Today:

- Signatures:
  - Recall definition
  - Hash & Sign paradigm
  - RSA signatures & full domain hash
  - El-Gamal signature scheme

Security:

Def: Existential unforgeability against adaptive chosen message attacks:

(i) Challanger generates $(PK, SK) \leftarrow KeyGen(1^\lambda)$

(ii) Adversary obtains oracle access to $Sign(SK, \cdot)$.

I.e., adversary obtains signatures to a sequence of msgs of his choice: $m_1, \ldots, m_q$ $q = poly(\lambda)$, where m_i can depend on signatures to $m_1, \ldots, m_{i-1}$.

Let $\sigma_i = Sign(SK, m_i)$

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

Adversary wins if

- $Verify(PK, m, \sigma^*) = 1$
- $m \notin \{m_1, \ldots, m_q\}$

Def: A scheme is <u>secure</u> (i.e., existentially unforgeable against adaptive chosen msg attacks) if

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

Def: A scheme is <u>strongly secure</u> if adv. cannot 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~~ sign $h(msg)$ rather than msg (where h is a cryptographic hash function), since hashing (say, SHA 256) is extremely efficient compared to signing operations (such as modular exponentiations).

* Hash function needs to be collision resistant !

<u>Claim</u> : If (KeyGen, Sign, Verify) is secure & $\mathcal{H} = \{h_k\}$ is collision resistant hash family, then the hash & sign version of (KeyGen, Sign, Verify) is also secure.

<u>Interestingly</u> : Hash & Sign paradigm is also useful for security !

Signing with RSA

Diffie & Hellman (1976) suggested a (general) method for using ~~and~~ a (deterministic) public-key encryption scheme as a signature scheme:

Idea: $\mathrm{Sign}(sk, m) = \mathrm{Dec}(sk, m)$

$\mathrm{Verify}(pk, m, \sigma) = 1$ iff $\mathrm{Enc}(pk, \sigma) = m$

Signing with RSA: Fist Attempt

$\mathrm{KeyGen}(1^\lambda)$: Choose $n = p \cdot q$ $\quad$ p, q random λ-bit primes.

Choose e, d random s.t. $e \cdot d = 1 \bmod \varphi(n)$.

$PK = (n, e)$

$SK = (n, d)$

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

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

Correctness $\forall m \in \mathbb{Z}_n$

$(m^d)^e = m^{de} = m \bmod n$ ✓

Is this secure? No!

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

one can easily sign $m^2 \bmod n$.

Idea: Use hash & sign

$\text{Sign}((sk, h), m) = (h(m))^d \bmod n.$

$\text{Verify}((pk, h), m, \sigma) = 1$ iff $\sigma^e = h(m) \bmod n.$

Is this secure??

Depends on h ...

Bellare-Rogaway 93:

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

Introduced ROM (Random Oracle Model).

[BR93] Proved that Hash & Sign RSA (a.k.a full domain hash FDH) is secure in the ROM assuming RSA assumption (i.e. RSA func. is hard to invert on avg).

(Generalizes to any trapdoor permutation...)

Security reduction is not tight ...

Loosely speaking, if RSA function is (t', ε')-secure
(i.e. $\forall$adv running in time t' can invert w.p. $\leq \varepsilon'$)

then FDH scheme is $(t, q_{SIG}, q_{hash}, \varepsilon)$-secure
(i.e., $\forall$adv running in time t, making $\leq q_{SIG}$ signature calls & $\leq q_{hash}$ hash calls, can forge a new signature w.p. $\leq \varepsilon$)

where:

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

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

Probabilistic Signature Scheme (PSS)

[Bellare-Rogaway 96]

RSA-based signature scheme secure in the ROM with tighter security proof.

$$m, r \xrightarrow[\text{encoding using ROM}]{} y \longrightarrow y^d \bmod n$$

El-Gamal Signatures

Note: The paradigm Enc(Dec(m)) doesn't work for El-Gamal, since El-Gamal is not a trapdoor permutation (it is randomized).

Scheme: PP: prime p

g generator of prime order subgroup q (order $q \mid p-1$).

KeyGen: $x \leftarrow \mathbb{Z}_q$ $\qquad$ $SK = x$

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

Sign(PP, SK, m):

- Choose $k \leftarrow \mathbb{Z}_q^*$
- Output $(r, s) = \left(g^k \bmod p, \ \dfrac{h(m) + rx}{k} \bmod q \right)$

Verify(PP, PK, m, (r, s)):

- Check that $0 < r < p$
- Check that $y^{r/s} \cdot g^{h(m)/s} = r$

Correctness:

$$y^{r/s}\, g^{h(m)/s} = g^{\frac{xr+h(m)}{s}} = g^{k} = r \bmod p$$

Security:

- Insecure with h = identity (exercise).
- Not known to be secure in ROM
- Secure in ROM if $h(m)$ is replaced with $h(m \| r)$ [Pointcheval – Stern 96]:

  [Intuition: If $h(m\|r)$ then adv. needs to choose r and succ for many values of $h(m\|r)$.
  $\Rightarrow$ knowledge of k. $\Rightarrow$ knowledge of sk]

Thm: 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)

Public Parameters: p prime, $q \mid p-1$

$|p| = 1024$ bits, $|q| = 160$ bits

g generator of subgroup of $\mathbb{Z}_p^*$ of order q.

<u>KeyGen</u>: $x \leftarrow \mathbb{Z}_q$ $\quad SK = x \quad |x| = 160$ bits

$y = g^x \quad PK = y \quad |y| = 1024$ bits

<u>$\text{Sign}_{SK}(m)$</u>: $k \leftarrow \mathbb{Z}_q^*$

$r = (g^k \bmod p) \bmod q \quad |r| = 160$ bits

$s = \dfrac{h(m) + rx}{k} \bmod q \quad |s| = 160$ bits

Redo if $r = 0$ or $s = 0$

Output (r, s).

<u>$\text{Verify}_{PK}(m, (r, s))$</u>:

- Check $0 < r, s < q$
- Check $y^{r/s} \cdot g^{h(m)/s} \pmod p \pmod q = r$

<u>Correctness</u>: $y^{r/s} g^{h(m)/s} = g^{\frac{xr + h(m)}{s}} = g^k =$

$r \pmod p \pmod q$.

<u>Security</u>: As before, provably secure if $h(m)$ is replaced with $h(m \| r)$.