

Criptografia assimétrica

SIO

deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

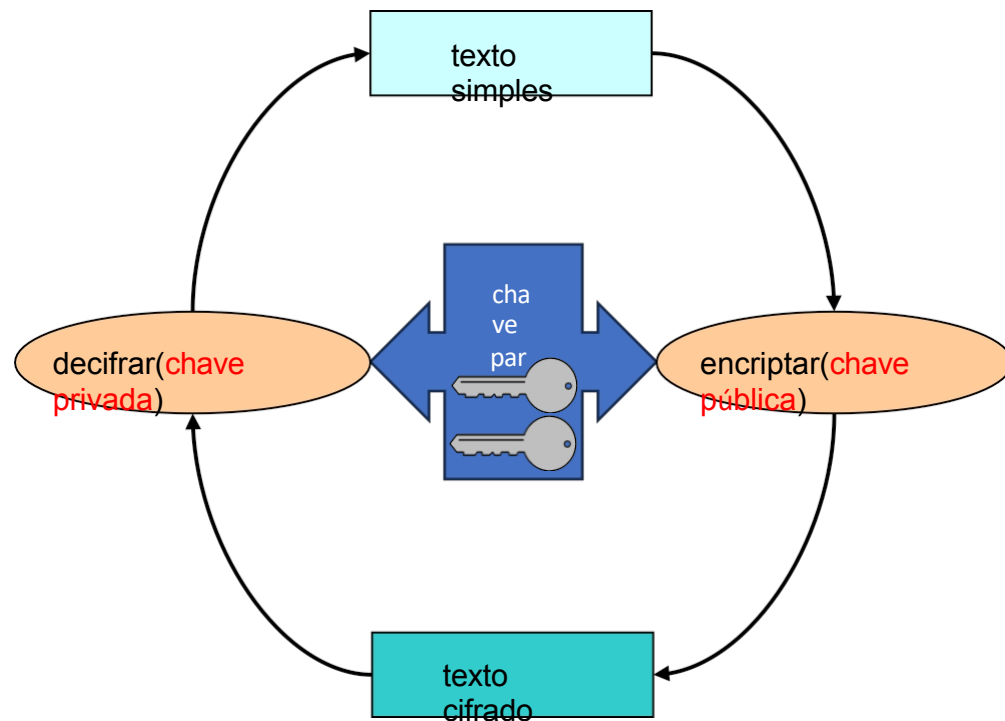
João Paulo Barraca

Cifras assimétricas (de bloco)

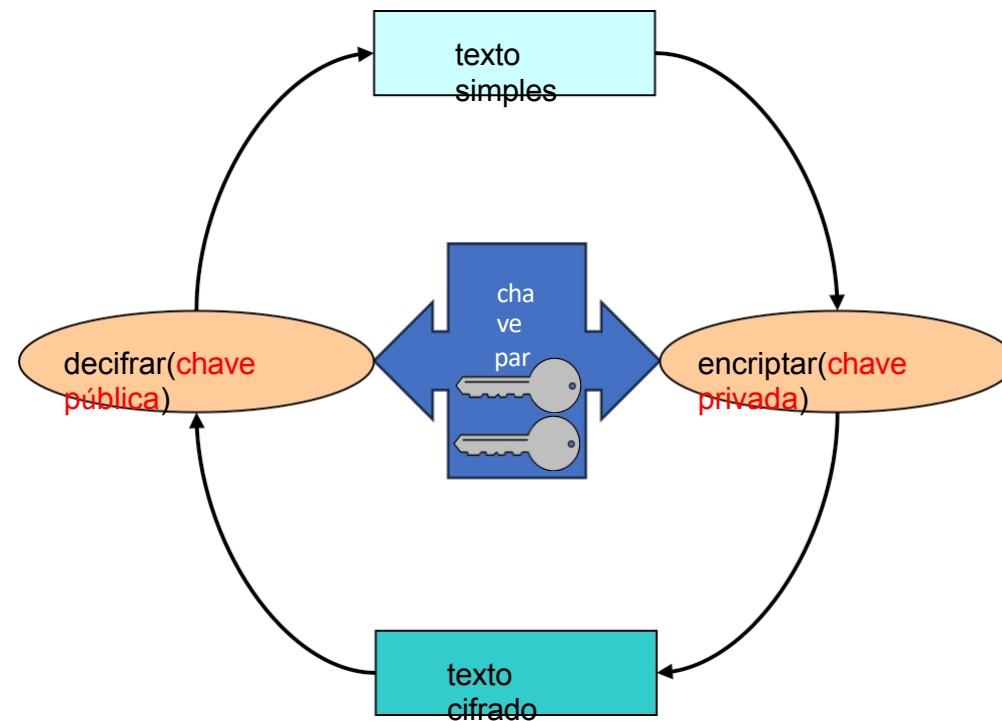
- Utilizar pares de chaves
 - **Uma chave privada:** pessoal, não transmissível
 - **Uma chave pública:** disponível para todos
- Permitir
 - Confidencialidade sem qualquer troca prévia de segredos
 - Autenticação
 - De conteúdo (integridade dos dados)
 - Da origem dos dados (autenticação da fonte ou assinatura digital)

