Today:
- Digital Signatures
- Security of digital signatures
- Hash & Sign
- RSA - EDH
- RSA - PSS
- El-Gamal digital sign
- 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

- \( \text{KeyGen}(1^n) \rightarrow (PK, SK) \)
  - security parameter
  - verification key
  - secret key

- \( \text{Sign}(sk, m) \rightarrow \sigma_{sk}(m) \) (may be randomized)

- \( \text{Verify}(PK, m, \sigma) = 1/0 \) (acc/reject)

**Correctness:** \( \forall M, \exists \sigma: \text{Verify}(PK, m, \text{Sign}(sk, m)) = 1 \)

- \( (PK, SK) \leftarrow \text{KeyGen}(1^n) \)

**Security:**

Weak existential unforgeability against adaptive chosen message attacks

(i) Challenger generates \( (PK, SK) \leftarrow \text{KeyGen}(1^n) \), and sends \( PK \) to Adversary.
(iii) Adversary obtains signature to a sequence of msgs of his choice:

\[ m_1, m_2, \ldots, m_g, \quad g = \text{poly}(\lambda), \]

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

Let \( C_i = \text{Sign}(sk, m_i) \).

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

Adversary wins if:

- \( \text{Verify}(pk, m, s^*) = 1 \)
- \( m \not\in \{ m_1, \ldots, m_g \} \)

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

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

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

Namely: Adv wins if

- \( \text{Verify}(pk, m, s^*) = 1 \)
- \( m \not\in \{ (m_1, s_1), \ldots, (m_g, s_g) \} \).
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: $Sign(sk, m) = Dec(sk, m)$

Verify $(pk, m, s) = 1$ iff $Enc(pk, s) = m$
First Attempt:

- KeyGen($k$): choose $n=p\cdot q$ random $\lambda$-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 $(PK,mc) = 1 \iff (c^e) = m \mod n$.

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

Is this secure?

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

What if we use hash & sign?

Given $(m_1, (h(m_1))^d \mod n), \ldots, (m_g, (h(m_g))^d \mod n)$ is it easy to sign a new $msg$?
Ans.: Depends on \( h \)...

Bellare-Rogaway 93:

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

Idea: Think of the hash function as being truly random. 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 \( h: \mathbb{Z}_n^* \rightarrow \mathbb{Z}_n^* \)

\[
\text{Sign}(sk, m) = (h(m))^d \mod n
\]

\[
\text{Verify}(pk, m, \tau) = 1 \iff (\tau^e \mod n = h(m))
\]

[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 \((t', \varepsilon')\)-secure

\[ \text{Adv running in time } t' \]
\[ \text{can invert with } \omega < \varepsilon' \]

then FDH signature scheme is \((t', g_{\text{sig}}, g_{\text{hash}}, \varepsilon)\)-secure

\[ \text{Adv running in time } t \]
\[ \text{making } g_{\text{sig}} \text{ signature calls } \leq t \]
\[ \text{and } g_{\text{hash}} \text{ hash calls } \]
\[ \text{can forge a new sig with } \omega < \varepsilon \]

where

\[ t = t' - \text{poly}(g_{\text{sig}}, g_{\text{hash}}, \lambda) \]

\[ \varepsilon = (g_{\text{sig}} + g_{\text{hash}}) \cdot \varepsilon' \]

\[ \vdash \varepsilon \text{ is much larger than } \varepsilon' \]

**PSS - Probabilistic Signature Scheme** [BE96]

**RSA based:**

\[ \text{Sign}(sk, m) = y^d \mod n \]

\[(n, d)\]

\[ y = ? \text{ probabilistic hash of } m \]

---

![Diagram](image)
Namely, \[ y = \| w \| r^\ast \| g_2(w) \]

\[ r^\ast \leftarrow \{0,1\}^{k_0} \]
\[ w \leftarrow h(m\|r) \quad |w| = k_1 \]
\[ r^* = g_1(w) \oplus r \quad |r^*| = k_0 \]
\[ y = \| w \| r^\ast \| g_2(w) \]
\[ |y| = 1 + k_1 + k_0 + k - (k_0 + k_1 + 1) = k \]

\[ \text{Sign}(sk,m) = y^d \mod n \]

\[ (m',d') \]

\[ \text{Verify}(pk,m,r) : \]
compute \[ y = r^e \mod n \]
parse \[ y = b \| w \| r^\ast \| x \]
\[ = 1 \quad k_1 \quad k_0 \quad k - (k_0 + k_1 + 1) \]
let \[ r = g_1(w) \oplus r^* \]

