PUBLIC KEY CRYPTOSYSTEMS

Md. Alamgir Hossain
Senior Lecturer,
Dept. of CSE, Prime University
Mail: alamgir.cse14.just@gmail.com

ASYMMETRIC KEY ENCRYPTION

- The concept of using ***different keys*** at the encryption and decryption ends.
- Depends on different mathematical principles than symmetric encryption.
- Usually done through a combination of hardware and software.
- Can be used for several different applications, other than just encryption.

MISCONCEPTIONS CONCERNING PUBLIC-KEY ENCRYPTION

- ***Public-key encryption*** is more secure from cryptanalysis than symmetric encryption.
  - Not true – they depend on different principles but can be equally secure.
- ***Public-key encryption*** has made symmetric encryption obsolete.
  - Not true – symmetric encryption is still used in several areas, quite successfully.

ASYMMETRIC ENCRYPTION TERMINOLOGY

- ***Asymmetric Keys***
  - Two related keys – a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification
- ***Public Key Certificate***
  - A digital document issued and digitally signed by the private key of the certification authority that binds the name of a subscriber to a public key. The certificate indicates that the subscriber identified in the certificate has sole control and access to the corresponding private key.
- ***Public Key Algorithm***
  - A cryptographic algorithm that uses the related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.
- ***Public Key Infrastructure***
  - A set of policies, processes, server platforms, software, and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue and revoke public key certificates.

