Cryptography, CSCI 57313, Summer 2022 (D. Nguyen)
William Jessup University

SUMMARY 1

Generally, we assume that the message that we wish to send has been converted to an integer in the set $J_m = \{0, 1, 2, 3, \ldots, m-1\}$, where m is some integer (positive) to be determined.

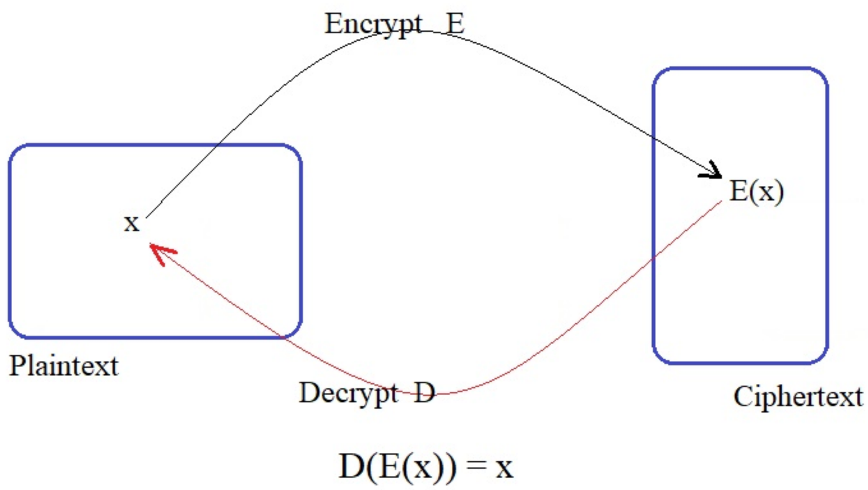
For **Encipher**

$$\begin{aligned} E : J_m &\longrightarrow \quad J_m \\ x &\longmapsto \quad E(x) \end{aligned}$$

For **Decipher**

$$D : J_m \longrightarrow \quad J_m$$

$$\text{such that} \quad D\left(E(x)\right) = x$$



1 Encoding a Phrase in a Number

In order to use almost cryptosystem to encrypt messages(plaintext), it is necessary to encode them as a sequence of numbers of size less than a number n. We now describe a simple way to do this. We use **Python** (*or Sagemath, Python language*) to implement conversion between a string and a number. The input string s on a computer is stored in a format called ASCII, so each "letter" corresponds to an integer between 0 and 255, inclusive (or Advanced ASCII, to 1024, or 2048, or 4096 for some languages like German, Russian, ...). This number is obtained from the letter using the **ord** command.
(See the Sample1)

2 The simplest encryption : Caesar encryption

Caesar encryption is a type of shift cryptosystem, is a symmetric-key cryptosystem in which each secret key **k** is an element of $\mathbb{Z}/n\mathbb{Z}$, it is clear that the key space $\mathbb{Z}/n\mathbb{Z}$ consists of n possible keys. Let $P = (p_0, p_1, p_2, ..., p_{m-1})$ be a non-empty plaintext consisting of m elements,

i.) Convert the elements of the plaintext into numbers **P** (including any notations : ' ' (blank), #, !, ?, ...,)

ii.) Use the secret key **k** to produce a ciphertext **C**:

$$C = P + k \ (mod\ n)$$

(n may be chosen as the base, the number of alphabets or, if using the ASCII table : $n = 256$, or 1024, 2048, ...). The ciphertext is **C**, may convert to alphabets before sending.

iii.) Decryption Key

$$P = C + k \ (mod \ n)$$

and convert to the alphabets to get the original plaintext. *(See the Sample2)*

3 Exponential encryption (Deffie-Hellman encryption)

i.) Choose a (large) prime $\boldsymbol{p}$ and a positive integer $\boldsymbol{e}$ such that $gcd(e, p-1) = 1$ (the number $\boldsymbol{e}$ should be chosen a prime). **The key of the encryption is the pair** (e, p) **(keep secret !)**

