<u>Today</u>:

- 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
≡
Chosen Plaintext Attacks
Chosen Ciphertext Attacks.

CPA & CCA2 security

<u>Def</u>: Every eff. adv. wins in the following game w.p. at most negl.

<u>Phase I</u> ("Find"):

- Adv is given PK generated by $(PK, SK) \leftarrow \text{KeyGen}(1^\lambda)$
- Adv is given oracle to $\text{Dec}(SK, \cdot)$ (Only in CCA2)
- Adv generates two msgs $m_0, m_1 \in \mathcal{M}$ and "state" Q
  msg space

Phase II ("Guess"):

- Adv is given $c \leftarrow Enc(PK, m_b)$ for random $b \leftarrow \{0,1\}$
- Adv is given oracle access to $Dec(sk, \cdot)$ everywhere except c (only in CCA2).
- Adv. outputs b' (a guess for b)

Adv. wins iff $b' = b$.

Note: A CPA (or CCA2) secure scheme must be randomized.

Thm: El-Gamal is CPA secure iff DDH holds in $G = \langle g \rangle$.

$\left[Enc(g^x, m) = g^y, g^{xy} \cdot m \right]$

(We saw last time).

- El-Gamal is not CCA2 secure

$\underbrace{Enc(g^x, m)}_{PK} = g^y, g^{xy} \cdot m \longrightarrow$ can easily generate $Enc(2m): g^y, g^{xy} \cdot 2m$

Moreover, El-Gamal is homomorphic:

$Enc(g^x, m_1), Enc(g^x, m_2) \longrightarrow Enc(g^x, m_1 \cdot m_2)$

- El-Gamal is rerandomizable

$$Enc(g^x, m) = (g^y, g^{xy} \cdot m) \Rightarrow (g^y \cdot g^z, g^{xy} \cdot m \cdot g^{xz})$$

$$= g^{y+z}, g^{x(y+z)} \cdot m$$

* Cramer-Shoup (1998) extended El-Gamal to be CCA2 secure.

Idea: Added to the ciphertext a "test".

Decryption checks that the test passes

If not outputs $\perp$.

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-Adleman 77]

First public key encryption scheme.

Follows the Diffie-Hellman model:

- KeyGen (1^λ): (PK, SK, M, C)

msg space (M)

ciphertext space (C)

$$|M| = |C|$$

- $Enc(PK, \cdot)$ is an efficiently computable deterministic function from $\mathcal{M}$ to $\mathcal{C}$.
- $Dec(SK, \cdot)$ is an efficiently computable inverse:

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

Deterministic encryption!

$\Rightarrow$ Not semantic secure

SK is "trapdoor" information that enables inversion of the (a.w. one-way) function $Enc(PK, \cdot)$.

RSA Enc : Trapdoor one-way function family:
permutation

KeyGen : • Sample two large primes p, q

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

$n = p \cdot q$

• Sample $e \leftarrow \mathbb{Z}^*_{\varphi(n)}$ and compute $d = e^{-1} \bmod \varphi(n)$

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

e^{-1} can be computed given $\varphi(n)$ using Extended Euclid's alg.

$PK = (n, e)$

$SK = (n, d)$

$$\mathcal{M} = \mathcal{C} = \mathbb{Z}_n$$

Encypt: $Enc((n,e), m) = m^e \bmod n$

Decrypt: $Dec((n,d), c) = c^d \bmod n$

Correctness: If $\mathcal{M} = \mathcal{C} = \mathbb{Z}_n^*$ then

$\forall m \in \mathbb{Z}_n^*$

$$Dec(sk, Enc(pk, m)) =$$

$$= Dec(sk, m^e \bmod n) =$$

$$= m^{ed} \bmod n = m^{1 + c \cdot \varphi(n)} \bmod n = m$$

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

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

$$m^{ed} = m \bmod p \ \ \& \ \ m^{ed} = m \bmod q.$$

<u>Recall CRT:</u>
For $n = p \cdot q$ where p & q are distinct primes, $\forall x, y \in \mathbb{Z}_n$
$x \equiv y \mod n$ iff $(x \equiv y \mod p$ & $x \equiv y \mod q)$

<u>Security</u>: Assumes it is hard to factor.
If one can factor one can compute $d = e^{-1} \mod \varphi(n)$.

<u>Key insight</u>: The size of the group $\mathbb{Z}_n^*$ is <u>unknown</u>

Knowing $\varphi(n) = |\mathbb{Z}_n^*|$ $\equiv$ knowing p & q.

<u>How hard is Factoring</u>: Can be done in time
$$2^{O(\log n)^3 \cdot (\log \log n)^{2/3}}$$

- 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 eff on a quantum computer

RSA is <u>not</u> semantic secure.

It is not even randomized.

