I – Basic Notions

David Pointcheval

MPRI – Paris

Ecole normale supérieure/PSL, CNRS & INRIA

Outline

Cryptography

Provable Security

Basic Security Notions

Conclusion

Cryptography

Outline

Cryptography

Introduction

Kerckhoffs' Principles

Formal Notations

Provable Security

Basic Security Notions

Conclusion

Secrecy of Communications

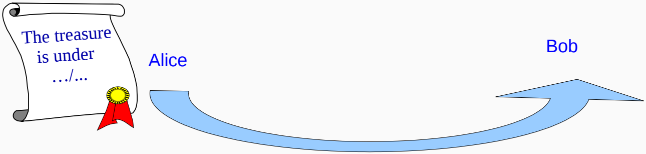
One ever wanted to communicate secretly

Alice

Bob

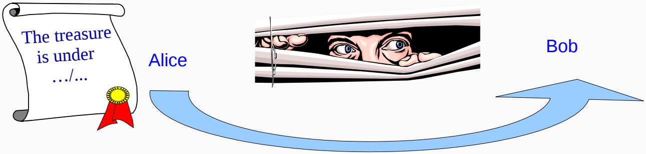
Secrecy of Communications

One ever wanted to communicate secretly



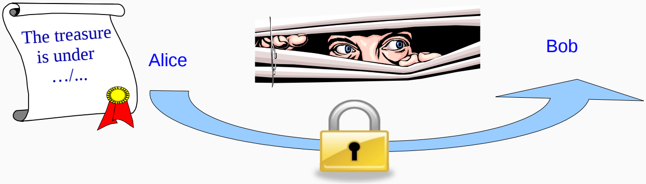
Secrecy of Communications

One ever wanted to communicate secretly



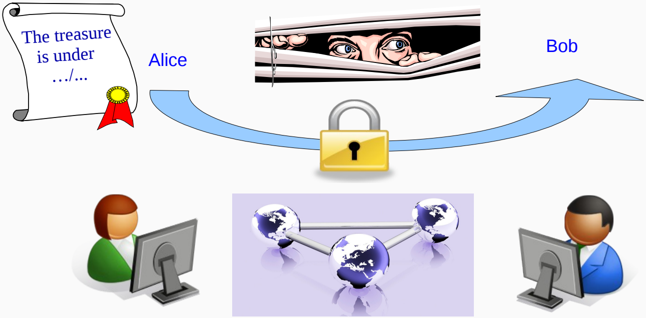
Secrecy of Communications

One ever wanted to communicate secretly



Secrecy of Communications

One ever wanted to communicate secretly



With the all-digital world, security needs are even stronger

Old Methods

Substitutions and permutations

Security relies on
the secrecy of the mechanism

Scytale - Permutation

Substitutions and permutations

Security relies on
the secrecy of the mechanism

Old Methods

Scytale - Permutation

Alberti's disk
Mono-alphabetical Substitution

Substitutions and permutations

Security relies on
the secrecy of the mechanism

Scytale - Permutation

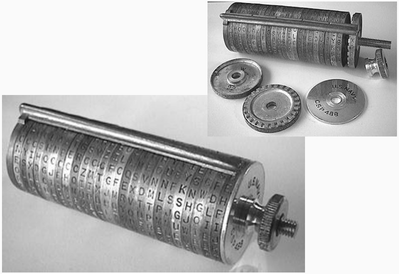
Substitutions and permutations

Security relies on

the secrecy of the mechanism

Alberti's disk

Mono-alphabetical Substitution



Wheel – M 94 (CSP 488)

Poly-alphabetical Substitution

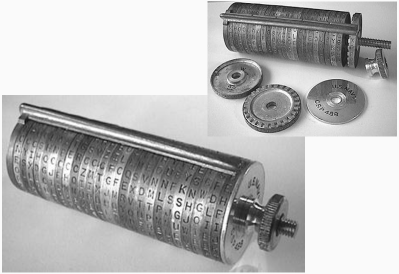
Old Methods

Scytale - Permutation

Substitutions and permutations

Security relies on
the secrecy of the mechanism

Alberti's disk
Mono-alphabetical Substitution



Wheel – M 94 (CSP 488)
Poly-alphabetical Substitution

Outline

Cryptography

Introduction

Kerckhoffs' Principles

Formal Notations

Provable Security

Basic Security Notions

Conclusion

Kerckhoffs' Principles (1)

La Cryptographie Militaire (1883)

Le système doit être matèriellement,
sinon mathématiquement, indéchiffrable

The system should be, if not theoretically unbreakable,
unbreakable in practice

$\longrightarrow$ If the security cannot be formally proven,
heuristics should provide some confidence.

Kerckhoffs' Principles (1)

La Cryptographie Militaire (1883)

Le système doit être matèriellement,
sinon mathématiquement, indéchiffrable

The system should be, if not theoretically unbreakable,
unbreakable in practice

$\longrightarrow$ If the security cannot be formally proven,
heuristics should provide some confidence.

Kerckhoffs' Principles (2)

La Cryptographie Militaire (1883)

Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi

Compromise of the system should not inconvenience the correspondents

$\longrightarrow$ The description of the mechanism should be public

Kerckhoffs' Principles (2)

La Cryptographie Militaire (1883)

Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi

Compromise of the system should not inconvenience the correspondents

$\longrightarrow$ The description of the mechanism should be public

Kerckhoffs' Principles (3)

La Cryptographie Militaire (1883)

La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants

The key should be rememberable without notes and should be easily changeable

$\longrightarrow$ The parameters specific to the users (the key) should be short

Kerckhoffs' Principles (3)

La Cryptographie Militaire (1883)

La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants

The key should be rememberable without notes and should be easily changeable

$\longrightarrow$ The parameters specific to the users (the key) should be short

Use of (Secret) Key

A shared information (secret key) between the sender and the receiver parameterizes the mechanism:

- Vigenère: each key letter tells the shift
- Enigma: connectors and rotors

Use of (Secret) Key

A shared information (secret key) between the sender and the receiver parameterizes the mechanism:

- Vigenère: each key letter tells the shift
- Enigma: connectors and rotors

Use of (Secret) Key

A shared information (secret key) between the sender and the receiver parameterizes the mechanism:

- Vigenère: each key letter tells the shift
- Enigma: connectors and rotors

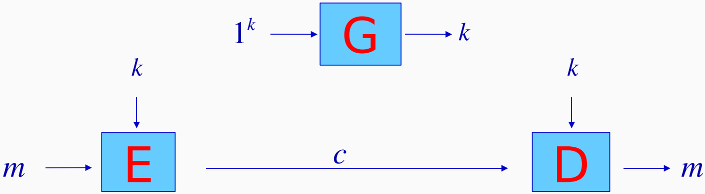
Use of (Secret) Key

A shared information (secret key) between the sender and the receiver parameterizes the mechanism:

- Vigenère: each key letter tells the shift
- Enigma: connectors and rotors

Security **looks** better: but broken (Alan Turing *et al.*)

Principles 2 and 3 define the concepts of symmetric cryptography:



Secrecy

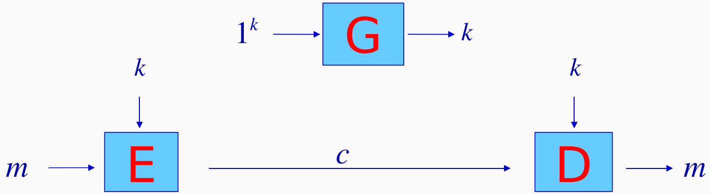
It is impossible/hard to recover *m* from *c* only (without *k*)

Security

It is heuristic only: 1st principle

Symmetric Encryption

Principles 2 and 3 define the concepts of symmetric cryptography:



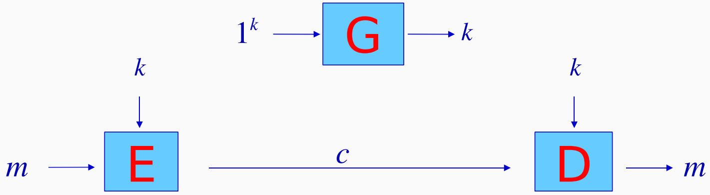
Secrecy

It is impossible/hard to recover *m* from *c* only (without *k*)

Security

It is heuristic only: 1st principle

Principles 2 and 3 define the concepts of symmetric cryptography:



Secrecy

It is impossible/hard to recover m from c only (without k)

Security

It is heuristic only: 1st principle

Perfect Secrecy?

Any security indeed vanished with statistical attacks!

Perfect Secrecy?

Any security indeed vanished with statistical attacks!

Perfect secrecy? Is it possible?

Perfect Secrecy?

Any security indeed vanished with statistical attacks!

Perfect secrecy? Is it possible?

Perfect Secrecy

The ciphertext does not reveal any (additional) information about the plaintext: no more than known before

- a priori information about the plaintext,
  defined by the distribution probability of the plaintext
- a posteriori information about the plaintext,
  defined by the distribution probability of the plaintext,
  given the ciphertext

Both distributions should be perfectly identical

One-Time Pad Encryption

Vernam's Cipher (1929)

- Encryption of $m \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$
\begin{array}{rcl}
m = & \boxed{1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1} & \text{plaintext} \\
 & \oplus & \text{XOR (+ modulo 2)} \\
k = & \boxed{1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0} & \text{key = random mask} \\
 & = & \\
c = & \boxed{0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1} & \text{ciphertext}
\end{array}
$$

- Decryption of $c \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m$$

One-Time Pad Encryption

Vernam's Cipher (1929)

- Encryption of $m \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$m =$	1	0	0	1	0	1	1	plaintext
				$\oplus$				XOR (+ modulo 2)
$k =$	1	1	0	1	0	0	0	key = random mask
				$=$				
$c =$	0	1	0	0	0	1	1	ciphertext

- Decryption of $c \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m$$

One-Time Pad Encryption

Vernam's Cipher (1929)

- Encryption of $m \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$\begin{array}{rcl} m = & \boxed{1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1} & \text{plaintext} \\ & \oplus & \text{XOR (+ modulo 2)} \\ k = & \boxed{1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0} & \text{key = random mask} \\ & = & \\ c = & \boxed{0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1} & \text{ciphertext} \end{array}$$

- Decryption of $c \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m$$

One-Time Pad Encryption

Vernam's Cipher (1929)

- Encryption of $m \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$\begin{array}{rcl} m = & \boxed{1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1} & \text{plaintext} \\ & \oplus & \text{XOR (+ modulo 2)} \\ k = & \boxed{1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0} & \text{key = random mask} \\ & = & \\ c = & \boxed{0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1} & \text{ciphertext} \end{array}$$

- Decryption of $c \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m$$

Which message is encrypted in the ciphertext $c \in \{0,1\}^n$?

One-Time Pad Encryption

Vernam's Cipher (1929)

- Encryption of $m \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

m =	1 0 0 1 0 1 1	plaintext
	$\oplus$	XOR (+ modulo 2)
k =	1 1 0 1 0 0 0	key = random mask
	=	
c =	0 1 0 0 0 1 1	ciphertext

- Decryption of $c \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m$$

Which message is encrypted in the ciphertext $c \in \{0,1\}^n$?

For any candidate $m \in \{0,1\}^n$, the key $k = c \oplus m$ would lead to c

One-Time Pad Encryption

Vernam's Cipher (1929)

- Encryption of $m \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$m =$	1	0	0	1	0	1	1	plaintext
				$\oplus$				XOR (+ modulo 2)
$k =$	1	1	0	1	0	0	0	key = random mask
				$=$				
$c =$	0	1	0	0	0	1	1	ciphertext

- Decryption of $c \in \{0,1\}^n$ under the key $k \in \{0,1\}^n$:

$$c \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m$$

Which message is encrypted in the ciphertext $c \in \{0,1\}^n$?

For any candidate $m \in \{0,1\}^n$, the key $k = c \oplus m$ would lead to c

$\Rightarrow$ no information about m is leaked with c!

Drawbacks

- The key must be as long as the plaintext
- This key must be used once only (one-time pad)

Theorem (Shannon – 1949)

To achieve perfect secrecy, A and B have to share a common string truly random and as long as the whole communication.

Thus, the above one-time pad technique is optimal…

Information Theory

Drawbacks

- The key must be as long as the plaintext
- This key must be used once only (one-time pad)

Theorem (Shannon – 1949)

To achieve perfect secrecy, A and B have to share a common string truly random and as long as the whole communication.

Thus, the above one-time pad technique is optimal…

Information Theory

Drawbacks

- The key must be as long as the plaintext
- This key must be used once only (one-time pad)

Theorem (Shannon – 1949)

To achieve perfect secrecy, A and B have to share a common string truly random and as long as the whole communication.

Thus, the above one-time pad technique is optimal...

Practical Secrecy

Perfect Secrecy vs. Practical Secrecy

- No information about the plaintext m is in the ciphertext c without the knowledge of the key k
  $\Rightarrow$ information theory
  No information about the plaintext m can be extracted from the ciphertext c, even for a powerful adversary (unlimited time and/or unlimited power): perfect secrecy
- In practice: adversaries are limited in time/power
  $\Rightarrow$ complexity theory

Practical Secrecy

Perfect Secrecy vs. Practical Secrecy

- No information about the plaintext m is in the ciphertext c without the knowledge of the key k
  $\Rightarrow$ information theory
  No information about the plaintext m can be extracted from the ciphertext c, even for a powerful adversary (unlimited time and/or unlimited power): perfect secrecy
- In practice: adversaries are limited in time/power
  $\Rightarrow$ complexity theory

Practical Secrecy

Perfect Secrecy vs. Practical Secrecy

- No information about the plaintext m is in the ciphertext c without the knowledge of the key k
  $\Rightarrow$ information theory
  No information about the plaintext m can be extracted from the ciphertext c, even for a powerful adversary (unlimited time and/or unlimited power): perfect secrecy
- In practice: adversaries are limited in time/power
  $\Rightarrow$ complexity theory

Shannon also showed that combining appropriately permutations and substitutions can hide information: extracting information from the ciphertext is time consuming

Modern Symmetric Encryption: DES and AES

Combination of substitutions and permutations

DES (1977)
Data Encryption Standard

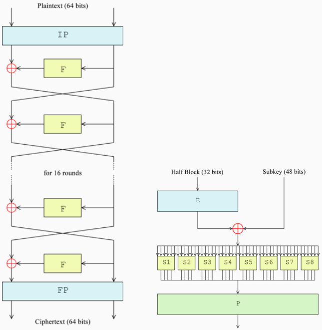
AES (2001)
Advanced Encryption Standard

Combination of substitutions and permutations

DES (1977)
Data Encryption Standard

AES (2001)
Advanced Encryption Standard

Combination of substitutions and permutations



DES (1977)

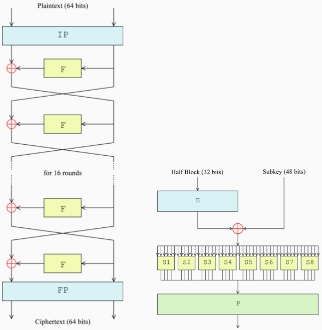
Data Encryption Standard

AES (2001)

Advanced Encryption Standard

Modern Symmetric Encryption: DES and AES

Combination of substitutions and permutations



DES (1977)

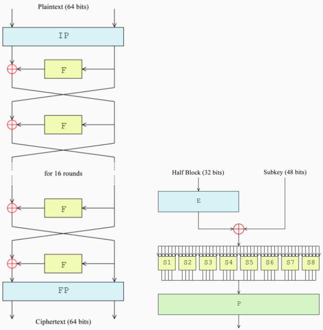
Data Encryption Standard

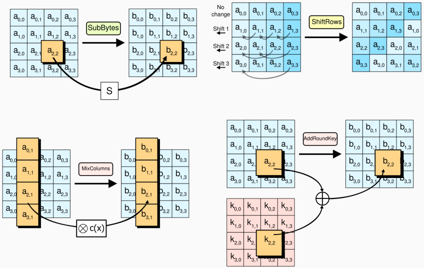
AES (2001)

Advanced Encryption Standard

Modern Symmetric Encryption: DES and AES

Combination of substitutions and permutations





DES (1977)

Data Encryption Standard

AES (2001)

Advanced Encryption Standard

Outline

Cryptography

Introduction

Kerckhoffs' Principles

Formal Notations

Provable Security

Basic Security Notions

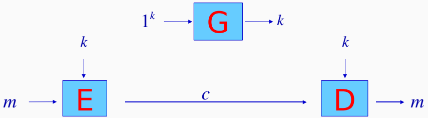
Conclusion

Symmetric Encryption: Formalism

Symmetric Encryption – Secret Key Encryption

One secret key only shared by Alice and Bob: this is a common parameter for the encryption and the decryption algorithms

This secret key has a symmetric capability

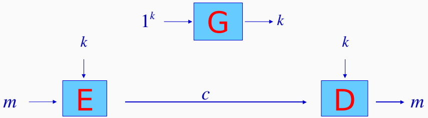


Symmetric Encryption: Formalism

Symmetric Encryption – Secret Key Encryption

One secret key only shared by Alice and Bob: this is a common parameter for the encryption and the decryption algorithms

This secret key has a symmetric capability



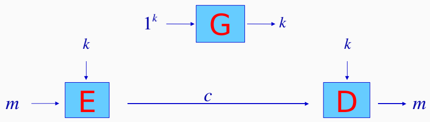
The secrecy of the key *k* guarantees the secrecy of communications but requires such a common secret key!