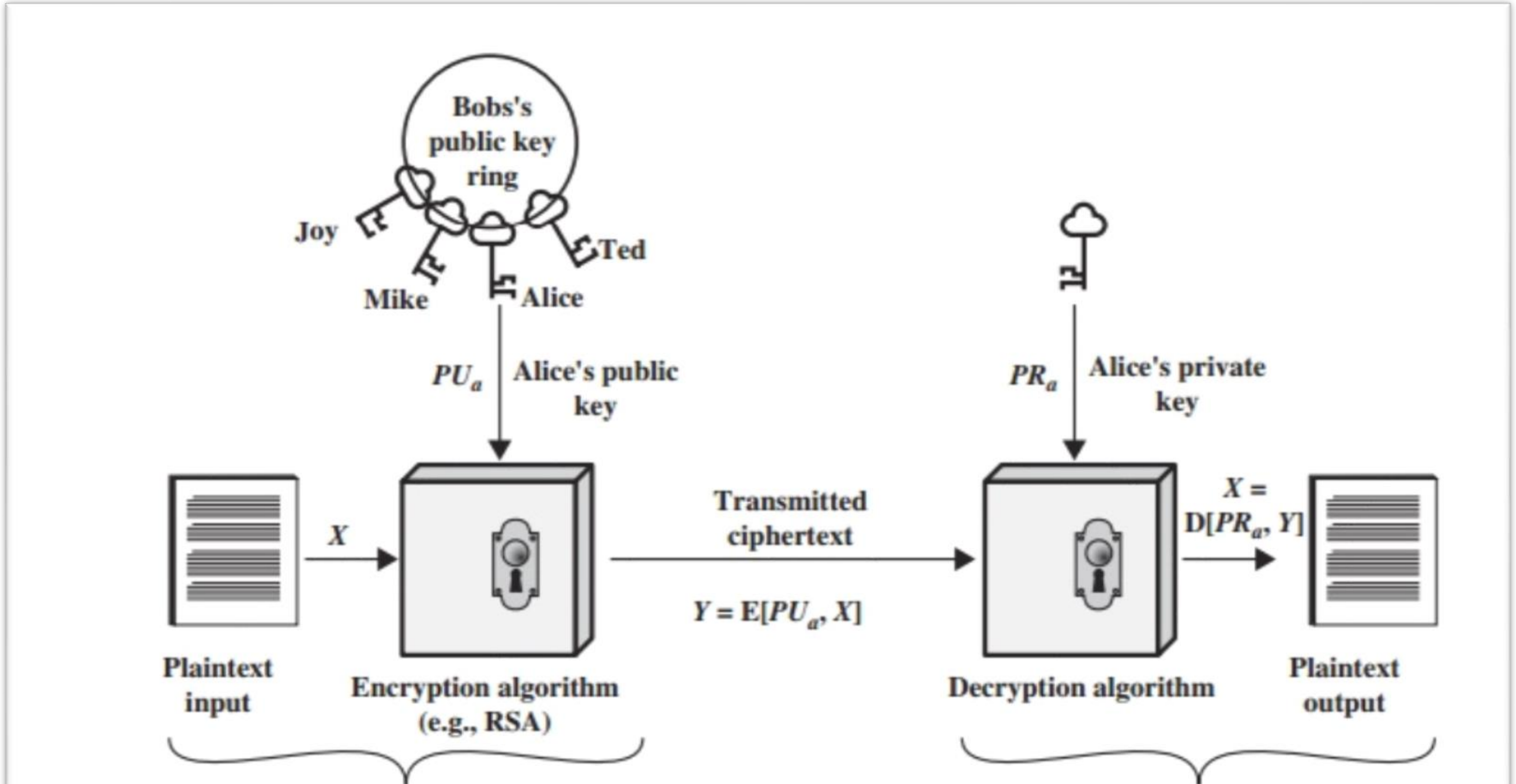
PRINCIPLES OF PUBLIC-KEY CRYPTOSYSTEMS

- Public-key cryptography evolved from an attempt to address the two basic limitations of symmetric encryption:
  - ***Key Distribution*** - How to have secure communication without having to trust a KDC (key distribution center) with your key?
  - ***Digital Signatures*** - How to verify that a message comes intact from the claimed sender?
- Whit Diffie and Martin Hellman proposed a method that addressed both problems and was radically different from all previous approaches to cryptography.

PUBLIC-KEY CRYPTOSYSTEMS TERMINOLOGY

- Plaintext – the input data
- Encryption Algorithm – Performs various transformations on the plaintext
- Public Key – Used for encryption
- Private Key – Used for decryption
- Ciphertext – The output data
- Decryption Algorithm – Used for decryption

