MODULE 3

INFORMATION SECURITY [3 0 0 3]

ICT 3172:

Diffie hellman key exchange

10.4 ELGAMAL CRYPTOSYSTEM

Public-key cryptosystem.

Inventor, Taher ElGamal

Based on the discrete logarithm

10.4 ELGAMAL CRYPTOSYSTEM

If p is a very large prime, e_1 is a primitive root in the group G = <Zp*, × > and r is an integer, then $e_2 = e_1^r$ mod p is easy to compute using the fast exponential algorithm,

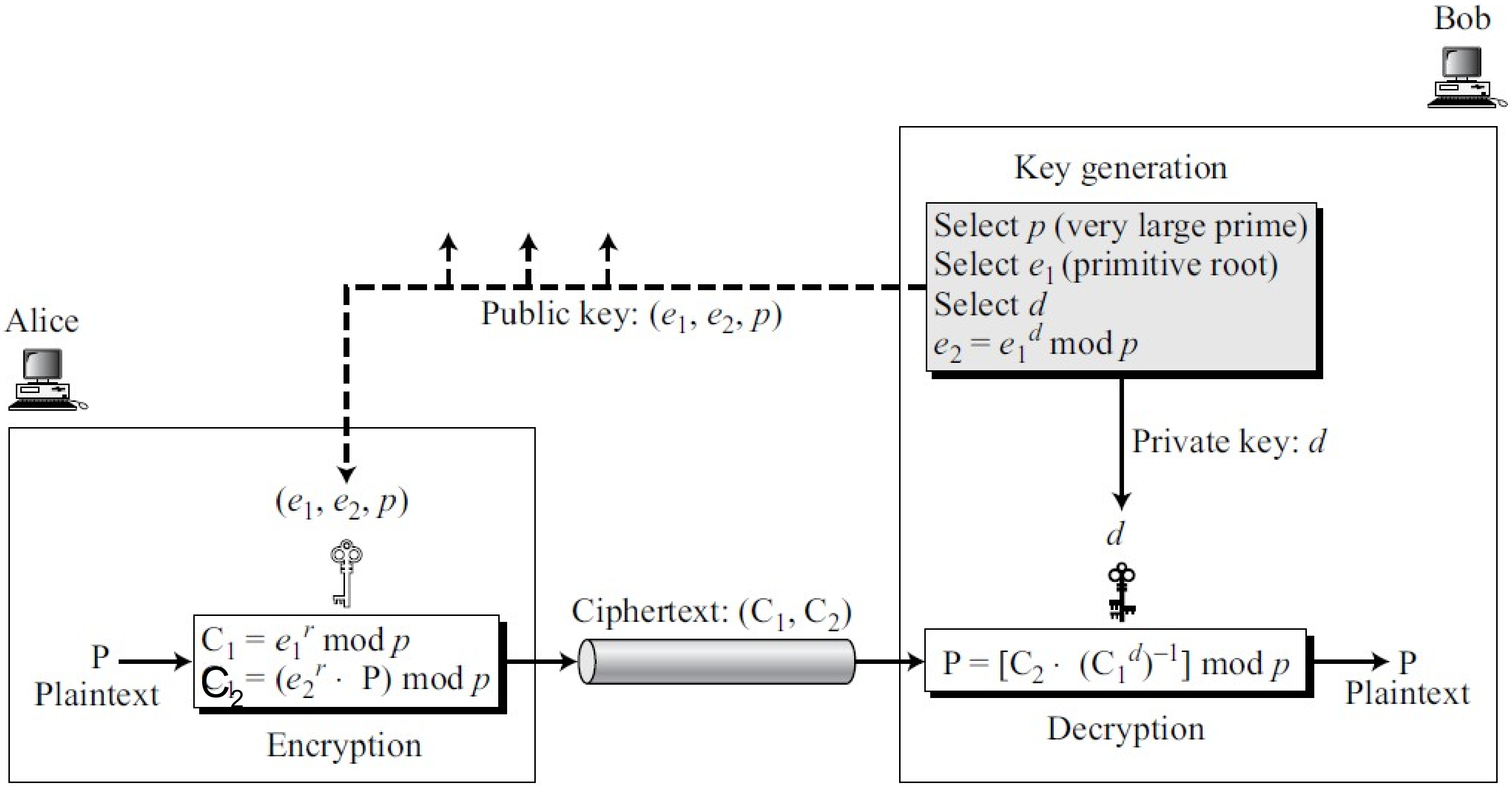
