Learning Outcome 7 – Encryption

Encryption is a method to hide a message so that only the intended person can read it. Plaintext (P) is the original message, and Ciphertext (C) is the scrambled version created by an encryption function. A key (k) determines how the message is scrambled or unscrambled.

Terminology

Plaintext (P) – the original message to be encrypted

Ciphertext (C) – the output of the encryption process or the encrypted (scrambled) version

Notation

Encryption can be described using the following function:

$$C = E_k(P)$$

where E_k is the encryption function based on some key k, where key (k) is used to encrypt and decrypt.

Decryption can be described using the following function:

$$P = D_k(C)$$

Where D_k is the decryption process based on some key k.

Substitution ciphers (Replacing letters)

Each letter or group of letters is replaced by another letter or group of letters to disguise it. As an example, the alphabet can be mapped to other letters as follows:

P	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
C	q	w	e	r	t	y	u	i	o	p	a	s	d	f	g	h	j	k	l	z	x	c	v	b	n	m

So, for instance, for the plaintext P = “hello”

the ciphertext would be C = “itssg”.

Easy idea—but not secure today

This system is known as monoalphabetic substitution. This may appear to be safe since there are 26! = 4×10^{26} possible keys. However, statistical properties of the natural language make the code easy to break. Also, this process requires that each of the sender and the receiver has the key – how do we transmit the key securely?

Because patterns in English (like how often certain letters appear) let attackers guess the key.

Transposition ciphers (Rearranging letters)

Instead of replacing letters, a transposition cipher reorders them. Substitution ciphers preserve the order of the plaintext symbols but disguise them. Transposition ciphers reorder the letters, but do not disguise them. The cipher is keyed on a word or phrase not containing any repeating letters. For example, suppose we want to encode the phrase:
Plaintext (P) = robmillerisawesome

The following cipher uses the key “linux”

Key word: **linux**

l	**i**	**n**	**u**	**x**
2	**1**	**3**	**4**	**5**
r	o	b	m	i
l	l	e	r	i
s	a	w	e	s
o	m	e	a	b
c	d	e	f	g

The purpose of the key is to number the columns with 1 being the closest to the start of the alphabet and so on. We then read the columns vertically, starting with the lowest number:

Step 1: Write numbers below key based on alphabetical order

Step 2: Write message in rows

Step 3: Read columns in order 1 → 5

Ciphertext (C) = olamdrlsocbeweemreafiisbg

To decode the message, only the key is needed. The problem then becomes “How is the key transmitted securely to the intended receiver?” Also, the process is relatively easy to crack.

Because this is easy to reverse if you know the **key word**.
But the key must be shared secretly — a big problem in old systems.

Modern encryption systems involve a public and private key. The method of encryption is also public. One of the strengths of the system is that the **private key is not transmitted**.

RSA (Modern System)

Modern systems (like RSA) fix the key-sharing problem.
Public key → shared with everyone
Private key → kept secret
Anyone can encrypt a message to you using your public key,
but only you can decrypt it with your private key.
This works because of one-way functions (easy one way, hard reverse).

The challenge of public key cryptography is to develop a system in which it is impossible/extremely difficult to determine the private key. This is done through a one-way function. With a one-way function, it is relatively easy to compute a result given a set of input values. However,

it is extremely difficult to determine the original value starting only with the result. The one-way function used in RSA is simply the multiplication of two large prime numbers (1024-bit or more). It is easy to multiply two big primes to get a result, but given the result, it can be extremely time-consuming and hard to factor it.

21 = 3 * 7

91 = 7 * 13

1073 = 29 * 37

Must determine the factors using "brute-force techniques". A brute force method is a problem-solving technique that involves systematically trying every possible solution until the correct one is found.

Public key cryptography uses functions like this to build a cryptosystem, or in other words, to build private key and the product of primes to build the public key. The function is one-way for large prime numbers (1024-bit minimum).

Steps involved in RSA

1. Begin by taking two large prime numbers (1024-bit minimum) called *p* and *q*. Calculate their product as $n = p * q$.
2. Create the public key and private key as follows:
    a. Calculate the **modulus** $z = (p-1)*(q-1)$
    b. Pick a random number *e* such that:
        i. $e < z$
        ii. *e* and *z* must be **relatively prime**, which means their greatest common factor is 1.
    c. Compute a number *d* such that $ed\%z = 1$. (% is modulo operator, e.g., 5 % 2 = 1, meaning the remainder when 5 is divided by 2).

d. The public key is (*n*, *e*) and the private key is (*n*, *d*).

