
# Cryptography
# ECE5632 - Spring 2025

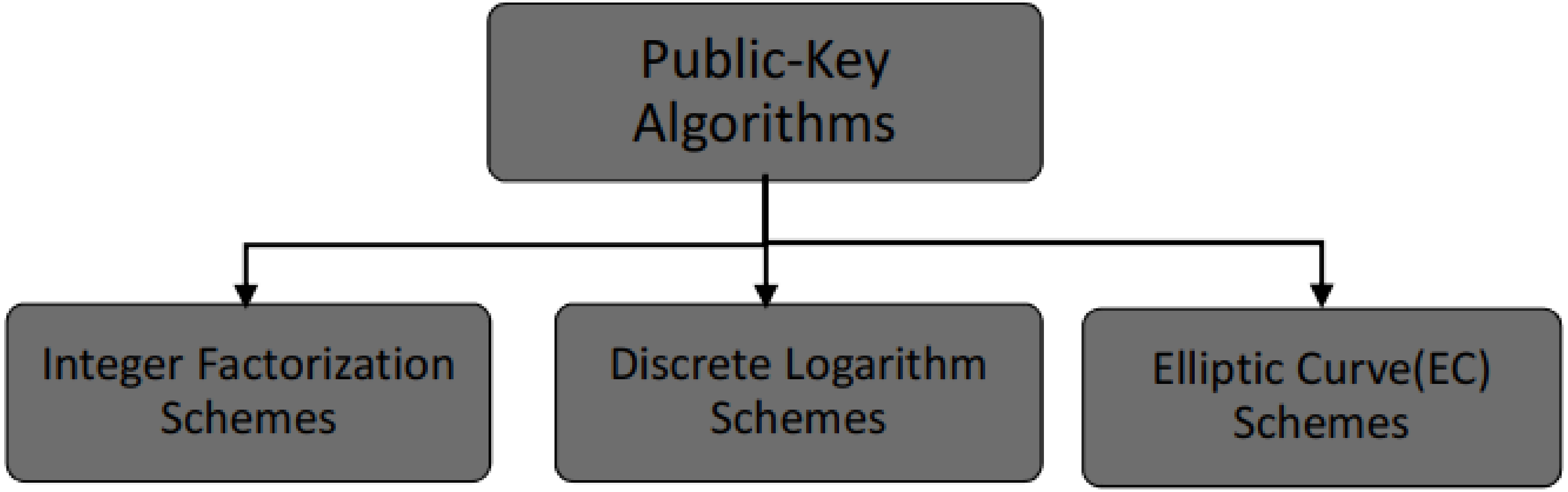
Lecture 8A

**Dr. Farah Raad**

# Lecture Topic

# ElGamal Encryption

# PKC Algorithms: Three Families




e.g.

- RSA algorithm.

e.g.:

- Diffie–Hellman key exchange.
- ElGamal encryption.
- Digital Signature Algorithm (DSA).

e.g.:

- EC Diffie–Hellman key exchange (ECDH).
- EC Digital Signature Algorithm (ECDSA).

# ElGamal Encryption

- Invented in 1985 by Taher ElGamal.
- Can be viewed as an extension of the DHKE protocol
- Based on the intractability of the discrete logarithm problem and the Diffie–Hellman problem

# The Elgamal Encryption Protocol

**Alice** | **Bob**

**Bob:**

choose large prime $p$

choose primitive element $\alpha \in Z_p^*$ or in a subgroup of $Z_p^*$

choose $d = k_{prB} \in \{2,\ldots,p-2\}$

compute $\beta = k_{pubB} = \alpha^d \bmod p$

Bob → Alice: $k_{pubB} = (p, \alpha, \beta)$

**Alice:**

choose i = $k_{prA} \in \{2,\ldots,p-2\}$

compute $k_E = k_{pubA} = \alpha^i \bmod p$

compute masking key $k_M = \beta^i \bmod p$

encrypt message $x \in Z_p^*$:
$y = x \cdot k_M \bmod p$

Alice → Bob: $(k_E, y)$

**Bob:**

compute masking key $k_M = k_E^{\,d} \bmod p$

decrypt $x = y \cdot k_M^{-1} \bmod p$

✓ This looks very similar to the DHKE! The actual Elgamal protocol re-orders the computations which helps to save one communication

# ElGamal Encryption

***Example:*** **In this example, Bob generates the Elgamal keys and Alice encrypts the message *x* = 26.**

| **Alice** | | **Bob** |
|---|---|---|
| message $x = 26$ | | generate $p = 29$ and $\alpha = 2$ |
| | | choose $k_{pr,B} = d = 12$ |
| | | compute $\beta = \alpha^d \equiv 7 \bmod 29$ |
| | $\xleftarrow{k_{pub,B}=(p,\alpha,\beta)}$ | |
| choose $i = 5$ | | |
| compute $k_E = \alpha^i \equiv 3 \bmod 29$ | | |
| compute $k_M = \beta^i \equiv 16 \bmod 29$ | | |
| encrypt $y = x \cdot k_M \equiv 10 \bmod 29$ | | |
| | $\xrightarrow{y,k_E}$ | |
| | | compute $k_M = k_E^d \equiv 16 \bmod 29$ |
| | | decrypt |
| | | $x = y \cdot k_M^{-1} \equiv 10 \cdot 20 \equiv 26 \bmod 29$ |

