CST803: ADVANCED CRYPTOGRAPHY

AFRICA CENTRE OF EXCELLENCE ON TECHNOLOGY ENHANCED LEARNING (ACETEL)

NATIONAL OPEN UNIVERSITY OF NIGERIA

Course Guide for CST803

Introduction

CST803 – Advanced Cryptography is a 3-credit unit. The course is a core course in first semester. It will take you 15 weeks to complete the course. You are to spend 91 hours of study for a period of 13 weeks while the first week is for orientation and the last week is for end of semester examination. The credit earned in this course is part of the requirement for graduation.

You will receive the course material which you can read online or download and read off-line. The online course material is integrated in the Learning Management System (LMS). All activities in this course will be held in the LMS. All you need to know in this course is presented in the following sub-headings.

Course Competencies

By the end of this course, you will gain competency to:

- Protect data at rest and during transmission

Course Objective

The course objective is to:

- Design and implement security applications

Working Through this Course

The course is divided into modules and units. The modules are derived from the course competencies and objectives. The competencies will guide you on the skills you will gain at the end of this course. So, as you work through the course, reflect on the competencies to ensure mastery. The units are components of the modules. Each unit is sub-divided into introduction, intended learning outcome(s), main content, self-assessment exercise(s), conclusion, summary, and further readings. The introduction introduces you to the unit topic. The intended learning outcome(s) is the central point which help to measure your achievement or success in the course. Therefore, study the intended learning outcome(s) before going to the main content and at the end of the unit, revisit the intended learning outcome(s) to check if you have achieved the learning outcomes. Work through the unit again if you have not attained the stated learning outcomes.

The main content is the body of knowledge in the unit. Self-assessment exercises are embedded in the content which helps you to evaluate your mastery of the competencies. The conclusion gives you the takeaway while the summary is a brief of the knowledge presented in the unit. The final part is the further readings. This takes you to where you can read more on the knowledge or topic presented in the unit. The modules and units are presented as follows:

Module 1: Cryptanalysis and Shannon Theory
- Unit 1: Introduction to simple cryptosystem
- Unit 2: Types, Techniques and application of Cryptanalysis
- Unit 3: Probability theory
- Unit 4: Entropy
- Unit 5: Product Cryptosystem

Module 2: Block Cypher and Advanced Encryption Scheme
- Unit 1: Linear Cryptanalysis
- Unit 2: Differential Cryptanalysis
- Unit 3: Data and Advanced Encryption Standard

Module 3: Public Key Cryptography and Discrete Logarithm
- Unit 1: ElGamal Cryptosystem
- Unit 2: Algorithm for the discrete logarithm problem
- Unit 3: Elliptics Curves

Module 4: Private Key Encryption
- Unit 1: Symetric Encryption Scheme
- Unit 2: Issues in Privacy

There are thirteen units in this course. Each unit represent a week of study.

Presentation Schedule

The weekly activities are presented in Table 1 while the required hours of study and the activities are presented in Table 2. This will guide your study time. You may spend more time in completing each module or unit.

Table I: Weekly Activities

Week	**Activity**
1	Orientation and course guide
2	Module 1 Unit 1
3	Module 1 Unit 2
4	Module 1 Unit 3
5	Module 1 Unit 4
6	Module 1 Unit 5
7	Module 2 Unit 1

8	Module 2 Unit 2
9	Module 2 Unit 3
10	Module 3 Unit 1
11	Module 3 Unit 2
12	Module 3 Unit 3
13	Module 4 Units 1 and 2
14	Revision and response to questionnaire
15	Examination

The activities in Table I include facilitation hours (synchronous and asynchronous), assignments, mini projects, and laboratory practical. How do you know the hours to spend on each? A guide is presented in Table 2.

Table 2: Required Minimum Hours of Study

S/N	Activity	Hour per Week	Hour per Semester
1	Synchronous Facilitation (Video Conferencing)	2	26
2	Asynchronous Facilitation (Read and respond to posts including facilitator's comment, self-study)	4	52
3	Assignments, mini-project, laboratory practical and portfolios	1	13
	Total	7	91

Assessment

Table 3 presents the mode you will be assessed.

Table 3: Assessment

S/N	**Method of Assessment**	**Score (%)**
1	Portfolios	10
2	Mini Projects with presentation	30
3	Assignments	20
4	Final Examination	40
Total		100

Portfolio

A portfolio has been created for you tagged "**My Portfolio**". With the use of Microsoft Word, state the knowledge you gained in every Module and in not more than three sentences explain how you were able to apply the knowledge to solve problems or challenges in your context or how you intend to apply the knowledge. Use this Table format:

Application of Knowledge Gained

Module	Topic	Knowledge Gained	Application of Knowledge Gained

You may be required to present your portfolio to a constituted panel.

Mini Projects with presentation

You are to work on the project according to specification. You may be required to defend your project. You will receive feedback on your project defence or after scoring. This project is different from your thesis.

Assignments

Take the assignment and click on the submission button to submit. The assignment will be scored, and you will receive a feedback.

Examination

Finally, the examination will help to test the cognitive domain. The test items will be mostly application, and evaluation test items that will lead to creation of new knowledge/idea.

How to get the Most from the Course

To get the most in this course, you:

- Need a personal laptop. The use of mobile phone only may not give you the desirable environment to work.
- Need regular and stable internet.
- Need to install the recommended software.
- Must work through the course step by step starting with the programme orientation.
- Must not plagiarise or impersonate. These are serious offences that could terminate your studentship. Plagiarism check will be used to run all your submissions.
- Must do all the assessments following given instructions.
- Must create time daily to attend to your study.

Facilitation

There will be two forms of facilitation – synchronous and asynchronous. The synchronous will be held through video conferencing according to weekly schedule. During the synchronous facilitation:

- There will be two hours of online real time contact per week making a total of 26 hours for thirteen weeks of study time.
- At the end of each video conferencing, the video will be uploaded for view at your pace.
- You are to read the course material and do other assignments as may be given before video conferencing time.
- The facilitator will concentrate on main themes.
- The facilitator will take you through the course guide in the first lecture at the start date of facilitation

For the asynchronous facilitation, your facilitator will:

- Present the theme for the week.
- Direct and summarise forum discussions.
- Coordinate activities in the platform.
- Score and grade activities when need be.
- Support you to learn. In this regard personal mails may be sent.
- Send you videos and audio lectures, and podcasts if need be.

Read all the comments and notes of your facilitator especially on your assignments, participate in forum discussions. This will give you opportunity to socialise with others in the course and build your skill for teamwork. You can raise any challenge encountered during your study. To gain the maximum benefit from course facilitation, prepare a list of questions before the synchronous session. You will learn a lot from participating actively in the discussions.

Finally, respond to the questionnaire. This will help ACETEL to know your areas of challenges and how to improve on them for the review of the course materials and lectures.

Learner Support

You will receive the following support:

- Technical Support: There will be contact number(s), email address and chatbot on the Learning Management System where you can chat or send message to get assistance and guidance any time during the course.

- 24/7 communication: You can send personal mail to your facilitator and the centre at any time of the day. You will receive answer to you mails within 24 hours. There is also opportunity for personal or group chats at any time of the day with those that are online.

- You will receive guidance and feedback on your assessments, academic progress, and receive help to resolve challenges facing your stuides.

Course Information

Course Code:	CST 803
Course Title:	Advanced Cryptography
Credit Unit:	3
Course Status:	Compulsory
Course Blurb:	This covers intelligence foundation, lifecycle, attack, defence and tools; cyber threat intelligence landscape including tactical, operational and strategic dimensions and threat intelligence maturity model. It includes techniques gathering intelligence, counterintelligence methods and attribution.
Semester:	Second
Course Duration:	13 weeks
Required Hours for Study:	65

Course Team

Course Developer:	ACETEL
Course Writer:	J. A. Ojeniyi (PhD) and O. B. Longe (PhD)
Content Editor:	Ismaila Idris (PhD)
Instructional Designers:	Inegbedion, Juliet O. (Ph.D.) and Dr. Lukuman Bello
Learning Technologists:	Dr. Adewale Adesina & Mr. Miracle David
Graphic Artist:	Mr. Henry Udeh
Proofreader:	Mr Awe Olaniyan Joseph

Module 1: Cryptanalysis and Shannon Theory

Unit 1: Introduction to a Simple Cryptosystem

Contents

1.0 Introduction

Security of Information and Communication Technology has become major issues of concern in every discourse of Information Technology deployment. The privacy and security of data at rest and data in motion against malicious activities is the focus of this unit. In particular, I will take you through acquiring the requisite knowledge and skills to provide the confidentiality of your message over an insecure channel.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain the fundamental goals of cryptography
- elaborate the conditions for a tuple-based mathematical definition of a cryptosystem
- encrypt a message using a given cryptographic scheme
- decrypt a given ciphertext using an appropriate key.

3.0 Main Content

3.1 Fundamental Goals of Cryptography

Cryptography is the study of mathematical techniques for ensuring the security of information and communication over an insecure channel in a way that the un-intended person in the middle cannot understand the sent information. The channel could be a telephone line, wired or wireless network medium. The four fundamental goals of cryptography in securing information over a communication medium are (1) confidentiality, (2) data integrity, (3) authentication and (4) non-repudiation.

(1) Confidentiality: This is to ensure that the content of information is intelligently readable to only the authorized user while it is unreadable to unintended users. It maintains secrecy of the meaning of the information.

(2) Data integrity: This is a mechanism to check unauthorized manipulation of data such as insertion, deletion and substitution.

(3) Authentication: This is a service that ensures that the communicating parties can identify each other.

(4) Non-repudiation: This is a mechanism that prevents an entity in a communication scenario from denying previous actions. It is a means of resolving communication disputes involving denial of prior actions taken.

3.2 Illustration and Mathematical Notation of a Simple Cryptosystem

A cryptosystem is a suite of cryptographic algorithms or schemes to implement a particular fundamental information security service, as mentioned in section 3.1 or any other derived security service. Basically, a cryptosystem consists of three algorithms: (1) key generation algorithm, (2) encryption algorithm and (3) decryption algorithm.

An illustration of communication between two people over an insecure channel is given in figure 1 below to explain the components of a cryptosystem:

***General Scenario**: The basic idea about cryptography is keeping your information secret. Information such as your debit card pin, computer password, mobile phone password and even email password. The goal is secrecy. The closest example of a cryptographic system is our email*

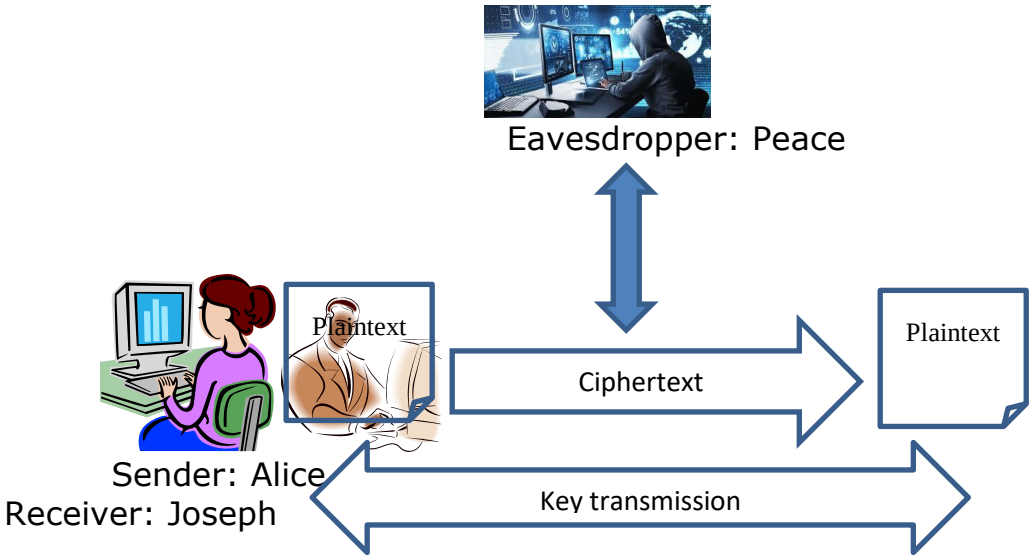


Fig. 1: A Simple Cryptosystem

As indicated in figure 1, the two people communicating are Alice and Joseph with an eavesdropper named Peace, who is interested in illegally understanding the message. The message Alice is sending is referred to as 'plaintext'. She encrypts this message using a key before sending it over the insecure channel. The encrypted message is called ciphertext. The process of converting plaintext into ciphertext using a key is called encryption. This ciphertext consists of scrambled text which is un-intelligent to Peace even if eavesdropped. At the receiver's end, Joseph uses the key to decrypt the ciphertext back into plaintext. On the other hand, the process of converting ciphertext back into plaintext using a key is referred to as decryption. Alice and Joseph communicate the key over a secured medium.

The cryptosystem ideas can be defined mathematically as:
A cryptosystem is a five-tuple $(P, C, K, \mathcal{E}, D)$, provided the following conditions are satisfied:

(1) P is a finite set of possible plaintexts;
(2) C is a finite set of possible ciphertexts;
(3) K, the keyspace, is a finite set of possible keys;
(4) For every $k \in K$, there exists an encryption rule $e_k \in \mathcal{E}$ and a corresponding decryption rule $d_k \in D$. Each $e_k: P \rightarrow C$ and $d_k: C \rightarrow P$ are functions such that $d_k(e_k(x)) = x$ for every plaintext element $x \in P$.

3.3 Types of Cryptosystem

Considering the simple and classical cryptography, there are different major types of cryptosystems. Some of the identified types are: Monoalphabetic, polyalphabetic, permutation and stream cryptosystems.

3.3.1 Monoalphabetic Cryptosystems

This is a cryptosystem in which once a key is chosen, each alphabetic character in the plaintext is mapped to a unique alphabetic character as ciphertext. Examples of monoalphabetic cryptosystems are Shift cipher, substitution cipher and affine cipher.

(1) Shift cipher

This is a monoalphabetic cryptosystem that is based on modular arithmetic. Modular arithmetic is an arithmetic operation on integers in which a number or numbers are replaced by their remainders after dividing them with a fixed number.

Solved Example 1: Compute 25 mod 4 pronounced 25 moduli 4
Solution:
You write $25 = 6 \times 4 + 1$,
Mathematically,
25 is the dividend
6 is the quotient
4 is the divisor
1 is the remainder,
Now, since $0 \leq 1 < 4$, then
$25 \bmod 4 = 1$

Solved Example 2: Compute 17 mod 5
Solution:
$17 = 3 \times 5 + 2$
Therefore, $17 \bmod 5 = 2$

Watch the following videos:

Video 1: What is Modular Arithmetic - Introduction to Modular Arithmetic – Cryptography - https://www.youtube.com/watch?v= Eg6CTCu8iio/

Video 2: How to Convert a Positive Integer in Modular Arithmetic – Cryptography - https://www.youtube.com/watch?v=LcUKSu-1Adw,

Video 3: How to Convert a Negative Integer in Modular Arithmetic – Cryptography - https://www.youtube.com/watch?v=2rbeCUMBYgk/

Shift Cipher Cryptosystem
As the name implies, shift cipher cryptosystem is a system of encryption or decryption based on modular shift (addition or subtraction) in the encryption or decryption function.

In mathematical terms, shift cipher cryptosystem can be defined as follows:

If given plaintext, P, ciphertext, C and key, K to be integer modulus 26 (i.e. $\mathbb{Z}_{26}$), for every $0 \leq k \leq 25$, then
$e_k(x) = (x + k)$ mod 26 and
$d_k(y) = (y - k)$ mod 26
for all $x, y \in \mathbb{Z}_{26}$

In order to encrypt using shift cipher, the fundamental principle to follow is explained thus:
If given P be a set of simple lower case English text or alphabetic characters as the plaintext, you can encrypt P by setting up a correspondence between a set of simple upper case English alphabetic characters and modulus 26 since there are 26 texts in English alphabets. This correspondence is to generate the ciphertext denoted in upper case or capital letters.

The correspondence is set up as follows:

A ↔ 0, B ↔ 1, C ↔ 2, …, Z ↔ 25,
This implies that,

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*

N	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*
13	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*

The table of correspondence given above will be used for encryption in several cryptosystem examples.

The computations of the key or set of keys, K is also in modulus 26. For instance,
if K = 9, then 9 mod 26 = 9,
if K = 30, then 30 mod 26 = 4,
if K = 27, then 27 mod 26 = 1

Solved Example 3: Given the plaintext: nounacetelteamisworking and K = 14, encrypt the plaintext using shift cipher cryptosystem.

Solution:

First, you convert the plaintext to a sequence of integers using the defined 26 integer modulo correspondence as follows,

N	*O*	*U*	*n*	*a*	*c*	*e*	*t*	*e*	*l*	*t*	*e*	*a*	*m*	*i*	*s*	*w*	*o*	*r*	*k*	*i*	*n*	*g*
13	*14*	*20*	*13*	*0*	*2*	*4*	*19*	*4*	*11*	*19*	*4*	*0*	*12*	*8*	*18*	*22*	*14*	*17*	*10*	*8*	*13*	*6*

Then, you ensure that the key value is in modulo 26, i.e. k = 14 mod 26 = 14,

Now, you add 14 to each value

N	*o*	*U*	*n*	*A*	*c*	*e*	*t*	*e*	*l*	*t*	*e*	*a*	*m*	*i*	*s*	*w*	*o*	*r*	*k*	*i*	*n*	*g*
27	*28*	*34*	*27*	*14*	*16*	*18*	*33*	*18*	*25*	*33*	*18*	*14*	*26*	*22*	*32*	*36*	*28*	*31*	*24*	*22*	*27*	*20*

The next step is to ensure that the integer values are in the required modulus, which is 26. This is given as follows,

N	*o*	*U*	*n*	*A*	*c*	*e*	*t*	*e*	*l*	*t*	*e*	*a*	*m*	*i*	*s*	*w*	*o*	*r*	*k*	*i*	*n*	*g*
1	*2*	*8*	*1*	*14*	*16*	*18*	*7*	*18*	*25*	*7*	*18*	*14*	*0*	*22*	*6*	*10*	*2*	*5*	*24*	*22*	*1*	*20*

Finally, you encrypt the plaintext into ciphertext by converting the modulo-based sequence of integer values to upper case alphabetic characters using the table of correspondence as follows:

N	*o*	*U*	*n*	*a*	*c*	*e*	*t*	*e*	*l*	*t*	*e*	*a*	*M*	*i*	*s*	*w*	*o*	*r*	*k*	*i*	*n*	*g*
1	*2*	*8*	*1*	*14*	*16*	*18*	*7*	*18*	*25*	*7*	*18*	*14*	*0*	*22*	*6*	*10*	*2*	*5*	*24*	*22*	*1*	*20*
B	*C*	*I*	*B*	*O*	*Q*	*S*	*H*	*S*	*Z*	*H*	*S*	*O*	*A*	*W*	*G*	*K*	*C*	*F*	*Y*	*W*	*B*	*U*

Therefore, the required ciphertext is:

BCIBOQSHSZHSOAWGKCFYWBU

Solved Example 4: Given the plaintext: open mode and K = 28, encrypt the plaintext using shift cipher cryptosystem.

First, you convert the plaintext to a sequence of integers using the defined 26 integer modulo correspondence as follows,

o	*p*	*E*	*n*	*m*	*O*	*d*	*E*

14	15	4	13	12	14	3	4

Then, convert the key into modulo 26, i.e. k = 28 mod 26 = 28 – 26 = 2, so the key is 2 in modulus 26,

Now, you add 2 to each value

o	p	E	n	m	O	d	E
16	17	6	15	14	16	5	6

Next, convert the integer values to mod 26. This is given as follows,

o	p	E	n	m	O	d	E
16	17	6	15	14	16	5	6

Finally, you encrypt the plaintext into ciphertext by converting the modulo-based sequence of integer values to upper case alphabetic characters using the specified correspondence as follows:

o	p	E	n	m	o	d	E
16	17	6	15	14	16	5	6
Q	R	G	P	O	Q	F	G

Therefore, the required ciphertext is:

QRGPOQFG

(2) Substitution cipher

A substitution cipher is another example of a monoalphabetic cryptosystem. Like shift cipher, you can use 26-lettered English alphabets for the plaintext, P and ciphertext, C. But unlike in shift cipher where encryption/decryption is treated as algebraic operation, it is more of permutation of algebraic characters in the substitution cipher.
Mathematically, substitution cipher cryptosystem can be defined as follows:

> If given plaintext, P and ciphertext, C in integer modulus 26 (i.e. $\mathbb{Z}_{26}$) and key K consist of all possible permutations of the 26 symbols 0, 1, 2, …, 25.For every permutation $\phi \in K$, then the encryption function is defined as,
> $e_\phi(x) = \phi(x)$ and decryption function as $d_\phi(y) = \phi^{-1}(y)$
> where$\phi^{-1}$is the inverse permutation to ϕ.