Symmetric Encryption: Formalism

Symmetric Encryption – Secret Key Encryption

One secret key only shared by Alice and Bob: this is a common parameter for the encryption and the decryption algorithms

This secret key has a symmetric capability



The secrecy of the key k guarantees the secrecy of communications but requires such a common secret key!

How can we establish such a common secret key?

Or, how to avoid it?

Asymmetric Encryption: Intuition

[Diffie-Hellman 1976]

Secrecy

- The recipient only should be able to open the message
- No requirement about the sender

Why would the sender need a secret key to encrypt a message?

Secrecy

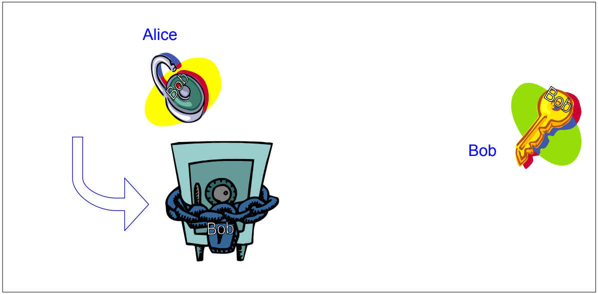
- The recipient only should be able to open the message
- No requirement about the sender

Why would the sender need a secret key to encrypt a message?

Secrecy

- The recipient only should be able to open the message
- No requirement about the sender

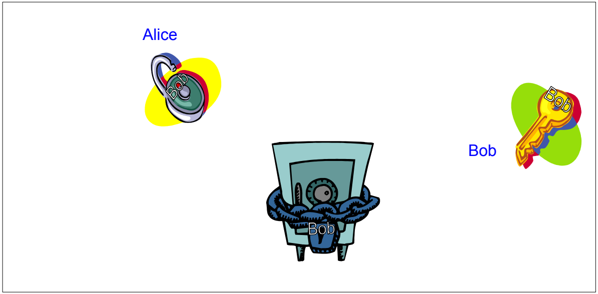
Why would the sender need a secret key to encrypt a message?



Secrecy

- The recipient only should be able to open the message
- No requirement about the sender

Why would the sender need a secret key to encrypt a message?



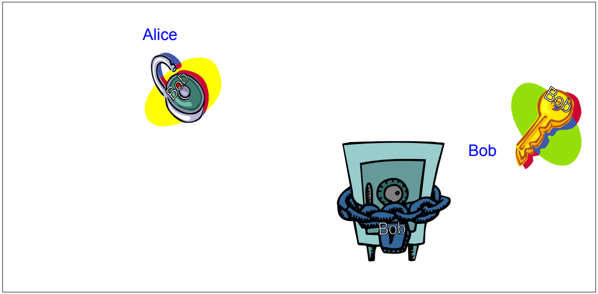
Asymmetric Encryption: Intuition

[Diffie-Hellman 1976]

Secrecy

- The recipient only should be able to open the message
- No requirement about the sender

Why would the sender need a secret key to encrypt a message?



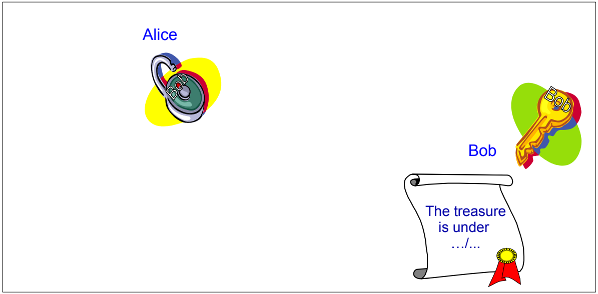
Asymmetric Encryption: Intuition

[Diffie-Hellman 1976]

Secrecy

- The recipient only should be able to open the message
- No requirement about the sender

Why would the sender need a secret key to encrypt a message?

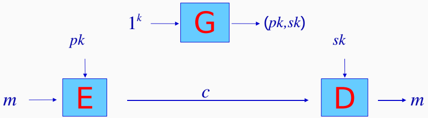


Asymmetric Encryption: Formalism

Public Key Cryptography – Diffie-Hellman (1976)

- Bob's public key is used by Alice as a parameter to encrypt a message to Bob
- Bob's private key is used by Bob as a parameter to decrypt ciphertexts

Asymmetric cryptography extends the 2nd principle:

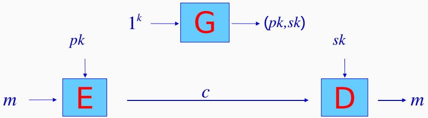


Asymmetric Encryption: Formalism

Public Key Cryptography – Diffie-Hellman (1976)

- Bob's public key is used by Alice as a parameter to encrypt a message to Bob
- Bob's private key is used by Bob as a parameter to decrypt ciphertexts

Asymmetric cryptography extends the 2nd principle:



The secrecy of the private key *sk* guarantees the secrecy of communications

Provable Security

Outline

Cryptography

Provable Security

Definition

Computational Assumptions

Some Reductions

Basic Security Notions

Conclusion

What is a Secure Cryptographic Scheme/Protocol?

- Symmetric encryption:
  The secrecy of the key k guarantees the secrecy of communications
- Asymmetric encryption:
  The secrecy of the private key sk guarantees the secrecy of communications
- What does mean secrecy?
  $\rightarrow$ Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  $\rightarrow$ Provable security

What is a Secure Cryptographic Scheme/Protocol?

- Symmetric encryption:
  The secrecy of the key k guarantees the secrecy of communications
- Asymmetric encryption:
  The secrecy of the private key sk guarantees the secrecy of communications

- What does mean secrecy?
  → Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  → Provable security

What is a Secure Cryptographic Scheme/Protocol?

- Symmetric encryption:
  The secrecy of the key k guarantees the secrecy of communications
- Asymmetric encryption:
  The secrecy of the private key sk guarantees the secrecy of communications
- What does mean secrecy?
  $\rightarrow$ Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  $\rightarrow$ Provable security

What is a Secure Cryptographic Scheme/Protocol?

- Symmetric encryption:
  The secrecy of the key k guarantees the secrecy of communications
- Asymmetric encryption:
  The secrecy of the private key sk guarantees the secrecy of communications
- What does mean secrecy?
  → Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  → Provable security

What is a Secure Cryptographic Scheme/Protocol?

- Symmetric encryption:
  The secrecy of the key k guarantees the secrecy
  of communications
- Asymmetric encryption:
  The secrecy of the private key *sk* guarantees
  the secrecy of communications
- What does mean secrecy?
  $\rightarrow$ Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  $\rightarrow$ Provable security

Provable Security

One can prove that:

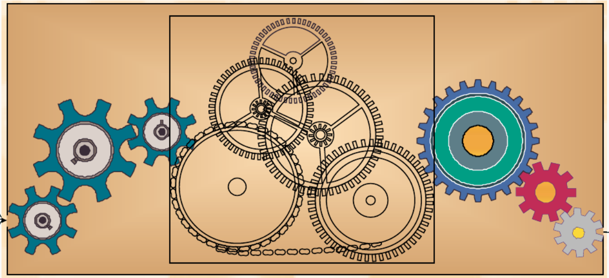
- if an adversary is able to break the cryptographic scheme
- then one can break a well-known hard problem

Provable Security

One can prove that:

- if an adversary is able to break the cryptographic scheme
- then one can break a well-known hard problem

hard instance $\rightarrow$



$\rightarrow$ solution

General Method

Computational Security Proofs

In order to prove the security of a cryptographic scheme/protocol, one needs

- a formal security model (security notions)
- acceptable computational assumptions (hard problems)
- a reduction: if one can break the security notions, then one can break the hard problem

General Method

Computational Security Proofs

In order to prove the security of a cryptographic scheme/protocol, one needs

- a formal security model (security notions)
- acceptable computational assumptions (hard problems)
- a reduction: if one can break the security notions, then one can break the hard problem

General Method

Computational Security Proofs

In order to prove the security of a cryptographic scheme/protocol, one needs

- a formal security model (security notions)
- acceptable computational assumptions (hard problems)
- a reduction: if one can break the security notions, then one can break the hard problem

General Method

Computational Security Proofs

In order to prove the security of a cryptographic scheme/protocol, one needs

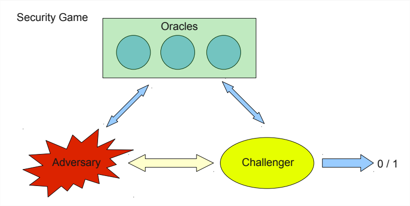
- a formal security model (security notions)
- acceptable computational assumptions (hard problems)
- a reduction: if one can break the security notions,
  then one can break the hard problem

General Method

Computational Security Proofs

In order to prove the security of a cryptographic scheme/protocol, one needs

- a formal security model (security notions)
- acceptable computational assumptions (hard problems)
- a reduction: if one can break the security notions, then one can break the hard problem

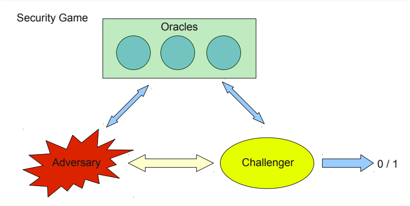


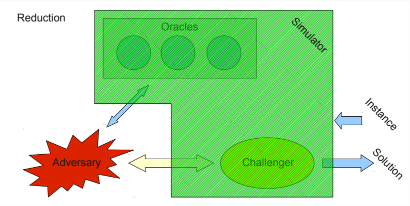
General Method

Computational Security Proofs

In order to prove the security of a cryptographic scheme/protocol, one needs

- a formal security model (security notions)
- acceptable computational assumptions (hard problems)
- a reduction: if one can break the security notions, then one can break the hard problem





Outline

Cryptography

Provable Security

Definition

Computational Assumptions

Some Reductions

Basic Security Notions

Conclusion

Integer Factoring
[Lenstra-Verheul 2000]

Integer Factoring

- Given $n = pq$
- Find p and q

Year	Required Complexity	n bitlength
before 2000	64	768
before 2010	80	1024
before 2020	112	2048
before 2030	128	3072
	192	7680
	256	15360

Note that the reduction may be lossy: extra bits are then required

Integer Factoring Records

Integer Factoring

- Given $n = pq$
- Find p and q

Digits	Date	Details
129	April 1994	Quadratic Sieve
130	April 1996	Algebraic Sieve
140	February 1999	
155	August 1999	512 bits
160	April 2003	
200	May 2005	
232	December 2009	768 bits

Integer Factoring Variants

RSA [Rivest-Shamir-Adleman 1978]

- Given $n = pq$, e and $y \in \mathbb{Z}_n^\star$
- Find x such that $y = x^e \bmod n$

Note that this problem is hard without the prime factors p and q, but becomes easy with them: if $d = e^{-1} \bmod \varphi(n)$, then $x = y^d \bmod n$

Flexible RSA [Baric-Pfitzmann and Fujisaki-Okamoto 1997]

- Given $n = pq$ and $y \in \mathbb{Z}_n^\star$
- Find x and $e > 1$ such that $y = x^e \bmod n$

Both problems are assumed as hard as integer factoring:
the prime factors are a trapdoor to find solutions

Integer Factoring Variants

RSA

[Rivest-Shamir-Adleman 1978]

- Given $n = pq$, e and $y \in \mathbb{Z}_n^\star$
- Find x such that $y = x^e \bmod n$

Note that this problem is hard without the prime factors p and q, but becomes easy with them: if $d = e^{-1} \bmod \varphi(n)$, then $x = y^d \bmod n$

Flexible RSA

[Baric-Pfitzmann and Fujisaki-Okamoto 1997]

- Given $n = pq$ and $y \in \mathbb{Z}_n^\star$
- Find x and $e > 1$ such that $y = x^e \bmod n$

Both problems are assumed as hard as integer factoring:
the prime factors are a trapdoor to find solutions

Integer Factoring Variants

RSA

[Rivest-Shamir-Adleman 1978]

- Given $n = pq$, e and $y \in \mathbb{Z}_n^\star$
- Find x such that $y = x^e \bmod n$

Note that this problem is hard without the prime factors p and q, but becomes easy with them: if $d = e^{-1} \bmod \varphi(n)$, then $x = y^d \bmod n$

Flexible RSA

[Baric-Pfitzmann and Fujisaki-Okamoto 1997]

- Given $n = pq$ and $y \in \mathbb{Z}_n^\star$
- Find x and $e > 1$ such that $y = x^e \bmod n$

Both problems are assumed as hard as integer factoring:
the prime factors are a trapdoor to find solutions

Discrete Logarithm

Discrete Logarithm Problem

- Given $\mathbb{G} = \langle g \rangle$ a cyclic group of order q, and $y \in \mathbb{G}$
- Find x such that $y = g^x$

Possible groups: $\mathbb{G} \in (\mathbb{Z}_p^\star, \times)$, or an elliptic curve

(Computational) Diffie Hellman Problem

- Given $\mathbb{G} = \langle g \rangle$ a cyclic group of order q, and $X = g^x$, $Y = g^y$
- Find $Z = g^{xy}$

The knowledge of x or y helps to solve this problem (trapdoor)

Discrete Logarithm

Discrete Logarithm Problem

- Given $\mathbb{G} = \langle g \rangle$ a cyclic group of order q, and $y \in \mathbb{G}$
- Find x such that $y = g^x$

Possible groups: $\mathbb{G} \in (\mathbb{Z}_p^\star, \times)$, or an elliptic curve

(Computational) Diffie Hellman Problem

- Given $\mathbb{G} = \langle g \rangle$ a cyclic group of order q, and $X = g^x$, $Y = g^y$
- Find $Z = g^{xy}$

The knowledge of x or y helps to solve this problem (trapdoor)

Success Probabilities

For any computational problem P, we quantify the quality of an adversary $\mathcal{A}$ by its success probability in finding the solution:

$$\mathbf{Succ}^P(\mathcal{A}) = \Pr[\mathcal{A}(\text{instance}) \rightarrow \text{solution}].$$

We quantify the hardness of the problem by the success probability of the best adversary within time t: $\mathbf{Succ}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Succ}(\mathcal{A})\}$.

