CRYPTOGRAPHIC PROTOCOLS
2018, LECTURE 4

Elgamal. IND-CPA Security. Malleability

Helger Lipmaa
University of Tartu, Estonia
UP TO NOW

- Assumptions, reductions
UP TO NOW

- Assumptions, reductions
- Basic assumptions in DL family:
UP TO NOW

- Assumptions, reductions
- Basic assumptions in DL family:
  - DL, CDH, DDH
Assumptions, reductions

Basic assumptions in DL family:
- DL, CDH, DDH
- (DH)KE. Security of KE <=> assumptions
Assumptions, reductions

Basic assumptions in DL family:
- DL, CDH, DDH

(DH)KE. Security of KE \(\iff\) assumptions
- Owners of public keys can agree on joint secret key
REMINDER: KEY EXCHANGE
REMINDER: KEY EXCHANGE

I want to send secret information to Bob, but he is in Jamaica
REMINDER: KEY EXCHANGE

Let us agree on a joint secret key for further communication.
SECURE COMPUTATION
I want to know whether I have more hash than Bob. It is ok if Bob also gets to know this, but I do not want to reveal how much exactly I have.
Let us do secure computation of "≥"
Let us do secure computation of "≥"

Ditto
SECURE COMPUTATION

Let us do secure computation of "≥"

Ditto

Millionaires' protocol
2-MESSAGE S.C. IN A NUTSHELL

sk,pk,a
2-MESSAGE S.C. IN A NUTSHELL

sk, pk, a

1 Encode a by using pk
2-MESSAGE S.C. IN A NUTSHELL

 sk, pk, a

 Encode a by using pk

(pk, Encode (pk; a))
2-MESSAGE S.C. IN A NUTSHELL

1. Encode $a$ by using $pk$

($pk, \text{Encode}(pk; a)$)

2. Compute $f(a, b)$ on "encoded" inputs
2-MESSAGE S.C. IN A NUTSHELL

1. Encode $a$ by using $pk$

$(pk, \text{Encode}(pk; a))$

$c = \text{Encode}(pk; f(a, b))$

2. Compute $f(a, b)$ on "encoded" inputs
2-MESSAGE S.C. IN A NUTSHELL

1 Encode \( a \) by using \( pk \)

2 Compute \( f(a, b) \) on "encoded" inputs

3 Decode by using \( sk \), obtain \( f(a, b) \)

\[ c = \text{Encode} \ (pk; f(a, b)) \]

\[ (pk, \text{Encode} \ (pk; a)) \]
2-MESSAGE S.C. IN A NUTSHELL

\[ c = \text{Encode} (pk; f(a, b)) \]

Million dollar question: define “Encode” and “Decode"
THIS LECTURE: MODULARIZATION

- Assumption
  - DDH, ...
- Primitive
  - Encrypt, garble, ...
- Protocol
  - Millionaires', ...

DDH, ... Encrypt, garble, ... Millionaires', ...
THIS LECTURE: MODULARIZATION

Assumption
DDH, ...

Primitive
Encrypt, garble, ...

Protocol
Millionaires', ...

We have a goal
THIS LECTURE: MODULARIZATION

**Assumption**
- DDH, ...

**Primitive**
- Encrypt, garble, ...
  - We find an existing encoding that looks suitable

**Protocol**
- Millionaires', ...
  - We have a goal
This lecture: Modularization

Assumption
DDH, ...
Make the necessary assumption

Primitive
Encrypt, garble, ...
We find an existing encoding that looks suitable

Protocol
Millionaires', ...
We have a goal
THIS LECTURE: MODULARIZATION

**Top-down approach:**
good when you have an application, already know the field, and there are enough primitives/assumptions available

- **Assumption**
  - DDH, ...
  - Make the necessary assumption

- **Primitive**
  - Encrypt, garble, ...
  - We find an existing encoding that looks suitable

- **Protocol**
  - Millionaires', ...
  - We have a goal
THIS LECTURE: MODULARIZATION

Assumption  Primitive  Protocol
THIS LECTURE: MODULARIZATION

Assumption
DDH, ...

Primitive

Protocol
THIS LECTURE: MODULARIZATION

Assumption
- DDH, ...

Primitive
- Encrypt, garble, ...

Protocol
THIS LECTURE: MODULARIZATION

Assumption
- DDH, ...

Primitive
- Encrypt,
- garble, ...

Protocol
- Millionaires',
- ...
THIS LECTURE: MODULARIZATION

We know assumption

Assumption
DDH, ...

Primitive
Encrypt, garble, ...

Protocol
Millionaires', ...

We know assumption
THIS LECTURE: MODULARIZATION

We know assumption

DDH, ...

We build some encoding on top of it

Encrypt, garble, ...

Protocol

Millionaires', ...

Assumption

Primitive
We know assumption

DDH, ...

We build some encoding on top of it

Encrypt, garble, ...

We build the final protocol on top of it

Millionaires', ...

THIS LECTURE: MODULARIZATION
**This Lecture: Modularization**

**Assumption**
- DDH, ...

**Primitive**
- Encrypt, garble, ...
- We build some encoding on top of it

**Protocol**
- Millionaires', ...
- We build the final protocol on top of it

**Bottom-up approach:** good for teaching, research (easier to start with basics) and getting papers published
THIS LECTURE: MODULARIZATION

Assumption

Elgamal (homomorphic encryption)
We know assumption
Assumption
DDH

Elgamal
(homomorphic encryption)
We know assumption

Assumption

DDH

Elgamal
(homomorphic encryption)
We know assumption

DDH

We build some encoding on top of it

Elgamal (homomorphic encryption)

**Bottom-up approach:** good for teaching, research (easier to start with basics) and getting papers published
REMINDER: DHKE

Asymmetric, public key

\[ \text{sk}_a, \text{pk}_a \leftarrow g^{\text{sk}_a} \]

Asymmetric, public key

\[ \text{sk}_b, \text{pk}_b \leftarrow g^{\text{sk}_b} \]

\[ \text{pk}_a \]

\[ \text{pk}_b \]

Symmetric, shared key

\[ \text{sk} \leftarrow \text{SK}(\text{sk}_a, g^{\text{sk}_b}) = \text{sk} \leftarrow \text{SK}(\text{sk}_b, g^{\text{sk}_a}) \]
REMINDER: ONE-TIME PAD

\[
c \leftarrow m \oplus sk
\]

\[
m \leftarrow c \oplus sk
\]
REMINDER: ONE-TIME PAD

\[
(m \oplus sk) \oplus sk = m \oplus (sk \oplus sk) = m
\]
HOW TO COMBINE
HOW TO COMBINE

- We have key-exchange
HOW TO COMBINE

- We have key-exchange
- Returns secret key $sk_1$
HOW TO COMBINE

- We have key-exchange
  - Returns secret key $sk_1$
- ... and one-time pad
HOW TO COMBINE

- We have key-exchange
  - Returns secret key $sk_1$
- ... and one-time pad
  - Returns ciphertext, given $sk_2$
HOW TO COMBINE

- We have key-exchange
  - Returns secret key $sk_1$
- ... and one-time pad
  - Returns ciphertext, given $sk_2$
- However, they work in different domains
HOW TO COMBINE

- We have key-exchange
  - Returns secret key $sk_1$
- ... and one-time pad
  - Returns ciphertext, given $sk_2$
- However, they work in different domains
  - $sk_1$ is a group element, $sk_2$ is a bitstring
ONE-TIME PAD IN A GROUP

\[ c \leftarrow m \cdot sk \]

\[ m \leftarrow c / sk \]
ONE-TIME PAD IN A GROUP

$\text{sk, } m \in G$

$c \leftarrow m \cdot \text{sk}$

$m \leftarrow c / \text{sk}$

Quiz: what property of groups is needed for decryption to work?
ONE-TIME PAD IN A GROUP

$sk, m \in G$

$c \leftarrow m \cdot sk$

$m^*$

$m \leftarrow c / sk$

Quiz: what property of groups is needed for decryption to work?
If $sk$ is uniformly distributed in $G$, then $m \cdot sk$ is also uniformly distributed.
If $sk$ is uniformly distributed in $G$, then $m \cdot sk$ is also uniformly distributed.

We assume $G$ does not contain $0$. 
If \( sk \) is uniformly distributed in \( G \), then \( m \cdot sk \) is also uniformly distributed.

We assume \( G \) does not contain \( 0 \).

