Capítulo 4

Assinaturas e certificados

Este capítulo apresenta uma introdução aos mecanismos de assinatura digital e aos certificados de chave pública. Este texto não tem a mínima pretensão de ser completo sobre esse vasto tema; leitores em busca de uma abordagem mais profunda e completa devem procurar livros específicos sobre criptografia.

4.1 Resumo criptográfico

Um *resumo criptográfico* (*cryptographic hash*) [Menezes et al., 1996] é uma função $y = hash(x)$ que gera uma sequência de bytes y de tamanho pequeno e fixo (algumas dezenas ou centenas de bytes) a partir de um conjunto de dados x de tamanho variável aplicado como entrada. Os resumos criptográficos são frequentemente usados para identificar unicamente um arquivo ou outra informação digital, ou para atestar sua integridade: caso o conteúdo de um documento digital seja modificado, seu resumo também será alterado.

Em termos matemáticos, os resumos criptográficos são um tipo de *função unidirecional* (*one-way function*). Uma função $f(x)$ é chamada unidirecional quando seu cálculo direto ($y = f(x)$) é rápido, mas o cálculo de sua inversa ($x = f^{-1}(y)$) é impossível ou computacionalmente inviável. Um exemplo clássico de função unidirecional é a fatoração do produto de dois números primos muito grandes, como os usados no algoritmo RSA. Considerando a função $f(p, q) = p \times q$, onde p e q são inteiros primos, calcular $y = f(p, q)$ é simples e rápido, mesmo se p e q forem grandes. Entretanto, fatorar y para obter de volta os primos p e q pode ser computacionalmente inviável, se y tiver muitos dígitos[1].

Os algoritmos de resumo criptográfico mais conhecidos e utilizados atualmente são os da família SHA (*Secure Hash Algorithms*). Os algoritmos MD5 e SHA1 foram muito utilizados, mas se mostraram inseguros a colisões e não devem mais ser utilizadom para aplicações criptográficas; seu uso continua viável em outras aplicações, como a detecção de réplicas de arquivos [Menezes et al., 1996; Stamp, 2011]. No Linux, comandos como `md5sum` e `sha1sum` permitem calcular respectivamente resumos criptográficos de arquivos:

[1] Em 2014, um grupo de pesquisadores conseguiu fatorar o inteiro $2^{1199} - 1$ (que tem 361 dígitos), em um projeto que consumiu cerca de 7.500 anos-CPU.

```
~:> md5sum livro.*
371f456d68720a3c0ba5950fe2708d37  livro.pdf
d4a593dc3d44f6eae54fc62600581b11  livro.tex

~:> sha1sum livro.*
9664a393b533d5d82cfe505aa3ca12410aa1f3b7  livro.pdf
d5bd8d809bb234ba8d2289d4fa13c319e227ac25  livro.tex

~:> sha224sum livro.*
36049e03abf47df178593f79c3cdd0c018406232a0f300d872351631  livro.pdf
edac4154fe0263da86befa8d5072046b96b75c2f91764cc6b5b2f5c0  livro.tex

~:> sha256sum livro.*
c5fc543d1758301feacdc5c6bfd7bc12ef87036fbc589a902856d306cb999d50  livro.pdf
da5075006c6f951e40c9e99cef3218d8a2d16db28e746d0f4e4b18cf365a8099  livro.tex
```

Uma boa função de resumo criptográfico deve gerar sempre a mesma saída para a mesma entrada ($hash(m_1) = hash(m_2) \iff m_1 = m_2$) e saídas diferentes para entradas diferentes (($hash(m_1) \neq hash(m_2) \iff m_1 \neq m_2$). No entanto, como o número de bytes do resumo é pequeno, podem ocorrer *colisões*: duas entradas distintas m_1 e m_2 gerando o mesmo resumo ($m_1 \neq m_2$ mas $hash(m_1) = hash(m_2)$). Idealmente, uma função de *hash* criptográfico deve apresentar as seguintes propriedades:

Determinismo: para uma dada entrada m, a saída é sempre a mesma.

Rapidez: o cálculo de $x = hash(m)$ é rápido para qualquer m.

Resistência à pré-imagem: dado um valor de x, é difícil encontrar m tal que $x = hash(m)$ (ou seja, a função é difícil de inverter).

Resistência à colisão: é difícil encontrar duas mensagens quaisquer $m_1 \neq m_2$ tal que $hash(m_1) = hash(m_2)$.

