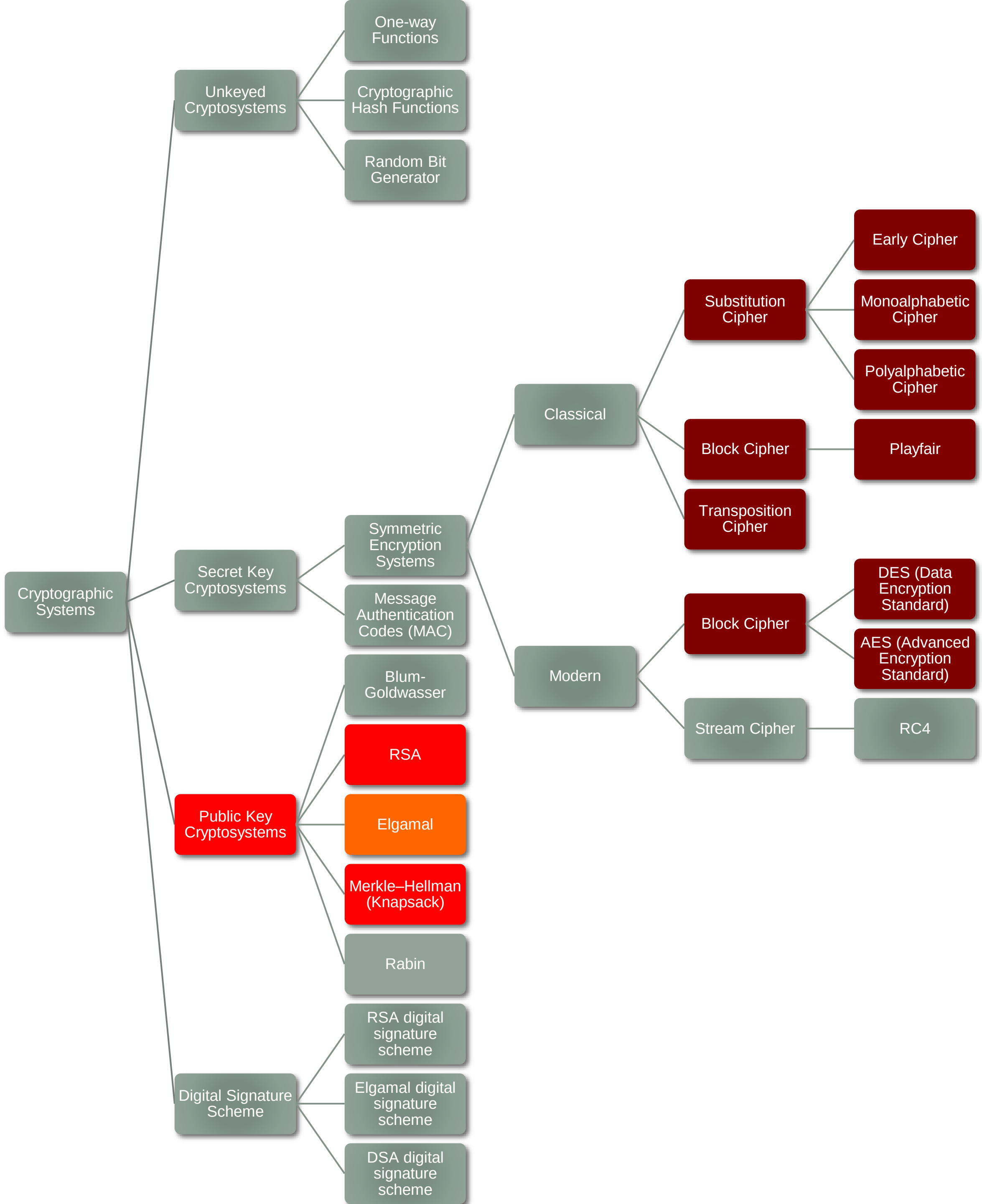
TUTORIAL

CSCI361 – Computer Security

Sionggo Japit

sjapit@uow.edu.au

12 February 2024


Cryptographic Systems
Unkeyed Cryptosystems
One-way Functions
Cryptographic Hash Functions
Random Bit Generator
Secret Key Cryptosystems
Symmetric Encryption Systems
Message Authentication Codes (MAC)
Classical
Substitution Cipher
Early Cipher
Monoalphabetic Cipher
Polyalphabetic Cipher
Block Cipher
Playfair
Transposition Cipher
Modern
Block Cipher
DES (Data Encryption Standard)
AES (Advanced Encryption Standard)
Stream Cipher
RC4
Public Key Cryptosystems
Blum-Goldwasser
RSA
Elgamal
Merkle–Hellman (Knapsack)
Rabin
Digital Signature Scheme
RSA digital signature scheme
Elgamal digital signature scheme
DSA digital signature scheme

GENERATOR IN Z_p^*

Generator in z_p^*

- A **generator** is an element of Z_p^* whose first *p-1* powers generate all the nonzero elements of the field F. It is also known as **primitive element** of a finite field *F* containing *p* number of elements.

For example, in a finite field GF(11), all the nonzero elements are 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. If $a^1 \bmod 11$, $a^2 \bmod 11$, $a^3 \bmod 11, \ldots, a^{10} \bmod 11$ produce all the nonzero elements of F, then ***a*** is said to be a generator in GF(11).

Generator in z_p^*

For example:

$2^1 \bmod 11 = 2$
$2^2 \bmod 11 = 4$
$2^3 \bmod 11 = 8$
$2^4 \bmod 11 = 5$
$2^5 \bmod 11 = 10$
$2^6 \bmod 11 = 9$