Solved Example 5: Using substitution cipher, encrypt the plaintext 'openmode' given the encryption function, $e_\phi(x)$, in the table below (as

used conventionally, plaintext characters are written in lower case while ciphertext characters are written in upper case):

X	a	B	c	D	e	f	g	H	i	j	k	l	m
$e_\phi(x)$	N	O	P	Q	R	S	A	U	V	W	X	Y	D

N	o	P	q	r	S	t	U	v	w	x	y	z
T	B	C	Z	E	F	G	H	I	J	K	L	M

Solution:

Step 1: Use the encryption function to convert the given plaintext 'openmode' to ciphertext as follows:

$e_\phi(o)=B$, $e_\phi(p)=C$, $e_\phi(e)=R$ *and so on*

The complete encryption result is given in the table below:

x	O	p	e	N	m	o	d	e
$e_\phi(x)$	B	C	R	T	D	B	Q	R

Step 2: Extract the ciphertext as BCRTDBQR

[**Hint:** In order to decrypt the ciphertext, we use the reverse of the encryption table by putting the upper case ciphertext characters as the first row and lower case plaintext characters as the second row and carry out the corresponding substitution]

(3) Affine cipher

By comparing the first two monoalphabetic ciphers, you will discover that shift cipher is a special case of substitution cipher. The number of possible permutations of 26 elements or characters in substitution cipher is 26!, pronounced 26 – factorial (i.e. 26 x 25 x 24 x … x 1 ways). On the other hands, only 26 permutations are possible in shift cipher. Now, affine cipher is another special case of substitution cipher in which the encryption functions take the form of:

$e(x) = (ax + b) \bmod 26$,

$a, b \in \mathbb{Z}_{26}$

Note that if $a = 1$ in the affine encryption, it will become,

$e(x) = (x + b) \bmod 26$, which is equivalent to the shift cipher encryption function

The mathematical definition of affine cipher is given as follows:

Given that $P = C = \mathbb{Z}_{26}$ *and assume that* $K = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26}: \gcd(a, 26) = 1\}$. *For the key* $k = (a, b) \in K$, *then the encryption function is,*

$e_k(x) = (ax + b) \bmod 26$ *and*

decryption function is

$d_k(y) = a^{-1}(y - b) \bmod 26$

$(x, y \in \mathbb{Z}_{26})$

Solved Example 6:
If the key, k = (7, 3) and the plaintext is 'noun', use affine cipher to encrypt the given plaintext.

Solution:
Step 1: First compute the encryption function
From the mathematical definition of affine cipher, the encryption function is,
$e_k(x) = (ax + b) \bmod 26$, given $k = (7, 3)$ which implies $a = 7$ and $b = 3$
Therefore, $e_k(x) = (7x + 3) \bmod 26$

Step 2: Convert the letters n, o, u, n to residues modulo 26 as follows:

n	*O*	*u*	*N*
13	*14*	*20*	*13*

Step 3: Substitute integer modulo values of plaintext characters into the encryption function as follows:
n: $(7 \times 13 + 3) \bmod 26 = 94 \bmod 26 = 16$
o: $(7 \times 14 + 3) \bmod 26 = 101 \bmod 26 = 23$
u: $(7 \times 20 + 3) \bmod 26 = 143 \bmod 26 = 13$
n: $(7 \times 13 + 3) \bmod 26 = 94 \bmod 26 = 16$

Step 4: Convert the encrypted modulo values into the corresponding ciphertext characters.
The corresponding ciphertext characters for 16, 23, 13, 16 are Q, X, N and Q. Therefore, the string of the ciphertext is QXNQ.

3.3.2 Polyalphabetic Cryptosystems

In section 3.3.1, I introduced you to monoalphabetic cryptosystems in which once a key is chosen, each alphabetic character in the plaintext is mapped to a unique alphabetic character in the ciphertext. In contrast, a polyalphabetic cryptosystem is a system of encryption in which each alphabetic characters in the plaintext can be mapped to different alphabetic characters in the ciphertext and vice versa.

(1) Vigenere cipher
This is a polyalphabetic cryptosystem in which each key *k* can be associated with an alphabetic string of length *m* called a keyword. Then, *m* alphabetic characters are encrypted at a time. Therefore, each plaintext element is equivalent to *m* alphabetic characters.

Mathematically, vigenere cipher can be expressed as: If given $P = C = K = \mathbb{Z}_{26}$ and for a key $k = (k_1, k_2, \ldots, k_m)$ where *m* is a positive integer, the encryption function is defined as
$e_k(x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$
and decryption function as

$d_k(y_1, y_2, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m)$,
where all operations are in $\mathbb{Z}_{26}$.

Solved Example 7: Given m = 4, the keyword is 'NOUN', and the plaintext is the string 'acetelproject'. Encrypt the plaintext using Vigenere cipher.

Solution:
Step 1: Convert the keyword to the numerical equivalent in modulus 26 (i.e., using the table of correspondence in mod 26)

N	*O*	*U*	*N*	*← this is the keyword*
13	*14*	*20*	*13*	*← numerical equivalents*
13	*14*	*20*	*13*	*← modulo 26*

This implies that k = (13, 14, 20, 13)
Step 2: Convert the plaintext elements to residues modulo 26 as follows:

A	*c*	*E*	*t*	*e*	*l*	*p*	*r*	*o*	*J*	*e*	*c*	*t*
0	*2*	*4*	*19*	*4*	*11*	*15*	*17*	*14*	*9*	*4*	*2*	*19*

Step 3: Write the plaintext modulo values in groups of m which is 4 and add k (13, 14, 20, 13) to each group

A	*c*	*E*	*t*	*e*	*l*	*p*	*r*	*o*	*J*	*e*	*c*	*t*
0	*2*	*4*	*19*	*4*	*11*	*15*	*17*	*14*	*9*	*4*	*2*	*19*
13	*14*	*20*	*13*	*13*	*14*	*20*	*13*	*13*	*14*	*20*	*13*	*13*
13	*16*	*24*	*32*	*17*	*25*	*35*	*30*	*27*	*23*	*24*	*15*	*32*

Step 4: Convert the results of the addition into modulo 26

13	*16*	*24*	*32*	*17*	*25*	*35*	*30*	*27*	*23*	*24*	*15*	*32*
13	*16*	*24*	*6*	*17*	*25*	*9*	*4*	*1*	*23*	*24*	*15*	*6*

Step 5: Then, convert the modulo-based sum into the alphabetic equivalent to form the ciphertext string

13	*16*	*24*	*6*	*17*	*25*	*9*	*4*	*1*	*23*	*24*	*15*	*6*
N	*Q*	*Y*	*G*	*R*	*Z*	*J*	*E*	*B*	*X*	*Y*	*P*	*G*

Therefore, the ciphertext string is: NQYGRZJEBXYPG

From all the types of cryptosystems, as you can encrypt a plaintext using an encryption function, you can equally decrypt the ciphertext using decryption function. Decryption function is always the inverse of encryption function as seen in all the mathematical definitions of each cryptosystem. All the operations in the decryption process are the opposite of the operations in the corresponding encryption process.

(2) Hill cipher

Hill cipher is another polyalphabetic cryptosystem in which m linear combinations of the m alphabetic characters in one plaintext element produces m alphabetic characters in one ciphertext element.

For example, if $m = 3$, you can write a plaintext element as $x = (x_1, x_2, x_3)$ and a ciphertext element as $y = (y_1, y_2, y_3)$, then each of y_1, y_2, y_3 is a linear combination of x_1, x_2, x_3 as follow:

$y_1 = ax_1 + b\,x_2 + c\,x_3$

$y_2 = dx_1 + e\,x_2 + f\,x_3$

$y_3 = gx_1 + h\,x_2 + i\,x_3$

This can be expressed as:

$$(y_1, y_2, y_3) = (x_1, x_2, x_3)\begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix}$$

Where all operations are performed in $\mathbb{Z}_{26}$,

The key k = $\begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix}$

Using matrix notation, the whole equation can be reduced to:

$y = xk$

Where x is the plaintext, y is the ciphertext and k is the key

Formally, the encryption function of the Hill cipher can be defined as,

$e_k(x) = xk$, for a key k and

decryption function defined as,

$d_k(y) = yk^{-1}$ *with all operations in* $\mathbb{Z}_{26}$,

given that an integer $m \geq 2$, $P = C = (\mathbb{Z}_{26})^m$ *and K = {m x m invertible matrices over* $\mathbb{Z}_{26}$*)*

3.3.3 Permutation Cryptosystem

All the cryptosystems discussed so far both monoalphabetic and polyalphabetic involve substituting plaintext characters with different ciphertext characters. But, a permutation cipher is meant to keep the characters unsubstituted. The only aspects being altered is the character position by rearranging them by the principle called a permutation. By this, a permutation cipher is also referred to as Transposition cipher.

Mathematically, transposition or permutation cipher can be defined as follows:

Given m is a positive integer, P = C = $(\mathbb{Z}_{26})^m$ and K consists all permutations of {1, …, m}. For every key k (i.e. a permutation) ϕ, an encryption function can be defined as,

$$e_\phi(x_1, \ldots, x_m) = (x_{\phi(1)}, \ldots, x_{\phi(m)})$$

and decryption function as,

$$d_\phi(y_1,\ldots,y_m) = (y_{\phi^{-1}(1)},\ldots,y_{\phi^{-1}(m)}),$$

where ϕ^{-1} is the inverse permutation to ϕ.

Solved Example 8: Given the key used to encrypt a ciphertext 'PNOEOEMD' in the table below. Compute the decryption key and use it to decrypt the given ciphertext given that m=4..

The encryption key is:

x	1	2	3	4
$\phi(x)$	2	4	1	3

Solution:
Step 1: Compute the inverse permutation $\phi^{-1}(x)$ by interchanging the two rows

$\phi(x)$	2	4	1	3
x	1	2	3	4

Step 2: Re-arrange the columns so that the first row is in ascending order. This gives you the key

y	1	2	3	4
$\phi^{-1}(y)$	3	1	4	2

Step 3: Partition the ciphertext into groups of four letters:

P	N	O	E	O	E	M	D
1	2	3	4	1	2	3	4

Step 4: Re-arrange each group according to the inverse permutation $\phi^{-1}(x)$

P	N	O	E	O	E	M	D
1	2	3	4	1	2	3	4
3	1	4	2	3	1	4	2
o	P	e	n	m	o	d	e

Therefore, the plaintext is: 'openmode'

3.3.4 Stream cryptosystem

All the cryptosystems discussed in the previous sections use the same key, k to encrypt successive plaintext elements. These types of cryptosystems are called **block ciphers**. Another cryptosystem that does not use the same key is a **stream cipher**. In a stream cipher, a keystream is generated and used to encrypt the plaintext.

One of the simplified mathematical expressions for defining stream cipher is given as follows:
If $P = C = K = L = \mathbb{Z}_{26}$ where L is a finite set called the keystream alphabet, given that $z_1 = k$ and $z_i = x_{i-1}$ for all $i \geq 2$. *For* $0 \leq z \leq 25$, the encryption function is,

$e_z(x) = (x + z) \bmod 26$

and the decryption function is,

$d_z(y) = (y - z) \bmod 26$
$(x, y \in \mathbb{Z}_{26})$

Solved Example 9: Encrypt a plaintext 'excellence' with a key k = 5 using stream cipher scheme.

Solution:
Step 1: Convert the plaintext to a sequence of integers:

e	*x*	*C*	*e*	*l*	*l*	*e*	*n*	*c*	*E*
4	*23*	*2*	*4*	*11*	*11*	*4*	*13*	*2*	*4*

Step 2: Generate the keystream
To generate the keystream, you should take note of some mathematical expressions in the stream cipher mathematical definition.
First of all, $z_1 = k$ implies $z_1 = 5$ (i.e. the first value, z_1 in the keystream is the key 5)
Then other values in the keystream are generated by $z_i = x_{i-1}$ starting from *i = 2, this gives the values of the plaintext.*
So, the keystream is generated as follows:

5	*4*	*23*	*2*	*4*	*11*	*11*	*4*	*13*	*2*

Step 3: Add the corresponding plaintext elements and the keystream elements

4	*23*	*2*	*4*	*11*	*11*	*4*	*13*	*2*	*4*
5	*4*	*23*	*2*	*4*	*11*	*11*	*4*	*13*	*2*
9	*27*	*25*	*6*	*15*	*22*	*15*	*17*	*15*	*6*

Step 4: Reduce the values to modulo 26

9	*27*	*25*	*6*	*15*	*22*	*15*	*17*	*15*	*6*
9	*1*	*25*	*6*	*15*	*22*	*15*	*17*	*15*	*6*

Step 5: Convert to ciphertext

9	*1*	*25*	*6*	*15*	*22*	*15*	*17*	*15*	*6*
J	*B*	*Z*	*G*	*P*	*W*	*P*	*R*	*P*	*G*

Therefore, the ciphertext is: JBZGPWPRPG

Assignment 1:
Discuss the strength and weaknesses of the various cryptosystem talked about in this unit. Tyye your answer in Microsoft Word and clcik on the submit button to submit your answer.

4.0 Self-Assessment Exercise(s)

1. Decrypt the ciphertext (NQYGRZJEBXYPG) in Solved Example 7 using the keyword 'NOUN' with m = 4.

Answer:

Step 1: Convert the keyword to the numerical equivalent in modolus 26

N	*O*	*U*	*N*	*← this is the keyword*
13	*14*	*20*	*13*	*← numerical equivalents*
13	*14*	*20*	*13*	*← modulo 26*

This implies that k = (13, 14, 20, 13)

Step 2: Convert the ciphertext elements to residues modulo 26 as follows:

N	*Q*	*Y*	*G*	*R*	*Z*	*J*	*E*	*B*	*X*	*Y*	*P*	*G*
13	*16*	*24*	*6*	*17*	*25*	*9*	*4*	*1*	*23*	*24*	*15*	*6*

Step 3: Write the ciphertext modulo values in groups of m (i.e. 4) and subtract k (13, 14, 20, 13) from each group

N	*Q*	*Y*	*G*	*R*	*Z*	*J*	*E*	*B*	*X*	*Y*	*P*	*G*
13	*16*	*24*	*6*	*17*	*25*	*9*	*4*	*1*	*23*	*24*	*15*	*6*
13	*14*	*20*	*13*	*13*	*14*	*20*	*13*	*13*	*14*	*20*	*13*	*13*
0	*2*	*4*	*-7*	*4*	*11*	*-11*	*-9*	*-12*	*9*	*4*	*2*	*-7*

Step 4: Convert the results of the subtraction into modulo 26

0	*2*	*4*	*-7*	*4*	*11*	*-11*	*-9*	*-12*	*9*	*4*	*2*	*-7*
0	*2*	*4*	*	*4*	*11*	*15*	*	*	*9*	*4*	*2*	*

* there are negative integers and their conversion to modulo 26 is not straight forward

Step 5: If there is or are, negative integer values, then convert the negative integers into modulo 26

Note: The conversion of negative integers is different from the conversion of positive integers. You can use the formula given below for easy conversion:

n = qm + r

where n is the number to be converted

q is the quotient

m is the required mod

r is the remainder which will be the new number in the required mod

Since n and m are already known, you are to compute q and r. The guiding principle is that you must pick a value for q such that when you multiply it with the mod m, it will give the immediate negative value less

than the negative number n to be converted. Then, the value of r to be added to qm is the final value in the required mod.

For example, to convert – 7 into mod 26, it implies that,

n = qm + r, n, q, m, r ∈ ℤ
where n = - 7
m = 26
q = ?
r = ?
therefore,
- 7 = (?) x 26 + (?)
Now, choose the value of q such that you get a negative value immediately less than – 7,
The value of q = - 1, → - 1 x 26 = - 26
But if you choose q = - 2 for example, → - 2 x 26 = - 52
The two values – 26 and – 52 are less than – 7, but – 26 is closer to – 7 than the second number – 52. So, - 26 is the immediate less than – 7 that is obtainable.
Therefore,
- 7 = (- 1) x 26 + (?),
- 7 = -26 + (r), then r = 19
So, apply this principle to compute the modulo values for all the negative integers.

0	*2*	*4*	*-7*	*4*	*11*	*-11*	*-9*	*-12*	*9*	*4*	*2*	*-7*
0	*2*	*4*	*19*	*4*	*11*	*15*	*17*	*14*	*9*	*4*	*2*	*19*
a	*c*	*E*	*t*	*e*	*L*	*p*	*r*	*O*	*j*	*e*	*c*	*t*

Step 5: Finally, the complete modulo-based subtracted values into the alphabetic equivalent to form the plaintext. But, remember to use the lower case conventional letters for the plaintext as upper case letters are used for the ciphertext

0	*2*	*4*	*19*	*4*	*11*	*15*	*17*	*14*	*9*	*4*	*2*	*19*
a	*c*	*E*	*t*	*e*	*L*	*p*	*r*	*O*	*j*	*e*	*c*	*t*

Therefore, the plaintext string is: 'acetelproject'
If you remember, this is the original plaintext for Solved Example 7.

2. Encrypt the plaintext 'openmode' given m = 4 and the key is given by the following permutation, ϕ.

x	*1*	*2*	*3*	*4*
ϕ(x)	*2*	*4*	*1*	*3*

Answer:
Step 1: Partition the plaintext into groups of four letters:

o	*p*	*e*	*n*	*M*	*o*	*d*	*E*

1	2	3	4	1	2	3	4

Step 2: Re-arrange each group according to the permutation $\phi(x)$

x	o	p	e	N	m	o	D	E
x	1	2	3	4	1	2	3	4
$\phi(x)$	2	4	1	3	2	4	1	3
y	P	N	O	E	O	E	M	D

Therefore, the ciphertext is: PNOEOEMD

3. Encrypt the plaintext 'noun' using a key k = $\begin{pmatrix} 10 & 5 \\ 2 & 6 \end{pmatrix}$.

Answer:

Step 1: Determine the value of *m* and the number of elements in the plaintext to encrypt

Since the k is of order *2 by 2* (i.e. *2 rows by 2 columns*) then *m = 2* and also the number of elements to encrypt in the plaintext is 2

Step 2: Partition the plaintext into different elements for encryption

Therefore, the plaintext 'noun' is separated into two as: 'no' and 'un',

Step 3: Convert the plaintext elements into corresponding modulo 26

This gives (13, 14) for 'no' and (20, 13) for 'un'.

Step 4: Compute the ciphertext for the plaintext elements from the encryption function

Recall and use the equation, y = xk

where x is the plaintext, y is the ciphertext and k is the key

For (13, 14), it implies

$y_1 = (13, 14) \begin{pmatrix} 10 & 5 \\ 2 & 6 \end{pmatrix} = (13 \times 10 + 14 \times 12 \quad 13 \times 5 + 14 \times 6) = (298 \quad 149)$

$y_2 = (20, 13) \begin{pmatrix} 10 & 5 \\ 2 & 6 \end{pmatrix} = (20 \times 10 + 13 \times 2 \quad 20 \times 5 + 13 \times 6) = (226 \quad 178)$

Step 5: Convert the resultant values to modulo 26

$y_1 = (298 \quad 149) = (12 \quad 19)$

$y_2 = (226 \quad 178) = (18 \quad 22)$

Step 6: Map the modulo 26 values to ciphertext alphabetic characters

$y_1 = (12 \quad 19) = (M, T)$

$y_2 = (18 \quad 22) = (S, W)$

Step 7: Merge the elements to form ciphertext string

$y = y_1y_2$ = MTSW

therefore encryption of the plaintext 'noun' using the given key produces MTSW

Self-Assessment Exercise(s)

1. In your opinion, what do you think are the component of a cryptosystem?

Answer:

Basically, a cryptosystem was defined to be a five-tuple system consisting of plaintext P, ciphertext C, key K, encryption function E and decryption function D. You also learnt the major categorisation of cryptosystems to be: monoalphabetic, polyalphabetic, permutation and stream cryptosystems.

With various types and solved examples introduced to you in this unit, you have been equipped with the ability of encryption and decryption in cryptosystems. The next unit will teach you how to cryptanalyse some of the cryptosystems you have learnt to determine the secret key of the cryptosystem. It will also enable you to know the security strength of your cryptosystem.

5.0 Conclusion

This unit has given you a detailed exposition to the essence of cryptography and the underpinning mathematical principles of various cryptosystems. With this knowledge, you are better armed with the ability to juxtapose the similarities and differences of the types of cryptosystems. Based on your security goal, you will be able to determine the most suitable cryptosystem to adopt.

6.0 Summary

In this unit, you have learnt that cryptography is principally application of mathematical principles for information and communication security. And the four fundamental goals are confidentiality, data integrity, authentication and non-repudiation. Beyond this, you are now conversant with mathematical notations in defining the various types of cryptosystems.

7.0 References/Further Reading

https://www.merriam-webster.com/dictionary/**modular**%20**arithmetic**

https://www.oercommons.org/courses/yet-another-introductory-number-theory-textbook-cryptology-emphasis-version/view

Menezes, A. J.; Paul C. van Oorschot & Vanstone, Scott A. (1996). *Handbook of Applied Cryptography*. CRC Press.

Stinson, D. R. (2006). *Cryptography: Theory and Practice*. (3rd ed.). CRC Press.