Operações de uma cifra assimétrica

Confidencialidade

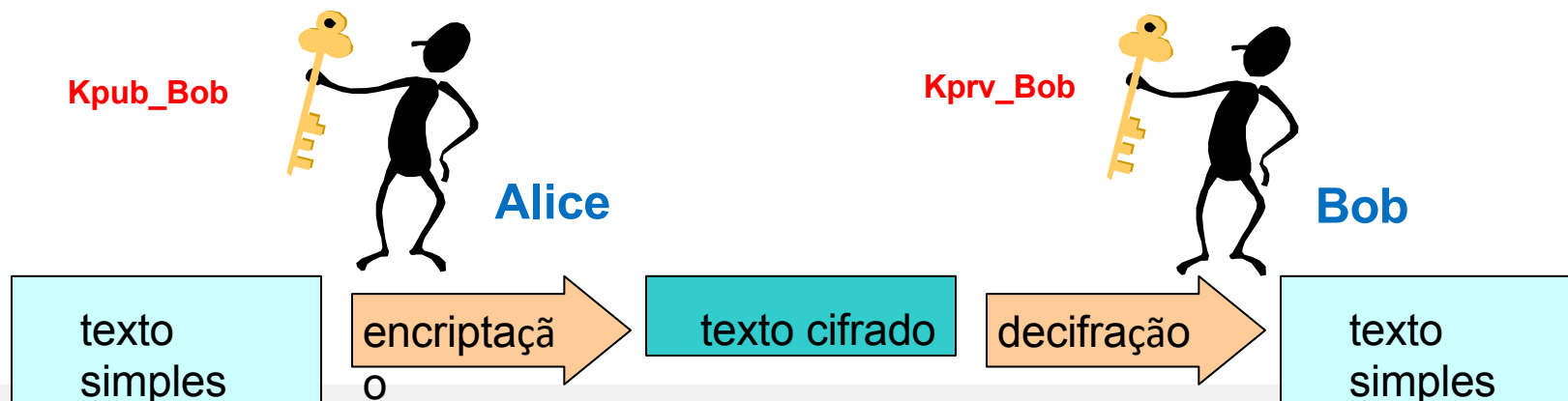


Autenticidade



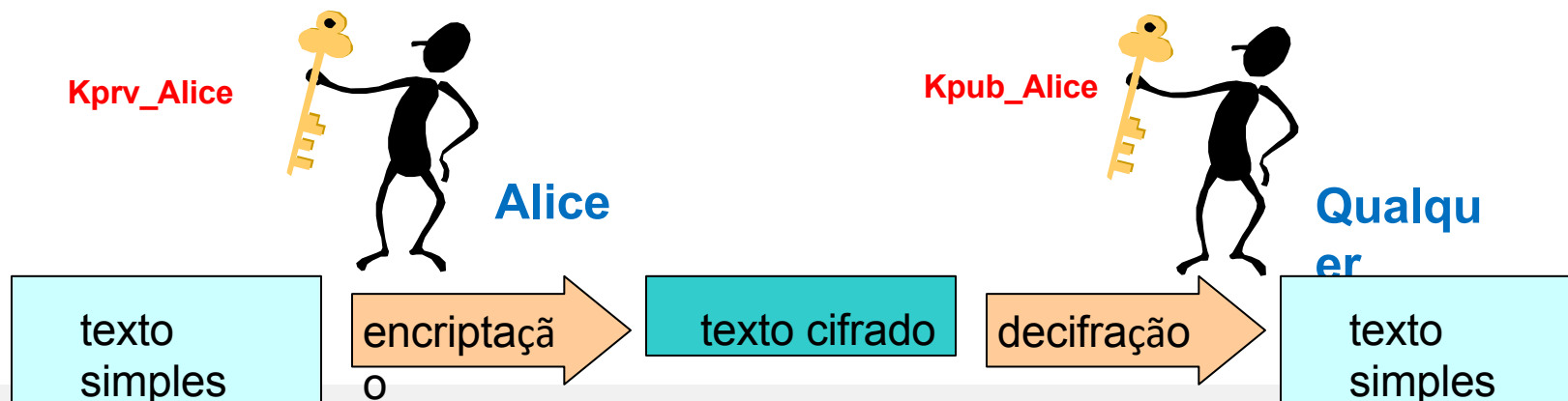
Casos de utilização: comunicação confidencial

- Comunicação segura com um alvo (Bob)
 - Alice cifra o texto simples **P** com a chave pública de Bob K_{pub_Bob}
Alice: $C = \{P\}_{K_{pub_Bob}}$
 - O Bob descripta o texto **cifrado C** com a sua chave privada K_{priv_Bob}
Bob: $P' = \{C\}_{K_{priv_Bob}}$
 - **P'** deve ser igual a **P** (requer verificação através do controlo da integridade)
 - K_{pub_Bob} precisa de ser conhecido por Alice



Casos de utilização: comunicação autenticada

- Autenticar a comunicação de Alice
 - Alice cifra o texto simples **P** com a sua chave privada $K_{\text{priv_Alice}}$
Alice: $C = \{P\}_{K_{\text{priv_Alice}}}$
 - Qualquer pessoa pode decifrar o texto **cifrado C** com a chave pública de Alices $K_{\text{pub_Alice}}$ **Qualquer pessoa:** $P' = \{C\}_{K_{\text{pub_Alice}}}$
 - Se $P' = P$, então **C** é a assinatura de Alice de **P**
 - $K_{\text{pub_Alice}}$ tem de ser conhecido pelos verificadores de mensagens



Cifras assimétricas

Questões

- Vantagens
 - Constituem um mecanismo de autenticação fundamental
 - Permitem explorar características que não são possíveis com cifras assimétricas
- Desvantagens
 - Desempenho: 2 ou 3 ordens de grandeza em relação ao AES
 - Muito ineficaz e consome muita memória: Chaves grandes
- Problemas
 - Distribuição fiável de chaves públicas: como saber se a chave pública é a correta?
 - Tempo de vida dos pares de chaves: Como garantir que podemos lidar com chaves perdidas/depreciadas/apagadas?