but

given e_2, e_1, and p, it is infeasible to calculate $r = \log_{e1} e_2$ mod p (discrete logarithm problem).

10.4 ELGAMAL CRYPTOSYSTEM

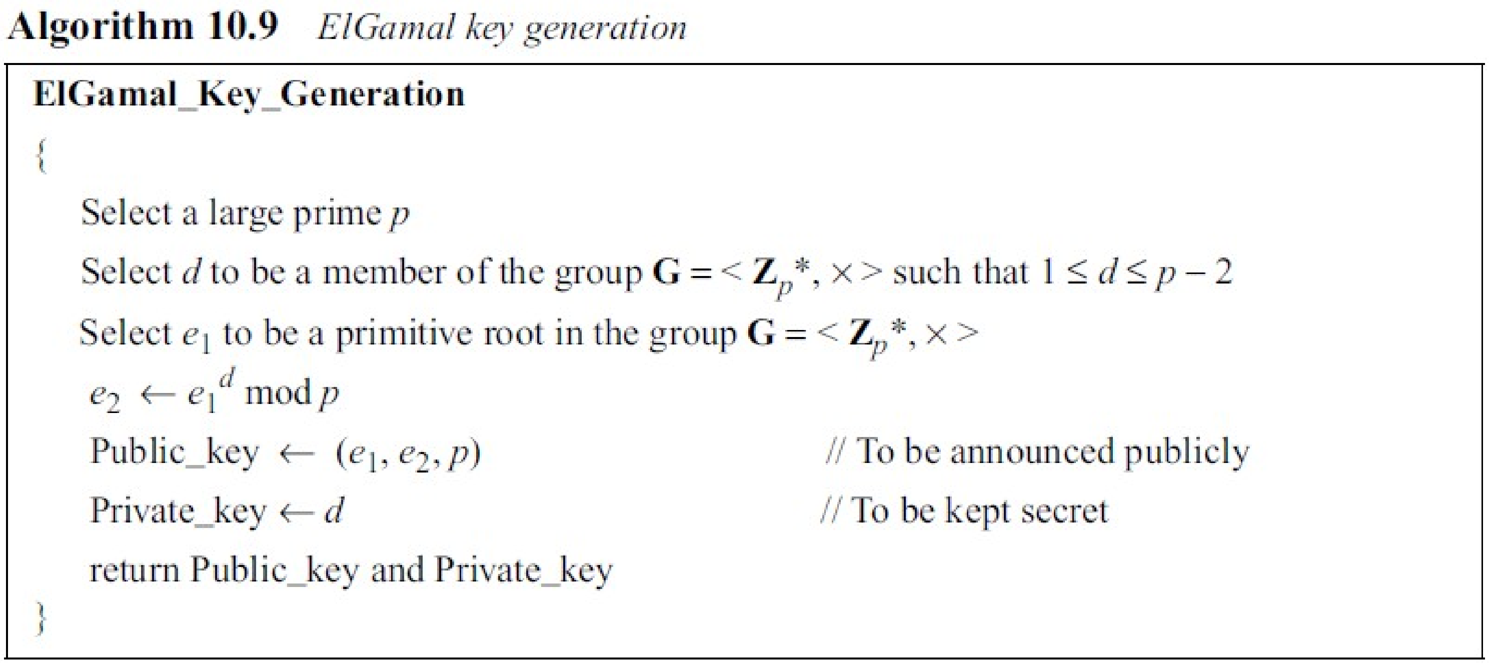
Figure shows key generation, encryption, and decryption in ElGamal

Figure 10.11 *Key generation, encryption, and decryption in ElGamal*



10.4 ELGAMAL CRYPTOSYSTEM

Key Generation algorithm to create public and private keys.

