

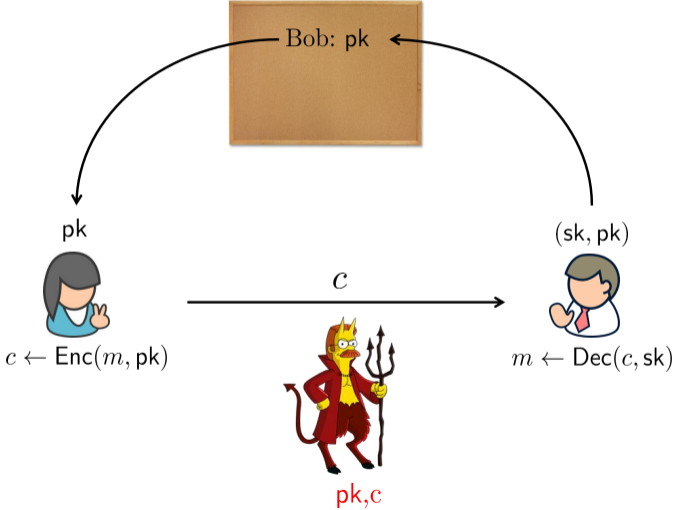
Modern Public Key Cryptography

Public Key Encryption

Daniel Kales based on slides by Sebastian Ramacher and David Derler

March 9, 2022

Public Key Encryption - The Setting



Formal Definition

Public Key Encryption

A PKE scheme is a tuple of PPT algorithms:

KeyGen(1^κ): This probabilistic algorithm takes a security parameter κ and outputs a pair of keys (sk, pk) (pk fixes plaintext space \mathcal{M} and ciphertext space \mathcal{C}).

Enc(m, pk): This (probabilistic) algorithm takes a message $m \in \mathcal{M}$ and a public key pk and outputs a ciphertext $c \leftarrow \text{Enc}(m, pk) \in \mathcal{C}$.

Dec(c, sk): This deterministic algorithm takes a ciphertext $c \in \mathcal{C}$ and a private key sk and outputs $m \leftarrow \text{Dec}(c, sk) \in \mathcal{M} \cup \{\perp\}$.

Security

Correctness

$$\forall (sk, pk) \leftarrow \text{KeyGen}(1^\kappa) : \Pr [\text{Dec}(\text{Enc}(m, pk), sk) = m] = 1 - \epsilon(\kappa)$$

How to define when a scheme is secure?

- Given c and pk it should be hard to find m ?
 - Very weak guarantees ...
- We will gradually develop the idea of security for PKE

Overview of Target and Attacks

Targets (hardest to easiest)

- **One-wayness (OW):** hard to invert
- **Semantically secure (Indistinguishable - (IND)):** no information about the message in $\text{Enc}(m, pk)$ is leaked
- **Non-malleable (NM):** for any non-trivial relation R it is hard to compute $\text{Enc}(R(m), pk)$ from $\text{Enc}(m, pk)$

Overview of Target and Attacks

Attacks (weak to strong)

- **Passive attacks:** Chosen plaintext attack (CPA)
- **Active attacks:** Chosen ciphertext attacks (CCA)

Highest security guarantees if strongest attacker can not even achieve the weakest target: NM-CCA2 (IND-CCA2)

Overview of Target and Attacks

Attacks (weak to strong)

- **Passive attacks:** Chosen plaintext attack (CPA)
- **Active attacks:** Chosen ciphertext attacks (CCA)

Highest security guarantees if strongest attacker can not even achieve the weakest target: NM-CCA2 (IND-CCA2)

Textbook RSA Encryption

Use a trapdoor one-way function for encryption (e.g., RSA, Rabin)

KeyGen(1^κ): Pick two random κ -bit primes p, q , set $N = pq$, pick e s.t. $\gcd(e, \varphi(N)) = 1$, compute $d \leftarrow e^{-1} \pmod{\varphi(N)}$ output $(\text{sk}, \text{pk}) \leftarrow ((d, N), (e, N))$

Enc(m, pk): On input $m \in \mathbb{Z}_N^*$ and $\text{pk} = (e, N)$, compute and output $c \leftarrow m^e \pmod{N}$

Dec(c, sk): On input c and $\text{sk} = (d, N)$, compute and output $m \leftarrow c^d \pmod{N}$