Espalhamento: uma modificação em um trecho específico dos dados de entrada m gera modificações em várias partes do resumo $hash(m)$.

Sensibilidade: uma pequena mudança nos dados de entrada m (mesmo um só bit) gera mudanças significativas no resumo $hash(m)$.

4.2 Assinatura digital

Os algoritmos de criptografia assimétrica e resumos criptográficos previamente apresentados permitem efetuar a *assinatura digital* de documentos eletrônicos. A assinatura digital é uma forma de verificar a autoria e integridade de um documento, sendo por isso o mecanismo básico utilizado na construção dos *certificados digitais*, amplamente empregados para a autenticação de servidores na Internet.

Em termos gerais, a assinatura digital de um documento consiste de um resumo digital do mesmo, cifrado usando a chave privada de seu autor (ou de quem o está

assinando). Sendo um documento d emitido pelo usuário u, sua assinatura digital $s(d, u)$ é definida por:

$$s(d, u) = \{\ hash(d)\ \}_{kv(u)}$$

onde $hash(x)$ é uma função de resumo criptográfico conhecida, $\{x\}_k$ indica a cifragem de x usando uma chave k e $kv(u)$ é a chave privada do usuário u. Para verificar a validade da assinatura, basta calcular novamente o resumo $r' = hash(d)$ e compará-lo com o resumo obtido da assinatura, decifrada usando a chave pública de u ($r'' = \{s\}^{-1}_{kp(u)}$). Se ambos forem iguais ($r' = r''$), o documento foi realmente assinado por u e está íntegro, ou seja, não foi modificado desde que u o assinou [Menezes et al., 1996].

A Figura 4.1 ilustra o processo de assinatura digital e verificação de um documento. Os passos do processo são:

1. Alice divulga sua chave pública kp_a em um repositório acessível publicamente;
2. Alice calcula o resumo digital r do documento d a ser assinado;
3. Alice cifra o resumo r usando sua chave privada kv_a, obtendo uma assinatura digital s;
4. A assinatura s e o documento original d, em conjunto, constituem o documento assinado por Alice: $[d, s]$;
5. Bob obtém o documento assinado por Alice ($[d', s']$, com $d' = d$ e $s' = s$ se ambos estiverem íntegros);
6. Bob recalcula o resumo digital $r' = hash(d')$ do documento, usando o mesmo algoritmo empregado por Alice;
7. Bob obtém a chave pública kp_a de Alice e a usa para decifrar a assinatura s' do documento, obtendo um resumo r'' ($r'' = r$ se s foi realmente cifrado com a chave kv_a e se $s' = s$) ;
8. Bob compara o resumo r' do documento com o resumo r'' obtido da assinatura digital; se ambos forem iguais ($r' = r''$), o documento foi assinado por Alice e está íntegro, assim como sua assinatura.

