Infraestruturas de Chave Pública

Criptografia

António Santos

ISTEC

Tópicos

Definição

O PKI (Public Key Infrastructure) é um sistema de recursos, políticas e serviços que suportam a utilização de criptografia de chave pública para autenticar as partes envolvidas.
Uma PKI é um órgão, iniciativa pública ou privada que tem como objetivo manter uma estrutura de emissão de chaves públicas, baseando-se no princípio da terceira parte confiável, oferecendo uma mediação de credibilidade e confiança em transações entre partes que utilizam certificados digitais. A principal função do PKI é definir um conjunto de técnicas, práticas e procedimentos a serem adotados pelas entidades a fim de estabelecer um sistema de certificação digital baseado em chave pública.

Definição

- Emissão de certificados digitais
- Validação de certificados digitais
- Revogação de certificados digitais
- Distribuição de chaves públicas

Vantagens PKI

Garantia de sigilo e privacidade - Quando se visita um site "seguro"da web, o computador recebe o certificado contendo a chave pública desse site, o que é suficiente para criar um túnel criptográfico, tornando os dados incompreensíveis durante o tráfego, sendo possível apenas ao servidor web recuperar a informação original.
Controle de acesso a aplicativos - O servidor web pode solicitar ao utilizador que apresente um certificado digital, em vez de digitar utilizador e password. Os utilizadores não poderão colocar em perigo a aplicação por falta de cuidado no uso e armazenamento da password.

Vantagens PKI

Assinatura de formulários e impossibilidade de repúdio - Os utilizadores poderão assinar os formulários que submetem preenchidos na web da mesma forma que fariam pessoalmente num serviço. Além disso, qualquer documento digital passa a valer como documento assinado, com validade jurídica, dispensando-se o uso de papel.

Garantia de sigilo e privacidade - O sistema de email utilizado para troca de mensagens através da Internet não possui recursos nativos para impedir a violação da correspondência eletrónica. Com o uso de certificados digitais, pode-se selar a correspondência num envelope digital criptográfico e certificar-se de que apenas o destinatário será capaz de compreender seu conteúdo.

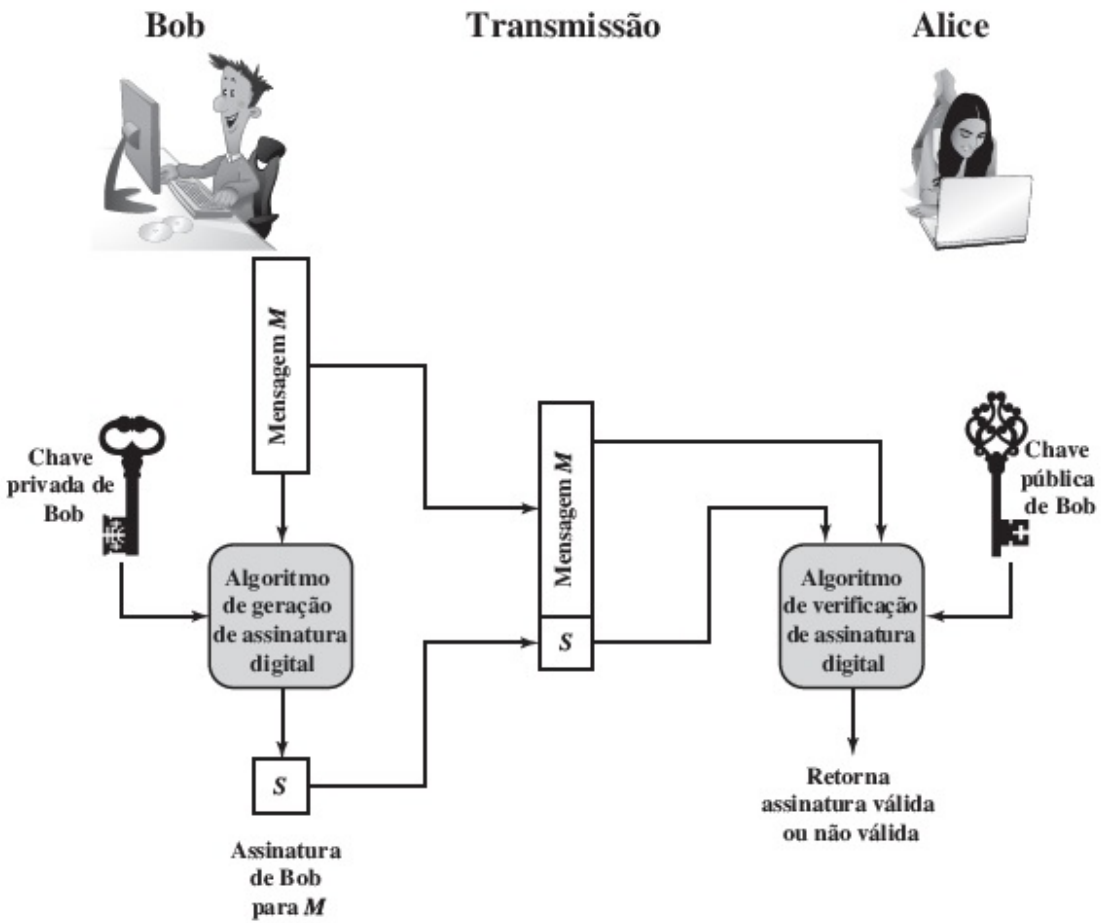
Vantagens PKI

Identificação do remetente - Não existirá mais dúvidas sobre a origem de uma mensagem, pois será possível certificar-se da identidade do emissor.