Security of Textbook RSA

- How hard is it to recover m given c and $pk = (e, N)$
 - This has been formalized as the **RSA problem** and is assumed to be hard
 - Assumes that c (and thus m) is a random element of \mathbb{Z}_N
 - Very strong assumption for a secure PKE
- Some of the problems of textbook RSA
 - RSA encryption is **deterministic**: Small message space $\mathcal{M}' \subseteq \mathcal{M}$, just test any $m \in \mathcal{M}'$
 - RSA encryption function is a **homomorphism**:

$$\text{Enc}(m_0, pk) \cdot \text{Enc}(m_1, pk) = m_0^e \cdot m_1^e = (m_0 \cdot m_1)^e = \text{Enc}(m_0 \cdot m_1, pk)$$

Security of Textbook RSA

- How hard is it to recover m given c and $pk = (e, N)$
 - This has been formalized as the **RSA problem** and is assumed to be hard
 - Assumes that c (and thus m) is a random element of \mathbb{Z}_N
 - Very strong assumption for a secure PKE
- Some of the problems of textbook RSA
 - RSA encryption is **deterministic**: Small message space $\mathcal{M}' \subseteq \mathcal{M}$, just test any $m \in \mathcal{M}'$
 - RSA encryption function is a **homomorphism**:

$$\text{Enc}(m_0, pk) \cdot \text{Enc}(m_1, pk) = m_0^e \cdot m_1^e = (m_0 \cdot m_1)^e = \text{Enc}(m_0 \cdot m_1, pk)$$

One-Wayness

One-Wayness

For all PPT adversaries \mathcal{A} and security parameters κ there is a negligible function ϵ such that:

$$\Pr \left[\begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\kappa), m \xleftarrow{R} \mathcal{M}, \\ m^* \leftarrow \mathcal{A}(\text{pk}, \text{Enc}(m, \text{pk})) : m^* = m \end{array} \right] \leq \epsilon(\kappa).$$

- Not meaningful for most applications of PKE (but okay for RSA-KEM)
- \mathcal{A} may still compute some information about m
- How to formalize "does not leak any information"?

One-Wayness

One-Wayness

For all PPT adversaries \mathcal{A} and security parameters κ there is a negligible function ϵ such that:

$$\Pr \left[\begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\kappa), m \xleftarrow{R} \mathcal{M}, \\ m^* \leftarrow \mathcal{A}(\text{pk}, \text{Enc}(m, \text{pk})) : m^* = m \end{array} \right] \leq \epsilon(\kappa).$$

- Not meaningful for most applications of PKE (but okay for RSA-KEM)
- \mathcal{A} may still compute some information about m
- How to formalize "does not leak any information"?

More Powerful Passive Adversaries (CPA)

Experiment $\text{Exp}_{\Pi, \mathcal{A}}^{\text{IND-CPA}}(\cdot)$:

Adversary \mathcal{A}



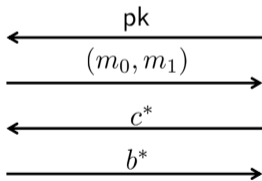
Challenger \mathcal{C}

$(sk, pk) \leftarrow \text{KeyGen}(1^\kappa)$

$b \leftarrow^R \{0, 1\}$

$c^* \leftarrow \text{Enc}(m_b, pk)$

if $b^* = b$ return 1;
else return 0;



IND-CPA Security

- We can define the advantage of adversary \mathcal{A} for the IND-CPA experiment for scheme Π as

$$\mathbf{Adv}_{\Pi, \mathcal{A}}^{\text{IND-CPA}}(\kappa) = \left| \Pr[\mathbf{Exp}_{\Pi, \mathcal{A}}^{\text{IND-CPA}}(\kappa) = 1] - \frac{1}{2} \right|.$$

- For a secure scheme, the advantage is negligible as a function of κ for any PPT \mathcal{A}

IND-CPA

For all PPT adversaries \mathcal{A} and security parameters κ there is a negligible function ϵ such that:

$$\Pr \left[\begin{array}{l} (\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^\kappa), ((m_0, m_1), \text{state}) \leftarrow \mathcal{A}(\text{pk}), \\ b \leftarrow \{0, 1\}, c \leftarrow \text{Enc}(m_b, \text{pk}), b^* \leftarrow \mathcal{A}(\text{state}, c) : b^* = b \end{array} \right] \leq \frac{1}{2} + \epsilon(\kappa).$$