Note that the probability space can be restricted:
some inputs are fixed, and others only are randomly chosen.

Discrete Logarithm Problem

We usually fix the group $\mathbb{G} = \langle g \rangle$ of order q, and the generator g, but x is randomly chosen:

$$\mathbf{Succ}^{\mathbf{dlp}}_{\mathbb{G}}(\mathcal{A}) = \Pr_{x \overset{R}{\leftarrow} \mathbb{Z}_q}[\mathcal{A}(g^x) \rightarrow x].$$

Success Probabilities

For any computational problem P, we quantify the quality of an adversary $\mathcal{A}$ by its success probability in finding the solution:

$$\mathbf{Succ}^{P}(\mathcal{A}) = \Pr[\mathcal{A}(\text{instance}) \rightarrow \text{solution}].$$

We quantify the hardness of the problem by the success probability of the best adversary within time t: $\mathbf{Succ}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Succ}(\mathcal{A})\}$.

Note that the probability space can be restricted:

some inputs are fixed, and others only are randomly chosen.

Discrete Logarithm Problem

We usually fix the group $\mathbb{G} = \langle g \rangle$ of order q, and the generator g, but x is randomly chosen:

$$\mathbf{Succ}^{\mathsf{dlp}}_{\mathbb{G}}(\mathcal{A}) = \Pr_{x \stackrel{R}{\leftarrow} \mathbb{Z}_q}[\mathcal{A}(g^x) \rightarrow x].$$

Success Probabilities

For any computational problem P, we quantify the quality of an adversary $\mathcal{A}$ by its success probability in finding the solution:

$$\mathbf{Succ}^{P}(\mathcal{A}) = \Pr[\mathcal{A}(\text{instance}) \rightarrow \text{solution}].$$

We quantify the hardness of the problem by the success probability of the best adversary within time t: $\mathbf{Succ}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Succ}(\mathcal{A})\}$.

Note that the probability space can be restricted:
some inputs are fixed, and others only are randomly chosen.

Discrete Logarithm Problem

We usually fix the group $\mathbb{G} = \langle g \rangle$ of order q, and the generator g, but x is randomly chosen:

$$\mathbf{Succ}^{\mathbf{dlp}}_{\mathbb{G}}(\mathcal{A}) = \Pr_{x \stackrel{R}{\leftarrow} \mathbb{Z}_q}[\mathcal{A}(g^x) \rightarrow x].$$

Success Probabilities

For any computational problem P, we quantify the quality of an adversary $\mathcal{A}$ by its success probability in finding the solution:

$$\mathbf{Succ}^{P}(\mathcal{A}) = \Pr[\mathcal{A}(\text{instance}) \to \text{solution}].$$

We quantify the hardness of the problem by the success probability of the best adversary within time t: $\mathbf{Succ}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Succ}(\mathcal{A})\}$.

Note that the probability space can be restricted:

some inputs are fixed, and others only are randomly chosen.

Discrete Logarithm Problem

We usually fix the group $\mathbb{G} = \langle g \rangle$ of order q, and the generator g, but x is randomly chosen:

$$\mathbf{Succ}^{\mathsf{dlp}}_{\mathbb{G}}(\mathcal{A}) = \Pr_{x \stackrel{R}{\leftarrow} \mathbb{Z}_q}[\mathcal{A}(g^x) \to x].$$

Decisional Problem

(Decisional) Diffie Hellman Problem

- Given $\mathbb{G} = \langle g \rangle$ a cyclic group of order q, and $X = g^x$, $Y = g^y$, as well as a candidate $Z \in \mathbb{G}$
- Decide whether $Z = g^{xy}$

The adversary is called a distinguisher (outputs 1 bit).
A good distinguisher should behave in significantly different manners according to the input distribution:

$$\begin{aligned}\mathbf{Adv}_{\mathbb{G}}^{\mathsf{ddh}}(\mathcal{A}) = & \Pr[\mathcal{A}(X, Y, Z) = 1 | Z = g^{xy}] \\ & - \Pr[\mathcal{A}(X, Y, Z) = 1 | Z \stackrel{R}{\leftarrow} \mathbb{G}]\end{aligned}$$

Outline

Cryptography

Provable Security

- Definition
- Computational Assumptions
- Some Reductions

Basic Security Notions

Conclusion

CDH $\leq$ DLP

Let $\mathcal{A}$ be an adversary against the **DLP** within time t, then we build an adversary $\mathcal{B}$ against the **CDH**: given X and Y, $\mathcal{B}$ runs $\mathcal{A}$ on X, that outputs x' (correct or not); then $\mathcal{B}$ outputs $Y^{x'}$.

The running time t' of $\mathcal{B}$ is the same as $\mathcal{A}$, plus one exponentiation:

$$\begin{aligned}\mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t') \geq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(\mathcal{B}) &= \Pr[\mathcal{B}(X, Y) \to g^{xy} = Y^x] \\ &= \Pr[\mathcal{A}(X) \to x] = \mathbf{Succ}_{\mathbb{G}}^{\mathbf{dlp}}(\mathcal{A})\end{aligned}$$

Taking the maximum on the adversaries $\mathcal{A}$:

$$\mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t + \tau_{\mathsf{exp}}) \geq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{dlp}}(t)$$

DDH $\leq$ CDH $\leq$ DLP

CDH $\leq$ DLP

Let $\mathcal{A}$ be an adversary against the **DLP** within time t, then we build an adversary $\mathcal{B}$ against the **CDH**: given X and Y, $\mathcal{B}$ runs $\mathcal{A}$ on X, that outputs x' (correct or not); then $\mathcal{B}$ outputs $Y^{x'}$.

The running time t' of $\mathcal{B}$ is the same as $\mathcal{A}$, plus one exponentiation:

$$\begin{aligned} \mathbf{Succ}_{\mathbb{G}}^{\mathsf{cdh}}(t') \geq \mathbf{Succ}_{\mathbb{G}}^{\mathsf{cdh}}(\mathcal{B}) &= \Pr[\mathcal{B}(X,Y) \rightarrow g^{xy} = Y^x] \\ &= \Pr[\mathcal{A}(X) \rightarrow x] = \mathbf{Succ}_{\mathbb{G}}^{\mathsf{dlp}}(\mathcal{A}) \end{aligned}$$

Taking the maximum on the adversaries $\mathcal{A}$:

$$\mathbf{Succ}_{\mathbb{G}}^{\mathsf{cdh}}(t + \tau_{\mathsf{exp}}) \geq \mathbf{Succ}_{\mathbb{G}}^{\mathsf{dlp}}(t)$$

DDH $\leq$ CDH

Let $\mathcal{A}$ be an adversary against the **CDH** within time t, we build an adversary $\mathcal{B}$ against the **DDH**: given X, Y and Z, $\mathcal{B}$ runs $\mathcal{A}$ on (X, Y), that outputs Z'; then $\mathcal{B}$ outputs 1 if $Z' = Z$ and 0 otherwise.

The running time of $\mathcal{B}$ is the same as $\mathcal{A}$: $\mathbf{Adv}_{\mathbb{G}}^{\mathsf{ddh}}(t)$ is greater than

$$\begin{aligned}
\mathbf{Adv}_{\mathbb{G}}^{\mathsf{ddh}}(\mathcal{B}) &= \Pr[\mathcal{B} \to 1 | Z = g^{xy}] - \Pr[\mathcal{B} \to 1 | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \Pr[\mathcal{A}(X, Y) \to Z | Z = g^{xy}] - \Pr[\mathcal{A}(X, Y) \to Z | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \Pr[\mathcal{A}(X, Y) \to g^{xy}] - \Pr[\mathcal{A}(X, Y) \to Z | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \mathbf{Succ}_{\mathbb{G}}^{\mathsf{cdh}}(\mathcal{A}) - 1/q
\end{aligned}$$

Taking the maximum on the adversaries $\mathcal{A}$:

$$\mathbf{Adv}_{\mathbb{G}}^{\mathsf{ddh}}(t) \geq \mathbf{Succ}_{\mathbb{G}}^{\mathsf{cdh}}(t) - 1/q$$

DDH $\leq$ CDH

Let $\mathcal{A}$ be an adversary against the **CDH** within time t, we build an adversary $\mathcal{B}$ against the **DDH**: given X, Y and Z, $\mathcal{B}$ runs $\mathcal{A}$ on (X, Y), that outputs Z'; then $\mathcal{B}$ outputs 1 if $Z' = Z$ and 0 otherwise.

The running time of $\mathcal{B}$ is the same as $\mathcal{A}$: $\mathbf{Adv}_{\mathbb{G}}^{\textbf{ddh}}(t)$ is greater than

$$\begin{aligned}
\mathbf{Adv}_{\mathbb{G}}^{\textbf{ddh}}(\mathcal{B}) &= \Pr[\mathcal{B} \rightarrow 1 | Z = g^{xy}] - \Pr[\mathcal{B} \rightarrow 1 | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \Pr[\mathcal{A}(X, Y) \rightarrow Z | Z = g^{xy}] - \Pr[\mathcal{A}(X, Y) \rightarrow Z | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \Pr[\mathcal{A}(X, Y) \rightarrow g^{xy}] - \Pr[\mathcal{A}(X, Y) \rightarrow Z | Z \stackrel{R}{\leftarrow} \mathbb{G}] \\
&= \mathbf{Succ}_{\mathbb{G}}^{\textbf{cdh}}(\mathcal{A}) - 1/q
\end{aligned}$$

Taking the maximum on the adversaries $\mathcal{A}$:

$$\mathbf{Adv}_{\mathbb{G}}^{\textbf{ddh}}(t) \geq \mathbf{Succ}_{\mathbb{G}}^{\textbf{cdh}}(t) - 1/q$$

Distribution Indistinguishability

Indistinguishabilities

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X:

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are perfectly indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = 0$$

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are statistically indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = \mathrm{negl}()$$

Distribution Indistinguishability

Indistinguishabilities

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X:

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are perfectly indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = 0$$

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are statistically indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = \mathsf{negl}()$$

Distribution Indistinguishability

Indistinguishabilities

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X:

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are perfectly indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = 0$$

- $\mathcal{D}_0$ and $\mathcal{D}_1$ are statistically indistinguishable if

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1) = \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_1}[a = x] - \Pr_{a \in \mathcal{D}_0}[a = x] \right| = \mathrm{negl}()$$

Distribution Indistinguishability

Computational Indistinguishability

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X,

- a distinguisher $\mathcal{A}$ between $\mathcal{D}_0$ and $\mathcal{D}_1$ is characterized by its advantage

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) = \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a) = 1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a) = 1]$$

- the computational indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_1$ is measured by

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(t) = \max_{|\mathcal{A}|\leq t}\{\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A})\}$$

Distribution Indistinguishability

Computational Indistinguishability

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X,

- a distinguisher $\mathcal{A}$ between $\mathcal{D}_0$ and $\mathcal{D}_1$ is characterized by its advantage

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) = \Pr_{a \in \mathcal{D}_1}[\mathcal{A}(a) = 1] - \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(a) = 1]$$

- the computational indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_1$ is measured by

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(t) = \max_{|\mathcal{A}| \leq t}\{\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A})\}$$

Distribution Indistinguishability

Computational Indistinguishability

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X,

- a distinguisher $\mathcal{A}$ between $\mathcal{D}_0$ and $\mathcal{D}_1$ is characterized by its advantage

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) = \Pr_{a \in \mathcal{D}_1}[\mathcal{A}(a) = 1] - \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(a) = 1]$$

- the computational indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_1$ is measured by

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(t) = \max_{|\mathcal{A}| \leq t} \{\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A})\}$$

Computational Indistinguishability

$$\begin{aligned}\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a \in \mathcal{D}_1}[\mathcal{A}(a) = 1] - \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(a) = 1] \\ &= \Pr[a \in \mathcal{D}_1 : \mathcal{A}(a) = 1] - \Pr[a \in \mathcal{D}_0 : \mathcal{A}(a) = 1]\end{aligned}$$

Computational Indistinguishability

$$
\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a \in \mathcal{D}_1}[\mathcal{A}(a) = 1] - \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(a) = 1] \\
&= \Pr[a \in \mathcal{D}_1 : \mathcal{A}(a) = 1] - \Pr[a \in \mathcal{D}_0 : \mathcal{A}(a) = 1] \\
&= \Pr[b \leftarrow 1; a \in \mathcal{D}_b : \mathcal{A}(a) = 1] \\
&\quad - \Pr[b \leftarrow 0; a \in \mathcal{D}_b : \mathcal{A}(a) = 1]
\end{aligned}
$$

Computational Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= \Pr[b\leftarrow 1; a\in\mathcal{D}_b : \mathcal{A}(a)=1] \\
&\quad - \Pr[b\leftarrow 0; a\in\mathcal{D}_b : \mathcal{A}(a)=1] \\
&= \Pr[b\leftarrow 1; a\in\mathcal{D}_b : \mathcal{A}(a)=1] \\
&\quad -1 + \Pr[b\leftarrow 0; a\in\mathcal{D}_b : \mathcal{A}(a)=0]
\end{aligned}$$

Computational Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= \Pr[b\leftarrow 1; a\in\mathcal{D}_b : \mathcal{A}(a)=1] \\
&\quad -1 + \Pr[b\leftarrow 0; a\in\mathcal{D}_b : \mathcal{A}(a)=0]
\end{aligned}$$

Computational Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= \Pr[b\leftarrow 1; a\in\mathcal{D}_b : \mathcal{A}(a)=1] \\
&\quad -1 + \Pr[b\leftarrow 0; a\in\mathcal{D}_b : \mathcal{A}(a)=0] \\
&= \Pr[b\leftarrow 1; a\in\mathcal{D}_b : \mathcal{A}(a)=b] \\
&\quad + \Pr[b\leftarrow 0; a\in\mathcal{D}_b : \mathcal{A}(a)=b] - 1
\end{aligned}$$

Computational Indistinguishability

$$
\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= \Pr[b\leftarrow 1; a\in\mathcal{D}_b : \mathcal{A}(a)=b] \\
&\quad + \Pr[b\leftarrow 0; a\in\mathcal{D}_b : \mathcal{A}(a)=b] - 1
\end{aligned}
$$

Computational Indistinguishability

$$
\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= \Pr[b\leftarrow 1; a\in\mathcal{D}_b : \mathcal{A}(a)=b] \\
&\quad + \Pr[b\leftarrow 0; a\in\mathcal{D}_b : \mathcal{A}(a)=b] - 1 \\
&= \Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=1]/\Pr[b=1] \\
&\quad + \Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=0]/\Pr[b=0] - 1
\end{aligned}
$$

Computational Indistinguishability

$$
\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=1]/\Pr[b=1] \\
&\quad + \Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=0]/\Pr[b=0] - 1
\end{aligned}
$$

Computational Indistinguishability

$$
\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=1]/\Pr[b=1] \\
&\quad + \Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=0]/\Pr[b=0] - 1 \\
&= 2\times\Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=1] \\
&\quad + 2\times\Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=0] - 1
\end{aligned}
$$

