<u>Today</u>: Digital Signature Schemes (Cont.).

- Hash & Sign
- El-Gamal signature scheme
- DSS (Digital Signature Standard)

<u>Recall</u>: A digital signature scheme w. msg space $\mathcal{M}$ consists of PPT alg's: (KeyGen, Sign, Verify)

- KeyGen(1^λ) generates (PK, SK)
- Sign(SK, m) generates a signature σ
- Verify(PK, m, σ) = 0/1 ("acc" or "rej")

<u>Correctness</u>: $\forall m \in \mathcal{M}$ for $(PK, SK) \leftarrow KeyGen(1^\lambda)$

$$\Pr[\,Verify(PK, m, Sign(SK, m)) = 1\,] = 1$$

<u>Security</u> (against adaptive chosen msg attacks): for $\forall$ PPT Adv, given PK and oracle to Sign(SK, $\cdot$), $(PK, SK) \leftarrow KeyGen(1^\lambda)$, denoting by $m_1, \ldots, m_q$ its oracle queries, the prob that $\mathcal{A}$ outputs (m^*, σ^*) s.t. $m^* \notin \{m_i\}$ & Verify$(PK, m^*, \sigma^*) = 1$, is negl.

<u>Last lecture</u>: <u>RSA</u> digital sig scheme.

Follows Diffie-Hellman blue print:

$$f_{n,e}: \mathbb{Z}_n \to \mathbb{Z}_n \qquad x \mapsto x^e \bmod n$$

$$\underset{(n,d)}{\mathrm{Sign}(\overset{\shortparallel}{SK}, m)} = f^{-1}_{n,e}(m) = m^d \bmod n$$

$e \cdot d = 1 \bmod \varphi(n)$.

<u>Correctness</u>: $\forall m \in \mathbb{Z}_n \quad (m^d)^e = m^{d \cdot e} = m \mod n$ ✓

<u>Not secure</u>: Given $\text{Sign}(sk, m) = m^d \mod n$

one can easily sign $m^2 \mod n \rightarrow (m^d)^2 \mod n$.

To make RSA secure use hash & sign:

Hash & Sign

Rather than signing m, sign $\underline{h(m)}$,
where h is a hash function (part of the public key)

* <u>Better efficiency</u>: Hashing is extremely eff compared to signing.
* Allow <u>flexibility</u>: signing any msg m $\{0,1\}^*$.
* <u>Interestingly</u>: Useful for <u>security</u>.

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

Moreover: : Hash & Sign paradigm enhances security for RSA

Hash & Sign with RSA

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

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

$$\sigma^e = h(m) \bmod n.$$

Is this secure ? Depends on h... [not secure if $m, h(m)^d \xrightarrow{\text{easy}} h(m^2)^d$

It is secure in the Random Oracle Model (if h is RO) [Bellare-Rogaway 93]

a.k.a. Full Domain Hash (FDH)

Intuition: pairs (m_i, σ_i) are dist. like (m_i, r_i) (random) where $h(m_i) = r_i^e \bmod n$.

Doesn't give any useful info in ROM $\equiv$ Can be simulated.

If Adv generates (m^*, σ^*) $\Longrightarrow$ Adv breaks RSA (in ROM)

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) (a.k.a. RSA-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 [1984]

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

$g \in \mathbb{Z}_p^*$ generator of prime order subgroup q (order $q \mid p-1$).

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

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

Sign(PP, SK, m):

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

$$\Downarrow$$

$$r^s = g^{h(m)+rx} = g^{h(m)} \cdot y^r$$

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

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

Correctness:

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

Security:

Idea: Generating a signature $\underbrace{\text{require}}_{\text{seems to}}$ knowledge of k and a signer that knows k must know $sk = x$.

- Insecure with h = identity (exercise). ← Original proposal

- 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 $h(m\|r)$. ⇒ knowledge of k. ⇒ knowledge of sk

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

* Rarely used in practice. The following variant is used instead.

Digital Signature Standard

(DSS-NIST 91) (a.k.a. Digital Signature Alg (DSA))

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.

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

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

$\text{Sign}_{SK}(m)$: $k \leftarrow \mathbb{Z}_q$

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

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

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

Output (r, s).

$\text{Verify}_{PK}(m, (r, s))$:

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

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

$r \pmod p \pmod q$.

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