For all \( m \in G \), \( c \in G \):

\[
\Pr_{sk}[m \cdot sk = c] = \Pr_{sk}[sk = c/m] = 1/q
\]

Thus, an adversary who only sees \( c \), has no information about \( m \).
ELGAMAL ENCRYPTION

\[ sk_b, pk_b = g^{sk_b} \]
ELGAMAL ENCRYPTION

public key $pk_b$

Long-time keys

$sk_b, pk_b = g^{sk_b}$
ELGAMAL ENCRYPTION

One-time keys $r, g^r, m$

Public key $pk_b$

$sk = SK(r, pk_b)$

Long-time keys $sk_b, pk_b = g^{sk_b}$
ELGAMAL ENCRYPTION

One-time keys $r, g^r, m$

Long-time keys

Public key $pk_b$

$pk_b$ ↔ $sk = SK(r, pk_b)$

$(c_1, c_2) \leftarrow (m \cdot sk, g^r)$
ELGAMAL ENCRYPTION

One-time keys

\[ r, g^r, m \]

Long-time keys

\[ \text{public key } pk_b \]

\[ sk = SK(r, pk_b) \]

\[ (c_1, c_2) \leftarrow (m \cdot sk, g^r) \]

\[ sk \leftarrow SK(sk_b, g^r) \]

\[ m \leftarrow c_1 / sk \]
**ELGAMAL ENCRYPTION**

**Elgamal.Setup (κ):**
1. Choose a group $G$ of order $q$ where breaking DDH has complexity $2^\kappa$
2. Choose a generator $g$ of $G$
3. Return $g^k \leftarrow \text{desc}(G) = (... , q , g)$

**Public key $pk_b$**

$pk_b \leftarrow \text{Elgamal.Setup}(κ)$

$$\begin{align*}
(c_1 , c_2) &\leftarrow (m \cdot sk , g^r) \\
sk &\leftarrow \text{SK}(sk_b , g^r) \\
m &\leftarrow c_1 / sk
\end{align*}$$
**ELGAMAL ENCRYPTION**

**Elgamal.Setup (κ):**
1. Choose a group $G$ of order $q$ where breaking DDH has complexity $2^κ$
2. Choose a generator $g$ of $G$
3. Return $g^k ← \text{desc} (G) = (\ldots, q, g)$

**Elgamal.Keygen (gk):**
1. $sk ← \mathbb{Z}_q$
2. $pk = b ← g^{sk}$
3. Return $(sk, pk)$ // secret key, public key

$(c_1, c_2) ← (m \cdot sk, gr)$

$sk ← \text{SK}(sk_b, g^r)$

$m ← c_1 / sk$
**ELGAMAL ENCRYPTION**

**Elgamal.Setup (κ):**
1. Choose a group $G$ of order $q$ where breaking DDH has complexity $2^κ$
2. Choose a generator $g$ of $G$
3. Return $g^k ← \text{desc}(G) = (... , q , g)$

**Elgamal.Keygen (gk):**
1. $sk ← \mathbb{Z}_q$
2. $pk = b ← g^{sk}$
3. Return $(sk, pk)$ // secret key, public key

**Elgamal.Enc_{gk, pk} (m; r):**
1. // Assumes $r ← \mathcal{R}$: randomized alg.
2. $(c_1, c_2) ← (m · pk^r , g^r)$
3. Return $(c_1 , c_2)$

$m ← c_1 / sk$

$sk ← SK(sk_b, g^r)$
**ELGAMAL ENCRYPTION**

---

**Elgamal.Setup (κ):**
1. Choose a group $G$ of order $q$ where breaking DDH has complexity $2^κ$
2. Choose a generator $g$ of $G$
3. Return $g^k ← \text{desc} (G) = (... , q, g)$

**Elgamal.Keygen (gk):**
1. $sk ← \mathbb{Z}_q$
2. $pk = b ← g^{sk}$
3. Return $(sk, pk)$ // secret key, public key

**Elgamal.Enc_{gk, pk} (m; r):**
1. // Assumes $r ← R$: randomized alg.
2. $(c_1, c_2) ← (m \cdot pk^r, g^r)$
3. Return $(c_1, c_2)$

**Elgamal.Dec_{gk, sk} (c_1, c_2):**
1. $m ← c_1 / c_2^{sk}$
2. Return $m$
ELGAMAL ENCRYPTION

Elgamal.Setup ($\kappa$):
1. Choose a group $G$ of order $q$ where breaking DDH has complexity $2^\kappa$
2. Choose a generator $g$ of $G$
3. Return $g^k \leftarrow \text{desc}(G) = (\ldots, q, g)$

Elgamal.Keygen ($g^k$):
1. $sk \leftarrow \mathbb{Z}_q$
2. $pk = h \leftarrow g^{sk}$
3. Return $(sk, pk)$ // secret key, public key

Elgamal.Enc$_{g^k, pk}$ ($m; r$):
1. // Assumes $r \leftarrow R$: randomized alg.
2. $(c_1, c_2) \leftarrow (m \cdot pk^r, g^r)$
3. Return $(c_1, c_2)$

Elgamal.Dec$_{g^k, sk}$ ($c_1, c_2$):
1. $m \leftarrow c_1 / c_2^{sk}$
2. Return $m$

For the sake of simplicity, we will assume that $g^k$ is a fixed system parameter, shared by all participants. Will not mention $g^k$ unless explicitly needed.
CORRECTNESS: FORMALLY
CORRECTNESS: FORMALLY

- Key generation: \( pk = g^{sk} \)
Key generation: \( pk = g^{sk} \)

Encryption: \((c_1, c_2) = \text{Enc}_{pk} (m; r) = (m \cdot pk^r, g^r)\)
**CORRECTNESS: FORMALLY**

- **Key generation:** $pk = g^{sk}$
- **Encryption:** $(c_1, c_2) = Enc_{pk}(m; r) = (m \cdot pk^r, g^r)$
- **Decryption:** $c_1 / c_2^{sk} = m \cdot g^r \cdot sk / g^r \cdot sk = m$
CORRECTNESS: FORMALLY

- Key generation: \( pk = g^{sk} \)
- Encryption: \((c_1, c_2) = \text{Enc}_{pk} (m; r) = (m \cdot pk^r, g^r)\)
- Decryption: \( c_1 / c_2^{sk} = m \cdot g^r \cdot sk / g^r \cdot sk = m \)
- Thus decryption is always successful
SECURITY: INTUITION
SECURITY: INTUITION

- Elgamal = DHKE + one-time pad
SECURITY: INTUITION

- Elgamal = DHKE + one-time pad
- DHKE is whatke-secure under X assumption
SECURITY: INTUITION

- Elgamal = DHKE + one-time pad
- DHKE is *whatke*-secure under X assumption
- one-time pad is unconditionally secure
SECURITY: INTUITION

- Elgamal = DHKE + one-time pad
  - DHKE is whatke-secure under X assumption
  - one-time pad is unconditionally secure
- => Elgamal is whatenc-secure under X assumption
SECURITY: INTUITION

- Elgamal = DHKE + one-time pad
  - DHKE is whatke-secure under $X$ assumption
  - one-time pad is unconditionally secure
  - $\Rightarrow$ Elgamal is whatenc-secure under $X$ assumption
- Intuitively, $X$ (=DDH) should suffice for the security of Elgamal
Elgamal = DHKE + one-time pad
- DHKE is whatke-secure under \( X \) assumption
- one-time pad is unconditionally secure
- \( \Rightarrow \) Elgamal is whatenc-secure under \( X \) assumption
- Intuitively, \( X \) (=DDH) should suffice for the security of Elgamal
SECURITY: INTUITION

- Elgamal = DHKE + one-time pad
  - DHKE is whatke-secure under $X$ assumption
  - one-time pad is unconditionally secure
  - $\Rightarrow$ Elgamal is whatenc-secure under $X$ assumption
- Intuitively, $X$ (=DDH) should suffice for the security of Elgamal

But how to formalize?
ELGAMAL ENCRYPTION

One-time keys

\[ r, g^r, m \]

Public key \( pk_b \)

Long-time keys

\[ sk_b, pk_b = g^{sk_b} \]

\[ sk = SK(r, pk_b) \]

\[ (c_1, c_2) \leftarrow (m \cdot sk, g^r) \]

\[ sk \leftarrow SK(sk_b, g^r) \]

\[ m \leftarrow c_1/sk \]
ELGAMAL ENCRYPTION

public keys \( pk_b \)

One-time keys

\( r, g^r, m \)

Long-time keys

\( sk_b, pk_b = g^{sk_b} \)

\( sk = SK(r, pk) \)

\( (c_1, c_2) \leftarrow (m \cdot sk, g^r) \)

\( sk \leftarrow SK(sk_b, g^r) \)

\( m \leftarrow c_1 / sk \)
ENCRYPTION SECURITY: ISSUES
Message recovery security:
ENCRYPTION SECURITY: ISSUES

❖ **Message recovery security:**
  ❖ Eve should not be able to compute \( m \)
ENCRYPTION SECURITY: ISSUES

- **Message recovery security:**
  - Eve should not be able to compute \( m \)
  - Just difficulty of computing \( m \) is not good enough
ENCRIPTION SECURITY: ISSUES