Assinatura Digital

- **Autenticidade**: quem recebe a mensagem consegue confirmar que a assinatura foi feita por quem a emitiu;
- **Integridade**: se ocorre qualquer alteração no documento, a assinatura automaticamente não corresponde mais a ele;
- **Irretratabilidade ou não-repúdio**: quem emite a mensagem não pode voltar atrás ou negar a autenticidade dela.

Assinatura Digital



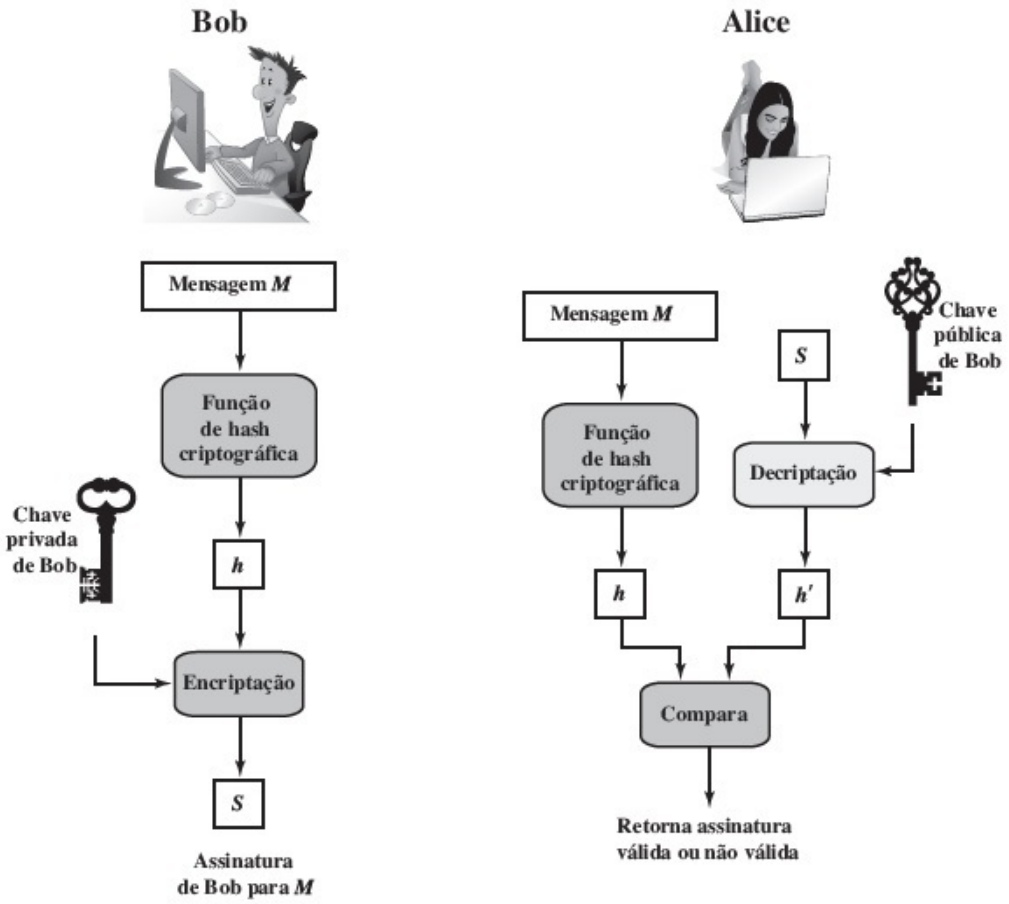
Propriedades

1. A Alice pode forjar uma mensagem diferente e reivindicar que ela veio de Bob. Ela simplesmente teria que criar uma mensagem e anexar um código de autenticação usando a chave que eles compartilham.
2. Bob pode negar o envio da mensagem. Como é possível que Alice falsifique uma mensagem, não há como provar que ele realmente a enviou.

Propriedades

- Verificar o autor, a data e a hora da assinatura.
- Autenticar o conteúdo no momento da assinatura.
- Ser verificável por terceiros, para resolver disputas.

Propriedades



Ataques e falsificações

- Ataque somente de chave: C só conhece a chave pública de A.
- Ataque de mensagem conhecida: C recebe acesso a um conjunto de mensagens e suas assinaturas.
- Ataque de mensagem escolhida genérica: C escolhe uma lista de mensagens antes de tentar quebrar o esquema de assinatura de A, independente da chave pública de A. C, então, obtém de A assinaturas válidas para as mensagens escolhidas. O ataque é genérico, pois não depende da chave pública de A; o mesmo ataque é usado contra todos.