Cifras assimétricas

Visão geral

- Abordagens: problemas matemáticos complexos
 - **Logaritmos discretos** de números grandes
 - **Factorização** de números **inteiros** grandes
- Algoritmos mais comuns
 - RSA
 - ElGamal
 - Curvas elípticas (ECC)
- Outras técnicas com pares de chaves assimétricas
 - Diffie-Hellman (acordo de chaves)

RSA

Rivest, Shamir, Adelman, 1978

- Chaves: Privado: (d, n) Público: (e, n)
- Encriptação de chave pública (confidencialidade) de **P**
 - $C = P^e \bmod n$
 - $P = C^d \bmod n$
- Encriptação de chave privada (autenticidade) de **P**
 - $C = P^d \bmod n$
 - $P = C^e \bmod n$

P, C são números!
A mensagem é convertida de/para
números

$$0 \leq P, C < n$$

RSA

Rivest, Shamir, Adelman, 1978

- Complexidade computacional: **Logaritmo discreto e factorização de números inteiros**
- Seleção de chaves
 - Grande **n** (centenas ou milhares de bits)
 - **$n = p \times q$** , sendo **p** e **q** números primos grandes (secretos)
 - Selecionar um **e** co-primo com **$(p-1) \times (q-1)$**
 - Calcular **d** tal que **$e \times d \equiv 1 \pmod{(p-1) \times (q-1)}$**
 - Rejeitar **p** e **q**
 - O valor de **d** não pode ser calculado a partir de e e **n**
 - Apenas de **p** e **q**

coprimo $\rightarrow \gcd(a, b) = 1$

$\times \rightarrow$ multiplicação

mod \rightarrow operação de módulo

$\equiv \rightarrow$ congruência modular

$a \equiv b \pmod n$ iff $\text{rem}(a,n) = \text{rem}(b,n)$

Brincar com o RSA

- $p = 5$ $q = 11$ (números primos)
 - $n = p \times q = 55$
 - $(p-1) \times (q-1) = 40$
- $e = 3$ (chave pública = e, n)
 - Coprimo de 40
- $d = 27$ (chave privada = d, n)
 - $e \times d \equiv 1 \pmod{40}$ \rightarrow $d \times e \pmod{40} = 1$ \rightarrow $(27 \times 3) \pmod{40} = 1$
- Para uma mensagem a cifrar, $P = 26$ (note que $P, C \in [0, n-1]$)
 - $C = P^e \pmod{n}$ $= 26^3 \pmod{55} = 31$
 - $P = C^d \pmod{n}$ $= 31^{27} \pmod{55} = 26$

Encriptação híbrida

- Combina criptografia simétrica com criptografia assimétrica
 - Utilizar o melhor dos dois mundos, evitando problemas
 - Cifra assimétrica: Utiliza chaves públicas (mas é lenta)
 - Cifra simétrica: Rápida (mas com métodos de troca de chaves fracos)
- Método:
 - Obter K_{pub} do recetor
 - Gerar um aleatório K_{sym}
 - Calcular $C1 = E_{sym} (K_{sym} , P)$
 - Calcular $C2 = E_{asym} (K_{pub} , K_{sym})$
 - Enviar **C1 + C2**
 - C1 = Texto cifrado com chave simétrica
 - C2 = Chave simétrica cifrada com a chave pública do recetor
 - Pode também conter o IV

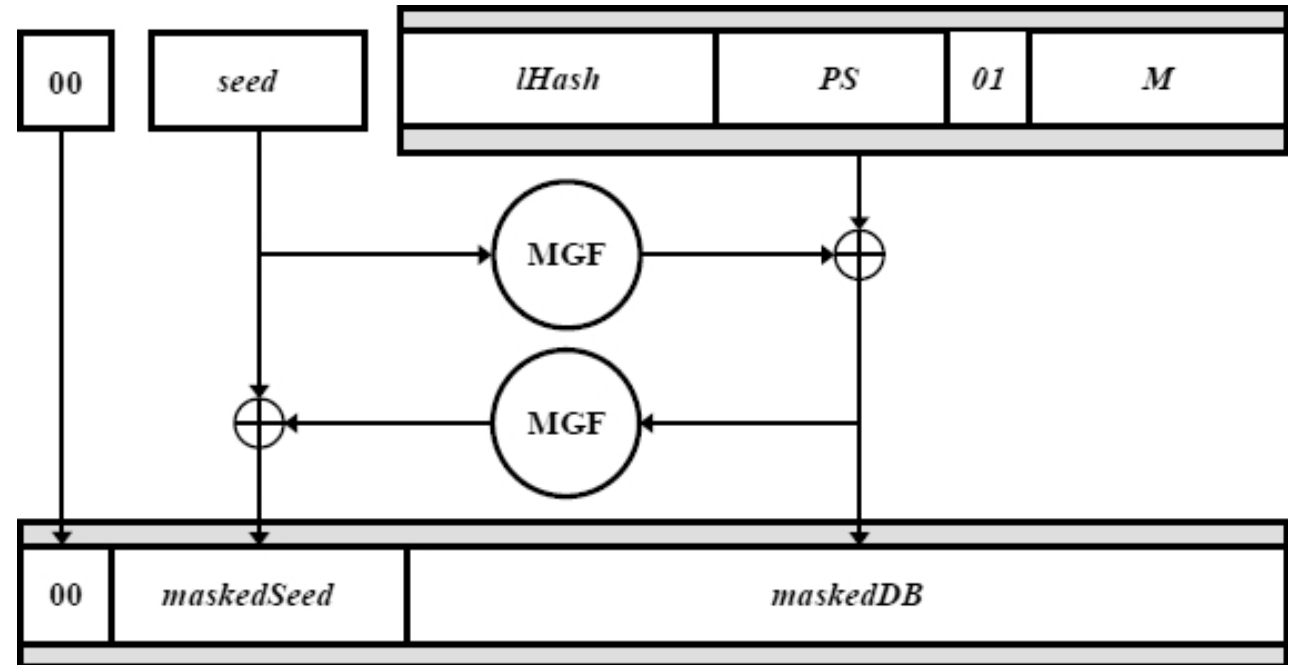
Randomização de encriptações assimétricas