IND-CPA Security With RSA

- Textbook RSA is obviously not IND-CPA secure
 - It is deterministic: \mathcal{A} simply computes $c' \leftarrow \text{Enc}(m_0, \text{pk})$ and outputs 0 if $c' = c$ and 1 otherwise
 - Leaks Jacobi symbol
 - If $e = 3$ and $m < N^{\frac{1}{3}}$, then $m = c^{\frac{1}{3}}$ (in the integers)
- No deterministic PKE scheme can be IND-CPA secure: encryption has to be randomized

IND-CPA Security With RSA

- Hard-core bit for RSA and IND-CPA security
 - Modify RSA assumption to output z such that z is least significant bit (LSB) of m
 - If you can compute LSB, then you can invert RSA
 - LSB is hardest bit to compute in RSA (a **hard-core bit**)
 - Can be used for encryption, but inefficient (bitwise)

$$\text{Enc}(m, \text{pk}) := (\text{LSB}(x) \oplus m, x^e \bmod N) \text{ for } m \in \{0, 1\} \text{ and } x \xleftarrow{R} \mathbb{Z}_N$$

$$\text{Dec}((c_1, c_2), \text{sk}) := \text{LSB}(c_2^d \bmod N) \oplus c_1$$

IND-CPA Security With RSA

- More efficiency with random oracles (RSA-CPA)

- Let $H : \mathbb{Z}_N \rightarrow \{0, 1\}^\ell$ be a random oracle

$$\text{Enc}(m, \text{pk}) := (H(x) \oplus m, x^e \bmod N) \text{ for } m \in \{0, 1\}^\ell \text{ and } x \xleftarrow{R} \mathbb{Z}_N$$

$$\text{Dec}((c_1, c_2), \text{sk}) := H(c_2^d \bmod N) \oplus c_1$$

- IND-CPA secure in the random oracle model

RSA-CPA IND-CPA Proof Idea

- To obtain information about m from $(H(x) \oplus m, x^e \pmod{N})$, one has to learn some information about $H(x)$
- As H is a random oracle, the only way to learn any information about $H(x)$ is to evaluate H at x
- An adversary who learns anything about m thus knows x
- The adversary thus can break the RSA assumption
- If adversary does not query $H(x)$, then challenge ciphertext c is independent from m_b

Concrete vs. Asymptotic Security

- Asymptotic security does not care about the runtime of the reduction (as long as polynomial time)
- Concrete security relates runtime t and success probability ϵ of adversary to t' and ϵ' of reduction
- Reduction is tight if $\epsilon \approx \epsilon'$ and $t \approx t'$ ($\frac{t'}{\epsilon'} \approx \frac{t}{\epsilon}$)
- Non-tight if $t \ll t'$ or if $\epsilon \gg \epsilon'$ (tightness gap is $\frac{t'\epsilon}{t\epsilon'}$)
 - $\frac{t'}{\epsilon'} \geq q_{\mathcal{O}} \cdot \frac{t}{\epsilon}$, where \mathcal{O} is some oracle (RO, signing, etc.)
- Tightness relates security of the scheme to the problem

RSA-CPA IND-CPA Proof

Theorem

If there exists an (t, q_H, ϵ) IND-CPA adversary against RSA-CPA, then there is a (t', ϵ') solver for the RSA assumption with $\epsilon' \geq 2\epsilon$ and $t' \leq t + (q_H^2 + q_H \cdot t_{exp})$.

Proof (by Reduction):

- Reduction \mathcal{B} obtains RSA challenge (e, y, N) (want to find x s.t. $y \equiv x^e \pmod{N}$)
- \mathcal{B} runs \mathcal{A} on $\text{pk} = (e, N)$ and obtains challenge (m_0, m_1)
- \mathcal{B} gives ciphertext (r, y) to \mathcal{A} for $r \xleftarrow{R} \{0, 1\}^\ell$ and RSA challenge y (as long as x s.t. $y \equiv x^e \pmod{N}$ not queried to H , challenge ciphertext information theoretically hidden)

RSA-CPA IND-CPA Proof

Theorem

If there exists an (t, q_H, ϵ) IND-CPA adversary against RSA-CPA, then there is a (t', ϵ') solver for the RSA assumption with $\epsilon' \geq 2\epsilon$ and $t' \leq t + (q_H^2 + q_H \cdot t_{exp})$.