Computational Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= 2\times\Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=1] \\
&\quad + 2\times\Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=0] - 1
\end{aligned}$$

Computational Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\
&= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1] \\
&= 2\times\Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=1] \\
&\quad + 2\times\Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b \wedge b=0] - 1 \\
&= 2\times\Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b] - 1
\end{aligned}$$

Computational Indistinguishability

$$\begin{aligned}\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a \in \mathcal{D}_1}[\mathcal{A}(a) = 1] - \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(a) = 1] \\ &= \Pr[a \in \mathcal{D}_1 : \mathcal{A}(a) = 1] - \Pr[a \in \mathcal{D}_0 : \mathcal{A}(a) = 1] \\ &= 2 \times \Pr[a \in \mathcal{D}_b : \mathcal{A}(a) = b] - 1\end{aligned}$$

Computational Indistinguishability

$$\begin{aligned}\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] \\ &= \Pr[a\in\mathcal{D}_1 : \mathcal{A}(a)=1] - \Pr[a\in\mathcal{D}_0 : \mathcal{A}(a)=1]\end{aligned}$$

Equivalent Notation

Let $\mathcal{D}_0$ and $\mathcal{D}_1$, two distributions on a finite set X,

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) = 2 \times \Pr[a\in\mathcal{D}_b : \mathcal{A}(a)=b] - 1$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) = \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1]$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1] - \Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1] \\
&= \sum_{x\in X} \begin{pmatrix} \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1 \wedge a=x] \\ -\Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1 \wedge a=x] \end{pmatrix}
\end{aligned}$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) \;=\; \sum_{x\in X}\begin{pmatrix} \Pr_{a\in\mathcal{D}_0}[\mathcal{A}(a)=1 \wedge a=x] \\ -\Pr_{a\in\mathcal{D}_1}[\mathcal{A}(a)=1 \wedge a=x]\end{pmatrix}$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \sum_{x \in X} \begin{pmatrix} \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(a) = 1 \wedge a = x] \\ -\Pr_{a \in \mathcal{D}_1}[\mathcal{A}(a) = 1 \wedge a = x] \end{pmatrix} \\
&= \sum_{x \in X} \begin{pmatrix} \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(x) = 1 \wedge a = x] \\ -\Pr_{a \in \mathcal{D}_1}[\mathcal{A}(x) = 1 \wedge a = x] \end{pmatrix}
\end{aligned}$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) \;=\; \sum_{x \in X} \begin{pmatrix} \Pr_{a \in \mathcal{D}_0}[\mathcal{A}(x) = 1 \wedge a = x] \\ - \Pr_{a \in \mathcal{D}_1}[\mathcal{A}(x) = 1 \wedge a = x] \end{pmatrix}$$

For any distinguisher $\mathcal{A}$, we have

$$\begin{aligned}\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \sum_{x\in X}\begin{pmatrix}\Pr_{a\in\mathcal{D}_0}[\mathcal{A}(x)=1\wedge a=x]\\ -\Pr_{a\in\mathcal{D}_1}[\mathcal{A}(x)=1\wedge a=x]\end{pmatrix}\\ &= \sum_{x\in X}\Pr[\mathcal{A}(x)=1]\times\begin{pmatrix}\Pr_{a\in\mathcal{D}_0}[a=x]\\ -\Pr_{a\in\mathcal{D}_1}[a=x]\end{pmatrix}\end{aligned}$$

$a=x$ and $\mathcal{A}(x)=1$ are independent events

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) = \sum_{x \in X} \Pr[\mathcal{A}(x) = 1] \times \begin{pmatrix} \Pr_{a \in \mathcal{D}_0}[a = x] \\ -\Pr_{a \in \mathcal{D}_1}[a = x] \end{pmatrix}$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &= \sum_{x \in X} \Pr[\mathcal{A}(x) = 1] \times \begin{pmatrix} \Pr_{a \in \mathcal{D}_0}[a = x] \\ - \Pr_{a \in \mathcal{D}_1}[a = x] \end{pmatrix} \\
&\leq \sum_{x \in X} |\Pr[\mathcal{A}(x) = 1]| \times \begin{vmatrix} \Pr_{a \in \mathcal{D}_0}[a = x] \\ - \Pr_{a \in \mathcal{D}_1}[a = x] \end{vmatrix}
\end{aligned}$$

A better analysis could be done here

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) \leq \sum_{x \in X} |\Pr[\mathcal{A}(x) = 1]| \times \left| \begin{array}{c} \Pr_{a \in \mathcal{D}_0}[a = x] \\ -\Pr_{a \in \mathcal{D}_1}[a = x] \end{array} \right|$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &\leq \sum_{x \in X} |\Pr[\mathcal{A}(x) = 1]| \times \left| \begin{array}{c} \Pr_{a \in \mathcal{D}_0}[a = x] \\ - \Pr_{a \in \mathcal{D}_1}[a = x] \end{array} \right| \\
&\leq \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_0}[a = x] - \Pr_{a \in \mathcal{D}_1}[a = x] \right|
\end{aligned}$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) \;\leq\; \sum_{x \in X} \left| \Pr_{a \in \mathcal{D}_0}[a = x] - \Pr_{a \in \mathcal{D}_1}[a = x] \right|$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\begin{aligned}\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) &\leq \sum_{x\in X} \left| \Pr_{a\in\mathcal{D}_0}[a = x] - \Pr_{a\in\mathcal{D}_1}[a = x] \right| \\ &\leq \mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_1)\end{aligned}$$

Relations between Indistinguishability Notions

For any distinguisher $\mathcal{A}$, we have

$$\mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(\mathcal{A}) \leq \textbf{Dist}(\mathcal{D}_0,\mathcal{D}_1)$$

Theorem

$\textbf{Dist}(\mathcal{D}_0,\mathcal{D}_1)$ *is the best advantage any adversary could get, even within an unbounded time.*

$$\forall t, \quad \mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(t) \leq \textbf{Dist}(\mathcal{D}_0,\mathcal{D}_1).$$

With a better analysis, we can even get

$$\forall t, \quad \mathbf{Adv}^{\mathcal{D}_0,\mathcal{D}_1}(t) \leq \frac{1}{2} \cdot \textbf{Dist}(\mathcal{D}_0,\mathcal{D}_1).$$

Hybrid Technique

Let us consider the distributions $\mathcal{D}_A$ and $\mathcal{D}_B$:

$$\mathcal{D}_A = (g^x, g^{y_1}, g^{xy_1}, \ldots, g^{y_n}, g^{xy_n}) \subseteq \mathbb{G}^{2n+1}$$
$$\mathcal{D}_B = (g^x, g^{y_1}, g^{z_1}, \ldots, g^{y_n}, g^{z_n}) \subseteq \mathbb{G}^{2n+1}$$

$$\mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(t)?$$

We define the hybrid distribution

$$\mathcal{D}_i = (g^x, g^{y_1}, g^{xy_1}, \ldots, g^{y_i}, g^{xy_i}, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

$$\mathcal{D}_0 = \mathcal{D}_B \qquad \mathcal{D}_n = \mathcal{D}_A.$$

Hybrid Technique

Let us consider the distributions $\mathcal{D}_A$ and $\mathcal{D}_B$:

$$\mathcal{D}_A = (g^x, g^{y_1}, g^{xy_1}, \ldots, g^{y_n}, g^{xy_n}) \subseteq \mathbb{G}^{2n+1}$$
$$\mathcal{D}_B = (g^x, g^{y_1}, g^{z_1}, \ldots, g^{y_n}, g^{z_n}) \subseteq \mathbb{G}^{2n+1}$$

$$\mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(t)?$$

We define the hybrid distribution

$$\mathcal{D}_i = (g^x, g^{y_1}, g^{xy_1}, \ldots, g^{y_i}, g^{xy_i}, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

$$\mathcal{D}_0 = \mathcal{D}_B \qquad \mathcal{D}_n = \mathcal{D}_A.$$

Hybrid Technique

Let us consider the distributions $\mathcal{D}_A$ and $\mathcal{D}_B$:

$$\mathcal{D}_A = (g^x, g^{y_1}, g^{xy_1}, \ldots, g^{y_n}, g^{xy_n}) \subseteq \mathbb{G}^{2n+1}$$
$$\mathcal{D}_B = (g^x, g^{y_1}, g^{z_1}, \ldots, g^{y_n}, g^{z_n}) \subseteq \mathbb{G}^{2n+1}$$

$$\mathbf{Adv}^{\mathcal{D}_A,\mathcal{D}_B}(t)?$$

We define the hybrid distribution

$$\mathcal{D}_i = (g^x, g^{y_1}, g^{xy_1}, \ldots, g^{y_i}, g^{xy_i}, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

$$\mathcal{D}_0 = \mathcal{D}_B \qquad \mathcal{D}_n = \mathcal{D}_A.$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \xleftarrow{R} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) \quad = \quad \mathbf{Adv}^{\mathcal{D}_n, \mathcal{D}_0}(\mathcal{A})$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}\text{, then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}\text{, then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i,\mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\begin{aligned} \mathbf{Adv}^{\mathcal{D}_A,\mathcal{D}_B}(\mathcal{A}) &= \mathbf{Adv}^{\mathcal{D}_n,\mathcal{D}_0}(\mathcal{A}) \\ &= \left| \Pr_{\mathcal{D}_0}[\mathcal{A} \to 1] - \Pr_{\mathcal{D}_n}[\mathcal{A} \to 1] \right| \end{aligned}$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathbb{G}}^{\mathsf{ddh}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) = \left| \Pr_{\mathcal{D}_0}[\mathcal{A} \to 1] - \Pr_{\mathcal{D}_n}[\mathcal{A} \to 1] \right|$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}\text{, then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}\text{, then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\begin{aligned} \mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) &= \left| \Pr_{\mathcal{D}_0}[\mathcal{A} \to 1] - \Pr_{\mathcal{D}_n}[\mathcal{A} \to 1] \right| \\ &= \sum_{i=1}^{n} \left| \Pr_{\mathcal{D}_{i-1}}[\mathcal{A} \to 1] - \Pr_{\mathcal{D}_i}[\mathcal{A} \to 1] \right| \end{aligned}$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet\ \text{if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet\ \text{if } Z \stackrel{R}{\leftarrow} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathrm{exp}} \end{array}$$

$$\mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) = \sum_{i=1}^{n} \left| \Pr_{\mathcal{D}_{i-1}}[\mathcal{A} \to 1] - \Pr_{\mathcal{D}_i}[\mathcal{A} \to 1] \right|$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet\ \text{if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet\ \text{if } Z \xleftarrow{R} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i,\mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\begin{aligned} \mathbf{Adv}^{\mathcal{D}_A,\mathcal{D}_B}(\mathcal{A}) &= \sum_{i=1}^{n} \left| \Pr_{\mathcal{D}_{i-1}}[\mathcal{A} \to 1] - \Pr_{\mathcal{D}_i}[\mathcal{A} \to 1] \right| \\ &= \sum_{i=1}^{n} \mathbf{Adv}^{\mathcal{D}_i,\mathcal{D}_{i-1}}(\mathcal{A}) \end{aligned}$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i,\mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathrm{exp}} \end{array}$$

$$\mathbf{Adv}^{\mathcal{D}_A,\mathcal{D}_B}(\mathcal{A}) = \sum_{i=1}^{n} \mathbf{Adv}^{\mathcal{D}_i,\mathcal{D}_{i-1}}(\mathcal{A})$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\begin{aligned} \mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) &= \sum_{i=1}^{n} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \\ &\leq \sum_{i=1}^{n} \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t') \end{aligned}$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i,\mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\mathbf{Adv}^{\mathcal{D}_A,\mathcal{D}_B}(\mathcal{A}) \leq \sum_{i=1}^{n} \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t')$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}\text{, then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}\text{, then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\begin{aligned} \mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) &\leq \sum_{i=1}^{n} \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t') \\ &\leq n \times \mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}(t') \end{aligned}$$

Hybrid Technique

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\mathsf{exp}} \end{array}$$

$$\mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) \quad \leq \quad n \times \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t')$$

Let $\mathcal{A}$ be an adversary within time t, against $\mathcal{D}_A$ vs. $\mathcal{D}_B$.

Given a **DDH** input (X, Y, Z), we generate the hybrid instance:

$$\mathcal{I}_i = (X, g^{y_1}, X^{y_1}, \ldots, g^{y_{i-1}}, X^{y_{i-1}}, Y, Z, g^{y_{i+1}}, g^{z_{i+1}}, \ldots, g^{y_n}, g^{z_n})$$

Note that

$$\left.\begin{array}{l} \bullet \text{ if } Z = g^{xy}, \text{ then } \mathcal{I} \in \mathcal{D}_i \\ \bullet \text{ if } Z \stackrel{R}{\leftarrow} \mathbb{G}, \text{ then } \mathcal{I} \in \mathcal{D}_{i-1} \end{array}\right\} \quad \begin{array}{l} \mathbf{Adv}^{\mathcal{D}_i, \mathcal{D}_{i-1}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathbb{G}}^{\textbf{ddh}}(t') \\ \text{where } t' \leq t + 2(n-1)\tau_{\text{exp}} \end{array}$$

$$\mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(\mathcal{A}) \quad \leq \quad n \times \mathbf{Adv}_{\mathbb{G}}^{\textbf{ddh}}(t')$$

Theorem

$$\forall t, \qquad \mathbf{Adv}^{\mathcal{D}_A, \mathcal{D}_B}(t) \leq n \times \mathbf{Adv}_{\mathbb{G}}^{\textbf{ddh}}(t + 2(n-1)\tau_{\text{exp}})$$

Basic Security Notions

Outline

Cryptography

Provable Security

Basic Security Notions

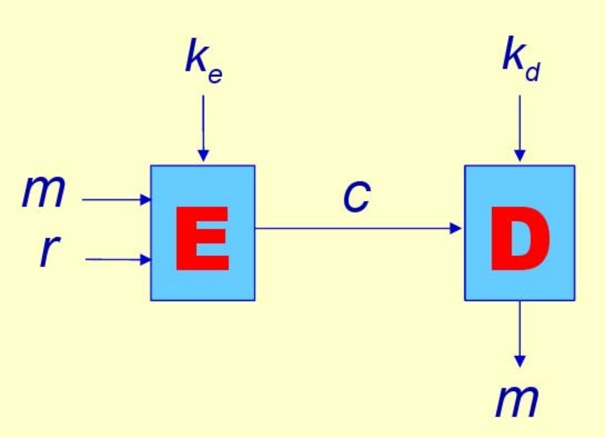
Public-Key Encryption

Variants of Indistinguishability

Signatures

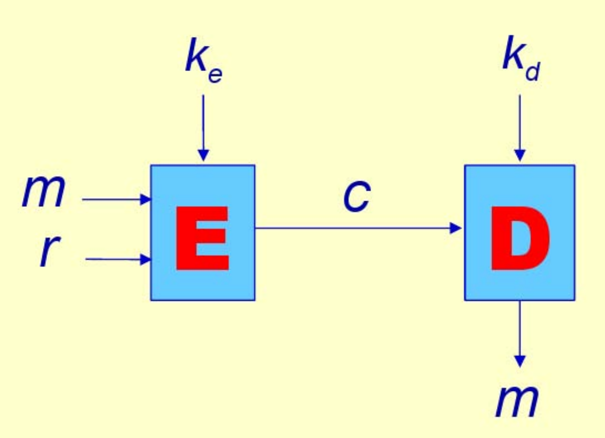
Conclusion

Public-Key Encryption



Goal: Privacy/Secrecy of the plaintext

Public-Key Encryption



Goal: Privacy/Secrecy of the plaintext

OW − CPA

One-Wayness

For a public-key encryption scheme $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, without the secrete key *sk*, it should be computationally impossible to recover the plaintext *m* from the ciphertext *c*:

$$\mathbf{Succ}_{\mathcal{S}}^{\mathsf{ow}}(\mathcal{A}) = \Pr[(sk, pk) \leftarrow \mathcal{K}(); m \stackrel{R}{\leftarrow} \mathcal{M}; c = \mathcal{E}_{pk}(m) : \mathcal{A}(pk, c) \rightarrow m]$$

should be negligible.