$2^7 \bmod 11 = 7$
$2^8 \bmod 11 = 3$
$2^9 \bmod 11 = 6$
$2^{10} \bmod 11 = 1$

The values *2, 4, 8, 5, 10, 9, 7, 3, 6*, and *1* are nonzero elements of F (GF(11)).

Hence ***2*** is said to be a generator or primitive element.

Generator in z_p^*

Generator in Z^*_{11}:

	1	2	3	4	5	6	7	8	9	10
1	1	1								
2	**2**	**4**	**8**	**5**	**10**	**9**	**7**	**3**	**6**	**1**
3	3	9	5	4	1	3				
4	4	5	9	3	1	4				
5	5	3	4	9	1	5				
6	**6**	**3**	**7**	**9**	**10**	**5**	**8**	**4**	**2**	**1**
7	**7**	**5**	**2**	**3**	**10**	**4**	**6**	**9**	**8**	**1**
8	**8**	**9**	**6**	**4**	**10**	**3**	**2**	**5**	**7**	**1**
9	9	4	3	5	1	9				
10	10	1	10							

Generator in z_p^*

- Unfortunately there is **NO** simple general formula to compute generator is known.
- However, there are methods to determine a generator that are faster than simply trying out all candidates.

Generator in z_p^*

Finding generator using $g^{\frac{p-1}{2}} \bmod p$ test.

- **To use this test, *p* must be a safe prime.**
- A prime number p is a safe prime if $p = 2q + 1$, and q is a prime.

> For a safe prime p, test if $g^{\frac{p-1}{2}} \bmod p \neq 1$.
> If yes, then g is a generator,
> otherwise $(-g) \bmod p$ is a generator.

Generator in z_p^*

For example, a prime number 11 is a safe prime

because $11 = (2 \times 5) + 1$, and 5 is a prime number.

Hence in Z_{11}^*,

$2^{\frac{11-1}{2}} \bmod 11 = 2^5 \bmod 11 = 10$, is not equal to 1. Thus **2** is a generator in Z_{11}^*.