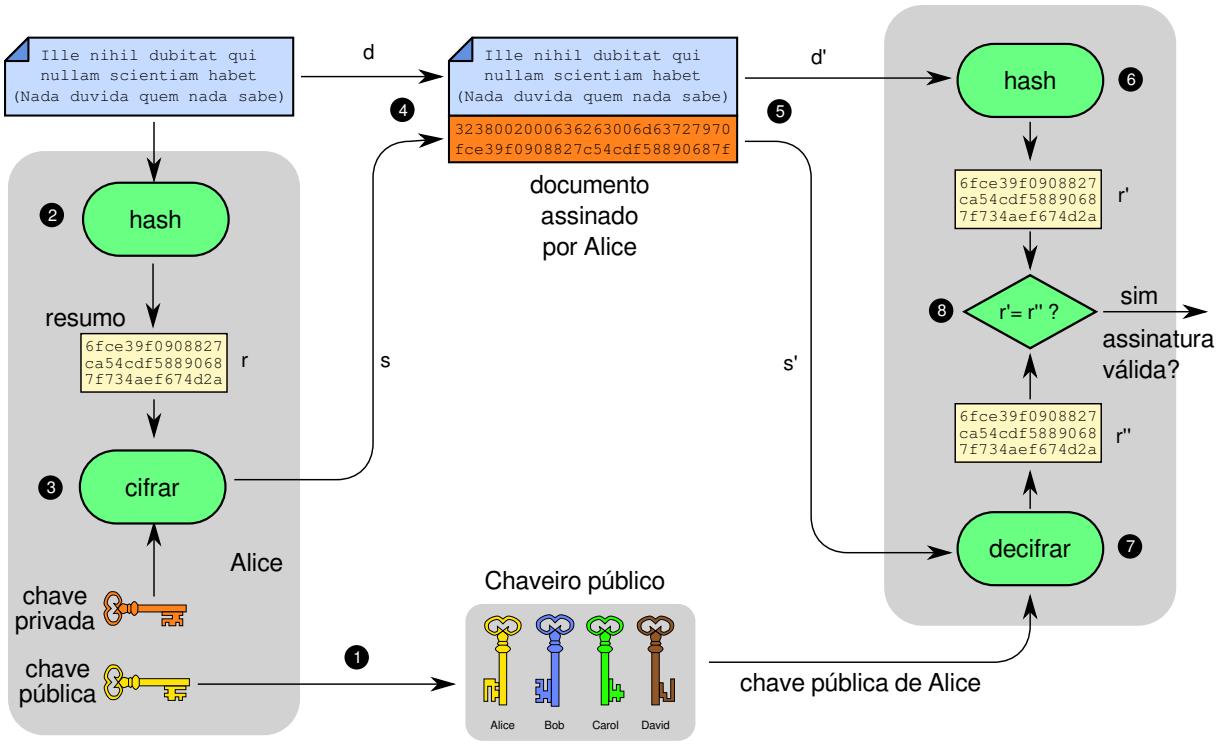

Figura 4.1: Assinatura e verificação de uma assinatura digital.

4.3 Certificado de chave pública

A identificação confiável do proprietário de uma chave pública é fundamental para o funcionamento correto das técnicas de criptografia assimétrica e de assinatura digital. Uma chave pública é composta por uma mera sequência de bytes que não permite a identificação direta de seu proprietário. Por isso, torna-se necessária uma estrutura complementar para fazer essa identificação. A associação entre chaves públicas e seus respectivos proprietários é realizada através dos *certificados digitais*. Um certificado digital é um documento digital assinado, composto das seguintes partes [Menezes et al., 1996]:

- Identidade do proprietário do certificado (nome, endereço, e-mail, URL, número IP e/ou outras informações que permitam identificá-lo unicamente)[2];
- Chave pública do proprietário do certificado;
- Identificação da entidade que emitiu/assinou o certificado;
- Outras informações, como período de validade do certificado, algoritmos de criptografia e resumos utilizados, etc.;
- Uma ou mais assinaturas digitais do conteúdo, emitidas por entidades consideradas confiáveis pelos usuários do certificado.

[2]Deve-se ressaltar que um certificado pode pertencer a um usuário humano, a um sistema computacional ou qualquer módulo de software que precise ser identificado de forma inequívoca.

Dessa forma, um certificado digital "amarra" uma identidade a uma chave pública. Para verificar a validade de um certificado, basta usar a chave pública da entidade que o assinou. Existem vários tipos de certificados digitais com seus formatos e conteúdos próprios, sendo os certificados PGP e X.509 aqueles mais difundidos [Mollin, 2000]. Os certificados no padrão X509 são extensivamente utilizados na Internet para a autenticação de chaves públicas de servidores Web, de e-mail, etc. Um exemplo de certificado X.509, destacando sua estrutura básica e principais componentes, é apresentado na Figura 4.2.

```
Certificate Data:
   Version: 3 (0x2)
   Serial Number: 05:f1:3c:83:7e:0e:bb:86:ed:f8:c4:9b
   Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign Extended Validation CA-SHA256-G3
   Validity                                                          informações básicas
      Not Before: Feb  7 12:41:03 2017 GMT
      Not After : May  9 23:59:59 2018 GMT
   Subject: businessCategory=Private Organization/serialNumber=00.000.000/7297-44/
            jurisdictionC=BR, C=BR, ST=Distrito Federal, L=Brasilia/
            street=ST STN SN QD 716 CONJ C EDIF SEDE IV ANDAR 1 ASA NORTE,   proprietário
            OU=DITEC, O=Banco do Brasil S.A., CN=www2.bancobrasil.com.br     do certificado
   Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
         Public-Key: (2048 bit)
         Modulus:                                                    chave pública
            00:db:4a:0e:92:da:5b:f3:38:3f:d5:63:9d:6d:f9:            do proprietário
            91:6c:16:fc:24:84:28:e8:aa:86:aa:9c:a3:aa:1a:            do certificado
            2e:b6:09:74:6a:f8:1e:31:4a:60:81:0f:ac:76:59:
            ... (linhas omitidas)
            8e:0b
         Exponent: 65537 (0x10001)
   X509v3 extensions:
      X509v3 Key Usage: critical                                     campos opcionais
         Digital Signature, Key Encipherment
      Authority Information Access:
         CA Issuers - URI:http://secure.globalsign.com/cacert/gsextendvalsha2g3r3.crt
         OCSP - URI:http://ocsp2.globalsign.com/gsextendvalsha2g3r3
      X509v3 Extended Key Usage:
         TLS Web Server Authentication, TLS Web Client Authentication
Signature Algorithm: sha256WithRSAEncryption
   94:8e:14:c6:38:30:78:77:80:fc:92:f1:5b:8b:72:6a:b6:b6:
   95:66:c5:7b:ba:be:51:a4:b8:8a:f5:37:0a:4a:74:4d:82:27:          assinatura do emissor
   ... (linhas omitidas)
   b6:44:e8:8c
```

