Admin:

Pset #4 out.

Quiz in-class Wed 4/16. Open notes (No laptops or books)

Projects: Next week: meet with TA to review progress.

Presentation schedule out soon (let us know if conflict)

Today:

- Digital signatures
- Security defn for digital signatures
- Hash & Sign
- RSA - PKCS
- RSA - PSS
- El Gamal dig. sigs.
- DSA - (NIST standard)
Digital Signatures (compare "electronic signature", "cryptographic signature")

- Invented by Diffie & Hellman in 1976
  ("New Directions in Cryptography")
- First implementation: RSA (1977)
- Initial idea: switch PK/SK
  (enc with secret key $\Rightarrow$ signature)
  (if PK decrypt; it & looks ok then sig ok??)

Current way of describing digital signatures

- $\text{Keygen}(1^l) \rightarrow (\text{PK}, \text{SK})$
  \[ \text{verification key} \rightarrow \text{signing key} \]
- $\text{Sign}(\text{SK}, m) \rightarrow \sigma_{\text{SK}}(m)$ \[\text{signature} \quad \text{[may be randomized]}\]
- $\text{Verify}(\text{PK}, m, \sigma) = \text{True/False (accept/reject)}$

Correctness:

$(\forall m) \text{Verify}(\text{PK}, m, \text{Sign}(\text{SK}, m)) = \text{True}$
Security of digital signature schemes:

**Def:** (weak) existential unforgeability under adaptive chosen message attack.

1. Challenger obtains \((PK, SK)\) from Keygen(\(\lambda\)).
   Challenger sends \(PK\) to Adversary.

2. Adversary obtains signatures to a sequence \(m_1, m_2, \ldots, m_g\) of messages of his choice. Here \(g = \text{poly}(\lambda)\), and \(m_i\) may depend on signatures to \(m_1, m_2, \ldots, m_{i-1}\).
   Let \(\sigma_i = \text{Sign}(SK, m_i)\).

3. Adversary outputs pair \((m, \sigma_x)\).

Adversary wins if \(\text{Verify}(PK, m, \sigma_x) = \text{True}\) and \(m \notin \{m_1, m_2, \ldots, m_g\}\).

Scheme is secure (i.e., weakly existentially unforgeable under adaptive chosen message attack) if
\[
\text{Prob}[\text{Adv wins}] = \text{negligible}
\]
Scheme is strongly secure if adversary can't even produce new signature for a message that was previously signed for him. I.e. Adv wins if \( \text{Verify}(PK, m, \sigma_{\text{Adv}}) = \text{True} \)
and \((m, \sigma_{\text{Adv}}) \notin \{(m_1, \sigma_1), (m_2, \sigma_2), \ldots, (m_g, \sigma_g)\}\).
Digital signatures

- Def of digital signature scheme
- Def of weak/strong existential unforgeability \{ see notes \}
  under adaptive chosen message attack, \{ from last lecture \}

Hash & Sign:

For efficiency reasons, usually best to sign

cryptographic hash \( h(M) \) of message, rather
than signing \( M \). Modular exponentiations are slow compared to (say) SHA-256.

Hash function \( h \) should be collision-resistant.
Signing with RSA - PKCS

- PKCS = "Public key cryptography standard"
  (early industry standard)
- Hash & sign method. Let \( H \) be CR hash fn.
- Given message \( M \) to sign:
  
  Let \( m = H(M) \)

Define \( \text{pad}(m) = \)

\[
\text{Ox } 00 \text{ 01 FF } \ldots \text{FF 00 || hash-name || m}
\]

where \# FF bytes enough to make \( |\text{pad}(m)| = |n| \) in bytes.
where hash-name is given in ASN.1 syntax (ugh!)

- Seems secure, but no proofs (even assuming \( H \) is CR
  and RSA is hard to invert)

\[
\sigma(M) = (\text{pad}(m))^d \pmod{n}
\]
PSS - Probabilistic Signature Scheme [Bellare & Rogaway, 1996]

- RSA-based
- "Probabilistic" = randomized [one M has many sigs]

\[
\sigma(M) = y^d \pmod{n}
\]

**Sign(M):**

\[
\begin{align*}
& r \leftarrow \mathcal{R}_{\mathbb{Z}_n} \\
& w \leftarrow h(M || r) \\
& r^* \leftarrow g_1(w) \oplus r \quad |r^*| = k_0 \\
& y \leftarrow \mathcal{O}(w||r^*||g_2(w)) \quad |y| = |n| \\
& \text{output } \sigma(M) = y^d \pmod{n}
\end{align*}
\]

**Verify(M, \sigma):**

\[
y \leftarrow \sigma^e \pmod{n}
\]

Parse \( y \) as \( b || w || r^* || \gamma \)

\[
\begin{align*}
& b = 0 & h(M || r) = w & g_2(w) = \gamma \\
& r^* \leftarrow r^* \oplus g_1(w)
\end{align*}
\]

return True iff \( b = 0 \& h(M || r) = w \& g_2(w) = \gamma \)
• We can model $h$, $g_1$, and $g_2$ as random oracles.