❖ **Message recovery security:**
  ❖ Eve should not be able to compute $m$
  ❖ Just difficulty of computing $m$ is not good enough
  ❖ Like KR security was not enough in KE
ENCRIPTION SECURITY: ISSUES

✧ **Message recovery security:**
  ✧ Eve should not be able to compute $m$
  ✧ Just difficulty of computing $m$ is not good enough
  ✧ Like KR security was not enough in KE
  ✧ Need **indistinguishability** again
ELGAMAL ENCRYPTION

\[ (c_1, c_2) \leftarrow (m \cdot \text{sk}, g^r) \]

\[ \text{sk} \leftarrow \text{SK}(\text{sk}_b, g^r) \]

\[ m \leftarrow c_1/\text{sk} \]

\[ m^* = m_0 \text{ or } m^* = m_1 \]
IND-CPA SECURITY GAME

\[ A_{\text{INDCPA}} (G) \]
IND-CPA SECURITY GAME

\[ g \leftarrow \$ G \setminus \{1\} \]
\[ sk \leftarrow \$ \mathbb{Z}_q \]
\[ pk \leftarrow g^{sk} \]

**Chal\textsuperscript{INDCPA} (G)**
**IND-CPA SECURITY GAME**

\[ \text{Chal}_{\text{INDCPA}} (G) \]

\[ g \leftarrow G \setminus \{1\} \]
\[ \text{sk} \leftarrow \mathbb{Z}_q \]
\[ \text{pk} \leftarrow g^{\text{sk}} \]

\[ \mathcal{A}_{\text{INDCPA}} (G) \]

\[ (g, \text{pk}) \]
IND-CPA SECURITY GAME

Chal_{INDCPA} (G)

$g \leftarrow \mathbb{G} \setminus \{1\}$

$sk \leftarrow \mathbb{Z}_q$

$pk \leftarrow g^{sk}$

$A_{INDCPA} (G)$

$(g, pk) \quad \rightarrow \quad (m_0, m_1) \leftarrow A(G, g, pk)$
**IND-CPA Security Game**

- **Challenge** \( \text{Chal}_{\text{INDCPA}} (G) \):
  - \( g \leftarrow \$ G \setminus \{1\} \)
  - \( \text{sk} \leftarrow \$ \mathbb{Z}_q \)
  - \( \text{pk} \leftarrow g^\text{sk} \)

- **Adversary** \( \mathcal{A}_{\text{INDCPA}} (G) \):
  - \( (m_0, m_1) \leftarrow \mathcal{A}(G, g, \text{pk}) \)
IND-CPA SECURITY GAME

\[ g \leftarrow \$ G \setminus \{1\} \]
\[ sk \leftarrow \$ \mathbb{Z}_q \]
\[ pk \leftarrow g^{sk} \]
\[ \beta \leftarrow \$ \{0, 1\} \]
\[ r \leftarrow \$ \mathbb{Z}_q \]

(g, pk) \rightarrow (m_0, m_1) \leftarrow A(G, g, pk)
IND-CPA SECURITY GAME

\[\text{Chal}_{\text{INDCPA}} (G)\]

- \(g \leftarrow \mathbb{G} \setminus \{1\}\)
- \(\text{sk} \leftarrow \mathbb{Z}_q\)
- \(\text{pk} \leftarrow g^{\text{sk}}\)
- \(\beta \leftarrow \{0, 1\}\)
- \(r \leftarrow \mathbb{Z}_q\)

\[\mathcal{A}_{\text{INDCPA}} (G)\]

- \((g, \text{pk})\)
- \((m_0, m_1) \leftarrow \mathcal{A}(G, g, \text{pk})\)
- \(c = (m_\beta \cdot \text{pk}^r, g^r)\)
IND-CPA SECURITY GAME

$g \leftarrow \$ G \setminus \{1\}$
$\text{sk} \leftarrow \$ \mathbb{Z}_q$
$pk \leftarrow g^{\text{sk}}$

$\beta \leftarrow \$ \{0, 1\}$
$r \leftarrow \$ \mathbb{Z}_q$

$(g, pk)$

$(m_0, m_1) \leftarrow \mathcal{A}(G, g, pk)$
$c = (m_\beta \cdot pk^r, g^r)$

$\beta^* \leftarrow \mathcal{A}(G, g, pk, c)$
IND-CPA SECURITY GAME

\[ \text{Chal}_{\text{INDCPA}} (G) \]

- \( g \leftarrow \$ G \setminus \{1\} \)
- \( \text{sk} \leftarrow \$ \mathbb{Z}_q \)
- \( \text{pk} \leftarrow g^{\text{sk}} \)

- \( \beta \leftarrow \$ \{0, 1\} \)
- \( r \leftarrow \$ \mathbb{Z}_q \)

\[ \text{A}_{\text{INDCPA}} (G) \]

- \( (g, \text{pk}) \)
- \( (m_0, m_1) \leftarrow \text{A} (G, g, \text{pk}) \)

- \( c = (m_\beta \cdot \text{pk}^r, g^r) \)
- \( \beta^* \leftarrow \text{A} (G, g, \text{pk}, c) \)
IND-CPA SECURITY GAME

Challenger (G):

\[ g \leftarrow \$ G \setminus \{1\} \]
\[ sk \leftarrow \$ \mathbb{Z}_q \]
\[ pk \leftarrow g^{sk} \]
\[ \beta \leftarrow \$ \{0, 1\} \]
\[ r \leftarrow \$ \mathbb{Z}_q \]

if \( \beta = \beta^* \):
\[ d \leftarrow 1 \]
else:
\[ d \leftarrow 0 \]

\( (g, pk) \)

\( (m_0, m_1) \leftarrow \mathcal{A}(G, g, pk) \)

\( c = (m_\beta \cdot pk^r, g^r) \)

\( \beta^* \leftarrow \mathcal{A}(G, g, pk, c) \)
IND-CPA SECURITY GAME

**Chal**\textsubscript{INDCPA} \((G)\)

\[ g \leftarrow \{1\} \backslash \{0\} \]
\[ sk \leftarrow \mathbb{Z}_q \]
\[ pk \leftarrow g^{sk} \]
\[ \beta \leftarrow \{0, 1\} \]
\[ r \leftarrow \mathbb{Z}_q \]

if \(\beta = \beta^*\):
\[ d \leftarrow 1 \]
else:
\[ d \leftarrow 0 \]

**A**\textsubscript{INDCPA} \((G)\)

\[ (g, pk) \leftarrow \]
\[ (m_0, m_1) \leftarrow A(G, g, pk) \]
\[ c = (m_\beta \cdot pk^r, g^r) \]
\[ \beta^* \leftarrow A(G, g, pk, c) \]
IND-CPA SECURITY
IND-CPA SECURITY

\[ \Pi = (\text{Setup, Keygen, Enc, Dec}) \]
\[ \Pi = (\text{Setup}, \text{Keygen}, \text{Enc}, \text{Dec}) \]

**Game INDCPA}_{\Pi, \mathcal{A}(\kappa)}**

- \( gk \leftarrow \text{Setup}(1^\kappa) \)
- \((sk, pk) \leftarrow \text{Keygen}(gk)\)
- \((m_0, m_1) \leftarrow \mathcal{A}(gk, pk)\)
- \(\beta \leftarrow \{0, 1\}\)
- \(r \leftarrow \mathbb{Z}_q\)
- \(c \leftarrow \text{Enc}_{pk}(m_\beta; r)\)
- \(\beta^* \leftarrow \mathcal{A}(gk, pk, c)\)
- Return \(\beta = \beta^* \oplus 1\)
**IND-CPA SECURITY**

\[ \Pi = (\text{Setup, Keygen, Enc, Dec}) \]

\[
\text{Adv}^{\text{INDCPA}}_{\Pi, \mathcal{A}}(\kappa) := 2 \cdot \left| \Pr[\text{INDCPA}_{\Pi, \mathcal{A}}(\kappa) = 1] - \frac{1}{2} \right|
\]

**Game INDCPA_{\Pi, \mathcal{A}}(\kappa)**

- \( gk \leftarrow \text{Setup}(\kappa) \)
- \( (sk, pk) \leftarrow \text{Keygen}(gk) \)
- \( (m_0, m_1) \leftarrow \mathcal{A}(gk, pk) \)
- \( \beta \leftarrow \{0, 1\} \)
- \( r \leftarrow \mathbb{Z}_q \)
- \( c \leftarrow \text{Enc}_{pk}(m_\beta; r) \)
- \( \beta^* \leftarrow \mathcal{A}(gk, pk, c) \)
- \text{Return } \beta = \beta^* \oplus 1 \oplus 0
IND-CPA SECURITY

- $\Pi = (\text{Setup}, \text{Keygen}, \text{Enc}, \text{Dec})$
  
  $\text{Adv}_{\Pi, A}^{\text{IND-CPA}}(\kappa) := 2 \cdot \left| \Pr[\text{INDCPA}_{\Pi, A}(\kappa) = 1] - 1/2 \right|$

