Computer security
Digital signatures

Olivier Markowitch

Digital signatures

Unlike handwritten signatures, the digital signatures:

- are linked to the content of the signed document
- is not compared with a witness signature, but is verified with an algorithm
- is universally verifiable

Non-repudiation

A digital signature ensures *non-repudiation of origin*

A signer cannot repudiate his signatures (he cannot convince anybody that he is not at the origin of his signatures)

A digital signature is generated on the basis of a private key and is verifiable, by anybody, thanks to the corresponding public key

Definitions

A **digital signature** is produced by a **digital signature generation algorithm** and is verified by a **digital signature verification algorithm**

A **digital signature scheme** is composed by a digital signature generation algorithm and the corresponding digital signature verification algorithm

Definitions

Two classes of digital signature schemes exist:

- with **appendix**: where the original message must be provided to the digital signature verification algorithm

- with **message recovery**: where the origin message can be recovered from the digital signature

Digital signature schemes with appendix

Each signer has a private key to sign and a corresponding public key to allow the verification of his digital signatures

Let M be a finite set of messages, S a finite set of digital signatures and K finite set of pair of keys (private and public)

For all pairs of private and public keys (k, k'), it exists a digital signature with appendix generation algorithm $\text{Sig}_{k'}$ and a corresponding digital signature verification algorithm Ver_k such that the digital signature of a message x is:

$$y = \text{Sig}_{k'}(x) \ : M \to S$$

and

$$\text{Ver}_k(x, y) \ : M \times S \to \{\text{true}, \text{false}\}$$

Digital signature schemes with appendix

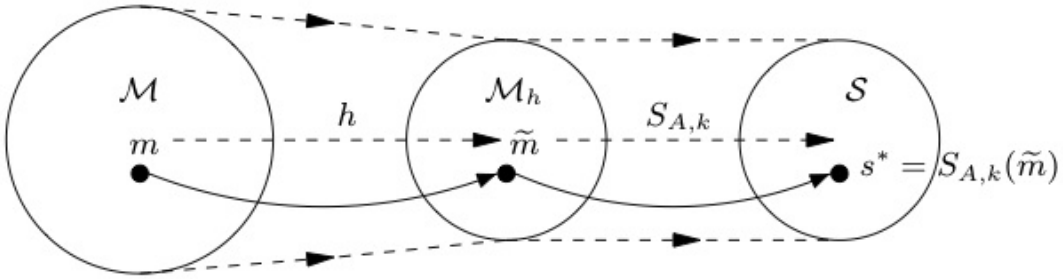
Cryptographic hash functions are usually used in digital signature schemes with appendix

The signer computes $m' = h(m)$ and $s = \text{Sig}_{k'}(m')$ where k' is the signer's private key

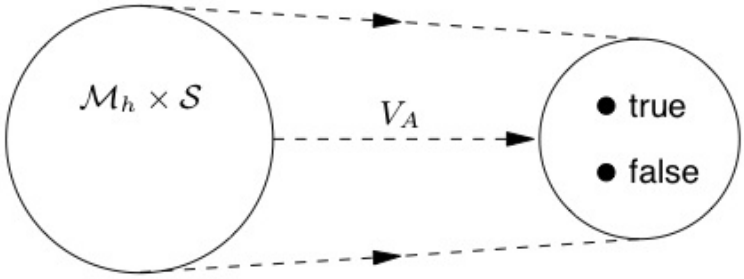
The verifier obtains m, s, the signer public key k, computes $m' = h(m)$ and accepts the signature if and only if $\text{Ver}_k(m', s) = true$

it should be computationally infeasible for an entity other than the signer to find a message M and the corresponding signature s such that $\text{Ver}_k(m', s) = true$ when k is the signer's public key and $m' = h(m)$

Digital signature schemes with appendix



(a) The signing process



(b) The verification process

Digital signature schemes with message recovery

Each signer has a private key to sign and a corresponding public key to allow the verification of his digital signatures

Let M be a finite set of messages, M_S a finite set of signable messages, S a finite set of digital signatures and K finite set of pair of keys (private and public)