Proof (by Reduction):

- Reduction \mathcal{B} obtains RSA challenge (e, y, N) (want to find x s.t. $y \equiv x^e \pmod{N}$)
- \mathcal{B} runs \mathcal{A} on $\text{pk} = (e, N)$ and obtains challenge (m_0, m_1)
- \mathcal{B} gives ciphertext (r, y) to \mathcal{A} for $r \xleftarrow{R} \{0, 1\}^\ell$ and RSA challenge y (as long as x s.t. $y \equiv x^e \pmod{N}$ not queried to H , challenge ciphertext information theoretically hidden)

RSA-CPA IND-CPA Proof (ctd.)

Simulation of the random oracle H (maintainig a list Q of tuples (x_i, h_i) - initially empty)

$H(x_j)$

```
1  $z \leftarrow x_j^e \pmod{N}$ 
2 if  $z = y$  then
3 output  $x_j$  and  $\mathcal{B}$  aborts // solved RSA challenge
4 else
5     if  $x_j$  in  $Q$  then
6         return  $h_j$ 
7     else
8          $h_j \xleftarrow{R} \{0,1\}^\ell$ 
9         store  $(x_j, h_j)$  in  $Q$ 
10        return  $h_j$ 
11    end if
12 end if
```

RSA-CPA IND-CPA Proof (ctd.)

- W event that \mathcal{A} wins the IND-CPA game (with prob. $\frac{1}{2} + \epsilon$); Q event that \mathcal{A} queries $H(x)$ s.t. $y = x^e \pmod{N}$.

$$\begin{aligned}\Pr[W] &= \Pr[W|Q] \cdot \Pr[Q] + \Pr[W|\neg Q] \cdot \Pr[\neg Q] \\ &\leq \Pr[Q] + \frac{1}{2} \cdot \Pr[\neg Q] \\ &= \Pr[Q] + \frac{1}{2}(1 - \Pr[Q]) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ \frac{1}{2} + \epsilon &\leq \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ 2\epsilon &\leq \underbrace{\Pr[Q]}_{\leq \epsilon'}.\end{aligned}$$

- If ϵ non-negl., so is ϵ' ; contradicting RSA assumption. □
- $t' \leq t + (q_H^2 + q_H \cdot t_{exp})$ (search in Q and one exp. per call)

RSA-CPA IND-CPA Proof (ctd.)

- W event that \mathcal{A} wins the IND-CPA game (with prob. $\frac{1}{2} + \epsilon$); Q event that \mathcal{A} queries $H(x)$ s.t. $y = x^e \pmod{N}$.

$$\begin{aligned}\Pr[W] &= \Pr[W|Q] \cdot \Pr[Q] + \Pr[W|\neg Q] \cdot \Pr[\neg Q] \\ &\leq \Pr[Q] + \frac{1}{2} \cdot \Pr[\neg Q] \\ &= \Pr[Q] + \frac{1}{2}(1 - \Pr[Q]) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ \frac{1}{2} + \epsilon &\leq \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ 2\epsilon &\leq \underbrace{\Pr[Q]}_{\leq \epsilon'}.\end{aligned}$$

- If ϵ non-negl., so is ϵ' ; contradicting RSA assumption. □
- $t' \leq t + (q_H^2 + q_H \cdot t_{exp})$ (search in Q and one exp. per call)

RSA-CPA IND-CPA Proof (ctd.)

- W event that \mathcal{A} wins the IND-CPA game (with prob. $\frac{1}{2} + \epsilon$); Q event that \mathcal{A} queries $H(x)$ s.t. $y = x^e \pmod{N}$.

$$\begin{aligned}\Pr[W] &= \Pr[W|Q] \cdot \Pr[Q] + \Pr[W|\neg Q] \cdot \Pr[\neg Q] \\ &\leq \Pr[Q] + \frac{1}{2} \cdot \Pr[\neg Q] \\ &= \Pr[Q] + \frac{1}{2}(1 - \Pr[Q]) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ \frac{1}{2} + \epsilon &\leq \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ 2\epsilon &\leq \underbrace{\Pr[Q]}_{\leq \epsilon'}.\end{aligned}$$

- If ϵ non-negl., so is ϵ' ; contradicting RSA assumption. □

- $t' \leq t + (q_H^2 + q_H \cdot t_{exp})$ (search in Q and one exp. per call)

RSA-CPA IND-CPA Proof (ctd.)