Another example using Z_{11}^* :

$3^{\frac{11-1}{2}} \bmod 11 = 3^5 \bmod 11 = 1$, is equal to 1. Thus **3** is **NOT** a generator in Z_{11}^*,

but (-3) mod 11 = **8** is a generator in Z_{11}^*

Generator in z_p^*

- If prime *p* is **NOT** a safe prime; i.e., $p = 2q + 1$, and *q* is NOT a prime,
  - Randomly choose a number g such that $g \in z_p^*$. Exclude 1 and $p-1$.
  - Test if $g^{\frac{p-1}{p_j}} \neq 1 \bmod p$ for all prime numbers p_j where $1 \le j \le k$. (k = number of different prime number between 1 and $p-1$)

Note: use this test if *p* is not a safe prime.

Generator in z_p^*

For example, find a generator of Z_{13}^*.

Generator in z_p^*

For example, to test if 2 is a generator of Z_{13}^*,

we test if $2^{\frac{13-1}{2}} \bmod 13$, $2^{\frac{13-1}{3}} \bmod 13$, $2^{\frac{13-1}{5}} \bmod 13$,

$2^{\frac{13-1}{7}} \bmod 13$, and $2^{\frac{13-1}{11}} \bmod 13$ are not equal 1.

Taking $2^{\frac{13-1}{2}} \bmod 13$ as an example, it is computed as follow :

1. Compute $2^{-1} \bmod 13 = -6 \bmod 13 = 7$
2. Compute $(13-1) \times 7 \bmod 13 = 84 \bmod 13 = 6$
3. $2^6 \bmod 13 = 12$

Do the same calculation to test the rest.

Generator in z_p^*

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	3	6	12	11	9	5	10	7	1
3	3	9	1	3	9	1	3	9	1	3	9	1
4	4	3	12	9	10	1	4	3	12	9	10	1
5	5	12	8	1	5	12	8	1	5	12	8	1
6	6	10	8	9	2	12	7	3	5	4	11	1
7	7	10	5	9	11	12	6	3	8	4	2	1
8	8	12	5	1	8	12	5	1	8	12	5	1
9	9	3	1	9	3	1	9	3	1	9	3	1
10	10	9	12	3	4	1	10	9	12	3	4	1
11	11	4	5	3	7	12	2	9	8	10	6	1
12	12	1	12	1	12	1	12	1	12	1	12	1

2	3	5	7	11
1	1	1	1	1
12	3	6	7	11
1	3	9	9	3
1	9	10	10	4
12	1	5	8	8
12	9	2	11	7
12	9	11	2	6
12	1	8	5	5
1	9	3	3	9
1	3	4	4	10
12	3	7	6	2
1	1	12	12	12

Test if $g^{\frac{p-1}{p_j}} \neq 1 \bmod p$ for $1 \le j \le k$. (k=number of different prime number between 1 and p-1)

ELGAMAL

Elgamal

- Public-key cryptosystem
- Based on discrete logarithm problem
    - Hard to find $k < p$, such that $y = g^k \bmod p$
    - g is a generator of Z_p^*
    - k is in Z_p^*
- Proposed by Tather Elgamal in 1984
- A variant of Diffie-Hellman providing a one-pass protocol with unilateral key authentication.

Elgamal

Key generation (participant A):

1. Choose a prime p and a generator $g \in Z_p^*$.
2. Select a random integer k_A, $1 < k_A < p - 1$, and
3. Compute public key (receiver)

$$y_A = g^{k_A} \bmod p$$

$$public\ key: (p, g, y_A)$$

$$private\ key: k_A$$

Elgamal

To Encrypt (participant B):

1. Obtain recipient's (A's) authentic public key $(p,\ g,\ y_A)$
2. Represent the message as integer m in the range $\{0,\ 1,\ \ldots,\ p-1\}$.
3. Pick a random k_B
4. Compute a shared key $K = (y_A)^{k_B} \bmod p$
5. Encrypt the message m as a pair of integers $(\mathrm{C}_1,\ \mathrm{C}_2)$ where

$$C_1 = g^{k_B} \bmod p$$

$$C_2 = m \times K \bmod p$$

$$\textit{for a message } 0 < m < p$$

Note:
k_B = B's private key

The encrypted text (or ciphertext) is $(C_1,\ C_2)$

Elgamal

To Decrypt (participant A):

Participant A receives the encrypted message $(C_1, \ C_2)$ from B.