PUBLIC-KEY CRYPTOGRAPHY-ENCRYPTION

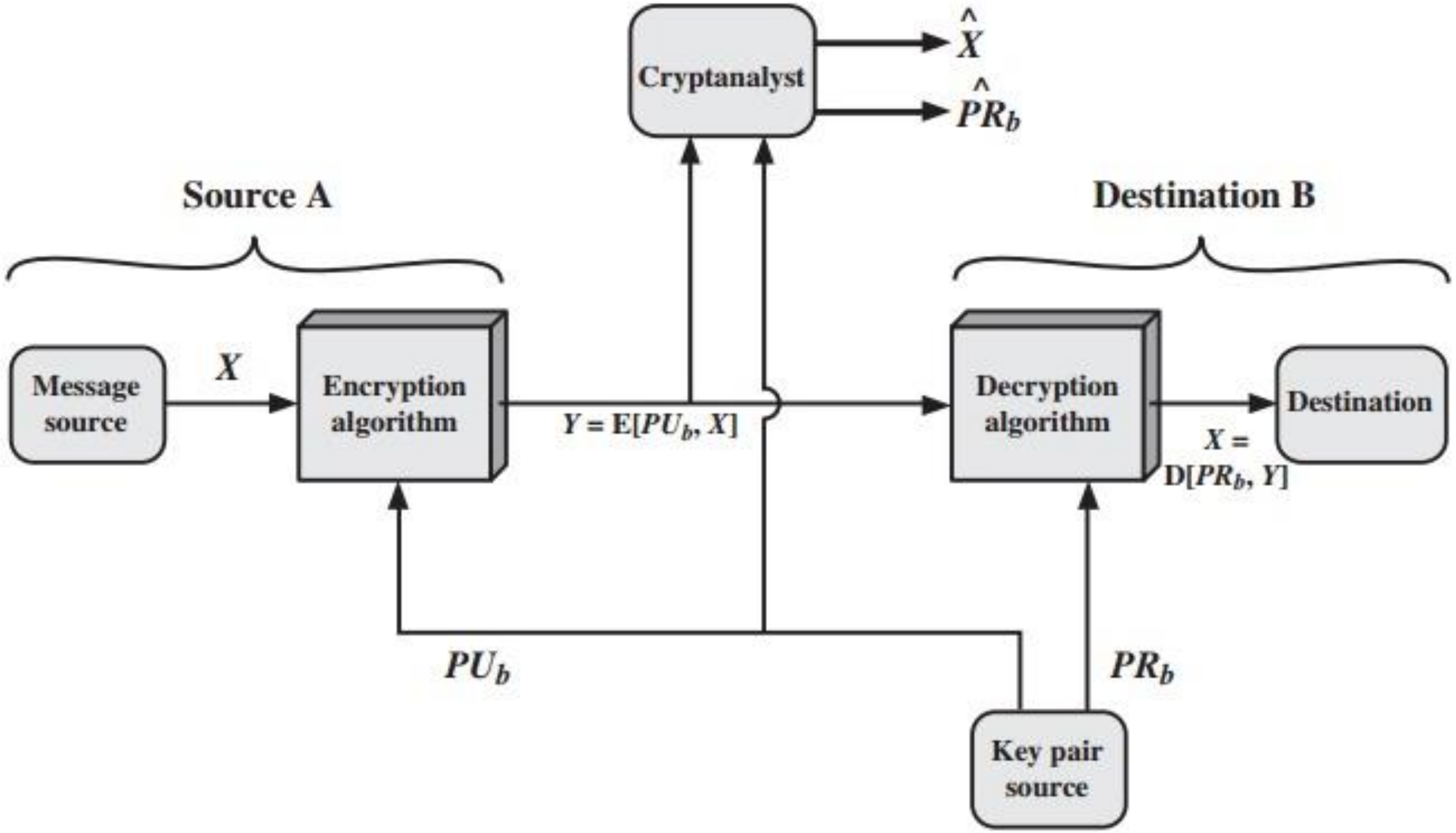


(a) Encryption with public key

CONVENTIONAL AND PUBLIC-KEY ENRYPTION

Conventional Encryption	Public-Key Encryption
Needed to Work: I. The same algorithm with the same key is used for encryption and decryption. II. The sender and receiver must share the algorithm and the key.	***Needed to Work:*** I. One algorithm is used for encryption and a related one for decryption. II. The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security:	***Needed for Security:***
I. The key must be kept secret. II. It must be impossible or at least impractical to decipher a message if the key is kept secret. III. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the other key.	I. One of the two keys must be kept secret. II. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret. III. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other keys.

PUBLIC-KEY CRYPTOSYSTEM: SECRECY



APPLICATIONS FOR PUBLIC KEY CRYPTOSYSTEMS

Public-key cryptosystems can be classified into three categories:

- ***Encryption/decryption:*** The sender encrypts a message with the recipient's public key
- ***Digital Signatures:*** The sender "signs" a message with its private key
- ***Key Exchange:*** Two sides cooperate to exchange a session key
- Some algorithms are suitable for all three applications, whereas others can be used only for one or two.

PUBLIC KEY REQUIREMENTS

- ***Computationally easy***
  - for party B to generate a pair (public-key PUb , private key PRb)
  - for sender A, knowing the public key and the message, to generate the corresponding ciphertext
  - for receiver B to decrypt the resulting ciphertext using the private key to recover the
- ***Computationally infeasible for an adversary***
  - knowing the public key, to determine the private key
  - knowing the public key and a ciphertext, to recover the original message

PUBLIC KEY REQUIREMENTS

- Need a trap-door one-way function
  - f : $\{0,1\}^n$ -> $\{0,1\}$n is a one-way function if
    - $Y = f(X)$ can easily be computed for X in $\{0,1\}^n$
    - $X = f^{-1}(Y)$ infeasible for Y in $\{0,1\}^n$
- A trap-door one-way function is a family of invertible functions f_k , such that computing
  - $Y = f_k(X)$ is easy, if k and X are known
  - $X = f_k^{-1}(Y)$ is easy, if k and Y are known
  - $X = f_k^{-1}(Y)$ infeasible, if Y is known but k not known
- A practical public-key scheme depends on a suitable trapdoor one-way function

RIVEST-SHAMIR-ADLEMAN (RSA) SCHEME

- Developed in 1977 by Ron Rivest, Adi Shamir & Len Adleman.
- Most widely used general-purpose public-key encryption.
- A cipher for which the plaintext and ciphertext are integers between 0 and n – 1 for some n.
  - A typical size for n is 1024 bits, or 309 decimal digits.

