# Security Reduction

## How to understand Breaking Assumption


**Fuchun Guo**
University of Wollongong
fuchun@uow.edu.au

卧村密碼学報

# Outline

- What is breaking assumption
- How to understand breaking assumption
- How to use breaking assumption
  - ➢ Digital Signatures
  - ➢ Encryption



# What is breaking assumption?

# Breaking Assumption

- When we propose a scheme, we need to prove its security.
- One of security proofs is called "security reduction"

**Proof.** Suppose there is an adversary who can break the proposed scheme in defined security model with non-negligible advantage. **(This is called breaking assumption)**

……
……
……
……
……
……

# Breaking Assumption

**Breaking Assumption**： Suppose there is an adversary who can break the proposed scheme in defined security model with non-negligible advantage.

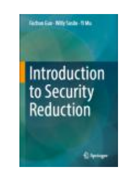
- Adversary
- Break the propose scheme
- Security model
- Non-negligible advantage



# Breaking Assumption: Adversary


- The adversary can break the proposed scheme.

- The adversary stays in a very safe box (blackbox).
- It can hear what we say, and we can hear what it says.
- However, we cannot see what it writes and calculates when it is breaking our proposed scheme.



# Breaking Assumption: Adversary


- The security proof is to prove that the adversary in the blackbox cannot break (meaning the scheme is secure).
- If we could see what it writes and calculates, the adversary must exist and the scheme must be insecure!



# Breaking Assumption: Adversary


What will the adversary say?

Well, we partially know what the adversary will say.

# Breaking Assumption: Break the Proposed Scheme


Are you the proposed scheme? →

← Yes

I can break you in XX security model →

Scheme

# Breaking Assumption: Break the Proposed Scheme


Are you the proposed scheme? →

← Yes

I can break you in XX security model →

← Okay!  Let's start!

**Scheme**

## Then they interact following the security model!

(The defined security model in breaking assumption)



# Breaking Assumption: Example


Give me the public key →

← pk

Give me the signature on m_1 →

← Here is the signature $\sigma$_1

…………

Give me the signature on m_q →

← Here is the signature $\sigma\_q$

Here is my forged signature $\sigma$* on m* →

**Scheme**

The EUF-CMA security model for signatures



# Breaking Assumption: Non-Negligible Advantage


Are you the proposed scheme? →

← Yes

I can break you in XX security model →

← Okay!  Let's start!

**Scheme**

- The adversary might fail in breaking the scheme.
- The adversary will successfully break it with non-negligible advantage $\epsilon$ only.
- It is advantage not probability  (for universal definition).
- We cannot consider $\epsilon$=1 because non-negligible is already dangerous.



# How to understand breaking assumption?

# Breaking Assumption: Adversary


If we meet the adversary in Wollongong, he will say
"Hey I am the Alien!"

Question: What will the adversary say if we meet him in Sydney?

Answer:________________?

# Breaking Assumption: Adversary


If we meet the adversary in Wollongong, he will say
"Hey I am the Alien!"

Question: What will the adversary say if we meet him in Sydney?

Answer: **We don't know!**

# Breaking Assumption: Adversary


If we meet the adversary in Wollongong, he will say "Hey I am the Alien!"

Question: What will the adversary say if we meet him in Sydney?

Will the adversary still say "Hey I am the Alien!"? Yes, possible.

What is the probability of saying the same? We don't know.



# Breaking Assumption: Break the Proposed Scheme


Are you the proposed scheme? →

← Yes

I can break you in XX security model →

**Scheme**


Are you the proposed scheme? →

← **NO**

**?????????????????????** →

**Scheme**



# Breaking Assumption: Break the Proposed Scheme


Are you the proposed scheme?

NO

??????????????????????

Scheme

?????  means

**We don't know what will happen!**

# Breaking Assumption: Break the Proposed Scheme


Scheme

Give me the public key →

← pk

Give me the signature on m_1 →

← Here is the signature $\sigma$_1

…………

Give me the signature on m_q →

← **Can I give you signatures on other messages?**

??????????????????????????? →

The EUF-CMA security model for signatures



# Breaking Assumption: Break the Proposed Scheme


Give me the public key →

← pk

Give me the signature on m_1 →

← Here is the signature $\sigma$_1

............