- O RSA é um algoritmo determinístico: mensagens iguais resultam em resultados iguais
- O que precisamos: Resultado não determinístico de encriptações assimétricas
 - **N** encriptações do mesmo valor, com a mesma chave, devem produzir N resultados diferentes
 - **Objetivo: evitar a descoberta de valores encriptados por tentativa e erro**
- Abordagens
 - Concatenação do valor a encriptar com dois valores
 - Um fixo (para controlo da integridade)
 - Um aleatório (para aleatorização)

Randomização de encriptações assimétricas

OAEP (Optimal Asymmetric Encryption Padding)

- iHash: digest over Label
- semente: valor aleatório
- PS: zeros
- M: texto simples
- MGF: Função de geração de máscaras
 - Semelhante ao Hash, mas com tamanho variável



Acordo de chaves Diffie-Hellman (1976)



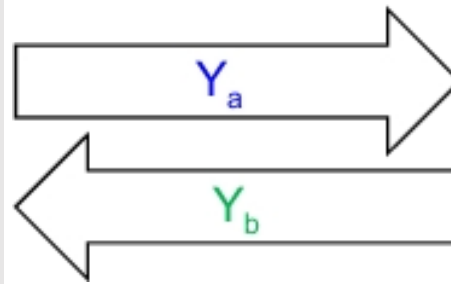
q (primo grande)
 α (raiz primitiva mod q)



a = aleatório

$$Y_a = \alpha^a \text{ mod } q$$

$$K_{ab} = Y_b^a \text{ mod } q$$



$$K_{ab} = K_{ba}$$

b = aleatório

$$Y_b = \alpha^b \text{ mod } q$$

$$K_{ba} = Y_a^b \text{ mod } q$$

Acordo de chaves Diffie-Hellman (1976)



a = aleatório

$$Y_a = \alpha^a \bmod q$$

$$K_{ac} = Y_c^a \bmod q$$



c = aleatório

$$Y_c = \alpha^c \bmod q$$

$$K_{ca} = Y_a^c \bmod q$$

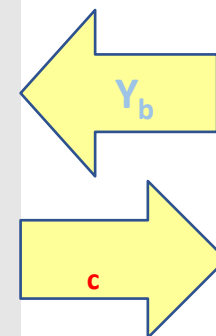
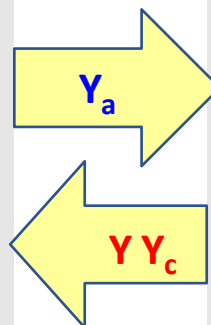
$$K_{cb} = Y_b^c \bmod q$$



b = aleatório

$$Y_b = \alpha^b \bmod q$$

$$K_{bc} = Y_c^b \bmod q$$



Criptografia de curva elíptica (ECC)

- As curvas elípticas são funções específicas
 - Têm um gerador (G)
 - Uma chave privada K_{prv} é um número inteiro com um máximo de bits permitido pela curva
 - Uma chave pública K_{pub} é um ponto $(x,y) = K_{\text{prv}} \times G$
 - Dado K_{pub} , deve ser difícil adivinhar K_{prv}
- Curvas
 - Curvas NIST (15)
 - P-192, P-224, P-256, P-384, P-521
 - B-163, B-233, B-283, B-409, B-571
 - K-163, K-233, K-283, K-409, K-571

Outras curvas

- Curva25519 (256 bits)
- Curva448 (448 bits)

