Today:
- CPA & CCA2 Secure public key Encryption
- RSA Encryption scheme
- RSA OAEP CCA2 secure scheme
  \Optimal Asymmetric Encryption Padding
- Digital signatures
  - Def
  - RSA
  - Hash & Sign

Semantic security
\[ \text{Chosen Plaintext Attacks} \leq \text{Chosen Ciphertext Attacks} \leq \text{CPA & CCA2 security} \]

\underline{Def:} Every eff. adv. wins in the following game w.p. at most negl.

\underline{Phase I ("Find") :}
- Adv is given PK generated by \((PK, SK) \leftarrow \text{Key Gen}(1^n)\)
- Adv is given oracle to \(\text{Dec}(sk, \cdot)\) \(\text{(Only in CCA2)}\)
- Adv generates two msgs \(m_0, m_1 \in \mathcal{M}\) and "state" \(\theta\)
Phase II ("Guess"):

- \text{Adv} is given \( c \leftarrow \text{Enc}(PK, M_b) \) for random \( \hat{b} = \hat{0} \hat{1} \).

- \text{Adv} is given oracle access to \( \text{Dec}(sk, \cdot) \) everywhere except \( c \) (only in CCA2).

- \text{Adv. outputs} \( \hat{b} \) (a guess for \( b \)).

\text{Adv. wins iff} \( \hat{b} = b \).

\textit{Note}: A CPA (or CCA2) secure scheme must be randomized.

\textbf{Thm.} El-Gamal is CPA secure iff DDH holds in [\text{Enc}(g^x, m) = g^y, g^{xy}, m]

(We saw last time).

- El-Gamal is not CCA2 secure

\text{Moreover,} \text{Enc}(g^x, m) = g^y, g^{xy} \cdot m \rightarrow \text{can easily generate} \text{Enc}(2m) : g^y, g^{xy} \cdot 2m

Moreover, El-Gamal is homomorphic:

\text{Enc}(g^x, m_1), \text{Enc}(g^x, m_2) \rightarrow \text{Enc}(g^x, m_1 \cdot m_2)
El-Gamal is re-randomizable

\[
\text{Enc}(g^x, m) = (g^y, g^{xy} \cdot m) \Rightarrow (g^{y+z}, g^{x(y+z)} \cdot m)
\]

(1998)

Cramer-Shoup extended El-Gamal to be CCA2 secure.

Idea: Added to the ciphertext a "test."
Decryption checks that the test passes.
If not outputs 1.
Decrypts only if test passes.

Idea: To pass the test one needs to "know" the msg
(and hence decryption oracle is useless).

RSA Encryption

[Rivest-Shamir-Adleman77]

First public key encryption scheme.

Follows the Diffie-Hellman model:

1. \text{Key Gen}(1^n): (PK, SK, M, C)
2. \text{msg space} \rightarrow \text{ciphertext space}
3. |M| = |C|
- $\text{Enc}(\text{PK}, \cdot)$ is an efficiently computable deterministic function from $\mathcal{M}$ to $\mathcal{C}$.

- $\text{Dec}(\text{sk}, \cdot)$ is an efficiently computable inverse:

  $$\text{Dec}(\text{sk}, \text{Enc}(\text{PK}, m)) = m \quad \forall m \in \mathcal{M}.$$  

  Deterministic encryption!

  $\Rightarrow$ Not semantic secure

  SK is "trapdoor" information that enables inversion of the (aw. one-way) function $\text{Enc}(\text{PK}, \cdot)$.

  RSA $\text{Enc}$: Trapdoor one-way function family:

  $\text{KeyGen}$: Sample two large primes $p, q$

  (eg. $\lambda = 1024$ bits each).

  $n = p \cdot q$

  - Sample $e \in \mathbb{Z}_n^*$ and compute $d = e^{-1} \mod \lambda(n)$

  Recall $\lambda(n) = |\mathbb{Z}_n^*| = (p-1) \cdot (q-1)$

  $e^{-1}$ can be computed given $\lambda(n)$ using Extended Euclid's alg.
\[ PK = (n, e) \]
\[ SK = (n, d) \]

\[ H = C = \mathbb{Z}_n^\times \]

\[ \text{Encrypt: } \quad Enc((n, e), m) = m^e \mod n \]

\[ \text{Decrypt: } \quad Dec((n, d), c) = c^d \mod n \]

Correctness: If \( M = C = \mathbb{Z}_n^\times \) then

\[ \forall m \in \mathbb{Z}_n^\times \]

\[ \text{Dec}(sk, Enc(pk, m)) = \]
\[ = \text{Dec}(sk, m^e \mod n) = \]
\[ = m^{ed} \mod n = m \]

Correctness also holds in \( \mathbb{Z}_n \) via the Chinese Remainder Thm. (though we will not see msgs in \( \mathbb{Z}_n \setminus \mathbb{Z}_n^\times \)),

which implies that it suffices to prove that \( \forall m \in \mathbb{Z}_n \)

\[ m^{ed} = m \mod p \] \& \[ m^{ed} = m \mod q \].
Recall CRT:

\[
\text{For } n=p \cdot g \text{ where } p \text{ and } g \text{ are distinct primes, } \forall x, y \in \mathbb{Z}_n,
\]

\[x = y \mod n \iff (x = y \mod p \text{ and } x = y \mod g)\]

Security: Assumes it is hard to factor.

If one can factor one can compute \(d = e^{-1} \mod \varphi(n)\).

Key insight: The size of the group \(\mathbb{Z}_n^*\) is unknown.