To decrypt the encrypted message, Participant A does the following:

- recovers the shared key $K = (C_1)^{k_A} \bmod p,$ then
- computes $m = \dfrac{C_2}{K} \bmod p$

Note: Need to compute the above in 3 steps:

1. $K = (C_1)^{k_A} \bmod p$ (need to use fast exponentiation)
2. $K_i = K^{-1} \bmod p$ (need to use extended Euclidean)
3. $m = C_2 K_i \bmod p$ or $m = C_2 K^{-1} \bmod p$

Elgamal

Let's see how the encryption can be done:

Example: Encrypt and decrypt "JAPIT" using Elgamal.

The message to encrypt is **JAPIT**

- Convert to ASCII decimal:

J	A	P	I	T
74	65	80	73	84

Elgamal

$Key\ generation:$

1. Choose a prime p and a generator $g \in Z_p^*$

$p = 89$ Why 89? Is 89 good?

$g \in Z_p^*$ How?

Elgamal

Key generation:

1. Choose a prime p and a generator g ∈ Z_p^*

$$p = 83$$

$$g = 5. \quad Test\ if\ 5^{\frac{82}{2}}\ mod\ 83 \neq 1?$$

5 x 27 x 75 mod 83
= 10125 mod 83 = 82 != 1
=> 5 is a generator!

2^0=1 (1)	**5^1 mod 83**	**5 mod 83 = 5**
2^1=2 (1)	5^2 mod 83	5 x 5 mod 83 = 25
2^2=4 (0)	5^4 mod 83	25 x 25 mod 83 = 625 mod 83 = 44
2^3=8(0)	**5^8 mod 83**	**44 x 44 mod 83 = 27**
2^4=16 (1)	5^{16} mod 83	27 x 27 mod 85 = 729 mod 83 = 65
2^5=32 (1)	**5^{32} mod 83**	**65 x 65 mod 83 = 4225 mod 83 = 75**

Elgamal

- Choose a random number $1 < k_r < p - 1$ and generate y_r (a component of the public key.)

  I choose $k_r = 3$.
- With my choices of $p = 83$, $g = 5$, and $k_r = 3$, I compute:

$$y_r = g^{k_r} \; mod \; p$$
$$y_r = 5^3 \; mod \; 83$$
$$y_r = 125 \; mod \; 83 = 42$$
$$\therefore \; The \; public \; key \; is \; (g, y_r, p) = (5, 42, 83), and$$
$$the \; private \; key \; is \; (k_r) = 3.$$

End of key generation!

Elgamal

Key generation:

- Encrypt letter J with $k_s = 5$.
- First compute the shared key $K = (y_r)^{k_s} \mod p$

$$K = 42^5 \mod 83$$

$$K = 13$$

Note: The public key (g, y, p) of the recipient is (5, 42, 83)..

How to calculate 42^5 mod 83?

Need to compute using fast exponentiation.

Elgamal

- I will use fast exponentiation!

2^0=1 (1)	**42^1 mod 83**	**42 mod 83 = 42**
2^1=2 (0)	42^2 mod 83	42 x 42 mod 83 = 1764 mod 83 = 21
2^2=4 (1)	**42^4 mod 83**	**21 x 21 mod 83 = 441 mod 83 =26**

$$\therefore 42^5 \bmod 83 = 42 \times 26 \bmod 83$$

$$= 1092 \bmod 83 = 13.$$

Elgamal

- Next calculate C_1 and C_2 pair as follow :

$$C_1 = g^{k_s} \mod p$$

$$= 5^5 \mod 83$$

$$= 54$$

$$C_2 = Km \mod p \quad where\ m = 74$$

$$= 13 \times 74 \mod 83$$

$$= 962 \mod 83 = 49$$

Hence the encrypted message $(C_1, C_2) = (54, 49)$

Elgamal

- The sender needs to choose a key $1 < k_s < p - 1$.
- NOTE: To introduce, confusion effect, the key for each letter of the message must be different!
- I choose my second $k_s = 11$ to encrypt the letter A.

