Admin:

Outline:

- RSA-OAEP
- Digital Signatures
  - def
  - secure def
- RSA-PSS
- El Gamal
- DSS 3 ← didn't do
How to make RSA IND-CCA2 secure?

"OAEP" = Optimal asymmetric encryption padding [BR94]

Given $m$, $|m| = t$ bits

Pick $r$ at random, $|r| = k_0$

On decryption: invert RSA, invert OAEP, reject if $O^{k_1}$ not present

Thm: RSA with OAEP secure against ACCA, assuming RO model & that RSA hard to invert on random inputs.

OAEP: used in practice

theory: (we don't have random oracles...)

$G: 50,13 \rightarrow 50,13^{k_0}^{t+k_1}$

$H: 50,13 \rightarrow 50,13^{k_0}$

$G, H$: "random oracles" like UFE(!) of Desai
Digital Signatures

- Invented by Diffie/Hellman in 1976 (New Directions)
- First implementation: RSA (1977) [key motivation for me for PK... !]
- Initial idea: switch PK/SK - enc with secret key = sig
  - if PK decrypts it - then sig OK

- Current way of describing digital signatures

  (Note: law is confused - includes hashes, MACs, etc...) - ignore it

- $\text{Keygen}(1^\lambda) \rightarrow (PK, SK)$
- $\text{Verifation key}$
  $\text{Signing key}$

  $\lambda =$ "secure parameter"
  all terms are polynomial
  security may be negligible fun of $\lambda$.

- Ignore for now - "PKI" issue:
  knowing that you have "right" PK

- $\text{Sign} (SK, M) \rightarrow \sigma_{sk} (M)$
  $M \in \{0, 1\}^*$

  (must be randomized)

- $\text{Verify} (PK, M, \sigma) = \text{True/False}$

  $\text{Correctness: } (\forall M) \text{ Verify} (PK, M, \text{Sign}(SK, M)) = \text{True}$
Security: (Weak) existential unforgeability under adaptive chosen message attack:

**Game:**

Challenger

(\(PK, sk\)) \(\leftarrow\) KeyGen(1^λ)

\[\begin{align*}
\text{PK} & \rightarrow \ni_1 \\
\sigma(\ni_1) & \leftarrow \ni_2 \\
\sigma(\ni_2) & \leftarrow \vdots \\
\sigma(\ni_κ) & \leftarrow \ni_κ \\
\text{Adv wins if } & \text{ Verify } (PK, \ni_κ, \sigma_κ) = \text{True} \\
& \land \ni_κ \not\in \mathcal{E}(\ni_1, \ldots, \ni_κ) \end{align*}\]

Scheme is secure (i.e. weakly existentially unforgeable against adaptive chosen message attack)

\[\Pr[\text{Adv wins}] \text{ is negligible (i.e. } \leq \frac{1}{2^κ} \text{ for } \frac{1}{2^κ} \text{ c & all sub. } k \geq \lambda\)]

Scheme is strongly secure if adversary can't even produce new sig for previous message, previously signed

i.e. Adv wins if \(\text{Verify } (PK, \ni, \sigma_κ) = \text{True}\)

\[\land (\ni, \sigma_κ) \not\in \mathcal{E}(\ni_1, \sigma_1), (\ni_2, \sigma_2), \ldots, (\ni_κ, \sigma_κ)\]
Signing with RSA

1. Hash & sign with PKCS
   Let $H(M) = SHA_{256}(M)$
   Let $H'(M) = 0x\ 00\ 01\ FF\ FF\ ...\ FF\ 00||H(M)$
   $\sigma(M) = (h'(M))^d \mod n$
   Some problems with $e=3$ (bad implementations can find $0$
   parse ASN.1
   take $H(M)$
   miss other shift after $H(M)$)
   Otherwise seems OK, but no proofs. (even
   assuming collision resistance & RSA hard to invert...)

Commonly used, none the less...