William, Stallings Pearson (2016). *Cryptography and Network Security: Principles and Practice*. (7th ed.). CRC Press SBN-10: 0134444280.

Unit 2: Types, Techniques and Application of Cryptanalysis

Contents

1.0 Introduction

As the previous unit has explored the various types of cryptosystems, the determination of the strength of their security keys is equally very important. This unit will explain the types of cryptanalysis techniques and their applications. With the intended knowledge to be acquired in this unit, you will know the requisite approaches deployed by attackers against cryptosystems.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- juxtapose cryptanalysis in relations to the basis of operations
- compare and contrast the techniques of cryptanalysis
- apply cryptanalytic scheme for getting encryption key.

3.0 Main Content

3.1 Cryptanalysis and its basis of operation

Cryptanalysis is the art of breaking codes or coded information to get the key or secret information used for the encoding. It can also be defined as the study of ciphertext in order to convert it back to the plaintext. As cryptography and cryptosystem operations are based on strong mathematical principles, so also cryptanalysis techniques.

The activity of cryptanalyst is similar to the malicious activity of an un-intended user depicted as eavesdropper Peace in the illustrative diagram

of a cryptosystem in Unit 1. A diagrammatic illustration of cryptanalysis is given in Figure 1:

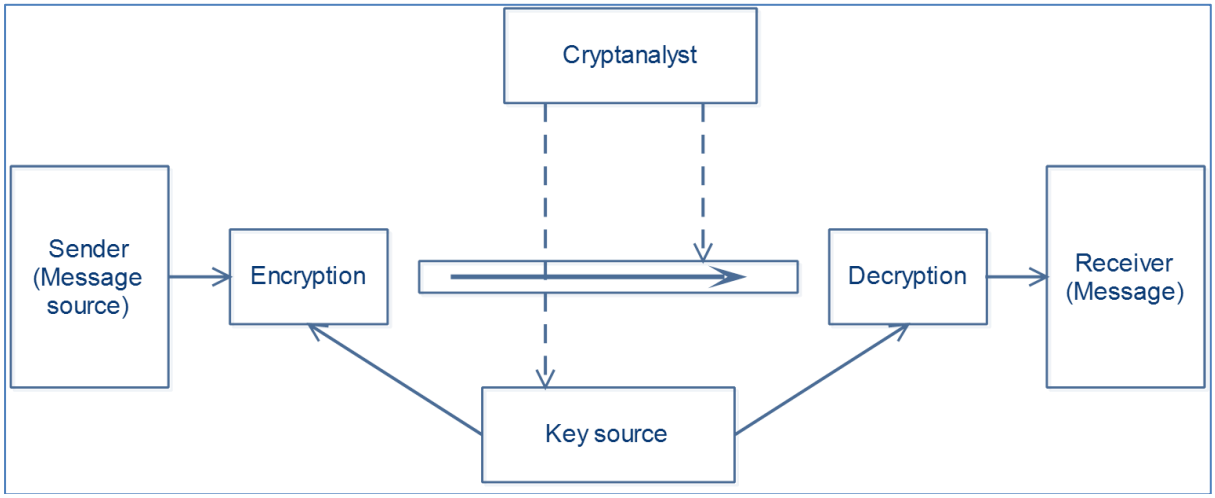


Fig. 2: Secrecy system for cryptanalysis

There is a fundamental basis guiding the operation of cryptanalysis. The three identified basis are:

(1) Amount of information available to an attacker

The target of an attacker is to get the key used for encrypting the plaintext into the ciphertext. So, the amount or quantity of useful tips like plaintext and ciphertext available to the attacker will influence the success of cryptanalysis.

(2) Computational resource required in terms of time, memory and data

The time resource is the number of computational steps the attacker must perform for the cryptanalysis like encryption and decryption algorithmic steps. The number of system storage location required for the cryptanalysis is the resource. Data computational resource is all about the required quantity of plaintext or ciphertext needed for performing the cryptanalysis.

(3) Types or classes of ciphers

Another very critical basic consideration for cryptanalysis in the category of the cipher. There are classical ciphers as treated in Unit 1 and there several other modern ciphers with more complex computations. So, the complexity of cryptanalysis operations also depends on the types or classes of cipher itself.

3.2 Types and Techniques of Cryptanalysis

The activities of cryptanalyst or attacker are categorised into types that are equivalent to their techniques of operations. Some of the most commonly identified types and techniques are:

(1) Ciphertext Only Attack (COA)
In this type of attack, the attacker can access only ciphertext or encrypted data but cannot access the plaintext.
This is depicted in Figure 2:

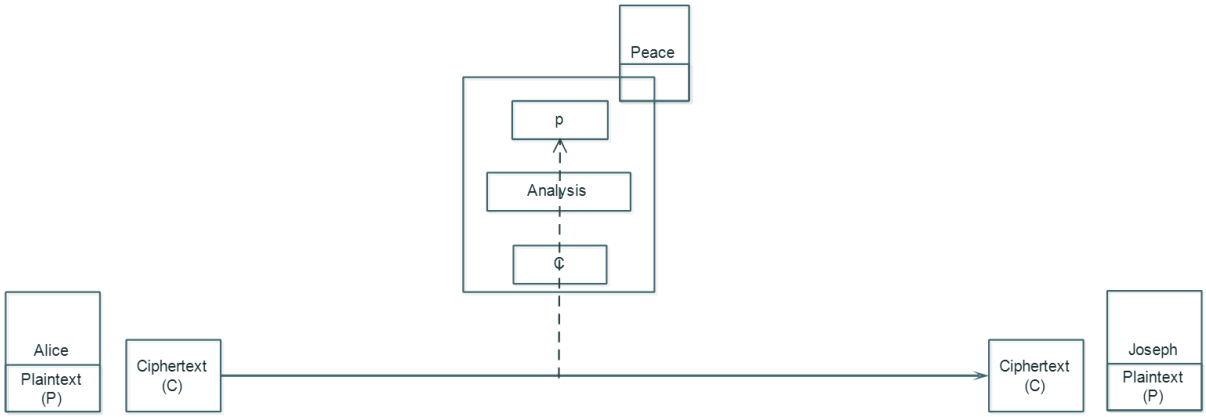


Fig. 3: Diagram for Ciphertext-Only Attack illustration

There are three basic methods to carry out a ciphertext-only attack. They are brute force attack, statistical attack and dictionary attack. Brute force attack is the process of guessing and trial of every possible key in the keyspace. The statistical attack is the guessing of the secret key based on the model of the language being used about the distribution of letters, probabilities or frequency of occurrence, digrams (pairs of letters), trigrams (triplets of letters). This will be discussed in details in the subsequent section under application of cryptanalysis. The last one is dictionary attack which is the process of building a table of possible plaintext messages and their corresponding ciphertext strings.

(2) Known Plaintext Attack (KPA)
This is a situation when the attacker possesses a string of the plaintext and its corresponding ciphertext. Then, the attacker tries to find out the relationship or mapping between them. The diagram depicting KPA is in Figure 3:

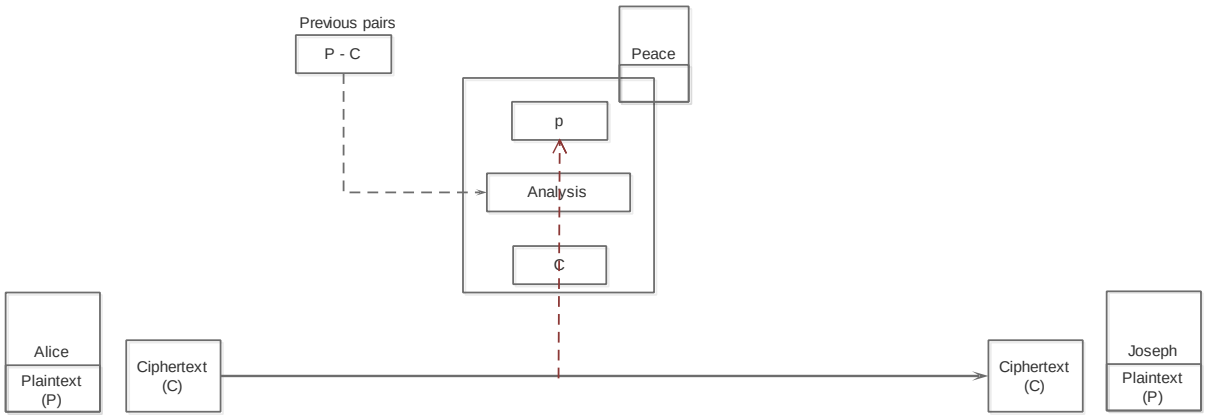


Fig. 4: Known Plaintext Attack Diagram

(3) Chosen Plaintext Attack (CPA)

This is a situation in which the attacker obtains a temporary or arbitrary access to the encryption system. Consequently, the attacker can choose a string of the plaintext and construct the corresponding string of the ciphertext.

The illustration for CPA is as in Figure 4:

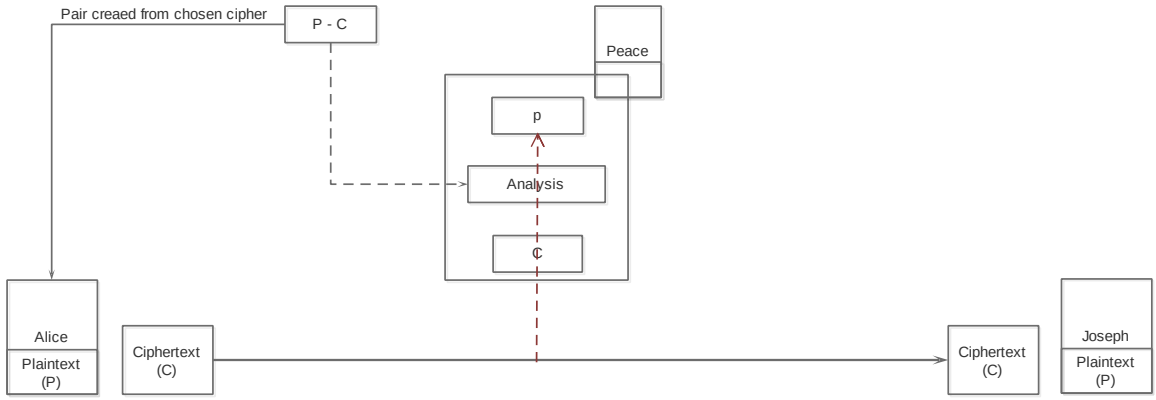


Fig. 5: Chosen Plaintext Attack Diagram

(4) Chosen Ciphertext Attack (CCA)

Similarly, the attacker has secured arbitrary or temporary access to the decryption machine. He or she can subsequently choose a ciphertext string and construct a corresponding string of the plaintext. What is the scale of the required information? The diagram in Figure 5 explains CCA the more.

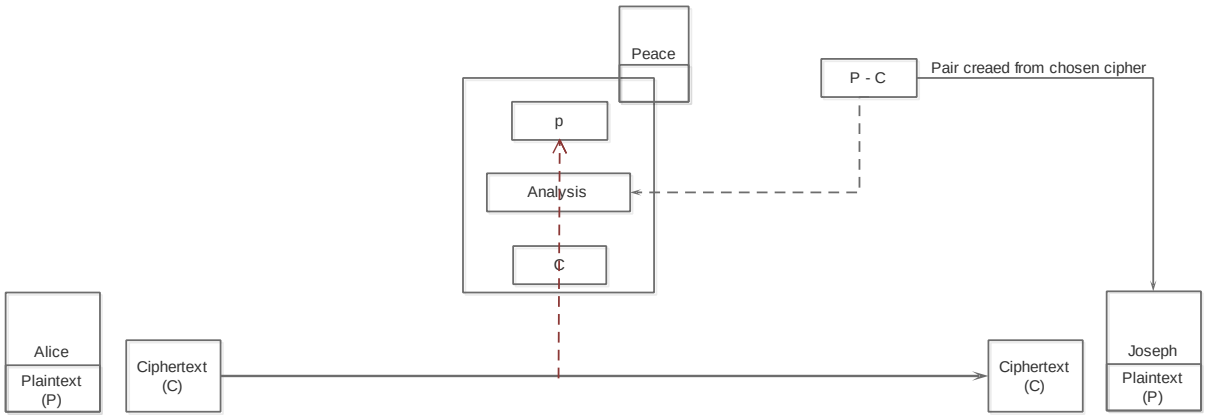


Fig. 6: Chosen Ciphertext Attack

(5) Adaptive Chosen Plaintext (ACP)
The attacker obtains temporary access to the encryption system, chooses a plaintext string and constructs the corresponding ciphertext string as in the chosen-plaintext attack. But, also, the attacker chooses a subsequent set of plaintext based on the information from the previous encryption methods.

(6) Adaptive Chosen Ciphertext (ACC)
In addition to the processes taken in the Chosen ciphertext attack, the attacker chooses the subsequent set of ciphertext using the previous information obtained from the decryption mechanism.

(7) Related Key Attack (RKA)
This is an attack similar to the chosen plaintext. The attacker can obtain only ciphertext encrypted with the help of the relationship between two unknown known keys.

The objective of the adversary in all the cases mentioned above is to determine the key that was used to encrypt the ciphertext. Once this is done, the attacker has access to decrypt a specific or targeted ciphertext string and any other ciphertext strings encrypted with the same key. A summary of the comparison of these cryptanalytic attacks is presented in Table 1.

Table 1: Comparison of the cryptanalysis attack models

S/N	Attack Type	Information known to the cryptanalyst
1	Ciphertext Only Attack (COA)	(i) Encryption algorithm (ii) Ciphertext to be decrypted
2	Known Plaintext Attack (KPA)	(i) Encryption algorithm (ii) Ciphertext to be decrypted (iii) One or more plaintext – ciphertext pairs formed with the same key
3	Chosen Plaintext Attack (CPA)	(i) Encryption algorithm (ii) Ciphertext to be decrypted (iii) Arbitrary plaintext message chosen by cryptanalyst, with its corresponding ciphertext generated using the same key
4	Chosen Ciphertext Attack (CCA)	(i) Encryption algorithm (ii) Ciphertext to be decrypted (iii) Arbitrary ciphertext chosen by cryptanalyst with its corresponding decrypted plaintext generated with unknown key.
5	Adaptive Chosen Plaintext (ACP)	(i) Encryption algorithm (i) Ciphertext to be decrypted (ii) Arbitrary plaintext message chosen by cryptanalyst, with its corresponding ciphertext generated using the same key (iii) Chooses an addition set of plaintext from a previous encryption
6	Adaptive Chosen Ciphertext (ACC)	(i) Encryption algorithm (ii) Ciphertext to be decrypted (iii) Arbitrary ciphertext chosen by cryptanalyst with its corresponding decrypted plaintext generated with unknown key. (iv) Chooses subsequent set of ciphertext from a previous encryption
7	Related Key Attack (RKA)	(i) Encryption algorithm (ii) Ciphertext to be decrypted (iii) An arbitrary plaintext message is chosen by cryptanalyst, with its corresponding ciphertext generated using the relationship between the two unknown keys

3.3 Application of Cryptanalysis

There are two aspects to the application of cryptanalysis. The first one is software designs, applications and deployment that are vulnerable to these attacks and the second aspect is the application of cryptanalysis techniques in breaking the codes to obtain the secret key.

3.3.1 Vulnerable applications to cryptanalysis

Different application deployments are vulnerable to each of the four cryptanalytic attacks.

(1) Applications vulnerable to ciphertext only attack (COA)
Some of the applications are:

(i) Early versions of Microsoft's PPTP virtual private network software that used the same RC4 key for the sender and the receiver
(ii) Wired Equivalent Privacy (WEP). Wi-Fi first security protocol was prone to several attacks most among them was ciphertext only.
(iii) Some designs for modern cipher such as Akelarre.

(2) Applications prone to known-plaintext attack (KPA)
The deployments that are vulnerable to known-plaintext attack are:

(i) Classical ciphers are mostly prone to this attack, such as Caesar or substitution cipher.
(ii) ZIP files that are encrypted and archived are also prone to this attack. For example, the only thing the cryptanalyst needs is an encrypted ZIP and unencrypted file from the archive to form "known-plaintext".
(iii) With some publicly available software, the secret key can decrypt the whole archive.

(3) Applications vulnerable to chosen-plaintext attack (CPA)
The application areas that are vulnerable to this cryptanalytic attack are:

(i) Non-randomised (deterministic) public-key encryption algorithms. So, for the public-key definitions to be more secured against this type of attack, there is a need for probabilistic encryption or randomised encryption.
(ii) Conventional symmetric ciphers. It is when the key used for both encryption and decryption is the same. A Gardening technique was used to solve the problem of encrypted messages during World War II. This Gardening technique worked based on the principle of a chosen-plaintext attack.

(4) Applications prone to chosen-ciphertext attack (CCA)
The applications that vulnerable to CCA are as follow:

(i) The semantic security of the El Gamal cryptosystem can be broken by CCA.
(ii) The early versions of RSA padding used in the SSL protocol were prone to a more advanced adaptive chosen-ciphertext (ACC).

(iii) Self-synchronizing stream cipher
(iv) Tamper-resistant cryptographic smart cards design may be prone to an adversary that could launch a large number of chosen-ciphertext in order to recover the hidden key.

3.3.2 Applications of Cryptanalysis Attack Techniques

In order to apply or deploy the techniques of cryptanalysis, several approaches are used. One of the approaches is the use of statistical properties of the English language. There are estimated relative frequencies of the 26 English letters by various researchers. These frequencies denote the probabilities of occurrence of the 26 letters of English alphabet. The compilation of these relative frequencies was by robust statistics from numerous newspapers, magazines, novels and so on. One of such compiled frequencies is given in Table 2.

Table 2: Probabilities of occurrence of the 26-English letters

Letter	Probability	Letter	probability
A	0.082	N	0.067
B	0.015	O	0.075
C	0.028	P	0.019
D	0.043	Q	0.001
E	0.127	R	0.060
F	0.022	S	0.063
G	0.020	T	0.091
H	0.061	U	0.028
I	0.070	V	0.010
J	0.002	W	0.023
K	0.008	X	0.001
L	0.040	Y	0.020
M	0.024	Z	0.001

The 26 – English letters were partitioned into five groups on the basis of the probabilities of occurrence. They are:
(1) E, with a probability of about 0.120
(2) T, A, O, I, N, S, H, R, with a probability between 0.06 and 0.09
(3) D, L, with a probability of around 0.04
(4) C, U, M, W, F, G, Y, P, B, with a probability between 0.015 and 0.028
(5) V, K, J, X, Q, Z, with probability less than 0.01

Also, further consideration was made for the commonness of two or three consecutive letters, respectively referred to as digrams and trigrams.
The 30 most common digrams in decreasing order are:

TH, HE, IN, ER, AN, RE, ED, ON, ES, ST,
EN, AT, TO, NT, HA, ND, OU, EA, NG, AS,
OR, TI, IS, ET, IT, AR, TE, SE, HI, OF.

And the twelve most common trigrams are:

THE, ING, AND, HER, ERE, ENT,
THA, NTH, WAS, ETH, FOR, DTH

Solved Example 1: Given that a ciphertext string below was obtained from an Affine Cipher, cryptanalysis it using statistical analysis of the ciphertext characters:

FMXVEDKAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDKAPRKDLYE VLRHHRH

Solution:

Step 1: Develop the frequency analysis table for the ciphertext characters

Letter	Frequency	Letter	Frequency
A	2	N	1
B	1	O	1
C	0	P	2
D	7	Q	0
E	5	R	8
F	4	S	3
G	0	T	0
H	5	U	2
I	0	V	4
J	0	W	0
K	5	X	2
L	2	Y	1
M	2	Z	0

Note: there are 57 characters in the ciphertext string which is enough for cryptanalysis of Affine cipher using statistical analysis.

Step 2: Rate the ciphertext characters based on the occurrences from highest to lowest

The ciphertext characters with most frequencies or occurrences are rated as:

Character	Occurrence
R	8
D	7
E, H, K	5 each
F, S, V	4 each
and so on	

Step 3: Formulate hypothesis based on the frequencies of ciphertext characters in Step 2 against the table of probabilities of English language alphabets

Since the characters of the ciphertext are English, then you compare the table in Step 1 or Step 2 with Table 2 (Probabilities of occurrence of the 26-English letters)

Then, hypothesise the most common character in the ciphertext as the encryption of the most common character in the plaintext. Continue this to the next most common to form at least two instances of mapping

This implies that,

R is the encryption of *e*, that is,

(since *e* is the letter with the highest probability in the English alphabet)

Similarly,

D is the encryption of *t*

Converting to modulo, it implies that

R encrypting e means $e_k(4) = 17$

D encrypting t means $e_k(19) = 3$

Step 4: Use the mapping in Step 3 to formulate simultaneous equation using the Affine cipher encryption function

Since Affine cipher encryption equation is: $e_k(x) = ax + b$

where x is the plaintext character,

a and b are constants that form key k = (a, b)

Now, substituting the first mapping ($e_k(4) = 17$) in Step 3 into the encryption equation, *you will have:*

$4a + b = 17$,

Similarly, substituting the second mapping ($e_k(19) = 3$) into the Affine encryption function, you will obtain

$19a + b = 3$

These two gives you simultaneous equations:

$4a + b = 17$

$19a + b = 3$

Step 5: Solve the resultant simultaneous equations

Given the equations:

$4a + b = 17$

$19a + b = 3$

Use elimination method to solve them as:

$4a + b = 17$
$(-)19a + b = 3$
$-15a = 14$

$a = -14/15 = -14 \times 15^{-1}$,

Converting this to modulo 26 gives $a = 6$

and subsequently $b = 19$:

But, since the greatest common divisor between *an* (i.e. 6) and 26 is 2, which is greater than 1 then the hypothesis is incorrect.

