CiberSegurança

MEET, MEIC, MEIM

Assinaturas digitais

2021–2022

Integridade, autenticação e não repúdio

Recorde-se que os objetivos principais da criptografia atual são:

- a **confidencialidade** (*confidentiality, secrecy, privacy*)
- a **integridade da informação** (*data integrity*);
- a **autenticação** (*authentication*) ;
- o **não repúdio** (*non-repudiation*);

As primitivas apresentadas até agora estavam orientadas a garantir a confidencialidade (sistemas de cifra), a integridade dos dados (funções Hash, sem chave) e autenticar as entidades participantes (MAC, HMAC). Para assegurar também o não repúdio, são usadas as **assinaturas digitais**.

Assinaturas digitais

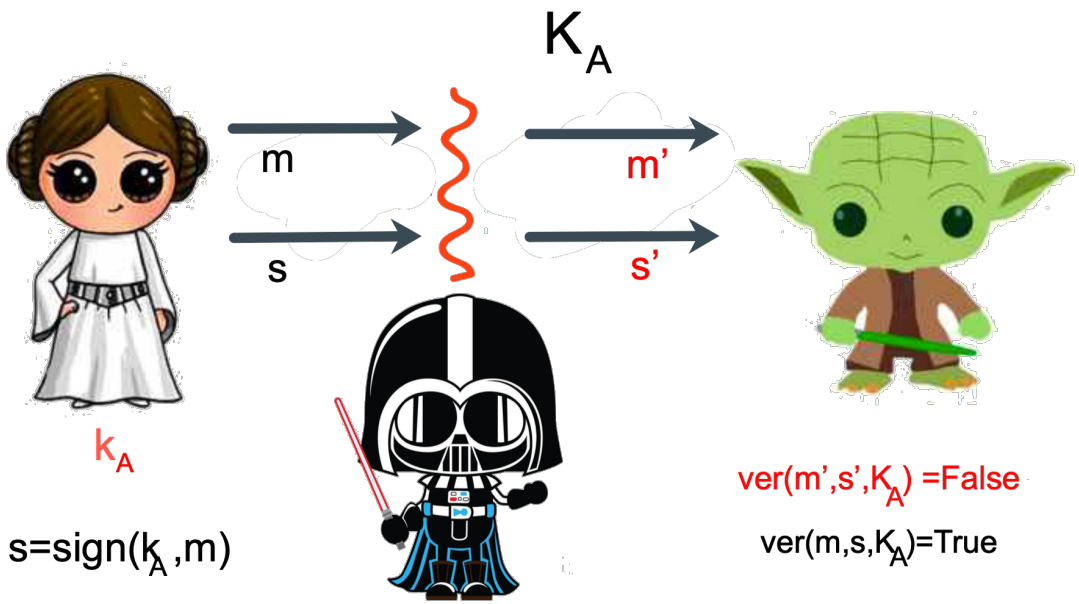
Um esquema de **assinatura digital** é um processo que produce valores fixos, chamados **assinaturas** de mensagens digitais usando uma chave privada.

A assinatura pode depois ser **verificada** usando a correspondente chave pública.

As assinaturas digitais diferem dos MAC no uso da chave privada: no caso da MAC, a chave secreta é partilhada pelo emissor e pelo destinatário, pelo que uma MAC não proporciona a propriedade de *não repúdio*, **qualquer usuário que possa verificar uma MAC é capaz também de gerar MACs**.

As assinaturas digitais são geradas usando a chave privada do **emissor/titular** e só ele pode as gerar: as assinaturas digitais oferecem **não repúdio.**

Esquema de uma assinatura digital



Uma assinatura digital consiste em duas funções, uma delas **sign** que calcula a assinatura a partir da informação privada k_A e da mensagem m e outra, **ver**, a função de verificação, com valores booleanos (True, False) que determina se a assinatura é válida, a partir da informação pública K_A, da assinatura s e da mensagem m recebida.

Segurança de uma assinatura digital

Para que a assinatura seja segura, é preciso que seja não invertível, mais precisamente:

1. Conhecidos K_A e m é computacionalmente intratável determinar k_A;
2. Conhecidos m e uma assinatura qualquer s de m, é computacionalmente intratável encontrar $m' \neq m$ tal que $\text{ver}(m') = True$.

$\rightsquigarrow$ **Por outras palavras, a assinatura será válida só se tiver sido calculada por alguém que conhece a chave privada da Alice.**

Assinaturas digitais

A combinação de uma cifra de chave pública e uma função *hash* criptográfica permitem construir assinaturas digitais. Por exemplo:

- Algoritmo para assinaturas digitais baseado no RSA;
- O *Digital Signature Algorithm* (DSA), que é um standard do FIPS (*Federal Information Processing Standard*) para assinaturas digitais, baseado no problema do logaritmo discreto.
- O *Elliptic Curve Digital Signature Algorithm* (ECDSA), um standard mais recente do FIPS (*Federal Information Processing Standard*) para assinaturas digitais, baseado no problema do logaritmo discreto generalizado em curvas elípticas.