RSA ALGORITHM

- Plaintext is encrypted in blocks with whose value less than some number n
- Encryption and decryption are of the following form, for plaintext block M and ciphertext block C
- $C = M^e \bmod n$
- $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$
- Both sender and receiver must know the value of n
- The sender knows the value of e, and only the receiver knows the value of d
- This is a public-key encryption algorithm with a public key of PU={e, n} and a private key of PR={d, n}

ALGORITHM REQUIREMENTS

- It should be possible to find values of e, d, n such that
  $M^{ed} \bmod n = M$ for all $M < n$
- It should be relatively easy to calculate
  $M^e \bmod n$ and $C^d \bmod n$ for all values of $M < n$
- It should be infeasible to determine d given e and n.

RSA ALGORITHM

Key Generation	
Select p, q	p and q both prime $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \bmod \phi(n)$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption	
Plaintext	$M < n$
Ciphertext	$C = M^e \pmod{n}$

Decryption	
Ciphertext	C
Plaintext	$M = C^d \pmod{n}$

EXAMPLE OF RSA ALGORITHM

Select two prime no. Suppose P = 53 and Q = 59.

Now First part of the Public key : n = P*Q = 3127.

We also need a small exponent say e : But e Must be An integer.

1 < e < Φ(n) [Φ(n) is discussed below], Let us now consider it to be equal to 3.

Our Public Key is made of n and e.

Generating Private Key :

We need to calculate Φ(n) : Such that Φ(n) = (P-1)(Q-1). so, Φ(n) = 3016

Now calculate Private Key, d : d = (k*Φ(n) + 1) / e for some integer k

For k = 2, value of d is 2011.

Now we are ready with our – Public Key (n = 3127 and e = 3) and Private Key(d = 2011)

FINDING THE VALUE OF "d"

- d = e^{-1} mod Φ(n). => de = 1 mod Φ(n). What does it mean? It means "***de mod Φ(n) = 1***"(Basic theorem of inverse modular arithmetic).

