CST903: ADVANCED CRYPTOGRAPHY

AFRICA CENTRE OF EXCELLENCE ON TECHNOLOGY ENHANCED LEARNING (ACETEL)

NATIONAL OPEN UNIVERSITY OF NIGERIA

Course Guide for CST903

Introduction

CST903 – 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:

On the various research in the design and implementation of security and its applications

Course Objective

The course objective is to:

Protect data at rest and during transmission in systems and network

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 Symmetric Cryptography

Unit 1 Symmetric Encryption Scheme
Unit 2 Indistinguishability Under Chosen Plaintext Attack
Unit 3 Indistinguishability Under Chosen Cipher Text Attack

Module 2 Asymmetric Cryptography

Unit 1 Asymmetric Encryption Scheme
Unit 2 Problem with Deterministic Encryption
Unit 3 RSA Cryptosystem
Unit 4 Probabilistic Public Key Encryption

Module 3 Algorithms of Modern Cipher

Unit 1 AES, DES and RSA
Unit 2 RC4 and Key Exchange Management

Module 4 Signature Scheme

Unit 1 Security Requirement for Signature Scheme
Unit 2 Digital Signature Algorithm
Unit 3 Secure Hash
Unit 4 Steganography

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 2 Unit 1
6	Module 2 Unit 2
7	Module 2 Unit 3
8	Module 2 Unit 4
9	Module 3 Units 1 and 2
10	Module 4 Unit 1
11	Module 4 Unit 2
12	Module 4 Unit 3
13	Module 4 Unit 4
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	20
3	Laboratory Practical	20
4	Assignments	10
5	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.

Laboratory Practical

The laboratory practical may be virtual or face-to-face or both depending on the nature of the activity. You will receive further guidance from your facilitator.

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 903
Course Title:	ADVANCED CRYPTOGRAPHY
Credit Unit:	3
Course Status:	Compulsory
Course Blurb:	This covers symmetric and asymmetric cryptography including the history of cryptography and cryptanalysis, algorithms for modern ciphers such as AES, DES, RSA, and RC4, key exchange and management, digital signatures, secure hashes, as well as steganography
Semester:	First
Course Duration:	13 weeks
Required Hours for Study:	91

Course Team

Course Developer:	ACETEL
Course Writers:	Dr Olawale Surajudeen Adebayo & Dr Ahmad Aliyu
Content Editor:	Dr Ismaila Idris
Instructional Designers:	Inegbedion, Juliet O. (PhD) and Dr Lukuman Bello
Learning Technologists:	Dr Adewale Adesina and Mr Miracle David
Graphic Artist:	Mr Henry Udeh
Proofreader:	Mr Awe Olaniyan Joseph

Module 1: Symmetric Cryptography

Unit 1 Symmetric Encryption Scheme
Unit 2 Indistinguishability under Chosen Plaintext Attack
Unit 3 Indistinguishability under Chosen Ciphertext Attack

Unit 1 Symmetric Encryption Scheme

Contents

1.0 Introduction

In this unit, you will learn the nature and structure of communicating data, together with their security features. The communicating data could be plaintext or ciphertext. The data or message or word in original or intelligible form is called plaintext. This data could be hijacked, attacked, or damage over an insecure communication channel. The ciphertext, on the other hand, is the encrypted data that has been transformed into unintelligible form to prevent an attacker from comprehending the contents.

In the unit, the symmetric cryptography is discussed. This is an encryption technique where one key called private key is used for message encryption and decryption. The two basic types of symmetric cipher will be discussed. These are substitution and transposition ciphers. Substitution cipher like shift cipher, affine cipher, vigenere cipher, and hill cipher shall be discussed. The rail fence technique of transposition technique shall also be examined. Finally, the privacy and authenticity of communicating data over an insecure communication channel shall be examined and the best ways to ensure the security of data.

2.0 Intended Learning Outcomes (ILOs)

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

- describe symmetric encryption
- explain the security of symmetric cryptography
- identify communicating data with the various security attributes
- discuss the privacy and authenticity of the communicated data.