- $A(\tau, \varepsilon)$-breaks IND-CPA security of $\Pi$ iff $A$ runs in time $\leq \tau$ and $\text{Adv}_{\Pi, A}^{\text{IND-CPA}}(\kappa) \geq \varepsilon$

**Game $\text{INDCPA}_{\Pi, A}(\kappa)$**

- $gk \leftarrow \text{Setup}(\kappa)$
- $(sk, pk) \leftarrow \text{Keygen}(gk)$
- $(m_0, m_1) \leftarrow A(gk, pk)$
- $\beta \leftarrow \$ \{0, 1\}$
- $r \leftarrow \$ \mathbb{Z}_q$
- $c \leftarrow \text{Enc}_{pk}(m_\beta; r)$
- $\beta^* \leftarrow A(gk, pk, c)$
- Return $\beta = \beta^* ? 1 : 0$
IND-CPA SECURITY

- $\Pi = (\text{Setup, Keygen, Enc, Dec})$
  \[ \text{Adv}_{\Pi,A}(\kappa) := 2 \cdot \left| \Pr[\text{INDCPA}_{\Pi,A}(\kappa) = 1] - 1/2 \right| \]

- $A(\tau, \varepsilon)$-breaks IND-CPA security of $\Pi$ iff $A$ runs in time $\leq \tau$ and
  \[ \text{Adv}_{\Pi,A}(\kappa) \geq \varepsilon \]

- $\Pi$ is $(\tau, \varepsilon)$-IND-CPA secure iff no adversary $(\tau, \varepsilon)$-breaks IND-CPA security of $\Pi$

Game $\text{INDCPA}_{\Pi,A}(\kappa)$

- $gk \leftarrow \text{Setup}(\kappa)$
- $(sk, pk) \leftarrow \text{Keygen}(gk)$
- $(m_0, m_1) \leftarrow A(gk, pk)$
- $\beta \leftarrow \{0, 1\}$
- $r \leftarrow \mathbb{Z}_q$
- $c \leftarrow \text{Enc}_{pk}(m_\beta; r)$
- $\beta^* \leftarrow A(gk, pk, c)$
- Return $\beta = \beta^* ? 1 : 0$
IND-CPA SECURITY

- $\Pi = (\text{Setup}, \text{Keygen}, \text{Enc}, \text{Dec})$
  
  $\text{Adv}_{\Pi, A}^{\text{IND-CPA}}(\kappa) := 2 \cdot \left| \Pr[\text{INDCPA}_{\Pi, A}(\kappa) = 1] - 1/2 \right|

- $A(\tau, \varepsilon)$-breaks IND-CPA security of $\Pi$ iff $A$ runs in time $\leq \tau$ and
  $\text{Adv}_{\Pi, A}^{\text{IND-CPA}}(\kappa) \geq \varepsilon$

- $\Pi$ is $(\tau, \varepsilon)$-IND-CPA secure iff no adversary $(\tau, \varepsilon)$-breaks IND-CPA security of $\Pi$

- $\Pi$ is IND-CPA secure iff it is $(\text{poly}(\kappa), \text{negl}(\kappa))$-IND-CPA secure

---

Game $\text{INDCPA}_{\Pi, A}(\kappa)$

- $gk \leftarrow \text{Setup}(1^\kappa)$
- $(\text{sk}, \text{pk}) \leftarrow \text{Keygen}(gk)$
- $(m_0, m_1) \leftarrow A(gk, pk)$
- $\beta \leftarrow \{0, 1\}$
- $r \leftarrow \mathbb{Z}_q$
- $c \leftarrow \text{Enc}_{\text{pk}}(m_\beta; r)$
- $\beta^* \leftarrow A(gk, pk, c)$
- Return $\beta = \beta^* \ ? \ 1 : 0$
ELGAMAL IS IND-CPA SECURE

**Theorem.** If $G$ is a $(\approx\tau, \approx\varepsilon)$-DDH group, then Elgamal is $(\tau, \varepsilon)$-IND-CPA secure in $G$. 
Theorem. If $G$ is a $(\approx\tau, \approx\varepsilon)$-DDH group, then Elgamal is $(\tau, \varepsilon)$-IND-CPA secure in $G$.

Proof idea. Reduction to absurd: we show that if Elgamal is not secure in $G$, then DDH must be easy in $G$. 
Theorem. If $G$ is a $(\approx \tau, \approx \varepsilon)$-DDH group, then Elgamal is $(\tau, \varepsilon)$-IND-CPA secure in $G$.

Proof idea. Reduction to absurd: we show that if Elgamal is not secure in $G$, then DDH must be easy in $G$.

- Elgamal is not secure $\Rightarrow$ there exists an adversary $\mathcal{D}$ that breaks Elgamal
**Theorem.** If $G$ is a $(\approx \tau, \approx \varepsilon)$-DDH group, then Elgamal is $(\tau, \varepsilon)$-IND-CPA secure in $G$.

**Proof idea.** Reduction to absurd: we show that if Elgamal is not secure in $G$, then DDH must be easy in $G$.

- Elgamal is not secure $\Rightarrow$ there exists an adversary $\mathcal{D}$ that breaks Elgamal
- We show DDH is easy by constructing an adversary $\mathcal{C}$ that breaks DDH
Theorem. If $G$ is a $(\approx \tau, \approx \varepsilon)$-DDH group, then Elgamal is $(\tau, \varepsilon)$-IND-CPA secure in $G$.

Proof idea. Reduction to absurd: we show that if Elgamal is not secure in $G$, then DDH must be easy in $G$.

- Elgamal is not secure $\Rightarrow$ there exists an adversary $\mathcal{D}$ that breaks Elgamal
- We show DDH is easy by constructing an adversary $\mathcal{C}$ that breaks DDH
- $\mathcal{C}$ can use help from adversary $\mathcal{D}$, by sending inputs to $\mathcal{D}$ and receiving outputs
**Theorem.** If $G$ is a $(\approx \tau, \approx \varepsilon)$-DDH group, then Elgamal is $(\tau, \varepsilon)$-IND-CPA secure in $G$.

**Proof idea.** Reduction to absurd: we show that if Elgamal is not secure in $G$, then DDH must be easy in $G$.

- Elgamal is not secure $\Rightarrow$ there exists an adversary $D$ that breaks Elgamal
- We show DDH is easy by constructing an adversary $C$ that breaks DDH
- $C$ can use help from adversary $D$, by sending inputs to $D$ and receiving outputs

Simple, a home exercise
Theorem. If Elgamal is $(\tau + \text{small}, \varepsilon)$-IND-CPA secure, then $G$ is $(\tau, \varepsilon)$-DDH group. (Thus, equivalent)
**Theorem.** If Elgamal is \((\tau + \text{small}, \varepsilon)-\text{IND-CPA}\) secure, then \(G\) is \((\tau, \varepsilon)-\text{DDH group}\). (Thus, equivalent)

**Proof idea.** Reduction to absurd: we show that if DDH is not secure in \(G\), then breaking Elgamal must be easy in \(G\).
Theorem. If Elgamal is $(\tau + \text{small}, \varepsilon)$-IND-CPA secure, then $G$ is $(\tau, \varepsilon)$-DDH group. (Thus, equivalent)

Proof idea. Reduction to absurd: we show that if DDH is not secure in $G$, then breaking Elgamal must be easy in $G$.

- DDH is not secure $\Rightarrow$ there exists an adversary $D$ that breaks DDH
Theorem. If Elgamal is \((\tau + \text{small}, \varepsilon)\)-IND-CPA secure, then \(G\) is \((\tau, \varepsilon)\)-DDH group. (Thus, equivalent)

Proof idea. Reduction to absurd: we show that if DDH is not secure in \(G\), then breaking Elgamal must be easy in \(G\).

- DDH is not secure \(\Rightarrow\) there exists an adversary \(D\) that breaks DDH
- Show Elgamal is easy by constructing an adversary \(C\) that breaks Elgamal
Theorem. If Elgamal is \((\tau + \text{small}, \varepsilon)\)-IND-CPA secure, then \(G\) is \((\tau, \varepsilon)\)-DDH group. (Thus, equivalent)

Proof idea. Reduction to absurd: we show that if DDH is not secure in \(G\), then breaking Elgamal must be easy in \(G\).