Φ(n)) de (x

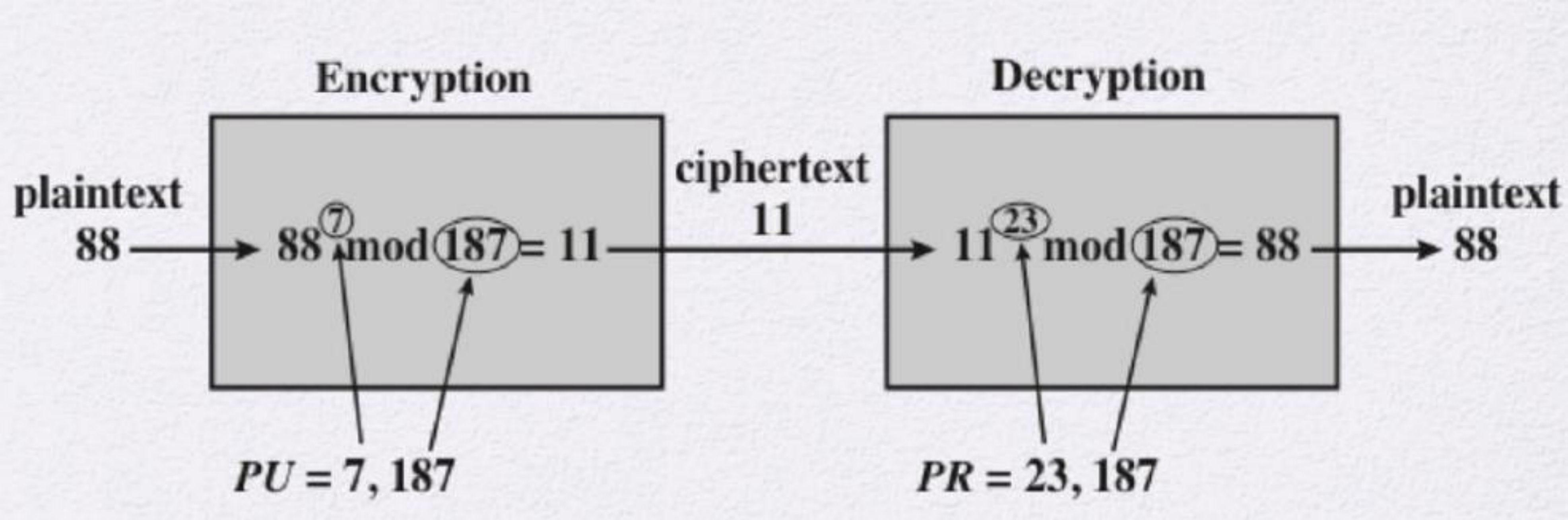
1

- ***=> Φ(n) * X + 1 = de; => d = {Φ(n) * X + 1} / e; d must be an integer number.***

EXAMPLE OF RSA ALGORITHM

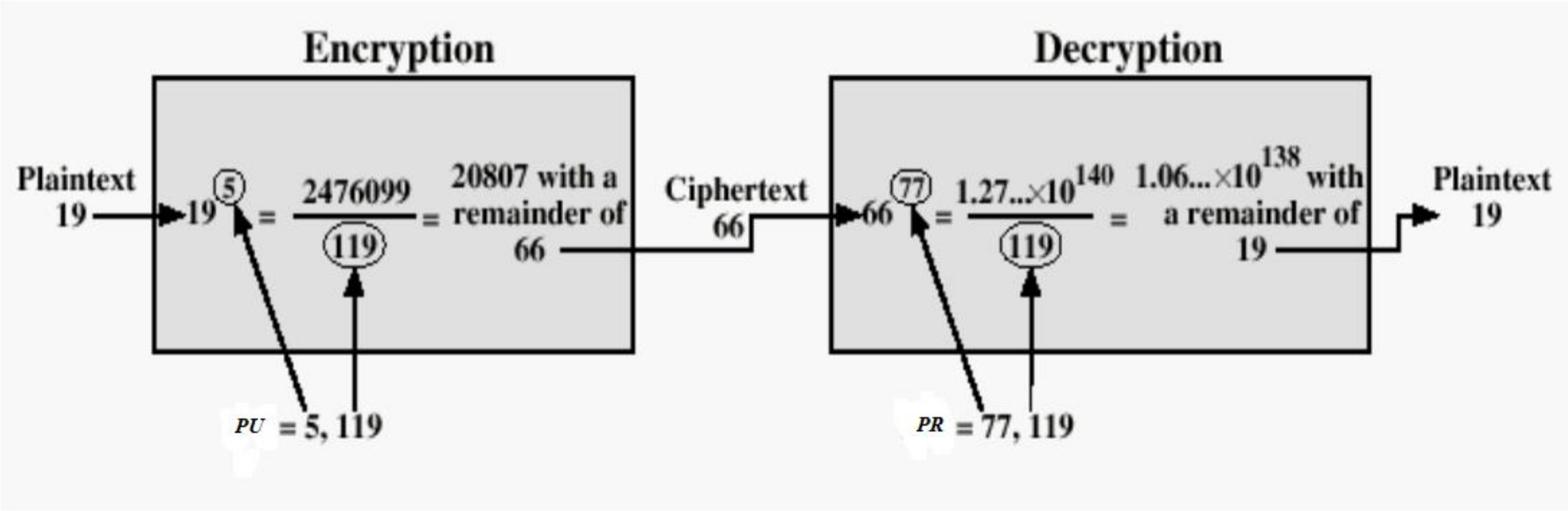
- M = f/6, p = 7, q = 17, n = 119, Φ(n) = 96, e = 5, C = 41, d = 77, M = 6.
- M = 2, p = 3, q = 11, n = 33, Φ(n) = 20, e = 7, C = 29, d = 3, M = 2.
- M = 9, p = 7, q = 11, n = 77, Φ(n) = 60, e = 7, C = 37, d = 43, M = 9.