3.0 Main Content

3.1 Symmetric Encryption (Substitution and Transposition)

This is single-key cryptography used before the development of public-key cryptography, also known as asymmetric cryptography. Most of the symmetric ciphers are block-oriented in operation. That is, the data is being executed in bit by bit of 8 bits or block at a time. It means the data will store in the hard disk or other memory before the execution of another batch. This led to the security vulnerability of block cipher cryptosystems. Symmetric encryption is characterised with the use of a single private key for encryption and decryption. Symmetric encryption consists of transposition techniques, substitution techniques, rotor-machines, and other techniques. All these forms of symmetric encryption are called classical encryptions. In this encryption, the decryption algorithm is the reverse of the encryption algorithm. That is, different computations technique is used in the encryption and decryption algorithms, unlike asymmetric, which use the same computations. The most commonly used symmetric ciphers are the Advanced Encryption Standard (AES) and Data Encryption Standard (DES).

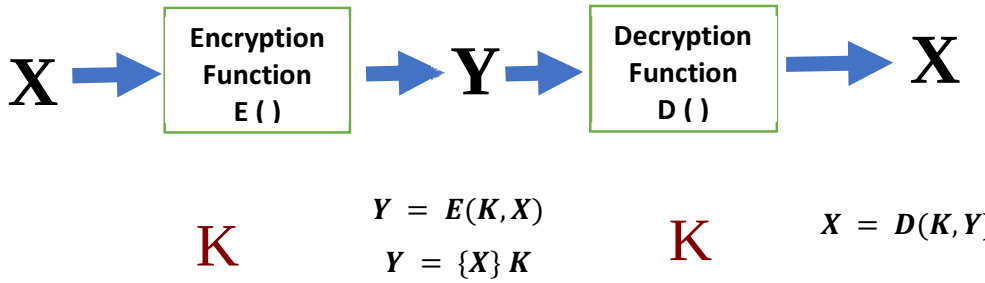

Fig. 1.1: Symmetric Cryptography Models

3.1.1Symmetric Cipher Model

Fig. 1.1 is a Symmetric encryption model having a five-tuple scheme with plaintext, encryption algorithm, secret key, decryption algorithm, and ciphertext. These tuples are defined as follows:

1. **Plaintext** or **Cleartext:**
   This is the original message that is unintelligible in nature.
2. **Encryption / Cipher**
   This is a technique or algorithm used to transform or encode plaintext to ciphertext.
3. **Decryption** / **Decipher**
   This is a technique or algorithm used to revert or decode ciphertext into plaintext.
4. **Key**
   An input variable or parameter for the cipher/decipher, encoding /decoding or encryption/decryption algorithms.
5. **Secret key or secret shared key**
   This is an input parameter used in symmetric encryption
6. **Encipher / encrypt**
   This is a conversion/encoding of plaintext into ciphertext.
7. **Decipher, decrypt** – reverting/decoding plaintext from the ciphertext.
8. **Ciphertext:**
   This connotes an encrypted message derived from encryption algorithm and a secret key.

Figure 1.1 shows a simplified symmetric cryptography model with X as input message or plaintext, Y as ciphertext, Kx as the symmetric secret key, encryption function E() given as Y = E(Kx, X) / Y={X}Kx, and decryption function D() given as X = D(Ky,Y).

Given that $X = X_1, X_2, X_3, \ldots X_i$ as plaintext to be encrypted using the key $X = K_1, K_2, K_3, \ldots K_n$. The i elements of X and n elements of K are either letters of finite alphabets A to Z or binary alphabet {0,1}.

The ciphertext $Y = Y_1, Y_2, Y_3, \ldots Y_j$ is generated from the combination of plaintext X and key K as inputs using the encryption function Y and encryption algorithm E where

$$\boldsymbol{Y = E(K, X)}$$

and the ciphertext Y is transformed to plaintext X by the receiver using the decryption function X and encryption algorithm D, where

$$\boldsymbol{X = D(K, Y)}$$

Features of Cryptosystems

1. The technique used in converting plaintext to ciphertext
   The two conventional principles used by cryptosystems are substitution and transposition techniques. In substitution system, the bit or letter elements are mapped into another bit or letter elements respectively while in transposition, the bits, or letters in the plaintext are rearranged.
2. The number of keys used for encryption and decryption.
   When a single key is used to encrypt and decrypt a plaintext, then the system is called symmetric cryptosystem. At the same time, the asymmetric cryptosystem involves the use of more than one key for encryption.
3. The method of processing the plaintext.
   When a cryptosystem encrypts input parameters in one block or 8 bit of elements at a time, yielding an equivalent block or 8 bit of output for each input parameters, then the type of enciphering is called a block. A stream cipher, on the other hand, encrypts the input parameters in bit by bit as it arrives and produces the output parameter at a time.