$\rightsquigarrow$ **As assinaturas digitais de uma mensagem usam, normalmente, um hash da mensagem a assinar, não a totalidade da mensagem.**

I. Geração dos parâmetros do domínio

1. É escolhida uma função *hash* criptográfica H
2. São escolhidos p e q primos tais que $p-1$ é múltiplo de q;
3. É escolhido um inteiro h aleatoriamente, tal que $2 \leq h \leq p-2$;
4. É calculado $g \equiv h^{(p-1)/q} (\mathrm{mod}\, p)$. No caso improvável de $g = 1$, escolher outro h.

   Os parâmetros públicos do algoritmo são (p, q, g), que podem ser partilhados pelos diferentes usuários do sistema.

⇝ **Nas publicações do NIST encontram-se detalhadas recomendações atuais para o comprimento do valor hash e dos comprimentos em bits dos parâmetros escolhidos**.

II. Geração de chaves por sessão

A partir do conjunto de parâmetros (p, q, g), calcula-se um par (chave pública/chave privada) para cada uso:

1. É escolhido um inteiro aleatório x entre 1 e $q - 1$;
2. É calculado $y = g^x \pmod{p}$.

x é a chave privada do emissor da sessão, y é a chave pública da sessão que deve ser enviada ao destinatário através de um canal de confiança (não necessáriamente secreto).

III . Assinatura

A mensagem m é assinada do seguinte modo:

1. É escolhido aleatoriamente um inteiro k, $1 \leq k \leq q-1$;
2. Calcula-se $r \equiv (g^k \bmod p)(\bmod q)$. No caso improvável de $r = 0$, começar de novo com outro k aleatório.
3. Calcula-se $s = \left(k^{-1}(H(m) + xr)\right)(\bmod q)$. No caso improvável de $s = 0$, começar de novo com outro k aleatório.

A assinatura é o par (r, s).

IV. Verificação

O **destinatário/verificador** verifica que a assinatura (r, s) é válida para a mensagem m do seguinte modo:

1. Verifica que $0 < r < q$ e $0 < s < q$;
2. Calcula $w \equiv s^{-1} (\mathrm{mod}\, q)$;
3. Calcula $u_1 = H(m) \cdot w (\mathrm{mod}\, q)$
4. Calcula $u_2 = r \cdot w (\mathrm{mod}\, q)$
5. Calcula $v = (g^{u_1} y^{u_2} (\mathrm{mod}\, p)) (\mathrm{mod}\, q)$
6. A assinatura é válida se e só se $v = r$.

I. Geração dos parâmetros do domínio

1. Uma função *hash* criptográfica H;
2. Os parâmetros que definem a curva ellíptica que será usada: a, b, p;
3. um número primo n;
4. o ponto base G da curva elíptica de ordem n;
5. o inteiro h tal que $n \cdot h = N$ com N ordem da curva elíptica.

II. Assinatura

A mensagem m é assinada do seguinte modo:

1. é calculado o *hash* da mensagem m e truncado como um inteiro que denotamos z.
2. é escolhido um inteiro k em $\{1, 2, \ldots, n-1\}$;
3. é calculado o ponto $P = kG$;
4. é calculado $r = x_P$ (com x_P a coordenada x de P);
5. se $r = 0$, escolhe-se outro k e tenta-se de novo;
6. é calculado $s = k^{-1}(z + rd_A) \bmod n$; (com d_A a chave privada do emissor)
7. se $s = 0$, então tenta-se com outro k.

A assinatura é o par (r, s).

III. Verificação

Para verificar a assinatura é necessária a chave pública de Alice, o ponto H_A, o *hash* z e a assinatura (r, s). Realiza-se o seguinte procedimento:

1. Calcula-se $u_1 = s^{-1} z \bmod n$;
2. Calcula-se $u_2 = s^{-1} r \bmod n$;
3. Calcula-se o ponto $P = u_1 G + u_2 H_A$

A assinatura é válida se e só se $r = x_P \bmod n$.

A assinatura funciona ...

Efetivamente, a partir de $P = u_1 G + u_2 H_A$, por definição da chave pública $H_A = d_A G$, com d_A a chave privada de Alice

$$P = u_1 G + u_2 H_A = u_1 G + u_2 d_A G = (u_1 + u_2 d_A) G$$

$$= (s^{-1} z + s^{-1} r d_A) G = s^{-1} (z + r d_A) G$$

Recorde-se que

$$s = k^{-1} (z + r d_A)$$

donde $k = s^{-1}(z + r d_A)$ e sustituindo acima obtemos

$$P = kG$$

(Calcula o mesmo ponto que tínhamos na geração do algoritmo,)

Os Trusted Third-party (TTP)

A gestão das chaves costuma ser realizada através de um **Trusted Third-party (TTP)**.

- No caso de cifrados simétricos, se duas entidades A_i e A_j desejam comunicar entre sim, contactam o TTP que gera então uma chave de cifra simétrica para a comunicação entre A_i e A_j e envia às entidades;
- No caso de cifrados assimétricos os TTP guardam, para cada entidade A_j, a sua chave pública k_j.

$\rightsquigarrow$ **PROBLEMA: Como conseguir um TTP realmente de confiança???** ... é um problema sem solução ...