Information & System Security

Lecture 30

VIT-AP UNIVERSITY

Asymmetric
or
Public Key
Cryptography

10-3 RABIN CRYPTOSYSTEM

The Rabin cryptosystem can be thought of as an RSA cryptosystem in which the value of e and d are fixed.

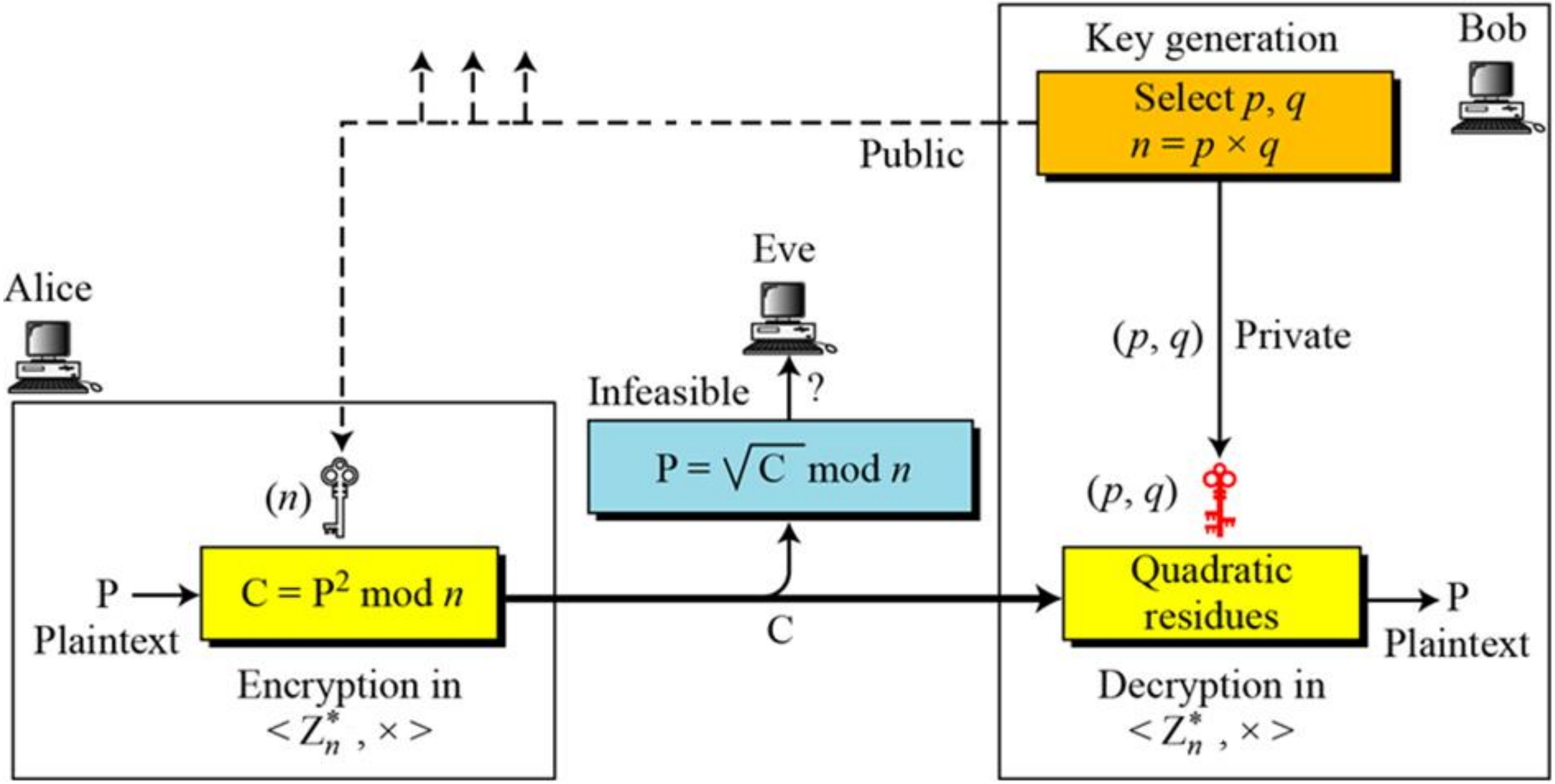
Encryption: $C \equiv P^2 \pmod{n}$

Decryption: $P \equiv C^{1/2} \pmod{n}$.