- DDH is not secure => there exists an adversary \(D\) that breaks DDH
- Show Elgamal is easy by constructing an adversary \(C\) that breaks Elgamal
- \(C\) uses help from adversary \(D\), sending inputs to \(D\) and receiving outputs
MAIN IDEA
Recall: \( pk = g^{sk} \), \((c_1, c_2) = Enc_{pk}(m; r) = (m \cdot pk^r, g^r)\)
Recall: \( pk = g^{sk} \), \((c_1, c_2) = \text{Enc}_{pk} (m; r) = (m \cdot pk^r, g^r) \)

Hence \((g, pk, c_2, c_1) = (g, g^{sk}, g^r, m \cdot g^r \cdot sk)\)
Recall: $pk = g^{sk}$, $(c_1, c_2) = Enc_{pk}(m; r) = (m \cdot pk^r, g^r)$

Hence $(g, pk, c_2, c_1) = (g, g^{sk}, g^r, m \cdot g^r \cdot sk)$

If $m = 1$: $(g, pk, c_2, c_1)$ is a random DDH tuple
Recall: \( pk = g^{sk}, (c_1, c_2) = \text{Enc}_{pk}(m; r) = (m \cdot pk^r, g^r) \)

Hence \((g, pk, c_2, c_1) = (g, g^{sk}, g^r, m \cdot g^r \cdot sk)\)

If \( m = 1 \): \((g, pk, c_2, c_1)\) is a random DDH tuple

If \( m = \text{random} \):
Recall: \( pk = g^{sk}, (c_1, c_2) = \text{Enc}_{pk} (m; r) = (m \cdot pk^r, g^r) \)

Hence \((g, pk, c_2, c_1) = (g, g^{sk}, g^r, m \cdot g^r \cdot sk)\)

If \( m = 1 \): \((g, pk, c_2, c_1)\) is a random DDH tuple

If \( m = \text{random} \):

\((g, pk, c_2, c_1)\) is a random tuple

since random * anything = random in cyclic group
Recall: $pk = g^{sk}, \ (c_1, c_2) = \text{Enc}_{pk} (m; r) = (m \cdot pk^r, g^r)$

Hence $(g, pk, c_2, c_1) = (g, g^{sk}, g^r, m \cdot g^r \cdot sk)$

If $m = 1$: $(g, pk, c_2, c_1)$ is a random DDH tuple

If $m$ = random:

$(g, pk, c_2, c_1)$ is a random tuple

DDH assumption: indistinguishable

since random * anything = random in cyclic group
REDUCTION: DDH $\rightarrow$ ELGAMAL

\text{Challenger}_{\text{Elgamal}} (G) \quad \exists$

\text{To be constructed}

\mathcal{D}_{\text{DDH}} (G) \quad \exists$

// Black box
**REDUCTION: DDH → ELGAMAL**

<table>
<thead>
<tr>
<th>Challenger$_{\text{Elgamal}}$ ($G$)</th>
<th>$C_{\text{Elgamal}}$ ($G$)</th>
<th>$D_{\text{DDH}}$ ($G$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>To be constructed</td>
<td>// Black box</td>
</tr>
</tbody>
</table>

Exists

To be constructed
**REDUCTION: DDH $\rightarrow$ ELGAMAL**

<table>
<thead>
<tr>
<th>Challenger$_{Ellamal} (G)$</th>
<th>$C_{Ellamal} (G)$</th>
<th>$D_{DDH} (G)$</th>
</tr>
</thead>
</table>
| $g \leftarrow_{S} G \setminus \{1\}$ | \[
\text{To be constructed}
\] | \[
// Black box
\] |
| $sk \leftarrow_{S} \mathbb{Z}_q$ |
| $pk \leftarrow g^{sk}$ |

Exists

To be constructed

Exists
**REDUCTION: DDH → ELGAMAL**

Challenger\_Elgamal (G)

\[\begin{align*}
g & \leftarrow_G G \setminus \{1\} \\
\text{sk} & \leftarrow \mathbb{Z}_q \\
\text{pk} & \leftarrow g^{\text{sk}}
\end{align*}\]

C\_Elgamal (G)

_EXISTS_

D\_DDH (G)

// Black box

_EXISTS_

**To be constructed**
REDUCTION: DDH $\rightarrow$ ELGAMAL

Challenger_{Elgamal} $(G)$

$g \leftarrow_{S} G \setminus \{1\}$
$sk \leftarrow_{S} \mathbb{Z}_q$
$pk \leftarrow g^{sk}$

$(g, pk) \rightarrow \mathcal{C}_{Elgamal} (G)$

$m_0 \leftarrow 1_G = g^0$
$m_1 \leftarrow_{S} \mathbb{G}$

$\mathcal{D}_{DDH} (G)$

// Black box

Exists

To be constructed

Exists
REDUCTION: DDH $\rightarrow$ ELGAMAL

Challenger ${}_{Elgamal} (G)$

- $g \leftarrow G \setminus \{1\}$
- $sk \leftarrow \mathbb{Z}_q$
- $pk \leftarrow g^{sk}$

$\exists$

$\rightarrow$

$\rightarrow$

$C_{Elgamal} (G)$

- $(g, pk)$
- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow \mathbb{G}$

To be constructed

$\rightarrow$

$\rightarrow$

$\mathcal{D}_{DDH} (G)$

$\exists$

$\rightarrow$

$\rightarrow$

// Black box
REDUCTION: DDH $\rightarrow$ ELGAMAL

### Challenger$_{Elgamal}$ $(G)$
- $g \leftarrow$ $\mathbb{G}\setminus\{1\}$
- $sk \leftarrow$ $\mathbb{Z}_q$
- $pk \leftarrow g^{sk}$
- $\beta \leftarrow$ $\{0,1\}$
- $r \leftarrow$ $\mathbb{Z}_q$
- $(c_1, c_2) \leftarrow$ $\text{Enc}_{pk}(m_\beta; r)$

### $C_{Elgamal} (G)$
- $(g, pk)$
- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow$ $\mathbb{G}$

### $D_{DDH} (G)$
- $//$ Black box

### Exists
- To be constructed

### Exists
REDUCTION: DDH $\rightarrow$ ELGAMAL

**Challenger**_{Elgamal} ($G$)

- $g \leftarrow G \setminus \{1\}$
- $sk \leftarrow \mathbb{Z}_q$
- $pk \leftarrow g^{sk}$
- $\beta \leftarrow \{0,1\}$
- $r \leftarrow \mathbb{Z}_q$
- $(c_1, c_2) \leftarrow \text{Enc}_{pk}(m_\beta; r)$

**$C_{Elgamal}$ ($G$)**

- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow \mathbb{G}$

**$D_{DDH}$ ($G$)**

- // Black box

Exists

To be constructed

Exists
REDUCTION: DDH $\rightarrow$ ELGAMAL

Challenger\text{Elgamal} $(G)$

$g \leftarrow \$ G\setminus\{1\}$
$\text{sk} \leftarrow \$ \mathbb{Z}_q$
$\text{pk} \leftarrow g^{\text{sk}}$
$\beta \leftarrow \$ \{0,1\}$
$r \leftarrow \$ \mathbb{Z}_q$
$(c_1, c_2) \leftarrow \text{Enc}_{\text{pk}}(m_\beta; r)$

$\mathcal{C}_{\text{Elgamal}} (G)$

$m_0 \leftarrow 1_G = g^0$
$m_1 \leftarrow \$ \mathbb{G}$
$c = (c_1, c_2)$

$\mathcal{D}_{\text{DDH}} (G)$

// Black box

$(g, \text{pk}, c_2, c_1)$

Exists

To be constructed

Exists
**REDUCTION: DDH → ELGAMAL**

**Challenger**

\[ g \leftarrow G \setminus \{1\} \]
\[ \text{sk} \leftarrow \mathbb{Z}_q \]
\[ \text{pk} \leftarrow g^\text{sk} \]
\[ \beta \leftarrow \{0,1\} \]
\[ r \leftarrow \mathbb{Z}_q \]
\[ (c_1, c_2) \leftarrow \text{Enc}_{\text{pk}}(m_\beta; r) \]

**C**

\[ m_0 \leftarrow 1_G = g^0 \]
\[ m_1 \leftarrow \$ \mathcal{G} \]

**D**

\[ \beta' \]

// Black box

**Exists**

**To be constructed**

**Exists**
REDUCTION: DDH $\rightarrow$ ELGAMAL

**Challenger**$_{\text{Elgamal}}$ $(G)$
- $g \leftarrow$ $G \setminus \{1\}$
- $\text{sk} \leftarrow$ $\mathbb{Z}_q$
- $\text{pk} \leftarrow g^{\text{sk}}$
- $\beta \leftarrow$ $\{0,1\}$
- $r \leftarrow$ $\mathbb{Z}_q$
- $(c_1, c_2) \leftarrow \text{Enc}_{\text{pk}}(m_\beta; r)$

