Admin: PSet #4 out today

Today:
- Digital Signatures
- Security of digital signatures
- Hash & Sign
- RSA-FDH
- RSA-PSS
- El-Gamal digital sigs
- DSA-NIST standard

Digital Signatures

- Invented by Diffie & Hellman in 1976 ("New Directions in Cryptography").

Idea:
- signature depends on the msg.
- How to verify?

  Each user has a pair of keys (PK, SK).

  PK is public, SK is kept secret.

  Use PK to verify, & use SK to sign.

- First implementation: RSA (1977)

Current way of describing digital signatures:

Def: A signature scheme consists of 3 algorithms

- $KeyGen(1^\lambda) \longrightarrow (PK, SK)$

  security parameter — verification key — secret key

- $Sign(SK, m) \longrightarrow \sigma_{SK}(m)$ (may be randomized).

- $Verify(PK, m, \sigma) = 1/0$ (acc/reject).

Correctness: $\forall m$, $Verify(PK, m, Sign(SK, m)) = 1$
$\forall (PK, SK) \leftarrow KeyGen(1^\lambda)$

Security:

1. Weak existential unforgeability against adaptive chosen message attacks

   (i) Challenger generates $(PK, SK) \leftarrow KeyGen(1^\lambda)$, and sends PK to Adversary.

(ii) Adversary obtains signature to a sequence of msgs of his choice:

$$m_1, m_2, \ldots, m_q \quad , \quad q = poly(\lambda),$$

where m_i can depend on signatures to $m_1, \ldots, m_{i-1}$ (i.e., adaptive).

Let $\sigma_i = \text{Sign}(SK, m_i)$.

(iii) Adversary outputs a pair (m, σ^*).

Adversary wins if:

- $\text{Verify}(PK, m, \sigma^*) = 1$

&

- $m \notin \{m_1, \ldots, m_q\}$.

Scheme is <u>secure</u> (i.e., weakly existentially unforgeable against adaptive chosen msg attacks) if

$$\Pr[\text{Adv wins}] = negl(\lambda).$$

Scheme is <u>strongly secure</u> if adv can't even produce a new signature for a msg that was previously signed for him.

Namely: Adv wins if

- $\text{Verify}(PK, m, \sigma^*) = 1$

& $(m, \sigma^*) \notin \{(m_1, \sigma_1), \ldots, (m_q, \sigma_q)\}$.

Hash & Sign:

For efficiency reasons, often better to sign $h(msg)$ rather than msg (where h is a cryptographic hash function), since hashing (say, SHA256) is extremely efficient compared to signing operations (such as modular exponentiation).

- Hash function should be collision resistant.
- Claim: If (KeyGen, Sign, Verify) is secure & h is collision resistant then the hash& Sign version of (KeyGen, Sign, Verify) is also secure.
- Interestingly: Hash & Sign paradigm is also useful for security!

Signing with RSA

Diffie & Hellman (1976) suggested a (general) method for using any (det.) public key encryption scheme as a signature scheme:

Idea: $\text{Sign}(SK, m) = \text{Dec}(SK, m)$
$\text{Verify}(PK, m, \sigma) = 1$ iff $\text{Enc}(PK, \sigma) = m$

First Attempt:

- KeyGen(1^λ): choose $n = p \cdot q$ p, q random λ-bit primes.

  choose e, d s.t. $e \cdot d = 1 \mod \varphi(n)$.

  $pk = (n, e)$

  $sk = (n, d)$.

- Sign$(sk, m) = m^d \mod n$.

- Verify$(pk, m, \sigma) = 1$ iff $(\sigma^e) = m \mod n$.

  Note: $(m^d)^e = m^{d \cdot e} = m \mod n$ ✓

Is this secure?

No! Since given sign(sk, m) one can easily sign $2m$.

What if we use hash & sign?

Given $(m_1, (h(m_1))^d \bmod n), \ldots, (m_q, h(m_q)^d \bmod n)$

is it easy to sign a new msg?

<u>Ans.</u>: Depends on h...

<u>Bellare-Rogaway 93</u>:

"Random oracles are practical: a paradigm for designing efficient protocols."

<u>Idea</u>: Think of the hash function as being truly <u>random</u>.

I.e., as a random oracle. This is called the random oracle model (ROM).

Prove security in ROM.