Digital signature schemes with message recovery

For all pairs of private and public keys (k, k'), it exists a digital signature with message recovery algorithm $\text{Sig}_{k'}$ that applies $M_S \rightarrow S$, a redundancy function $R : M \rightarrow M_S$ and a corresponding digital signature verification algorithm Ver_k: $S \rightarrow M_S$ such that the digital signature of a message x is:

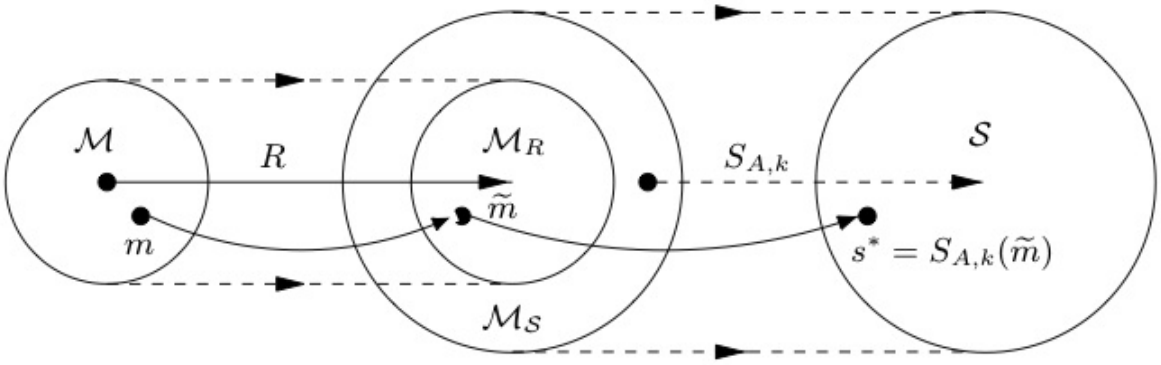
$$y = \text{Sig}_{k'}(R(x))$$

and

$$x' = \text{Ver}_k(y)$$

If $x' \notin M_S$ then the digital signature is rejected, otherwise the digital signature is accepted and the message $x = R^{-1}(x')$ is retrieved

Digital signature schemes with message recovery



Attacks

The aim of an opponent is to forge a digital signature that will be verified with the public key of another entity

If the opponent is either able to compute the private key of a genuine signer or is able to forge a digital signature for all the possible messages on the name of this genuine signer, the digital signature scheme is said to be *totally broken*

If the opponent is able to create a valid signature for a particular message or for a set of chosen messages chosen a priori, the digital signature scheme is said to allow *selective forgeries*

If the opponent is able to forge a signature for at least one message but having no control over the message whose signature is obtained, the digital signature scheme is said to allow *existential forgeries*

RSA

Key generation

- choose p and q two large primes approximatively of the same size
- let $n = pq$
- choose $e \in]1, \phi(n)[$ such that $gcd(e, \phi(n)) = 1$
- compute d such that $e \cdot d \equiv 1 \pmod{\phi(n)}$

The digital signature generation private key is d, the public verification key is (n, e)

RSA with message recovery

Digital signature generation

Let m the message to be signed:

- $\tilde{m} = R(m)$ where R is the redundancy function
- $s = \tilde{m}^d \mod n$ is the digital signature of m

Digital signature verification

Only s is needed to verify the digital signature:

- $\tilde{m} = s^e \mod n$
- if $\tilde{m} \in M_s$ then $m = R^{-1}(\tilde{m})$, otherwise the digital signature is rejected

RSA with appendix

Digital signature generation

Let m the message to be signed:

- $\tilde{m} = h(m)$ where h is a MDC
- $s = \tilde{m}^d \mod n$ is the digital signature of m

Digital signature verification

s and m are needed to verify the digital signature:

- $\tilde{m} = s^e \mod n$
- if $h(m) = \tilde{m}$ then the digital signature is accepted

El Gamal

Key generation

- choose a large prime p
- choose α a generator of $\mathbb{Z}^*_p$
- choose $a \in [1, p-2]$
- compute $\beta = \alpha^a \mod p$

The digital signature generation private key is a, the public verification key is (p, α, β)