**$C_{\text{Elgamal}}$ $(G)$**
- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow$ $\mathbb{G}$
- $(m_0, m_1)$

**$D_{\text{DDH}}$ $(G)$**
- $\beta'$
- $(g, \text{pk}, c_2, c_1)$
- $\beta'$

Exists

To be constructed

Exists

// Black box
REDUCTION: DDH → ELGAMAL

**Challenger**_{Elgamal} \((G)\)

\[
g \leftarrow G \setminus \{1\} \\
\text{sk} \leftarrow \mathbb{Z}_q \\
pk \leftarrow g^{\text{sk}} \\
\beta \leftarrow \{0, 1\} \\
r \leftarrow \mathbb{Z}_q \\
(c_1, c_2) \leftarrow \text{Enc}_{pk}(m_\beta; r)
\]

**\(C_{Elgamal}\) (G)**

\[
m_0 \leftarrow 1^G = g^0 \\
m_1 \leftarrow \$ \ G
\]

\[
(c_1, c_2) \leftarrow \text{Enc}_{pk}(m_\beta; r)
\]

\[
\beta' = 0 \\
\text{if } \beta' = \beta: \\
\text{else: } d \leftarrow 0
\]

\[
\beta' \leftarrow \text{Black box}
\]

\[
(g, pk, c_2, c_1)
\]

\[
\exists (m_0, m_1)
\]

\[
\exists (c_1, c_2)
\]

\[
\exists \beta'
\]
REDUCTION: DDH ⟷ ELGAMAL

Challenger\textsubscript{Elgamal} \((G)\)

\begin{align*}
g &\leftarrow \_ G \setminus \{1\} \\
\text{sk} &\leftarrow \_ \mathbb{Z}_q \\
\text{pk} &\leftarrow g^{\text{sk}} \\
\beta &\leftarrow \_ \{0,1\} \\
r &\leftarrow \_ \mathbb{Z}_q \\
(c_1, c_2) &\leftarrow \text{Enc}_{\text{pk}}(m_\beta; r)
\end{align*}

\text{if } \beta' = \beta:
\quad d \leftarrow 1
\quad \text{else: } d \leftarrow 0

\text{Exists}

\text{C}_{\text{Elgamal}} \((G)\)

\begin{align*}
m_0 &\leftarrow 1_G = g^0 \\
m_1 &\leftarrow \_ G \\
c &\leftarrow (c_1, c_2)
\end{align*}

\text{To be constructed}

\text{D}_{\text{DDH}} \((G)\)

\begin{align*}
(g, \text{pk}, c_2, c_1) &\leftarrow \\
\beta' &\leftarrow \\
\text{Exists}
\end{align*}

// Black box
Assume $D$ works in time $\tau$ and is successful with prob. $\varepsilon + 1/2$.

**Challenger** (Elgamal) $(G)$
- \( g \leftarrow_{S} G \setminus \{1\} \)
- \( sk \leftarrow_{S} \mathbb{Z}_q \)
- \( pk \leftarrow g^{sk} \)
- \( \beta \leftarrow_{S} \{0,1\} \)
- \( r \leftarrow_{S} \mathbb{Z}_q \)
- \((c_1, c_2) \leftarrow \text{Enc}_{pk}(m_\beta; r)\)
- \[ \text{if } \beta' = \beta: \]
  - \( d \leftarrow 1 \)
- \[ \text{else: } d \leftarrow 0 \]

**C** (Elgamal) $(G)$
- \( m_0 \leftarrow 1_G = g^0 \)
- \( m_1 \leftarrow_{S} G \)
- \( c = (c_1, c_2) \)

**D** (DDH) $(G)$
- \( (g, pk, c_2, c_1) \)
- \[ \text{if } \beta' = \beta: \]
  - \( d \leftarrow 1 \)
- \[ \text{else: } d \leftarrow 0 \]

Exists

To be constructed

Exists
**REDUCTION: DDH → ELGAMAL**

**Challenger** $\text{Elgamal}(G)$

- $g \leftarrow \mathbb{G}\setminus\{1\}$
- $sk \leftarrow \mathbb{Z}_q$
- $pk \leftarrow g^{sk}$
- $\beta \leftarrow \{0,1\}$
- $r \leftarrow \mathbb{Z}_q$
- $(c_1, c_2) \leftarrow \text{Enc}_{pk}(m_{\beta}; r)$

**C$\text{Elgamal}(G)$**

- $(g, pk)$
- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow \mathbb{G}$
- $c=(c_1, c_2)$

- $d \leftarrow 1$ if $\beta' = \beta$
- $d \leftarrow 0$ else

To be constructed

 EXISTS

**C$\text{Elgamal}(G)$**

- $(g, pk, c_2, c_1)$
- $\beta' = \beta$

// Black box

**Exist**

**Assume $D$ works in time $\tau$ and is successful with prob. $\epsilon + 1/2$**

**If $D$ is successful then $C$ is successful and takes time $\tau^*=\tau + \text{small}$**
REDUCTION: DDH $\rightarrow$ ELGAMAL

**Challenger**

- $g \leftarrow G \setminus \{1\}$
- $sk \leftarrow \mathbb{Z}_q$
- $pk \leftarrow g^{sk}$
- $\beta \leftarrow \{0, 1\}$
- $r \leftarrow \mathbb{Z}_q$
- $(c_1, c_2) \leftarrow Enc_{pk}(m_\beta; r)$

**C\text{Elgamal}(G)\)**

- $(g, pk) \leftarrow$ (g, pk)
- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow \$ \mathbb{G}$
- $c = (c_1, c_2)$
- $(g, pk, c_2, c_1) \leftarrow$ (g, pk, c_2, c_1)

**Exists**

- $\beta' = \beta$
- $d \leftarrow 1$

**To be constructed**

- $\exists$

**Exists**

- $d \leftarrow 0$

---

Assume $D$ works in time $\tau$ and is successful with prob. $\varepsilon + 1/2$

If $D$ is successful then $C$ is successful and takes time $\tau^* = \tau + $ small

If $D$ is unsuccessful then $C$ is successful with prob. 0, and takes time $\tau^*$

// Black box
**REDUCTION: DDH to ElGamal**

**Challenger**\text{Elgamal } (G)

- $g \leftarrow G \setminus \{1\}$
- $sk \leftarrow \mathbb{Z}_q$
- $pk \leftarrow g^{sk}$
- $\beta \leftarrow \{0,1\}$
- $r \leftarrow \mathbb{Z}_q$
- $(c_1, c_2) \leftarrow \text{Enc}_{pk}(m_0; r)$

**C\text{Elgamal } (G)**

- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow \$ G$
- $\beta' \leftarrow \{0,1\}$

**if** $\beta' = \beta$:
- $d \leftarrow 1$
**else:** $d \leftarrow 0$

**Exists**

**To be constructed**

**Exists**

\begin{align*}
\text{Assume } D & \text{ works in time } \tau \text{ and is successful with prob. } \varepsilon + 1/2 \\
\text{If } D \text{ is successful then } \text{C is successful and takes time } \tau^* = \tau + \text{small} \\
\text{If } D \text{ is unsuccessful then } \text{C is successful with prob. } 0, \text{ and takes time } \tau^* \\
\text{Pr} [\text{C successful}] & = \varepsilon + 1/2
\end{align*}
REDUCTION: DDH $\rightarrow$ ELGAMAL

Challenger$_{Elgamal}$ ($G$)

- $g \leftarrow G \setminus \{1\}$
- $sk \leftarrow \mathbb{Z}_q$
- $pk \leftarrow g^{sk}$
- $\beta \leftarrow \{0,1\}$
- $r \leftarrow \mathbb{Z}_q$
- $(c_1, c_2) \leftarrow Enc_{pk}(m_\beta; r)$

\[
\begin{align*}
d & \leftarrow 1 \\
\text{if } \beta' = \beta: & \\
\text{else: } d & \leftarrow 0
\end{align*}
\]

$\exists$ To be constructed

$C_{Elgamal}$ ($G$)

- $(g, pk)$
- $m_0 \leftarrow 1_G = g^0$
- $m_1 \leftarrow \$ \mathbb{G}$

\[
\begin{align*}
\beta' & \leftarrow \{0,1\} \\
\beta' & \leftarrow \mathbb{Z}/\mathbb{Z}_q \\
\beta' & \leftarrow \{0,1\}
\end{align*}
\]

$\exists$

Assume $D$ works in time $\tau$ and is successful with prob. $\varepsilon + 1/2$

If $D$ is successful then $C$ is successful and takes time $\tau^* = \tau + \text{small}$

If $D$ is unsuccessful then $C$ is successful with prob. 0, and takes time $\tau^*$

Pr[$C$ successful] = $\varepsilon + 1/2$

Thus, if Elgamal is $(\tau^*, \varepsilon)$-IND-CPA secure, then $G$ is a $(\tau, \varepsilon)$-DDH group
IND-CPA AND BEYOND

- Actual security requirements even more stringent
IND-CPA AND BEYOND

- Actual security requirements even more stringent
- IND-CPA security only demonstrates basic concepts
IND-CPA AND BEYOND