Theorem:

PSS is (weakly) existentially unforgeable against a chosen message attack in random oracle model if RSA is not invertible on random inputs.
El Gamal digital signatures

Public system parameters: prime $p$

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

Keygen: $x \leftarrow \mathbb{Z}_{p-1}$, $y = g^x \pmod{p}$, $SK = x$, $PK = y$

$Sign(M)$:

$m = \text{hash}(M)$

Pick $k \leftarrow \mathbb{Z}_{p-1}$, $r = g^k$

$s = \frac{m-\text{hash}(M) \cdot r}{k} \pmod{p-1}$

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

$Verify(M, x, (r, s))$:

Check that $0 < r < p$ (else reject)

Check that $y^r r^s = g^m \pmod{p}$

where $m = \text{hash}(M)$
Correctness of El Gamal signatures:

\[ y^r s = g^{rx} g^{sk} = g^{rx+sk} \equiv g^m \pmod{p} \]

\[ \equiv\]

\[ rx + ks \equiv m \pmod{p-1} \]

or

\[ s \equiv (m-rx) \pmod{p-1} \]

(assuming \( k \in \mathbb{Z}_{p-1}^* \))
Theorem: El Gamal signatures are existentially forgeable
(without $h$, or $h$=identity (note: this is CR!))

Proofs Let $e \leftarrow \mathbb{Z}_{p-1}$

$r \leftarrow g^e \cdot y \pmod{p}$

$s \leftarrow -r \pmod{p}$

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

Check:

$y^r r^s \equiv g^m$

$g^{xr} (g^e)^{r^e} = g^{-er} = g^{es} = g^m \checkmark$

But: It is easy to fix.

Modified El Gamal (Pointcheval & Stern 1996)

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

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

$m = h(M \| r)$

$s = (m-rx)/k \pmod{p-1}$

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

Verify: check $0 < r < p$ and $y^r r^s = g^m$ where $m = h(M \| r)$.

Theorem: Modified El Gamal is existentially unforgeable
against adaptive chosen message attack, in ROM,
assuming DLP is hard.
Digital Signature Standard (DSS - NIST 1991)

Public parameters (same for everyone):

- \( q \) prime, \(|q| = 160 \text{ bits} \)
- \( p = nq + 1 \) prime, \(|p| = 1024 \text{ bits} \)
- \( g_0 \) generates \( \mathbb{Z}_p^* \)
- \( g = g_0^k \) generates subgroup \( G_q \) of \( \mathbb{Z}_p^* \) of order \( q \)

Keygen:

- \( x \leftarrow \mathbb{Z}_q \) SK \(|x| = 160 \text{ bits} \)
- \( y \leftarrow g^x \mod p \) PK \(|y| = 1024 \text{ bits} \)

Sign(\( m \)):

- \( k \leftarrow \mathbb{Z}_q^* \) (i.e. \( 1 \leq k < q \) )
- \( r = (g^k \mod p) \mod q \) \(|r| = 160 \text{ bits} \)
- \( m = h(M) \)
- \( s = (m + rx) / k \mod q \) \(|s| = 160 \text{ bits} \)

Note: if \( k \) is reused for different messages \( m \), one could solve for \( x \) so it is not secure.

If \( k \) is reused for the same \( m \), we obtain the same signature so this is not a problem. If \( k \) is different for the same \( m \), it should be random and unknown (any known relation between the two \( k \)-s allows to solve for \( x \)).

Bottomline: All of the above are enforced by \( k \) chosen at random from \( \mathbb{Z}_q^{\omega} \) for large enough \( q \).
Verify:

Check $0 < r < q$ & $0 < s < q$

Check $y^{r/s} m^s (mod \ p)(mod \ q) = r$

where $m = h(M)$

Correctness:

$g^{(rx+m)/s} \equiv r (mod \ p)(mod \ q)$

$\equiv g^k \equiv r (mod \ p)(mod \ q)$  \ $\checkmark$

As it stands, existentially forgeable for $h = \text{identity}$. Provably secure (as with Modified El Gamal)

if we replace $m = h(M)$ by $m = h(M || r)$, as before.

Note: As with El Gamal, secrecy & uniqueness of $k$

is essential to security.