Knowing \(\varphi(n) = |\mathbb{Z}_n^*| \equiv \) knowing \(p\) and \(g\).

How hard is factoring? Can be done in time \(2 (\log n)^3 \cdot (\log \log n)^{4/3}\).

- In 2009 RSA keys of length 768 were factored.
- Can expect 1024 bit keys to be factored in near future.
- RSA keys of length 2048 are believed to be secure for a long time, unless there will be an alg\breakthrough, or quantum computers.
- Factoring can be done \textit{eff} on a quantum computer.
RSA is not semantic secure.

It is not even randomized.

RSA is a trapdoor permutation

\[ f_{n,e} : \mathbb{Z}_n \to \mathbb{Z}_n \]

\[ a \to a^e \mod n \]

\[ \text{Easy to compute.} \]
\[ \text{Believed to be hard to invert} \]

(RSA assumption)

\[ \text{Easy to invert given a trapdoor d.} \]

Making RSA CCA2 Secure

\[ \text{OAEP} = \text{"Optimal Asymmetric Encryption Padding."} \]
\[ \text{[Bellare - Rogaway '94]} \]

Idea: Apply the RSA encryption to an encoding of the msg.

\[ X_0 = G(r) \oplus (m \circ O^{K_1}) \]
\[ X_1 = r \oplus H(X_0) \]

G, H Random Oracles

\[ G : \{0,1\}^k \to \{0,1\}^{t+4} \]
\[ H : \{0,1\}^t \to \{0,1\}^{t_0} \]

\[ \text{Similar to UFE of Desai for symmetric enc} \]
Note:

OAES is randomized. \( \forall m \rightarrow \text{Enc}(m) = (x_0, x_1) \) is random.

Moreover one cannot learn any bit of information about \( m \) w.o. revealing the encoding \( (x_0, x_1) \) entirely. [in ROM]

Any trapdoor permutation w. OAES encoding is CPA secure.

RSA w. OAES is CCA2 secure!

---

**Digital Signatures**

- Proposed 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)
Def: A digital signature scheme consists of 3 alg:
- \text{KeyGen} (1^k) \rightarrow (PK, SK)
- \text{Sign} (SK, m) \rightarrow \sigma_{sk} (m) \leftarrow \text{may be randomized}
- \text{Verify} (PK, m, \sigma) = 0/1 \quad \text{(acc/res)}.

Correctness: \forall m \in \mathcal{M}
\begin{equation}
P_r \left[ \text{Verify} (PK, m, \text{Sign}(SK, m)) = 1 \right] = 1
\end{equation}

Security: Existential Unforgeability against
- adaptive chosen msg attacks:
  (i) Adv gets oracle access to \text{Sign}(SK, \cdot) for
  \((PK, SK) \leftarrow \text{KeyGen}(1^k)\)
  Namely, adv obtains signatures for msgs of his choice
  \(m_1, \ldots, m_g, \sigma_1, \ldots, \sigma_g\) \quad g = \text{poly}(k) \quad \sigma_i \leftarrow \text{Sign}(SK, m_i)
  \(m_i\) can depend on \(m_1, \ldots, m_{i-1}, \sigma_1, \ldots, \sigma_{i-1}\).
  (ii) Adv outputs a pair \((m, \sigma^*\))
  Adv wins if \(\text{Verify}(PK, m, \sigma^*) = 1 \land m \in \{m_1, \ldots, m_g\}\)
Def: A scheme is secure (i.e. existentially unforgeable against adaptive chosen msg attacks) if
\[ P[\text{Adv wins}] = \text{negl}(\lambda) \]

Diffie & Hellman (1976) suggested a general method for using a deterministic public key encryption scheme as a signature scheme.

Idea: \[ \text{Sign} (sk, m) = \text{Dec}(sk, m) \]
\[ \text{Verify} (pk, m, \sigma) = 1 \text{ iff } \text{Enc}(pk, \sigma) = m. \]

**Signing w. RSA**

**KeyGen** (1^\circ): Choose \( n = p \cdot q \) \( p, q \) random \( >\)-bit primes.

Choose \( e \in \mathbb{Z}_{\phi(n)}^{\ast} \), \( d = e^{-1} \mod \phi(n) \)

\( PK = (n, e) \)
\( SK = (n, d) \)

\[ \text{Sign} (sk, m) = m^d \mod n \]
\[ \text{Verify} (pk, m, \sigma) = 1 \text{ iff } \sigma^e = m \mod n \]
Correctness: $\forall m \in \mathbb{Z}_n \, \, (m^d)^e = m^{d \cdot e} \equiv m \mod n$

Not secure: 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 $h(m)$, where $h$ is a collision resistant hash function.

* Better efficiency: Hashing is extremely efficient compared to signing.

* Allow flexibility: Signing any msg $m \in \{0,1\}^*$.

Claim: If $(\text{KeyGen}, \text{Sign}, \text{Verify})$ is secure & $H = \{h_k\}$ is a collision resistant hash family then the hash & sign version of $(\text{KeyGen}, \text{Sign}, \text{Verify})$ is also secure.
Interestingly, Hash & Sign paradigm is also useful for security.

Hash & Sign with RSA

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

\[
\text{Verify} \left( (n,e,h), m, \sigma \right) = 1 \iff 
\sigma^e = h(m) \mod n.
\]

Is this secure? Depends on $h$.

It is secure in the Random Oracle Model (if $h$ is RO) \([\text{Bellare-Rogaway 93}]\)

aka Full Domain Hash (FDH)