Give me the signature on m_q →

← **Here is the signature (But actually it is invalid)**

??????????????????????????? →

Scheme

The EUF-CMA security model for signatures



# Breaking Assumption: Summary


Give me the public key →

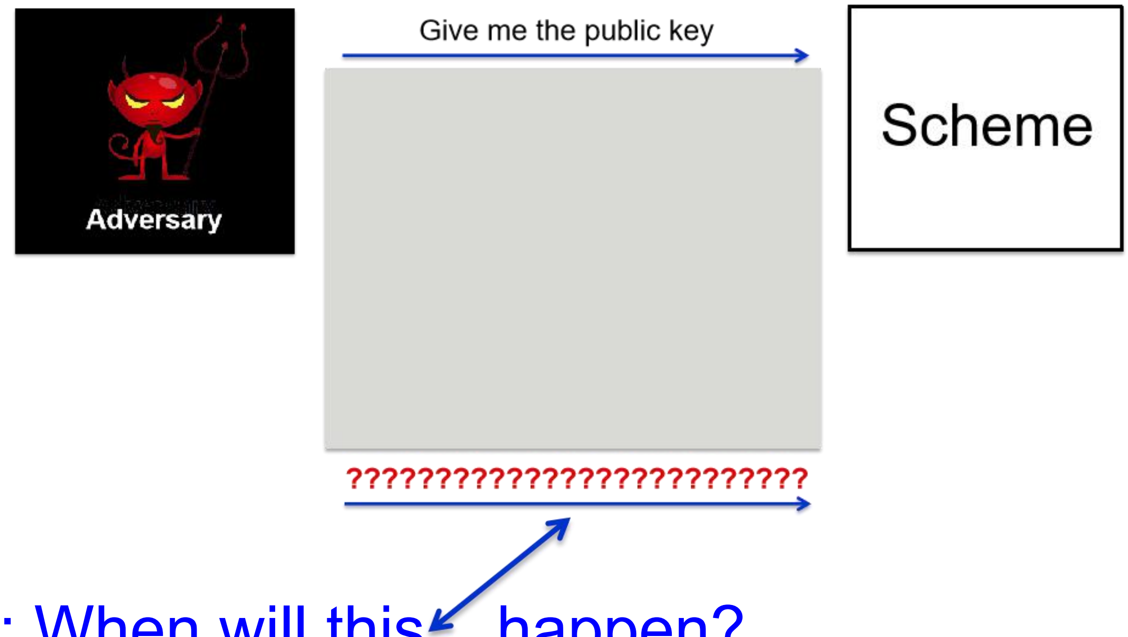
Scheme

???????????????????????????

Question: When will this happen?



# Breaking Assumption: Summary




Question: When will this happen?

**It happens if**

- **The adversary has a proof showing that it is not real, or**
- **The scheme doesn't follow the security model.**

# Breaking Assumption: Summary


Are you the proposed scheme? →

← Yes

Will you follow the security model? →

← Yes

I will break you but with non-negligible advantage only! →

Scheme

**Then they interact following the security model!**

(The defined security model in breaking assumption)



# How to use breaking assumption?



# Breaking Assumption: How to use

In security reduction, we use a hard problem instance to create a simulated scheme.


Are you the proposed scheme? →

← ~~NO~~ Yes

Will you follow the security model? →

← ~~Try my best~~ Yes

I will break you but with non-negligible advantage only! →

Simulated Scheme

We reduce the attack to solving hard problem

# Breaking Assumption: How to use

In security reduction, we use a hard problem instance to create a simulated scheme.


Are you the proposed scheme? →

← ~~NO~~ Yes

Will you follow the security model? →

← ~~Try my best~~ Yes

I will break you but with non-negligible advantage only! →

**Simulated Scheme**

We reduce the attack to solving hard problem

Well, it looks easy but we have not yet seen the difficulity due to security model.

# How to use breaking assumption in Digital Signatures?

# Digital Signatures and Breaking Assumption


Give me the public key →

← pk

Give me the signature on m_1 →

← Here is the signature $\sigma$_1

…………

Give me the signature on m_q →

← Here is the signature $\sigma\_q$

Here is my forged signature $\sigma$* on m* →

**Scheme**

The EUF-CMA security model for signatures



# Digital Signatures and Breaking Assumption

In security reduction, we use a hard problem instance to create a simulated scheme.


Give me the public key →

← pk

............

↔

Here is my forged signature $\sigma$* on m* →

**Simulated Scheme**

We wish:

- The simulated scheme should look like the proposed scheme.
- We can use the forged signature to solve hard problem.



# Digital Signatures and Breaking Assumption

We wish:

- The simulated scheme should look like the proposed scheme.
- We can use the forged signature to solve hard problem.