NOTE: You can only have a unique solution for every ciphertext character y if and only if greatest common divisor gcd(a, 26) = 1. Else you will have more than one solutions.

Although we can use the key to cryptanalysis, it is most likely to get a plaintext string that is unreasonable.

So you continue the iteration of the mapping until you get a desired key or result.

With the mapping of R to e and K to t, and you follow through the steps, you will get the value of a = 3 and b = 5.

The gcd (3, 26) = 1, this key (3, 5) will give you unique decryption. So, it is acceptable.

Step 6: Use the acceptable key *(a, b)* to compute the decryption function m

Recall that $d_k(y) = a^{-1}(y - b) \bmod 26$
Therefore, $d_k(y) = 3^{-1}(y - 5) \bmod 26$
Computing the inverse, 3^{-1} in mod 26, gives,
$d_k(y) = 9y - 19$,
If you use this encryption function to decrypt each letter of the ciphertext, you will get

Algorithmsarequitegeneraldefinitionsofarithmeticsprocesses

Solved Example 2: Determine the keyword length of the given ciphertext enciphered by Vigenere.

CHREEVOAHMAERATBIAXXWTNXBEEOPHBSBQMQEQERBW
RVXUOAKXAOSXXWEAHBWGJMMQMNKGRFVGXWTRZXWIAK
LXFPSKAUTEMNDCMGTSXMXBTUIADNGMGPSRELXNJELX
VRVPRTULHDNQWTWDTYGBPHXTFALJHASVBFXNGLLCHR
ZBWELEKMSJIKNBHWRJGNMGJSGLXEYPHAGNRBIEQJT
AMRVLCRREMNDGLXRRIMGNSNRWCHRQHAEYEVTAQEBBI
PEEWEVKAKOEWADREMXMTBHHCHRTKDNVRZCHRCLQOHP
WQAIIWXNRMGWOIIFKEE

Solution:

Step 1: In order to find the keyword length, you can use Kasiski test

Note: Kasiski test searches the ciphertext for pairs of identical segments of length at least three and record the distance between the starting positions of the two segments as the keyword length. In a case of several

of such distances, you can compute the greatest common divisor as the key length.

From the ciphertext string, CHR occurs in 5 different places,
The positions of its occurrence, are: 1, 166, 236, 276 and 286
The distances from the first occurrence to the other four occurrences are:
166 – 1, 236 – 1, 276 – 1 and 286 – 1
= 165, 235, 275 and 285
So, the greatest common divisor for these four distances is 5,
Therefore, it highly possible for the keyword length to be 5

Step 2: Carry out a further confirmatory test on the derived keyword length using indices of coincidences

Note: The index of coincidence of x, denoted as $I_c(x)$, is defined as the probability that two random elements of x are identical where $x = x_1x_2\ldots x_n$ is a string of n alphabetic characters. Hence, the mathematical computation of $I_c(x)$ is,

$$I_c = \frac{\sum_{i=0}^{25}\binom{f_i}{2}}{\binom{n}{2}} = \frac{\sum_{=0}^{25} f_i(f_i-1)}{n(n-1)}$$

where$f_0, f_1, \ldots, f_{25}$ are the frequencies of *A, B, … Z in x* respectively

n = the total number of characters in the plaintext or (given ciphertext)

2 = denote the two elements picked at a time

Taking x to be a string of English alphabet and assume that probabilities of the English alphabet are denoted by $p_0, p_1, \ldots, p_{25}$
Then, it implies that,

$$p_i = \frac{f_i}{n} \quad and \quad p_{i+1} = \frac{f_i - 1}{n-1}$$

Therefore,

$$I_c = \frac{\sum_{=0}^{25} f_i(f_i-1)}{n(n-1)} = \sum_{0}^{25} \frac{f_i}{n}\frac{(f_i-1)}{(n-1)} = \sum_{0}^{25} p_i p_{i+1} = \sum_{0}^{25} {p_i}^2$$

By substituting the probabilities of occurrences of the English alphabet into:

$$I_c = \sum_{0}^{25} {p_i}^2 ,$$

$$I_c \approx \sum_{0}^{25} {p_i}^2 = 0.082^2 + 0.015^2 + \ldots + 0.001^2 = 0.065$$

Now, instead of taking x to be a string of English alphabet, let us assume it is a string of ciphertext, then the index of coincidence will be calculated as follows:

Given the ciphertext string $y = y_1y_2 \ldots y_n$ constructed with Vigenere cipher

You can then define m substrings of y, denoted by y_1, y_2, …,y_m, by writing out the ciphertext, in columns, in a rectangular array of dimensions m x (n/m).

The rows of this matrix are the substrings y_i, $1 \leq i \leq m$

That is, you will have a matrix like this:

$$\begin{pmatrix} y_1 & = & y_1 y_{m+1} y_{2m+1} & \cdots \\ y_2 & = & y_2 y_{m+2} y_{2m+2} & \cdots \\ \vdots & \vdots & \vdots & \cdots \\ ym & = & y_m y_{2m} y_{3m} & \cdots \end{pmatrix}$$

With the equation constructed above, the index of coincidence should be $I_c(y_i) \approx 0.065$ if m is the keyword length, but if it is not, then the index of coincidence will be more random away from 0.065. It could be something like 0.028. Clearly then, the value of m is not the keyword length.

Now, that you have background knowledge of how to compute, the index of coincidence, then compute for m = 5 from the given question,

When m = 1, $I_c(y_i)$ = 0.045

m = 2, the two indices are $I_c(y_i)$ = 0.46 and 0.41

m = 3, the indices are 0.043, 0.050, 0.047

m = 4, 0.042, 0.039, 0.045 and 0.040

m = 5, 0.063 0.068, 0.069, 0.061 and 0.072

Clearly, the values of $I_c(y_i)$ when m = 5 show closeness to 0.065

This further confirms that the keyword length is 5.

Solved Example 3: Given that a plaintext 'friday' is encrypted using a Hill Cipher with m = 2, and the resulting ciphertext is 'PQCFKU', cryptanalyse to find the secret key.

Solution:

Since the pairs of plaintext-ciphertext, the attack type to be used is referred to as known plaintext attack (KPA)

Step 1: Convert the two plaintext-ciphertext pairs to their equivalent integer values

This implies that,

f	*r*	*i*	*d*	*a*	*y*
5	*17*	*8*	*3*	*0*	*24*

P	*Q*	*C*	*F*	*K*	*U*
15	*16*	*2*	*5*	*10*	*20*

Step 2: Use the Hill cipher encryption function to form two equations

Recall: The encryption function is: $e_k(x) = xk$, for a key k

This implies that,

e_k = (5, 17) = (15, 16), e_k = (8, 3) = (2, 5)

Then, form a matrix equation as:

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} \times k$$

Therefore,

$$k = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$$ {Note: you will need to compute the inverse of a matrix and carry out matrix multiplication before you arrive at the answer}

4.0 Self-Assessment Exercise(s)

1: Given the k = $\begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$, use Hill cipher to decrypt the ciphertext 'PQCFKU'

A. monday
B. tuesday
C. friday
D. Sunday

Answer: C

2: One the fundamental basis guiding the operation of cryptanalysis is:

A. The power of your calculator
B. The amount of information available
C.The quality of your certificate
D. How fast you are

Answer: B

3: Given the first two letters of Affine enciphered ciphertext to be 'PQ' and the corresponding first two letters of the plaintext to be 'if', using the encryption function of the form: y = ax + b, where key k = (a, b), y is the ciphertext character and x is the plaintext character, find the key k.

A. k = (1, 2)
B. k = (17, 9)
C. k = (3, 6)
D. k = (3, 2)

Answer: B

5.0 Conclusion

Beyond being equipped with the requisite knowledge of how to secure your information over insecure communication channels in Unit 1, Unit 2 has given you a better understanding of techniques adopted by attackers to compromise the security. The quality and quantity of information made available to attackers either consciously or unconsciously determine the

type of cryptanalysis that can be launched. So, as much as possible, ensure that you proactively protected against any malicious activities.

6.0 Summary

You have learnt the types and techniques of cryptanalytic attacks and the basis of their operations. As clearly explained, the main goal of a cryptanalytic attacker is to explore available information to determine the secret key used to encrypt the ciphertext. The bases of their operations were itemised as the amount of information available, computational resources and types of cipher used. I also explored seven types of cryptanalysis with illustrative diagrams. The types are based on the information available to the cryptanalyst. The cryptanalytic attack types are Ciphertext Only Attack (COA), Known Plaintext Attack (KPA), Chosen Plaintext Attack (CPA), Chosen Ciphertext Attack (CCA), Adaptive Chosen Plaintext (ACP), Adaptive Chosen Ciphertext (ACC), Relative Key Attack (RKA).

In the application section, you have learnt the vulnerability of some existing applications to some of these cryptanalytic techniques. You were also equipped with some procedures of cryptanalysis in determining the secret key used for encryption. The essence of this is to ensure you are better informed and armed against the activities of malicious users. The next unit will teach you some probability methods that may be of importance to the security of information and communication.

7.0 References/Further Reading

https://www.researchgate.net/publication/319413716_Intelligent_Techniques_in_Cryptanalysis_Review_and_Future_Directions

http://www.uobabylon.edu.iq/eprints/paper_5_7264_649.pdf

Unit 3: Probability Theory

Contents

1.0 Introduction

In this unit, I will take you through the rudimentary of how to quantify the uncertainty about the occurrences of events. This is the essence of probability. The knowledge of probability theory enhances rational action and communication. You will be taught both the classical and modern probability theory.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- solve scenario-based probability combinatorial problems by making relevant inferences
- apply the conditional probability and Bayes' theorems in solving practical problems
- describe decisions based on scenarios using distribution functions of discrete variables

3.0 Main Content

3.1 Combinatorial Analysis and Combination of Events

Combinatorial analysis has to do with a single event being counted repeatedly. This principle is the one applicable to permutation and combination.

A class of problems can be considered in which the assignment of probabilities can be made naturally.

Let S be a finite or countably infinite set, and let B consist of all subsets of S.

For each point $w_i \in S$, i = 1, 2, …, let a nonnegative value p_i, wih $\sum p_i = 1$

If A is any subset of S, let $P(A) = \sum_{w_i \in A} p_i$

Then, P is a probability measure.

Basically, there are two combinatorial processes: they are permutation and combination. Permutation deals with arrangement while combination deals with selection. As the case may be, issues of with replacement or without replacement and ordered or unordered may come in. All these may affect the final results of processing or probability.

The formula for computing each is as follows:

Permutation = $^{n}P_{r} = \frac{n!}{(n-r)!}$

where n = total number of possibilities

r = number under consideration

Combination = $^{n}C_{r} = \frac{n!}{(n-r)!\,r!}$

Solved Example 1: How many combinations of two letter ‘C’ and one letter ‘P’ can be formed from a ciphertext string with 4 ‘C’s and 3 ‘P’s? Comments on the result of the probability.

Solution:
Step 1: Compute the combination using the formula

$${}^{n}C_{r} = \frac{n!}{(n-r)!r!}$$

$= {}^{4}C_{2} \times {}^{3}C_{1}$

$= \frac{4!}{(4-2)!2!} \times \frac{3!}{(3-1)!1!}$

= 6 x 3
= 18

Step 2: Compute the probability
Given the number of the event is 18,

The number of possibility = ${}^{7}C_{3}$ = $\frac{7!}{(7-3)!3!}$ = 35

Therefore, the probability = $\frac{18}{35}$ = 0.514

With this combination will form the keyword for cryptanalysis, then chances of success are 50%. Well, the small number of the ciphertext trying, you can go ahead to check the authenticity of the keyword, else try another keyword.

INQ1: In a given plaintext, there are three categories of letters noticed: the first category consist of 8 trigrams, the second category 6 digrams and the third category consists of 13 single letters. In how many of all possible samples of size 4, chosen without replacement, will every category of the letter be represented?

ANSWER: 7488

3.2 Conditional Probability

(1) Conditional probability

Given an event A and a discrete sample space B, then the probability of A in relation to the sample space B is given as follows:

$$P(A \text{ given } B) = \frac{\text{the number of elements in } A \cap B}{\text{the number of elements in } B}$$

From the above, the conditional probability of an event A, given the event B has occurred, is defined by

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

Provided $P(B) > 0$

This conditional probability measure satisfies three conditions below:

(1) $P(A/B) \geq 0$ for all event A

(2) $P(B/B) = 1$

(3) If A_1, A_2, …, are mutually exclusive events, then

(4) $$P\left(\bigcup_{k=1}^{\infty} A_k / B\right) = \sum_{k=1}^{\infty} P(A_k / B)$$

Solved Example 2: There are 240 characters in a ciphertext, out which 15 were used as the keyword for the secret key. During cryptanalysis, you pick two letters from the whole characters in succession (i.e. without replacement), what is the probability that they will belong to the keyword?

Solution:

$$P(A \cap B) = P(A)\,P(B/A)$$
$$= \left(\frac{15}{240}\right)\left(\frac{14}{239}\right) = \left(\frac{7}{1912}\right)$$

If an event A is independent of the event B, then

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

Provided $P(B) > 0$

Therefore, $$P(A/B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A)$$

It implies that,
$P(A/B) = P(A)$

(2) Bayes' Theorem

In many scenarios and situations around you, the ultimate outcome of an experiment depends on the occurrences in various intermediate stages. The problem of this nature is solved by Bayes' Theorem.

Given that S be a set or sample space and $P = \{A_i\}_{i=1}^{m}$ be a collection of subsets of S. Then the collection of P is called a partition of S if:

(i) $$S = \bigcup_{i=1}^{m} A_i$$

(ii) $$A_i \cap A_j = 0 \quad for\ i \neq j$$

Given the events $\{B_i\}_{i=1}^{m}$ constituting a partition of the sample space S such that $P(B_i) \neq 0$ $for\ i = 1, 2, \ldots, m,$ then for any event *A* in *S,*

$$P(A) = \sum_{i=1}^{m} P(B_i)\,P(A/B_i)$$

Given the events $\{B_i\}_{i=1}^{m}$ constituting a partition of the sample space S and $P(B_i) \neq 0 \text{ for } i = 1,2,\ldots,m,$, and if $P(A) \neq 0$, then for any event A in S,

$$P(B_k / A) = \frac{P(B_k)\,P(A/B_k)}{\sum_{i=1}^{m} P(B_i)\,P(A/B_i)} \quad k = 1,2,\ldots,m$$

Note: $P(B_k)$ is called prior probability while $P(B_k / A)$ is called the posterior probability

Solved Example 3: Two boxes containing tallies inscribed with ciphertext characters that can be cryptanalysed to obtain a secret key. Each tally was inscribed with either a lower case alphabet, upper case alphabet or a special character. The boxes are labelled B_1 and B_2. Box B_1 contains 7 tallies with lower case inscription and 4 tallies with upper case inscription. Box B_2 contains 3 tallies with lower case inscription and 10 tallies with special characters. The arrangement of these boxes gives the probability of selecting box B_1 to be $\frac{1}{3}$ and the probability of selecting box B_2 to be $\frac{2}{3}$. A cryptanalyst *Peace* is blindfolded and required to pick a tally. The cryptanalysis will be at least 50% correct if he selects a lower case inscribed tally.

(a) What is the probability that Peace will be at least 50% correct in cryptanalysing the ciphertext?

(b) If Peace achieves a minimum of 50% success in the cryptanalysis, what is the probability that the selected tally was from the first box?

Solution:

Note: For Peace to be at least 50% correct, he must select a lower case inscribed tally.

Assume A is the event of picking a tally with a lower case inscription

The prior probabilities are: $P(B_1) = \frac{1}{3}$ and $P(B_2) = \frac{2}{3}$

(a) $P(A) = P(A \cap B_1) + P(A \cap B_2)$

$= P(A/B_1)\,P(B_1) + P(A/B_2)\,P(B_2)$ [Substituting into conditional probability]

$$= \left(\frac{7}{11}\right)\left(\frac{1}{3}\right) + \left(\frac{3}{13}\right)\left(\frac{2}{3}\right)$$

$$= \frac{91}{429} + \frac{66}{429} = \frac{157}{429}$$

(b) Given that Peace attained at least 50% cryptanalysis success, the probability that the lower case inscribed tally was selected from B_1 is:

$$P(B_1 / A) = \frac{P(A/B_1)\,P(B_1)}{P(A/B_1)\,P(B_1) + P(A/B_2)\,P(B_2)}$$

$$= \frac{\left(\frac{7}{11}\right)\left(\frac{1}{3}\right)}{\left(\frac{7}{11}\right)\left(\frac{1}{3}\right)+\left(\frac{3}{13}\right)\left(\frac{2}{3}\right)}$$

$$= \frac{91}{157}$$

Note: $P(A/B_1)$ is the probability of picking a lower case inscribed tally from the box B_1 while $P(B_1/A)$ is the probability that the lower case inscribed tally was selected from box B_1.

Assignment 2:
Sixty per cent of new computer undergraduate student had cryptography knowledge. During their first year in the University, new students without the knowledge of cryptography have probability 0.08 of having sustaining cyber-attack, but the new students with knowledge of cryptography have only a 0.05 probability of an attack. What is the probability a new student has had cryptography knowledge, given that the student has had no cyber-attack the first year? Show your step.

ANSWER: = 0.6077

3.3 Distribution Functions of Discrete Variables

Given X is a discrete random variable with space S_x and probability density function *f(x)*, then

(a) $f(x) \geq 0$ for all x in S_x, and

(b) $\sum_{x \in S_X} f(x) = 1$

Solved Example 4: If the probability of decrypting a ciphertext forms a random variable X with sample space $S_x = \{1, 2, 3, \ldots, 12\}$ is given by

$f(x) = k(2x - 1)$

then, what is the value of the constant k?

Solution:

Recall: $\sum_{x \in S_X} f(x) = 1$

$$1 = \sum_{x \in S_X} f(x) = \sum_{x \in S_X} k(2x-1)$$

$$= \sum_{x=1}^{12} k(2x-1) = k\left[2\sum_{x=1}^{12} x - 12\right] = k\left[2\frac{(12)(13)}{2} - 12\right] = k \times 144$$

therefore,

$$k = \frac{1}{144}$$

Given a random variable X, cumulative distribution function C(x) of the random variable can be defined as:

$C(x) = P(X \leq x)$

for all real numbers x.

If X is a random variable with the space S_x, then

$$C(x) = \sum_{t \leq x} f(t) \text{ for } x \in S_X$$

Solved Example 5: If the probability density function of the random variable X for decrypting a ciphertext is given by

$$\frac{1}{144}(2x-1) \text{ for } x = 1, 2, 3, \ldots, 12,$$

Then find the cumulative distribution function of x = 3.

Solution:

Space of the random variable X is defined by $S_x = \{1, 2, \ldots, 12\}$

Therefore,

$$C(x) = \sum_{t \leq x} f(t)$$

$$C(3) = \sum_{t \leq 3} f(t) = f(1) + f(2) + f(3) = \frac{1}{144} + \frac{3}{144} + \frac{5}{144} = \frac{9}{144}$$

4.0 Self-Assessment Exercise(s)

1: A ciphertext string of 500 characters consists of 382 trigrams and 362 digrams. How many characters belong to both trigrams and digrams? If each of the characters belongs to at least one of the grams

[Hint: Use formula: $P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$]

A. – 244

B. 20

C. 1244

D. 244

ANSWER: 244

2. Let A and B be independent events with P(A) = P(B) = P(A u B) = 0.5. What is the probability of event A?

ANSWER: 0.2929

5.0 Conclusion

This unit has explored the fundamentals of taking rational decisions for informed actions. You are bound to encounter uncertainties in life which will demand a prompt and wise decision. With what I have taken you through, you are in a better position to take the required actions needed for enhanced information security.

6.0 Summary

In this unit, you are taught some classical probability theory needed for information security. The highlights are combinatorial analysis which deals more with repeated counting of a single event. This includes aspects of permutation and combination of two events which is a step further in combining more than one event in a scenario. Conditional probability which exposed you to probabilities depending on previous events. Bayes' theorem which is an extension of conditional probabilities but plays emphasis on the effects of intermediate stages. Distribution functions of discrete variables. Since most of the operations in computing are binary, this aspect exposed you to the functions of discrete or digital variables.

In the next unit, I will introduce you to entropy which is a measure of randomness, which is a major determinant factor in the security of a cryptographic key.

