeiro, deve-se previamente localizar a chave pública de qualquer pessoa com quem se deseja comunicar e, segundo, deve-se obter uma garantia de que a chave pública encontrada seja proveniente daquela pessoa (*Bob*). Sem esta garantia, um intruso *Eve* pode convencer os interlocutores (*Alice* e *Bob*) de que chaves públicas falsas pertencem a eles. Estabelecendo um processo de confiança entre os interlocutores, *Eve* pode fazer-se passar por ambos.

Deste modo, quando um interlocutor (*Alice*) enviar uma mensagem ao outro (*Bob*) solicitando sua chave pública, o intruso poderá interceptá-la e devolver-lhe uma chave pública forjada por ele. Ele também pode fazer o mesmo com o recebedor (*Bob*), fazendo com que cada lado pense que está se comunicando com o outro, quando na verdade estão sendo interceptados pelo intruso. *Eve* então pode decifrar todas as mensagens, cifrá-las novamente ou, se preferir, pode até substituí-las por outras mensagens. Através deste ataque, um intruso pode causar tantos danos ou até mais do que causaria se conseguisse quebrar o algoritmo de ciframento empregado pelos interlocutores.

A garantia para evitar este ataque é representada pelos *certificados de chave pública*. Tais certificados consistem em chaves públicas assinadas por uma pessoa de confiança, geralmente no formato padrão ITU X.509v3. Servem para evitar tentativas de substituição de uma chave pública por outra. O certificado de *Bob* contém algo mais do que sua chave pública: contém informações sobre *Bob* - seu nome, endereço e outras dados pessoais - e é assinado por alguém em quem *Alice* deposita sua confiança: uma *autoridade de certificação* ou *CA* (*Certification Authority*), que funciona como um cartório eletrônico.

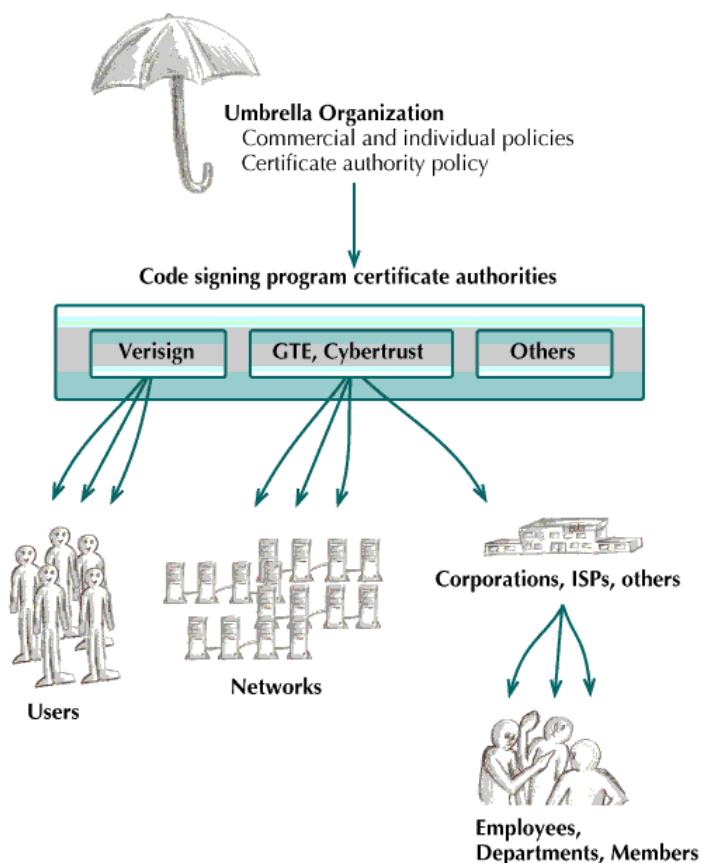
versão
número de série
algoritmo utilizado
nome X.500 da CA
nome X.500 do detentor
período de validade
extensões
chave pública do detentor
assinatura da CA

Assim, um certificado digital pode ser definido como um documento eletrônico, assinado digitalmente por uma terceira parte confiável, que associa o nome (e atributos) de uma pessoa ou instituição a uma chave criptográfica pública.

Pela assinatura da chave pública e das informações sobre *Bob*, a CA garante que a informação sobre *Bob* está correta e que a chave pública em questão realmente

pertence a *Bob*. *Alice*, por sua vez, confere a assinatura da CA e então utiliza a chave pública em pauta, segura de que esta pertence a *Bob* e a ninguém mais. Certificados desempenham um importante papel em um grande número de protocolos e padrões utilizados na proteção de sistemas de comércio eletrônico.

Autoridades de certificação, como Verisign, Cybertrust e Nortel, assinam certificados digitais garantindo sua validade. Uma CA também tem a responsabilidade de manter e divulgar uma lista com os certificados revogados (Certificate Revocation List - CRL). Certificados nesta lista podem ter sido roubados, perdidos ou, simplesmente, estar sem utilidade. As CAs podem estar encadeadas em hierarquias de certificação, onde a CA de um nível inferior valida sua assinatura com a assinatura de uma CA mais alta na hierarquia.



Existem diversos tipos de certificados, conforme descrição feita a seguir.

- Certificados de CA: utilizados para validar outros certificados; são auto-assinados ou assinados por outra CA.
- Certificados de servidor: utilizados para identificar um servidor seguro; contém o nome da organização e o nome DNS do servidor.
- Certificados pessoais: contém nome do portador e, eventualmente, informações como endereço eletrônico, endereço postal, etc.
- Certificados de desenvolvedores de software: utilizados para validar assinaturas associadas a programas.

A infra-estrutura para lidar com o gerenciamento de chaves públicas é definido pelo padrão Public Key Infrastructure (PKI), que define onde os certificados digitais serão armazenados e recuperados, de que forma estão armazenados, como um certificado é revogado, entre outras informações.

Referências:

Security and Electronic Commerce, David Kosiur, Microsoft Press, 1997.

<http://mspress.microsoft.com/prod/books/sampchap/1252.htm>

Cryptography and Network Security 2 ed., William Stallings, Prentice Hall, 1998.

Tese de Mestrado: "Introdução de Mecanismos de Segurança em Sistemas de Correio Eletrônico"

Paulo Sergio Pagliusi, orientada por Cláudio Leonardo Lucchesi e Luiz Eduardo Buzato,
UNICAMP, 1998.

Bruce Schneier: E-Mail Security - How To Keep Your Electronic Messages Private.
John Wiley & Sons, Inc., EUA, 1995.

Bruce Schneier: Applied Cryptography. John Wiley & Sons, Inc., EUA, 1996.

Sites e Links interessantes: www.rsa.com; www.pgpi.com; e www.hackers.com.br.