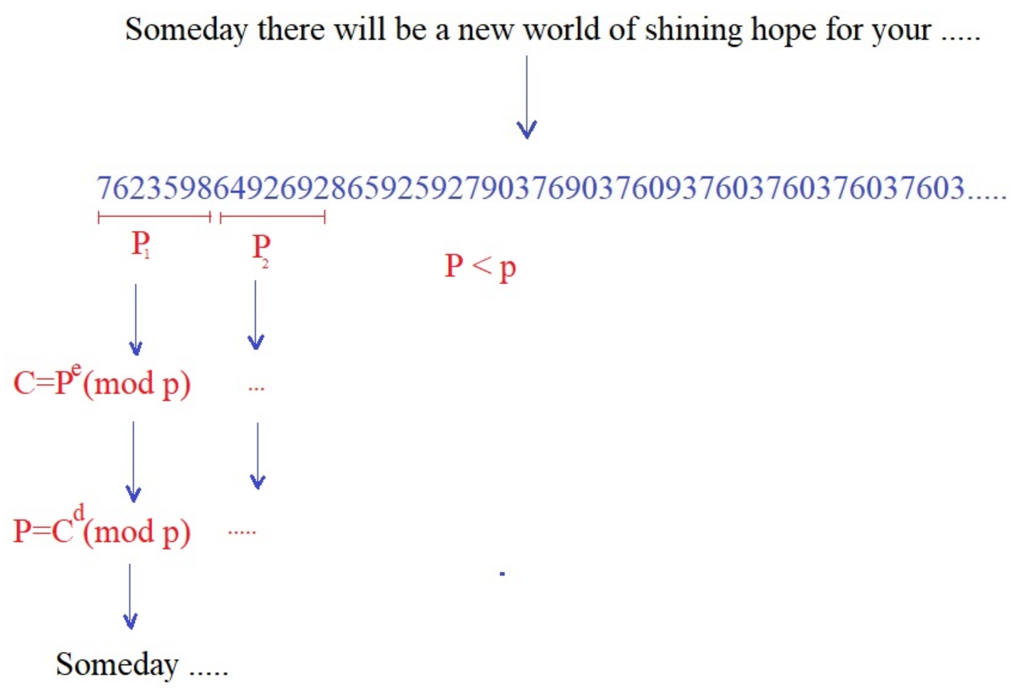
ii.) Convert the elements of the plaintext into numbers (including any notations : ' ' (blank), #, !, ?, ...,)

iii.) Grouping the numbers in groups, such that the value P of each group: $P < \boldsymbol{p}$

iv.) Encryption: For the number P of each group, using the encryption key to produce the ciphertext C:

$$C = P^e \ (mod \ p)$$

C is the ciphertext.



v.) Decryption: using the encryption key (e, p) to determine the decryption key $\boldsymbol{d}$ as the multiplicative inverse of $\boldsymbol{e}$ modulo $(p-1)$:

$$e.d = 1 \ (mod \ (p-1))$$

and for each number C (the ciphertext) get back to the number P by

$$P = C^d \ (mod \ p)$$

From P , convert to the original plaintext.

(See the Sample3)

From (e, p) someone can find the decryption key d, therefore the key of the encryption (e, p) must keep

secret, at least keep secrete e. Knowing p only, it is very difficult to find d, the fastest algorithm currently needs about $e^{\sqrt{\log(p)\log(\log(p))}}$ operations of bits.

4 The RSA encryption (Rivest-Shamir-Adleman)

RSA is one of the first public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, **the encryption key is public** and **distinct from the decryption key which is kept secret** (private). In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem".

i.) Choose two large primes p and q and let $n = pq$. Choose a positive integer e such that $gcd(e, \varphi(n)) = 1$, where $\varphi(n)$ is the Euler function. Since p, q are primes : $\varphi(n) = \varphi(pq) = (p-1)(q-1)$. The number e should be chosen as a large prime. **The encryption key (public) is the pair** (e, n), **but** p, q **keep secret**!

ii.) Let d be the multiplicative inverse of e modulo $\varphi(n)$:

$$e.d = 1\ (mod\ \varphi(n))$$

The decryption (private) key is the pair (d, n).

iii.) Encryption: Similarly as the Exponential encryption, step i.) to step iii.), where $P < n$, using the public key

(e, n) to produce the ciphertext

$$C = P^e \ (mod\ n)$$

C is the ciphertext.

iv.) Decryption: using the decryption (private) key (d, n) to get back to the number P

$$P = C^d \ (mod\ n)$$

From P , convert to the original plaintext.

(See the Sample4)