Digital Signature

Yu Zhang

HIT/CST/NIS

Cryptography, Autumn, 2014

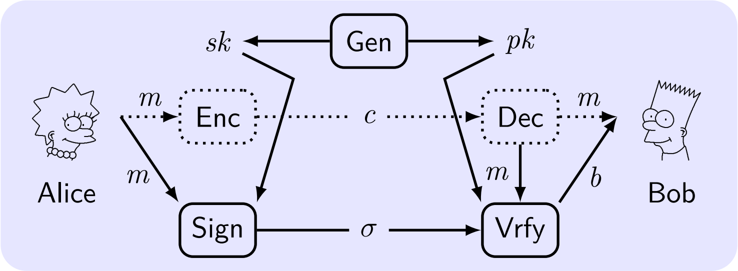
Outline

Content

Digital Signatures – An Overview

- **Digital signature scheme** is a mathematical scheme for demonstrating the authenticity/integrity of a digital message.
- allow a **signer** S to "**sign**" a message with its own sk, anyone who knows S's pk can **verify** the authenticity/integrity.
- (Comparing to MAC) digital signature is:
  - publicly verifiable.
  - transferable.
  - non-repudiation.
  - but slow.
- Digital signature is NOT the "inverse" of public-key encryption.

The Syntax of Digital Signature Scheme



- signature σ, a bit b means valid if $b = 1$; invalid if $b = 0$.
- **Key-generation** algorithm $(pk, sk) \leftarrow \mathsf{Gen}(1^n), |pk|, |sk| \geq n$.
- **Signing** algorithm $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$.
- **Verification** algorithm $b := \mathsf{Vrfy}_{pk}(m, \sigma)$.
- **Basic correctness requirement**: $\mathsf{Vrfy}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1$.

Defining of Signature Security

The signature experiment $\mathsf{Sigforge}_{\mathcal{A},\Pi}(n)$:

1. $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given input 1^n and oracle access to $\mathsf{Sign}_{sk}(\cdot)$, and outputs (m, σ). $\mathcal{Q}$ is the set of queries to its oracle.
3. $\mathsf{Sigforge}_{\mathcal{A},\Pi}(n) = 1 \iff \mathsf{Vrfy}_{pk}(m, \sigma) = 1 \wedge m \notin \mathcal{Q}$.

Definition 1

A signature scheme Π is **existentially unforgeable under an adaptive CMA** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that:

$$\Pr[\mathsf{Sigforge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Content

"Textbook RSA"

Construction 2

- Gen: *on input* 1^n *run* GenRSA(1^n) *to obtain* N, e, d.
  $pk = \langle N, e \rangle$ *and* $sk = \langle N, d \rangle$.
- Sign: *on input* sk *and* $m \in \mathbb{Z}_N^*$, $\sigma := [m^d \bmod N]$.
- Vrfy: *on input* pk *and* $m \in \mathbb{Z}_N^*$, $m \stackrel{?}{=} [\sigma^e \bmod N]$.

Insecurity

The "textbook RSA" is insecure.

Insecurity of "Textbook RSA"

- **A no-message attack**: choose an arbitrary $\sigma \in \mathbb{Z}_N^*$ and compute $m := [\sigma^e \bmod N]$. Output the forgery (m, σ).
- **Forging a signature on an arbitrary message**:
  To forge a signature on m, choose a random m_1, set $m_2 := [m/m_1 \bmod N]$, obtain signatures σ_1, σ_2 on m_1, m_2. $\sigma := [\sigma_1 \cdot \sigma_2 \bmod N]$ is a valid signature on m.

$$\sigma^e \equiv (\sigma_1 \cdot \sigma_2)^e \equiv (m_1^d \cdot m_2^d)^e \equiv m_1^{ed} \cdot m_2^{ed} \equiv m_1 m_2 \equiv m \pmod N.$$

Hashed RSA

- Gen: a hash function $H : \{0,1\}^* \to \mathbb{Z}_N^*$ is part of public key.
- Sign: $\sigma := [H(m)^d \bmod N]$.
- Vrfy: $\sigma^e \stackrel{?}{=} H(m) \bmod N$.

If H is not efficiently invertible, then the no-message attack and forging a signature on an arbitrary message is difficult.

Insecurity

There is NO known function H for which hashed RSA signatures are secure.

RSA-FDH Signature Scheme: Random Oracle as a **Full Domain Hash (FDH)** whose image size = the RSA modulus $N - 1$.