Chosen-Plaintext Attacks

In the public-key setting, the adversary has access to the encryption key (the public key), and thus can encrypt any plaintext of its choice: chosen-plaintext attack

OW − CPA

One-Wayness

For a public-key encryption scheme $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, without the secrete key *sk*, it should be computationally impossible to recover the plaintext *m* from the ciphertext *c*:

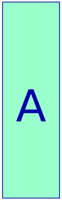
$$\mathbf{Succ}_{\mathcal{S}}^{\mathsf{ow}}(\mathcal{A}) = \Pr[(sk, pk) \leftarrow \mathcal{K}(); m \stackrel{R}{\leftarrow} \mathcal{M}; c = \mathcal{E}_{pk}(m) : \mathcal{A}(pk, c) \to m]$$

should be negligible.

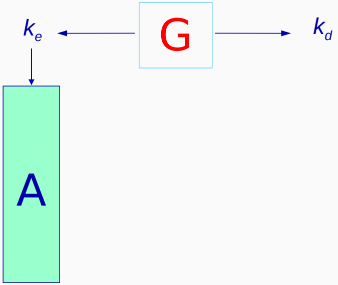
Chosen-Plaintext Attacks

In the public-key setting, the adversary has access to the encryption key (the public key), and thus can encrypt any plaintext of its choice: chosen-plaintext attack

$\mathbf{OW} - \mathbf{CPA}$ Security Game

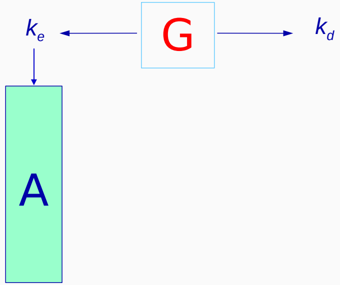


$\mathrm{OW}-\mathrm{CPA}$ Security Game



OW − CPA Security Game

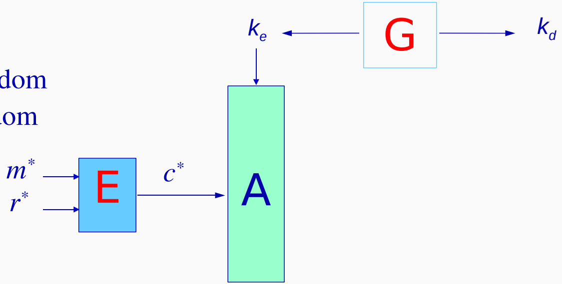
m^* random
r^* random



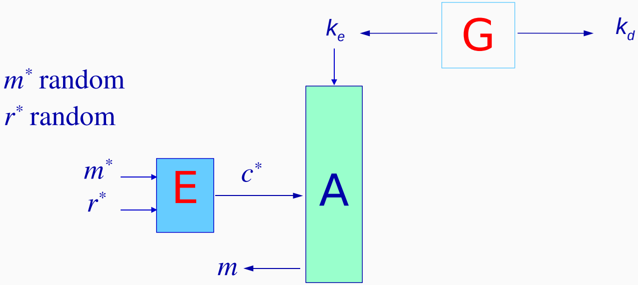
$\mathrm{OW} - \mathrm{CPA}$ Security Game

m^* random

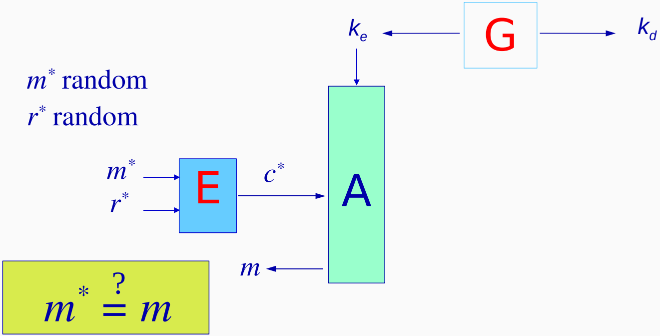
r^* random



$\mathrm{OW} - \mathrm{CPA}$ Security Game



OW − CPA Security Game



ElGamal Encryption

The ElGamal encryption scheme $\mathcal{EG}$ is defined,
in a group $\mathbb{G} = \langle g \rangle$ of order q

- $\mathcal{K}(\mathbb{G}, g, q)$: $x \stackrel{R}{\leftarrow} \mathbb{Z}_q$, and $sk \leftarrow x$ and $pk \leftarrow y = g^x$
- $\mathcal{E}_{pk}(m)$: $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$, $c_1 \leftarrow g^r$ and $c_2 \leftarrow y^r \times m = pk^r \times m$.
  Then, the ciphertext is $c = (c_1, c_2)$
- $\mathcal{D}_{sk}(c)$ outputs $c_2/c_1^x = c_2/c_1^{sk}$

Theorem (ElGamal is $\mathrm{OW-CPA}$)

$$\mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(t) \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$$

ElGamal Encryption

The ElGamal encryption scheme $\mathcal{EG}$ is defined,
in a group $\mathbb{G} = \langle g \rangle$ of order q

- $\mathcal{K}(\mathbb{G}, g, q)$: $x \stackrel{R}{\leftarrow} \mathbb{Z}_q$, and $sk \leftarrow x$ and $pk \leftarrow y = g^x$
- $\mathcal{E}_{pk}(m)$: $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$, $c_1 \leftarrow g^r$ and $c_2 \leftarrow y^r \times m = pk^r \times m$.
  Then, the ciphertext is $c = (c_1, c_2)$
- $\mathcal{D}_{sk}(c)$ outputs $c_2/c_1^x = c_2/c_1^{sk}$

Theorem (ElGamal is $\mathbf{OW-CPA}$)

$$\mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(t) \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$$

ElGamal is $\mathbf{OW}-\mathbf{CPA}$: Proof

$$\mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(t) \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$

ElGamal is $\mathrm{OW}-\mathrm{CPA}$: Proof

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(t) \leq \mathbf{Succ}^{\mathbf{cdh}}_{\mathbb{G}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$

ElGamal is $\mathrm{OW}-\mathrm{CPA}$: Proof

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(t) \leq \mathbf{Succ}^{\mathbf{cdh}}_{\mathbb{G}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $m^* \stackrel{R}{\leftarrow} \mathcal{M}$, sets $c_2 \leftarrow y^{r^*} \times m^*$ and sends $c = (c_1, c_2)$

ElGamal is $\mathbf{OW-CPA}$: Proof

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(t) \leq \mathbf{Succ}^{\mathbf{cdh}}_{\mathbb{G}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $m^* \stackrel{R}{\leftarrow} \mathcal{M}$, sets $c_2 \leftarrow y^{r^*} \times m^*$ and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs m

ElGamal is $\mathbf{OW}-\mathbf{CPA}$: Proof

$$\mathbf{Succ}^{\text{ow}-\text{cpa}}_{\mathcal{EG}}(t) \leq \mathbf{Succ}^{\textbf{cdh}}_{\mathbb{G}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $m^* \stackrel{R}{\leftarrow} \mathcal{M}$, sets $c_2 \leftarrow y^{r^*} \times m^*$ and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs m
- $\Pr[m = m^*] = \mathbf{Succ}^{\text{ow}-\text{cpa}}_{\mathcal{EG}}(\mathcal{A})$

$$\mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(t) \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $m^* \stackrel{R}{\leftarrow} \mathcal{M}$, sets $c_2 \leftarrow y^{r^*} \times m^*$ and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs m
- $\Pr[m = m^*] = \mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(\mathcal{A})$

ElGamal is $\mathbf{OW-CPA}$: Proof

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(t) \leq \mathbf{Succ}^{\mathbf{cdh}}_{\mathbb{G}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- The challenger chooses $m^* \stackrel{R}{\leftarrow} \mathcal{M}$, sets $c_2 \leftarrow y^{r^*} \times m^*$ and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs m
- $\Pr[m = m^*] = \mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(\mathcal{A})$

ElGamal is $\mathbf{OW}-\mathbf{CPA}$: Proof

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(t) \leq \mathbf{Succ}^{\mathbf{cdh}}_{\mathbb{G}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $c_2 \stackrel{R}{\leftarrow} \mathbb{G}$ (which virtually sets $m^* \leftarrow c_2/\mathbf{CDH}(X, Y)$), and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs m
- $\Pr[m = m^*] = \mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(\mathcal{A})$

ElGamal is $\mathrm{OW}-\mathrm{CPA}$: Proof

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(t) \leq \mathbf{Succ}^{\mathbf{cdh}}_{\mathbb{G}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $c_2 \stackrel{R}{\leftarrow} \mathbb{G}$ (which virtually sets $m^* \leftarrow c_2/\mathbf{CDH}(X, Y)$), and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives m from $\mathcal{A}$ and outputs c_2/m
- $\Pr[m = m^*] = \mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{EG}}(\mathcal{A})$

ElGamal is $\mathrm{OW}-\mathrm{CPA}$: Proof

$$\mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(t) \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$$

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **CDH**: let us be given a **CDH** instance (X, Y)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $c_2 \stackrel{R}{\leftarrow} \mathbb{G}$ (which virtually sets $m^* \leftarrow c_2/\mathbf{CDH}(X, Y)$), and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives m from $\mathcal{A}$ and outputs c_2/m
- $\Pr[m = m^*] = \mathbf{Succ}_{\mathcal{EG}}^{\mathsf{ow-cpa}}(\mathcal{A})$
  $= \Pr[c_2/m = c_2/m^*] = \Pr[c_2/m = \mathbf{CDH}(X, Y)] \leq \mathbf{Succ}_{\mathbb{G}}^{\mathbf{cdh}}(t)$

Is $\mathrm{OW}-\mathrm{CPA}$ Enough?

For a yes/no answer or sell/buy order,
one bit of information may be enough for the adversary!

How to model that no bit of information leaks?

Semantic Security [Goldwasser-Micali 1984]

For any predicate f, $\mathcal{E}(m)$ does not help to guess $f(m)$, with better probability than $f(m')$ (for a random but private m'): in the game

$$(sk, pk) \leftarrow \mathcal{K}(); (\mathcal{M}, f, \text{state}) \leftarrow \mathcal{A}(pk);$$
$$m, m' \stackrel{R}{\leftarrow} \mathcal{M}; c = \mathcal{E}_{pk}(m); p \leftarrow \mathcal{A}(\text{state}, c)$$

then,

$$\mathbf{Adv}^{\text{sem}}_{\mathcal{S}}(\mathcal{A}) = \left|\Pr[p = f(m)] - \Pr[p = f(m')]\right|.$$

Is $\mathrm{OW} - \mathrm{CPA}$ Enough?

For a yes/no answer or sell/buy order,
one bit of information may be enough for the adversary!

How to model that no bit of information leaks?

Semantic Security [Goldwasser-Micali 1984]

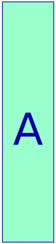
For any predicate f, $\mathcal{E}(m)$ does not help to guess $f(m)$, with better probability than $f(m')$ (for a random but private m'): in the game

$$(sk, pk) \leftarrow \mathcal{K}(); (\mathcal{M}, f, \text{state}) \leftarrow \mathcal{A}(pk);$$
$$m, m' \stackrel{R}{\leftarrow} \mathcal{M}; c = \mathcal{E}_{pk}(m); p \leftarrow \mathcal{A}(\text{state}, c)$$

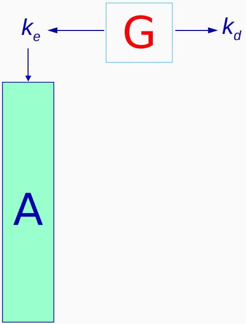
then,

$$\mathbf{Adv}_{\mathcal{S}}^{\text{sem}}(\mathcal{A}) = \left|\Pr[p = f(m)] - \Pr[p = f(m')]\right|.$$

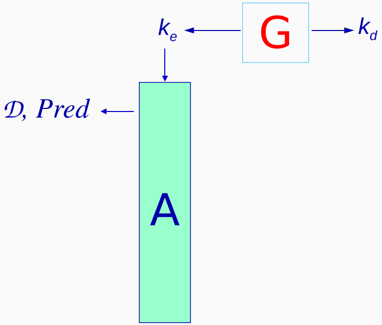
Semantic Security



Semantic Security

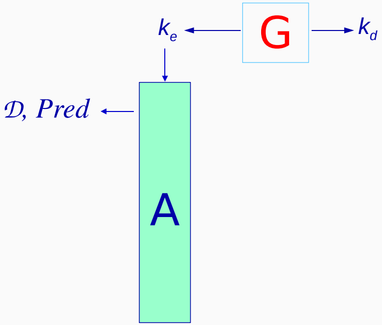


Semantic Security



$m^*, m' \leftarrow \mathcal{D}$

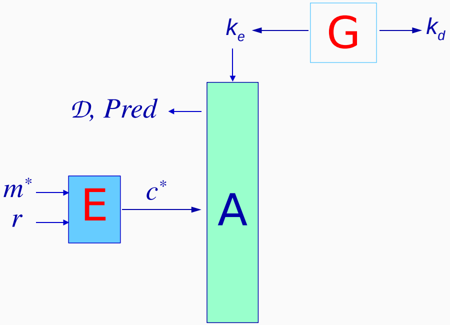
r random



Semantic Security

$m^*, m' \leftarrow \mathcal{D}$

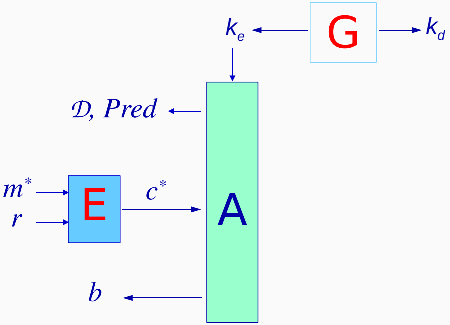
r random



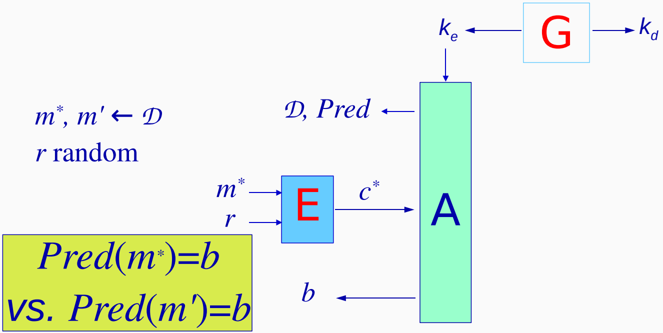
Semantic Security

$m^*, m' \leftarrow \mathcal{D}$

r random



Semantic Security



Another equivalent formulation (if efficiently computable predicate):

IND − CPA

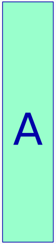
After having chosen two plaintexts m_0 and m_1, upon receiving the encryption of m_b (for a random bit b), it should be hard to guess which message has been encrypted: in the game

$$(sk, pk) \leftarrow \mathcal{K}(); (m_0, m_1, \text{state}) \leftarrow \mathcal{A}(pk);$$
$$b \stackrel{R}{\leftarrow} \{0,1\}; c = \mathcal{E}_{pk}(m_b); b' \leftarrow \mathcal{A}(\text{state}, c)$$