Full Domain Hash (FDH)

Hash & Sign RSA with $h: \mathbb{Z}_n^* \to \mathbb{Z}_n^*$

$$\mathrm{Sign}(\underset{(n,d)}{SK}, m) = (h(m))^d \bmod n$$

$$\mathrm{Verify}(\underset{(n,e)}{PK}, m, \sigma) = 1 \text{ iff } (\sigma^e) \bmod n = h(m)$$

[BR93] Proved that FDH is secure in the ROM. assuming RSA func. is hard to invert.

However, security reduction was not tight...

Loosely speaking, if the RSA function is (t', ε')-secure

$\forall$adv running in time $\le t'$ can invert w.p. $\le \varepsilon'$

then FDH sig scheme is $(t, q_{SIG}, q_{hash}, \varepsilon)$-secure

$\forall$adv running in time $\le t$ making $\le q_{SIG}$ signature calls & $\le q_{hash}$ hash calls can forge a new sig w.p. $\le \varepsilon$

where $t = t' - \text{poly}(q_{SIG}, q_{hash}, \lambda)$

$$\varepsilon = (q_{sig} + q_{hash}) \cdot \varepsilon'$$

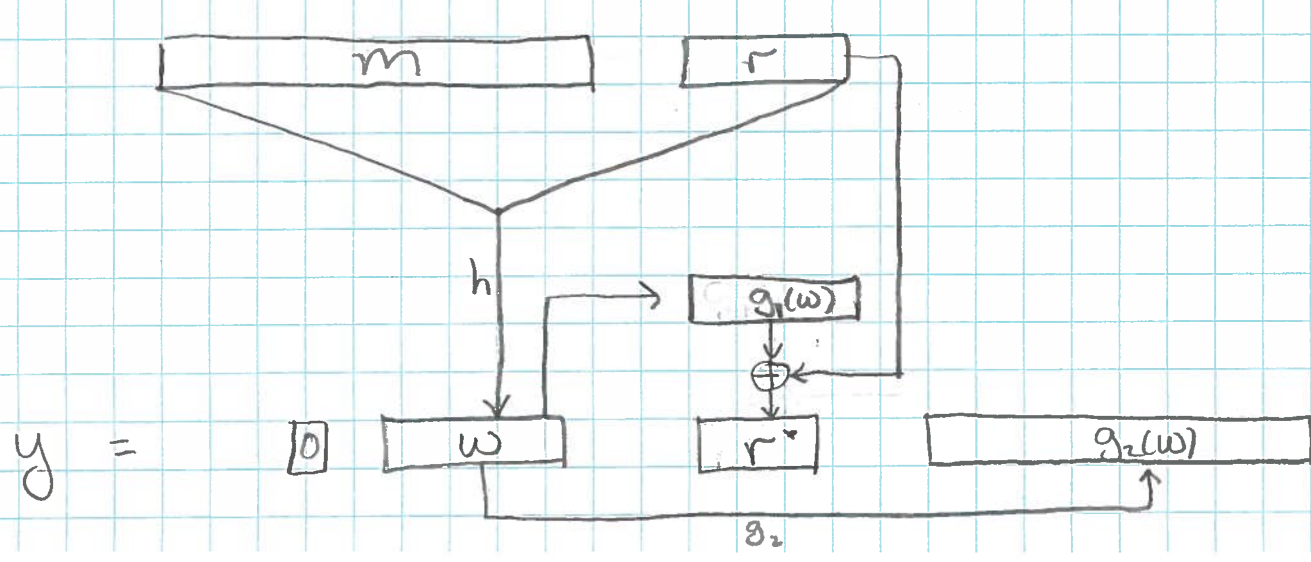
$\leftarrow$ ε is much larger than ε'

PSS - Probabilistic Signature Scheme [BR96]

RSA based:

$$\underset{\underset{(n,d)}{\shortparallel}}{\text{Sign}(sk, m)} = y^d \bmod n$$

$y = ?$ probabilistic hash of m



Namely, $y = 0 \| \omega \| r^* \| g_2(\omega)$

$r \leftarrow \{0,1\}^{k_0}$

$\omega \leftarrow h(m \| r) \qquad |\omega| = k_1$

$r^* = g_1(\omega) \oplus r \qquad |r^*| = k_0$

$y = 0 \| \omega \| r^* \| \underbrace{g_2(\omega)}_{k-(k_0+k_1+1)} \qquad |y| = 1 + k_1 + k_0 + k - (k_0+k_1+1) = k.$

$\text{Sign}(\underset{(n,d)}{SK}, m) = y^d \bmod n$

Verify(PK, m, σ):

compute $y = \sigma^e \bmod n$

Parse $y = \underbrace{b}_{1} \| \underbrace{\omega}_{k_1} \| \underbrace{r^*}_{k_0} \| \underbrace{\gamma}_{k-(k_0+k_1+1)}$

Let $r = g_1(\omega) \oplus r^*$

Output 1 iff $h(m, r) = \omega$ & $g_2(\omega) = \gamma$ & $b = 0$.