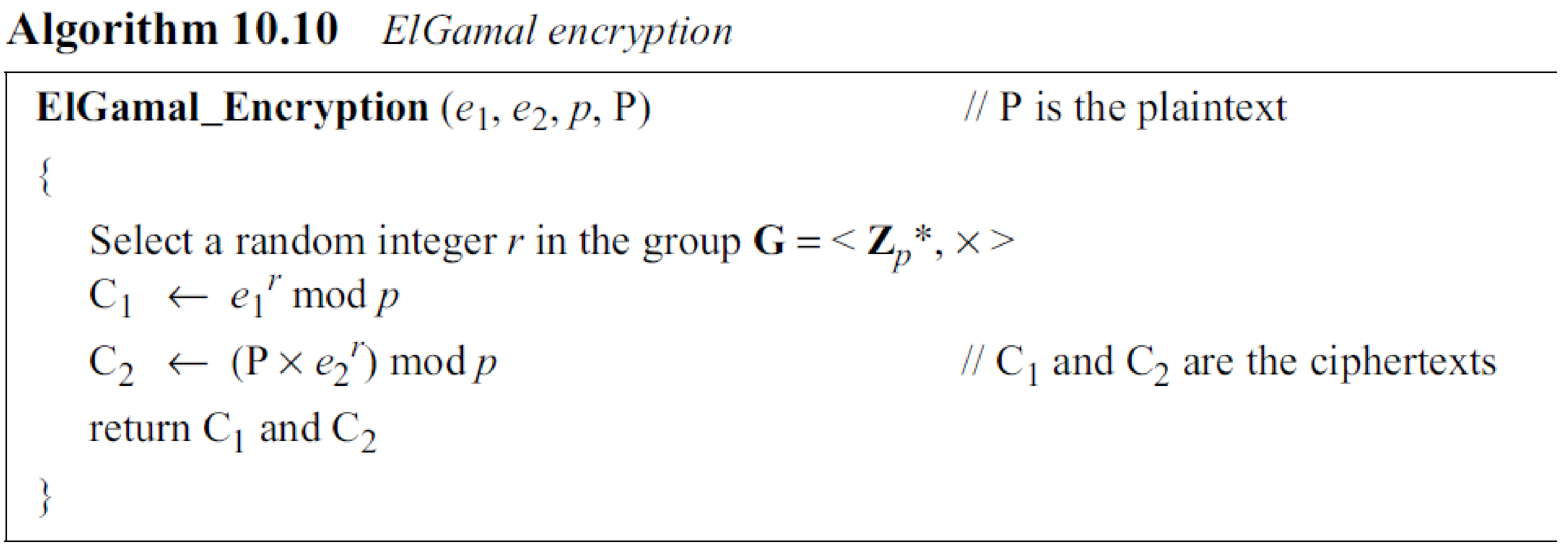


Algorithm 10.9 *ElGamal key generation*

```
ElGamal_Key_Generation
{
    Select a large prime $p$
    Select $d$ to be a member of the group $\mathbf{G} = <\mathbf{Z}_p{}^*, \times>$ such that $1 \le d \le p - 2$
    Select $e_1$ to be a primitive root in the group $\mathbf{G} = <\mathbf{Z}_p{}^*, \times>$
    $e_2 \leftarrow e_1{}^d \bmod p$
    Public_key $\leftarrow$ $(e_1, e_2, p)$                // To be announced publicly
    Private_key $\leftarrow d$                             // To be kept secret
    return Public_key and Private_key
}
```