ECDH: DH com ECC



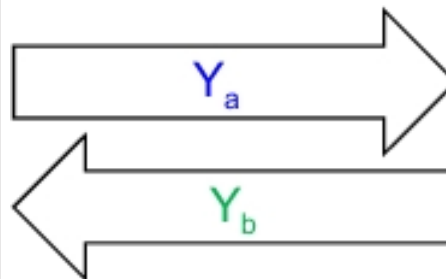
Curva ECC $\rightarrow G$



a = aleatório

$$Y_a = a G$$

$$K_{ab} = a Y_b$$



b = aleatório

$$Y_b = b G$$

$$K_{ba} = b Y_a$$

$$K_{ab} = K_{ba}$$

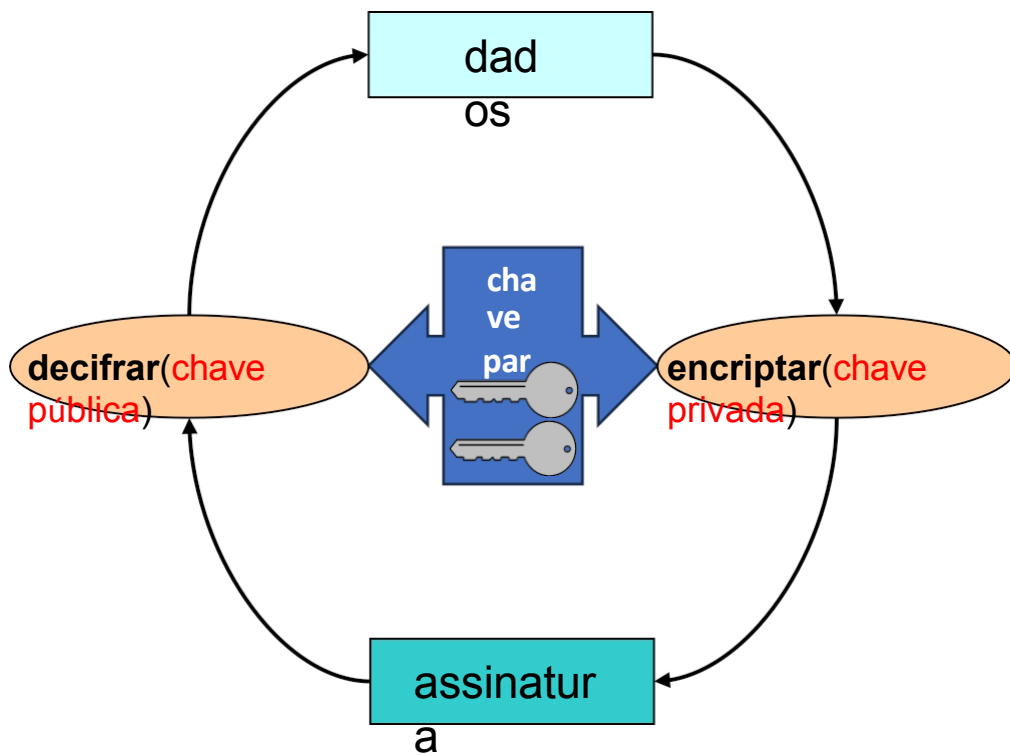
Encriptação de chave pública ECC

Combina a encriptação híbrida com a ECDH

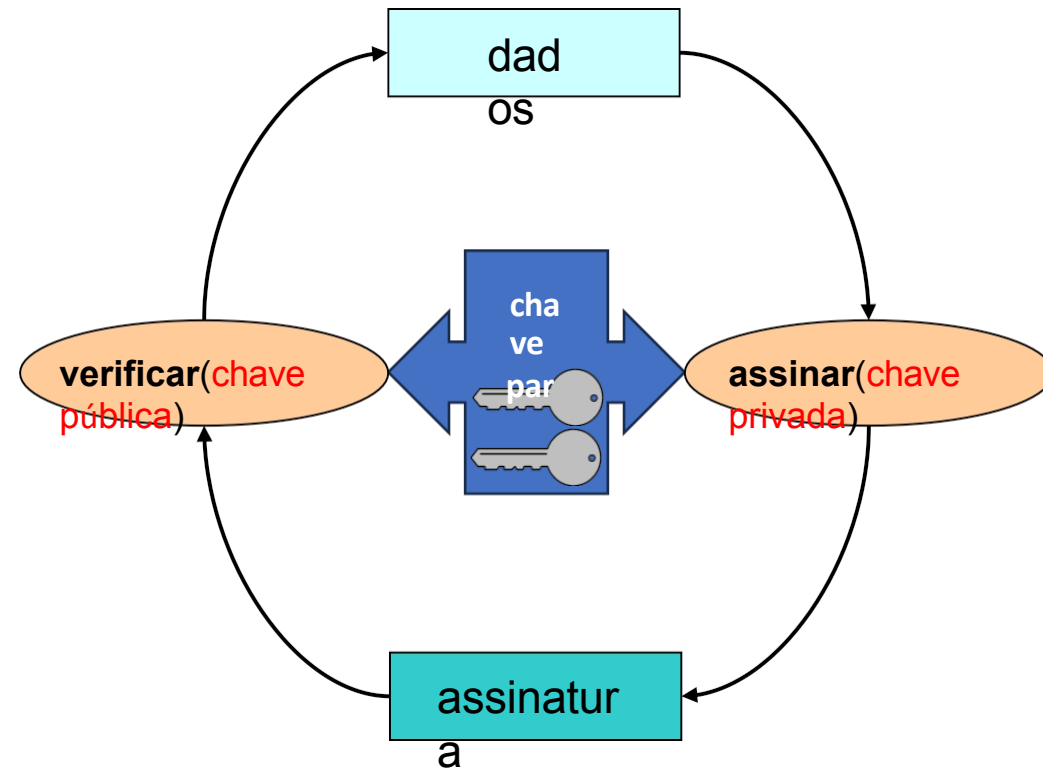
- Obter K_{pub_recv} do recetor
- Gerar um K_{priv_send} aleatório e o K_{pub_send} correspondente
- Calcular $K_{sym} = K_{priv_send} K_{pub_recv}$
- $C = E(P, K)_{sym}$
- Enviar $C + K_{pub_send}$
- O recetor calcula $K_{sym} = K_{pub_send} K_{priv_recv}$
- $P = D(C, K)_{sym}$

Assinaturas digitais

Encriptar/Desencriptar (RSA)



Assinar/Verificar (ElGamal, EC)



Operações com chaves privadas

- Autenticar o conteúdo de um documento
 - Garantir a sua integridade (não foi alterado)
- Autenticar o seu autor
 - Garantir a identidade do criador/originador
- Impedir o repúdio da carga encriptada
 - Não repúdio
 - Os autores genuínos não podem negar a sua autoria
 - Apenas o autor identificado poderia ter gerado uma determinada carga útil
 - Porque só o autor tem a chave privada

Assinaturas digitais

- Autenticar o conteúdo de um documento
 - Garantir a sua integridade (não foi alterado)
- Autenticar o seu autor
 - Garantir a identidade do criador/originador
- Impedir o repúdio de assinaturas
 - Propriedade de não repúdio
 - Os autores genuínos não podem negar a sua autoria
 - Apenas o autor identificado poderia ter gerado uma determinada assinatura

Considerações práticas

- A encriptação com chave privada é vital para a autenticação
 - Apenas o autor o pode fazer, todos o podem verificar
- Mas... o envio de textos autenticados seguros exigirá dois (lentos) encriptações
 - Lembre-se: As cifras assimétricas são lentas e ineficientes
- Abordagem preferida: **Encriptar Hash(T), criando assinaturas digitais**