7.0 References/Further Reading

Sahoo, P. (2013). *Probability and Mathematical Statistics*. Louisville, KY USA: University of Louisville: https://www.academia.edu/32442387/probability_and_mathematical_statistics

http://people.math.harvard.edu/~ctm/papers/home/text/class/harvard/154/course/course.pdf

Lo, G. S. (2016). *Statistics and Probability African Society Books Series*. Calgary, Alberta: SPAS Textbooks

Series:.https://arxiv.org/pdf/1808.01713.pdf

Unit 4: Entropy

Contents

1.0 Introduction

Having learnt some basics on the classical probability theory, I will introduce you to some related aspects of it in information theory. You will learn about the randomness of the secret key and its significance in information security. You will also be abreast with the concepts of entropy and perfect secrecy in addition to relevant theories.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- compare and contrast the concepts and properties of entropy and perfect secrecy
- apply relevant theories to back up information security proposition
- describe entropy and perfect secrecy principles in enhancing the security of secret keys against cryptanalytic attacks.

3.0 Main Content

3.1 Concepts of Perfect Secrecy and Entropy

3.1.1 Perfect secrecy

The concept of perfect secrecy is about the procedure of ensuring that no meaningful information about the plaintext can be obtained by observing the ciphertext. For the concept to be more precise and understandable, the following definitions with illustrative example are given.

Given that *x* belongs to a plaintext P and *y* belongs to a ciphertext C, then a cryptosystem is said to have perfect secrecy if and only if:

$P(x/y) = P(x)$

That is, a prior probability of the plaintext x and a posterior probability of that plaintext x are identical.

Solved Example 1: Given a set of plaintext characters P = {r, q}, a set of ciphertext characters C = {1, 2, 3, 4} encrypted with a set of keys K = {k_1, k_2, k_3} such that P(r) = $\frac{1}{4}$, P(q) = $\frac{3}{4}$, P(k_1) = $\frac{1}{2}$, P(k_2) = P(k_3) = $\frac{1}{4}$ and assuming the encryption functions are defined to be:

$$e_{k_1}(r)=1, e_{k_1}(q)=2, e_{k_2}(r)=2, e_{k_2}(q)=3, e_{k_3}(r)=3, e_{k_3}(q)=4$$

Using the information provided above to attempt the following questions:

(i) Depict the cryptosystem using a table of encryption matrix

(ii) Compute the probability distribution on the ciphertext

(iii) Compute the conditional probability distributions on the plaintext given that a particular ciphertext character has been observed

(iv) Based on your results in (ii) and (iii), provide a comment with respect to the issue of perfect secrecy.

Solution:

(i) Encryption matrix is given as:

	r	q
k_1	1	2
k_2	2	3
k_3	3	4

(ii) Computation of the probability distribution on the ciphertext characters

Use the table of encryption matrix above to check and compute the probabilities of the keys and the plaintext characters that generated each of the ciphertext elements as follows:

$P(1) = P(k_1) \times P(r) = \frac{1}{2} \times \frac{1}{4} = \frac{1}{8}$

$P(2) = P(k_1) \times P(q) + (k_2) + P(r) = \frac{1}{2} \times \frac{3}{4} + \frac{1}{4} \times \frac{1}{4} = \frac{7}{16}$

$P(3) = P(k_2) \times P(q) + P(k_3) \times P(r) = \frac{1}{4} \times \frac{3}{4} + \frac{1}{4} \times \frac{1}{4} = \frac{1}{4}$

$P(4) = P(k_3) \times P(q) = \frac{1}{4} \times \frac{3}{4} = \frac{3}{16}$

(iii) Computing the conditional probability distribution on the plaintext, given that a certain ciphertext has been observed.

$$P(r/1) = \frac{P(r \cap 1)}{P(1)}$$

Use the Encryption matrix table to evaluate, $P(r \cap 1)$

Which gives, $P(r \cap 1) = P(r) \times P(k_1)$

Therefore,

$$P(r/1) = \frac{P(r \cap 1)}{P(1)} = \frac{P(r) \times P(k_1)}{P(1)} = \frac{\frac{1}{4} + \frac{1}{2}}{\frac{1}{8}} = 1$$

$$P(r/2) = \frac{P(r \cap 2)}{P(2)} = \frac{P(r) \times P(k_2)}{P(2)} = \frac{\frac{1}{4} \times \frac{1}{4}}{\frac{7}{16}} = \frac{1}{7}$$

$$P(r/3) = \frac{P(r \cap 3)}{P(3)} = \frac{P(r) \times P(k_3)}{P(3)} = \frac{\frac{1}{4} \times \frac{1}{4}}{\frac{1}{4}} = \frac{1}{4}$$

$$P(r/4) = \frac{P(r \cap 4)}{P(4)} = \frac{P(r) \times P(k_3)}{P(4)} = \frac{\frac{1}{4} \times 0}{\frac{3}{16}} = 0$$

$$P(q/1) = \frac{P(q \cap 1)}{P(1)} = \frac{P(q) \times P(k_3)}{P(1)} = \frac{\frac{1}{4} \times 0}{\frac{1}{8}} = 0$$

$$P(q/2) = \frac{P(q \cap 2)}{P(2)} = \frac{P(q) \times P(k_1)}{P(2)} = \frac{\frac{3}{4} \times \frac{1}{2}}{\frac{7}{16}} = \frac{6}{7}$$

$$P(q/3) = \frac{P(q \cap 3)}{P(3)} = \frac{P(q) \times P(k_2)}{P(3)} = \frac{\frac{3}{4} \times \frac{1}{4}}{\frac{1}{4}} = \frac{3}{4}$$

$$P(q/4) = \frac{P(q \cap 4)}{P(4)} = \frac{P(q) \times P(k_3)}{P(4)} = \frac{\frac{3}{4} \times \frac{1}{4}}{\frac{3}{16}} = 1$$

(iv) Comparing the probabilities off ciphertext in (ii) and (iii), the property of perfect secrecy is satisfied with the ciphertext y = 3 but not for the three other
ciphertexts.

3.1.2 Entropy

Entropy is a measure of information or uncertainty. The higher the uncertainty, the higher the entropy and vice versa. The knowledge of entropy of information is used to build a more random key. The higher the randomness of a key or key generator, the more secure the ciphertext.

If given X to be a random variable having a finite set of values x_1, x_2, …, x_n, with probability $P(X = x_i) = p_i$, where $0 \le p_i \le 1$ for each I, $1 \le I \le n$, and where $\sum_{i=1}^{n} p_i = 1$, therefore the entropy of X is a mathematical measure of the amount of information provided by the observation of X.

The formula for the computation of entropy is as follows: Let entropy denoted by H(X) or uncertainty of X be defined by

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i = \sum_{i=1}^{n} \log p_i \left(\frac{1}{p_i}\right)$$

For you to find it more conveniently, you can use the short formula below for computing entropy:

$$H(X) = p_i \log p_i = p_i \log \frac{1}{p_i}$$

You can think of every component of a cryptosystem as a random variable. For example, if you take the characters of plaintext, ciphertext and the keys as random variables, then you can compute their entropies.

Solved Example 2: As in the previous example, given a set of plaintext characters P = {r, q}, a set of ciphertext characters C = {1, 2, 3, 4} encrypted with a set of keys K = $\{k_1, k_2, k_3\}$ such that $P(r) = \frac{1}{4}$, $P(q) = \frac{3}{4}$, $P(k_1) = \frac{1}{2}$, $P(k_2) = P(k_3) = \frac{1}{4}$ and assuming the encryption functions are defined to be:

$$e_{k_1}(r) = 1, e_{k_1}(q) = 2, e_{k_2}(r) = 2, e_{k_2}(q) = 3, e_{k_3}(r) = 3, e_{k_3}(q) = 4$$

Compute the entropies for the plaintext and the ciphertext.

Note: $\log_2 y =$ undefined if y= 0, but to solve this problem in entropy, you use limit as y tends to zero as,

$$\lim_{y \to} y \log_2 y = 0$$

Solution:

Recall: $H(X) = -\sum_{i=1}^{n} p_i \log p_i = \sum_{i=1}^{n} \log p_i \left(\frac{1}{p_i}\right)$

To compute the entropy for the plaintext P = {r, q},
Let the entropy be denoted by H,
Then $H(r, q) = - P(r) \log_2 P(r) + [- P(q) \log_2 (q)]$

$$= -\frac{1}{4}\log_2 \frac{1}{4} - \frac{3}{4}\log_2 \frac{3}{4}$$

$$= -\frac{1}{4}\log_2 \frac{1}{2^2} - \frac{3}{4}\log_2 \frac{3}{2^2}$$

$$= -\frac{1}{4}\log_2 2^{-2} - \frac{3}{4}\left[\log_2 3 - \log_2 2^2\right]$$

$$= -\frac{1}{4}(-2)\log_2 2 - \frac{3}{4}\left[\log_2 3 - 2\log_2 2\right]$$

Recall that $\log_2 2 = 1$[according laws of logarithms]
therefore, this implies that

$$= \frac{1}{2} - \frac{3}{4}[\log_2 3 - 2]$$
$$= \frac{1}{2} - \frac{3}{4}\log_2 3 + \frac{3}{2}$$
$$= 2 - \frac{3}{4}\log_2 3$$
$$\approx 0.81$$

Use the same question in Solved Example 2 to calculate the entropy of the key K

ANSWERS:
H(K) = 1.5

3.2 Properties and Theoretical Framework of Entropy

Some of the essential properties of entropy are as follows:
Assuming X be a random variable which n values.

(i) *$0 \leq H(X) \leq \log n$*

(ii) *$H(X) = 0$ if and only if $p_i = 1$ for some i, and $p_j = 0$ for all $j \neq i$* (that is, there is no uncertainty of the outcome.

(iii) *$H(X) = \log n$ if and only if $p_i = \frac{1}{n}$ for each i, $1 \leq i \leq n$* such that all outcomes are equally likely

Theorem 1: *Given a random variable X with a probability distribution having values $p_1, p_2, \ldots, p_n$, where $p_i > 0$, $1 \leq i \leq n$. Then $H(X) \leq \log_2 n$, with equality if and only if $p_i = \frac{1}{n}$, $1 \leq i \leq n$.*

Theorem 2: When you have more than one random variable to compute entropy, then it is referred to as joint entropy. Given two random variables X and Y, their joint entropy is given as,

$$H(X,Y) = -\sum_{x,y} P(X = x, Y = y)\log(P(X = x, Y = y)),$$

Where the summation indices x and y respectively range overall X and Y. The number of random variables can be more than two, and the definition can be extended to cover the number.

Theorem 3: Given two random variables X and Y, $H(X, Y) \leq H(X) + H(Y)$, with equality if and only if X and Y are independent random variables

Theorem 4: Given that X and Y are two random variables,
(i)the conditional entropy of X given that Y = y is given as,

$$H(X/Y=y)=-\sum_{x} P(X=x/Y=y)\log(P(X=x/Y=y)),$$

Where the summation index x ranges overall values of X

(ii) The conditional entropy of X given Y also referred to as equivocation of Y about X, is given as,

$$H(X/Y)=\sum_{y} P(Y=y)H(X/Y=y)$$

Where the summation index y ranges overall values of Y

Some properties of conditional entropy are: Given that X and Y are random variables,

(i) The quantity H(X/Y) measures the amount of uncertainty remaining about X after Y has been observed.

(ii) H(X/Y) ≥ 0 and H(X/Y) = 0

(iii) H(X/Y) = H(X) + H(Y/X) = H(Y) + H(X/Y)

(iv) H(X/Y) ≤ H(X), with equality if and only if X and Y are independent.

If given the quantity H(X/Y) as a measure of the amount of uncertainty in the random variables X and Y, what is the condition required for H(X/Y) to be equal to H(X)?

ANSWER: The random variables X and Y must be independent.

3.3 Applications of Entropy

3.3.1 Key Equivocation

There are several applications of entropy. One of them is using the concept of conditional entropy to measure the amount of uncertainty of the secret key remaining when the ciphertext is known. This is also referred to as the key equivocation. It is computed based on the following theorem:

Given a cryptosystem (P, C, K, E, D), then

H(K/C) = H(K) + H(P) – H(C)

Where H(K) is the entropy of the key

H(P) is the entropy of the plaintext

H(C) is the entropy of the ciphertext

Solved Example 3: Compute the key equivocation of the secret key used in the Solved Example 2.H(P) has already been calculated, and the value is 0.81.

Solution:

You should first of all calculate H(C) and H(K). Assuming the answers are: H(C) = 1.85 and H(K) = 1.5

H(K/C) = H(K) + H(P) – H (C)

H(K/C) = 1.5 + 0.81 – 1.85

H(K/C) = 0.46

Compute the key equivocation of the secret key if the entropies of the plaintext, ciphertext and key are 0.63, 1.79 and 1.4 respectively
ANSWER: 0.24

3.3.2 Entropy of Natural Language: Redundancy, Spurious Keys and Unicity Distance

The characters of plaintext are from a natural language such as English. Generally during cryptanalysis, an attacker will rule out some keys. However, other possible keys will remain for the purpose of cryptanalysis, especially in ciphertext only attack. Out of these possible keys, only one key is correct. The remaining possible but incorrect keys are referred to as spurious keys.

Given that L is a natural language, H_L is the entropy of L, then,

$$H_L = \lim_{n\to\infty} \frac{H(P^n)}{n}$$

where P^n is the random variable whose probability distribution is that of all n-grams of plaintext.
And the redundancy of L which is the number of 'excess characters' denoted y R_L is expressed as,

$$R_L = 1 - \frac{H_L}{\log_2 |P|}$$

Given a cryptosystem where $|C| = |P|$ and keys are chosen equiprobably. Assuming the redundancy of the underlying language is R_L. If given a sufficiently large string of ciphertext of length n, the expected number of spurious keys, $\overline{s_n}$, satisfies

$$\overline{s_n} \geq \frac{|K|}{|P|^{nR_L}} - 1$$

The quantity $\frac{|K|}{|P|^{nR_L}} - 1$ approaches 0 exponentially quickly as n increases.
This accuracy of this estimation may be poor for small values of n, especially since $\frac{H(P^n)}{n}$ also affects the estimation if H_L if n is small.
Another very important area of application is the calculation of the unicity distance of a cryptosystem. It is the value of n, usually denoted as n_0, at which the expected number of spurious keys becomes zero, i.e. it is the average amount of ciphertext required for an opponent to uniquely compute the key if given enough computing time.
The unicity distance is computed using the following formula:

$$n_0 \approx \frac{\log_2 |K|}{R_L \log_2 |P|}$$

Solved Example 4: Considering substitution cipher cryptosystem in which $|P| = 26 \; and \; |K| = 26!$. Assuming the redundancy of the language is 0.82, compute the unicity distance.

Solution:

$$n_0 \approx \frac{\log_2 |K|}{R_L \log_2 |P|}$$

$$= \frac{\log_2 |26!|}{0.82 \times \log_2 26} = \frac{\log_2 |26 \times 25 \times 24 \times ... \times 1|}{0.82 \times \log_2 26}$$

$$= \frac{88.4}{(0.82 \times 4.7)} = \frac{88.4}{3.854} \approx 22.9$$

This implies that unique decryption is possible if given a ciphertext string of length at least 22.9%.

4.0 Self-Assessment Exercise(s)

1. Given a set of plaintext characters P = {r, q}, a set of ciphertext characters C = {1, 2, 3, 4} encrypted with a set of keys K = {k_1, k_2, k_3} such that P(r) = $\frac{1}{4}$, P(q) = $\frac{3}{4}$, P(k_1) = $\frac{1}{2}$, P(k_2) = P(k_3) = $\frac{1}{4}$ and assuming the encryption functions are defined to be:

   $$e_{k_1}(r) = 1, e_{k_1}(q) = 2, e_{k_2}(r) = 2, e_{k_2}(q) = 3, e_{k_3}(r) = 3, e_{k_3}(q) = 4$$

   Compute the entropy for the ciphertext.

   ANSWER: H(C) = 1.85

2. _____ is the average amount of ciphertext required for an opponent to uniquely compute the key if given enough computing time.

   ANSWER: Unicity distance

5.0 Conclusion

No better security can be achieved without the knowledge of associated uncertainties. With informed concepts and theories of perfect secrecy and entropy, you have been guided on the exposure of information to attackers and the uncertainty level of your protection.

6.0 Summary

In this unit, you have been taught the concepts of perfect secrecy and entropy. As perfect secrecy exposed you to the procedures of ensuring security, entropy shows you the uncertainties of your security mechanisms based on controllable and some other parameters beyond your control. I made available some relevant theories and definitions to you. These are to guide you in the application areas. In a nutshell, some application areas like the entropy of natural language, key equivocation, redundancy, spurious keys and unicity distance were covered. The next unit will introduce you to a combination of more than one cryptosystems to form another cryptosystem in order to have a system with a better entropy.

7.0 References/Further Reading

https://www.eit.lth.se/fileadmin/eit/courses/edi051/lecture_notes/LN3.pdf

https://www.mdpi.com/1099-4300/19/3/100/htm

Unit 5: Product Cryptosystem

Contents

1.0 Introduction

Building on the previous four units and particularly the last unit, you will learn how to combine more than one cryptosystem to form a better cryptosystem. With this improvement, you will have a better entropy. The cryptosystem formed as a result of this combination is referred to as **product cryptosystem**.

Product cryptosystem is a cryptosystem basically formed by combining two cryptosystems together. This type of cryptosystem has been fundamental to the design and building of more recent cryptosystems such as Advanced Encryption Standard.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- examine the essential definitions and theorems of product cryptosystems
- build an endomorphic product cryptosystems
- analyse the relevant properties of classical and product cryptosystems.

3.0 Main Content

3.1 Essentials of Product Cryptosystems

Product cryptosystem is the combination of two cryptosystems in encoding the plaintext. That is, you will first encrypt the given plaintext

with the first cryptosystem and then encrypt the resultant ciphertext using the next cryptosystem.

Given two cryptosystems:

$T_1 = (P_1, C_1, K_1, E_1, D_1)$ and $T_2 = (P_2, C_2, K_2, E_2, D_2)$

The product cryptosystem of T_1 and T_2 is denoted by $T_1 \times T_2$

For simplicity, the focus is to define the product cryptosystems in which the ciphertext is equal to plaintext. This type of cryptosystem is referred to as an endomorphic cryptosystem.

Given two cryptosystems A and B. If A is defined as (P_1, C_1, K_1, E_1, D_1) and B as (P_2, C_2, K_2, E_2, D_2),
Where P_i, C_i, K_i, E_i, D_i are plaintexts, ciphertexts, keys, encryption functions and decryption functions of A and B respectively for i = 1, 2
Using appropriate notation, write the expressions for the encryption and decryption functions.
The answer will be,
the encryption function is: $e_{(k_1,k_2)}(x) = e_{k_2}(e_{k_1}(x))$
decryption function is: $d_{(k_1,k_2)}(y) = e_{k_1}(e_{k_2}(y))$

3.2 Building of Endomorphic Product Cryptosystems

A cryptosystem is said to be endomorphic if the plaintext, P is equal to the ciphertext, C. For endomorphic product cryptosystems, it implies that the plaintext and ciphertext in both cryptosystems are the same.

Therefore, given two cryptosystems:

$T_1 = (P_1, C_1, K_1, E_1, D_1)$ and $T_2 = (P_2, C_2, K_2, E_2, D_2)$

Then the endomorphic cryptosystem ($T_1 \times T_2$) is given by:

$(P, P, K_1 \times K_2, E, D)$

where $P = P_1 = C_1 = P_2 = C_2$

the key $k = (K_1, K_2)$

the encryption function E is: $e_{(k_1,k_2)}(x) = e_{k_2}(e_{k_1}(x))$

the decryption function D is: $d_{(k_1,k_2)}(y) = e_{k_1}(e_{k_2}(y))$

ITQ2: Two cryptosystems are said to be endomorphic is the encryption and decryption functions are the same. (True or False)

Answer: False. It is if the plaintext and ciphertext are the same.

3.3 Properties of Classical and Product Cryptosystems

At this point, you need to be conversant with some relevant properties of cryptosystems. The properties are commutativity, associativity and idempotency.

(1) Commutativity:
Two cryptosystems T_1 and T_2 are said to commute or to be commutative if:
$T_1 \times T_2 = T_2 \times T_1$
Note: Not all pairs of cryptosystems do commute. Therefore, product operations of cryptosystems are not always commutative

(2) Associativity:
Three cryptosystems T_1, T_2 and T_3 are said to be associative if:
$(T_1 \times T_2) \times T_3 = T_1 \times (T_2 \times T_3)$
Note: Product operation on cryptosystems is always associative.
If you take the product of an endomorphic cryptosystem T with itself, you will definitely obtain the cryptosystem T x T, which is denoted as T^2.
If the n-fold product is considered, then the resulting cryptosystem is denoted by T^n.

(3) Idempotent property:
A cryptosystem T is said to be an idempotent cryptosystem if $T^2 = T$. Many of the classical cryptosystems you learnt in Unit 1 are idempotent. For example, Shift, Substitution, Affine, Hill, Vignere and Permutation Ciphers are all idempotent. For idempotent cryptosystem T, there is no need using the product system T^2 because there is no provision of extra security but requires an extra key.