10.4 ELGAMAL CRYPTOSYSTEM

Encryption

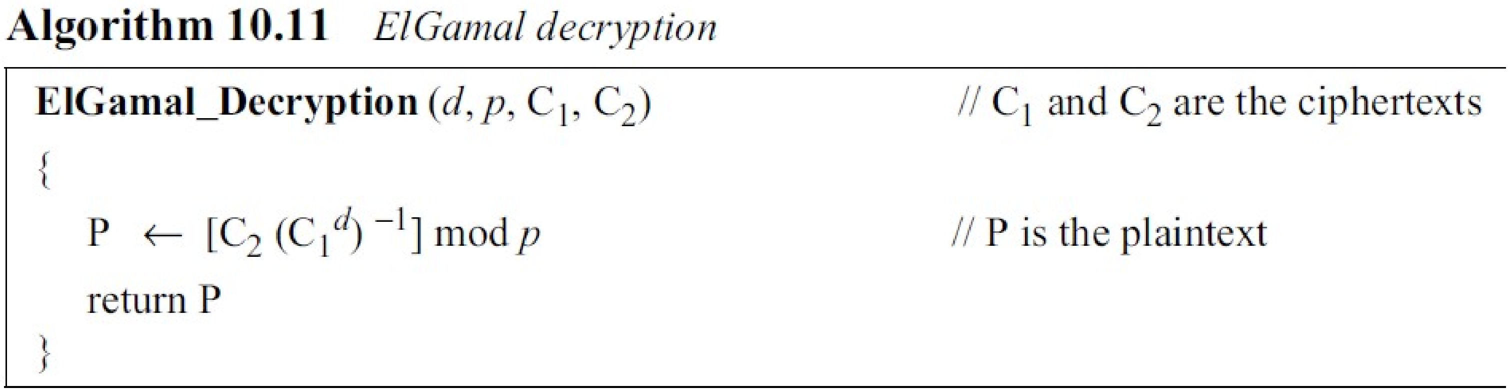


Algorithm 10.10 *ElGamal encryption*

```
ElGamal_Encryption ($e_1$, $e_2$, $p$, P)                    // P is the plaintext
{
    Select a random integer $r$ in the group $\mathbf{G} = < \mathbf{Z}_p{}^*, \times >$
    $C_1 \leftarrow e_1{}^r \bmod p$
    $C_2 \leftarrow (P \times e_2{}^r) \bmod p$                // $C_1$ and $C_2$ are the ciphertexts
    return $C_1$ and $C_2$
}
```

10.4 ELGAMAL CRYPTOSYSTEM

Decryption



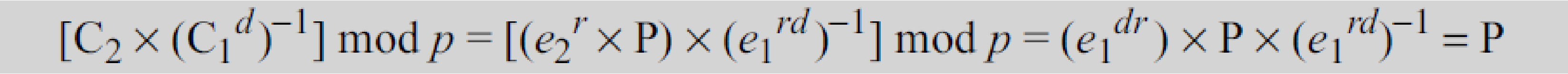
Algorithm 10.11 *ElGamal decryption*

```
ElGamal_Decryption ($d$, $p$, $C_1$, $C_2$)                    // $C_1$ and $C_2$ are the ciphertexts
{
    P ← $[C_2 (C_1^{d})^{-1}] \bmod p$                        // P is the plaintext
    return P
}
```

10.4 ELGAMAL CRYPTOSYSTEM

Proof

The ElGamal decryption expression C_2 × $(C_1^{d})^{-1}$ can be verified to be P through substitution:



$$[C_2 \times (C_1{}^{d})^{-1}] \bmod p = [(e_2{}^{r} \times P) \times (e_1{}^{rd})^{-1}] \bmod p = (e_1{}^{dr}) \times P \times (e_1{}^{rd})^{-1} = P$$