In the EUF-CMA security model:

Question 1： What messages will the adversary query?

Adversary: **?????????????**(We don't know!)

We could have problems in simulation because we don't know what will be queried.



# Digital Signatures and Breaking Assumption

We wish:

- The simulated scheme should look like the proposed scheme.
- We can use the forged signature to solve hard problem.

In the EUF-CMA security model:

Question 1： What messages will the adversary query?

Adversary: ????????????(We don’t know!)

We could have problems in simulation because we don’t know what will be queried.

Question 2: What message will the adversary forge its signature?

Adversary: ????????????(We don’t know!)

We could have problems in reduction because we don’t know what will be forged.



# Idea for successful reduction


Block the Dead Road

↓

No matter what **?????????????** is, we can simulate it or reduce it!

# Example (Basic Scheme)

**KeyGen**: $\mathrm{pk} = (g, g^s, \mathrm{e}, \mathrm{p}),\ \mathrm{sk} = s$
**Sign:** Given message $m \in Z_p$,compute the signature $\sigma_m$ as

$$\sigma_m = g^{\frac{1}{s+m}}$$

Suppose that we will reduce its security to solve **exponent inverse problem**.
Input: $(g, g^a)$, Output: $g^{\frac{1}{a}}$

Proof. We set $g^s$=$g^a$ . In the key-only attack, the adversary should forge a signature on m* after seeing the public key.

Let the forged signature be $g^{\frac{1}{s+m*}} = g^{\frac{1}{a+m*}}$.
We cannot extract the problem solution from the forged signature because m* can be any integer chosen by the adversary.



# Example (To extact problem solution)

**KeyGen**: $\mathrm{pk} = (g, g^s, g^t, \mathrm{e}, \mathrm{p}),\ \mathrm{sk} = (\mathrm{s}, \mathrm{t})$
**Sign:** Given message $m \in Z_p$,compute the signature $\sigma_m$ as

$$\sigma_m = g^{\frac{1}{s+m \times t}}$$

Suppose that we will reduce its security to solve **exponent inverse problem**.
Input: $(g, g^a)$, Output: $g^{\frac{1}{a}}$

Proof. We set $g^s$=$g^a$ and $g^t$=$g^{k*a}$ using a random inetger k. In the key-only attack, the adversary should forge a signature on m* after seeing the public key.

Let the forged signature be $g^{\frac{1}{s+m* \times t}} = g^{\frac{1}{a+m* \times k \times a}} = g^{\frac{1}{a(1+m* \times k)}}$.
We can extract the problem solution from the forged signature no matter what m* is in the forged signature as long as it is randomly chosen.



# Example (Basic Scheme)

**KeyGen**: $\mathrm{pk} = (g, g^s, \mathrm{e}, \mathrm{p}),\ \mathrm{sk} = \mathrm{s}$
**Sign:** Given message $m \in Z_p$,compute the signature $\sigma_m$ as

$$\sigma_m = g^{\frac{1}{s+m}}$$

Suppose that we will reduce its security to solve **exponent inverse problem**.
Input: $(g, g^a)$, Output: $g^{\frac{1}{a}}$

Proof. We set $g^s$=$g^a$ . In the chosen-message attack (one query), the adversary will make signature query on m that is chosen by itself.

The queried signature is $g^{\frac{1}{s+m}} = g^{\frac{1}{a+m}}$.
We cannot compute the signature for the adversary because it is hard to compute it without knowing $a$ .

# Example (To simulate signature)

**KeyGen**: $\mathrm{pk} = (g, g^s, \mathrm{h}, \mathrm{e}, \mathrm{p})$, $\mathrm{sk} = (\mathrm{s}, \mathrm{t})$
**Sign:** Given message $m \in Z_p$, choose a random $r$ and compute signature $\sigma_m$ as

$$\sigma_m = (r, h^{\frac{1}{s+m+r}})$$

Suppose that we will reduce its security to solve **exponent inverse problem**.
Input: $(g, g^a)$, Output: $g^{\frac{1}{a}}$

Proof. We set $g^s$=$g^a$. We choose random w,z and set $h = g^{z(a+w)}$

For a signature query on message m, we set $r = w - m$.
We have $h^{\frac{1}{s+m+r}}$=$h^{\frac{1}{a+m+r}} = h^{\frac{1}{a+w}} = g^z$

We can simulate the signature for the adversary no matter what the queried message m is by the adversary.