Mini Project: Use appropriate mathematical notations, expression the following properties about cryptosystems and product cryptosystem: commutativity, associativity and idempotency. And solve the expression to solve a relevant example.

4.0 Self-Assessment Exercise(s)

1. Based on your knowledge of cryptosystem so far, critically give the major difference between the classical cryptosystems and product cryptosystem
   Answer:
   Classical cryptosystems are single-typed cryptosystem while product cryptosystem as its name suggests is a result or product of two cryptosystems.

2. If a cryptosystem A is idempotent, then there is no point in using A^2 to encrypt instead of A. Justify.

Answer:
For idempotent cryptosystem A, it implies that:

$$A^2 = A$$

Therefore, the extra key generated by A^2 used for encryption will be a waste as no additional security added to A^2

5.0 Conclusion

You have been abreast with the essence of product cryptosystems. For simplicity, the endomorphic product cryptosystem is a special type that you can easily examine. The knowledge acquired in this unit will help you to figure out how can own combine two cryptosystems to achieve better security.

6.0 Summary

The fundamentals of enhancing the security of classical cryptosystems have been explored. You have been taught the basic definitions and theorems relevant to the building of product cryptosystem. You learnt the special type of product cryptosystem in which the plaintext and ciphertext are the same, which is referred to endomorphic cryptosystem. You were also exposed to the relevant properties of classical and product cryptosystems. In the next unit, you will go further in the category of cryptanalysis of cryptosystems, by learning linear cryptanalysis.

7.0 References/Further Reading

http://www.maths.unp.ac.za/coursework/Math236/2008/chapter%209%20DES.pdf

https://crypto.stanford.edu/~dabo/cryptobook/draft_0_2.pdf

Module 2: Block Cipher and Advanced Encryption Scheme

Module Introduction

In this module, you will learn block cipher and Advance Encryption Scheme (AES) and get yourself ready to explore the world of cryptanalysis. By the time you complete the module, you should be able to explain the concept of differential and linear cryptanalysis.

The module is organised into three units. These are as follows:

Unit 1: Linear Cryptanalysis
Unit 2: Differential Cryptanalysis
Unit 3: Data and Advanced Encryption Standard

Unit 1: Linear Cryptanalysis

Contents

1.0 Introduction

Building on previous knowledge, in this Unit, you are going to learn the basic of linear cryptanalysis as regards to information security. Also, you will learn the concept of block cipher design.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- examine the concept of block cipher design
- demonstrate the use of substitution-permutation networks mathematical model in finding solution to challenges in cyber security
- apply linear cryptanalysis in solving real life problems.

3.0 Main Content

3.1 Concept of Block Cipher Design

Block ciphers are the building blocks of many cryptographic constructions including hash functions, encryption schemes and message authentication codes. The two popular design models for block ciphers are Feistel Networks and Substitution-Permutation Networks. In this unit, you will learn about the latter. The former is for more advance study.

3.2 Demonstration of Substitution – Permutation Networks

Substitution-Permutation Network (SPN) is a model of a block cipher that is being iterated in its operation. The plaintext and ciphertext are taken as blocks consisting of bits that is equivalent to *l x m.* In each round, there is a combination of the round key with the input to the round through XOR operation. This is called a key mixing layer. Also, there is a successive application of a substitution function and a permutation function on the *l x m* bit input to that round.

Solved Example 1: Given that $l = m = Nr = 4$, where they all belong to positive integers. If π_S is defined by the table below where z is the input and $\pi_S(z)$ written in hexadecimal notation:

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

So, build the Substitution – Permutation Network (SPN) and use it to encrypt a plaintext x = 0010 0110 1011 0111 supposing the key k is:

k = 0011 1010 1001 0100 1101 0110 0011 1111

and the round keys are as follows:

k^1 = 0011 1010 1001 0100
k^2 = 1010 1001 0100 1101
k^3 = 1001 0100 1101 0110
k^4 = 0100 1101 0110 0011
k^5 = 1101 0110 0011 1111

Solution:
Before you start solving, take note of the following:
$l, m, Nr \in$ *positive integer*
π_S is the permutation or substitution box (S-box) to replace *l* – bits (i.e. 4)
π_P is the permutation to replace *l m* bits (4 x 4= 16)
The number of bits of the plaintext and ciphertext will be equal to *l m (4 x 4 = 16)*
You will build the SPN from two components: π_S and π_P
Now, you should take note of the procedures to follow throughout the stages of solving the question. They are:

(1) Mixing key, i.e. Modulo-2 operation or XOR-operation between the initial key, k^1 and the plaintext binary values.

(2) Substitution using the substitution function in the hexadecimal, $\pi_S(z)$ the table given above

(3) Permutation using the Substitution-Permutation Network

Step 1: Develop the Table of Substitution and the Framework for the Substitution-Permutation Blocks

First, develop the substitution table since the values in the substitution function, $\pi_S(z)$, given above are in hexadecimal
This table provides the binary equivalent of the hexadecimal

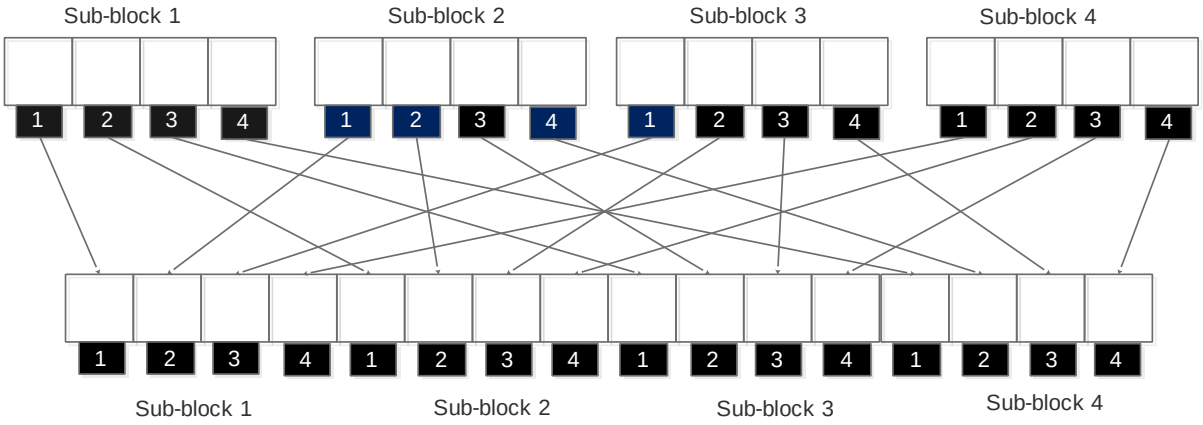
z	Input Bit-Stream to Substitution-Box(S-Box)				$\pi_S(z)$	Output Bit-Stream from the Substitution-Box(S-Box)			
0	0	0	0	0	E	1	1	1	0
1	0	0	0	1	4	0	1	0	0
2	0	0	1	0	D	1	1	0	1
3	0	0	1	1	1	0	0	0	1
4	0	1	0	0	2	0	0	1	0
5	0	1	0	1	F	1	1	1	1
6	0	1	1	0	B	1	0	1	1
7	0	1	1	1	8	1	0	0	0
8	1	0	0	0	3	0	0	1	1
9	1	0	0	1	A	1	0	1	0

A	1	0	1	0	6	0	1	1	0
B	1	0	1	1	C	1	1	0	0
C	1	1	0	0	5	0	1	0	1
D	1	1	0	1	9	1	0	0	1
E	1	1	1	0	0	0	0	0	0
F	1	1	1	1	7	0	1	1	1

Since l = 4, the total length of the permutation blocks is l x m = 4 x 4 = 16 but the length for the substitution block is 4 (since l = 4)
The framework of the permutation block will look like the diagram below:

Try and study the pattern of connecting arrows. Some of the identified principles are:

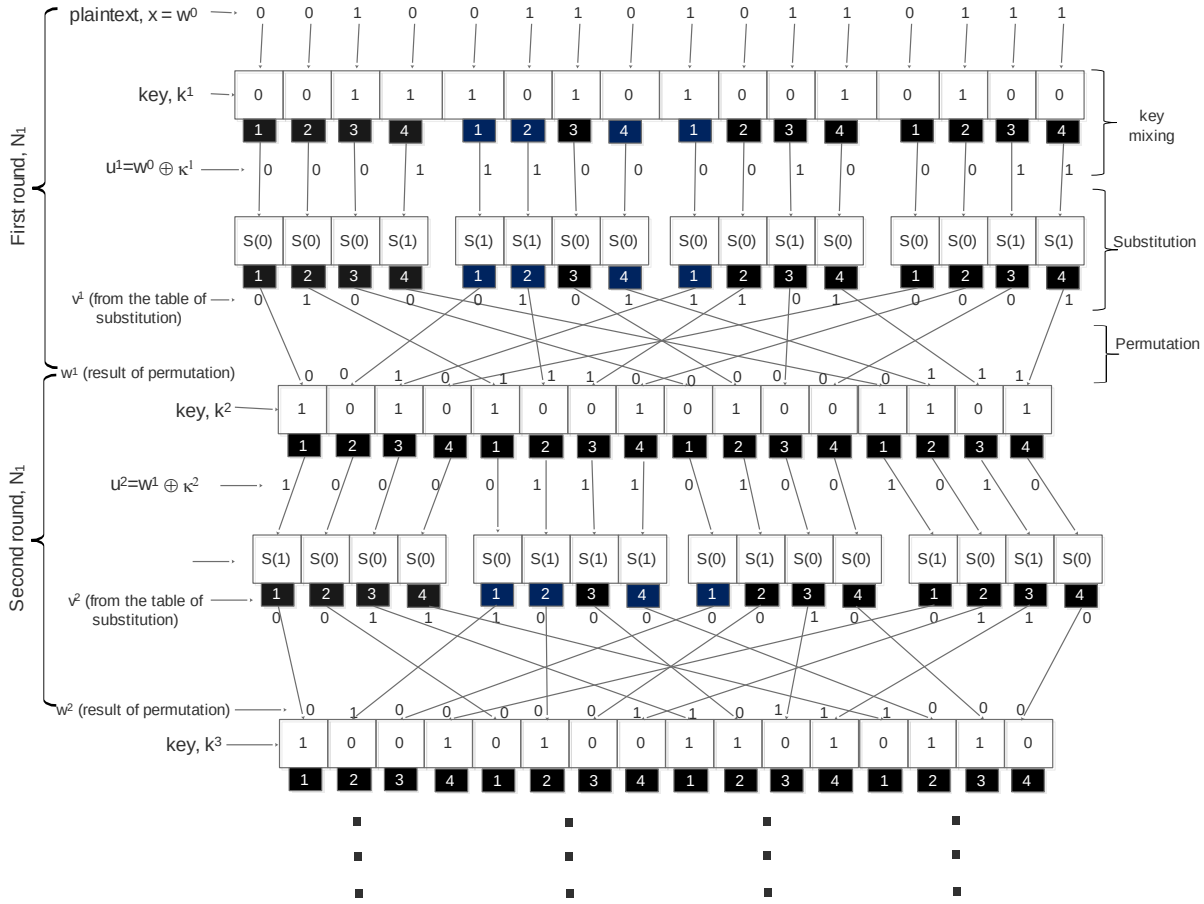
- Pin 1 in sub-block 1 connects to pin 1 in the merged sub-block 1, pin 2 in sub-block 2 connects pin 2 in the merged sub-block 2 and so on.
- Other pins in sub-block 1(i.e. pin 2, 3, 4) connect pins 1 in the other merged sub-blocks, other pins in sub-block 2(i.e. 1, 3, 4) connect pins 2 in the other merged sub-blocks and so on.



Step 2: Start the iteration of the encryption process by carrying out rounds of operation. In each round, you carry out three basic processes:

(1) Key mixing (i.e. modulo 2 addition or XOR operation between plaintext and the key),
(2) Substitution
(3) Permutation

A round of processes is depicted in the diagram below:



At the 5th round, the key, k = 1011110011010110 and the ciphertext y = 1011110011010110

3.3 Application of Linear Cryptanalysis

The major area of application of linear cryptanalysis is a known-plaintext attack. It requires accessing a large volume of plaintext and ciphertext pairs encrypted with the same unknown key. All the possible keys will be used to decrypt each of the plaintext-ciphertext pairs. The resulting intermediate ciphertext is studied to seek the least random result. A subkey which generates the least random intermediate cipher for all ciphertext becomes a candidate key.

4.0 Self-Assessment Exercise(s)

1. Block ciphers are the building blocks of many cryptographic constructions including hash functions, encryption schemes and message authentication codes.
   A. False
   B. True

   Answer: A

2. In each round, there is a combination of the round key with the input to the round through XOR operation.
   A. OR operation
   B. NOT operation
   C. XOR operation
   D. AND operator.

   Answer: C

5.0 Conclusion

In this unit, you have learnt, linear cryptanalysis as a general form of cryptanalysis based on finding affine approximations to the action of a cipher. And how attacks have been developed for block ciphers and stream ciphers. Linear cryptanalysis is one of the two most widely used attacks on block ciphers.; the other being differential cryptanalysis.

6.0 Summary

You can summarise that Substitution-Permutation Network (SPN) is a model of a block cipher that is being iterated in its operation. Also, the major area of application of linear cryptanalysis is known plaintext attack.

7.0 References/Further Reading

https://www.engr.mun.ca/~howard/PAPERS/ldc_tutorial.pdf

https://www.researchgate.net/publication/314864829_Linear_Cryptanalysis_Key_Schedules_and_Tweakable_Block_Ciphers

Unit 2 Differential Cryptanalysis

Contents

1.0 Introduction

In this unit, you will move forward from previous knowledge on linear cryptanalysis. Here, you are going to learn the basic of differential cryptanalysis and the concept of substitution-permutation model.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- demonstrate the use of substitution-permutation model in introducing many of the concepts of modern block cipher design
- compare and contrast differential cryptanalysis and linear cryptanalysis
- utilise relevant lemmas and theories in applying differential cryptanalysis to solve problems.

3.0 Main Content

3.1 Demonstration of Differential Cryptanalysis

In many aspects, differential cryptanalysis has operations similar to linear cryptanalysis. The major difference of differential cryptanalysis from linear cryptanalysis is its involvement in comparing the x-or operations of two inputs to the x-or operations of the corresponding two outputs.

Solved Example 1:
Suppose that l = m = Nr = 4, let π_S be defined as follows in Table 1, the input (i.e. z) and the output (i.e. π_S (z)) are written in hexadecimal notation in Table 2.

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
π_S(z)	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Table 2:

Hexadecimal	Bitstring			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Now, suppose you consider input x-or or x' = 1011. Carry out differential cryptanalysis.

Solution:

Δ(1011) = {(0000, 1011), (0001, 1010), …, (1111, 0100)}

For each ordered pair in the set Δ(1011), you compute output x-or of π_S. In each row of the following table, you have x $\otimes$x* = 1011, y = π_S(x), y* = π_S(x*) and y' = y $\otimes$y*:

X	x*	Y	y*	y'
0000	1011	1110	1100	0010
0001	1010	0100	0110	0010
0010	1001	1101	1010	0111
0011	1000	0001	0011	0010
0100	1111	0010	0111	0101
0101	1110	1111	0000	1111
0110	1101	1011	1001	0010

0111	1100	1000	0101	1101
1000	0011	0011	0001	0010
1001	0010	1010	1101	0111
1010	0001	0110	0100	0010
1011	0000	1100	1110	0010
1100	0111	0101	1000	1101
1101	0110	1001	1011	0010
1110	0101	0000	1111	1111
1111	0100	0111	0010	0101

Looking at the last column of the above table, you obtain the following distribution of output x-ors:

0000	0001	0010	0011	0100	0101	0110	0111
0	0	8	0	0	2	0	2

1000	1001	1010	1011	1100	1101	1110	1111
0	0	0	0	0	2	0	2

3.2 Application of Differential Cryptanalysis

Differential cryptanalysis is applied to a chosen-plaintext attack. Since differential cryptanalysis involves comparing the x-or of two inputs x and x* to the x-or of the corresponding two outputs y and y*. This is based on the assumption that x, x*, y and y* are binary strings having a specified (fixed) x-or value denoted by $x' = x \otimes x^*$. It is also based on the assumption that possesses a large number of tuples (x, x*, y, y*). The same unknown key, K is used to encrypt the plaintext elements (i.e. x and x*) to give the ciphertexts y and y*. In order for attacker to carry out differential cryptanalysis, the ciphertexts y and y* will be decrypted for each of these tuples using all the possible candidate keys for the last round of the cipher. For each candidate key, the values of certain state bits are computed and determined if their x-or has the most likely value for the gin input x-or. Then, a candidate key counter is incremented whenever such happens. Finally, the candidate key with the highest frequency count contains the correct values for these key bits at the end of this iterative process.

4.0 Self-Assessment Exercise(s)

1. Differential Cryptanalysis can be mounted on
   A. DES encryption algorithm
   B. AES encryption algorithm
   C. RSA encryption algorithm
   D. Deffie-Hellman key exchange algorithm.

   Answer: A

2. Chosen cipher text attack is based on
   A. Encryption
   B. Cryptography
   C. Cryptanalysis
   D. Decryption

   Answer: C

5.0 Conclusion

In this unit, you have learnt why differential cryptanalysis was introduced, and also the approach to analyse the security of DES-like cryptosystems.

6.0 Summary

You have seen that differential cryptanalysis is a general form of cryptanalysis applicable to block ciphers, and also to stream ciphers and cryptographic hash functions. In the broadest sense, it is the study of how differences in information input can affect the resultant difference at the output. In the case of a block cipher, it refers to a set of techniques for tracing differences through the network of transformation, discovering where the cipher exhibits non-random behaviour and exploiting such properties to recover the secret key.

7.0 References/Further Reading

http://koclab.cs.ucsb.edu/teaching/cren/project/2005past/natarara.pdf

https://ioactive.com/differential-cryptanalysis-for-dummies/

Mini Project:
Write a research paper on the application of differential cryptanalysis

Unit 3: Data and Advanced Encryption Standard

Contents

1.0 Introduction

In this unit, you are going to learn the Data Encryption Standard (DES) as a secret key encryption scheme adopted as the standard in the USA in 1977. It uses a 56-bit key, which is today considered by many to be insufficient as it can with moderate effort be cracked by brute force. A variant called Triple-DES (TDES or 3DES) uses a longer key and is more secure, but has never become popular. The Advanced Encryption Standard (AES) is expected to supersede DES (and 3DES) as the standard encryption algorithm.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- interpret the general principles of a block cipher is applied in solving problems
- describe data and advanced encryption standards
- analyse data and advanced encryption standards
- compare and contrast the modes of operations of data and advanced encryption standards.

3.0 Main Content

3.1 Data Encryption Standard

The Data Encryption Standard (DES) was jointly developed in 1974 by IBM and the U.S. government (US patent 3,962,539) to set a standard that everyone could use to securely communicate with each other. It operates on blocks of 64 bits using a secret key that is 56 bits long. The original proposal used a secret key that was 64 bits long. It is widely believed that the removal of these 8 bits from the key was done to make it possible for U.S. government agencies to secretly crack messages.

DES started out as the "Lucifer" algorithm developed by IBM. The US National Security Agency (NSA) made several modifications, after which it was adopted as Federal Information Processing Standard (FIPS) standard 46-3 and ANSI standard X3.92.