Ataques e falsificações

- Ataque de mensagem escolhida direcionada: semelhante ao ataque genérico, exceto que a lista de mensagens a serem assinadas é escolhida depois que C conhece a chave pública de A, mas antes que quaisquer assinaturas sejam vistas.
- Ataque de mensagem escolhida adaptativa: C tem permissão para usar A como um "oráculo". Isso significa que C pode solicitar de A assinaturas de mensagens que dependam de pares mensagem-assinatura previamente obtidos.

Ataques e falsificações

- Quebra total: C determina a chave privada de A.
- Falsificação universal: C encontra um algoritmo de assinatura eficiente que oferece um modo equivalente de construção de assinaturas sobre mensagens arbitrárias.
- Falsificação seletiva: C forja uma assinatura para determinada mensagem escolhida por C.
- Falsificação existencial: C forja uma assinatura para pelo menos uma mensagem. C não tem controle sobre a mensagem. Consequentemente, essa falsificação pode ser somente um pequeno incômodo para A.

Requisitos de assinatura digital

- A assinatura precisa ser um padrão de bits que depende da mensagem a ser assinada.
- A assinatura precisa usar alguma informação exclusiva do emissor, para impedir falsificação e negação.
- É preciso ser relativamente fácil produzir a assinatura digital.
- É preciso ser relativamente fácil reconhecer e verificar a assinatura digital.

Requisitos de assinatura digital

- É preciso ser computacionalmente inviável falsificar uma assinatura digital, seja construindo uma nova mensagem para uma assinatura digital existente ou uma assinatura digital fraudulenta para determinada mensagem.
- É preciso ser prático reter uma cópia da assinatura digital em termos de armazenamento.

Assinatura digital direta

O termo assinatura digital direta refere-se a um esquema de assinatura digital que envolve apenas as partes em comunicação (origem, destino). Considera-se que o destino conhece a chave pública da origem.
A confidencialidade pode ser fornecida pela encriptação da mensagem inteira mais a assinatura com uma chave secreta (encriptação simétrica).

Assinatura digital direta

Assinatura de um documento: Quando se utiliza uma assinatura convencional, esta assinatura está fisicamente ligada ao documento que foi assinado. Quando utilizamos uma assinatura digital, esta não está fisicamente ligada ao documento, sendo assim necessário que seja utilizado um algoritmo/aplicação que de alguma maneira deve ligar a assinatura ao documento que queremos assinar.

Assinatura digital direta

Verificação da assinatura: Numa assinatura convencional, a verificação da assinatura é verificada quando esta é comparada com uma assinatura autenticada (por exemplo, a assinatura do Cartão do Cidadão ou de um cartão de crédito). As assinaturas digitais são verificadas utilizando um algoritmo de verificação público. Qualquer pessoa pode verificar se a assinatura digital é autentica.

Assinatura com RSA

As assinaturas digitais do RSA possuem as mesmas características da criptografia e decriptografia de mensagens. Apesar de possuir uma fragilidade caso seja mal implementado, aplicar esta ferramenta garante uma maior segurança aos utilizadores e ao próprio receptor, afinal de contas o utilizador não poderá negar que assinou a mensagem.

Assinatura com RSA

Sejam (n_a, e_a) e (p_a, q_a, d_a) as chaves pública e privada RSA de Alice, respectivamente; e (n_b, e_b) e (p_b, q_b, d_b) as chaves pública e privada RSA de Bob, respectivamente.
Suponhamos que a Alice pretende enviar uma mensagem cifrada e assinada a Bob. Dada a mensagem original x, primeiro a Alice assina x utilizando a sua chave de RSA privada, d_a, obtendo

$$y = sig_{d_a}(x) \equiv x^{d_a} \ mod \ n_a.$$

Assinatura com RSA

Em seguida, cifram-se x e y utilizando a chave pública RSA de Bob, obtendo a mensagem cifrada z que transmite a Bob. Quando Bob recebe z, ele primeiro decifra z utilizando a sua chave privada RSA e obtém (x, y). Para certificar a autenticidade da assinatura y, Bob utiliza a chave pública RSA de Alice verificando se a seguinte congruência é verdadeira

$$y^{e_a} \equiv x \ mod \ n_a$$

.

Assinatura com RSA

Se Alice cifrar primeiro x, obtendo z, e só depois assinar z, obtendo y e enviar o par (z, y) a Bob, então Trudy poderá criar a sua assinatura y' de z e substituir a assinatura y de Alice, enviando o par (z, y') a Bob. Neste caso, Bob ficará convencido que quem lhe enviou a mensagem x foi Trudy. Note-se que nesta situação, Trudy sabendo ou não a mensagem x, consegue sempre assinar z. Por esta razão é recomendado que se assine sempre a mensagem antes de a cifrar.

