Admin:

Talk with TA this week about project.

Quiz: in-class 4/13 Wed. Open notes (No laptops or books)

Presentations start Wed 4/20/16

Today:

- Digital signatures
- Security of digital signatures
- Hash & Sign
- RSA - PKCS
- RSA - PSS
- El Gamal digital signatures
- 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

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

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

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

1. Challenger obtains (PK, SK) from KeyGen(1^t)
   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 \( g = \text{poly}(t) \),
   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 \((m, \sigma^*_m)\)

Adversary wins if \( \text{Verify}(PK, m, \sigma^*_m) = \text{True} \)
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 an adversary can't even produce a new signature for a message that was previously signed for him, i.e., Adv wins if \( \text{Verify}(PK, m, \sigma_x) = \text{True} \) and \( (m, \sigma_x) \notin \{(m_1, \sigma_1), (m_2, \sigma_2), \ldots, (m_n, \sigma_n)\} \).
Digital signatures:
- Def of digital signature scheme
- Def of weak/strong existential unforgeability under adaptive chosen message attack.

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)
- Given message M to sign:
  Let \( m = H(M) \)
  Define \( \text{pad}(m) = \)
  \[ 0x00 01 FF FF ... FF 00 \| \text{hash-name} \| m \]
  where \# FF bytes enough to make \( |\text{pad}(m)| = |n| \) in bytes.
  where hash-name is given in ASN.1 syntax (ugly!)
- 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]

\[ r \xleftarrow{\$} 50, 15 k_0 \]
\[ w \leftarrow h(M \parallel r) \quad |w| = k_1 \]
\[ r^* \leftarrow g_s(w) \oplus r \quad |r^*| = k_0 \]
\[ y \leftarrow 0 \parallel w \parallel r^* \parallel g_s(w) \quad |y| = |n| \]

output \( \sigma(M) = y^d (\text{mod} \ n) \)

Verify \((M, \sigma)\):
\[ y \leftarrow \sigma^e (\text{mod} \ n) \]
Parse \( y \) as \( b \parallel w \parallel r^* \parallel y \)
\[ r^* \leftarrow r^* \oplus g_s(w) \]
return True iff \( b = 0 \) \& \( h(M \parallel r) = w \) \& \( g_s(w) = y \)
• We can model $h$, $g_1$, and $g_2$ as random oracles.

**Theorems:**

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: $x \leftarrow \mathcal{R} \{0, 1, ..., p-2\}$  
$SK = x$ 
$y = g^x \pmod{p}$  
$PK = y$

Sign $(M)$:

- $m = \text{hash} (M)$  
  - [hash fn into $\mathbb{Z}_{p-1}$]
- $k \leftarrow \mathcal{R} \mathbb{Z}_{p-1}^*$  
  - [gcd$(k,p-1)=1$]
- $r = g^k$  
  - [hard work is indep of $M$]
- $s = \frac{(m-rx)}{k} \pmod{p-1}$
- $\sigma (M) = (r, s)$

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

- Check that $0 < r < p$  
  - (else reject)
- Check that $y^r s = g^m \pmod{p}$
  - where $m = \text{hash} (M)$
Correctness of El Gamal signatures:

\[ y^r s = g^{rx} g^{sk} = g^{rx+sk} = g^m \quad (\text{mod} \ p) \]

\[ \equiv \]

\[ rx + ks \equiv m \quad (\text{mod} \ p-1) \]

or

\[ s \equiv \frac{(m-rx)}{k} \quad (\text{mod} \ p-1) \]

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

Proofs

Let \( e \leftarrow \mathbb{Z}_{p-1} \)
\[
\begin{align*}
    r &\leftarrow g^e \cdot y \pmod{p} \\
    s &\leftarrow -r \pmod{p-1}
\end{align*}
\]

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

Check:
\[
\begin{align*}
    y^r r^s &= g^m \\
    g^{xr} (g^e y)^r &= g^{er} = g^e s = g^m \quad \square
\end{align*}
\]

But: It is easy to fix.

Modified ElGamal (Pointcheval & Stern 1996)

Sign \( (M) \):
\[
\begin{align*}
    k &\leftarrow \mathbb{Z}_p^* \\
    r &= g^k \pmod{p} \\
    m &= h (M \| r) \quad \Leftarrow *** \\
    s &= (m - rx) / k \pmod{p-1} \\
\end{align*}
\]

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

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

Theorem: Modified ElGamal 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, \( |q| = 160 \) bits
- \( p = nq + 1 \) prime, \( |p| = 1024 \) 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
- \( y = g^x \pmod{p} \) PK

Sign \((m)\):

- \( k \leftarrow \mathbb{Z}_q^* \) (i.e. \( 1 \leq k < q \))
- \( r = (g^k \pmod{p}) \pmod{q} \)
- \( |r| = 160 \) bits
- \( m = h(M) \)
- \( s = (m + rx) \pmod{q} \)
- \( |s| = 160 \) 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 \( \mathbb{Z}_q \) for large enough \( q \).
Verify:
Check $0 < r < q$ & $0 < s < q$
Check $y^{r/s} g^{m/s} \equiv r \pmod{p} \pmod{q}$
where $m = h(M)$

Correctness:
\[
g^{(rx+m)/s} \equiv r \pmod{p} \pmod{q}
\]
\[
\equiv g^k = r \pmod{p} \pmod{q} \checkmark
\]

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.