- W event that \mathcal{A} wins the IND-CPA game (with prob. $\frac{1}{2} + \epsilon$); Q event that \mathcal{A} queries $H(x)$ s.t. $y = x^e \pmod{N}$.

$$\begin{aligned}\Pr[W] &= \Pr[W|Q] \cdot \Pr[Q] + \Pr[W|\neg Q] \cdot \Pr[\neg Q] \\ &\leq \Pr[Q] + \frac{1}{2} \cdot \Pr[\neg Q] \\ &= \Pr[Q] + \frac{1}{2}(1 - \Pr[Q]) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ \frac{1}{2} + \epsilon &\leq \frac{1}{2} + \frac{1}{2} \cdot \Pr[Q] \\ 2\epsilon &\leq \underbrace{\Pr[Q]}_{\leq \epsilon'}.\end{aligned}$$

- If ϵ non-negl., so is ϵ' ; contradicting RSA assumption. □
- $t' \leq t + (q_H^2 + q_H \cdot t_{exp})$ (search in Q and one exp. per call)

Tightness of the Reduction

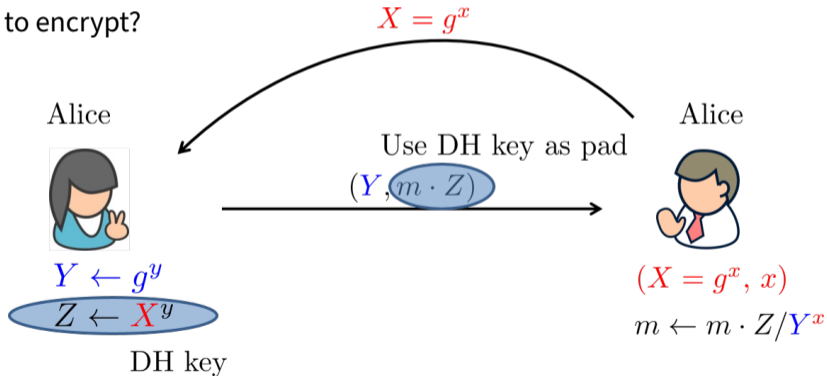
- RSA-CPA has a tight reduction (additive factor)
- RSA-FDH has **no** tight reduction (multiplicative factor)
 - Bellare et al.'s proof loses a factor of q_H
 - Coron's proof only loses a factor of q_S
 - It is often assumed that $q_H \leq 2^{60}$ and $q_S \leq 2^{30}$
 - Assume RSA with 80 bit security: 1248 bit modulus (ECRYPT II)
 - To obtain this security level w.r.t. Bellare et al.'s analysis we at least require 4000 bit RSA!

Encrypting With Diffie-Hellman

- Let \mathbb{G} be a group of prime order p and g a generator
- No trapdoor known to invert discrete exponentiation function
- How to encrypt?

Encrypting With Diffie-Hellman

- Let \mathbb{G} be a group of prime order p and g a generator
- No trapdoor known to invert discrete exponentiation function
- How to encrypt?



ElGamal Encryption Scheme

ElGamal

KeyGen(1^κ): Pick group $\mathbb{G} = \langle g \rangle$ with $|\mathbb{G}| = p \approx 2^\kappa$ prime, pick $x \xleftarrow{R} \mathbb{Z}_p$ and output $(\text{sk}, \text{pk}) \leftarrow (x, X = g^x)$

Enc(m, pk): Let $m \in \mathbb{G}$, pick $y \xleftarrow{R} \mathbb{Z}_p$ and output $(c_1, c_2) \leftarrow (g^y, m \cdot X^y)$

Dec(c, sk): Let $c = (c_1, c_2)$, compute and output $m \leftarrow c_2 / c_1^x$

ElGamal Encryption Scheme (ctd.)

Recall: DDH Assumption

Let $\mathbb{G} = \langle g \rangle$ with $|\mathbb{G}| = p$ prime, $\log_2 p = \kappa$, then \forall PPT \mathcal{A}

$$|\Pr[x, y \xleftarrow{R} \mathbb{Z}_p : \mathcal{A}(g^x, g^y, g^{xy}) = 1] - \Pr[x, y, z \xleftarrow{R} \mathbb{Z}_p : \mathcal{A}(g^x, g^y, g^z) = 1]| \leq \epsilon(\kappa)$$

and let us denote this probability as $\mathbf{Adv}_{\mathbb{G}, g, p}^{\text{DDH}}(\mathcal{A})$.