Assinatura ElGamal

A assinatura digital ElGamal foi descrita pela primeira vez em 1985 e foi desenvolvida especificamente para ser uma assinatura, o que contrasta com o RSA que pode ser usado como um sistema criptográfico ou como uma assinatura. Uma modificação desta assinatura deu origem à assinatura digital standard adoptada pelo National Institute of Standards and Technology (NIST).

Assinatura ElGamal

Seja p um primo tal que a resolução computacional do problema do logaritmo discreto em $\mathbb{Z}_p$ é impraticável. Sejam α uma raiz primitiva de p, $1 < a < p - 1$ um valor aleatório e $\beta \equiv \alpha^a \ mod \ p$. Os valores p, α e β são públicos e a é secreto. Seja $K = (p, \alpha, a, \beta)$ a chave ElGamal de Bob. Para assinar uma mensagem x, gera-se aleatoriamente $k \in Z^*_{p-1}$ (k deve ser mantido secreto) e define-se como $sig_K(x, k) = (\gamma, \delta)$, onde $\gamma \equiv \alpha^k \ mod \ p$ e $\delta \equiv (x - a\gamma)k^{-1} \ mod \ p - 1$.

Assinatura ElGamal

A função de verificação é dada por
$ver_K(x, \gamma, \delta) = 1 \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \ mod \ p$.
Esta verificação está correcta porque
$\beta^\gamma \gamma^\delta \equiv \alpha^{a\gamma} \alpha^{k\delta} \equiv \alpha^x \ mod \ p$, porque $a\gamma + k\delta \equiv x \ mod \ p - 1$.
O Bob calcula a sua assinatura utilizando o valor secreto a, que faz parte da sua chave e o valor secreto k (que só deve ser alterado sempre que se quer assinar uma mensagem x). A verificação é obtida usando somente informação pública.

Assinatura ElGamal

Exemplo. Seja $p = 467$, $\alpha = 2$ e $a = 127$. Então $\beta \equiv 2^{127} \equiv 132 \ mod \ 467$.
Suponhamos que Bob quer assinar a mensagem $x = 100$.
Primeiro gera o número aleatório $k = 213$ que é primo com 466 (como tinha de ser). Então $213^{-1} \equiv 431 \ mod \ 466$. A assinatura de (x, k) passa a ser $(29, 51)$, porque $\gamma \equiv 2^{213} \equiv 29 \ mod \ 467$ e $\delta \equiv (100 - 127 \bullet 29) \bullet 431 \equiv 51 \ mod \ 466$.
Para verificar a assinatura basta calcular $132^{29}29^{51} \equiv 189 \ mod \ 467$ e $2^{100} \equiv 189 \ mod \ 467$.
Portanto, a assinatura é válida.

Assinatura Digital Schnorr

Assim como o esquema de assinatura digital Elgamal, o esquema de assinatura Schnorr é baseado em logaritmos discretos. O esquema Schnorr minimiza a quantidade de cálculos dependentes da mensagem exigidos para gerar uma assinatura. O trabalho principal para a geração de assinatura não depende da mensagem, e pode ser feito durante o tempo ocioso do processador. A parte da geração da assinatura dependente da mensagem exige multiplicar um inteiro de $2n$ bits por um inteiro de n bits.

Assinatura Digital Schnorr

O esquema é baseado no uso de um módulo primo p, com $p-1$ tendo um fator primo q do tamanho apropriado; ou seja,$p-1 \equiv (mod q)$. Normalmente, usamos $p \approx 2^{1024}$ e $q \approx 2^{160}$. Assim, p é um número de 1024 bits, e q é um número de 160 bits, que também é o tamanho do valor de hash do SHA-1.

Assinatura Digital Schnorr

1ª parte - geração de um par de chaves privada/pública:

1. Escolha primos p e q, de modo que q é um fator primo de $p-1$.
2. Escolha um inteiro a, tal que $\alpha^q = 1 mod p$. Os valores a, p e q compreendem uma chave pública global que pode ser comum a um grupo de utilizadores.
3. Escolha um inteiro aleatório s com $0 < s < q$. Esta é a chave privada do utilizador.
4. Calcule $v = a^{-s} mod p$. Esta é a chave pública do utilizador.

Assinatura Digital Schnorr

Um utilizador com chave privada s e chave pública v gera uma assinatura:

1. Escolha um inteiro aleatório r com $0 < r < q$ e calcule $x = a^r mod p$. Esse cálculo é um estágio de pré-processamento independente da mensagem M a ser assinada.
2. Concatene a mensagem com x e calcule o hash do resultado para obter o valor e: $e = H(M||x)$
3. Calcule $y = (r + se) mod q$. A assinatura consiste no par (e, y).