Thm: If h, g_1, g_2 are modelled as RO then PSS is existentially unforgeable against adaptive chosen msg attacks, assuming the RSA function is one-way (i.e., hand to invert on random inputs).

El-Gamal digital signatures

Note: The paradigm $Enc(dec(m)) = m$ doesn't work for El-Gamal (since El-Gamal is randomized)

New Scheme: PP: - prime p

- generator g of $\mathbb{Z}_p^*$

KeyGen: $x \leftarrow \{0, 1, \ldots, p-2\}$ $\quad SK = x$

$y = g^x \bmod p \quad PK = y$

$\text{Sign}(PP, \overset{x}{\overset{\shortparallel}{SK}}, m)$:

- Compute $h(m)$ assume range of h is $\mathbb{Z}_{p-1}$
- Choose $k \leftarrow \mathbb{Z}_{p-1}^*$
- Compute $r = g^k \pmod{p}$
- Compute $s = \frac{h(m) - rx}{k} \pmod{p-1}$

$\sigma(m) = (r, s)$

$\text{Verify}(\underset{(p,g)}{\underset{\shortparallel}{PP}}, \underset{y}{\underset{\shortparallel}{PK}}, m, (r, s))$:

- Check that $0 < r < p$
- Check that $\underbrace{y^r \cdot r^s}_{g^{x \cdot r} \cdot g^{k\left(\frac{h(m) - rx}{k}\right)}} = g^{h(m)} \pmod{p}$

correctness:

Security: With h = idenity, it is not secure (it is existentially forgeable).

Pf: Let $r = g^e \cdot y \bmod p$ for $e \leftarrow \mathbb{Z}_{p-1}$

$$s = -r$$

Then (r, s) is a valid El-Gamal sig of $m = e \cdot s \pmod{p-1}$

Check: $y^r \cdot r^s \stackrel{?}{=} g^m$

$$y^r \cdot (g^e \cdot y)^{-r} = g^{es}$$

What about security in ROM?

Not known how to reduce to DL problem.

Pointcheval-Stern 96: Modified version of El-Gamal:

Sign(m): $k \leftarrow \mathbb{Z}_p^*$

$$r = g^k \pmod p$$

$$s = \frac{h(m \| r) - rx}{k} \bmod (p-1)$$

$$\sigma = (r, s)$$

Verify: Check $0<r<p$ & $y^r \cdot r^s = g^{h(m\|r)}$

<u>Thm</u>. Modified El-Gamal is existentially unforgeable against adaptive chosen msg attacks, in ROM, assuming DLP is hard (on avg).

Digital Signature Standard (DSS-NIST 91)

<u>Public Params</u>:
- q prime $|q| = 160$ bits
- $p = n \cdot q + 1$ $|p| = 1024$ bits
- g_0 - generates $\mathbb{Z}_p^*$
- $g = g_0^n$ generates subgroup of $\mathbb{Z}_p^*$ of order q.

<u>KeyGen</u>:
- Choose $x \leftarrow \mathbb{Z}_q$ $SK = x$ $|x| = 160$ bits
- $y = g^x \bmod p$ $PK = y$ $|y| = 1024$ bits

<u>Sign (m)</u>:

$k \leftarrow \mathbb{Z}_q^*$ (i.e. $1 \le k < q$)

$r = g^k \pmod p \pmod q$ $|r| = 160$ bits

$s = \frac{h(m) + rx}{k} \pmod q$ $|s| = 160$ bits.

redo if $r = 0$ or $s = 0$

$\sigma(m) = (r, s)$.

Verify: • Check $0 < r, s < q$

• Check $y^{r/s} \cdot g^{h(m)/s} \pmod p \pmod q = r$

Correctness: $y^{r/s} \cdot g^{h(m)/s}$

$$= g^{\frac{xr + h(m)}{s}} = g^{k} = r \pmod p \pmod q$$

Security:

As before, insecure w. h = identity.

Provably secure if $h(m)$ is replaced w. $h(m\|r)$
(as with modified El-Gamal).