Topics discussed in this section:

10.3.1 Procedure

10-3 Continued



Rabin Cryptosystem

Key Generation

```
Rabin_Key_Generation
{
    Choose two large primes p and q in the
                        form 4k + 3 and p ≠ q.
    n ← p × q
    Public_key ← n          // To be announced publicly
    Private_key ← (p, q)  // To be kept secret
    return Public_key and Private_key
}
```

Encryption

```
Rabin_Encryption (n, P)
{ // n is the public key; P is the ciphertext from Z_n^*
    C ← P^2 mod n              // C is the ciphertext
    return C
}
```

Note

The Rabin cryptosystem is not deterministic: Decryption creates four plaintexts.

Decryption

Rabin_Decryption (p, q, C)

{ // C is the ciphertext; p and q are private keys

$a_1 \leftarrow +(\mathrm{C}^{(p+1)/4}) \bmod p$

$a_2 \leftarrow -(\mathrm{C}^{(p+1)/4}) \bmod p$

$b_1 \leftarrow +(\mathrm{C}^{(q+1)/4}) \bmod q$

$b_2 \leftarrow -(\mathrm{C}^{(q+1)/4}) \bmod q$

// The algorithm for the Chinese remainder

// algorithm is called four times.

$\mathrm{P}_1 \leftarrow$ Chinese_Remainder (a_1, b_1, p, q)

$\mathrm{P}_2 \leftarrow$ Chinese_Remainder (a_1, b_2, p, q)

$\mathrm{P}_3 \leftarrow$ Chinese_Remainder (a_2, b_1, p, q)

$\mathrm{P}_4 \leftarrow$ Chinese_Remainder (a_2, b_2, p, q)

return P_1, P_2, P_3, and P_4

}

Example

Here is a very trivial example to show the idea.

1. **Bob selects $p = 23$ and $q = 7$. Note that both are congruent to 3 mod 4.**
2. **Bob calculates $n = p \times q = 161$.**
3. **Bob announces n publicly; he keeps p and q private.**
4. **Alice wants to send the plaintext $P = 24$. Note that 161 and 24 are relatively prime; 24 is in Z_{161}^*. She calculates $C = 24^2 = 93 \bmod 161$ and sends the ciphertext 93 to Bob.**

10.3.1 Continued

Example

5. **Bob receives 93 and calculates four values:**

   $a_1 = +(93^{(23+1)/4}) \bmod 23 = 1 \bmod 23$

   $a_2 = -(93^{(23+1)/4}) \bmod 23 = 22 \bmod 23$

   $b_1 = +(93^{(7+1)/4}) \bmod 7 = 4 \bmod 7$

   $b_2 = -(93^{(7+1)/4}) \bmod 7 = 3 \bmod 7$

6. **Bob takes four possible answers, (a_1, b_1), (a_1, b_2), (a_2, b_1), and (a_2, b_2), and uses the Chinese remainder theorem to find four possible plaintexts: 116, 24, 137, and 45. Note that only the second answer is Alice's plaintext.**