Assinatura Digital Schnorr

Qualquer outro utilizador pode verificar a assinatura:

1. Calcule $x' = a^y v^e mod p$.
2. Verifique se $e = H(M||x')$.

Assinatura Digital Schnorr

Para ver se a verificação funciona, observe que
$x' \equiv a^y v^e \equiv a^y a^{-se} \equiv a^{y-se} \equiv a^r \equiv x(modp)$.
Logo, $H(M||x') = H(M||x)$.

Algoritmo de Assinatura Digital do NIST

O National Institute of Standards and Technology (NIST) publicou o Federal Information Processing Standard FIPS 186, conhecido como algoritmo de assinatura digital (Digital Signature Algorithm - DSA). O DSA utiliza o Secure Hash Algorithm (SHA). O DSA foi proposto originalmente em 1991 e revisto em 1993 em resposta ao feedback público em relação à segurança do esquema. Houve outra revisão secundária em 1996. Em 2000, uma versão expandida do padrão foi emitida como FIPS 186-2.

A técnica do DSA

O DSA utiliza um algoritmo que é projetado para oferecer apenas a função de assinatura digital. Diferente do RSA, ele não pode ser usado para encriptação ou troca de chave. Apesar disso, essa é uma técnica de chave pública.
A técnica do DSA também usa uma função de hash. O código de hash é fornecido como entrada de uma função de assinatura, junto com um número aleatório k, gerado para essa assinatura em particular. A função de assinatura também depende da chave privada do emissor (PR_a) e um conjunto de parâmetros conhecidos de um grupo de membros em comunicação. Esse conjunto possui uma chave pública global (PU_G). O resultado é uma assinatura que consiste em dois componentes, rotulados com s e r.

A técnica do DSA

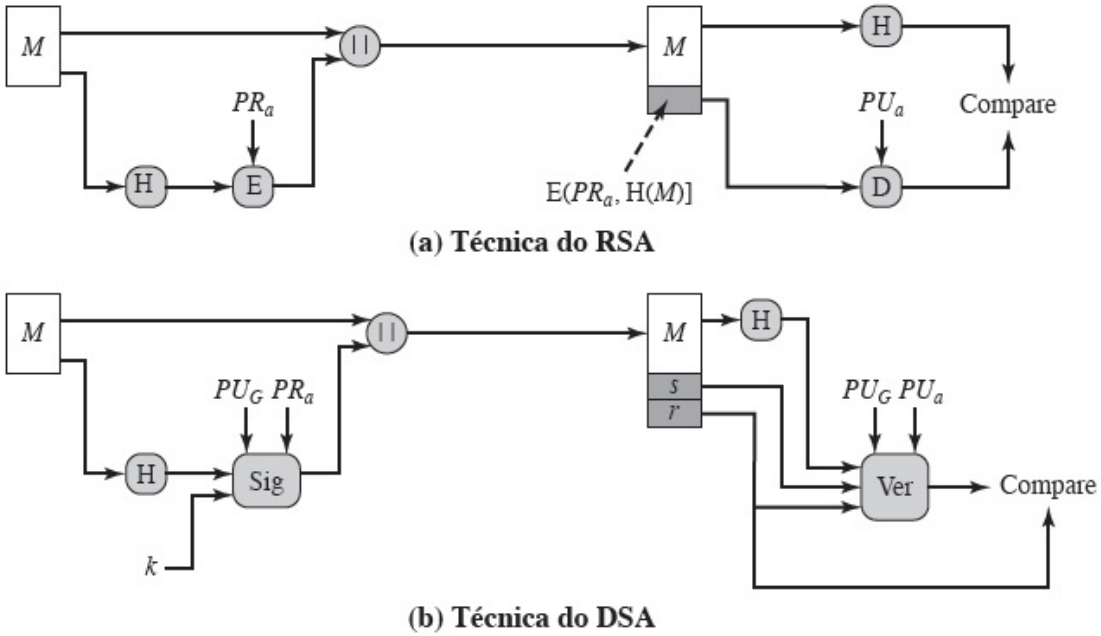


Figura: Duas técnicas para assinaturas digitais.

Digital Signature Algorithm

Componentes globais da chave pública

p número primo entre $2^{L-1} < p < 2^L$ para $512 \leq L \leq 1024$
e L um múltiplo de 64; ou seja, o tamanho entre 512 e 1024 bits em incrementos de 64 bits

q divisor primo de $(p-1)$, onde $2^{N-1} < q < 2^N$, ie, tamanho de N bits

$g = h(p-1)/qmodp$, onde h é qualquer inteiro em $1 < h < (p-1)$, tal que $h^{(p-1)/q}\ mod\ p > 1$

$M =$ mensagem a ser assinada
$H(M) =$ hash de M usando SHA-1
$M', r', s' =$versões recebidas de M, r, s