3.1.2 Cryptography (Privacy and Authenticity of Data)

Cryptography originated from the Greek word κρυπτός, kryptos, meaning "hidden, secret"; and γράφ, meaning gráph, and "writing", or -λογία, -logia, respectively. It is the study and practice of hiding information to ensure its security. Cryptography also connotes the study of encryption and decryption of message to ensure its protection over the communication network. The study of modern encryption and decryption cut across different field of disciplines, namely mathematical foundation of cryptographic techniques, computing and communication or information storage system, and security services and protocols in engineering technology.

The principle of cryptography essentially is to ensure the confidentiality, authentication, integrity, and availability. The task of confidentiality is ensure using the symmetric and asymmetric cryptography. Ensuring the authenticity of a message or document requires a digital signature being

used while cryptography hash functions are deployed to ensure messages' integrity.

Cryptography is divided into two distinct categories: can you identify these categories?

These are the categories of cryptography:

1. **Classic cryptography** (Criptografia clássica) or Symmetric
   This is a cryptography technique where encryption and decryption functions together with the keys kept secret and processed by key players. This cryptography is also called symmetric cryptography.

2. **Modern/computational cryptography** (criptografia moderna ou criptografia computaconal) or Asymmetric
   In this cryptography technique, the encryption and decryption algorithms are in the public domain, while the encryption and decryption keys are made secret to the sender and receiver. The modern cryptography applies to computers and its applications. This cryptography is also called asymmetric cryptography.

Identify the basic types of cryptography?

The three basic types of cryptography are:

- **Ciphers**: These involve the use of a general algorithm with a secret parameter known only to a select few.
- **Steganography**: The technique of hiding data or information in an image, audio, or video.
- **Codes**: These are ways of looking up data in a secret table.

The Basic Principles of Cryptography / Historical Cipher

- The channel between Ade and Ola is public.
- There available secret key $\mathcal{K}$ share by Ade and Ola.
- Ade encodes his message **X** using a public encryption algorithm $\mathcal{E}$ and key $\mathcal{K}$ represented by $\boldsymbol{Y = \mathcal{E}_{\mathcal{K}}(X)}$.
- Ola decrypts Ade's message using a public decryption algorithm $\mathfrak{D}$ and$\mathcal{K}$represented by $X = \mathfrak{D}_{\mathcal{K}}(Y)$.

3.1.3 Cryptanalysis and Brute-Force Attacks on Symmetric Cipher

These two approaches are used by an attacker to decipher the ciphertext without having the key. The primary aim is to obtain the key used for encryption.

Cryptanalysis or Criptanálise, otherwise called code-breaking, is an attempt to examine or discover the contents of plaintext or key used for encryption using publicly available information. This attack depends on

previous general knowledge of features of plaintext or type of encryption or decryption algorithm.

Brute-force Attack
The attacker used brute-force attack by trying all possible available keys on the ciphertext until a meaningful plaintext is obtained.

Categories of Attacks
1. **Ciphertext only:** In this attack, the encryption algorithm and ciphertext are known to the attacker.
2. **Known plaintext:** This is an attack where an attacker knows the encryption algorithm, ciphertext, or one or more plaintext-ciphertext pairs with the secret key.
3. **Chosen plaintext:** An attacker knows encryption algorithm, ciphertext, selected message in addition to its corresponding ciphertext generated with the secret key.
4. **Chosen ciphertext:** Encryption algorithm, ciphertext, preempt a chosen-ciphertext by an attacker, with corresponding deciphered plaintext using secret key k known to the attacker.
5. **Chosen text**: Encryption algorithm, ciphertext, attacker's chosen plaintext message, together with its corresponding ciphertext obtained using secret key k, assumed attacker's chosen-ciphertext, along with its corresponding decrypted plaintext generated with the secret key.

