

Admin:

Project proposals (see feedback on gradescope)

Quiz is 4/13 (two weeks) in class!

Today:

☐ El Gamal is malleable

☐ IND-CCA2 Security (& Cramer-Shoup)

☐ RSA

☐ Making RSA IND-CCA2 secure

☐ Other aspects of RSA security

Readings:

Katz: Ch 7, 10

Paar: Ch 6, 7, 8

Theorem (Tsiounis & Yung):

El Gamal is semantically secure in G



DDH holds in G

- Semantic security may not be enough for some applications.

- El Gamal is malleable:

$$\text{Given } E(m) = (g^k, m \cdot y^k)$$

it is easy to produce $E(2m) = (g^k, (2 \cdot m) \cdot y^k)$

without knowing m !

- More generally, El Gamal is homomorphic:

$$\text{Given } c_1 \in E(m_1) = (g^r, m_1 \cdot y^r)$$

$$\& \text{ given } c_2 \in E(m_2) = (g^s, m_2 \cdot y^s)$$

$$\text{can produce } c_1 \cdot c_2 = (g^{r+s}, (m_1 \cdot m_2) \cdot y^{r+s}) \\ \in E(m_1 \cdot m_2)$$

- Product of ciphertexts yields an encryption of product of plaintexts.

- Special case: multiplying by $E(1) = (g^s, y^s)$
re-randomizes encryption.

- What is stronger notion of security for PK encryption?
(e.g. one that excludes malleability...)
- "IND-CCA2 secure" (ACCA secure = secure under adaptive chosen ciphertext attack)
 $\approx$ IND-CCA secure defn we saw for symmetric enc.
- Similar to semantic security defn, except that Adv allowed access to decryption oracle, too.
(He has PK so access to encryption oracle already there.)
(As before, may not use oracle to decrypt challenge ciphertext during "guess" phase.)

IND-CCA2 (ACCA) security game:

Phase I ("Find"):

new $\Rightarrow$

- Examiner generates (PK, SK) using $\text{Keygen}(1^\lambda)$
- Examiner sends PK to Adversary
- Adversary computes for polynomial (in λ) time, having access to a decryption oracle $D(SK, \cdot)$ then outputs two messages m_0, m_1 of same length, and "state information" s . [$m_0 \neq m_1$ required]

Phase II ("Guess"):

new $\Rightarrow$ {

- Examiner picks $b \xleftarrow{R} \{0,1\}$, computes $c_b = E(PK, m_b)$
- Examiner sends c_b, s to Adversary
- Adversary computes for polynomial (in λ) time, having access to a decryption oracle $D(SK, \cdot)$ except on input c_b then outputs $\hat{b}$ (his "guess" for b).

Adversary wins if $\hat{b} = b$.

Def: PK encryption method is IND-CCA2 secure (ACCA-secure) if

$$\text{Prob}[\text{Adv wins}] \leq \frac{1}{2} + \text{negligible}$$

How to make El Gamal IND-CCA2 secure?

- Cramer-Shoup method is such an extension of El Gamal.
- Let G_g be a group of prime order g
(e.g. $G_g = \mathbb{Q}_p$, where $p=2g+1$, $p \& g$ prime).
- Keygen:

$$g_1, g_2 \xleftarrow{R} G_g$$

$$x_1, x_2, y_1, y_2, z \xleftarrow{R} \mathbb{Z}_g$$

$$c = g_1^{x_1} g_2^{x_2}$$

$$d = g_1^{y_1} g_2^{y_2}$$

$$h = g_1^z$$

EG

$$PK = (g_1, g_2, c, d, h)$$

$$H = \text{hash fn mapping } G_g^3 \text{ to } \mathbb{Z}_g$$

$$SK = (x_1, x_2, y_1, y_2, z)$$

• Enc(m) [where $m \in G_q$]:

$$r \xleftarrow{R} \mathbb{Z}_q$$

EG

$$u_1 = g_1^r$$

EG

$$u_2 = g_2^r$$

$$e = h^r \circ m$$

EG

$$\alpha = H(u_1, u_2, e)$$

$$v = c^r d^{r\alpha}$$

$$\text{ciphertext} = (\underline{u_1}, \underline{u_2}, \underline{e}, v)$$

EG

• Decrypt(u_1, u_2, e, v):

$$\alpha = H(u_1, u_2, e)$$

$$\text{Check: } u_1^{x_1 + y_1 \alpha} u_2^{x_2 + y_2 \alpha} \stackrel{?}{=} v$$

If not equal, reject

$$\text{else output } m = e / u_1^z$$

EG

$$\text{Note: } u_1^{x_1} u_2^{x_2} = g_1^{rx_1} g_2^{rx_2} = c^r$$

$$u_1^{y_1} u_2^{y_2} = d^r$$

$$u_1^z = g_1^{rz} = h^r$$

EG

Theorem: Cramer-Shoup is IND-CCA2
secure (i.e. secure against adaptive chosen
ciphertexts) if

- ① DDH holds in G_g
- ② H satisfies a certain condition
($\approx$ "target collision resistance")

Thus, our strongest notion of security for PK
encryption is in fact achievable, albeit at
some cost in terms of speed & complexity.