## Example (Basic Scheme)

**KeyGen**: $\mathrm{pk} = (g, g^s, \mathrm{e}, \mathrm{p}),\ \mathrm{sk} = \mathrm{s}$

**Sign:** Given message $m \in Z_p$,compute the signature $\sigma_m$ as

$$\sigma_m = g^{\frac{1}{s+m}}$$

## Example (To extact problem solution)

**KeyGen**: $\mathrm{pk} = (g, g^s, g^t, \mathrm{e}, \mathrm{p}),\ \mathrm{sk} = (\mathrm{s}, \mathrm{t})$

**Sign:** Given message $m \in Z_p$,compute the signature $\sigma_m$ as

$$\sigma_m = g^{\frac{1}{s+m\times t}}$$

## Example (To simulate signature)

**KeyGen**: $\mathrm{pk} = (g, g^s, \mathrm{h}, \mathrm{e}, \mathrm{p}),\ \mathrm{sk} = (\mathrm{s}, \mathrm{t})$

**Sign:** Given message $m \in Z_p$,choose a random $r$ and compute signature $\sigma_m$ as

$$\sigma_m = \left(r, h^{\frac{1}{s+m+r}}\right)$$



# Reduction in Digital Signatures (Conclusion)

We can simulate one signature no matter what message m is quried by the adversary

↓

We can simulate polynomial signatures no matter what messages are quried by the adversary

We can extract problem solution no matter what message m* in the forged signature

↓

All conditions must hold in one signature scheme!



# Reduction in Digital Signatures (Conclusion)

We can simulate one signature no matter what message m is quried by the adversary

↓

We can simulate polynomial signatures no matter what messages are quried by the adversary

We can extract problem solution no matter what message m* in the forged signature

↓

All conditions must hold in one signature scheme!

↓

Then?

# Reduction in Digital Signatures (Conclusion)

We can simulate one signature no matter what message m is quried by the adversary

↓

We can extract problem solution no matter what message m* in the forged signature

We can simulate polynomial signatures no matter what messages are quried by the adversary

↓

**All conditions must hold in one signature scheme!**

↓ Simulatable | Reducible

The adversary cannot distinguish which forged signatures are simulatable.
(The advantage of launching useful attack is non-negligible)

↓ Malicious | Unbounded

**We can solve the hard problem with non-negligible probability.**

# How to use breaking assumption in Encryption (Decisional Version)?

# Public-Key Encryption


Give me the public key →

← pk

Here are two messages M_0, M_1 →

← Challenge Cipher CT=E(pk,M_c)

Here is my guess c' →

**Scheme**

The IND-CPA security model for encryption



# Decisional Problem

A decisional problem generated with a security parameter $\lambda$ is hard if, given as input a problem instance whose target is $Z$, the advantage of returning a correct guess in polynomial time is a negligible function of $\lambda$, denoted by $\varepsilon(\lambda)$ ($\varepsilon$ for short),

$$\varepsilon = \Pr\left[\text{Guess } Z = \text{True} | Z = \text{True}\right] - \Pr\left[\text{Guess } Z = \text{True} | Z = \text{False}\right],$$

where

- $\Pr\left[\text{Guess } Z = \text{True} | Z = \text{True}\right]$ is the probability of correctly guessing $Z$ if $Z$ is true.
- $\Pr\left[\text{Guess } Z = \text{True} | Z = \text{False}\right]$ is the probability of wrongly guessing $Z$ if $Z$ is false.

# Public-Key Encryption and Reduction

In security reduction, we use problem instance (I,Z) to create a simulated scheme.


Give me the public key →

pk ←

Here are two messages M_0, M_1 →

Challenge Cipher CT=E(pk,M_c) ←

Here is my guess c' →

**Simulated Scheme**

We run the interactions for 1000 times and we wish:

- The adversary will guess c correctly for 938 times when Z is true.
- The adversary will guess c correctly for 504 times when Z is false.

# Public-Key Encryption and Reduction

$$\varepsilon = \Pr\left[\text{Guess } Z = \text{True} | Z = \text{True}\right] - \Pr\left[\text{Guess } Z = \text{True} | Z = \text{False}\right],$$

We run the interactions for 1000 times and we wish:

- The adversary will guess c correctly for 938 times when Z is true.
- The adversary will guess c correctly for 504 times when Z is false.