- Actual security requirements even more stringent
- IND-CPA security only demonstrates basic concepts
- Bob must be sure $c$ really came from Alice
IND-CPA AND BEYOND

- Actual security requirements even more stringent
- IND-CPA security only demonstrates basic concepts
- Bob must be sure $c$ really came from Alice
- Authentication, non-malleability
IND-CPA AND BEYOND

- Actual security requirements even more stringent
- IND-CPA security only demonstrates basic concepts
- Bob must be sure $c$ really came from Alice
- Authentication, non-malleability
- In protocol design, malleability is very useful
IND-CPA AND BEYOND

- Actual security requirements even more stringent
- IND-CPA security only demonstrates basic concepts
- Bob must be sure $c$ really came from Alice
- Authentication, non-malleability
- In protocol design, malleability is very useful

Crypto is full of trade-offs: do we want malleability or not?
MALLEABILITY
MALLEABILITY

\[ m \rightarrow f(m) \]
MALLEABILITY

Modifying ciphertexts, without knowing the secret key, so that the plaintext changes predictably.
MALLEABILITY

Modifying ciphertexts, *without* knowing the secret key, so that the plaintext changes predictably

Essential for secure computation
MALLEABILITY OF ELGAMAL
MALLEABILITY OF ELGAMAL

Recall: $\text{Enc}_{pk}(m; r) = (c_1, c_2) = (m \cdot pk^r, g^r)$
MALLEABILITY OF ELGAMAL

Recall: $\operatorname{Enc}_{pk}(m; r) = (c_1, c_2) = (m \cdot pk^r, g^r)$

$\operatorname{Enc}_{pk}(m_1; r_1) \cdot \operatorname{Enc}_{pk}(m_2; r_2) = \operatorname{Enc}_{pk}(m_1 \cdot m_2; r_1 + r_2)$

$(m_1 \cdot pk^{r_1}, g^{r_1}) \cdot (m_2 \cdot pk^{r_2}, g^{r_2}) = (m_1m_2 \cdot pk^{r_1+r_2}, g^{r_1+r_2})$
MALLEABILITY OF ELGAMAL

- Recall: $\text{Enc}_{pk} (m; r) = (c_1, c_2) = (m \cdot pk^r, g^r)$
- $\text{Enc}_{pk} (m_1; r_1) \cdot \text{Enc}_{pk} (m_2; r_2) = \text{Enc}_{pk} (m_1 \cdot m_2; r_1 + r_2)$
  - $(m_1 \cdot pk^{r_1}, g^{r_1}) \cdot (m_2 \cdot pk^{r_2}, g^{r_2}) = (m_1m_2 \cdot pk^{r_1+r_2}, g^{r_1+r_2})$

Componentwise multiplication
“RANDOMIZED ISOMORPHISM”

Isomorphism:  Homomorphic encryption:
“RANDOMIZED ISOMORPHISM”

**Isomorphism:**

\[ f(m_1) \cdot f(m_2) = f(m_1 \cdot m_2) \]

**Homomorphic encryption:**

\[ Enc(m_1; r_1) \cdot Enc(m_2; r_2) = Enc(m_1 \cdot m_2; r_1 + r_2) \]
**RANDOMIZED ISOMORPHISM**

**Isomorphism:**

\[ f(m_1)f(m_2) = f(m_1m_2) \]

\[ f(\bot) = \bot \]

**Homomorphic encryption:**

\[ \text{Enc}(m_1; r_1) \text{Enc}(m_2; r_2) = \text{Enc}(m_1m_2; r_1+r_2) \]

\[ \text{Enc}(\bot; o) = (\bot, o) \]
Isomorphism:

\[ f(m_1)f(m_2) = f(m_1m_2) \]

\[ f(1) = 1 \]

\[ f(m^{-1}) = \frac{1}{f(m)} \]

Homomorphic encryption:

\[ \text{Enc}(m_1; r_1) \text{Enc}(m_2; r_2) = \text{Enc}(m_1m_2; r_1+r_2) \]

\[ \text{Enc}(1; 0) = (1, 0) \]

\[ \text{Enc}(m^{-1}; -r_2) = \text{Enc}(1; 0) / \text{Enc}(m; r_2) \]
SOME COROLLARIES
Some Corollaries

Recall: $\text{Enc}(m_1; r_1) \cdot \text{Enc}(m_2; r_2) = \text{Enc}(m_1 \cdot m_2; r_1 + r_2)$
SOME COROLLARIES

- Recall: $\text{Enc}(m_1; r_1) \cdot \text{Enc}(m_2; r_2) = \text{Enc}(m_1 \cdot m_2; r_1 + r_2)$
- We omit $\text{pk}$ (public key) for brevity in what follows
SOME COROLLARIES

- Recall: \( \text{Enc} (m_1; r_1) \cdot \text{Enc} (m_2; r_2) = \text{Enc} (m_1 \cdot m_2; r_1 + r_2) \)
- We omit \( \text{pk} \) (public key) for brevity in what follows

1. \( \prod_i \text{Enc} (m_i; r_i) = \text{Enc} (\prod_i m_i; \Sigma_i r_i) \)
SOME COROLLARIES

- Recall: $\text{Enc}(m_1; r_1) \cdot \text{Enc}(m_2; r_2) = \text{Enc}(m_1 \cdot m_2; r_1 + r_2)$

- We omit $pk$ (public key) for brevity in what follows

1. $\prod_i \text{Enc}(m_i; r_i) = \text{Enc}(\prod_i m_i; \Sigma_i r_i)$

2. $\text{Enc}(m; r)^s = \text{Enc}(m^s; sr)$ for public $s \in \mathbb{Z}_q$
SOME COROLLARIES

- Recall: \( \text{Enc} (m_1; r_1) \cdot \text{Enc} (m_2; r_2) = \text{Enc} (m_1 \cdot m_2; r_1 + r_2) \)
- We omit \textit{pk} (public key) for brevity in what follows

1. \( \prod_i \text{Enc} (m_i; r_i) = \text{Enc} (\prod_i m_i; \sum_i r_i) \)
2. \( \text{Enc} (m; r)^s = \text{Enc} (m^s; sr) \) for public \( s \in \mathbb{Z}_q \)
3. \( \prod_i \text{Enc} (x^i; r_i)^{f_i} = \prod_i \text{Enc} ((x^i)^{f_i}; f_ir_i) = \text{Enc} \left( \prod_i (x^i)^{f_i}; \sum_i f_ir_i \right) \)
SOME COROLLARIES

- Recall: $\text{Enc} (m_1; r_1) \cdot \text{Enc} (m_2; r_2) = \text{Enc} (m_1 \cdot m_2; r_1 + r_2)$
- We omit $pk$ (public key) for brevity in what follows

1. $\prod_i \text{Enc} (m_i; r_i) = \text{Enc} (\prod_i m_i; \sum_i r_i)$
2. $\text{Enc} (m; r)^s = \text{Enc} (m^s; sr)$ for public $s \in \mathbb{Z}_q$
3. $\prod_i \text{Enc} (x^i; r_i)^{f_i} = \prod_i \text{Enc} ((x^i)^{f_i}; f_ir_i) = \text{Enc} \left( \prod_i (x^i)^{f_i}; \sum_i f_ir_i \right)$

- evaluating public "polynomial" at secret point
SOME COROLLARIES

- Recall: $\text{Enc}(m_1; r_1) \cdot \text{Enc}(m_2; r_2) = \text{Enc}(m_1 \cdot m_2; r_1 + r_2)$

- We omit $\text{pk}$ (public key) for brevity in what follows

1. $\prod_i \text{Enc}(m_i; r_i) = \text{Enc}(\prod_i m_i; \sum_i r_i)$

2. $\text{Enc}(m; r)^s = \text{Enc}(m^s; sr)$ for public $s \in \mathbb{Z}_q$

3. $\prod_i \text{Enc}(x^i; r_i)^{f_i} = \prod_i \text{Enc}((x^i)^{f_i}; f_ir_i) = \text{Enc}\left(\prod_i (x^i)^{f_i}; \sum_i f_ir_i\right)$

- Polynomials have a sum instead of product and product instead of exponentiation

- evaluating public "polynomial" at secret point
QUIZ: ENCRYPTING INTEGERS
QUIZ: Encrypting Integers

- Elgamal makes it possible to encrypt group elements
Elgamal makes it possible to encrypt group elements

**Question**: how to encrypt (short) integers?
QUIZ: ENCRYPTING INTEGERS

- Elgamal makes it possible to encrypt group elements
- **Question:** how to encrypt (short) integers?
- **Answer:**
Elgamal makes it possible to encrypt group elements

**Question:** how to encrypt (short) integers?

**Answer:**

- **Lifted Elgamal:** use exponentiation (isomorphism)
Elgamal makes it possible to encrypt group elements