Elgamal

- Encrypt letter A with $k_s = 11$.
- First compute the shared key $K = (y_r)^{k_s} \mod p$.

$$K = 42^{11} \mod 83$$

2^0=1 (1)	**42^1 mod 83**	**42 mod 83 = 42**
2^1=2 (1)	**42^2 mod 83**	**42 x 42 mod 83 = 1764 mod 83 = 21**
2^2=4 (0)	42^4 mod 83	21 x 21 mod 83 = 441 mod 83 = 26
23=8(1)	**42^8 mod 83**	**26 x 26 mod 83 = 676 mod 83 = 12**

$$\therefore 42^{11} \mod 83 = 42 \times 21 \times 12 \mod 83$$

$$K = 10584 \mod 83$$

$$K = 43$$

Elgamal

- Next, we calculate C_1 and C_2 pair as follow :

$$
\begin{aligned}
C_1 &= g^{k_s} \mod p \\
&= 5^{11} \mod 83 \\
&= 48828125 \mod 83 = 55
\end{aligned}
$$

$$
\begin{aligned}
C_2 &= Km \mod p \quad \textit{where } m = 65 \\
&= 43 \times 65 \mod 83 \\
&= 2795 \mod 83 = 56
\end{aligned}
$$

Hence the encrypted message $(C_1, C_2) = (55, 56)$

Elgamal

- Encrypt letter P with $k_s = 13$
- First compute the shared key $K = (y_r)^{k_s} \mod p$

$K = 42^{13} \mod 83$

$2^0=1$ (1)	**42^1 mod 83**	**42 mod 83 = 42**
$2^1=2$ (0)	42^2 mod 83	42 x 42 mod 83 = 1764 mod 83 = 21
$2^2=4$ (1)	**42^4 mod 83**	**21 x 21 mod 83 = 441 mod 83 = 26**
$2^3=8$(1)	**42^8 mod 83**	**26 x 26 mod 83 = 676 mod 83 = 12**

$$\therefore 42^{13} \mod 83 = 42 \times 26 \times 12 \mod 83$$

$$K = 13104 \mod 83$$

$$K = 73$$

Elgamal

- Next, we calculate C_1 and C_2 pair as follow :

$$
\begin{aligned}
C_1 &= g^{k_s} \mod p \\
&= 5^{13} \mod 83 \\
&= 1220703125 \mod 83 \\
&= 47 \\
C_2 &= Km \mod p \qquad where\ m = 80 \\
&= 73 \times 80 \mod 83 \\
&= 5840 \mod 83 = 30
\end{aligned}
$$

Hence the encrypted message $(C_1, C_2) = (47, 30)$

Elgamal

- Encrypt letter I with $k_s = 7$.
- First compute the shared key $K = (y_r)^{k_s} \mod p$

$$K = 42^7 \mod 83$$

2^0=1 (1)	42^1 mod 83	42 mod 83 = 42
2^1=2 (1)	42^2 mod 83	42 x 42 mod 83 = 1764 mod 83 = 21
2^2=4 (1)	42^4 mod 83	21 x 21 mod 83 = 441 mod 83 = 26

$$\therefore 42^7 \mod 83 = 42 \times 21 \times 26 \mod 83$$

$$\mathrm{K} = 22932 \mod 83$$

$$\mathrm{K} = 24$$

Elgamal

- Next, we calculate C1 and C2 pair as follow :

$$C_1 = g^{k_s} \mod p$$

$$= 5^7 \mod 83$$

$$= 78125 \mod 83 = 22$$

$$C_2 = Km \mod p \qquad where\ m = 73$$

$$= 24 \times 73 \mod 83$$

$$= 1752 \mod 83 = 9$$

Hence the encrypted message $(C_1, C_2) = (22, 9)$

Elgamal

- Encrypt letter T with $k_s = 19$.
- First copute the shared key $K = (y_r)^{k_s} \mod p$.