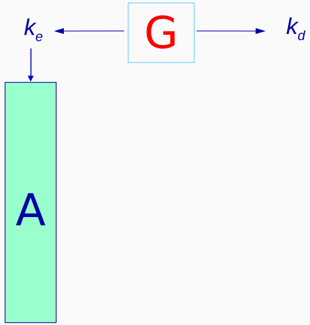
then,

$$\begin{aligned} \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{S}}(\mathcal{A}) &= \left|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]\right| \\ &= \left|2 \times \Pr[b' = b] - 1\right| \end{aligned}$$

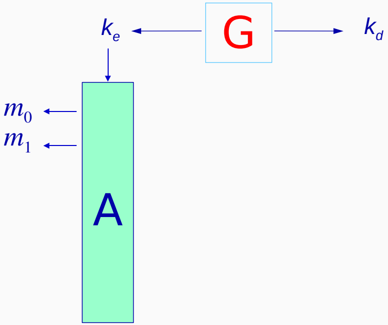
IND − CPA Security Game



$\mathsf{IND}-\mathsf{CPA}$ Security Game

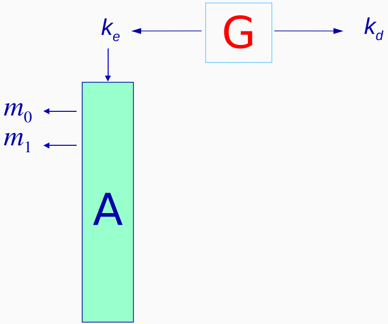


IND − CPA Security Game

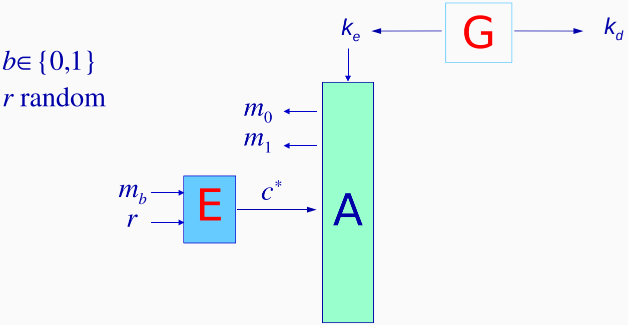


$\mathrm{IND}-\mathrm{CPA}$ Security Game

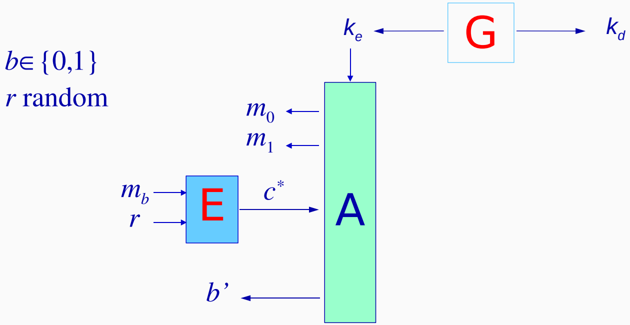
$b \in \{0,1\}$
r random



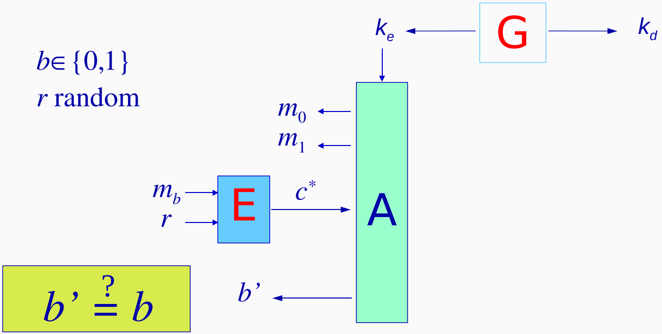
IND − CPA Security Game



$\mathrm{IND}-\mathrm{CPA}$ Security Game



IND – CPA Security Game



Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - if $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b';
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b';
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b',
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b',
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b',
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \overset{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b',
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b',
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Note that (if diff denotes the event that $\mathcal{P}(m) \neq \mathcal{P}(m'))$

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b',
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Note that (if diff denotes the event that $\mathcal{P}(m) \neq \mathcal{P}(m')$)

$$\mathbf{Adv}^{\mathsf{sem}}(\mathcal{A}) = \left|\Pr[p = \mathcal{P}(m) | c = \mathcal{E}(m)] - \Pr[p = \mathcal{P}(m') | c = \mathcal{E}(m)]\right|$$

Indistinguishability implies Semantic Security

Let $\mathcal{A}$ be an adversary within time t against semantic security, we build an adversary $\mathcal{B}$ against indistinguishability:

- $\mathcal{B}$ runs $\mathcal{A}$ to get $\mathcal{D}$ and a predicate $\mathcal{P}$;
  it gets $m_0, m_1 \stackrel{R}{\leftarrow} \mathcal{D}$, and outputs them;
- the challenger encrypts m_b in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get the guess p of $\mathcal{A}$ about the predicate $\mathcal{P}$ on the plaintext in c;
  - If $\mathcal{P}(m_0) = \mathcal{P}(m_1)$, $\mathcal{B}$ outputs a random bit b',
  - otherwise it outputs b' such that $\mathcal{P}(m_{b'}) = p$.

Note that (if diff denotes the event that $\mathcal{P}(m) \neq \mathcal{P}(m')$)

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{sem}}(\mathcal{A}) &= \left|\Pr[p = \mathcal{P}(m) | c = \mathcal{E}(m)] - \Pr[p = \mathcal{P}(m') | c = \mathcal{E}(m)]\right| \\
&= \left|\begin{array}{l} \Pr[p = \mathcal{P}(m) | c = \mathcal{E}(m) \wedge \mathsf{diff}] \\ - \Pr[p = \mathcal{P}(m') | c = \mathcal{E}(m) \wedge \mathsf{diff}] \end{array}\right| \times \Pr[\mathsf{diff}]
\end{aligned}$$

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

$$\mathbf{Adv}^{\mathsf{ind}}(\mathcal{B}) = \left|\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]\right|$$

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

$$
\begin{aligned}
\mathbf{Adv}^{\mathsf{ind}}(\mathcal{B}) &= \left| \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] \right| \\
&= \left| \begin{array}{l} \Pr[b' = 1 | b = 1 \wedge \mathsf{diff}] \\ - \Pr[b' = 1 | b = 0 \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}]
\end{aligned}
$$

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ind}}(\mathcal{B}) &= \left|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]\right| \\
&= \left| \begin{array}{l} \Pr[b' = 1 | b = 1 \wedge \mathsf{diff}] \\ - \Pr[b' = 1 | b = 0 \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}] \\
&= \left| \begin{array}{l} \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \\ - \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_0) \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}]
\end{aligned}$$

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

$$
\begin{aligned}
\mathbf{Adv}^{\mathsf{ind}}(\mathcal{B}) &= \left|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]\right| \\
&= \left| \begin{array}{l} \Pr[b' = 1 | b = 1 \wedge \mathsf{diff}] \\ - \Pr[b' = 1 | b = 0 \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}] \\
&= \left| \begin{array}{l} \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \\ - \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_0) \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}] \\
&= \left| \begin{array}{l} \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \\ - \Pr[\mathcal{P}(m_0) = p | c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}]
\end{aligned}
$$

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

$$
\begin{array}{rcl}
\mathbf{Adv}^{\text{ind}}(\mathcal{B}) & = & \left|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]\right| \\
& = & \left| \begin{array}{l} \Pr[b' = 1 | b = 1 \wedge \text{diff}] \\ - \Pr[b' = 1 | b = 0 \wedge \text{diff}] \end{array} \right| \times \Pr[\text{diff}] \\
& = & \left| \begin{array}{l} \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_1) \wedge \text{diff}] \\ - \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_0) \wedge \text{diff}] \end{array} \right| \times \Pr[\text{diff}] \\
& = & \left| \begin{array}{l} \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_1) \wedge \text{diff}] \\ - \Pr[\mathcal{P}(m_0) = p | c = \mathcal{E}(m_1) \wedge \text{diff}] \end{array} \right| \times \Pr[\text{diff}] \\
& = & \mathbf{Adv}^{\text{sem}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ind}}(t')
\end{array}
$$

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ind}}(\mathcal{B}) &= \left|\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]\right| \\
&= \left|\begin{array}{l} \Pr[b' = 1|b = 1 \wedge \mathsf{diff}] \\ - \Pr[b' = 1|b = 0 \wedge \mathsf{diff}] \end{array}\right| \times \Pr[\mathsf{diff}] \\
&= \left|\begin{array}{l} \Pr[\mathcal{P}(m_1) = p|c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \\ - \Pr[\mathcal{P}(m_1) = p|c = \mathcal{E}(m_0) \wedge \mathsf{diff}] \end{array}\right| \times \Pr[\mathsf{diff}] \\
&= \left|\begin{array}{l} \Pr[\mathcal{P}(m_1) = p|c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \\ - \Pr[\mathcal{P}(m_0) = p|c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \end{array}\right| \times \Pr[\mathsf{diff}] \\
&= \mathbf{Adv}^{\mathsf{sem}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{ind}}(t')
\end{aligned}$$

The running time t' of $\mathcal{B}$ = one execution of $\mathcal{A}$ (time t), two sampling from $\mathcal{D}$ (time τ_D), two evaluations of the predicate $\mathcal{P}$ (time $\tau_{\mathcal{P}}$)

Indistinguishability implies Semantic Security

If diff denotes the event that $\mathcal{P}(m_0) \neq \mathcal{P}(m_1)$

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ind}}(\mathcal{B}) &= \left|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]\right| \\
&= \left| \begin{array}{l} \Pr[b' = 1 | b = 1 \wedge \mathsf{diff}] \\ - \Pr[b' = 1 | b = 0 \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}] \\
&= \left| \begin{array}{l} \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \\ - \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_0) \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}] \\
&= \left| \begin{array}{l} \Pr[\mathcal{P}(m_1) = p | c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \\ - \Pr[\mathcal{P}(m_0) = p | c = \mathcal{E}(m_1) \wedge \mathsf{diff}] \end{array} \right| \times \Pr[\mathsf{diff}] \\
&= \mathbf{Adv}^{\mathsf{sem}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{ind}}(t')
\end{aligned}$$

The running time t' of $\mathcal{B}$ = one execution of $\mathcal{A}$ (time t), two sampling from $\mathcal{D}$ (time τ_D), two evaluations of the predicate $\mathcal{P}$ (time $\tau_{\mathcal{P}}$)

$$\mathbf{Adv}^{\mathsf{sem}}(t) \leq \mathbf{Adv}^{\mathsf{ind}}(t + 2\tau_{\mathcal{D}} + 2\tau_{\mathcal{P}})$$

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

$$\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) \quad = \quad \left|\Pr[p = \mathcal{P}(m)] - \Pr[p = \mathcal{P}(m')]\right|$$

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

$$\begin{aligned} \mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \left|\Pr[p = \mathcal{P}(m)] - \Pr[p = \mathcal{P}(m')]\right| \\ &= \left|\Pr[m = m_p] - \Pr[m' = m_p]\right| \end{aligned}$$

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \left|\Pr[p = \mathcal{P}(m)] - \Pr[p = \mathcal{P}(m')]\right| \\
&= \left|\Pr[m = m_p] - \Pr[m' = m_p]\right| \\
&= \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right|
\end{aligned}$$

Semantic Security implies Indistinguishability

Let $\mathcal{A}$ be an adversary within time t against indistinguishability, we build an adversary $\mathcal{B}$ against semantic security:

- $\mathcal{B}$ runs $\mathcal{A}$ to get (m_0, m_1);
  it sets $\mathcal{D} = \{m_0, m_1\}$, and $\mathcal{P}(m) = (m \stackrel{?}{=} m_1)$;
- the challenger chooses $m, m' \stackrel{R}{\leftarrow} \mathcal{D}$, and encrypts m in c
- $\mathcal{B}$ runs $\mathcal{A}$, to get b', that it forwards as its guess p

$$\begin{aligned}
\mathbf{Adv}^{\text{sem}}(\mathcal{B}) &= \left|\Pr[p = \mathcal{P}(m)] - \Pr[p = \mathcal{P}(m')]\right| \\
&= \left|\Pr[m = m_p] - \Pr[m' = m_p]\right| \\
&= \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right| \\
\mathbf{Adv}^{\text{ind}}(\mathcal{A}) &= \left|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]\right| \\
&\quad \text{where } m = m_b
\end{aligned}$$

Semantic Security implies Indistinguishability

Semantic Security implies Indistinguishability

$$\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) \quad = \quad \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right|$$

Semantic Security implies Indistinguishability

$$\begin{aligned}\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right| \\ &= |\Pr[m_b = m_{b'}] - \Pr[m_d = m_{b'}]| \\ &\quad \text{where } m = m_b \text{ and } m' = m_d\end{aligned}$$

Semantic Security implies Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right| \\
&= \left|\Pr[m_b = m_{b'}] - \Pr[m_d = m_{b'}]\right| \\
&\quad \text{where } m = m_b \text{ and } m' = m_d \\
&= \left|\Pr[b = b'] - \Pr[d = b']\right|
\end{aligned}$$

Semantic Security implies Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right| \\
&= |\Pr[m_b = m_{b'}] - \Pr[m_d = m_{b'}]| \\
&\quad \text{where } m = m_b \text{ and } m' = m_d \\
&= \left|\Pr[b = b'] - \Pr[d = b']\right| \\
&= \left|\Pr[b = b'] - 1/2\right|
\end{aligned}$$

Semantic Security implies Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \Big|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\Big| \\
&= |\Pr[m_b = m_{b'}] - \Pr[m_d = m_{b'}]| \\
&\quad \text{where } m = m_b \text{ and } m' = m_d \\
&= \Big|\Pr[b = b'] - \Pr[d = b']\Big| \\
&= \Big|\Pr[b = b'] - 1/2\Big| \\
&= \mathbf{Adv}^{\mathsf{ind}}(\mathcal{A})/2 \leq \mathbf{Adv}^{\mathsf{sem}}(t')
\end{aligned}$$

Semantic Security implies Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right| \\
&= |\Pr[m_b = m_{b'}] - \Pr[m_d = m_{b'}]| \\
&\quad \text{where } m = m_b \text{ and } m' = m_d \\
&= \left|\Pr[b = b'] - \Pr[d = b']\right| \\
&= \left|\Pr[b = b'] - 1/2\right| \\
&= \mathbf{Adv}^{\mathsf{ind}}(\mathcal{A})/2 \leq \mathbf{Adv}^{\mathsf{sem}}(t')
\end{aligned}$$

The running time t' of $\mathcal{B}$ = one execution of $\mathcal{A}$ (time t)

Semantic Security implies Indistinguishability

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{sem}}(\mathcal{B}) &= \left|\Pr[m = m_{b'}] - \Pr[m' = m_{b'}]\right| \\
&= |\Pr[m_b = m_{b'}] - \Pr[m_d = m_{b'}]| \\
&\quad \text{where } m = m_b \text{ and } m' = m_d \\
&= \left|\Pr[b = b'] - \Pr[d = b']\right| \\
&= \left|\Pr[b = b'] - 1/2\right| \\
&= \mathbf{Adv}^{\mathsf{ind}}(\mathcal{A})/2 \leq \mathbf{Adv}^{\mathsf{sem}}(t')
\end{aligned}$$