El Gamal

Digital signature generation

Let m the message to be signed:

- choose randomly $k \in [1, p-2]$ such that k is prime with $p-1$
- compute $\gamma = \alpha^k \mod p$
- compute $\delta = (h(m) - a \cdot \gamma) \cdot k^{-1} \mod p-1$

The digital signature of m is the pair (γ, δ)

El Gamal

Digital signature verification

s and m are needed to verify the digital signature:

If $\gamma \in [1, p-1]$ and if $\beta^{\gamma} \cdot \gamma^{\delta} \equiv \alpha^{h(m)} \mod p$ then the digital signature is accepted

Indeed: $\beta^{\gamma} \cdot \gamma^{\delta} \mod p$

$= \alpha^{a \cdot \gamma} \cdot \gamma^{(h(m) - a \cdot \gamma) \cdot k^{-1}} \mod p$

$= \alpha^{a \cdot \gamma} \cdot \alpha^{k \cdot (h(m) - a \cdot \gamma) \cdot k^{-1}} \mod p$

$= \alpha^{a \cdot \gamma} \cdot \alpha^{h(m) - a \cdot \gamma} \mod p$

$= \alpha^{a \cdot \gamma} \cdot \alpha^{h(m)} \cdot \alpha^{-a \cdot \gamma} \mod p$

$= \alpha^{h(m)} \mod p$

El Gamal

Precautions:

- the value of k cannot be disclosed by the signer
- the signer cannot use the same value k to sign two different messages

Digital Signature Algorithm (DSA)

Key generation

- choose a large prime $q \in \left]2^{159}, 2^{160}\right[$
- choose a prime p of $512+64 \cdot t$ bits with $t \in [0, 8]$ and such that q divides $p - 1$
- choose α a generator of the cyclic group of order q in ${\mathbb{Z}^*}_p$
- choose randomly $a \in [1, q - 1]$
- compute $\beta = \alpha^a \mod p$

The digital signature generation private key is a, the public verification key is (p, q, α, β)

DSA

Digital signature generation

Let m the message to be signed:

- choose randomly $k \in]0, q[$ such that k is prime with $p - 1$
- compute $\gamma = (\alpha^k \mod p) \mod q$
- compute $\delta = (h(m) + a \cdot \gamma) \cdot k^{-1} \mod q$

The digital signature of m is the pair (γ, δ)

DSA

Digital signature verification

s and m are needed to verify the digital signature:

If $\gamma \in]0, q[$ and $\delta \in]0, q[$:

- compute $e_1 = h(m) \cdot \delta^{-1} \mod q$
- compute $e_2 = \gamma \cdot \delta^{-1} \mod q$

If $(\alpha^{e_1} \cdot \beta^{e_2} \mod p) \mod q = \gamma$ then the digital signature is accepted

DSA

$$(\alpha^{e_1} \cdot \beta^{e_2} \mod p) \mod q$$

$$= (\alpha^{h(m)\cdot\delta^{-1}} \cdot \alpha^{a\cdot\gamma\cdot\delta^{-1}} \mod p) \mod q$$

$$= (\alpha^{(h(m)+a\cdot\gamma)\cdot\delta^{-1}} \mod p) \mod q$$

since $\delta = (h(m) + a \cdot \gamma) \cdot k^{-1} \mod q$

$$= (\alpha^{k\cdot\delta\cdot\delta^{-1}} \mod p) \mod q$$

$$= (\alpha^{k} \mod p) \mod q$$

$$= \gamma$$

Blind signatures

D. Chaum, *Blind signatures for untraceable payments*. Proceedings of Crypto'82, Plenum Press, 1983

D. Chaum, *Security without identification : transaction systems to make big brother obsolete*. Communication of the ACM, Vol 28, 1985

D. Chaum, *Blinding for unanticipated signatures*, Proceedings of Eurocrypt'87, Lecture Notes in Computer Science, Vol 304, 1988

J. Camenish, J-M Piveteau et M. Stadler,*Blind signatures based on the discrete logarithm problem*. Proceedings of Eurocrypt'94, Lecture Notes in Computer Science, Vol 950, 1995