Figura 4.2: Certificado digital no padrão X.509.

4.4 Infraestrutura de chaves públicas

Todo certificado deve ser assinado por alguma entidade considerada confiável pelos usuários do sistema. Essas entidades são normalmente denominadas *Autoridades*

Certificadoras (AC ou CA – *Certification Authorities*). Como as chaves públicas das ACs devem ser usadas para verificar a validade de um certificado, surge um problema: como garantir que uma chave pública realmente pertence a uma dada autoridade certificadora?

A solução para esse problema é simples: basta criar um certificado para essa AC, assinado por outra AC ainda mais confiável. Dessa forma, pode-se construir uma estrutura hierárquica de certificação, na qual a AC mais confiável (denominada "AC raiz") assina os certificados de outras ACs, e assim sucessivamente, até chegar aos certificados dos servidores, usuários e demais entidades do sistema. Uma estrutura de certificação dessa forma se chama *Infraestrutura de Chaves Públicas* (ICP ou PKI - *Public-Key Infrastructure*). Em uma ICP convencional (hierárquica), a chave pública da AC raiz deve ser conhecida de todos e é considerada íntegra [Mollin, 2000].

A Figura 4.3 traz um exemplo de infraestrutura de chaves públicas hierárquica. A chave pública AC raiz (vermelha) é usada para assinar os certificados das chaves verde e azul, e assim por diante. Reciprocamente, o certificado de chave roxo depende da confiança na chave azul, que por sua vez depende da confiança na chave vermelha. A sequência de certificados *roxo* → *azul* → *vermelho* é chamada de **cadeia de certificação** ou *cadeia de confiança*.