The running time t' of $\mathcal{B}$ = one execution of $\mathcal{A}$ (time t)

$$\mathbf{Adv}^{\mathsf{ind}}(t) \leq 2 \times \mathbf{Adv}^{\mathsf{sem}}(t)$$

ElGamal Encryption

The ElGamal encryption scheme $\mathcal{EG}$ is defined,
in a group $\mathbb{G} = \langle g \rangle$ of order q

- $\mathcal{K}(\mathbb{G}, g, q)$: $x \overset{R}{\leftarrow} \mathbb{Z}_q$, and $sk \leftarrow x$ and $pk \leftarrow y = g^x$
- $\mathcal{E}_{pk}(m)$: $r \overset{R}{\leftarrow} \mathbb{Z}_q$, $c_1 \leftarrow g^r$ and $c_2 \leftarrow y^r \times m = pk^r \times m$.
  Then, the ciphertext is $c = (c_1, c_2)$
- $\mathcal{D}_{sk}(c)$ outputs $c_2 / c_1^x = c_2 / c_1^{sk}$

Theorem (ElGamal is $\mathrm{IND-CPA}$)

$$\mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(t) \leq 2 \times \mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(t)$$

ElGamal Encryption

ElGamal Encryption

The ElGamal encryption scheme $\mathcal{EG}$ is defined,
in a group $\mathbb{G} = \langle g \rangle$ of order q

- $\mathcal{K}(\mathbb{G}, g, q)$: $x \stackrel{R}{\leftarrow} \mathbb{Z}_q$, and $sk \leftarrow x$ and $pk \leftarrow y = g^x$
- $\mathcal{E}_{pk}(m)$: $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$, $c_1 \leftarrow g^r$ and $c_2 \leftarrow y^r \times m = pk^r \times m$.
  Then, the ciphertext is $c = (c_1, c_2)$
- $\mathcal{D}_{sk}(c)$ outputs $c_2 / c_1^x = c_2 / c_1^{sk}$

Theorem (ElGamal is $\mathbf{IND-CPA}$)

$$\mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(t) \leq 2 \times \mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(t)$$

ElGamal is $\mathrm{IND}-\mathrm{CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$, and outputs (m_0, m_1)

ElGamal is $\mathrm{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$, and outputs (m_0, m_1)
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$

ElGamal is $\mathrm{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$, and outputs (m_0, m_1)
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow y^{r^*} \times m_b$, and sends $c = (c_1, c_2)$

ElGamal is $\mathbf{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$, and outputs (m_0, m_1)
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $b \stackrel{R}{\leftarrow} \{0,1\}$, sets $c_2 \leftarrow y^{r^*} \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs b'

ElGamal is $\mathbf{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow y = g^x$ from $\mathcal{K}$, and outputs (m_0, m_1)
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow y^{r^*} \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs b'
- $2 \times \Pr[b' = b] - 1 = \mathbf{Adv}_{\mathcal{EG}}^{\mathrm{ind-cpa}}(\mathcal{A})$

ElGamal is $\mathbf{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs (m_0, m_1)
- The challenger chooses $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and sets $c_1 \leftarrow g^{r^*}$
- The challenger chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow y^{r^*} \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs b'
- $2 \times \Pr[b' = b] - 1 = \mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(\mathcal{A})$

ElGamal is $\mathbf{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs (m_0, m_1)
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- The challenger chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow y^{r^*} \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs b'
- $2 \times \Pr[b' = b] - 1 = \mathbf{Adv}^{\mathsf{ind-cpa}}_{\mathcal{EG}}(\mathcal{A})$

ElGamal is $\mathbf{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs (m_0, m_1)
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow Z \times m_b$,
  and sends $c = (c_1, c_2)$
- $\mathcal{A}$ receives $c \leftarrow (c_1, c_2)$, and outputs b'
- $2 \times \Pr[b' = b] - 1 = \mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(\mathcal{A})$

ElGamal is $\mathbf{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs (m_0, m_1)
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow Z \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives b' from $\mathcal{A}$ and outputs $d = (b' = b)$
- $2 \times \Pr[b' = b] - 1 = \mathbf{Adv}^{\text{ind}-\text{cpa}}_{\mathcal{EG}}(\mathcal{A})$

ElGamal is $\mathrm{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs (m_0, m_1)
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow Z \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives b' from $\mathcal{A}$ and outputs $d = (b' = b)$
- $|2 \times \Pr[b' = b] - 1|$
  $= \mathbf{Adv}_{\mathcal{EG}}^{\mathrm{ind-cpa}}(\mathcal{A})$, if $Z = \mathbf{CDH}(X, Y)$

ElGamal is $\mathrm{IND-CPA}$: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{EG}$, we build an adversary $\mathcal{B}$ against **DDH**: let us be given a **DDH** instance (X, Y, Z)

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs (m_0, m_1)
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow Z \times m_b$,
  and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives b' from $\mathcal{A}$ and outputs $d = (b' = b)$
- $|2 \times \Pr[b' = b] - 1|$
  $= \mathbf{Adv}^{\mathrm{ind-cpa}}_{\mathcal{EG}}(\mathcal{A})$, if $Z = \mathbf{CDH}(X, Y)$
  $= 0$, otherwise

ElGamal is $\mathrm{IND}-\mathrm{CPA}$: Proof

As a consequence,

- $|2 \times \Pr[b' = b | Z = \mathbf{CDH}(X, Y)] - 1| = \mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(\mathcal{A})$
- $\left|2 \times \Pr[b' = b | Z \stackrel{R}{\leftarrow} \mathbb{G}] - 1\right| = 0$

$$\begin{aligned} \mathbf{Adv}_{\mathcal{EG}}^{\mathsf{ind-cpa}}(\mathcal{A}) &= 2 \times \left| \begin{array}{l} \Pr[d = 1 | Z = \mathbf{CDH}(X, Y)] \\ - \Pr[d = 1 | Z \stackrel{R}{\leftarrow} \mathbb{G}] \end{array} \right| \\ &= 2 \times \mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(\mathcal{B}) \leq 2 \times \mathbf{Adv}_{\mathbb{G}}^{\mathbf{ddh}}(t) \end{aligned}$$

ElGamal is $\mathbf{IND-CPA}$: Proof

As a consequence,

- $|2 \times \Pr[b' = b | Z = \mathbf{CDH}(X, Y)] - 1| = \mathbf{Adv}^{\mathsf{ind-cpa}}_{\mathcal{EG}}(\mathcal{A})$
- $\left| 2 \times \Pr[b' = b | Z \xleftarrow{R} \mathbb{G}] - 1 \right| = 0$

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ind-cpa}}_{\mathcal{EG}}(\mathcal{A}) &= 2 \times \left| \begin{array}{l} \Pr[d = 1 | Z = \mathbf{CDH}(X, Y)] \\ - \Pr[d = 1 | Z \xleftarrow{R} \mathbb{G}] \end{array} \right| \\
&= 2 \times \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(\mathcal{B}) \leq 2 \times \mathbf{Adv}^{\mathbf{ddh}}_{\mathbb{G}}(t)
\end{aligned}$$

$\mathcal{RSA}$ Encryption [Rivest-Shamir-Adleman 1978]

$\mathcal{RSA}$ Encryption

The RSA encryption scheme $\mathcal{RSA}$ is defined by

- $\mathcal{K}(1^k)$: p and q two random k-bit prime integers, and an exponent e (possibly fixed, or not): $sk \leftarrow d = e^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}_{pk}(m)$: the ciphertext is $c = m^e \bmod n$
- $\mathcal{D}_{sk}(c)$: the plaintext is $m = c^d \bmod n$

Theorem ($\mathcal{RSA}$ is $\mathrm{OW-CPA}$, but…)

A deterministic encryption scheme cannot be $\mathrm{IND-CPA}$

$\mathcal{RSA}$ Encryption [Rivest-Shamir-Adleman 1978]

$\mathcal{RSA}$ Encryption

The RSA encryption scheme $\mathcal{RSA}$ is defined by

- $\mathcal{K}(1^k)$: p and q two random k-bit prime integers, and an exponent e (possibly fixed, or not): $sk \leftarrow d = e^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}_{pk}(m)$: the ciphertext is $c = m^e \bmod n$
- $\mathcal{D}_{sk}(c)$: the plaintext is $m = c^d \bmod n$

Theorem ($\mathcal{RSA}$ is $\mathbf{OW-CPA}$, but...)

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{RSA}}(t) \leq \mathbf{Succ}^{\mathsf{rsa}}(t)$$

A deterministic encryption scheme cannot be $\mathbf{IND-CPA}$

$\mathcal{RSA}$ Encryption

The RSA encryption scheme $\mathcal{RSA}$ is defined by

- $\mathcal{K}(1^k)$: p and q two random k-bit prime integers, and an exponent e (possibly fixed, or not): $sk \leftarrow d = e^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}_{pk}(m)$: the ciphertext is $c = m^e \bmod n$
- $\mathcal{D}_{sk}(c)$: the plaintext is $m = c^d \bmod n$

Theorem ($\mathcal{RSA}$ is $\mathbf{OW-CPA}$, but…)

$$\mathbf{Succ}^{\mathsf{ow-cpa}}_{\mathcal{RSA}}(t) \leq \mathbf{Succ}^{\mathsf{rsa}}(t)$$

A deterministic encryption scheme cannot be $\mathbf{IND-CPA}$

Outline

Cryptography

Provable Security

Basic Security Notions

- Public-Key Encryption
- Variants of Indistinguishability
- Signatures

Conclusion

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and chooses 2 messages m_0 and m_1
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathrm{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and chooses 2 messages m_0 and m_1
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathrm{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathrm{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and chooses 2 messages m_0 and m_1
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and chooses 2 messages m_0 and m_1
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and chooses 2 messages m_0 and m_1
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and chooses 2 messages m_0 and m_1
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and chooses 2 messages m_0 and m_1
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and chooses 2 messages m_0 and m_1 Find stage
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and chooses 2 messages m_0 and m_1 Find stage
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b Guess stage

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Indistinguishability vs. Find-then-Guess

[Bellare-Desai-Jokipii-Rogaway 1997]

FtG – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and chooses 2 messages m_0 and m_1 — Find stage
- The challenger returns the encryption c of m_b under pk
- The adversary outputs its guess b' on the bit b — Guess stage

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ind-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Note: the adversary has access to the following oracle, only once:
$\mathsf{LR}_b(m_0, m_1)$: outputs the encryption of m_b under pk

Left-or-Right Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

LoR – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks LR on any pair (m_0, m_1) of its choice
- The challenger answers using LR_b
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Left-or-Right Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

LoR – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and asks LR on any pair (m_0, m_1) of its choice
- The challenger answers using LR_b
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Left-or-Right Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

LoR – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks LR on any pair (m_0, m_1) of its choice
- The challenger answers using LR_b
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{lor-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Left-or-Right Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

LoR − CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks LR on any pair (m_0, m_1) of its choice
- The challenger answers using LR_b
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\text{lor-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Left-or-Right Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

LoR – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks LR on any pair (m_0, m_1) of its choice
- The challenger answers using LR_b
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Left-or-Right Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

LoR – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks LR on any pair (m_0, m_1) of its choice
- The challenger answers using LR_b
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Left-or-Right Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

LoR – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks LR on any pair (m_0, m_1) of its choice
- The challenger answers using LR_b
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{lor-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Find-then-Guess vs. Left-or-Right

Theorem (FtG $\overset{n}{\sim}$ LoR)

$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t)$$
$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \leq n \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t)$$

where n is the number of LR queries

LoR $\Rightarrow$ FtG is clear

FtG $\Rightarrow$ LoR: hybrid distribution of the sequence of bits b

- The Left distribution is $(0, 0, \ldots, 0) \in \{0, 1\}^n$, for the LR queries
- The Right distribution is $(1, 1, \ldots, 1) \in \{0, 1\}^n$, for the LR queries
- Hybrid distribution: $\mathcal{D}_i = (0, \ldots, 0, 1, \ldots, 1) = 0^i 1^{n-i} \in \{0, 1\}^n$

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_n) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) \quad \mathbf{Dist}(\mathcal{D}_i, \mathcal{D}_{i+1}) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t)$$

Find-then-Guess vs. Left-or-Right

Theorem (FtG $\overset{n}{\sim}$ LoR)

$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t)$$
$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \leq n \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t)$$

where n is the number of LR queries

LoR $\Rightarrow$ FtG is clear

FtG $\Rightarrow$ LoR: hybrid distribution of the sequence of bits b

- The Left distribution is $(0, 0, \ldots, 0) \in \{0, 1\}^n$, for the LR queries
- The Right distribution is $(1, 1, \ldots, 1) \in \{0, 1\}^n$, for the LR queries
- Hybrid distribution: $\mathcal{D}_i = (0, \ldots, 0, 1, \ldots, 1) = 0^i 1^{n-i} \in \{0, 1\}^n$

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_n) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) \quad \mathbf{Dist}(\mathcal{D}_i, \mathcal{D}_{i+1}) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t)$$

Find-then-Guess vs. Left-or-Right

Theorem (FtG $\overset{n}{\sim}$ LoR)

$$\begin{aligned} \forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\text{ftg}-\text{cpa}}(t) &\leq \mathbf{Adv}_{\mathcal{S}}^{\text{lor}-\text{cpa}}(t) \\ \forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\text{lor}-\text{cpa}}(t) &\leq n \times \mathbf{Adv}_{\mathcal{S}}^{\text{ftg}-\text{cpa}}(t) \end{aligned}$$

where n is the number of LR queries

LoR $\Rightarrow$ FtG is clear

FtG $\Rightarrow$ LoR: hybrid distribution of the sequence of bits b

- The Left distribution is $(0, 0, \ldots, 0) \in \{0, 1\}^n$, for the LR queries
- The Right distribution is $(1, 1, \ldots, 1) \in \{0, 1\}^n$, for the LR queries
- Hybrid distribution: $\mathcal{D}_i = (0, \ldots, 0, 1, \ldots, 1) = 0^i 1^{n-i} \in \{0, 1\}^n$

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_n) = \mathbf{Adv}_{\mathcal{S}}^{\text{lor}-\text{cpa}}(\mathcal{A}) \quad \mathbf{Dist}(\mathcal{D}_i, \mathcal{D}_{i+1}) \leq \mathbf{Adv}_{\mathcal{S}}^{\text{ftg}-\text{cpa}}(t)$$

Find-then-Guess vs. Left-or-Right

Theorem (FtG $\overset{n}{\sim}$ LoR)

$$\begin{aligned}
\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t) &\leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \\
\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) &\leq n \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t)
\end{aligned}$$

where n is the number of LR queries

LoR $\Rightarrow$ FtG is clear

FtG $\Rightarrow$ LoR: hybrid distribution of the sequence of bits b

- The Left distribution is $(0, 0, \ldots, 0) \in \{0, 1\}^n$, for the LR queries
- The Right distribution is $(1, 1, \ldots, 1) \in \{0, 1\}^n$, for the LR queries
- Hybrid distribution: $\mathcal{D}_i = (0, \ldots, 0, 1, \ldots, 1) = 0^i 1^{n-i} \in \{0, 1\}^n$

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_n) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) \quad \mathbf{Dist}(\mathcal{D}_i, \mathcal{D}_{i+1}) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t)$$

