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

⇒ 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 \( \sigma \)

◆ the verification algorithm \( V \):
  
  public key + message \( m \) + alleged signature \( \sigma \)

  \( \rightarrow \) agrees or not
Digital signatures

Signing algorithm \( S \)
Verification algorithm \( V \)

\[ m \rightarrow S \rightarrow \sigma \]

\[ V \rightarrow \text{True/False} \]

Private key \( S \)
Public key \( P \)

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

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

\[ n = pq \text{ product of large primes} \]
\[ e : \text{public exponent} \]
\[ d = e^{-1} \mod \phi(n) : \text{private exponent} \]

Signature of the message \( m \in \mathbb{Z}_n \)
\[ \sigma = m^d \mod n \]

Verification of \((m, \sigma)\)

test whether \( m = \sigma^e \mod n \)

\begin{itemize}
  \item Only small messages (in \( \mathbb{Z}_n \)) can be signed
  \item Existentially forgeable
\end{itemize}

\( \Rightarrow \) in order to solve the former problem:

use of a collision-resistant hash function \( h \)

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

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 $\sigma$ it is easy to compute $m$ such that $m = \sigma^e \mod n$

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

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 $r$
- $\sigma_1 = S(m_1)$ and $\sigma = S(m_2)$

We thus sign two related messages
A DL-based example: DSA

\[ G = \langle g \rangle \text{ of prime order } q \]
\[ x : \text{secret key} \quad y = g^x : \text{public key} \]

- For signing \( m \in \mathbb{Z}_q \), \( S_x(m) = (c,d) \), where
  \[ 0 < u < q \quad c = (g^u) \mod q \quad c \neq 0 \]
  \[ \text{and} \quad d = (m + x \cdot c) / u \mod q \quad d \neq 0 \]

- Verification, \( V_y(m,c,d) : \)
  \[ h = 1/d \mod q, \quad h_1 = h \cdot m \mod q, \]
  \[ h_2 = h \cdot c \mod q, \quad c' = g^{h_1} \cdot y^{h_2} \]
  \[ \text{check whether } 0 < c, d < q \text{ and } c = c' \mod q \]

Twin-DSA

\( DSA_x(m) = S_x(\text{SHA}(m)) \)

- 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

\[ n = pq \text{ product of large primes} \]
\[ y \in \mathbb{Z}_n: \text{public element} \]

◆ For signing \( e, \quad S_{p,q}(e) = s \), where
\[ d = e^{-1} \mod \varphi(n), \quad s = y^d \mod n \]
◆ Verification,
\[ V_y(e,s) : s^e = y \mod n \]
◆ EuroC’ 99:
\[ \text{GHR}_{p,q}(m) = S_{p,q}(h(m)) \]
if \( h \) is divisible-intractable + chameleon
\[ \Rightarrow \text{no existential forgeries against adaptive chosen-message attacks} \]

Twin-GHR

◆ The chameleon property of \( h \) is required for simulating the signing oracle
\[ \Rightarrow \text{without it, no security against chosen-message attacks} \]
◆ Twin-GHR\(_{p,q}(m,a\|b) = (S_{p,q}(e_1), S_{p,q}(e_2)) \)
for \( e_i = h(m_i) \) where
\[ m_1 = (m\oplus a) \| (m\oplus b) \text{ and } m_2 = a \| b \]
◆ 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.