Diffie-Hellman model of PK encryption: (1976)

- $\text{Keygen}(1^\lambda) \rightarrow (\text{PK}, \text{SK}, \text{M}, \text{C})$
(public key, secret key, message space, ciphertext space)
Here $|\text{M}| = |\text{C}|$.
- $\text{E}(\text{PK}, \cdot)$ is an efficiently computable one-to-one (deterministic) map from M to C
 $\text{C} = \text{E}(\text{PK}, m)$ is (unique) ciphertext for m
- $\text{D}(\text{SK}, \cdot)$ is efficiently computable inverse:
 $\text{D}(\text{SK}, c) = \text{D}(\text{SK}, \text{E}(\text{PK}, m)) = m \quad (\forall m \in \text{M})$
- It is hard/infeasible to decrypt with knowledge of PK but without knowledge of SK .
 SK represents "trapdoor" information that enables inversion of the (otherwise one-way) function $\text{E}(\text{PK}, \cdot)$.

RSA PK encryption (Rivest, Shamir, Adleman 1977)Keygen:

Find two large primes p, q (e.g. $\lambda = 1024$ bits each)

$$n = p \cdot q$$

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

$$e \xleftarrow{R} \mathbb{Z}_{\varphi(n)}^* \quad [\text{i.e. } \gcd(e, \varphi(n)) = 1]$$

$$d = e^{-1} \pmod{\varphi(n)} \quad [\text{e.g. Euclid's extended alg}]$$

$$PK = (n, e)$$

$$SK = (d, p, q)$$

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

Encrypt:

Given $m \in \mathbb{Z}_n$ and $PK = (n, e)$:

$$c = E(PK, m) = m^e \pmod{n}$$

Decryption:

Given $c \in \mathbb{Z}_n$ and $SK = (d, p, q)$:

$$m = D(SK, c) = c^d \pmod{n}$$

$$(\text{where } n = p \cdot q)$$

Note:

p & q should be large randomly chosen primes, as security of RSA depends upon inability of adversary to factor n (from PK) into p, q .

Correctness of RSA:Lemma: (Chinese remainder theorem or CRT)Let $n = p \cdot q$ where p, q are distinct primes.Then $(\forall x, y \in \mathbb{Z}_n)$

$$x = y \pmod{n} \iff x = y \pmod{p} \ \& \ x = y \pmod{q}$$

Thus it suffices to prove RSA correct mod p ; the proof mod q is the same, and CRT then implies correctness mod n .

Given $e \cdot d = 1 \pmod{\varphi(n)}$ $[d = e^{-1} \pmod{\varphi(n)}]$

so $e \cdot d = 1 + t \cdot (p-1)(q-1)$ for some t

and $e \cdot d = 1 \pmod{p-1}$ $[d = e^{-1} \pmod{p-1}]$

Correctness of RSA means

$$(m^e)^d = m \pmod{n} \text{ for all } m \in \mathbb{Z}_n$$

By CRT we only need to prove

$$(m^e)^d = m \pmod{p} \text{ for all } m \in \mathbb{Z}_p$$

We consider two cases:

Case 1: $m = 0 \pmod{p}$

$$\text{Trivial: } 0^{ed} = 0 \pmod{p}$$

Case 2: $m \neq 0 \pmod{p}$

$$\text{i.e. } m \in \mathbb{Z}_p^*$$

$$\text{so } m^{p-1} = 1 \pmod{p} \quad [\text{Fermat}]$$

$$\text{Then } m^{ed} = m^{1+u \cdot (p-1)} \pmod{p}$$

$$\text{where } u = t \cdot (q-1)$$

$$m^{ed} = m \cdot (m^{p-1})^u \pmod{p}$$

$$= m \cdot 1^u$$

$$= m$$

$$\therefore m^{ed} = m \pmod{p} \text{ for all } m \in \mathbb{Z}_p$$

$$\& m^{ed} = m \pmod{q} \text{ for all } m \in \mathbb{Z}_q \text{ (similarly)}$$

$$\text{and } m^{ed} = m \pmod{n} \text{ for all } m \in \mathbb{Z}_n \text{ (by CRT)}$$

$$\text{Thus } (\forall m \in \mathbb{Z}_n) D(SK, E(PK, m)) = m \quad \square$$

Security of RSA

Factoring attacks:

If any adversary can factor n , then the adversary can compute $\varphi(n)$, and thus compute $d = e^{-1} \pmod{\varphi(n)}$. & vice versa

Key insight:
size of group $\mathbb{Z}_n^*$ is unknown and unknowable to Adversary.