Theorem

If DDH assumption holds, ElGamal is IND-CPA.

ElGamal IND-CPA Proof

Proof (by Game Hopping):

In future Lecture!

Need to first look at constructing proofs via Game Hopping!

- Basic Idea:
 - Write down scheme as algorithm
 - Transform algorithm by changing it slightly
 - Argue that Adversary cannot distinguish between old and new algorithm
 - Repeat until we arrive at one algorithm that cannot be broken
 - e.g., it does not have access to the secret key at all

ElGamal and DDH

- Careful with choice of groups
 - In \mathbb{Z}_p^* for p prime DDH is not hard (take prime order q subgroup, e.g., $p = 2q + 1$)
 - In symmetric pairings no DDH; in the XDH setting DDH is not hard in \mathbb{G}_2
- Can switch to Linear ElGamal (DLIN)

DLIN Assumption

Let \mathbb{G} with $|\mathbb{G}| = p$, $\log_2 p = \kappa$ and $u, v, h \in \mathbb{G}$, then \forall PPT \mathcal{A}

$$|\Pr[x, y \xleftarrow{R} \mathbb{Z}_p : \mathcal{A}(u^x, v^y, h^{x+y}) = 1] - \Pr[x, y, z \xleftarrow{R} \mathbb{Z}_p : \mathcal{A}(u^x, v^y, h^z) = 1]| \leq \epsilon(\kappa)$$

Linear ElGamal

Linear ElGamal

KeyGen(1^κ): Pick group $\mathbb{G} = \langle g \rangle$ with $|\mathbb{G}| = p \approx 2^\kappa$ prime, pick $u, v \xleftarrow{R} \mathbb{Z}_p, h \xleftarrow{R} \mathbb{G}$, set $(U, V, h) \leftarrow (h^{1/u}, h^{1/v}, h)$ and output $(\text{sk}, \text{pk}) \leftarrow ((u, v), (U, V, h))$

Enc(m, pk): Let $m \in \mathbb{G}$, pick $y, z \xleftarrow{R} \mathbb{Z}_p$ and output $(c_1, c_2, c_3) \leftarrow (U^y, V^z, m \cdot h^{y+z})$

Dec(c, sk): Let $c = (c_1, c_2, c_3)$, compute and output $m \leftarrow c_3 / (c_1^u \cdot c_2^v)$

Theorem

If DLIN assumption holds, Linear ElGamal is IND-CPA.

Problems With IND-CPA Security

Malleability: Adversary may change a ciphertext such that plaintexts are related

- RSA-CPA:

$$(H(x) \oplus m \oplus m', x^e \bmod N)$$

- ElGamal:

$$(g^{y_0}, m_0 X^{y_0}) \star (g^{y_1}, m_1 X^{y_1}) = (g^{y_0+y_1}, (m_0 \cdot m_1) X^{y_0+y_1})$$

- Sometimes desired (computing on encrypted data)
- Sometimes problematic (e.g., Bleichenbacher)

Active Adversaries (CCA)

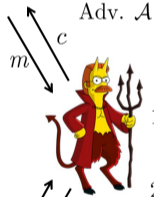
- How to formalize malleability (NM)? Dolev et al. have done this back in 1993 with a simulation-based notion
- Bellare et al. have shown that NM implies the IND notion
 - The strongest notion NM-CCA2 is equivalent to IND-CCA2
 - IND notion is more convenient to use
- Idea of a stronger IND notion
 - Give the adversary access to a decryption oracle
- IND-CCA2 automatically yields security in universal composability (UC) framework

Active Adversaries (CCA)

Oracle $\mathcal{O}_1^{\text{Dec}}(\cdot, \text{sk})$



Adv. \mathcal{A}



Oracle $\mathcal{O}_2^{\text{Dec}}(\cdot, \text{sk})$



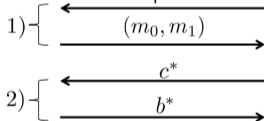
Challenger \mathcal{C}

$(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\kappa)$

$b \xleftarrow{R} \{0, 1\}$

$c^* \leftarrow \text{Enc}(m_b, \text{pk})$

if $b^* = b$ return 1;
else return 0;