10-4 ELGAMAL CRYPTOSYSTEM

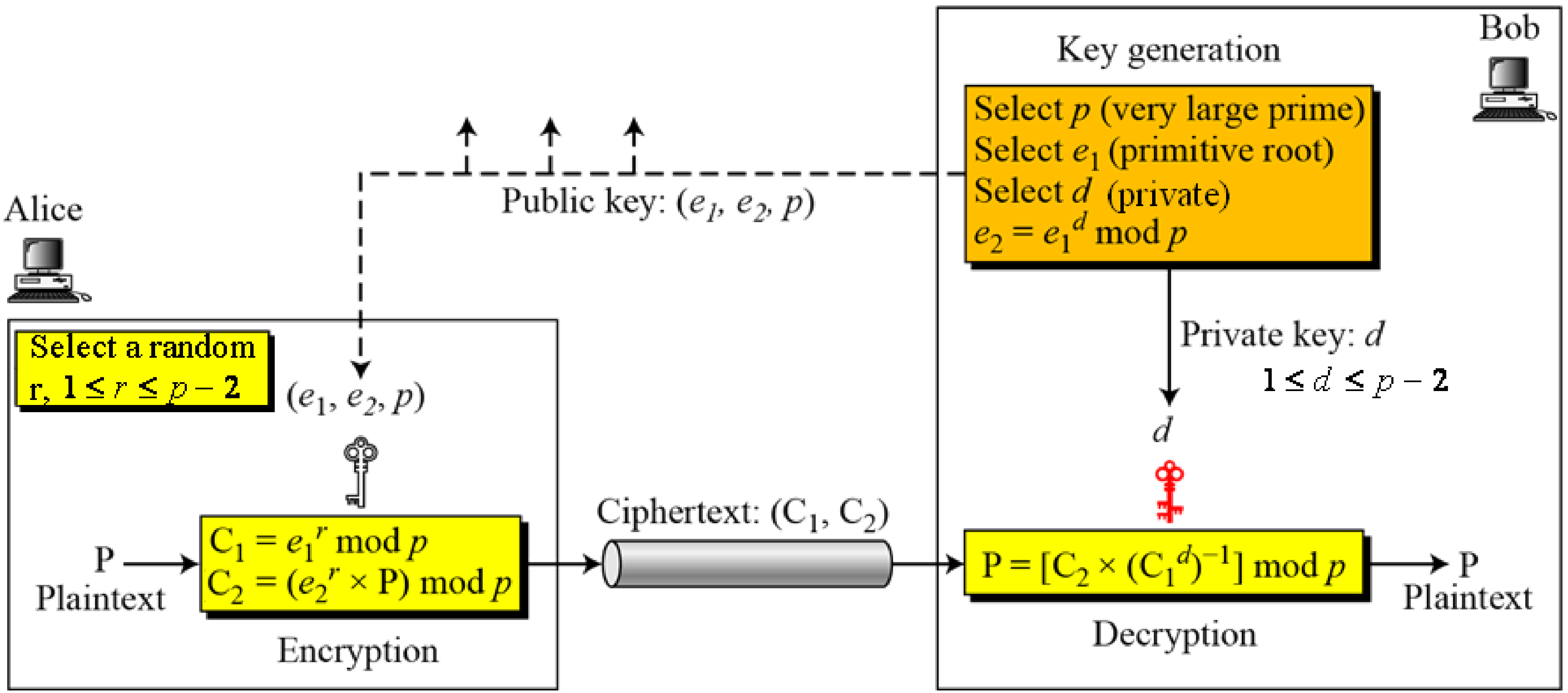
Besides RSA and Rabin, another public-key cryptosystem is ElGamal.

ElGamal is based on the discrete logarithm problem discussed earlier.