3.1.4 Substitution

General Simple Cryptosystem
A cryptosystem is a five-tuple element $(X, Y, \mathcal{K}, \mathcal{E}, \mathfrak{D})$ where:
X is the finite set of possible plaintexts
Y is the finite set of possible ciphertexts
$\mathcal{K}$ is a finite set of possible keys
For key $k \in \mathcal{K}$, there exists an encryption function $e_{\mathcal{K}} \in \mathcal{E}$ and a corresponding decryption function $\boldsymbol{d_{\mathcal{K}}} \in \mathfrak{D}$. Each $\boldsymbol{e_{\mathcal{K}}: X \rightarrow Y}$ and $\boldsymbol{d_{\mathcal{K}}: Y \rightarrow X}$ are functions $\ni\ d_k(e_k(x)) = x\ \forall\ plaintext\ x \in X$ where $Y = (e_k(x))$

Substitution Cipher Cryptosystem
Let $\boldsymbol{X = Y = \mathbb{Z}_{26}}$, with key $\mathcal{K}$ having all possible permutations of the 26 symbols 0, 1, 2, …, 25. For each permutation $\pi \epsilon \mathcal{K}$, we define

$e_\pi(x) = \pi(x)$, and

$d_\pi(y) = \pi^{-1}(y)$ where π^{-1} is the inverse permutation to π.

Example 1.1:
Given the following random permutation п, which comprises an encryption function:

Table 1.1. Subsititution Encryption

a	b	c	d	e	f	g	h	I	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
N	X	Y	A	H	P	O	G	Z	Q	W	B	T	F	S	L	R	C	V	M	I	E	K	J	D	U

Encrypt and decrypt the plaintext and ciphertext.

Solution 1.1

Thus, the encryption $e_\pi(a) = N$, $e_\pi(b) = X$ …

Decryption

The decryption

$d_\pi(y) = \pi^{-1}(y)$ =>

Table 1.2. Subtitution Decryption

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	c	r	Y	v	n	h	e	u	x	w	p	t	a	g	f	j	q	o	m	z	s	k	b	c	I

Hence,

$d_\pi(A) = d$, $d_\pi(F) = n$, $d_\pi(H) = e$

3.1.5 Permutation or Monoalphabetic Ciphers

3.1.5.1 Shift Cipher

Shift cipher is defined over $\mathbb{Z}_{26}$ since there is one 26 letters in the English alphabets, though it could be defined over $\mathbb{Z}_m$ for any modulus m. Again, $\forall\, x \in \mathbb{Z}_{26}, d_k(e_k(x)) = x$ where $y = (e_k(x))$

Shift Cipher Cryptosystem

Given that $X = Y = \mathcal{K} = \mathbb{Z}_{26}$
For $0 \leq \mathcal{K} \leq 25$, then $e_k(x) = (x + \mathcal{K}) mod\ 26$
and
$d_k(y) = (x - y) mod\ 26$

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

Example 1.2:

Encrypt and Decrypt the following plaintext "see me tomorrow" given that key k = 11.

Solution 1.2

Step1

Convert the plaintext to a sequence of integer →
The first step is to express each letter of the plaintext in alphabet equivalent. For instance, the alphabet-plaintext:

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10

K	L	M	N	O	P	Q	R	S	T
11	12	13	14	15	16	17	18	19	20

U	V	W	X	Y	Z
21	22	23	24	25	26

The plaintext "see me tomorrow" is

18 4 4 12 4 19 14 12 14 17 17 14 22

Step 2

Add key K to each value, and reduce each to module 26 i.e. (18 + 11) mod 26 = 29 mod 26 = 3

which yields

3 15 15 23 15 4 25 23 25 2 2 25 7

Step 3

Convert the integers to alphabetic characters =>

D PP X P E Z X Z C C Z H

Note: lowercase letters are used for plaintext while uppercase letters are used for ciphertext.

Example 1.3 (Decryption)

Given a ciphertext D P P X P E Z X Z C C Z H, use key K = 11 in shift cipher to decrypt the ciphertext.