The "Hash-and-Sign" Paradigm

Construction 3

$\Pi = (\mathsf{Gen}_S, \mathsf{Sign}, \mathsf{Vrfy})$, $\Pi_H = (\mathsf{Gen}_H, H)$. *A signature scheme* Π'*:*

- Gen'*: on input* 1^n *run* $\mathsf{Gen}_S(1^n)$ *to obtain* (pk, sk)*, and run* $\mathsf{Gen}_H(1^n)$ *to obtain* s*. The public key is* $pk' = \langle pk, s\rangle$ *and the private key is* $sk' = \langle sk, s\rangle$*.*
- Sign'*: on input* sk' *and* $m \in \{0,1\}^*$*,* $\sigma \leftarrow \mathsf{Sign}_{sk}(H^s(m))$*.*
- Vrfy'*: on input* pk'*,* $m \in \{0,1\}^*$ *and* σ*, output 1* $\iff$ $\mathsf{Vrfy}_{pk}(H^s(m), \sigma) = 1$*.*

Theorem 4

If Π *is existentially unforgeable under an adaptive CMA and* Π_H *is collision resistant, then Construction is existentially unforgeable under an adaptive CMA.*

Content

One-Time Signature (OTS)

One-Time Signature (OTS): sign only one message with one secret.

The OTS experiment $\mathsf{Sigforge}^{\text{1-time}}_{\mathcal{A},\Pi}(n)$:

1. $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given input 1^n and a single query m' to $\mathsf{Sign}_{sk}(\cdot)$, and outputs (m, σ), $m \neq m'$.
3. $\mathsf{Sigforge}^{\text{1-time}}_{\mathcal{A},\Pi}(n) = 1 \iff \mathsf{Vrfy}_{pk}(m, \sigma) = 1$.

Definition 5

A signature scheme Π is **existentially unforgeable under a single-message attack** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that:

$$\Pr[\mathsf{Sigforge}^{\text{1-time}}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Lamport's OTS

Idea: OTS from OWF; one mapping per bit.

Construction 6

f is a one-way function.

- Gen*: on input 1^n, for $i \in \{1, \ldots, \ell\}$:*
  1. *choose random $x_{i,0}, x_{i,1} \leftarrow \{0,1\}^n$.*
  2. *compute $y_{i,0} := f(x_{i,0})$ and $y_{i,1} := f(x_{i,1})$.*

  $$pk = \begin{pmatrix} y_{1,0} & y_{2,0} & \cdots & y_{\ell,0} \\ y_{1,1} & y_{2,1} & \cdots & y_{\ell,1} \end{pmatrix} \quad sk = \begin{pmatrix} x_{1,0} & x_{2,0} & \cdots & x_{\ell,0} \\ x_{1,1} & x_{2,1} & \cdots & x_{\ell,1} \end{pmatrix}.$$

- Sign*: $m = m_1 \cdots m_\ell$, output $\sigma = (x_{1,m_1}, \ldots, x_{\ell,m_\ell})$.*
- Vrfy*: $\sigma = (x_1, \ldots, x_\ell)$, output $1 \iff f(x_i) = y_{i,m_i}$, for all i.*

Theorem 7

If f is OWF, Π is OTS for messages of length polynomial ℓ.

Example of Lamport's OTS

Signing $m = 011$

$$sk = \begin{pmatrix} x_{1,0} & x_{2,0} & x_{3,0} \\ x_{1,1} & x_{2,1} & x_{3,1} \end{pmatrix} \implies \sigma = (x_{1,0}, x_{2,1}, x_{3,1})$$

$\sigma = (x_1, x_2, x_3)$:

$$pk = \begin{pmatrix} y_{1,0} & y_{2,0} & y_{3,0} \\ y_{1,1} & y_{2,1} & y_{3,1} \end{pmatrix} \implies \begin{array}{l} f(x_1) \stackrel{?}{=} y_{1,0} \\ f(x_2) \stackrel{?}{=} y_{2,1} \\ f(x_3) \stackrel{?}{=} y_{3,1} \end{array}$$

A Stateless Solution

Idea: use deterministic randomness to emulate the state of tree.

Use PRF F and two keys k, k' (secrets) to generate pk_w, sk_w:

1. compute $r_w := F_k(w)$.
2. compute $(pk_w, sk_w) :=$ Gen$(1^n; r_w)$, using r_w as random coins.

k' is used to generate r'_w that is used to compute σ_w.

Lemma 8

If OWF exist, then $\exists$ OTS (for messages of arbitrary length).

Theorem 9

If OWF exists, then $\exists$ (stateless) secure signature scheme.