$$K = 42^{19} \mod 83$$

$2^0=1$ (1)	42^1 mod 83	42 mod 83 = 42
$2^1=2$ (1)	42^2 mod 83	42 x 42 mod 83 = 1764 mod 83 = 21
$2^2=4$ (0)	42^4 mod 83	21 x 21 mod 83 = 441 mod 83 = 26
$2^3=8$ (0)	42^8 mod 83	26 x 26 mod 83 = 676 mod 83 = 12
$2^4=16$ (1)	42^{16} mod 83	12 x 12 mod 83 = 144 mod 83 = 61

$$\therefore \; 42^{19} \mod 83 = 42 \times 21 \times 61 \mod 83$$

$$K = 53802 \mod 83$$

$$K = 18$$

Elgamal

- Next, we calculate C1 and C2 as follow :

$$C_1 = g^{k_s} \mod p$$

$$= 5^{19} \mod 83$$

$$= 19073486328125 \mod 83 = 74$$

$$C_2 = Km \mod 83$$

$$= 18 \times 84 \mod 83 \quad where\ m = 84$$

$$= 1512 \mod 83 = 18$$

Hence the encrypted message $(C_1, C_2) = (74, 18)$

Elgamal

- The encrypted text:

J	(54, 49)
A	(55, 56)
P	(47, 30)
I	(22, 9)
T	(74, 18)

Elgamal

Decrypt:

Encrypted text: (54, 49)

To decrypt the ciphertext (54, 49), the recipient calculates the following:

1. Recover the shared key K:

$$K = C_1^{k_R} \bmod p$$

2. Compute inverse K; i.e., K^{-1} using Extended Euclidean Algorithm

3. Recover the message m:

$$\mathrm{m} = C_2 K^{-1} \bmod p$$

Elgamal

$$K = C_1^{k_r} \bmod p$$

$$= 54^3 \bmod 83 = 157464 \bmod 83 = 13$$

$$K_i = K^{-1} \bmod p$$

$$= 13^{-1} \bmod 83 \quad \text{(Using Extended Euclidean Algorithm)}$$

$$= 32$$

n1	n2	r	q	a1	b1	a2	b2
83	13	5	6	1	0	0	1
13	5	3	2	0	1	1	-6
5	3	2	1	1	-6	-2	13
3	2	1	1	-2	13	3	-19
2	1	0	2	3	-19	-5	32

Elgamal

$$m = C_2 K^{-1} \bmod p$$
$$= \; 49 \times 32 \bmod 83$$
$$= \; 1568 \bmod 83 \; = \; 74$$

- The recovered message is 74.
- Hence (54, 49) = **74 = J (in ASCII)** (decrypted)

Elgamal

Decrypt:

Encrypted text: (55, 56)

To decrypt ciphertext (55, 56), the recipient calculates the following:

1. Recover the shared key K:

$$K = C_1^{k_R} \bmod p$$

2. Compute inverse K; i.e., K^{-1} using Extended Euclidean Algorithm

3. Recover the message m:

$$\mathrm{m} = C_2 K^{-1} \bmod p$$

Elgamal

$$K = C_1^{k_r} \bmod p$$

$$= 55^3 \bmod 83 = 166375 \bmod 83 = 43$$

$$K_i = K^{-1} \bmod p$$

$$= 43^{-1} \bmod 83 \quad \text{(Using Extended Euclidean Algorithm)}$$

$$= -27 \bmod 83 = 56$$

n1	n2	r	q	a1	b1	a2	b2
83	43	40	1	1	0	0	1
43	40	3	1	0	1	1	-1
40	3	1	13	1	-1	-1	2
3	1	0	3	-1	2	14	-27

Elgamal

$$m = C_2 K^{-1} \bmod p$$

$$= 56 \times 56 \bmod 83$$

$$= 3136 \bmod 83 = 65$$

- The recovered message is 65.
- Hence (55, 56) = **65 = A (in ASCII)** (decrypted)

Elgamal

Decrypt:

Encrypted text: (47, 30)

To decrypt ciphertext (47, 30), the recipient calculates the following:

1. Recover the shared key K:

$$K = C_1^{k_R} \bmod p$$

2. Compute inverse K; i.e., K^{-1} using Extended Euclidean Algorithm

3. Recover the message m:

$$\mathrm{m} = C_2 K^{-1} \bmod p$$

Elgamal

$$K = C_1^{k_r} \bmod p$$

$$= 47^3 \bmod 83 = 103823 \bmod 83 = 73$$

$$K_i = K^{-1} \bmod p$$

$$= 73 \bmod 83 \quad \text{(Using Extended Euclidean Algorithm)}$$

$$= -25 \bmod 83 = 58$$

n1	n2	r	q	a1	b1	a2	b2
83	73	10	1	1	0	0	1
73	10	3	7	0	1	1	-1
10	3	1	3	1	-1	-7	8
3	1	0	3	-7	8	22	-25

Elgamal

$$m = C_2 K^{-1} \bmod p$$
$$= 30 \times 58 \bmod 83$$
$$= 1740 \bmod 83 = 80$$

- The recovered message is 80.
- Hence (47, 30) = **80 = P (in ASCII)** (decrypted)

Elgamal

Decrypt:

Encrypted text: (22, 9)

To decrypt ciphertext (22, 9), the recipient calculates the following:

1. Recover the shared key K:

$$K = C_1^{k_R} \bmod p$$

2. Compute inverse K; i.e., K^{-1} using Extended Euclidean Algorithm

3. Recover the message m:

$$\mathrm{m} = C_2 K^{-1} \bmod p$$

Elgamal

$$K = C_1^{k_r} \bmod p$$

$$= \ 22^3 \ \bmod \ 83 \ = \ 10648 \ \bmod \ 83 \ = \ 24$$

$$K_i = K^{-1} \ \bmod \ p$$

$$= \ 24^{-1} \ \bmod \ 83 \quad \text{(Using Extended Euclidean Algorithm)}$$

$$= \ -38 \ \bmod \ 83 \ = \ 45$$

n1	n2	r	q	a1	b1	a2	b2
83	24	11	3	1	0	0	1
24	11	2	2	0	1	1	-3
11	2	1	5	1	-3	-2	7
2	1	0	2	-2	7	11	-38

Elgamal

$$m = C_2 K^{-1} \bmod p$$
$$= 9 \times 45 \bmod 83$$
$$= 405 \bmod 83 = 73$$

- The recovered message is 73.
- Hence (22, 9) = **73 = I (in ASCII)** (decrypted)

Elgamal

Decrypt:

Encrypted text: (74, 18)

To decrypt ciphertext (74, 18), the recipient calculates the following:

1. Recover the shared key K:

$$K = C_1^{k_R} \bmod p$$

2. Compute inverse K; i.e., K^{-1} using Extended Euclidean Algorithm

3. Recover the message m:

$$m = C_2 K^{-1} \bmod p$$

Elgamal

$$K = C_1^{k_r} \bmod p$$

$$= \ 74^3 \ \bmod \ 83 \ = \ 405224 \ \bmod \ 83 = 18$$

$$K_i = K^{-1} \bmod p$$

$$= \ 18^{-1} \ \bmod \ 83 \quad \text{(Using Extended Euclidean Algorithm)}$$

$$= \ -23 \bmod 83 = 60$$

n1	n2	r	q	a1	b1	a2	b2
83	18	11	4	1	0	0	1
18	11	7	1	0	1	1	-4
11	7	4	1	1	-4	-1	5
7	4	3	1	-1	5	2	-9
4	3	1	1	2	-9	-3	14
3	1	0	3	-3	14	5	-23

Elgamal

$$
\begin{aligned}
m &= C_2 K^{-1} \bmod p \\
&= 18 \times 60 \bmod 83 \\
&= 1080 \bmod 83 = 1
\end{aligned}
$$

- The recovered message is 1.
- Hence (74, 18) = **84 = T (in ASCII)** (decrypted)

Elgamal

The decrypted texts are:

74 = J
65 = A
80 = P
73 = I
84 = T