Topics discussed in this section:

10.4.1 ElGamal Cryptosystem
10.4.2 Procedure
10.4.3 Proof

10.4.2 Procedure



Key generation, encryption, and decryption in ElGamal

10.4.2 Continued

Key Generation

ElGamal_Key_Generation

{

Select a large prime p

Select d to be a member of the group $\mathbf{G} = \langle \mathbf{Z}_p^*, \times \rangle$ such that $1 \le d \le p - 2$

Select e_1 to be a primitive root in the group $\mathbf{G} = \langle \mathbf{Z}_p^*, \times \rangle$

$e_2 \leftarrow e_1^{\,d} \bmod p$

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

}

Encryption

ElGamal_Encryption $(e_1, e_2, p, \mathrm{P})$ // P is the plaintext

{

Select a random integer r in the group $\mathbf{G} = <\mathbf{Z}_p{}^*, \times>$

$\mathrm{C}_1 \leftarrow e_1{}^r \bmod p$

$\mathrm{C}_2 \leftarrow (\mathrm{P} \times e_2{}^r) \bmod p$ // C_1 and C_2 are the ciphertexts

return C_1 and C_2

}

Decryption

```
ElGamal_Decryption (d, p, C1, C2)
{   // C1 and C2 are the ciphertexts
    P ← [C2 (C1^d)^-1] mod p // P is the plaintext
    return P
}
```

Note

The bit-operation complexity of encryption or decryption in ElGamal cryptosystem is polynomial.

Example

Here is a trivial example. Bob chooses p = 11 and $e_1 = 2$. and d = 3 $e_2 = e_1^d = 8$. So the public keys are (2, 8, 11) and the private key is 3. Alice chooses r = 4 and calculates C_1 and C_2 for the plaintext 7.

Plaintext: 7

$C_1 = e_1^r \bmod 11 = 16 \bmod 11 = 5 \bmod 11$

$C_2 = (P \times e_2^r) \bmod 11 = (7 \times 4096) \bmod 11 = 6 \bmod 11$

Ciphertext: (5, 6)

Bob receives the ciphertexts (5 and 6) and calculates the plaintext.

Plaintext: $[C_2 \times (C_1^d)^{-1}] \bmod 11 = 6 \times (5^3)^{-1} \bmod 11$

$= 6 \times 3 \bmod 11 = 7 \bmod 11$ **Plaintext: 7**

10.4.3 Continued

Example

Instead of using $P = [C_2 \times (C_1^{d})^{-1}] \bmod p$ for decryption, we can avoid the calculation of multiplicative inverse and use $P = [C_2 \times C_1^{p-1-d}] \bmod p$ (see Fermat's little theorem in Chapter 9). In this Example, we can calculate $P = [6 \times 5^{11-1-3}] \bmod 11 = 7 \bmod 11$.

Note

For the ElGamal cryptosystem, *p* must be at least 300 digits and *r* must be new for each encipherment.

10.4.3 Continued

Example

Bob uses a random integer of 512 bits. The integer p is a 155-digit number (the ideal is 300 digits). Bob then chooses e_1, d, and calculates e_2, as shown below:

p =	115348992725616762449253137170143317404900945326098349598143469219 056898698622645932129754737871895144368891765264730936159299937280 61165964347353440008577
e_1 =	2

d =	1007
e_2 =	978864130430091895087668569380977390438800628873376876100220622332 554507074156189212318317704610141673360150884132940857248537703158 2066010072558707455

10.4.3 Continued

Example

Alice has the plaintext P = 3200 to send to Bob. She chooses r = 545131, calculates C_1 and C_2, and sends them to Bob.

P =	3200
r **=**	545131
C_1 **=**	887297069383528471022570471492275663120260067256562125018188351429 417223599712681114105363661705173051581533189165400973736355080295 736788569060619152881
C_2 **=**	708454333048929944577016012380794999567436021836192446961774506921 244696155165800779455593080345889614402408599525919579209721628879 6813505827795664302950

Bob calculates the plaintext $P = C_2 \times ((C_1)^d)^{-1} \bmod p = 3200 \bmod p$.

P =	3200

References

- **Chapter 10 -** Behrouz A Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, Mc Graw Hill, 3rd Edition, 2015.

- **Chapter 9 -** William Stallings, Cryptography and Network Security Principles and Practices, 7th Edition, Pearson Education, 2017.