Given I: (same in all reductions)

- If the adversary guesses c correctly，we guess Z=True;
- Otherwise， we guess Z=False;



# Public-Key Encryption and Reduction

$$\varepsilon = \Pr\left[\text{Guess } Z = \text{True} | Z = \text{True}\right] - \Pr\left[\text{Guess } Z = \text{True} | Z = \text{False}\right],$$

We run the interactions for 1000 times and we wish:

- The adversary will guess c correctly for 938 times when Z is true.
- The adversary will guess c correctly for 504 times when Z is false.

Given I: (same in all reductions)

- If the adversary guesses c correctly， we guess Z=True;
- Otherwise， we guess Z=False;

If the above happens, we have

- Pr[Guess Z=True|Z=True]= 938/1000
- Pr[Guess Z=True|Z=False]= 504/1000

Therefore, we solve the problem with the help of the adversary.

# Public-Key Encryption and Reduction


Give me the public key →

pk ←

Here are two messages M_0, M_1 →

Challenge Cipher CT=E(pk,M_c) ←

Here is my guess c' →

**Simulated Scheme**

We wish:

- The adversary will guess c correctly for 938 times when Z is true.

We do:

- Use the adversary's advantage of breaking scheme.
- The simulated scheme should look like proposed scheme.



# Public-Key Encryption and Reduction


Give me the public key →

pk ←

Here are two messages M_0, M_1 →

Challenge Cipher CT=E(pk,M_c) ←

Here is my guess c' →

**Simulated Scheme**

We wish:

The adversary will guess it correctly for 504 times when Z is false.

We hope:

- The advesary will NOT help or simply output a random guess



# Public-Key Encryption and Reduction


Give me the public key →

← pk

Here are two messages M_0, M_1 →

← Challenge Cipher CT=E(pk,M_c)

Here is my guess c' →

Simulated

Scheme

We wish:

- The adversary will guess it correctly for 504 times when Z is false.

We hope:

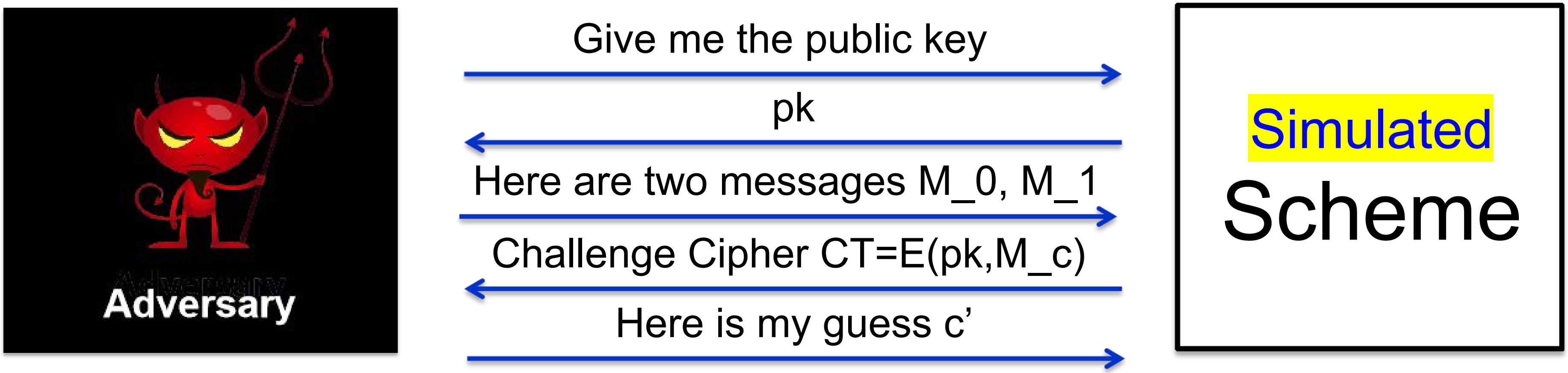
- The advesary will NOT help or simply output a random guess

Reality:

- If the simulated scheme looks like the proposed one, it will break/help.
- If it looks different, the adversary will ????????????????????????



# Public-Key Encryption and Reduction




We wish:

- The adversary will guess it correctly for 504 times when Z is false.

We hope:

- The advesary will NOT help or simply output a random guess

Reality:

- If the simulated scheme looks like the proposed one, it will break/help.
- If it looks different, the adversary will ?????????????????????????

**Solution:**

- The adversary has no advantage no matter what ??????????? is.

# Public-Key Encryption and Reduction

The adversary has no advantage no matter what ?????? is when Z is false

The simulated scheme should look like proposed scheme when Z is true.

The challenge cipher is a one-time pad

All conditions must hold in one simulated scheme!

Then?