How hard is factoring?

- time $\exp \{ c \cdot (\ln n)^{1/3} (\ln \ln n)^{2/3} \}$
- RSA keys of length 768 factored (2009);
can expect RSA key of length 1024 bits to be factored in the "near future".
- RSA keys of length 2048 secure for a very long time, unless there are algorithmic breakthroughs on problem of factoring.

Is (base) RSA semantically secure?

No. (It's not even randomized...)

∴ not IND-CCA2 secure either...

How to make RSA IND-CCA2 secure?

OAEP = "Optimal asymmetric encryption padding" [BR94]

{ Let message m be t bits in length.
Add k_0 bits of randomness $|r| = k_0$
Add k_1 bits of 0's 0^{k_1} (to check)

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

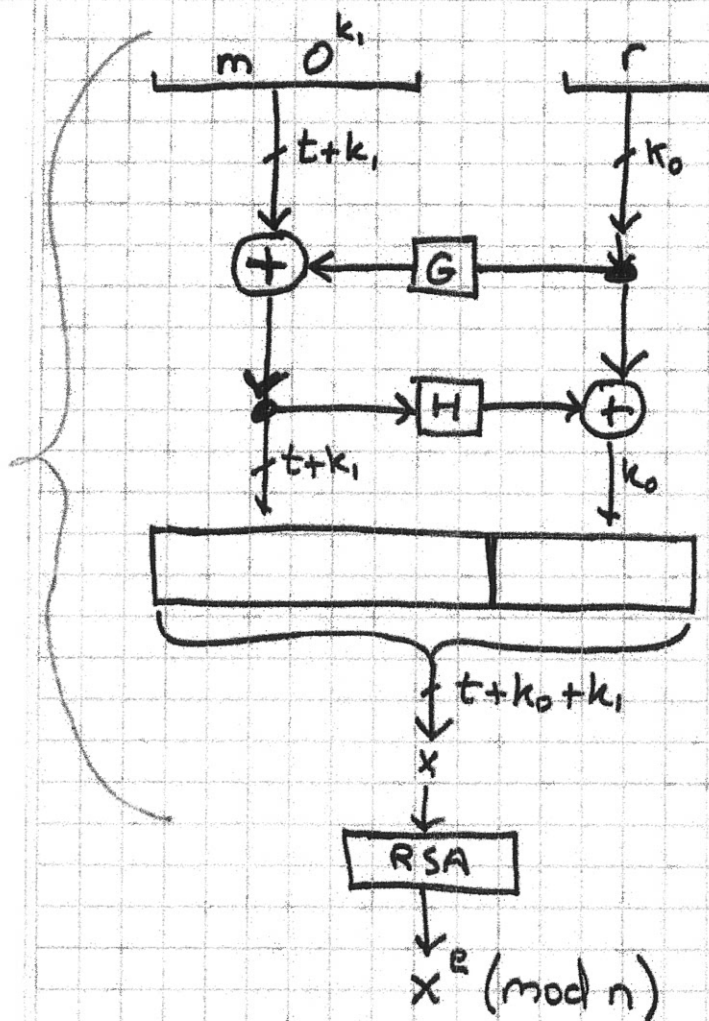
$H: \{0,1\}^{t+k_1} \rightarrow \{0,1\}^{k_0}$

G, H "random oracles"

[Compare to UFE of Desai for symmetric encryption]

OAEP Encryption

OAEP



On decryption:

invert RSA

invert OAEP

reject if 0^{k_1} not presentelse output m

Theorem: RSA with OAEP is IND-CCA2
secure, assuming ROM for G & H ,
and assuming RSA hard to invert on
random inputs.

[Bug in original proof, but OK with very
slightly modified assumptions (or OAEP⁺)]

OAEP used in practice

(But in practice we don't really have random oracles!)

Other aspects of RSA security:

[ref Boneh paper: 20 years of attacks on RSA]

Weak keys: small d is insecure

($d < n^{1/4}$ allows adversary to factor n)

Implementation issues:

- Power analysis
 - Timing attacks
 - Fault injection (introduce power supply glitch)
- (esp. if device is using CRT)

Quantum computing

Peter Shor (MIT) has shown that factoring in polynomial time is possible on a "quantum computer"