8th ACM Conference on Computer and Communications Security ‘ 2001

5-8 November 2001

Philadelphia - Pennsylvania - USA

Twin Signatures: an Alternative to the Hash-and-Sign Paradigm

David Naccache (Gemplus, France)
David Pointcheval (ENS, France)
Jacques Stern (ENS, France)

Overview

- Introduction
- Security notions for signatures
- The twinning paradigm
- A DL-based example
- An RSA-based example
- Conclusion

Introduction

- Digital signature = electronic version of handwritten signatures

$\Rightarrow$ authenticates the sender of a message

- the receiver knows the identity of the sender
- the sender cannot deny later having sent the message (non-repudiation)

Digital signatures

Defined by two algorithms

- the signing algorithm **S**:
  private key + message *m*
  $\rightarrow$ signature σ
- the verification algorithm **V**:
  public key + message *m*
  + alleged signature σ
  $\rightarrow$ agrees or not

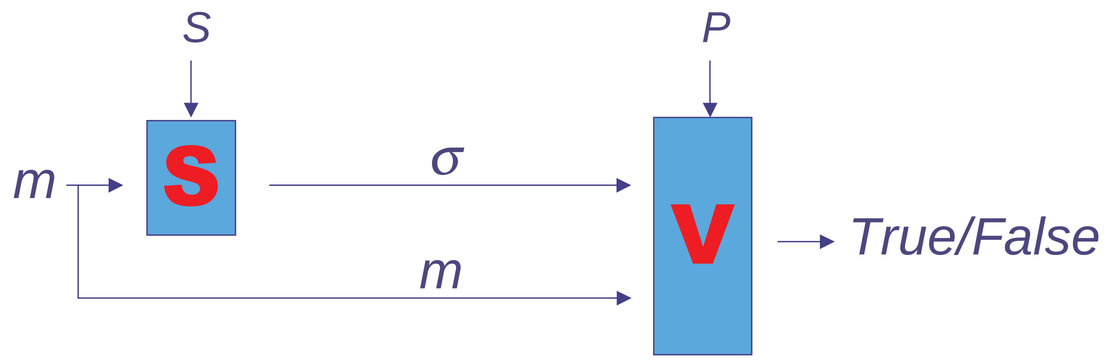
Digital signatures

Signing algorithm **S**

Verification algorithm **V**

Private key *S*

Public key *P*



Security: it is impossible to produce a new valid pair (m,σ)

Security notions

More precisely, one considers

- total break:

  the adversary recovers the private key
- universal forgery:

  the adversary can sign any message of her choice
- existential forgery:

  the adversary can produce accepted message/signature pairs

Adversaries

The information available to the adversary may be various, thus several attacks

- no-message attacks:

  the adversary just knows the verification algorithm (*i.e.* the public key)

- known-message attacks:

  she knows some message-signature pairs

- (adaptively) chosen-message attacks:

  she has access to a signing oracle

Secure signature schemes

For achieving non-repudiation, the scheme must prevent existential forgeries.

Furthermore, signatures are aimed to be published, thus known-message attacks should be withstood.

Secure signature scheme:
no existential forgery even against adaptively chosen-message attacks.

Example: RSA signature

$n = pq$ product of large primes
e : **public** exponent
$d = e^{-1} \bmod \varphi(n)$: **private** exponent

Signature of the message $m \in \mathbf{Z}_n$

$$\sigma = m^d \bmod n$$

Verification of (m,σ)

test whether $m = \sigma^e \bmod n$

RSA signature: problems

- Only small messages (in $\mathbf{Z}_n$) can be signed
- Existentially forgeable

$\Rightarrow$ in order to solve the former problem:
use of a collision-resistant hash function h

If h furthermore behaves like a truly random function $\{0,1\}^* \rightarrow \mathbf{Z}_n$: FDH in the ROM

FDH-RSA, provably secure [BR96, Co00]

$\Rightarrow$ hash-and-sign or hash-and-decrypt

An alternative: twinning

- Without the hash function, the RSA signature is insecure
  - even with it, the security proof only holds in the random oracle model

Insecure? Because from σ it is easy to compute m such that $m = \sigma^e \bmod n$

What about considering twin-signatures (σ, τ) such that $m = \sigma^e \bmod n$ and $m+1 = \tau^e \bmod n$?

Twin signatures

- Let **S** be a signature scheme (maybe weakly secure)
- We consider the signature scheme which consists in computing
  - $m_1 = f(m,r)$ and $m_2 = g(m,r)$ for some random r
  - $\sigma_1 = \mathbf{S}(m_1)$ and $\sigma = \mathbf{S}(m_2)$
- We thus sign two related messages

A DL-based example: DSA

$\mathbb{G} = \langle g \rangle$ of prime order q

x : **secret** key $y = g^x$: **public** key

- For signing $m \in \mathbb{Z}_q$, $\mathbf{S}_x(m) = (c,d)$, where

  $0 < u < q$ $c = (g^u) \bmod q$ $c \neq 0$

  and $d = (m + x\,c)/u \bmod q$ $d \neq 0$

- Verification, $\mathbf{V}_y(m,c,d)$:

  $h = 1/d \bmod q$, $h_1 = h\,m \bmod q$,

  $h_2 = h\,c \bmod q$, $c' = g^{h_1}\, y^{h_2}$

  check whether $0 < c, d < q$ and $c = c' \bmod q$

Twin-DSA

$\mathrm{DSA}_x(m) = \mathbf{S}_x(\mathrm{SHA}(m))$

- Unfortunately, no security result, even in the random oracle model, or the generic model.

$\text{Twin-DSA}_x(m) = ((c, d), (c', d'))$,
where (c, d) and (c', d') are two distinct signatures of m (with different random u, u')

Twin-DSA is secure in the generic model

An RSA-based example: GHR

$n = pq$ product of large primes
$y \in \mathbf{Z}_n$: **public** element

- For signing e, $\mathbf{S}_{p,q}(e) = s$, where
  $d = e^{-1} \bmod \varphi(n)$, $s = y^d \bmod n$
- Verification, $\mathbf{V}_y(e,s) : s^e = y \bmod n$
- EuroC' 99: $\mathrm{GHR}_{p,q}(m) = \mathbf{S}_{p,q}(h(m))$
  if h is divisible-intractable + chameleon
  $\Rightarrow$ no existential forgeries against
  adaptive chosen-message attacks

Twin-GHR

- The chameleon property of h is required for simulating the signing oracle
  $\Rightarrow$ without it, no security against
  chosen-message attacks
- $\text{Twin-GHR}_{p,q}(m,a||b) = (\mathbf{S}_{p,q}(e_1), \mathbf{S}_{p,q}(e_2))$
  for $e_i = h(m_i)$ where
  $m_1 = (m \oplus a) \,||\, (m \oplus b)$ and $m_2 = a \,||\, b$
- Verification: get m_1 and m_2, and $M = m_1 \oplus m_2$, check the redundancy $M = m \,||\, m$, output m

Twin-GHR: Security

The twinning replaces
the chameleon property:
if h simply achieves divisible-intractability
(or injection in the primes)

Twin-GHR prevents existential forgeries even against adaptive chosen-message attacks

- no generic model
- no random oracle
- just the flexible RSA problem.

Conclusion

Twinning is a new paradigm to

- prevent existential forgeries (*cf.* DSA)
  it may replace the random oracle model in some situations
- achieves security against adaptive chosen-message attacks (*cf.* GHR)
  it may replace chameleon hash function or the random oracle model
- this new direction should be more investigated.