EXAMPLE OF RSA ALGORITHM



P = 17, q = 11, n = 187, φ (n) = 160, e = 7, d = 23

EXAMPLE OF RSA ALGORITHM



P = 17, q = 7, n = 119, φ (n) = 96, e = 5, d = 77

EXAMPLE OF RSA ALGORITHM (TEXT)

- Encrypt and Decrypt: “HI”.
- Convert letters to numbers : H = 8 and I = 9.
- **P = 7, q = 17, n = 119, φ (n) = 96, e = 5, d = 77**
- Thus Encrypted Data c = 89^{e} mod n.
- Thus our Encrypted Data comes out to be 38
- Decrypted Data = c^{d} mod n.
- Thus our Deccrypted Data comes out to be 89
- H = 8 and I = 9 i.e. “HI”.

EXAMPLE OF RSA ALGORITHM (TEXT)

Plaintext: HELLO

1) Choose two prime numbers p = 5 and q = 7

2) Compute n = pq and m = (p – 1)(q – 1)

   a. n = pq = (5)(7) = 35

   b. m = (p – 1)(q – 1) = (5 – 1)(7 – 1) = (4)(6) = 24

3) Choose a number e < n such that it has no common factors with z other than

1, Let e = 5

4) Find a number d such that ed divided z has a remainder of 1, d = 5 or 29.

5) The public key becomes (n, e) and the private key becomes (n, d).

EXAMPLE OF RSA ALGORITHM (TEXT)

Encryption:

Plain Text	h	e	l	l	o
Number	8	5	12	12	15
Encrypted value	8	10	17	17	15

EXAMPLE OF RSA ALGORITHM (TEXT)

Decryption: Do it!!

EXAMPLE OF RSA ALGORITHM (TEXT)

- This time, to make life slightly less easy for those who can crack simple Caesar substitution codes, we will group the characters into blocks of three and compute a message representative integer for each block.
- ***Please note that this method is not secure in any way.***
- It just shows another example of the mechanism of RSA with small numbers.
- For this example, to keep things simple, we'll limit our characters to the letters A to Z and the space character.
- ***Message:*** ATTACK AT SEVEN = ATT ACK _AT _SE VEN

EXAMPLE OF RSA ALGORITHM (TEXT)

- In the same way that any decimal number can be represented uniquely as the sum of powers of ten, e.g. $135=1\times10^2+3\times10^1+5\times10^0$,
- we can represent our blocks of three characters as the sum of powers of 27 using SPACE=0, A=1, B=2, C=3, .. E=5, .. K=11, .. N=14, .. S=19, T=20, .. V=22, …, Z=26.

— ATT ⇒ 1 x 27^2 + 20 x 27^1 + 20 = 1289

— ACK ⇒ 1 x 27^2 + 3 x 27^1 + 11 = 821

— _AT ⇒ 0 x 27^2 + 1 x 27^1 + 20 = 47

— _SE ⇒ 0 x 27^2 + 19 x 27^1 + 5 = 518

— VEN ⇒ 22 x 27^2 + 5 x 27^1 + 14 = 16187

EXAMPLE OF RSA ALGORITHM (TEXT)

- Using this system of integer representation, the maximum value of a block (ZZZ) is $27^3-1=19682$, so we require a modulus n greater than this value.

1. We choose e = 3
2. We select primes p=173 and q=149 and check
   - gcd(e, p-1) = gcd(3, 172) = 1 ⇒ OK
   - gcd(e, q-1) = gcd(3, 148) = 1 ⇒ OK.
3. Thus we have $n = pq = 173 \times 149 = 25777$, and $\phi = (p-1)(q-1) = 172 \times 148 = 25456$.
4. We compute $d = e^{-1} \bmod \phi = 3^{-1} \bmod 25456 = 16971$.
   - Note that $ed = 3 \times 16971 = 50913 = 2 \times 25456 + 1$
   - That is, $ed \equiv 1 \bmod 25456 \equiv 1 \bmod \phi$
5. Hence our public key is (n, e) = (25777, 3) and our private key is (n, d) = (25777, 16971). We keep the values of p, q, d and ϕ secret.

EXAMPLE OF RSA ALGORITHM (TEXT)

- Overall, our plaintext ATTACK AT SEVEN is represented by the sequence of five integers m_1,m_2,m_3,m_4,m_5: (1289, 821, 47, 518, 16187).