RSA is a <u>trapdoor permutation</u>

$$f_{n,e} : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n$$

$$a \longmapsto a^e \bmod n$$

OW [Easy to compute.
Believed to be hard to invert $\quad a^e \xrightarrow[\text{}]{\text{HARD}} a \bmod n$ (given n, e) for random $a \leftarrow \mathbb{Z}_n$

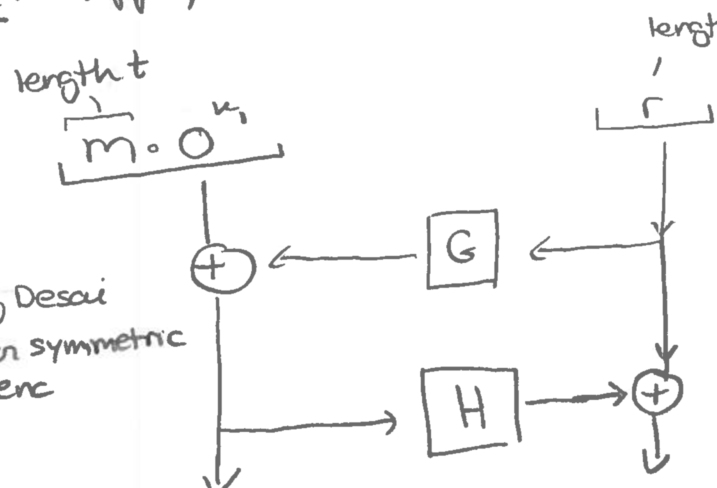
(RSA assumption)]

- Easy to invert given a trapdoor d.

Making RSA CCA2 secure

OAEP = "Optimal Asymmetric Encryption Padding" [Bellare-Rogaway 94]

<u>Idea:</u> Apply the RSA encryption to an <u>encoding</u> of the msg.



* Similar to UFE of Desai for symmetric enc

Unbalanced Feistel Enc.

$$X_0 = G(r) \oplus (m \circ 0^{k_1}) \qquad X_1 = r \oplus H(X_0)$$

$G : \{0,1\}^{k_0} \longrightarrow \{0,1\}^{t+k_1}$

$H : \{0,1\}^{t+k_1} \longrightarrow \{0,1\}^{k_0}$

G, H <u>Random Oracles</u>

$$Enc_{n,e}(X_0, X_1)$$

Note :

OAES is randomized. $\forall m \rightarrow Enc(m) = (x_0, x_1)$ 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's:

- KeyGen$(1^\lambda) \longrightarrow (PK, SK)$
- Sign$(SK, m) \longrightarrow \sigma_{SK}(m)$ ← may be randomized
- Verify$(PK, m, \sigma) = 0/1$ (acc/rej).

Correctness: $\forall m \in \mathcal{M}$ ← msg space, for $(PK, SK) \leftarrow$ KeyGen(1^λ)

$$\Pr[\text{Verify}(PK, m, \text{Sign}(SK, m)) = 1] = 1$$

Security: Existential Unforgeability against adaptive chosen msg attacks: ← similar to secure MACs.

(i) Adv gets oracle access to Sign$(SK, \cdot)$ for $(PK, SK) \leftarrow$ KeyGen(1^λ)

Namely, adv obtains signatures for msgs of his choice

$m_1, \ldots, m_q, \sigma_1 \ldots \sigma_q$ $\quad q = poly(\lambda)$ $\quad \sigma_i \leftarrow$ 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, σ^*).

Adv wins if Verify$(PK, m, \sigma^*) = 1$ & $m \notin \{m_1 \ldots m_q\}$

Def: A scheme is <u>secure</u> (i.e. existentially unforgeable against adaptive chosen msg attacks) if

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

Diffie & Hellman (1976) suggested a general method for using a <u>deterministic</u> public key encryption scheme as a signature scheme

Idea: $\text{Sign}(SK, m) = \text{Dec}(SK, m)$

$\text{Verify}(PK, m, \sigma) = 1$ iff $\text{Enc}(PK, \sigma) = m$.

Signing w. RSA

$\text{KeyGen}(1^\lambda)$: Choose $n = p \cdot q$ p, q random λ-bit primes.

Choose $e \leftarrow \mathbb{Z}^*_{\varphi(n)}$, $d = e^{-1} \bmod \varphi(n)$

$PK = (n, e)$

$SK = (n, d)$

$\text{Sign}(SK, m) = m^d \bmod n$

$\text{Verify}(PK, m, \sigma) = 1$ iff $\sigma^e = m \bmod n$

<u>Correctness</u>: $\forall m \in \mathbb{Z}_n \quad (m^d)^e = m^{d \cdot e} = m \mod n$ ✓

<u>Not secure</u>: Given $\text{Sign}(sk, m) = m^d \mod n$

one can easily sign $m^2 \mod n \to (m^d)^2 \mod n$.

To make RSA secure use hash & sign:

<u>Hash & Sign</u>

Rather than signing m, sign <u>$h(m)$</u>,
wher h is a collision resistant hash function.

* Better efficiency: Hashing is extremely eff compared to signing.
* Allow flexibility: signing any msg in $\{0,1\}^*$.

<u>Claim</u>: If (KeyGen, Sign, Verify) is secure & $H = \{h_k\}$ is a collision resistant hash family then the hash & Sign version of (KeyGen, Sign, Verify) is also secure.

Interestingly: Hash & Sign paradigm is also useful for security.

Hash & Sign with RSA

$$\text{Sign}((n,d,h), m) = h(m)^d \bmod n$$

$$\text{Verify}((n,e,h), m, \sigma) = 1 \text{ iff}$$

$$\sigma^e = h(m) \bmod n.$$

Is this secure? Depends on h...

It is secure in the Random Oracle Model (if h is RO) [Bellare-Rogaway 93]

a.k.a. Full Domain Hash (FDH)