# ElGamal Encryption

- Proof of Correctness:

Show that y· $K_M^{-1}$ mod p ≡ x mod p

$$
\begin{aligned}
y\cdot K_M^{-1} \bmod p &\equiv y\cdot (K_E^{d})^{-1} \bmod p \\
&\equiv x\cdot K_M\cdot K_E^{-d} \bmod p \\
&\equiv x\cdot \beta^{i}\cdot(\alpha^{i})^{-d} \bmod p \\
&\equiv x\cdot (\alpha^{d})^{i}\cdot(\alpha^{i})^{-d} \bmod p \\
&\equiv x\cdot \alpha^{0} \bmod p \\
&\equiv x \bmod p
\end{aligned}
$$

# ElGamal Encryption

- $K_E$ must be different for every x.
- ElGamal is a probabilistic encryption scheme, unlike schoolbook RSA.
- Since it depends on DLP, p should be at least 1024 bits long.
- i, $K_{pr}$ should result from a TRNG.

# Computational Aspects

- Key Generation
  - Generation of prime $p$
  - $p$ has to be at least 1024 bits long

- Encryption
  - Requires two modular exponentiations and a modular multiplication
  - All operands have a bit length of $\boldsymbol{log_2 P}$
  - Efficient execution requires methods such as the square-and-multiply algorithm

- Decryption
  - Requires one modular exponentiation and one modular inversion
  - As shown *in Understanding Cryptography*, the inversion can be computed from the ephemeral key

# ElGamal Encryption: Attack


**Bob:**

Choose p, $\alpha$

$k_{pr} = d \in \{2, \ldots, p-2\}$

$k_{pub} = \beta \equiv \alpha^d \bmod p$

$\xleftarrow{(\beta, p, \alpha)}$

**Alice:**

Choose $i \in \{2, \ldots, p-2\}$

$k_E \equiv \alpha^i \bmod p$

$k_M \equiv \beta^i \bmod p$

$y \equiv x \cdot K_M \bmod p$

$\xrightarrow{(k_E, y)}$

**Bob:**

$k_M \equiv K_E^d \bmod p$

Decrypt: $x \equiv y \cdot K_M^{-1} \bmod p$

A passive attacker's goal is to observe the channel and compute x.

He doesn't have i or d.

# ElGamal Encryption: Attack

Passive attack option 1:
Compute $d = \log_\alpha \beta$
or, compute $i = \log_\alpha K_E$

→ So, we must make sure that the DLP is very hard.
i.e., length of p is ≥ 1024 bits.

Passive attack option 2:
The attacker knows that i is being reused.

→ Make sure there's a fresh TRN i used every time.

# ElGamal Encryption: Attack

- Passive attacks
  - Attacker eavesdrops $p$, $\alpha$, $\beta = \alpha^d$ , $k_E = \alpha^i$, $y = x \cdot \beta^i$ and wants to recover $x$
  - Problem relies on the DLP
  - Key must be at leasst **1024 bits** long
- Active attacks
  - MITM attack defeats it
  - An attack is also possible if the secret exponent $i$ is being used more than once
  - If attacker can guess the plaintext of one message, it can be used to decrypt another message using the same key

# Example

- Find $3^{100000}$ mod 53
- Use Fermat's theorem to find a number x between 0 and 36 with $X^{145}$ equivalent to 7 modulo 37.

# Example :1

- Find $3^{100000}$ mod 53

Using Fermat Little Theorem:

$3^{53-1}=1 \bmod 53$

100000/52 : q=1923 , r=4

$3^{99996} = (3^{52})^{1923}=1 \bmod 53$

$3^{100000} = 3^4(3^{99996}) = 3^4 \bmod 53$

$3^{100000} = 81 \bmod 53 = 28 \bmod 53$

$$\boxed{a^p \equiv a \pmod{p}}$$

$$a^{p-1} \equiv 1 \pmod{p}$$

# Example :2

- Use Fermat's theorem to find a number x between 0 and 36 with $X^{145}$ equivalent to 7 modulo 37.

Using Fermat Little Theorem:

Since 37 is prime and with x not divisible by 37,

Then $x^{37-1} = x^{36} = 1 \bmod 37$.

$x^{145} = (x^{36})^4 . x = 7 \bmod 37$

Then $x = 7 \bmod 37$

# Thank You!

**See You next Lectures!!**

**Any Question?**

THE FIRST BRITISH HIGHER EDUCATION IN EGYPT

26th July Mehwar Road Intersection with Wahat Road, 6th of October City, Egypt

**Tel:** +202383711146 **Fax:** +20238371543 **Postal code:** 12451

**Email:** info@msa.eun.eg **Hotline:** 16672 **Website:** www.msa.edu.eg