Assinaturas digitais

- Abordagens

- Função de digestão do texto (apenas para o desempenho)
- Encriptação/desencriptação assimétrica ou assinatura/verificação

Assinatura:

$$A_x(\text{doc}) = \text{infor} + E(K_x^{-1}, \text{digerir}(\text{doc} + \text{info}))$$

maçã

$$A_x(\text{doc}) = \text{infor} + S(K_x^{-1}, \text{digerir}(\text{doc} + \text{info}))$$

maçã

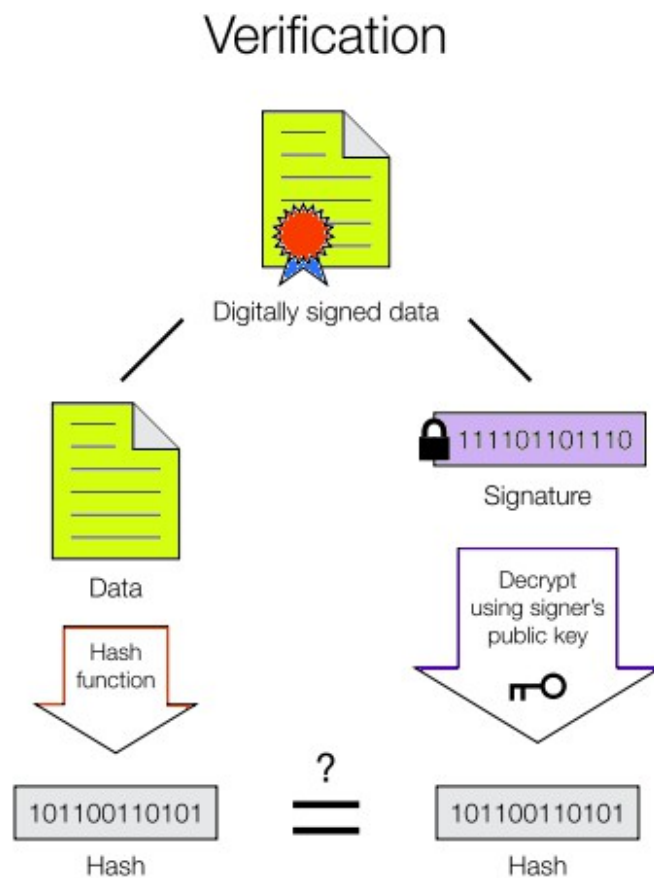
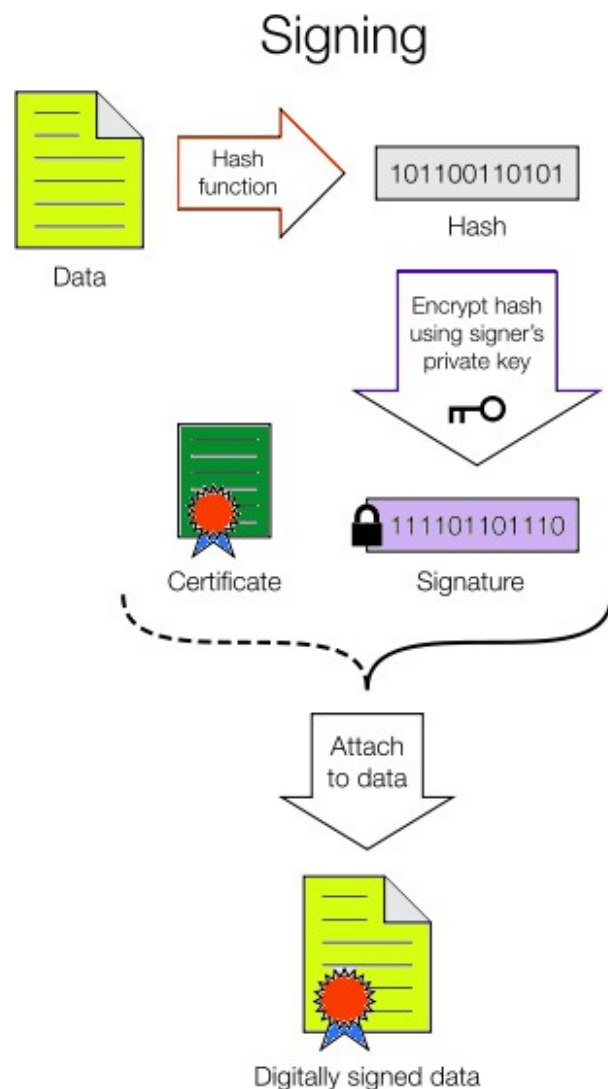
info = contexto de assinatura, identidade do signatário,
K_x

Verificação:

$$D(K_x, A_x(\text{doc})) \equiv \text{digest}(\text{doc} + \text{info})$$

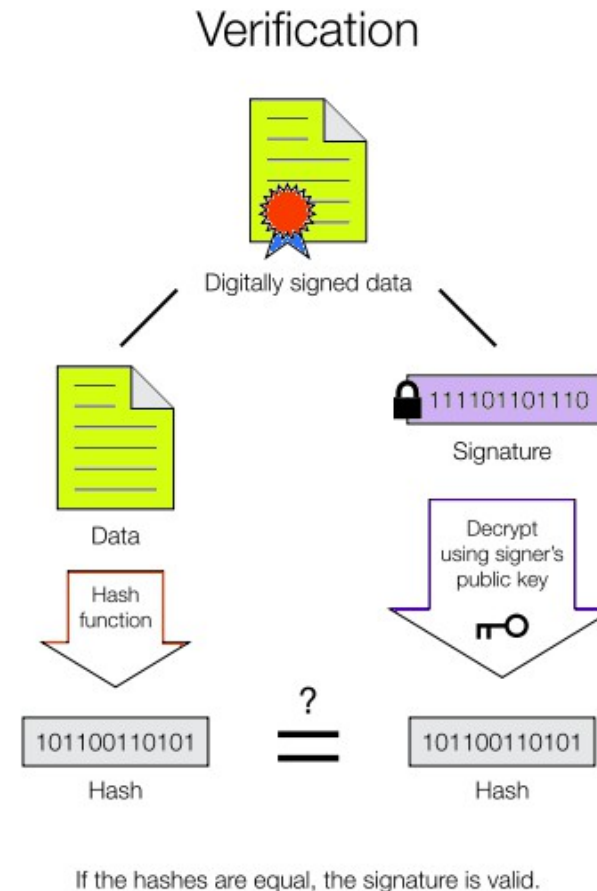
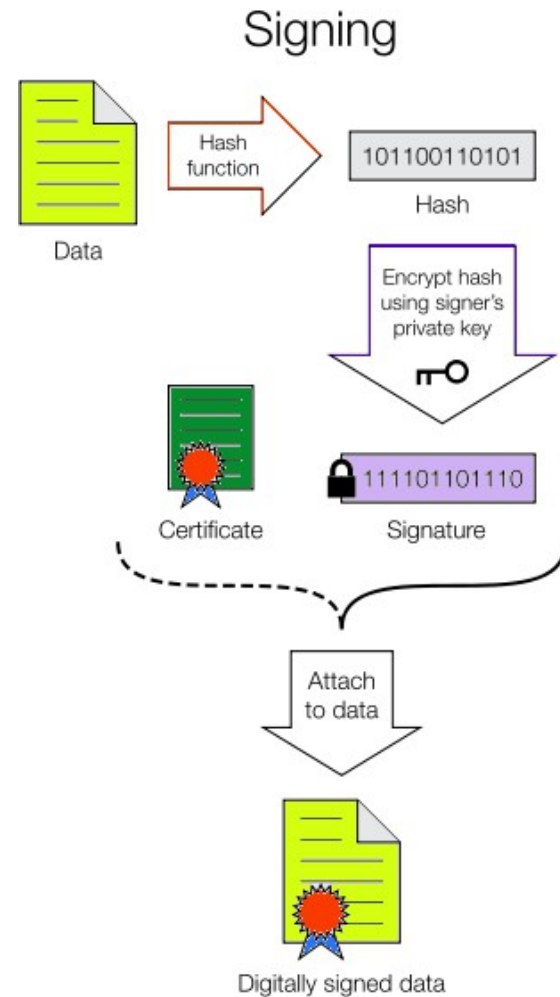
$$V(K_x, A_x(\text{doc}), \text{doc}, \text{info}) \rightarrow \text{Verdadeiro} / \text{Falso}$$

Assinaturas de encriptação / desencriptação



If the hashes are equal, the signature is valid.

Assinaturas de encriptação / desencriptação



Assinatura digital numa mensagem de correio eletrónico

Conteúdo de várias partes, assinatura com certificado

De - Sex Oct 02 15:37:14 2009
[...]
Data: Fri, 02 Oct 2009 15:35:55 +0100
De: Utilizador A <usera@domain.com>
MIME-Version: 1.0
Para: Utilizador B
<userb@domain.com> Assunto:
Teste
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature"; micalg=sha1; boundary="-----ms050405070101010502050101"

Esta é uma mensagem assinada criptograficamente no formato MIME.

-----ms050405070101010502050101
Content-Type: multipart/mixed;
boundary="-----060802050708070409030504"

Esta é uma mensagem com várias partes em formato MIME.

-----060802050708070409030504
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable

Corpo do correio

-----060802050708070409030504--
-----ms050405070101010502050101
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"
Content-Description: Assinatura criptográfica S/MIME

MIAGCSqGSib3DQEHAqCAMIACAQExCzAJBgUrDgMCGGUAMIAAGCSqGSib3DQEHAQAoIamTCCBUkwggSyoAMCAQICBAcnlaEwDQYJKoZIhvcNAQEFBQAwTElMAkGA1UEBhMCVVMxGDAWBgNV
[...] KoZIhvcNAQEBBQAEgY Cofks852BV77NVuww53vSxO1XtI2JhC1CDlu+tcTPoMD1wq5dc5v40Tgsaw0N8dqgVLk8aC/CdGMbRBu+J1LKrcVZa+khnjtB66HhDRLrjmEGDNtrEjbbqvpd2QO2
vxB3iPTIU+vCGXo47e6GyRydqTpbq0r49Zqmx+IJ6Z7iigAAAAA==