3. To send a message, the public key is passed to the sender. The sender encrypts the message using the formula:

   $C = P^e \% n$

4. The recipient then decodes the message using the formula:

   $P = C^d \% n$

Example of RSA

A recipient picks the values $p = 3$ and $q = 11$ (in reality, these numbers would be much bigger).

$n = p * q = 3 * 11 = 33$

Calculate the modulus

$z = (p-1)*(q-1) = (3-1)*(11-1) = 2 * 10 = 20$

Pick a number *e* according to the rules

$e = 7$

(Pick a random number *e* such that, $e < z$ *and,*

e and *z* must be **relatively prime**, which means their greatest common factor is 1) so, 7 is prime, but *e* does not have to be prime as long as it is relatively prime to 20 and it is less than 20. We could also have picked 9, which isn't prime.)

Compute *d* according to the formula:

$ed\%z = 1$

$7d\%20 = 1$

$7d = 20x + 1$

(For some whole number *x*, where *x* is the number of times 20 divides into the product 7*d*. Solve using trial and error.)

Solve for d:

$d = (20x + 1)/7$

Try $x = 1$

$d = (20 * 1 + 1)/7 = 21/7$

$d = 3$ (**an integer value**)

(Lucky, we got it on the first try; if not, try $x = 2$, $x = 3$, etc.)

Private key is $(n, d) = (33, 3)$

Public key is $(n, e) = (33, 7)$

To send a message, the receiver would send the public key to the sender of the message. The sender wants to send the message P = 13 (note that the value of P that we are encrypting must be less than the modulus z).

Encrypt using the formula:

$C = P^e \% n$

$C = (13)^7 \% 33$

$C = 7$

The sender transmits C = 7.

The recipient decodes using the private key (33, 3):

$P = C^d \% n$

$P = 7^3 \% 33$

$P = 13$

Example: One more example of RSA:

$p = 17$ and $q = 23$

$n = p*q = 391$

$z = (p-1)*(q-1) = 16*22 = 352$

Pick a number $e = 29$

Compute d according to $ed\%z = 1$, so $29d = 352x + 1$, so $d = (352x + 1)/29$; by trial and error, we arrive at $d = 85$ (**an integer value**) (when $x = 7$)

So public key is $(n, e) = (391, 29)$

Private key is $(n, d) = (391, 85)$

Suppose the sender wishes to send the message "HI", with "H" being encoded as 8 and "I" being encoded as 9.

$C = P^e \% n = 8^{29} \% 391 = 196$

$C = P^e \% n = 9^{29} \% 391 = 280$

So, the sender transmits 196 and 280

The recipient gets 196 and 280 and decodes it using:

$P = C^d \% n = 196^{85} \% 391 = 8$

$P = C^d \% n = 280^{85} \% 391 = 9$

Additional RSA details

a. A practical way to use the algorithm is to convert the message to hexadecimal and perform encryption and decryption steps on each octet (8 bits) individually
b. A popular choice for the public exponent e is $2^{16} + 1 = 65537$. Small devices often use smaller values to make encryption faster, but at the expense of greater security risks.

c. Exponentiation calculations in formulas $C = P^e \% n$ and $P = C^d \% n$ are done by **Exponentiation by Squaring** which has an efficiency of O(log n) vs. O(n) when multiplying a value by itself n times.
d. Plaintext values are padded in practice since certain values such as 0 and 1 are encrypted as 0 and 1 respectively.
e. Steps for determining *e* and *d* can be done using the Extended Euclidean algorithm.
f. For comparably secure systems, it has been possible to achieve a throughput 1000 to 10000 times higher for single-key algorithms than for two-key algorithms (like RSA). As a result, the main application of two-key cryptography is in hybrid systems. In such a system, a two-key algorithm is used for authentication and digital signatures or to exchange a randomly generated session key to be used with a single-key algorithm at high speed for main communication. At the end of the session, the session key is discarded.

Practice question:

Sam would like to start receiving encoded messages from her friend Rom. Rob wants to send the message '7'. Sam wants all messages to be encoded using the RSA encryption system.

a. What steps would Sam have to take before Rob could send the message?
b. What steps would Rob have to take to send the message?
c. What would Sam have to do to decode the message?
d. What is the strength of the RSA algorithm?
e. Show all necessary calculations using p = 13 and q = 17. Choose e as per the rule.