Content

Construction of DSS

DSS uses Digital Signature Algorithm (DSA, a variant of ElGamal signature scheme). [FIPS 186]

Construction 10

$\mathcal{G}$ outputs (p, q, g): (1) p and q are primes with $\|q\| = n$;
(2) $q|(p-1)$ but $q^2 \nmid (p-1)$;
(3) g is a generator of the subgroup of $\mathbb{Z}_p^*$ of order q.

- Gen: $(p, q, g) \leftarrow \mathcal{G}$. *hash function* $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$.
  $x \leftarrow \mathbb{Z}_q$ *and* $y := [g^x \bmod p]$.
  $pk = \langle H, p, q, g, y \rangle$. $sk = \langle H, p, q, g, x \rangle$.
- Sign: $k \leftarrow \mathbb{Z}_q^*$ *and* $r := [[g^k \bmod p] \bmod q]$,
  $s := [(H(m) + xr) \cdot k^{-1} \bmod q]$. *Output* (r, s).
- Vrfy: $u_1 := [H(m) \cdot s^{-1} \bmod q]$, $u_2 := [r \cdot s^{-1} \bmod q]$.
  Output $1 \iff r \stackrel{?}{=} [[g^{u_1} y^{u_2} \bmod p] \bmod q]$.

Correctness and Security of DSS

$r = [[g^k \bmod p] \bmod q]$ and $s = [(\hat{m}+xr)\cdot k^{-1} \bmod q],\ \hat{m} = H(m).$

$$\begin{aligned} g^{\hat{m}s^{-1}} y^{rs^{-1}} &= g^{\hat{m}\cdot(\hat{m}+xr)^{-1}k} g^{xr\cdot(\hat{m}+xr)^{-1}k} \pmod{p} \\ &= g^{(\hat{m}+xr)\cdot(\hat{m}+xr)^{-1}k} \pmod{p} \\ &= g^k \pmod{p}. \end{aligned}$$

$$[[g^k \bmod p] \bmod q] = r.$$

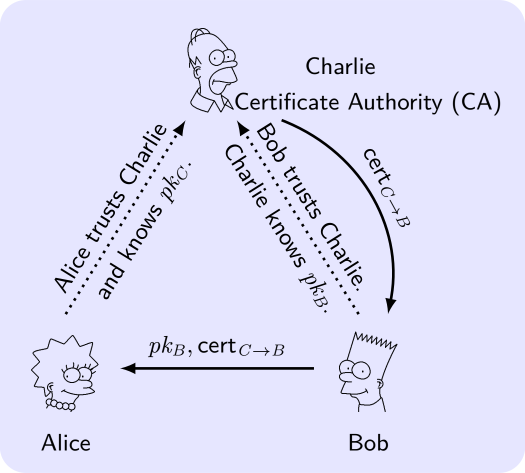
Security of DSS relies on the hardness of discrete log problem.
The entropy, secrecy and uniqueness of k is critical.

Insecurity

No proof of security for DSS based on discrete log assumption.

Content

Certificates



Certificates $\text{cert}_{C\to B} \stackrel{\text{def}}{=} \text{Sign}_{sk_C}(\text{'Bob's key is } pk_B\text{'})$.

Public-Key Infrastructure (PKI)

- **A single CA**: is trusted by everybody.
  - Strength: simple
  - Weakness: single-point-of-failure
- **Multiple CAs**: are trusted by everybody.
  - Strength: robust
  - Weakness: cannikin law
- **Delegation and certificate chains**: The trust is transitive.
  - Strength: ease the burden on the root CA.
  - Weakness: difficult for management, cannikin law.
- **"Web of trust"**: No central points of trust, e.g., PGP.
  - Strength: robust, work at "grass-roots" level.
  - Weakness: difficult to manage/give a guarantee on trust.

Invalidating Certificates

- **Expiration**: include an *expiry date* in the certificate.

$$\text{cert}_{C \to B} \stackrel{\text{def}}{=} \text{Sign}_{sk_C}(\text{`bob's key is } pk_B\text{'}, \text{ date}).$$

- **Revocation**: explicitly revoke the certificate.

$$\text{cert}_{C \to B} \stackrel{\text{def}}{=} \text{Sign}_{sk_C}(\text{`bob's key is } pk_B\text{'}, \ \#\#\#).$$

"###" represents the serial number of this certificate.
Cumulated Revocation: CA generates *certificate revocation list* (CRL) containing the serial numbers of all revoked certificates, signs CRL with the current date.

Summary

- Textbook RSA, Hashed RSA, Hash-and-Sign, Lamport's OTS, DSS.
- Stateful/Chain-based/Tree-based/Stateless Signature Scheme.
- Certificates, PKI, CA, Web-of-trust, Invalidation.