**Question:** how to encrypt (short) integers?

**Answer:**

- **Lifted Elgamal:** use exponentiation (isomorphism)

\[
\text{Enc}^*(m \in \mathbb{Z}_q; r) := \text{Enc}(g^m; r)
\]
Elgamal makes it possible to encrypt group elements

**Question:** how to encrypt (short) integers?

**Answer:**

- **Lifted Elgamal:** use exponentiation (isomorphism)

- $\text{Enc}^*(m \in \mathbb{Z}_q; r) := \text{Enc}(g^m; r)$
- $\text{Dec}^*(c_1, c_2) = \text{dlog}_g(\text{Dec}(c_1, c_2))$
SOME COROLLARIES
SOME COROLLARIES

1. $\Pi_i \text{Enc}^* (m_i; r_i) = \text{Enc}^* (\Sigma_i m_i; \Sigma_i r_i)$
SOME COROLLARIES

1. $\Pi_i \text{Enc}^* (m_i; r_i) = \text{Enc}^* (\Sigma_i m_i; \Sigma_i r_i)$

2. $\text{Enc}^* (m; r)^s = \text{Enc}^* (sm; sr)$ for public $s \in \mathbb{Z}_q$
SOME COROLLARIES

1. \( \prod_i \text{Enc}^* (m_i; r_i) = \text{Enc}^* (\sum_i m_i; \sum_i r_i) \)

2. \( \text{Enc}^* (m; r)^s = \text{Enc}^* (sm; sr) \) for public \( s \in \mathbb{Z}_q \)

3. \( \prod_i \text{Enc}^* (x^i; r_i)^{f_i} = \prod_i \text{Enc}^* (f_ix^i; f_ir_i) = \text{Enc}^* \left( \sum_i f_ix^i; \sum_i f_ir_i \right) \)
1. \( \prod_i \text{Enc}^* (m_i; r_i) = \text{Enc}^* (\sum_i m_i; \sum_i r_i) \)

2. \( \text{Enc}^* (m; r)^s = \text{Enc}^* (sm; sr) \) for public \( s \in \mathbb{Z}_q \)

3. \( \prod_i \text{Enc}^* (x^i; r_i)^{f_i} = \prod_i \text{Enc}^* (fx^i; fr_i) = \text{Enc}^* \left( \sum_i f_ix^i; \sum_i fr_i \right) \)
1. \( \Pi_i Enc^* (m_i; r_i) = Enc^* (\Sigma_i m_i; \Sigma_i r_i) \)

2. \( Enc^* (m; r)^s = Enc^* (sm; sr) \) for public \( s \in \mathbb{Z}_q \)

3. \( \Pi_i Enc^* (x^i; r_i)^{f_i} = \Pi_i Enc^* (f_i x^i; f_i r_i) = Enc^* \left( \Sigma_i f_i x^i; \Sigma_i f_i r_i \right) \)

\* evaluation of public polynomial at secret point
SOME COROLLARIES

1. $\Pi_i \text{Enc}^* (m_i; r_i) = \text{Enc}^* (\sum_i m_i; \sum_i r_i)$

2. $\text{Enc}^* (m; r)^s = \text{Enc}^* (sm; sr)$ for public $s \in \mathbb{Z}_q$

3. $\Pi_i \text{Enc}^* (x^i; r_i)^{f_i} = \Pi_i \text{Enc}^* (f_i x^i; f_i r_i) = \text{Enc}^* \left( \sum_i f_i x^i; \sum_i f_i r_i \right)$

   ◆ evaluation of public polynomial at secret point

4. $\Pi_i \text{Enc}^* (x^i; r_i)^{y_i} \cdot \text{Enc}^* (z; t) = \text{Enc}^* \left( \sum_i x_i y_i + z; \sum_i y_i r_i + t \right)$
SOME COROLLARIES

1. $\Pi_i \text{Enc}^* (m_i; r_i) = \text{Enc}^* (\sum_i m_i; \sum_i r_i)$

2. $\text{Enc}^* (m; r)^s = \text{Enc}^* (sm; sr)$ for public $s \in \mathbb{Z}_q$

3. $\Pi_i \text{Enc}^* (x_i; r_i)^{f_i} = \Pi_i \text{Enc}^* (f_i x_i; f_i r_i) = \text{Enc}^* \left( \sum_i f_i x_i; \sum_i f_i r_i \right)$

   ◊ evaluation of public polynomial at secret point

4. $\Pi_i \text{Enc}^* (x_i; r_i)^{y_i} \cdot \text{Enc}^* (z; t) = \text{Enc}^* \left( \sum_i x_i y_i + z; \sum_i y_i r_i + t \right)$

   ◊ arbitrary affine functions with public $y_i, z$
LIFTED ELGAMAL: MORE

\[\mathbb{Z}_q \times G \]
LIFTED ELGAMAL: MORE

- Enc, Dec --- efficient
**LIFTED ELGAMAL: MORE**

- **Enc, Dec --- efficient**
- **Enc* efficient**

![Diagram](image-url)
LIFTED ELGAMAL: MORE

- Enc, Dec --- efficient
- Enc* efficient
- Dec* inefficient
LIFTED ELGAMAL: MORE

- Enc, Dec --- efficient
- Enc* efficient
- Dec* inefficient
- Dec* can be computed in time $\Theta(\sqrt{L})$ when input is from $\{0, ..., L - 1\}$
LIFTED ELGAMAL: MORE

- Enc, Dec --- efficient
- Enc* efficient
- Dec* inefficient
- Dec* can be computed in time $\Theta(\sqrt{L})$ when input is from $\{0, ..., L - 1\}$
- Recall: BSGS, PH algorithms
LIFTED ELGAMAL: MORE

- Enc, Dec --- efficient
- Enc* efficient
- Dec* inefficient
- Dec* can be computed in time $\Theta (\sqrt{L})$ when input is from $\{0, \ldots, L - 1\}$
- Recall: BSGS, PH algorithms
- Lifted Elgamal useful when $L$ is small: $L \leq 2^{40}$?
2-MESSAGE S.C. IN A NUTSHELL

1. Encrypt \( a \) by using \( pk \)

2. For polynomial \( f \), compute \( f(a, b) \) on encrypted inputs

3. Decrypt by using \( sk \), obtain \( g(f(a, b)) \). Compute DL

\[ c = \text{Enc}_{pk}(f(a)) \]

\[ \text{(pk, Enc}_{pk}(a)) \]

Million dollar question: how to do efficiently?
SUMMARY

- One can "compute" on Elgamal ciphertexts
SUMMARY

- One can "compute" on Elgamal ciphertexts
- Plaintexts are group elements
SUMMARY

✧ One can "compute" on Elgamal ciphertexts
  ✧ Plaintexts are group elements
✧ More useful: lifted Elgamal
SUMMARY

- One can "compute" on Elgamal ciphertexts
  - Plaintexts are group elements
- More useful: lifted Elgamal
  - However, only efficient for small plaintexts
SUMMARY

- One can "compute" on Elgamal ciphertexts
  - Plaintexts are group elements
- More useful: lifted Elgamal
  - However, only efficient for small plaintexts
- Secret computation of arbitrary affine functions in $\mathbb{Z}_q$ that have small outputs
One can "compute" on Elgamal ciphertexts
- Plaintexts are group elements
More useful: lifted Elgamal
- However, only efficient for small plaintexts
- Secret computation of arbitrary affine functions in $\mathbb{Z}_q$ that have small outputs
Can do polynomials but then need to separately encrypt all monomials - inefficient when degree increases
SUMMARY

- One can "compute" on Elgamal ciphertexts
  - Plaintexts are group elements
- More useful: lifted Elgamal
  - However, only efficient for small plaintexts
- Secret computation of arbitrary affine functions in $\mathbb{Z}_q$ that have small outputs
- Can do polynomials but then need to separately encrypt all monomials - inefficient when degree increases

limited malleability
STUDY OUTCOMES
STUDY OUTCOMES

- Secure computation: idea
STUDY OUTCOMES

- Secure computation: idea
- Modularization
STUDY OUTCOMES

- Secure computation: idea
- Modularization
- Elgamal
STUDY OUTCOMES

- Secure computation: idea
- Modularization
- Elgamal
- IND-CPA security
STUDY OUTCOMES

- Secure computation: idea
- Modularization
- Elgamal
- IND-CPA security
- Malleability, homomorphic encryption
STUDY OUTCOMES

- Secure computation: idea
- Modularization
- Elgamal
- IND-CPA security
- Malleability, homomorphic encryption
- Lifted Elgamal, its limitations
WHAT NEXT?
WHAT NEXT?

- Examples of concrete protocols
WHAT NEXT?

- Examples of concrete protocols
  - quite cool non-trivial stuff can be done even under severe limitations (groups)
WHAT NEXT?

- Examples of concrete protocols
  - quite cool non-trivial stuff can be done even under severe limitations (groups)
- Formalization and proof of their security