Solution 1.3

$d_k(y) = (x - y) mod\ 26$
Ciphertext = D P P X P E Z X Z C C Z H

Step 1

Convert the alphabetic values to numeric =>

3 15 15 23 15 4 25 23 25 2 2 25 7

Step 2

Use $d_k(y) = (x - y) mod\ 26$ to reduce the ciphertext to modulo 26 i.e. when y = D = 3, k = 11, then

3 – 11 mod 26 = -8 mod 26 = 26 x (-1) + 18 = 18
Again, (4-11) mod 26 = -7 mod 26 = 26 x (-1) + 19 = 19

That is, what you are adding to 26 x (-1) that will give you the original negative number is the answer.

Therefore,

$d_k(y)$ = 18 4 4 12 4 19 14 12 14 17 17 14 22

Step 3

Convert the numeric values to alphabet letters plaintext equivalence =>

seemetomorrow

Example:

Encrypt and decrypt the following message "mabcdefgh" using the random permutation in example table 1.3

Solution

Table 1.3 Permutation Encryption and Decryption

m	A	b	c	d	e	f	g	h
T	N	X	Y	A	H	P	O	G

i.e. $e_\pi(m) = T$, $e_\pi(a) = N$ …

Encryption = TNXYAHPOG

For decryption:

$d_\pi(T) = m$, $d_\pi(N) = a$, $d_\pi(X) = b$, …

Decryption = mabcdefgh

3.1.5.2 Affine Cipher

This is a special version of substitution cipher with encryption functions

$e(x) = (ax + b)$, a,b, ϵ $\mathbb{Z}_{26}$

Note: when a = 1, then the affine cipher becomes shift cipher

For any $y \in \mathbb{Z}_{26}$, $(ax + b) = y\ (mod 26)$

and $ax = y - b\ (mod\ 26)$

because y varies over $\mathbb{Z}_{26}$, then $(ax = y(mo26)\ (y\ \in\ \mathbb{Z}_{26})$

Affine Cipher Cryptosystem

Let $X = Y = \mathbb{Z}_{26}$, let $\mathcal{K} = (a, b) \in \mathbb{Z}_{26}$ x $\mathbb{Z}_{26}$: $\gcd(a, 26) = 1$

For $k = (a, b) \in \mathcal{K}$ then

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

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

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

Example 1.4

Given k = (5,3), encrypt the word "shot."

Solution 1.5

a = 5, b = 3

$$e_k(x) = (ax + b) mod\ 26$$

$e_k(x) = (5x + 3) mod\ 26$ and

For each letter of the word "shot", we have

s = (5 x 18 + 3) mod 26 = 15 i.e. s=18 in alphabetic equivalence
h = (5 x 7 + 3) mod 26 = 12
o = (5 x 14 + 3) mod 26 = 21
t = (5 x 19 + 3) mod 26 = 20

therefore, the ciphertext characters are: 15, 12, 21, 20
which is equivalent to alphabetic string P M V U

Decryption

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

$$d_k(y) = 5^{-1}(y - b) mod\ 26$$

$$5^{-1} mod\ 26 = 21$$

i.e

For ciphertext 15, 12, 21, 20

15 => $d_k(15) = 21(15 - 3) mod\ 26 = 21(12) mod 26 = 18$

12 => $d_k(12) = 21(12 - 3) mod\ 26 = 21(9) mod 26 = 7$

21 => $d_k(21) = 21(21 - 3) mod\ 26 = 21(18) mod 26 = 14$

20 => $d_k(20) = 21(20 - 3) mod\ 26 = 21(17) mod 26 = 19$

Converting the numeric letters 18, 7, 14, 19 to alphabetic string

⇨ shot

3.1.6 Vigenere (Polyalphabetic) Cipher

This is a polyalphabetic cryptosystem or cipher named after the founder Blaise de Vigenere. It is an improvement on shift cipher, which is monoalphabetic. The polyalphabetic cipher is using a conversion of alphabetic letters to numeric equivalence, i.e. A=0, B=1, C=3, …, Z=25. Each associated key with an aliphatic string of length n is referred to as keyword.

Vigenere Cipher Cryptosystem

Given m a positive integer and define X = e = $\mathcal{K} = ((\mathbb{Z}_{26})^n$

For a given key $k = k_1, k_2, \ldots k_n$.