Digital Signature Algorithm

Chave privada do utilizador

x inteiro aleatório ou pseudoaleatório com $0 < x < q$

Chave pública do utilizador

$y = g^x \ mod \ p$

Número secreto por mensagem do utilizador

k inteiro aleatório ou pseudoaleatório com $0 < k < q$

$M =$ mensagem a ser assinada

$H(M) =$ hash de M usando SHA-1

$M', r', s' =$versões recebidas de M, r, s

Digital Signature Algorithm

Assinatura

$r = (g^k \ mod \ p) \ mod \ q \ s = [k^{-1}(H(M) + xr)] \ mod \ q$

Assinatura $= (r, s)$

Verificação

$w = (s^{'})^{-1} \ mod \ q$

$u_1 = [H(M^{'})w] \ mod \ q$

$u_2 = (r^{'})w \ mod \ q$

$v = [(g^{u1}y^{u2}) \ mod \ p] \ mod \ q$

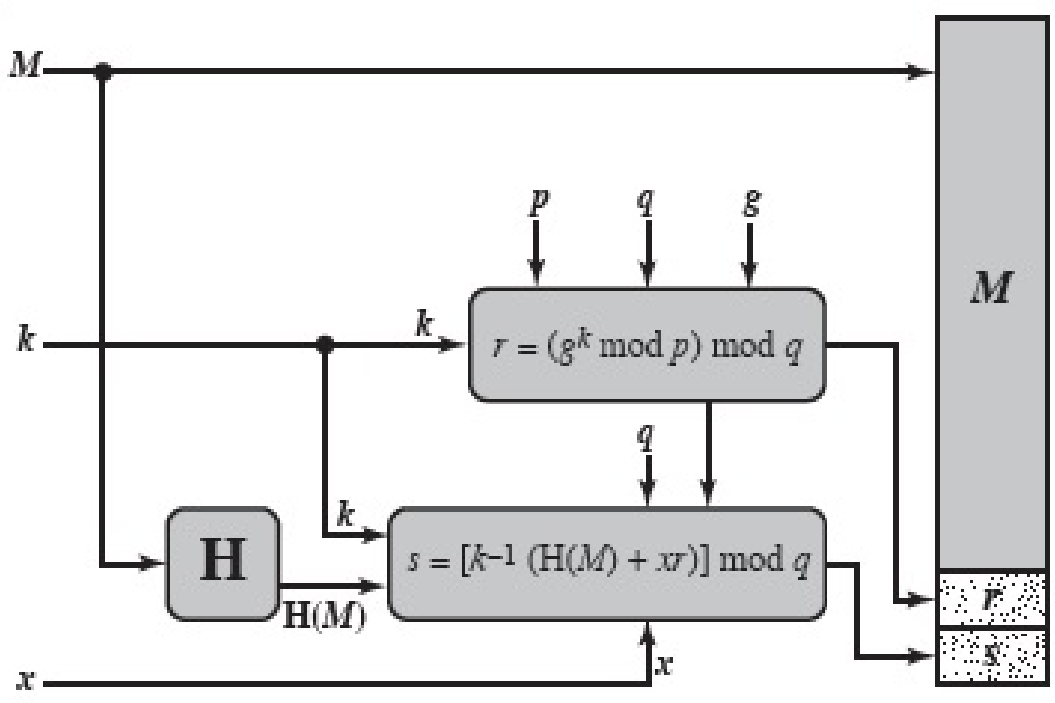
TESTE: $v = r^{'}$

$M =$ mensagem a ser assinada

$H(M) =$ hash de M usando SHA-1

$M^{'}, r^{'}, s^{'} =$ versões recebidas de M, r, s

Digital Signature Algorithm



(a) Assinatura

Algoritmo de Assinatura Digital de Curva Elíptica

A versão 2009 do FIPS 186 inclui uma nova técnica de assinatura digital baseada em criptografia de curva elíptica, conhecida como Elliptic Curve Digital Signature Algorithm (ECDSA).

Algoritmo de Assinatura Digital de Curva Elíptica

1. Todos aqueles que participam do esquema de assinatura digital usam os mesmos parâmetros de domínio global, que definem uma curva elíptica e um ponto de origem na curva.
2. Um assinante precisa primeiro gerar um par de chaves: pública e privada. Para a chave privada, o assinante seleciona um número aleatório ou pseudoaleatório. Usando esse número aleatório e o ponto de origem, o assinante calcula outro ponto na curva elíptica. Essa é a chave pública do assinante.

Algoritmo de Assinatura Digital de Curva Elíptica

1. Um valor de hash é gerado para a mensagem ser assinada. Usando a chave privada, os parâmetros de domínio e o valor de hash, a assinatura é gerada. A assinatura consiste em dois inteiros, r e s.
2. Para verificar a assinatura, o verificador usa como entrada a chave pública do assinante, os parâmetros do domínio e o inteiro s. A saída é um valor v que é comparado com r. A assinatura é válida se $v = r$.

Parâmetros de domínio global

q	um número primo
a, b	inteiros que especificam a equação de curva elíptica definida sobre Z_q com a equação $y^2 = x^3 + ax + b$
G	um ponto de base representado por $G = (x_g, y_g)$ sobre a equação da curva elíptica
n	ordem do ponto G; ou seja, n é o menor inteiro positivo tal que $nG = O$. Este também é o número de pontos na curva.