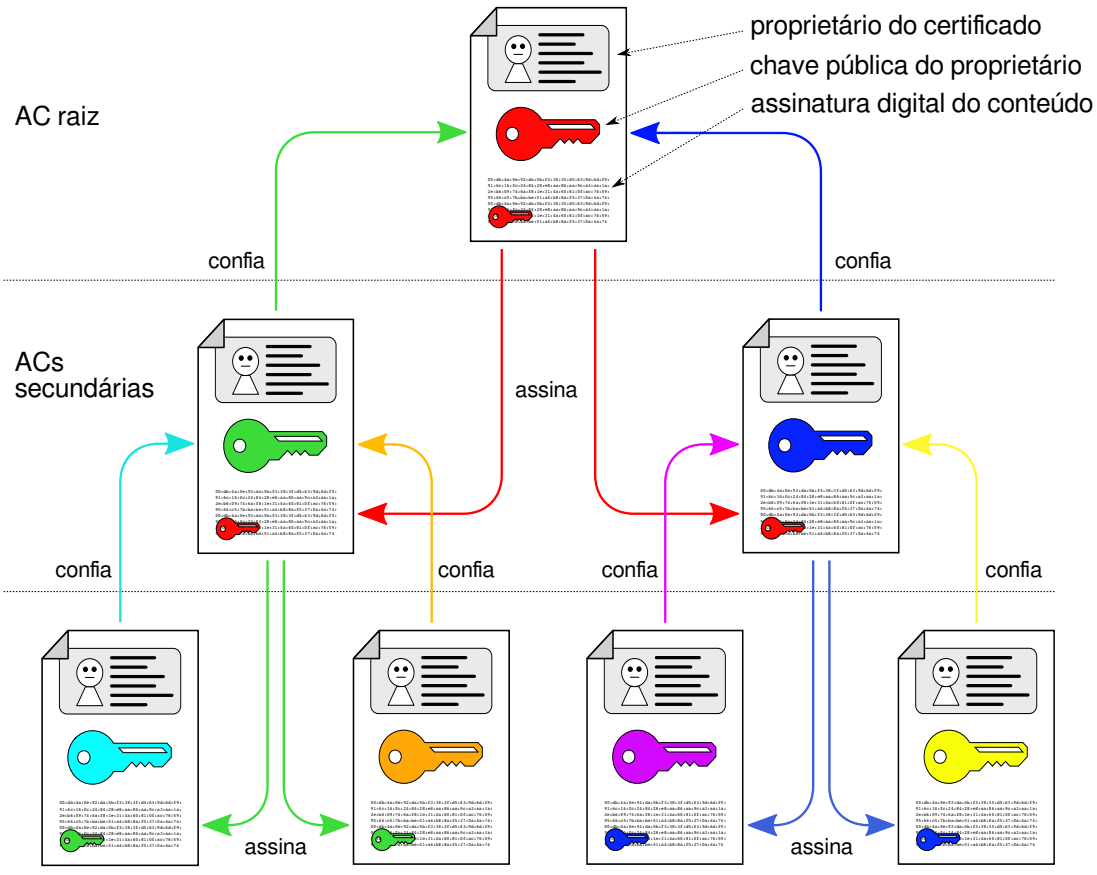

Figura 4.3: Infraestrutura de chaves públicas hierárquica.

O campo *Validity* de um certificado X509 (ver Figura 4.2) diz respeito ao seu prazo de validade, ou seja, o período de tempo em que o certificado é considerado válido. Entretanto, em algumas situações pode ser necessário **revogar certificados** antes do prazo final de validade. Casos típicos de revogação envolvem o vazamento da chave

privada do usuário ou da de alguma autoridade certificadora na cadeia de certificação que valida o certificado, ou então situações mais banais, como a cessação de atividade da empresa proprietária do certificado ou mudanças na finalidade do certificado (campos opcionais `Key Usage` na Figura 4.2).

Existem dois mecanismos básicos para revogar certificados: as CRLs - *Certificate Revocation Lists* são listas de certificados revogados mantidas pelas autoridades certificadoras, que pode ser descarregadas pelo software cliente através de um acesso HTTP. Contudo, em autoridades certificadoras populares, as CRLs podem conter muitos certificados e se tornar muito grandes. Por isso, mais recentemente foi definido o OCSP - *Online Certificate Status Protocol*, que permite ao software consultar junto à CA o status de um certificado digital específico.

Exercícios

1. Recentemente foi noticiado na imprensa que certificados digitais emitidos pela Autoridade Certificadora holandesa *DigiNotar* haviam sido falsificados e estavam sendo usados por um governo do oriente médio para monitorar as comunicações de seus cidadãos. Considerando o certificado falso do serviço de e-mails do *Google* (`mail.google.com`), explique:

   (a) Neste contexto, em que consiste um certificado falso?

   (b) Qual a utilidade de um certificado falso na interceptação de comunicações?

   (c) Por que somente os usuários do navegador *Chrome* (produzido pelo próprio *Google*) detectaram o certificado falso, enquanto usuários de outros navegadores não perceberam nada?

2. O provedor de conteúdo TOL (*Tabajara OnLine*) decidiu implementar um novo mecanismo de segurança em suas páginas web. Esse mecanismo consiste em adicionar uma etiqueta oculta (*HTML tag*) em cada página, contendo o nome do autor (*name*), a data de produção (*date*) e uma assinatura digital *s*. Essa assinatura é constituida pelo hash criptográfico do nome do autor e da data ($hash(name + date)$), cifrado usando a chave privada do autor da página. O conteúdo da página Web em si não é cifrado. As chaves públicas dos autores registrados podem ser obtidas em `http://www.tol.com.br/pubkeys.html`.

   Responda:

   (a) Que objetivo tinham em mente os proponentes desse mecanismo?

   (b) Esse esquema é seguro? Por que?

   (c) Se o esquema não for seguro, indique um possível ataque ao mesmo; caso seja seguro, explique por que esse mesmo ataque não funcionaria.

Referências

A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

R. A. Mollin. *An Introduction to Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 2000. ISBN 1584881275.

M. Stamp. *Information Security - Principles and Practice, 2nd edition*. Wiley, 2011.