# Public-Key Encryption and Reduction

The adversary has no advantage no matter what ?????? is when Z is false

↓

The simulated scheme should look like proposed scheme when Z is true.

The challenge cipher is a one-time pad

↓

All conditions must hold in one simulated scheme!

↓

**done!**

Note: It is harder to achieve both conditions in IND-CCA security model.



# Example (ElGamal Encryption)

**KeyGen**: $\text{pk} = (\text{g}, g_1, \text{p}, \text{H}) = (g, g^s, \text{p}, \text{H}),\ \text{sk} = \text{s}$
**Encrypt:** Given message $m \in G$,choose a random r and compute CT as

$$CT = (g^r, g_1^r * m)$$

Suppose that we will reduce its security to Decisional Diffie-Hellman problem.
Input: $(g, g^a, g^b, Z)$, Output: $Z = true$ if $Z = g^{ab}$ and false otherwise.

Proof. We set $g^s$=$g^a$ . In IND-CPA model, the pk is given to the adversary.

Given m_0 and m_1 from the challenger, we set $g^b = g^r$.
We set $CT = (g^b, Z * m\_c)$

If Z is true, it looks like the proposed scheme .
If Z is false, it is a one-time pad.

# How to use breaking assumption in Encryption (Computational Version)?



# Computational Problem

Given a problem instance I, the computational problem is to compute S.

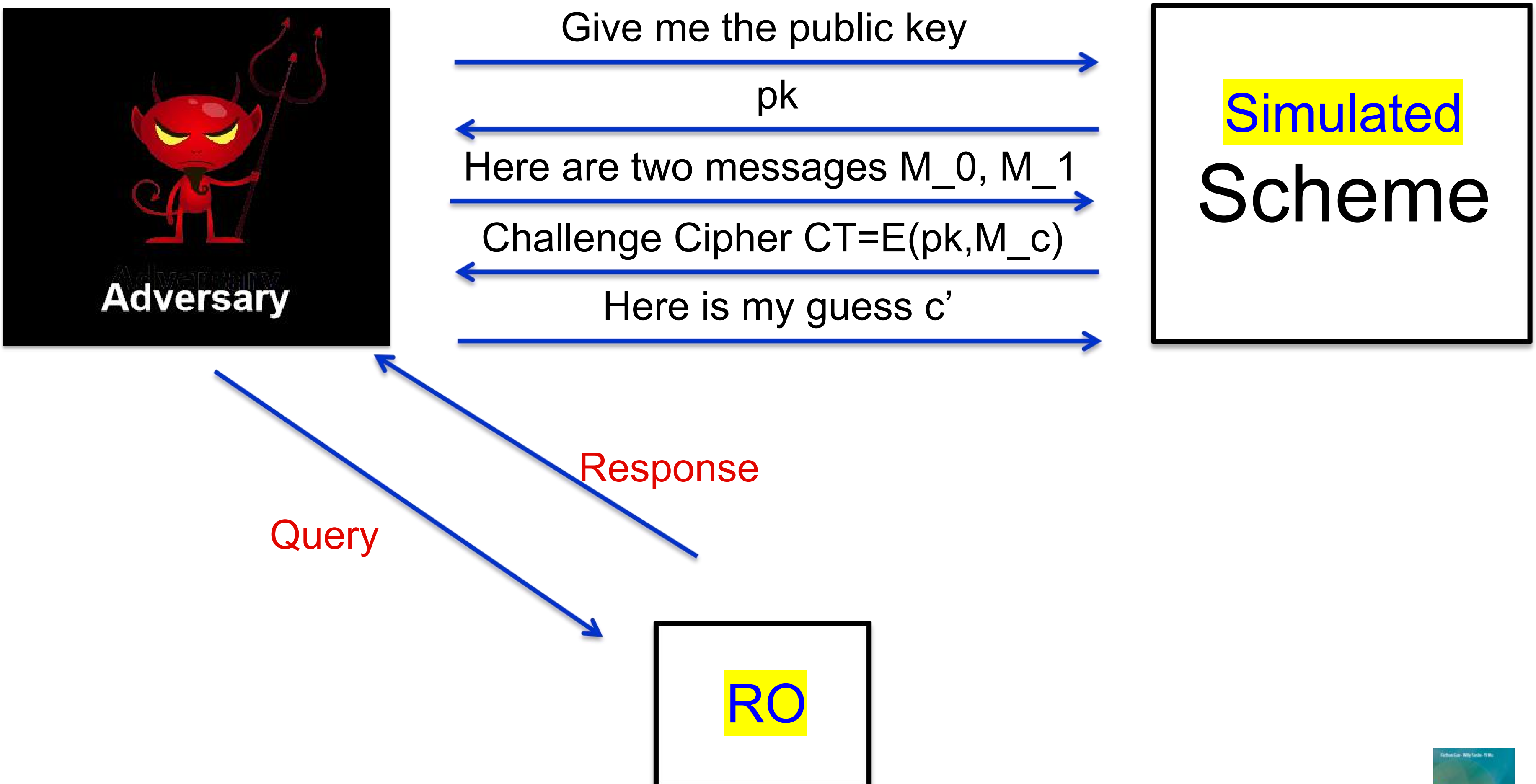
We say that the computational problem is hard for the PPT algorithm A() if