Geração de chave

1. Selecione um inteiro aleatório d, $d \in [1, n-1]$
2. Calcule $Q = dG$. Este é um ponto em $Eq(a, b)$.
3. A chave pública de Bob é Q e a chave privada é d.

Geração e autenticação de assinatura digital

Bob gera uma assinatura digital de 320 bytes para a mensagem m usando as seguintes etapas:

1. Selecione um inteiro aleatório ou pseudoaleatório k, $k \in [1, n-1]$
2. Calcule o ponto $P = (x, y) = kG$ e $r = x \ mod \ n$. Se $r = 0$, então vá para a etapa 1
3. Calcule $t = k^{-1} \ mod \ n$

Geração e autenticação de assinatura digital

Bob gera uma assinatura digital de 320 bytes para a mensagem m usando as seguintes etapas:

1. Calcule $e = H(m)$, onde H é a função de hash SHA-1, que produz um valor de hash de 160 bits
2. Calcule $s = k^{-1}(e + dr)\ mod\ n$. Se $s = O$, então vá para a etapa 1
3. A assinatura da mensagem m é o par (r, s).

Geração e autenticação de assinatura digital

Alice conhece os parâmetros de domínio público e a chave pública de Bob. Alice recebe a mensagem e a assinatura digital de Bob e a verifica usando as seguintes etapas:

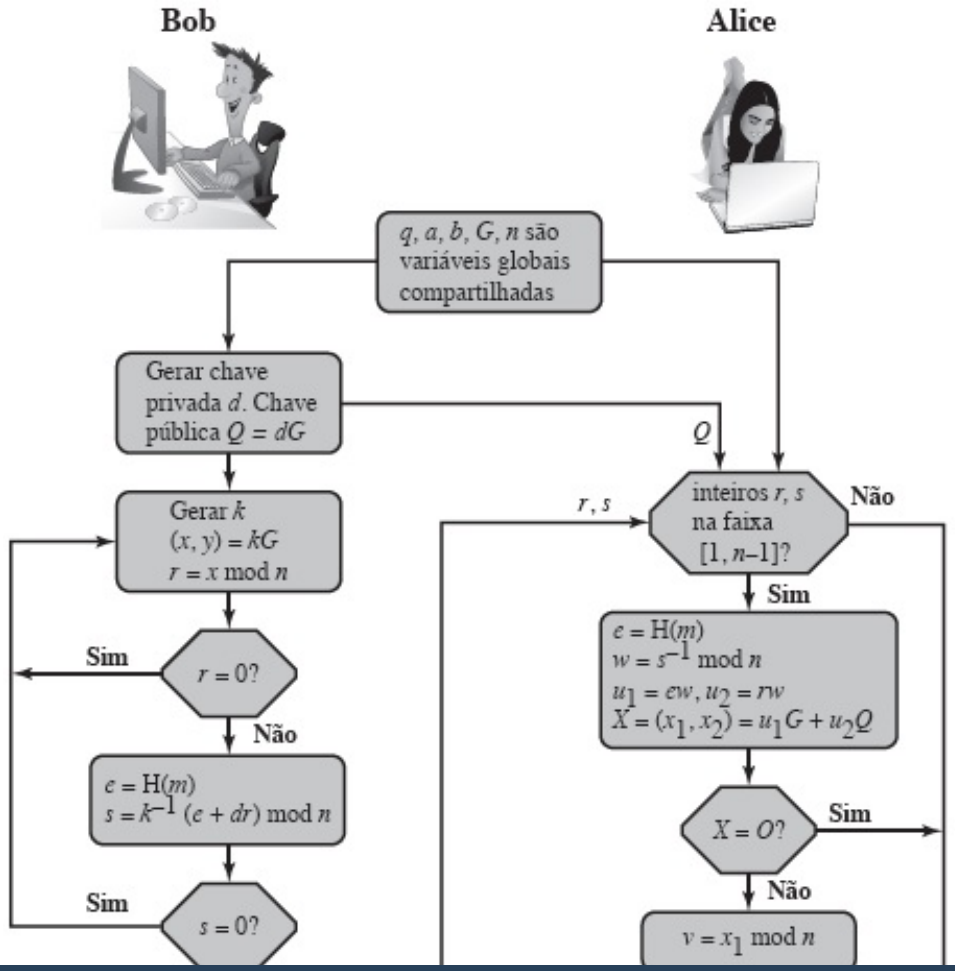
1. Verifique se r e s são inteiros no intervalo de 1 a $n - 1$
2. Usando SHA-1, calcule o valor de hash de 160 bits $e = H(m)$
3. Calcule $w = s^{-1} \ mod \ n$

Geração e autenticação de assinatura digital

Alice conhece os parâmetros de domínio público e a chave pública de Bob. Alice recebe a mensagem e a assinatura digital de Bob e a verifica usando as seguintes etapas:

1. Calcule $u_1 = ew$ e $u_2 = rw$
2. Calcule o ponto $X = (x_1, y_1) = u_1 G + u_2 Q$
3. Se $X = O$, rejeite a assinatura; se não, calcule $v = x_1 \ mod \ n$
4. Aceite a assinatura de Bob se e somente se $v = r$

Geração e autenticação de assinatura digital



Algoritmo de Assinatura Digital RSA-PSS

Além do DSA do NIST e do ECDSA, a versão de 2009 do FIPS 186 também inclui várias técnicas baseadas no RSA, todas desenvolvidas pela RSA Laboratories.
Os três principais esquemas de assinatura RSA diferem principalmente no formato de preenchimento que a operação de geração de assinatura emprega para embutir o valor de hash num representante da mensagem, e em como a operação de verificação de assinatura determina que o valor de hash e o representante da mensagem são consistentes.
Para todos os esquemas desenvolvidos antes do PSS, não foi possível desenvolver uma prova matemática de que o esquema de assinatura é tão preciso quanto a primitiva básica do RSA para encriptação/decriptação.

A Operação de Assinatura

1. Forme o bloco de dados $DB =$preenchimento$_2||sal$
2. Calcule o valor de MGF de H:
   $dbMask =$MGF$(H, emLen - hLen - 1)$
3. Calcule $maskedDB = DB \oplus dbMsk$
4. Defina os 8 bits $emLen - emBits$ do octeto mais à esquerda em $maskedDB$ como 0
5. $EM = maskedDB||H||bc$

A Operação de Assinatura

Formando a assinatura

Agora, mostramos como a assinatura é formada por um assinante com chave privada $\{d, n\}$ e chave pública $\{e, n\}$.
Trate a sequência de octetos EM como um inteiro binário não sinalizado, não negativo, m.
A assinatura s é formada encriptando m da seguinte forma:

$$s = m^d \ mod \ n$$

A Operação de Assinatura

Seja k o tamanho em octetos do RSA módulo n. Por exemplo, se o tamanho da chave para RSA for 2048 bits, então $k = 2048/8 = 256$. Depois converta o valor da assinatura s numa sequência de octetos S com tamanho de k octetos.

Verificação de Assinatura

Decriptação

Para a verificação da assinatura, trate a assinatura S como um inteiro binário s sem sinal, não negativo. O resumo de mensagem m é recuperado decriptando s da seguinte forma:

$$m = s^e \ mod \ n$$

Verificação de Assinatura

Opções	Hash	função de hash com saída de *hLen* octetos. A alternativa preferida atual é SHA-1, que produz um valor de hash de 20 octetos.
	MGF	Mask Generation Function. A especificação atual requer MGF1.
	sLen	tamanho do sal em octetos. Normalmente, $sLen = hLen$, que para a versão atual é 20 octetos.
Entrada	*M*	mensagem a ser codificada para assinatura.
	emBits	Esse valor é um a menos que o tamanho em bits do RSA módulo *n*.
Saída	*EM*	Encoded Message. Este é o resumo de mensagem que será encriptado para formar a assinatura digital.
Parâmetros	*emLen*	tamanho de *EM* em octetos = [*emBits*/8].
	preenchimento_1	string hexadecimal 00 00 00 00 00 00 00 00; ou seja, uma sequência de 64 bits zero.
	preenchimento_2	sequência hexadecimal de octetos 00 com tamanho ($emLen - sLen - hLen - 2$) octetos, seguida pelo octeto hexadecimal com valor 01.
	sal	um número pseudoaleatório.
	bc	o valor hexadecimal BC.

Verificação de Assinatura

1. Gere o valor de hash de M: $mHash = Hash(M)$
2. Se $emLen < hLen + sLen + 2$, informe "inconsistente"e termine
3. Se o octeto mais à direita de EM não tiver o valor hexadecimal BC, informe "inconsistente"e termine
4. Considere que $maskedDB$ seja os $emLen - hLen - 1$ octetos de EM, e seja H os próximos $hLen$ octetos
5. Se os 8 bits $emLen - emBits$ do octeto mais à esquerda em maskedDB não forem todos iguais a zero, informe "inconsistente"e termine

Verificação de Assinatura

1. Calcule $dbMask =$MGF$(H, emLen - hLen - 1)$
2. Calcule $DB = maskedDB \oplus dbMsk$
3. Defina os 8 bits $emLen - emBits$ do octeto mais à esquerda de DB como zero
4. Se os $(emLen - hLen - sLen - 1)$ octetos mais à esquerda de DB não forem iguais a preenchimento$_2$, informe "inconsistente"e termine

Verificação de Assinatura

1. Seja sal os últimos $sLen$ octetos de DB
2. Forme o bloco $M^{'}$ =preenchimento$_1||mHash||sal$
3. Gere o valor de hash de $M^{'}$: $H^{'} = Hash(M^{'})$
4. Se $H = H^{'}$, informe "consistente". Caso contrário, informe "inconsistente"

Verificação de Assinatura