2. PSS [Bellare & Rogaway 1996]
\[
\text{Sign}(m) : \begin{cases} 
\begin{gathered} 
\begin{align*} 
& r \xleftarrow{\text{R}} \mathbb{Z}_q \times \mathbb{Z}_r^* \times \mathbb{Z}_q^n \times \mathbb{Z}_r \\
& w \leftarrow h(M || r) \\
& r^* \leftarrow g_1(w) \oplus r \\
& y \leftarrow 0 \| w \| r^* \| g_2(w) \\
\end{align*} \\
\end{gathered} \\
\text{return } y^d \pmod{n} \\
\end{cases}
\]

\[
\text{Verify}(M, x) : \begin{cases} 
\begin{gathered} 
\begin{align*} 
& y \leftarrow x^e \pmod{n} \\
& \text{parse } y \text{ as } b \| w \| r^* \| g_2(w) \\
& r \leftarrow r^* \oplus g_1(w) \\
& \text{if } h(M || r) = w \& g_2(w) = 0 \& b = 0 \\
& \quad \text{return True} \\
& \quad \text{else return False} \\
\end{align*} \\
\end{gathered} \\
\end{cases}
\]

**Theorem:** PSS is (weakly) existentially unforgeable against chosen message attack in ROM if RSA is not invertible on random inputs. (\(
\text{HEAdv who can produce } x^d \text{ given } x. 
\))
El Gamal Signatures

Public system parameters

\[ p \text{ prime} \]
\[ g \text{ generator} \]

Keygen:
\[ x \in \mathbb{Z}_p \]
\[ y = g^x \]
\[ SK = x \]
\[ PK = y \]

Sign(m):
\[ m = h(M) \]
\[ k \in \mathbb{Z}_{p-1}^* \]
\[ r = g^k \]
\[ [\text{gcd}(k, p-1) = 1] \]
\[ \text{randomized signing} \]
\[ \text{hard work is independent of } M \]
\[ ks + rx = m \]
\[ s = \frac{(m - rx)}{k} \pmod{(p-1)} \]

\[ \sigma^*(M) = (r,s) \]

Verify:
\[ \text{check } 0 < r < p \]
\[ y^r s^c = g^m \pmod{p} \text{ where } m = h(M) \]
Return True if both checks pass else return False.

Correctness:
\[ g^{rx} g^{sk} = g^{rx + sk} \equiv g^m \pmod{p} \]
\[ \equiv \]
\[ r x + ks = m \pmod{p-1} \]
\[ \equiv \]
\[ s = \frac{(m - rx)}{k} \pmod{p-1} \]
\[ (\text{if gcd}(k, p-1) = 1) \]
That was original version.

**Theorem:** El Gamel is existentially forgeable (without h fn or \( h = \text{idem}_h \)).

**Proof:** Let \( \epsilon \in_R \mathbb{Z}_{p-1} \)

\[
    r \leftarrow g^e \pmod{p},
    s \leftarrow -r \pmod{p-1},
\]

\((r, s)\) is sig for message \( m = es \pmod{p-1} \).

\[
y^{r r^s} = g^m
\]

\[
g^{x^r (e^y)^{-r}} = g^{-er} = g^es = g^m \quad \text{for} \quad m = es \pmod{p-1}.
\]

**But:** It is easy to fix.

**Modified El Gamel (Pointcheval/Stern 1996)**

**Sign(M):** \( \epsilon \in_R \mathbb{Z}_p^* \)

\[
    r = g^k \pmod{p},
    m = h(M \| r),
    s = \frac{(m - rx)}{k} \pmod{p-1}
\]

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

**Verify:** check \( 0 < r < p \)

check \( y^{r s} = g^m \) where \( m = h(M \| r) \).
Thm: (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:
- $q$ prime, $|q| = 160$ bits
- $p = nq + 1$ prime, $|p| = 1024$ bits
- $g_0$ generates $\mathbb{Z}_p^*$
- $g = g_0^n$ generates $G_q$ - subgroup of $\mathbb{Z}_p^*$ of order $q$

Keygen:
- $x \in \mathbb{Z}_q^*$
- $y = g^x \pmod{p}$
- $SK = [x]$ 160 bits
- $PK = [y]$ 1024 bits

Sign($M$):
- $k \in \mathbb{Z}_q^*$ (i.e. $1 \leq k < q$)
- $r = (g^k \pmod{p})(\pmod{q})$ 160 bits
- $m = h(M)$
- $s = (m + rx)/k \pmod{q}$ 160 bits
- redo if $r = 0$ or $s = 0$
- $\sigma(M) = (r, s)$ 320 bits

Verify (PK, M, (r, s))
- Check $y^{r/s} m/s \pmod{p} \pmod{q} \stackrel{?}{=} r$
  where $m = h(M)$
- Correctness:
  $g^{(rx + m)/s} \equiv r \pmod{p} \pmod{q}$
  $g^k \equiv r \pmod{p} \pmod{q}$

Security proof works if we had done $m = h(M || r)$, as before.
As it stands, existentially forgeable for $h =$ identity.