3.1.1 Procedures of DES

Data Encryption Standard is a block cipher cryptosystem having a single key for encryption and decryption. Data Encryption Standard (DES) uses 64 bits block of data and 56 bits of the key. It involved 16 rounds of multiple encryptions. The algorithm of DES is depicted in Figure 1:

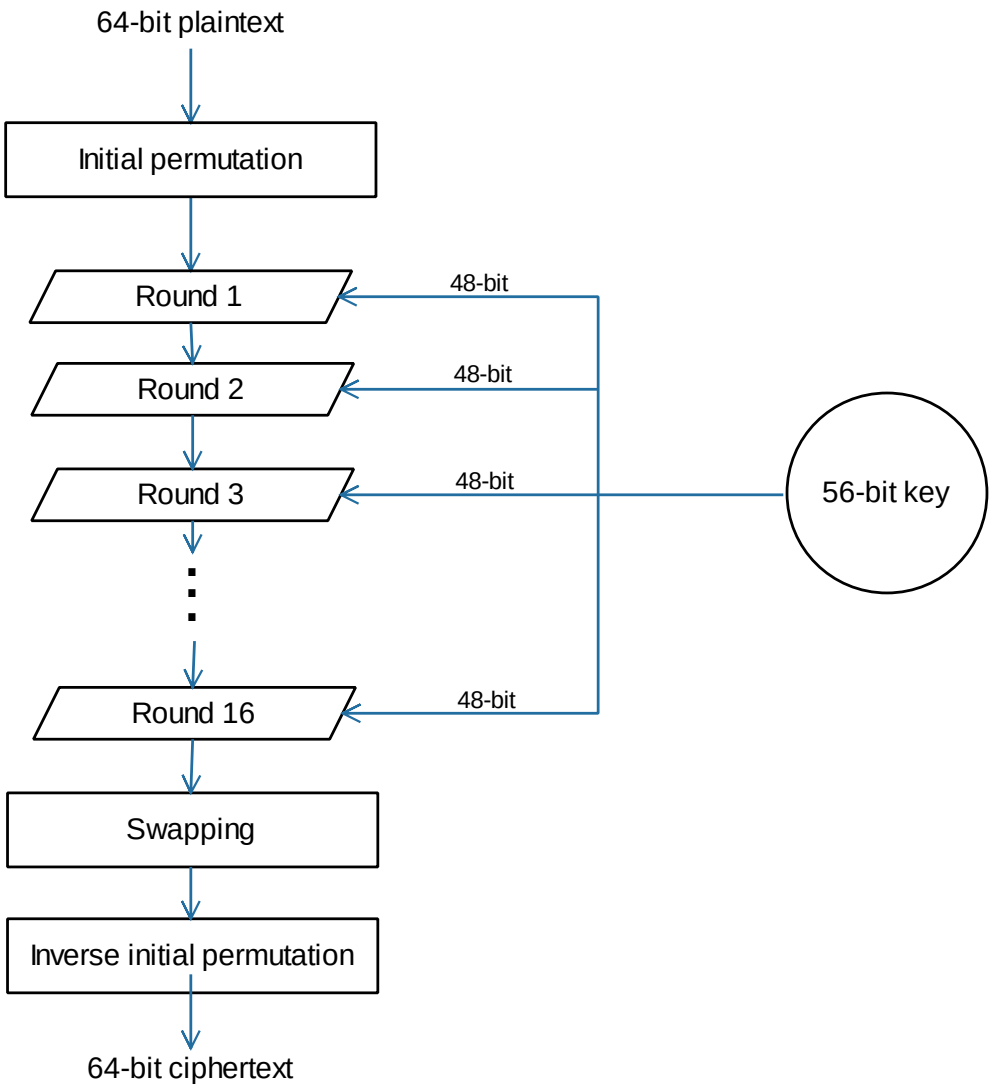


Fig. 7: Flowchart for DES

In DES, 64-bit plaintext is supplied into the cryptosystem after which the initial permutation is carried out. Then, 16-round application of 56-bit key even though the 48-bit key is actually used. After the completion of the 16 rounds key application, swapping is done. The permutation is then inversed to get 16-bit ciphertext, as shown in Figure 1.

3.1.2 Application of DES

The National Bureau of Standards (now known as the National Institute of Standards and Technology, or NIST) published a solicitation on May 15, 1973, for the need of cryptosystems in the Federal Register. This gave birth to wide adoption of the Data Encryption Standard in the world. DES was a modification of an earlier system known as Lucifer developed at IBM. On January 15, 1977, DES was adopted as a standard for "unclassified" applications. Its adoption was reviewed severally until the last review in January 1999 when the development of a replacement had

already begun. The proposed replacement was the Advanced Encryption Standard.

3.2 Advanced Encryption Standard

3.2.1 Procedures of AES

It is a symmetric key block cipher having 128-bit data. It was designed after DES. It is stronger and faster than DES. It is equally more secure than DES due to the larger-size.

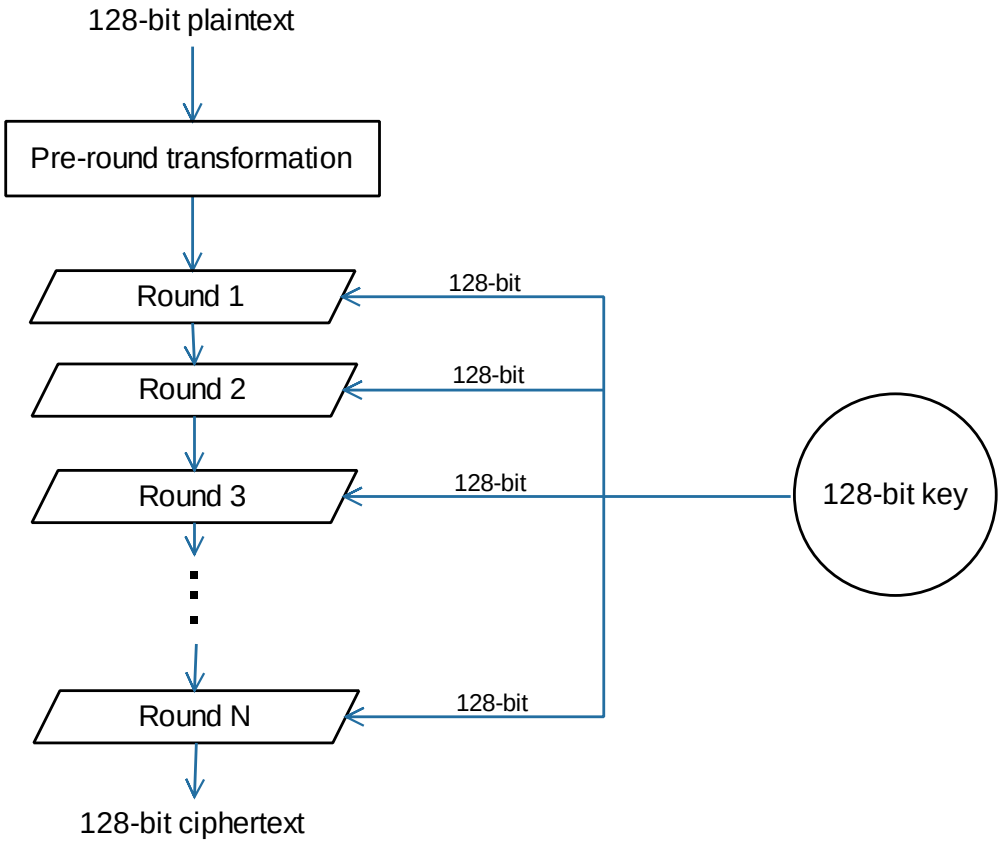


Fig. 8: Flowchart for AES

3.2.2 Application of AES

NIST started the formal process of choosing a replacement for DES on January 2, 1997. The replacement was to be called the Advanced Encryption Standard, or AES. Based on this, a formal call for algorithms was made on September 12, 1997. After the submissions and acceptance of submissions based on the relevant criteria, several conferences held for public reviews and comments. The selection process for the AES was generally accepted for its openness and international flavour. AES is applied to provide security against all known attacks. Also, some aspects of the design of AES has specific features that help in providing security against specific attacks. Taking an example of the use of the finite field inversion operation in the construction of the S-box yields linear approximation and difference distribution tables in which the entries are

close to uniform. This is applied to provide security against differential and linear attacks.

4.0 Self-Assessment Exercise(s)

1. Data Encryption Standard (DES), was designed by
   A. Intel
   B. IBM
   C. HP
   D. Sony
   Answer: B

2. Heart of Data Encryption Standard (DES), is the
   A. Cipher
   B. Rounds
   C. Encryption
   D. DES function
   Answer: D

5.0 Conclusion

In this unit you have learnt that the basic idea of differential cryptanalysis is first to cipher some plaintext, then make particular changes in that plaintext and cipher it again. Particular ciphertext differences occur more frequently with some key values than others, so when those differences occur, particular keys are (weakly) indicated.

6.0 Summary

In this unit, we have learnt how the differential attack can also be adapted for more than 2-rounds (and in the real world, it must be). The way to do this is by chaining differential characteristics together.

7.0 References/Further Reading

http://theamazingking.com/crypto-diff.php

https://ioactive.com/wp-ontent/uploads/2015/07/biham91differential.pdf

Module 3: Public Key Cryptography and Discrete Logarithm

Module Introduction

In this module, you will learn the concept of key cryptography and discrete logarithm function. By the time you complete the module, you will be able to explain the concept of key cryptography; discrete logarithm function; and the aim and outcome of public-key cryptography.

The module is organised into four units as follows:

Unit 1: ElGamal Cryptosystem
Unit 2: Algorithms for the Discrete Logarithm Problem
Unit 3: Elliptic Curves

Unit 1: ElGamal Cryptosystem

Contents

1.0 Introduction

In this unit you will learn about Elgamal cryptosystem, in addition, you will learn the basic concept ElGamal Cryptosystem in terms of Concepts and Principles; also you will learn the application areas of ElGamal Cryptosystem in Problem solving approach.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- illustrate public key cryptosystems such as ElGamal cryptosystems that is based on the Discrete Logarithm Problem
- describe ElGamal cryptosystem in terms of concepts and principles
- apply ElGamal cryptosystem in solving problems.

3.0 Main Content

3.1 Concept of Public-Key Cryptosystem

In the classical cryptography, you have learnt so far, the decryption rule or algorithm is the same with the encryption rule or algorithm or easily derived from it through the inverse procedure. This type of a cryptosystem is referred to as a symmetric-key cryptosystem since the exposure of either of encryption rule or decryption rule renders the system insecure. One major drawback of a symmetric-key system is that it requires the use of the secure channel for prior communication of the key K between the sender and the receiver before transmission of any ciphertext. Practical implementation of this is difficult, especially when the sender and receiver are far apart, and access to the reasonably secure channel is not guaranteed.

This limitation gave birth to the concept of a public-key cryptosystem. It is a cryptosystem where it is computationally impossible or infeasible for determination of decryption rule given the encryption rule. Then such encryption rule is referred to as a public key. One major benefit of this is that the sender can send an encrypted message to the receiver using the public encryption rule without prior communication of any shared secret key. It is only the receiver that can decrypt the ciphertext using the decryption rule, which is referred to as the private key.

The concept of a public-key cryptosystem was put forward in 1976 by Diffie and Hellman. The well-known cryptosystem was invented in 1977 by Rivest, Shamir and Adleman which they named after the acronym of their names as RSA Cryptosystem. Thereafter, several public-key systems have since been proposed whose security rests on different computational problems. There are variations of RSA Cryptosystem which security is based on the difficulty of factoring large integers. Another public-key cryptosystem is ElGamal Cryptosystem. It also has variations such as Elliptic Curve Cryptosystems. The security of these is based on the discrete logarithm problem.

3.2 History and Concept of ElgamalCryptosystem

It was proposed by an Egyptian Cryptographer called TaherElgamal in 1985. It is a public-key cryptosystem based on discrete logarithm problems.

Elgamal Cryptosystem is based on three phases referred to as Elgamal Protocol:

(1) The set-up phase: This is carried out by the party interested in receiving messages, i.e. the receiver.
(2) The encryption phase: This is carried out by the sender.
(3) The decryption phase: This is done by the receiver.

3.2.1 Set-up Phase of Elgamal Cryptosystem

The stages involved in the Set-up Phase are:

(1) The receiver chooses a large prime p (at least 1024 bits)
(2) The receiver then chooses $\alpha \in Z_p^*$
(3) The receiver chooses a random number which is a private key $b \in \{2, ..., p-2\}$
(4) The receiver computes the generator to the chosen random number $B = \alpha^b \bmod p$
(5) The public key of the receiver which has triple component is $k_{pub} = (p, \alpha, B)$

3.2.2 The Encryption Phase of Elgamal Cryptosystem

The stages in the encryption phase are:

(1) Given that the plaintext to be sent is denoted by x

To encrypt a message M to Bob, Alice first needs to obtain his public key triplet (p, g, g^b) from a key server or by receiving it from him via unencrypted electronic mail. There is no security issue involved in this transmission, as the only secret part, b, is sent in $g^{b.}$

Since the core assumption of the ElGamal cryptosystem says that it is infeasible to compute the discrete logarithm, this is safe.

For the encryption of the plaintext message M, Alice has to follow these steps:

1. Obtain the public key As described above, Alice has to acquire the public key part (p, g, g^b) of Bob from an official and trusted key server.
2. Prepare M for encoding Write M as set of integers (m1, m2, . . .) in the range of {1, . . . , p − 1}. These integers will be encoded one by one.
3. Select random exponent
   In this step, Alice will select a random exponent k that takes the place of the second party's private exponent in the Diffie-Hellman

key exchange. The randomness here is a crucial factor as the possibility to guess the k gives a sensible amount of the information necessary to decrypt the message to the attacker.

4. Compute public key
   To transmit the random exponent k to Bob, Alice computes g^k mod p and combines it with the ciphertext that shall be sent to Bob.
5. Encrypt the plaintext
   In this step, Alice encrypts the message M to the ciphertext C. For this, she iterates over the set created in step 2 and calculates for each of the m_i:

$c_i = m_{1*} (g^b)^k$

The ciphertext C is the set of all c_i with $0 < i \leq |M|$.
The resulting encrypted message C is sent to Bob together with the public key g^k mod p
derived from the random private exponent.

Even if an attacker would listen to this transmission, and in a second step would also acquire the public key part g^b of bob from a keyserver, he would still not be able to derive g^{b*k} as can be seen from the Discrete Logarithm problem.

Elgamal advises using a new random k for each of the single message blocks mi.

This greatly improves security, as knowledge of one message block m_j does not lead the attacker to the knowledge of all other mi.

The reason for this ability is that if
$c_1 = m_{1*} (g^b)^k$ mod p and $c_2 = m_{2*} (g^b)^k$ mod p, from knowing only m1 the next part of the message m2 can be calculated by the following formula:

$$\frac{m_1}{m_2} = \frac{c_1}{c_2}$$

3.2.3 The Decryption Phase of Elgamal Cryptosystem

Informal Procedures of the Working of ElGamal Cryptosystem
Informally, the workings of ElGamal Cryptosystem follow these procedures: Given a plaintext x, it is 'masked' by multiplying it by β^k, yielding y_2. The value α^k is also transmitted as part of the ciphertext. The receiver, who knows the private key, a, can compute β^k from α^k. Then he can 'remove the mask' by dividing y_2 by β^k to obtain x.

Solved Example:
Suppose p =2579 and α = 2. α is a primitive element modulo p. Let α =765, so

$\beta = 2^{765} \bmod 2579 = 949$

Now, suppose that Alice wishes to send the message x = 1299 to Joseph. Let say k = 853 is the random integer she chooses.

Solution:
She computes the following:
$y_1 = 2^{853} \bmod 2579$
$= 435$
and
$y_2 = 1299 \times 949^{853} \bmod 2579$
$= 2396$

When the receiver Joseph receives the ciphertext y = (435, 2396), he computes
$x = 2396 \times (435^{765})^{-1} \bmod 2579$
$= 1299$
which was the plaintext that Alice encrypted.

3.3 Application of Elgamal Cryptosystem

Like most public-key systems, the ElGamal cryptosystem is usually used as part of a hybrid cryptosystem where the message itself is encrypted using a symmetric cryptosystem and ElGamal is then used to encrypt only the symmetric key. This is because asymmetric cryptosystems like ElGamal are usually slower than symmetric ones for the same level of security, so it is faster to encrypt the message, which can be arbitrarily large, with a symmetric cipher, and then use ElGamal only to encrypt the symmetric key, which usually is quite small compared to the size of the message

4.0 Self-Assessment Exercise(s)

1. Elgamal Cryptosystem is based on three phases referred to as Elgamal Protocol they are:

   Answer
   1. The set-up phase: this is carried out by the party interested in receiving messages, i.e. the receiver
   2. The encryption phase: This is carried out by the sender
   3. The decryption phase: This is done by the receiver.

2. Why is the ElGamal cryptosystem usually used as part of a hybrid cryptosystem where the message itself is encrypted using a

symmetric cryptosystem and ElGamal is then used to encrypt only the symmetric key?

Answer
This is because asymmetric cryptosystems like ElGamal are usually slower than symmetric ones for the same level of security, so it is faster to encrypt the message, which can be arbitrarily large, with a symmetric cipher, and then use ElGamal only to encrypt the symmetric key, which usually is quite small compared to the size of the message.

5.0 Conclusion

You have learned from this unit that messages which are encrypted using a user's public key can only be decrypted using that user's private key. Cryptography following this paradigm is referred to as public-key cryptography, the "old" alternative being private key cryptography (the respective alternate names asymmetric and symmetric cryptography are also used).

6.0 Summary

We can deduct from this unit that using public-key cryptography, Alice and Bob need only exchange their public keys, without concern as to eavesdropping, before they can communicate privately. This allows secure communication between parties with no prior acquaintance or communication and without the need for a secure means of key transfer.

7.0 References/Further Reading

https://www.luke.maurits.id.au/files/misc/honours_thesis.pdf

https://wwwmayr.in.tum.de/konferenzen/Jass05/courses/1/papers/meier_paper.pdf

Unit 2: Algorithms for the Discrete Logarithm Problem

Contents

1.0 Introduction

In this unit, you will learn about the algorithm for discrete logarithm problem. Discrete logarithms are quickly computable in a few special cases. However, no efficient method is known for computing them in general. Several important algorithms in public-key cryptography base their security on the assumption that the discrete logarithm problem over carefully chosen groups has no efficient solution.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe the concepts of discrete logarithm problem
- analyse the algorithms for solving discrete logarithm problems
- solve the discrete logarithm problem
- apply discrete logarithm problem to real-life situation.

3.0 Main Content

3.1 Analysis of Algorithms for the Discrete Logarithm Problem

Let us assume that a multiplicative group is denoted by (G, ▪) and $\alpha \in G$ Has order n. Then, the discrete logarithm problem can be phrased as follows: Given $\beta \in \langle \alpha \rangle$, find the unique exponent ɑ, 0 ≤ɑ≤ n – 1, such that $\alpha^{a} = \beta$.

By analysing some elementary algorithms for solving Discrete Logarithm Problems (DLP), let us assume that computing a product of two elements in the group G requires constant (i.e., O(1)) time.

First approach:
DLP can be solved by exhaustive search in O(n) time and O(1) space by computing $\alpha, \alpha^2, \alpha^3, \ldots,$ until $\beta = \alpha^a$ is found. The computation of α^i requires a total time O(n).

Second approach:
By precomputing all possible values α^i, and then sort the list of ordered pairs (i, α^i) with respect to their second coordinates. If given β, then a binary search of the sorted list can be performed to find the value α such that $\alpha^a = \beta$

A precomputation time of O(n) is required to compute the n powers of α and time O(n log n) to sort the list of size n.

If the logarithmic factors are neglected, then the precomputation time is O(n).

The time for a binary search of a sorted list of size n is O(log n). By ignoring the logarithmic factors again then the DLP can be solved in O(1) time with O(n) precomputation and O(n) memory.

3.2 Algorithms for the Discrete Logarithm Problem

The four popular algorithms for solving the DLP are:
(1) Shanks' algorithm
(2) The Pollard Rho Discrete Logarithm Algorithm
(3) The Pohlig-Hellman Algorithm
(4) The Index Calculus Method.

3.2.1 Shanks' Algorithm

Shanks' algorithm is the first non-trivial algorithm based on a time-memory trade-off due to Shanks. The algorithm is presented as follows:

Algorithm: SHANKS (G, n, α, β)
1. $m \leftarrow \lfloor \sqrt{n} \rfloor$
2. for $j \leftarrow 0 \; to \; m-1$
   do compute α^{mj}
3. Sort the m ordered pairs (j, α^{mj}) with respect to their second coordinates, obtaining a list L_1
4. For i ← 0 to m – 1
   do compute $\beta\alpha^{-i}$
5. Sort the m ordered pair (i, $\beta\alpha^{-i}$) with respect to their second coordinates, obtaining a list L_2
6. Find a pair (j, y) $\in L_1$ and a pair (i, y) $\in L_2$ (i.e. find two pairs having identical second coordinates)
7. $\log_\alpha \beta \leftarrow (mj + i) \bmod n$

3.2.2 The Pollard Rho Discrete Logarithm Algorithm

The Pollard Rho Discrete Logarithm Algorithm is for factoring in which you seek for a collision of the form $x_i = x_{2i}$ in order to save time and memory. The algorithm is given as follows:

Algorithm: *POLLARD RHO DISCRETE LOG ALGORITHM (G, n, α, β)*

```
1    procedure f(x, a, b)
2        if  x ∈ S1
3        then f ← (β · x, a, (b + 1) mod n)
4        else if x ∈ S2
5        then  f ← (x^2, 2a mod n, 2b mod n)
6        else f ← ( α · x, (a + 1) mod n, b)
7        Return (f)
8
9    Main
10       define the partition G = S1 U S2 U S3
11       (x, a, b) ← f(1, 0, 0)
12       (x', a', b') ← f (x, a, b)
13       while x ≠ x'
14                (x,a,b) ← f(x,a,b)
15       do       (x', a', b') ← f (x', a', b')
16                (x', a', b') ← f (x', a', b')
17       If gcd (b' − b, n) ≠ 1
18       then return ("failure")
19       else return ((a − a')(b' − b)^−1)
```

3.2.3 The Pohlig-Hellman Algorithm

The Pohlig-Hellman Algorithm talks about distinct primes such that

$$n = \prod_{i=1}^{k} p_i^{c_i}$$

Where the p_i's are distinct primes.

The pseudo-code description of the algorithm is given as follows. To summarise the operation of this algorithm, α is an element of order n in a multiplicative group G, q is prime.