Active Adversaries (CCA)

- IND-CCA1 (lunchtime attacks)
 - \mathcal{A} only access to $\mathcal{O}_1^{\text{Dec}}$ (before seeing the challenge ciphertext)
 - Best we can get for homomorphic schemes (within our notions)
- IND-CCA2
 - \mathcal{A} also has access to $\mathcal{O}_2^{\text{Dec}}$ (after seeing the challenge)
 - Is not allowed to submit c^*

Active Adversaries (CCA)

- IND-CCA1 (lunchtime attacks)
 - \mathcal{A} only access to $\mathcal{O}_1^{\text{Dec}}$ (before seeing the challenge ciphertext)
 - Best we can get for homomorphic schemes (within our notions)
- IND-CCA2
 - \mathcal{A} also has access to $\mathcal{O}_2^{\text{Dec}}$ (after seeing the challenge)
 - Is not allowed to submit c^*

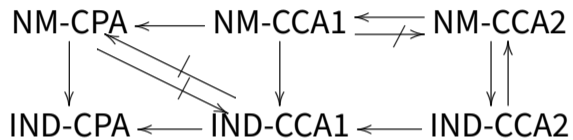
IND-CCA2 Schemes?

- In the random oracle model
 - RSA-OAEP(+) (RSA)
 - Hybrid ElGamal (strong DH)
 - Twin Hybrid ElGamal (DH)
- Without random oracles
 - Hash proof system (e.g., Cramer-Shoup)
 - From CPA secure IBE schemes
 - Twin-encryption using non-interactive zero-knowledge
- Conversion from IND-CPA (e.g., Fujisaki-Okamoto)

Theory vs. Practice

- Many "pure" PKES not useful for content encryption
- Use of hybrid encryption (KEM/DEM) approach
 - Combine PKE with symmetric encryption (and MAC or RO)
 - Generic conversions follow this approach
- Plain PKE schemes still very useful
 - Homomorphic encryption
 - Threshold encryption
 - Zero-knowledge proofs of knowledge (plaintext equality, inequality)

Relation Among Notions



Additional Security Notions for PKE

- **Replayable CCA (RCCA)**
 - IND-CPA too weak and IND-CCA2 often too strong
 - Capture schemes that are CCA2-secure except for allowing re-randomization of ciphertexts
 - Altering ciphertext is "ok", if it decrypts to original message
- **Circular Security**
 - Encrypt a secret key under the corresponding public key (1-cycle)
 - Important for fully homomorphic encryption (bootstrapping)
 - Homomorphically evaluating decryption function on ciphertext (use encryption of secret key)

Additional Security Notions for PKE

- Key Dependent Message (KDM) security
 - Generalization of circular security
 - Encrypted messages might depend on arbitrary function of secret keys
- Security under leakage
 - Leak a bounded number of bits of secret key
 - Leak an adversarially chosen function of secret key
 - ...

What you should know...

- Security models for PKE
 - Active and passive adversaries
 - “Games” for different Adversaries
- Asymptotic vs. concrete security
- Basic Idea of (Random) Oracles

Questions?

Further Reading I

- [1] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway.

Relations Among Notions of Security for Public-Key Encryption Schemes.

In Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings, pages 26–45, 1998.

- [2] Dan Boneh and Xavier Boyen.

Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups.

J. Cryptology, 21(2):149–177, 2008.

- [3] Dan Boneh, Ben Lynn, and Hovav Shacham.

Short Signatures from the Weil Pairing.

J. Cryptology, 17(4):297–319, 2004.

Further Reading II

- [4] Jean-Sébastien Coron.

On the exact security of full domain hash.

In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer, 2000.

- [5] Rosario Gennaro, Shai Halevi, and Tal Rabin.

Secure hash-and-sign signatures without the random oracle.

In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 123–139, 1999.

- [6] Susan Hohenberger and Brent Waters.

Short and stateless signatures from the RSA assumption.

In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 654–670, 2009.

Further Reading III

[7] Jonathan Katz.

Digital Signatures.

Springer, 2010.

[8] Brent Waters.

Efficient identity-based encryption without random oracles.

In Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, pages 114–127, 2005.