-----ms050405070101010502050101--

Assinaturas digitais em kernel.org

Index of /pub/linux/kernel/v6.x		
mirrors.edge.kernel....		
patch-6.7.9.xz	06-Mar-2024 15:09	703K
patch-6.7.xz	08-Jan-2024 06:00	8M
patch-6.8.1.xz	15-Mar-2024 19:04	5992
patch-6.8.10.xz	17-May-2024 10:24	730K
patch-6.8.11.xz	25-May-2024 14:46	740K
patch-6.8.12.xz	30-May-2024 07:59	878K
patch-6.8.2.xz	27-Mar-2024 05:24	241K
patch-6.8.3.xz	03-Apr-2024 13:44	374K
patch-6.8.4.xz	04-Apr-2024 18:39	366K
patch-6.8.5.xz	10-Apr-2024 14:49	461K
patch-6.8.6.xz	13-Apr-2024 11:27	498K
patch-6.8.7.xz	17-Apr-2024 09:38	537K
patch-6.8.8.xz	27-Apr-2024 15:28	583K
patch-6.8.9.xz	02-May-2024 14:54	643K
patch-6.8.xz	10-Mar-2024 21:45	7M
patch-6.9.1.xz	17-May-2024 10:28	3336
patch-6.9.10.xz	18-Jul-2024 11:57	603K
patch-6.9.11.xz	25-Jul-2024 08:15	647K
patch-6.9.12.xz	27-Jul-2024 09:48	652K
patch-6.9.2.xz	25-May-2024 14:54	16K
patch-6.9.3.xz	30-May-2024 07:55	151K
patch-6.9.4.xz	12-Jun-2024 09:49	263K
patch-6.9.5.xz	16-Jun-2024 12:04	306K
patch-6.9.6.xz	21-Jun-2024 12:54	388K
patch-6.9.7.xz	27-Jun-2024 12:04	465K
patch-6.9.8.xz	05-Jul-2024 07:53	521K
patch-6.9.9.xz	11-Jul-2024 11:08	572K
patch-6.9.xz	13-May-2024 05:20	7M
sha256sums.asc	10-Oct-2024 11:05	102K

João Paulo Barraca, André Zúquete

```
mirrors.edge.kernel.org/pub/lin x +
mirrors.edge.kerne...
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

3f6690efe8dc49751e33fbcc45d35fa9048f75e03bfeaaa2a28e1037ca8d85cf  ChangeLog-6.0
7eb504c0d87687a37753fbfe13e54ea979648b987e6f3f49c7f8f947bee7df3e  ChangeLog-6.0.1
38a40e43b4daeb3de10ad21c414c5e969f600ed4105883cab885392ada0b24c2  ChangeLog-6.0.2
1252dbe12a2bfc4320bc8721ffa6ac9755d21b355456189453949c3e34b40a30  ChangeLog-6.0.3
ed8991c1d0c78cb907af07648eacd889a8d05ce2c752fefbac52faa7a5e76e3c  ChangeLog-6.0.4
a135968b2ba483877b1e0d6c29f022df2ea2202b83b2d7a6367b1d218c402822  ChangeLog-6.0.5
23982b4a283f50f9eff4cdffc5a92e3cf188373e928dfdb529a0355b2f03e591  ChangeLog-6.0.6
685098787f5099393813af01dbf42be53d4cf66439101819e9d6812f9ea18b0d  ChangeLog-6.0.7
75ab6be0d282b450c847e4fd8d16a900c55b02dd1c2d4d367a0d72d6fa3ea6c0  ChangeLog-6.0.8
40c049dfd11dea11d06d9fa38268e6a4f1c46168ff6afa374d8977db75e4bc15  ChangeLog-6.0.9
ec14449d5d5f11d0c80cf1c1c33f2628333e1c4cf00779cd1dec66fcb9a34626  ChangeLog-6.0.10
d1aec42501f371cb0d46e428c56fd1b9e785a3b7ad884f641505486a1721a517  ChangeLog-6.0.11
5a7cc6b10574bf4ee627977173f6de69c150ed3a7ff039b1cc2ab2e9aea3045f  ChangeLog-6.0.12
05a014d458f50fda29cb92bffb99f7ff13506d245d21124e4b836e28f0b8197a  ChangeLog-6.0.13
44728440fadf4711f85d4bbcb59cf43614f720c6d1aa7c9235c9497371ca55536  ChangeLog-6.0.14
05b2597d94fe9674d5c575bb008953d1b548939a55f12709fdc6b0ce69abc211  ChangeLog-6.0.15
2b269f51babfd89937206aa0fcac6f93c94cdf2f24d7e54ebc304a93cf9e4929  patch-6.11.2.xz
4c808f6dd8814ab55a343649a2e2b925895b7f97044d15fa3424e5cf69349c3e  patch-6.11.3.xz
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2.0.22 (GNU/Linux)

iQIcBAEBCAAGBQJnB7TpAAoJEGMtOgZYnaax9wUP/izFOfkROcdC0YdH5pmlNQmQ
cBhqieYbwVm8lIOGUndjvcRRe/k7JaZKA73w7yGr456QGSIBu2jOvgytmdBQ2QSG
i6u1LF6npRNo41qWb5cHd1L2UTEEf0qDgqtBEnvSAWHgozoopfj/2VhfdJ5H4n/n
tVjGHQeXm00EVWV0rOhLKfR31YvRMbQwNDcB79Hxazd7qpCL6/yT1K5S8wUQe6B6
Nt5m9dgjR3WN0x1u9Wrg5akC3sSE3881P+TR/g4KhvhUzWzWMAPDHyXdv50/QEE5
0rr5XowtKkHFc2DENGf/b9egx0ojdy637JtV4kK2FbquRTAwgDNVZv3p8M95KKG2
v+8XEkPpTEthuvDI259Rfk6Q0D+aI/4uMnVGK1RzpfBz0Q9qwJGrWUF9nj1mT6Ud
GV0DegxenKfwoe8dUxR61HNTFHL1T1oXiZ3YZk9XqS18N51+uUydkuGkAWlchQZo
6qIS2dvVMmHnqC99rSxaYr/Qnn5WooCTd+u4iflmcqT7ss5aBnI5hgWN30vSwkZk
mJuPK8gCHBAPrXfb9G0d6esDBFP+szNBbpUn4K1nHKmrMQT7prGfDTmMjlfckIax
xTuHfjGgFZeGeg6BIEGZ37Mh5k+GmnztOK/RxUS55iytkXZOs2rYHvuO69KIJCWb
uda1IuRrgOZ4hoWw1whd
=s9Ry
-----END PGP SIGNATURE-----
```

SIO