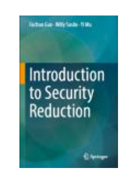
$$\Pr[A(I)=s] \leq \epsilon,$$

where the probability $\epsilon$ is a negligible function in security parameter.



# Public-Key Encryption and RO



# Public-Key Encryption and Reduction

In security reduction, we use problem instance I to create a simulated scheme.


Give me the public key →

pk ←

Here are two messages M_0, M_1 →

Challenge Cipher CT=E(pk,M_c) ←

Here is my guess c' →

Simulated
Scheme

We run the interactions for 1000 times and we wish:

- The adversary will query the problem solution S to RO for 938 times.



# Public-Key Encryption and Reduction


Give me the public key →

← pk

Here are two messages M_0, M_1 →

← Challenge Cipher CT=E(pk,M_c)

Here is my guess c' →

Simulated
Scheme

We wish:

- The adversary will query the problem solution S to RO for 938 times.

We do:

- The **???????** cannot appear before the adversary queries S



# Public-Key Encryption and Reduction


Give me the public key →

pk ←

Here are two messages M_0, M_1 →

Challenge Cipher CT=E(pk,M_c) ←

Here is my guess c' →

**Simulated Scheme**

We wish:

- The adversary will query the problem solution S to RO for 938 times.

We do:

- The **???????** cannot appear before the adversary queries S

- The simulate scheme looks like the proposed one before querying S.
- The adversary must query S in order to break.



# Public-Key Encryption and Reduction

In security reduction, we use problem instance I to create a simulated scheme.

The simulate scheme looks like the proposed one before querying S.

↓ Attack

The adversary has no advantage in breaking the scheme befor querying S.

↓ Attack by querying S (the only way)

**done!**

# Example (Hashed ElGamal Encryption)

**KeyGen**: $\mathrm{pk} = (\mathrm{g}, g_1, \mathrm{p}, \mathrm{H}) = (g, g^s, \mathrm{p}, \mathrm{H})$, $\mathrm{sk} = \mathrm{s}$
**Encrypt:** Given message $m \in \{0,1\}^n$,choose a random r and compute CT as

$$CT = (g^r, H(g_1^r) \oplus m)$$

Suppose that we will reduce its security to Computational Diffie-Hellman problem.
Input: $(g, g^a, g^b)$, Output: $g^{ab}$.

Proof. We set $g^s$=$g^a$ . In IND-CPA model, the pk is given to the adversary.

Given m_0 and m_1 from the challenger, we set $g^b = g^r$.
We set $CT = (g^b, R \oplus m\_c)$, here R is a randomly chosen bit string.

Before query S=$g^{ab}$, it looks like the proposed scheme .
Before query S=$g^{ab}$, the adversary has no advantage.



# Example (Hashed ElGamal Encryption)

**KeyGen**: $\mathrm{pk} = (\mathrm{g}, g_1, \mathrm{p}, \mathrm{H}) = (g, g^s, \mathrm{p}, \mathrm{H}),\ \mathrm{sk} = \mathrm{s}$

**Encrypt:** Given message $m \in \{0,1\}^n$,choose a random r and compute CT as

$$CT = (g^r, H(g_1^r) \oplus m)$$

Suppose that we will reduce its security to Computational Diffie-Hellman problem.
Input: $(g, g^a, g^b)$, Output: $g^{ab}$.

Proof. We set $g^s$=$g^a$ . In IND-CPA model, the pk is given to the adversary.

Given m_0 and m_1 from the challenger, we set $g^b = g^r$.
We set $CT = (g^b, (1||R) \oplus m\_c)$, here R is a random (n-1)- bit string.

Before query S=$g^{ab}$, it looks like the proposed scheme .
Before query S=$g^{ab}$, the adversary has advantage when m_0=0** and m_1=1**.