Algorithm:*ThePohlig-Hellman Algorithm(G, n, α, β ,q,c)*

```
1  j ← 0
2  β_j ← β
3  while j ≤ c − 1
4        δ ← β_j^(n/q^(J+1))
5        find i such that δ = α^(in/q)
6  do    a_j ← i
7        β_(j+1) ← β_j α^(−a,q^j)
8        j ← j + 1
9  Return (a0, …, ac-1)
```

3.2.4 The Index Calculus Method

The previous algorithms can be applied to any group, while the Index Calculus Method is more specialised. It is applicable in a particular situation of finding discrete logarithms in Z_p^* when p is prime and α is a primitive element modulo p. For this reason, the Index Calculus Algorithm is faster than the algorithms previously considered.

The Index Calculus Algorithm for computing discrete logarithms uses a factor base method and very much bears a high resemblance to many of the best factoring algorithms.

The steps involved in the algorithm are:
First step: It is a preprocessing step to find the logarithms of the B primes in the factor base.

Second step: It is the computation of the discrete logarithm of the desired element β, using the knowledge of the discrete logarithms of the elements in the factor base.

Portfolio:
Outline the algorithm for the index calculus method

4.0 Self-Assessment Exercise(s)

1. There are four most popular algorithms for solving the Discrete Logarithm
   Problem. Name them.

   Answer
   (1) Shanks' algorithm
   (2) The Pollard Rho Discrete Logarithm Algorithm
   (3) The Pohlig-Hellman Algorithm
   (4) The Index Calculus Method.

2. Shanks' algorithm is the first non-trivial algorithm based on a time-memory trade-off due to Shanks.
   A. True.
   B. False.
   Answer: A

5.0 Conclusion

Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups. If G is a multiplicative cyclic group and g is a generator of G, then from the definition of cyclic groups, we know every element h in G can be written as g^x for some x.

6.0 Summary

The discrete logarithm problem is defined as given a group G, a generator g of the group and an element h of G, to find the discrete logarithm to the base g of h in the group G. Discrete logarithm problem is not always hard. The hardness of finding discrete logarithms depends on the groups.

7.0 References/Further Reading

https://www.sciencedirect.com/topics/computer-science/discrete-logarithms

https://math.dartmouth.edu/~carlp/dltalk09.pdf

Unit 3: Elliptic Curves

Contents

1.0 Introduction

In this unit, you will learn the concepts and theories of elliptic curve cryptography as well as the properties of elliptic curves cryptography.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe the concepts and theories of Elliptic Curves
- examine the properties of Elliptic Curves
- compute Discrete Logarithms, Elliptic Curves and Diffie – Hellman Problems.

3.0 Main Content

3.1 Concepts and theories of Elliptic Curves

Elliptic curves are described by the set of solutions to certain equations in two variables. In public-key cryptography, Elliptic curves defined modulo a prime p are of major importance. Elliptic curves can be defined over the Reals and over Modulo a Prime.

To define Elliptic curves over the Reals, let $a,b \in R$ be constants such that $4a^3 + 27b^2 \neq 0$. A non-singular elliptic curve is the set E of solutions (x, y) $\in$ R x R to the equation

$$y^2 = x^3 + ax + b$$

together with a special point $\emptyset$ called the point of infinity.

To define Elliptic curves over Modulo a Prime (Z_p), the same process as its definition over the Reals can be followed provided that all operations over R are replaced by analogous operations in Z_p given that p > 3.
Formally, given p > 3, the elliptic curve $y^2 = x^3 + ax + b$ over Z_p is the set of solutions (x, y) $\in Z_p \times Z_p$ to the congruence
$y^2 \equiv x^3 + ax + b \pmod p$,
where a, b $\in Z_p$ are constants such that $4a^3 + 27b^2 \not\equiv 0 \pmod p$, together with a special $\wp$ called the point at infinity.

3.2 Properties of Elliptic Curves

These are some properties of Elliptic curves:
(1) An elliptic curve E defined over Z_p with p > 3 has roughly p points on it.
(2) It also satisfies the following inequality: $p+1-2\sqrt{p} \leq \#E \leq p+1+2\sqrt{p}$
(3) Genetic algorithms are applicable to the elliptic curve discrete logarithm problem, but there is no known adaptation of the index calculus algorithm to the setting of elliptic curves.
(4) A class of weak elliptic curves are the called curves of "trace one." These are elliptic curves defined over Z_p (where p is prime) having exactly p points on them.

4.0 Self-Assessment Exercise(s)

1. "Elliptic curve cryptography follows the associative property."
   A. True
   B. False
   Answer: A

2. In Singular elliptic curve, the equation x^3+ax+b=0 does _____ roots.
   A. does not have three distinct
   B. has three distinct
   C. has three unique
   D. has three distinct unique.
   Answer: A

5.0 Conclusion

Elliptic curve systems base their difficulty on the elliptic curve version of the DLP, which is simply called the Elliptic Curve Discrete Logarithm

Problem (ECDLP). Here, the underlying field of integers modulo prime p is replaced by points on an elliptic curve defined over a finite field. Since the ECDLP is significantly harder than the DLP, even a sophisticated hacker would require most of the world's computing power for a few years to break an elliptic curve cryptosystem

6.0 Summary

Elliptic curves are applicable for key agreement, digital signatures, pseudo-random generators and other tasks. Indirectly, they can be used for encryption by combining the key agreement with asymmetric encryption scheme

7.0 References/Further Reading

https://unacademy.com/lesson/elliptic-curve-cryptography-ecc/KVHWGMHN

https://scialert.net/fulltextmobile/?doi=jas.2005.604.633

Module 4: Private Key Encryption

Module Introduction

In this module, you will build on previous knowledge, and also, you will learn private key encryption.

The module is organised into two units as follows:
Unit 1: Symmetric Encryption Scheme
Unit 2: Issues in Privacy

Unit 1: Symmetric Encryption Scheme

Contents

1.0 Introduction

In this unit, we will learn base on previous knowledge the concept of symmetric encryption schemes.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- analyse the concepts and principles of Symmetric Encryption Scheme
- compare and contrast Symmetric Encryption and Public Key Cryptography.

3.0 Main Content

3.1 Definition of Symmetric Encryption Scheme

Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the development of public-key encryption in the 1970s. It remains by far the most widely used of the two types of encryption. An original message is known as the **plaintext**, while the coded message is called the **ciphertext**. The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**; restoring the plaintext from the ciphertext is **deciphering** or **decryption**. The many schemes used for encryption constitute the area of study known as **cryptography**. Such a scheme is known as a **cryptographic system** or a **cipher**.

Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**. Cryptanalysis is what the layperson calls "breaking the code." The areas of cryptography and cryptanalysis together are called **cryptology**. Figure 1 below shows the concept of the symmetric encryption scheme.

3.2 Ingredients of a Symmetric Encryption Scheme

There are five ingredients in asymmetric encryption scheme:
Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.

Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.

Secret key: The secret key is also input to the encryption algorithm. The key is value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.

Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

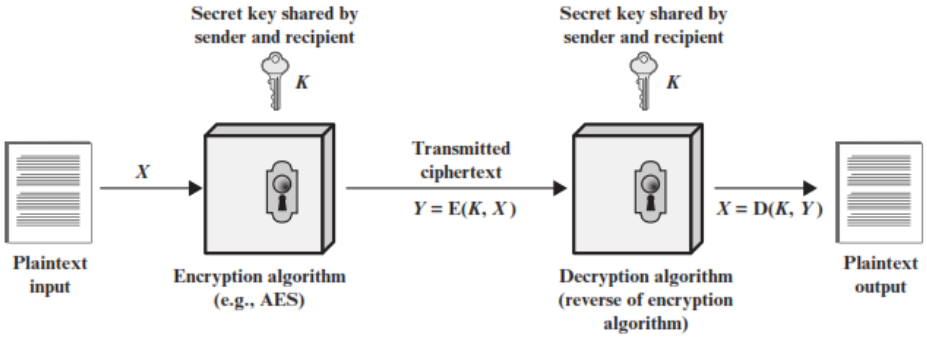


Fig. 9: A Simple Model of Symmetric Encryption

There are two requirements for the secure use of conventional encryption: We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or moreciphertexts would be unable to decipher the ciphertext or figure out the key.

This requirement is usually stated in a stronger form: The opponent should is unable to decrypt ciphertext or discover the key even if he or she is in possession of many ciphertexts together with the plaintext that produced each ciphertext. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

We assume that it is impractical to decrypt a message based on the ciphertext *plus* knowledge of the encryption/decryption algorithm. In other words, we do not need to keep the algorithm secret; we need to keep only the key secret.

This feature of symmetric encryption is what makes it feasible for widespread use. The fact that the algorithm need not be secret means that manufacturers can and have developed low-cost chip implementations of data encryption algorithms. These chips are widely available and incorporated into several products. With the use of symmetric encryption, the principal security problem is maintaining the secrecy of the key.

4.0 Self-Assessment Exercise(s)

1. Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of
   A. Cryptography
   B. Cryptology
   C. Cryptanalysis
   D. CryptoSystem
   Answer: C

2. _____ the encryption algorithm performs various substitutions and transformations on the plaintext.
   A. Decryption Algorithm
   B. Encryption Algorithm
   C. Substitution Algorithm
   D. Transformation Algorithm
   Answer: B

5.0 Conclusion

The security of symmetric encryption systems is based on how difficult it randomly guess the corresponding key to brute force them. A 128-bit key, for example, would take billions of years to guess using common computer hardware. The longer the encryption key, the harder it becomes to crack it. Keys that are 256-bits length are generally regarded as highly secure and theoretically resistant to quantum computer brute force attacks.

6.0 Summary

If the encryption scheme is strong enough, the only way for a person to read or access the information contained in the ciphertext is by using the corresponding key to decrypt it. The process of decryption is basically converting the ciphertext back to plaintext.

7.0 References/Further Reading

https://www.binance.vision/security/what-is-symmetric-key-cryptography

https://www.researchgate.net/publication/317426657_Symmetric_Key_Algorithms_A_Comparative_Analysis

Unit 2: Issues in Privacy

Contents

1.0 Introduction

In this unit, you will learn about privacy issues in a cryptographic system, even though there is plenty of advantage, a cryptosystem still faces some privacy issue.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain issues related to privacy, trust and security.
- discuss the ethics and law of data protection.

3.0 Main Content

3.1 Security, Privacy and Trust

Security, privacy and trust are inter-related. Security primarily sets up control measures to guide against breaches to information in terms of confidentiality, integrity and availability. Privacy preservation, on the other hands focuses on the protection of data or regulative measures on data or information in terms of accessibility, collection, use, inspection, update or modification. Trust is a relationship in which a party known as trustor believes in the second party (trustee) that he is good, honest, safe, reliable and unharmful.

Security measures and privacy preservation depend on trust. One does not feel secure if the security providers are not trusted. Equally, privacy measures will not be welcomed and embraced if preservation mechanisms or providers are not trusted to enter into the inaccessibility zone.

In a more specific term, privacy issues involve the right of a person to keep his personal data or information from other people. Such a person also has a right to view his data at any time. Also, data privacy is concerned with the costs or claim of compensation on the event of a breach of privacy with respect to relevant law, ethical principles and societal norms.

3.2 Privacy preservation and data protection: Ethics and law

In order to safeguard the privacy of data and information, there are ethics and laws for this purpose. The law provides a resolution when ethics cannot (e.g., ethics knows that stealing is wrong; the law punishes thieves); ethics can provide context to the law (e.g., the law allows trading for the purpose of making a profit, but ethics provides input into ensuring trade is conducted fairly). Privacy breaches disturb trust and run the risk of diluting or losing security; it is a show of disrespect to the law and a violation of ethical principles.

3.3 Data privacy: Protection and principles

3.3.1 Data Privacy Protection

Indeed, protecting data privacy is urgent and complex. This protection is necessary because of the ubiquity of the technology-driven and information-intensive environment. Technology-driven and information-intensive business operations are typical in contemporary corporations. The benefits of this trend are that, among other things, the marketplace is more transparent, consumers are better informed, and trade practices are fairer. The downsides include socio-techno risk, which originates with technology and human users (e.g., identity theft, information warfare, phishing scams, cyberterrorism, extortion), and the creation of more opportunities for organized and sophisticated cybercriminals to exploit. This risk results in information protection being propelled to the top of the corporate management agenda.

The need for data privacy protection is also urgent due to multidirectional demand. Information protection becomes an essential information security function to help develop and implement strategies to ensure that data privacy policies, standards, guidelines and processes are appropriately enhanced, communicated and complied with, and effective mitigation measures are implemented. The policies or standards need to

be technically efficient, economically/financially sound, legally justifiable, ethically consistent and socially acceptable since many of the problems commonly found after implementation and contract signing are of a technical and ethical nature, and information security decisions become more complex and difficult.

Data privacy protection is complex due to socio-techno risk, a new security concern. This risk occurs with the abuse of technology that is used to store and process data. For example, taking a company universal serial bus (USB) device home for personal convenience runs the risk of breaching a company regulation that no company property shall leave company premises without permission. That risk becomes a data risk if the USB contains confidential corporate data (e.g., data about the marketing strategy, personnel performance records) or employee data (e.g., employee addresses, dates of birth). The risk of taking the USB also includes theft or loss.

Using technology in a manner that is not consistent with ethical principles creates ethical risk, another new type of risk. In the previous example, not every staff member would take the company USB home, and those who decide to exploit the risk of taking the USB may do so based on their own sense of morality and understanding of ethical principles. The ethical risk (in addition to technical risk and financial risk) arises when considering the potential breach of corporate and personal confidentiality. This risk is related partly to technology (the USB) and partly to people (both the perpetrator and the victims) and is, therefore, a risk of a technological-cum-social nature - a socio-techno risk. Hence, taking home a USB is a vulnerability that may lead to a violation of data privacy.

However, the problem of data privacy is not unsolvable. The composite approach alluded to earlier that takes into consideration the tangible physical and financial conditions and intangible measures against logical loopholes, ethical violations, and social desirability is feasible. The method suggested in this article, which is built on a six-factor framework, can accomplish this objective.

3.3.2 Methods for Data Privacy Protection

The method is modelled on a framework initially perceived and developed to provide a fresh view to decision-makers and is based on the following three major instruments:

- The International Data Privacy Principles (IDPPs)[1] for establishing and maintaining data privacy policies, operating standards and mitigation measures
- Hong Kong's Data Protection Principles of personal data (DPPs)[2] for reinforcing those policies, standards and guidelines
- The hexa-dimension metric operationalization framework[3] for executing policies, standards and guidelines

3.3.3 International Data Privacy Principles

Data privacy can be achieved through technical and social solutions. Technical solutions include safeguarding data from unauthorized or accidental access or loss. Social solutions include creating acceptability awareness among customers about whether and how their data are being used and doing so transparently and confidentially. Employees must commit to complying with corporate privacy rules, and organisations should instruct them on how to avoid activities that may compromise privacy actively.

Next to technical and social solutions, the third element of achieving privacy is complying with data protection laws and regulations, which involves two issues. The first concern is that legal regulation is slow and, thus, unable to keep up with the rapid developments of information technology. Legal solutions are usually at least one step behind technological developments. Data privacy by electronic means should, therefore, be based not only on traditional jurisdiction but also on soft law, i.e., self-binding policies such as the existing data privacy principles. Soft law may be more effective than hard law. The reactions of disappointed customers, especially when those reactions are spread by social media, and the fact that noncompliance with corporate governance may result in unfair competition and/or liability toward affected customers (unfair competition by not complying with self-binding policies/liability toward customers by breach of contract) will often be more effective than mere fines or penalties.

The second problem of data protection has to do with the fact that these regulations are not internationally harmonized, causing severe complications (especially between the United States and the European Union) on a cross-border basis, which is the rule rather than the exception in modern business. To make data privacy rules work in a global environment, the principles outlined in this article consider US standards (e.g., the US Federal Trade Commission's Fair Information Practices), European standards (e.g., Data Protection Directive 95/46/EC and the General Data Protection Regulation [GDPR]), Asian regulations (e.g., Hong Kong Personal Data Privacy Ordinance [PDPO]) and international benchmarks (e.g., the Organisation for Economic Co-operation and Development [OECD] Privacy Framework Basic Principles).
This article also considers the fact that common data privacy regulations, especially in Europe, tend to focus on a traditional human rights approach, neglecting the fact that nowadays, data are usually given away voluntarily upon contractual agreement. When using sites such as Google, Baidu, Amazon, Alibaba or Facebook, users agree with the terms and conditions of these companies. Data privacy should consider not only mere data protection but also contractual principles, among which one of the oldest and most fundamental is **do ut des**, meaning a contract in

which there is a certain balance between what is given and what is received. That philosophy explains why companies such as Google or Facebook, for whose services the customer does not pay, have the right to use personal data. In other words, that tradeoff—data for services—is the balance.[4]

The consumer is less protected when receiving free services is a basic element of the European E-Commerce Directive, which does not apply to services that are offered free of charge. But this consideration is only a first step. In a modern data environment, a balance also must be struck concerning other parameters relevant to contractual aspects of data privacy. Since data are a contract matter, it is important to consider what kind of personal data is in consideration (e.g., sensitive and nonsensitive data have to be distinguished and treated differently). Since contracts are concluded by mutual consent, the extent of such consent also has to be taken into account. For example, does consent have to be declared explicitly or is accepting the terms of use sufficient?

The IDPPs approach takes into consideration the Asian, European, US and international data protection standards and focuses on personal data, but can apply to corporate data as well. These principles suggest that the three parameters (payment, consent and data category) should be balanced and combined with the previously mentioned, Asian, European, US and international standards, putting them into a set of privacy rules. Organisations in compliance with international data privacy standards should commit to the following 13 IDPPs:[5]

1. Comply with national data protection or privacy law, national contract law, and other legal requirements or regulations relating to data privacy.
2. Comply with current security standards to protect stored personal data from illegitimate or unauthorised access or from accidental access, processing, erasure, loss or use.
3. Implement an easily perceptible, accessible and comprehensible privacy policy with information on who is in charge of data privacy and how this person can be individually contacted, why and which personal data are collected, how these data are used, who will receive these data, how long these data are stored, and whether and which data will be deleted or rectified upon request.
4. Instruct employees to comply with such privacy policies and avoid activities that enable or facilitate illegitimate or unauthorized access in terms of IDPPs.
5. Do not use or divulge any customer data (except for statistical analysis and when the customer’s identity remains anonymous), unless the company is obliged to do so by law or the customer agrees to such use or circulation.

6. Do not collect customer data if such collection is unnecessary or excessive.
7. Use or divulge customer data in a fair way and only for a purpose related to the activities of the company.
8. Do not outsource customer data to third parties unless they also comply with standards comparable to these IDPPs.
9. Announce data breaches relating to sensitive data.
10. Do not keep personal data for longer than necessary.
11. Do not transfer personal data to countries with inadequate or unknown data protection standards unless the customer is informed about these standards being inadequate or unknown and agrees to such a transfer.
12. In the case of a contract between the company and the customer in which the customer commits to pay for services or goods:
    - Inform the customer individually and as soon as reasonably possible in the event of a data breach.
    - Inform the customer upon request about which specific data are stored, and delete such data upon request unless applicable laws or regulations require the company to continue storing such data.
    - Do not use or divulge content-related personal data.
    - Do not use or divulge any other personal data without the customer's explicit, separate and individual consent.
    - Do not store, use or divulge any customer data, unless applicable laws or regulations require the company to continue storing such data.
13. In the absence of a contract between the company and the customer in which the customer commits to pay for services or goods:
    - Inform the customer as soon as reasonably possible in the event of data breaches.
    - Inform the customer upon request what types of sensitive data are stored and delete such data upon request when such data are outdated unless applicable laws or regulations require the company to continue storing such data.
    - Do not use or divulge sensitive data without the customer's explicit, separate and individual consent.

4.0 Self-Assessment Exercise(s)

1. Security measures and privacy preservation depend on trust. Explain.

   Answer:
   One does not feel secure if the security providers are not trusted. Equally, privacy measures will not be welcomed and embraced if preservation mechanisms or providers are not trusted to enter into the inaccessibility zone.

2. Data privacy can be achieved through _____and _____ solutions.{Select two}
   A. Technical solutions
   B. Operational solutions
   C. Social solutions
   D. Media solutions
   Answer: A, C

5.0 Conclusion

Data privacy protection is complex due to socio-techno risk, a new security concern. This risk occurs with the abuse of technology that is used to store and process data. For example, taking a company universal serial bus (USB) device home for personal convenience runs the risk of breaching a company regulation that no company property shall leave company premises without permission.

6.0 Summary

The problem of data privacy is not unsolvable. The composite approach alluded to earlier that takes into consideration the tangible physical and financial conditions and intangible measures against logical loopholes, ethical violations, and social desirability is feasible.

7.0 Further Readings

https://medium.com/@the_manifest/data-privacy-concerns-an-overview-for-2019-2ccea79aa6f8

https://www.sciencedirect.com/topics/computer-science/privacy-problem