Output 1 iff \[ h(m,r) = w \& g_2(w) = y \& b = 0. \]

Thm: If \( h, g, g_2 \) are modelled as RO then

PSS is existentially unforgeable against adaptive chosen msg attacks, assuming the RSA function
is one-way (i.e., hard to invert on random inputs).
El-Gamal digital signatures

Note: The paradigm \( \text{Enc}(\text{dec}(m)) = m \) doesn't work for El-Gamal (since El-Gamal is randomized).

New Scheme: PP: - prime \( p \)
- generator \( g \) of \( \mathbb{Z}_p^* \)

\[ \text{KeyGen}: \quad x \leftarrow \{0, 1, \ldots, p-2\} \quad \text{sk} = x \]
\[ y = g^x \mod p \quad \text{pk} = y \]

\[ x \]

\[ \text{Sign}(\text{PP}, \text{sk}, m): \]
- Compute \( h(m) \) assume range of \( h \) is \( \mathbb{Z}_{p-1} \)
- Choose \( k \leftarrow \mathbb{Z}_{p-1} \)
- Compute \( r = g^k \mod p \)
- Compute \( \alpha = \frac{h(m) - rx}{k} \mod (p-1) \)

\( \sigma(m) = (r, \alpha) \)

\[ \text{Verify}(\text{PP}, \text{pk}, m, (r, \alpha)) : \]
- (\( p, g, y \)) \( y' \)
- Check that \( 0 < r < p \)
- Check that \( y^r \cdot r^\alpha = g^{h(m)} \mod p \)

Correctness:
\[ g^x = g^{h(m) + \alpha x} \]
Security: With \( h = \text{identity} \), it is \textbf{not secure} (it is existentially forgeable).

**Proof:** Let \( r = g^e \cdot y \mod p \) for \( e \in \mathbb{Z}_{p-1} \)

\[ \beta = -r \]

Then \((r, \beta)\) is a valid El-Gamal sig of \( m = e \cdot \beta \mod (p-1) \)

\[ \text{Check:} \quad y^r \cdot r^\beta = g^m \]

\[ y^r \cdot (g^e \cdot y)^{-r} = g^{e \beta} \]

What about security in ROM?

Not known how to reduce to DL problem

\textbf{Pointcheval-Stern 92 :} Modified version of El-Gamal:

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

\[ r = g^k \mod p \]

\[ \theta = \frac{h(m||r) - r \cdot x}{k} \mod (p-1) \]

\( \sigma = (r, \theta) \)
Verify: Check $cx \cdot r = y \cdot r^o = g^{h(m)||r}$

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

**Digital Signature Standard (DSS-NIST91)**

**Public Params**
- $g$ prime
- $|g| = 160$ bits
- $P = n \cdot g + 1$
- $|P| = 1024$ bits
- $g_0$ generates $Z_P^*$
- $g = g_0^n$ generates subgroup of $Z_P^*$ of order $g$

**KeyGen**
- Choose $x \in Z_q$
- $SK = x$
- $|x| = 160$ bits
- $y = g^x \mod P$
- $PK = y$
- $|y| = 1024$ bits

**Sign(m)**
- $k \leftarrow Z_q^*$ (i.e. $1 \leq k < g$)
- $r = g^k \mod P \mod g$
- $|r| = 160$ bits
- $\delta = h(m) + rx \mod g$
- $|\delta| = 160$ bits
- redo if $r = 0$ or $\delta = 0$
- $\sigma(m) = (r, \delta)$.
Verify:
- Check \( 0 < r, a < g \)
- Check \( y^{1/a} \cdot g^{h(m)/p} \pmod{p} \pmod{g} = r \)

\[
\text{Correctness:} \quad y^{1/a} \cdot g^{h(m)/a} = g^x = g^{x \cdot h(m)} = g^k = r \pmod{p} \pmod{g}
\]

Security:
As before, insecure as \( h = \text{identity} \).

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