We compute corresponding ciphertext integers $c_i = m_i^e \bmod n$, (which is still possible by using a calculator, honest):

$$c_1 = 1289^3 \bmod 25777 = 18524$$
$$c_2 = 821^3 \bmod 25777 = 7025$$
$$c_3 = 47^3 \bmod 25777 = 715$$
$$c_4 = 518^3 \bmod 25777 = 2248$$
$$c_5 = 16187^3 \bmod 25777 = 24465$$

We can send this sequence of integers, c_i, to the person who has the private key.

```
c_i = (18524, 7025, 715, 2248, 24465)
```

EXAMPLE OF RSA ALGORITHM (TEXT)

- We can compute the inverse of these ciphertext integers using $m = c^d \bmod n$.

$$m_1 = 18524^{16971} \bmod 25777 = 1289$$
$$m_2 = 7025^{16971} \bmod 25777 = 821$$
$$m_3 = 715^{16971} \bmod 25777 = 47$$
$$m_4 = 2248^{16971} \bmod 25777 = 518$$
$$m_5 = 24465^{16971} \bmod 25777 = 16187$$

To convert these integers back to the block of three letters, do the following. For example, given m = 16187,

```
16187 ÷ 27^2 = 16187 ÷ 729 = 22 rem 149, 22 → 'V'
149   ÷ 27^1 = 149   ÷ 27  = 5 rem 14,    5 → 'E'
14    ÷ 27^0 = 14    ÷ 1   = 14 rem 0,   14 → 'N'
```

Hence the integer m = 16187 represents the string "VEN".
Similarly, m = 47 is encoded as follows:

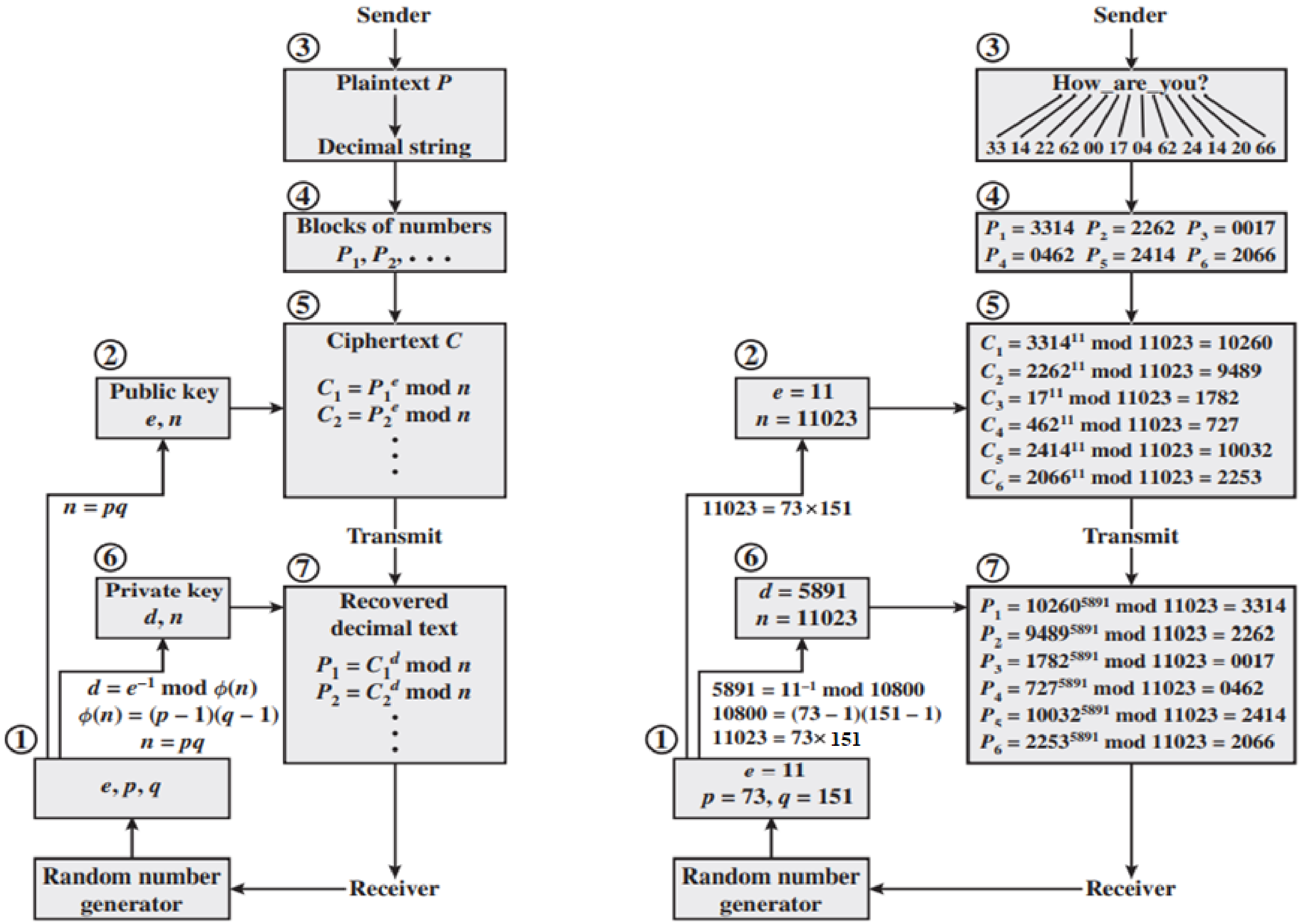
```
47 ÷ 27^2 = 0 rem 47, 0  → SPACE;
47 ÷ 27^1 = 1 rem 20, 1  → 'A';
20 ÷ 27^0 = 20 rem 0, 20 → 'T'
```