Find-then-Guess vs. Left-or-Right

Theorem (FtG $\overset{n}{\sim}$ LoR)

$$\begin{aligned} \forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t) &\leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \\ \forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) &\leq n \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t) \end{aligned}$$

where n is the number of LR queries

LoR $\Rightarrow$ FtG is clear

FtG $\Rightarrow$ LoR: hybrid distribution of the sequence of bits b

- The Left distribution is $(0, 0, \ldots, 0) \in \{0,1\}^n$, for the LR queries
- The Right distribution is $(1, 1, \ldots, 1) \in \{0,1\}^n$, for the LR queries
- Hybrid distribution: $\mathcal{D}_i = (0, \ldots, 0, 1, \ldots, 1) = 0^i 1^{n-i} \in \{0,1\}^n$

$$\mathbf{Dist}(\mathcal{D}_0, \mathcal{D}_n) = \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(\mathcal{A}) \quad \mathbf{Dist}(\mathcal{D}_i, \mathcal{D}_{i+1}) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ftg-cpa}}(t)$$

Real-or-Random Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR − CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks RR on any message m of its choice
- The challenger answers using RR_b:
  - if $b = 0$, the RR_0 encrypts m Real
  - if $b = 1$, the RR_1 encrypts a random message Random
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{ror-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR − CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks RR on any message m of its choice
- The challenger answers using RR_b:
  - if $b = 0$, the RR_0 encrypts m Real
  - if $b = 1$, the RR_1 encrypts a random message Random
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{ror-cpa}}(\mathcal{A}) = \left| 2 \times \Pr[b' = b] - 1 \right|$$

Real-or-Random Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR − CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks RR on any message m of its choice
- The challenger answers using RR_b:
  - if $b = 0$, the RR_0 encrypts m — Real
  - if $b = 1$, the RR_1 encrypts a random message — Random
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{ror-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Real-or-Random Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR − CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and asks RR on any message m of its choice
- The challenger answers using RR_b:
  - if $b = 0$, the RR_0 encrypts m — Real
  - if $b = 1$, the RR_1 encrypts a random message — Random
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Real-or-Random Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR − CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk, and asks RR on any message m of its choice
- The challenger answers using RR_b:
  - if $b = 0$, the RR_0 encrypts m Real
  - if $b = 1$, the RR_1 encrypts a random message Random
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Real-or-Random Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR – CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and asks RR on any message m of its choice
- The challenger answers using RR_b:
  - if $b = 0$, the RR_0 encrypts m — Real
  - if $b = 1$, the RR_1 encrypts a random message — Random
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(\mathcal{A}) = \left| 2 \times \Pr[b' = b] - 1 \right|$$

Real-or-Random Indistinguishability

[Bellare-Desai-Jokipii-Rogaway 1997]

RoR − CPA

- The challenger flips a bit b
- The challenger runs the key generation algorithm $(sk, pk) \leftarrow \mathcal{K}()$
- The adversary receives the public key pk,
  and asks RR on any message m of its choice
- The challenger answers using RR_b:
  - if $b = 0$, the RR_0 encrypts m — Real
  - if $b = 1$, the RR_1 encrypts a random message — Random
- The adversary outputs its guess b' on the bit b

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(\mathcal{A}) = \left|2 \times \Pr[b' = b] - 1\right|$$

Left-or-Right vs. Real-Random

Theorem (LoR $\sim$ RoR)

$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t)$$
$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \leq 2 \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t)$$

LoR $\Rightarrow$ RoR is clear (using $m_0 = m$ and $m_1 \stackrel{R}{\leftarrow} \mathcal{M}$)

RoR $\Rightarrow$ LoR: $\mathcal{B}$ flips a bit d, and uses m_d for the RR oracle, then forwards $\mathcal{A}$'s answer

$$\Pr[d \leftarrow \mathcal{B} | \mathsf{Real}] = \Pr[d \leftarrow \mathcal{A}] \quad \Pr[d \leftarrow \mathcal{B} | \mathsf{Random}] = 1/2$$

$$\begin{aligned} \mathbf{Adv}^{\mathsf{lor}}(\mathcal{A}) &= |2 \times \Pr[d \leftarrow \mathcal{A}] - 1| \\ &= |2 \times \Pr[d \leftarrow \mathcal{B} | \mathsf{Real}] - 2 \times \Pr[d \leftarrow \mathcal{B} | \mathsf{Random}]| \\ &\leq 2 \times \mathbf{Adv}^{\mathsf{ror}}(\mathcal{B}) \end{aligned}$$

Left-or-Right vs. Real-Random

Theorem (LoR $\sim$ RoR)

$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t) \leq \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t)$$
$$\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \leq 2 \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t)$$

LoR $\Rightarrow$ RoR is clear (using $m_0 = m$ and $m_1 \stackrel{R}{\leftarrow} \mathcal{M}$)

RoR $\Rightarrow$ LoR: $\mathcal{B}$ flips a bit d, and uses m_d for the RR oracle, then forwards $\mathcal{A}$'s answer

$$\Pr[d \leftarrow \mathcal{B}|\mathsf{Real}] = \Pr[d \leftarrow \mathcal{A}] \quad \Pr[d \leftarrow \mathcal{B}|\mathsf{Random}] = 1/2$$

$$\begin{aligned} \mathbf{Adv}^{\mathsf{lor}}(\mathcal{A}) &= |2 \times \Pr[d \leftarrow \mathcal{A}] - 1| \\ &= |2 \times \Pr[d \leftarrow \mathcal{B}|\mathsf{Real}] - 2 \times \Pr[d \leftarrow \mathcal{B}|\mathsf{Random}]| \\ &\leq 2 \times \mathbf{Adv}^{\mathsf{ror}}(\mathcal{B}) \end{aligned}$$

Theorem (LoR $\sim$ RoR)

$$\begin{array}{rrcl} \forall t, & \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t) & \leq & \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \\ \forall t, & \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) & \leq & 2 \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t) \end{array}$$

LoR $\Rightarrow$ RoR is clear (using $m_0 = m$ and $m_1 \stackrel{R}{\leftarrow} \mathcal{M}$)

RoR $\Rightarrow$ LoR: $\mathcal{B}$ flips a bit d, and uses m_d for the RR oracle, then forwards $\mathcal{A}$'s answer

$$\Pr[d \leftarrow \mathcal{B}|\mathsf{Real}] = \Pr[d \leftarrow \mathcal{A}] \quad \Pr[d \leftarrow \mathcal{B}|\mathsf{Random}] = 1/2$$

$$\begin{array}{rcl} \mathbf{Adv}^{\mathsf{lor}}(\mathcal{A}) & = & |2 \times \Pr[d \leftarrow \mathcal{A}] - 1| \\ & = & |2 \times \Pr[d \leftarrow \mathcal{B}|\mathsf{Real}] - 2 \times \Pr[d \leftarrow \mathcal{B}|\mathsf{Random}]| \\ & \leq & 2 \times \mathbf{Adv}^{\mathsf{ror}}(\mathcal{B}) \end{array}$$

Left-or-Right vs. Real-Random

Theorem (LoR $\sim$ RoR)

$$\begin{array}{rcl}\forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t) & \leq & \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) \\ \forall t, \quad \mathbf{Adv}_{\mathcal{S}}^{\mathsf{lor-cpa}}(t) & \leq & 2 \times \mathbf{Adv}_{\mathcal{S}}^{\mathsf{ror-cpa}}(t)\end{array}$$

LoR $\Rightarrow$ RoR is clear (using $m_0 = m$ and $m_1 \stackrel{R}{\leftarrow} \mathcal{M}$)

RoR $\Rightarrow$ LoR: $\mathcal{B}$ flips a bit d, and uses m_d for the RR oracle, then forwards $\mathcal{A}$'s answer

$$\Pr[d \leftarrow \mathcal{B} | \mathsf{Real}] = \Pr[d \leftarrow \mathcal{A}] \quad \Pr[d \leftarrow \mathcal{B} | \mathsf{Random}] = 1/2$$

$$\begin{array}{rcl}\mathbf{Adv}^{\mathsf{lor}}(\mathcal{A}) & = & |2 \times \Pr[d \leftarrow \mathcal{A}] - 1| \\ & = & |2 \times \Pr[d \leftarrow \mathcal{B} | \mathsf{Real}] - 2 \times \Pr[d \leftarrow \mathcal{B} | \mathsf{Random}]| \\ & \leq & 2 \times \mathbf{Adv}^{\mathsf{ror}}(\mathcal{B})\end{array}$$

Outline

Cryptography

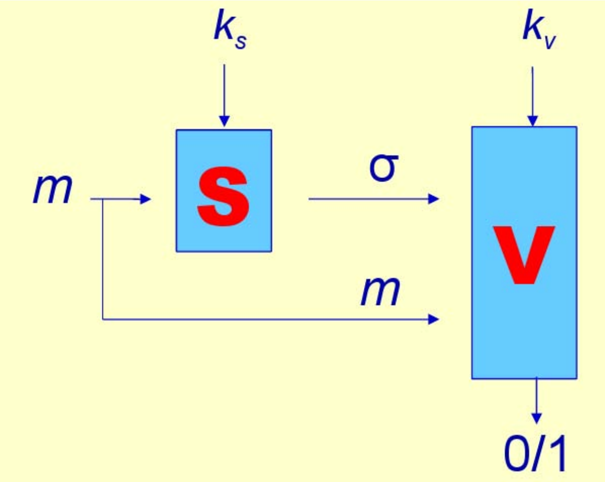
Provable Security

Basic Security Notions

- Public-Key Encryption
- Variants of Indistinguishability
- Signatures

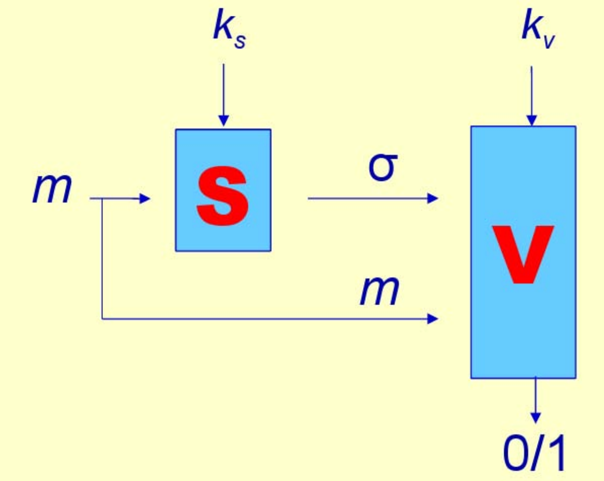
Conclusion

Signature



Goal: Authentication of the sender

Signature



Goal: Authentication of the sender

EUF − NMA

Existential Unforgeability

For a signature scheme $\mathcal{SG} = (\mathcal{K}, \mathcal{S}, \mathcal{V})$, without the secrete key *sk*, it should be computationally impossible to generate a valid pair (m, σ):

$$\mathbf{Succ}^{\mathsf{euf}}_{\mathcal{SG}}(\mathcal{A}) = \Pr[(sk, pk) \leftarrow \mathcal{K}(); (m, \sigma) \leftarrow \mathcal{A}(pk) : \mathcal{V}_{pk}(m, \sigma) = 1]$$

should be negligible.

No-Message Attacks

In the public-key setting, the adversary has access to the verification key (the public key), but not necessarily to valid signatures: no-message attack

EUF − NMA

Existential Unforgeability

For a signature scheme $\mathcal{SG} = (\mathcal{K}, \mathcal{S}, \mathcal{V})$, without the secrete key *sk*, it should be computationally impossible to generate a valid pair (m, σ):

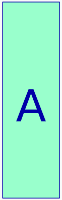
$$\mathbf{Succ}^{\mathsf{euf}}_{\mathcal{SG}}(\mathcal{A}) = \Pr[(sk, pk) \leftarrow \mathcal{K}(); (m, \sigma) \leftarrow \mathcal{A}(pk) : \mathcal{V}_{pk}(m, \sigma) = 1]$$

should be negligible.

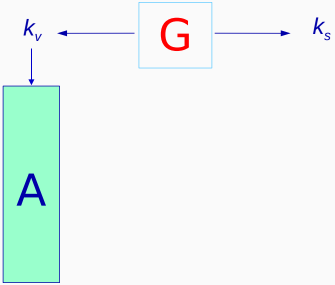
No-Message Attacks

In the public-key setting, the adversary has access to the verification key (the public key), but not necessarily to valid signatures: no-message attack

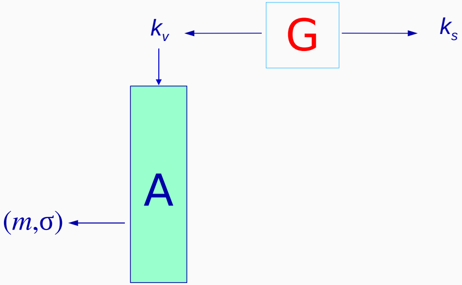
$\mathrm{EUF} - \mathrm{NMA}$ Security Game



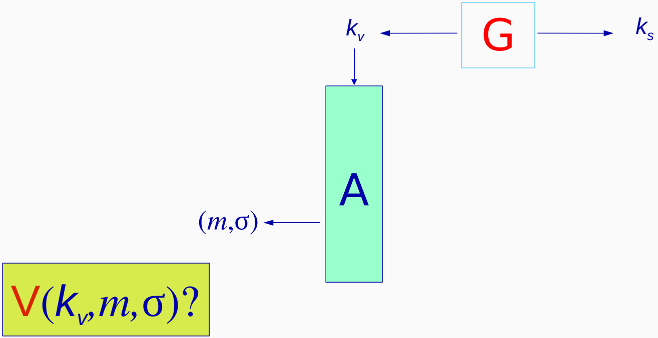
$\mathrm{EUF} - \mathrm{NMA}$ Security Game



EUF − NMA Security Game



$\mathsf{EUF} - \mathsf{NMA}$ Security Game



$\mathcal{RSA}$ Signature

The RSA signature scheme $\mathcal{RSA}$ is defined by

- $\mathcal{K}(1^k)$: p and q two random k-bit prime integers, and an exponent v (possibly fixed, or not): $sk \leftarrow s = v^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, v)$
- $\mathcal{S}_{sk}(m)$: the signature is $\sigma = m^s \bmod n$
- $\mathcal{V}_{pk}(m, \sigma)$ checks whether $m = \sigma^v \bmod n$

Theorem ($\mathcal{RSA}$ is not $\mathrm{EUF}-\mathrm{NMA}$)

The plain RSA signature is not secure at all!

$\mathcal{RSA}$ Signature

The RSA signature scheme $\mathcal{RSA}$ is defined by

- $\mathcal{K}(1^k)$: p and q two random k-bit prime integers, and an exponent v (possibly fixed, or not): $sk \leftarrow s = v^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, v)$
- $\mathcal{S}_{sk}(m)$: the signature is $\sigma = m^s \bmod n$
- $\mathcal{V}_{pk}(m, \sigma)$ checks whether $m = \sigma^v \bmod n$

Theorem ($\mathcal{RSA}$ is not $\mathbf{EUF-NMA}$)

The plain RSA signature is not secure at all!

Conclusion

Outline

Cryptography

Provable Security

Basic Security Notions

Conclusion

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available
- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available
- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
    - encryption:
        - indistinguishability is not enough
        - some information may leak
    - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions

  Proofs will become more intricate!

- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques

Conclusion

- Provable security provides guarantees on the security level
- But strong security notions have to be defined
  - encryption:
    - indistinguishability is not enough
    - some information may leak
  - signature: some signatures may be available

- We will provide stronger security notions
  Proofs will become more intricate!
- We will provide new proof techniques