# Summary

- Breaking assumption. ???unpredictable???

- Make sure that the reduction works no matter what ????????????? is.

UNIVERSITY OF WOLLONGONG AUSTRALIA

Institute of Cybersecurity and Cryptology (IC²)

Journal of WoCrypt

# Q&A

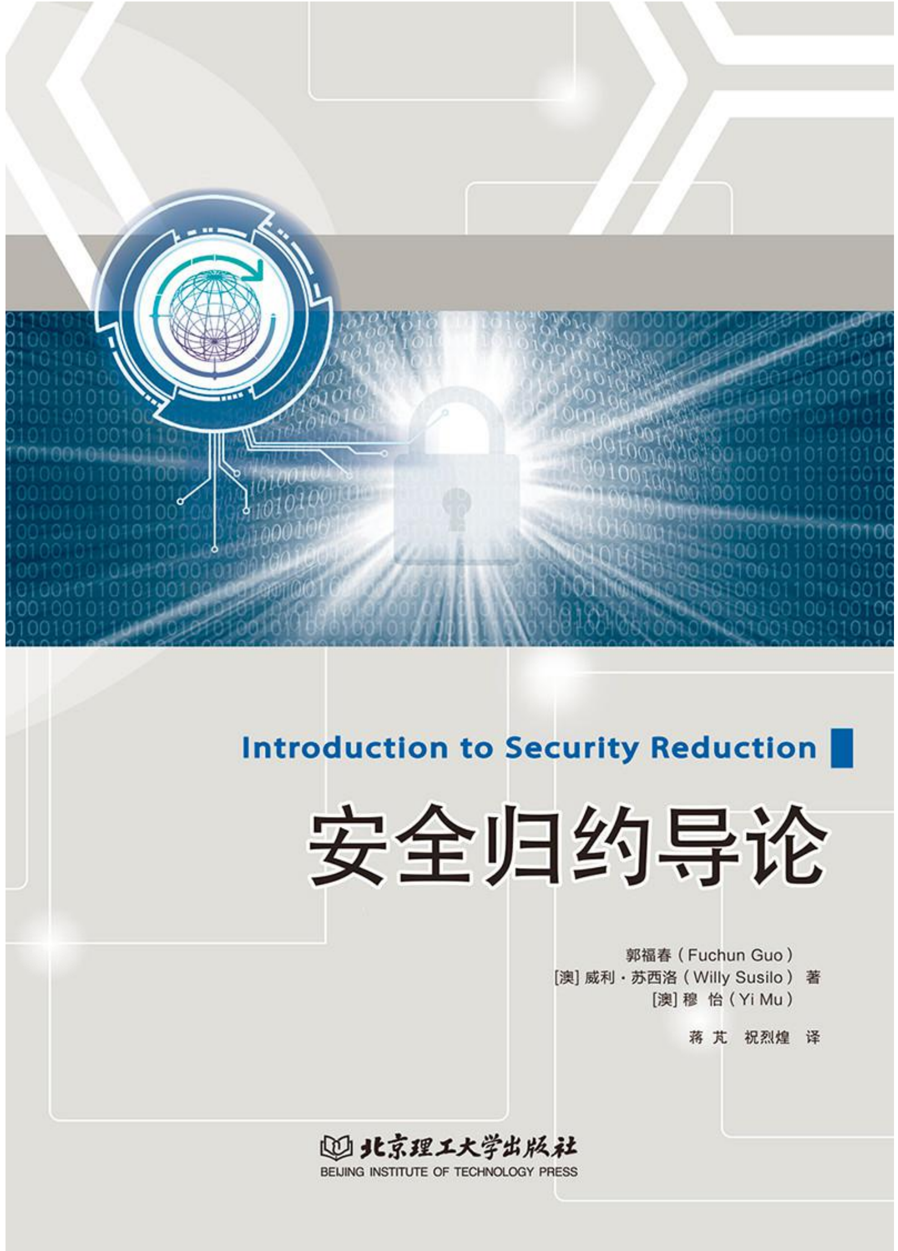
Adversary

My IQ is up to 186.

My interest is breaking schemes.

You want me to help you solve problem?

Fool me first!


Introduction to Security Reduction
安全归约导论
郭福春（Fuchun Guo）
[澳] 威利·苏西洛（Willy Susilo）著
[澳] 穆 怡（Yi Mu）
蒋 芃　祝烈煌　译
北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS


Introduction
to Security
Reduction