giving the string "_AT".

EXAMPLE OF RSA ALGORITHM (TEXT)

- RSA algorithm is used to process multiple blocks of data. In this simple example, the plaintext is an alphanumeric string.
- Each plaintext symbol is assigned a unique code of two decimal digits (e.g., a = 00, A = 26).
- A plaintext block consists of four decimal digits, or two alphanumeric characters.
- The sequence of events for the encryption of multiple blocks is as shown (the circled numbers indicate the order in which operations are performed).

EXAMPLE OF RSA ALGORITHM (TEXT)



(a) General approach (b) Example

ATTCKS IN RSA ALGORITHM

- ***Brute force***: This involves trying all possible private keys.
- ***Mathematical attacks:*** There are several approaches, all equivalent in effort to factoring the product of two primes.
- ***Timing attacks:*** These depend on the running time of the decryption algorithm.
- ***Hardware fault-based attack:*** This involves inducing hardware faults in the processor that is generating digital signatures.
- ***Chosen ciphertext attacks:*** This type of attack exploits properties of the RSA algorithm.

EXPONENTIATION IN MODULAR ARITHMETIC

- Both encryption and decryption in RSA involve raising an integer to an integer power, mod n.
- Can make use of a property of modular arithmetic:

  [(a mod n) x (b mod n)] mod n =(a x b) mod n
- With RSA you are dealing with potentially large exponents, so efficiency of exponentiation is a consideration.

EFFICIENT OPERATION USING THE PRIVATE KEY

- Decryption uses exponentiation to power d
- A small value of d is vulnerable to a brute force attack and to other forms of cryptanalysis
- Can use the Chinese Remainder Theorem (to speed up computation
  - The quantities d mod (p - 1) and d mod (q - 1) can be precalculated
  - Result is that the calculation is approximately four times as fast as evaluating $M = C^d \bmod n$ directly

ADVANTAGES OF ASYMMETRIC ENCRYPTION

- It's more secure than ***Symmetric Encryption.***
- It's useful when more endpoints are involved.
- Makes key distribution easy.
- Makes digital signatures possible.

DISADVANTAGES OF ASYMMETRIC ENCRYPTION

- Slower Speed
- It’s Too Bulky to Be Used at Scale

RECOMMENDATION OF READING

- https://www.brainkart.com/article/Principles-of-Public-Key-Cryptosystems-and-its-Applications,-Requirements,-Cryptanalysis_8435/
- https://sectigostore.com/blog/what-is-asymmetric-encryption-how-does-it-work/
- https://www.geeksforgeeks.org/rsa-algorithm-cryptography/
- https://www.di-mgt.com.au/rsa_alg.html
- YouTube Videos
- Online PowerMod Calculator: https://www.mtholyoke.edu/courses/quenell/s2003/ma139/js/powermod.html

Thank You