Let $e_k(x_1, x_2, \ldots x_n) = (x_1 + k_1, x_2 + k_2, + \cdots x_n + k_n) mod\ 26$
and

$$d_k(y_1, y_2, \ldots y_n) = (y_1 - k_1, \quad y_2 - k_2, + \cdots y_n - k_n) mod\ 26$$

Example 1.5
Suppose the length of word n = 5 and the keyword is CLOSE. Encrypt and decrypt the plaintext "cryptosystem is systemic."

Solution 1.6

Step 1

Convert the plaintext to residue modulo 26, in group 5, and then "add" the keyword modulo 26.

N = 5, keyword CLOSE => K = (2, 11, 14, 18, 4)

Then

Table 1.4. Vigenere Cipher Encryption and Decryption

Plaintext (P) Key (K)	2 17 24 15 19 2 11 14 18 4	14 18 24 18 19 2 11 14 18 4	4 12 8 18 18 2 11 14 18 4	24 18 19 4 12 2 11 14 18 4	8 12 2 11
P + K	4 28 38 33 23	16 29 38 36 23	6 23 22 36 22	26 29 33 22 16	10 13
(P+K)mod 26	4 2 12 7 23	16 3 12 10 23	6 23 22 10 22	0 3 7 22 16	10 13

Step 2

Convert the numeric letters to alphabetical letters

The ciphertext is: E C M H X Q D M K X G X W K W A D H W Q K N

Example 1.6

Decrypt the following ciphertext "ECMHXQ" given m = 5 and keyword is CLOSE

Solution 1.7

Step 1

Convert ciphertext to the numeric equivalent

⇨ 4 2 12 7 23 16

Step 2

Subtract the keywords modulo 26

Ciphertext (Y)	4	2	12	7	23	16
Key (K)	2	11	14	18	4	2
(Y-K) mod 26	2	17	24	15	19	14

Step 3

Convert numeric letters to alphabetic equivalent => crypto

Note: The number of possible keywords of length n in Vigenere cipher is 26^n. For example, the exhaustive keyword search requires a long time.

3.1.7 Hill Cipher

This is another polyalphabetic cryptosystem named after its inventor, Lester S. Hill. The Hill's idea is to take n linear combination of n alphabetic letters in one plaintext letter to produce n alphabetic characters in one ciphertext element, given n as a positive integer, and Y = X = $(\mathbb{Z}_{26})^n$

For example, if n = 2, then the plaintext could be $x = (x_1, x_2)$ and ciphertext could be $y = (y_1, y_2)$

If $y_1 = (3x_1 + 5x_2)\ mod\ 26$ and
$y_2 = (6x_1 + 9x_2)\ mod\ 26$
Then

$$(y_1 y_2) = (x_1 x_2) \begin{pmatrix} 3 & 6 \\ 5 & 9 \end{pmatrix} mod\ 26$$

Hill Cipher Cryptosystem

Given n ≥ 2 be an integer variable. Given Y = X = $(\mathbb{Z}_{26})^n$ and keyspace $\mathcal{K}$ = {n x n invertible matrices over $(\mathbb{Z}_{26})$ }. For any given key k,

$e_k(x) = xk$ **ϵ** ($\mathbb{Z}_{26}$and

$d_k(y) = yk^{-1}$ **ϵ** ($\mathbb{Z}_{26}$

Example 1.7

Given key k = $\begin{pmatrix} 4 & 1 \\ 3 & 2 \end{pmatrix}$, encrypt the plaintext "come."

Solution 1.8

To encrypt plaintext "come."

Step 1

Convert the alphabetic letters to the numeric equivalent

There are two corresponding elements in the plaintext, i.e. co and me, where co = (2, 14) and me = (12, 4). Then

co => (2, 14) and
me => (12, 4)

Step 2

Compute the product of plaintext part with key k

$$e_k(x) = xk$$

co => (2, 14) $\begin{pmatrix} 4 & 1 \\ 3 & 2 \end{pmatrix} = (8 + 42 \quad 2 + 28) = (50 \quad 30)\ mod\ 26 = (24 \quad 4)$
and

me => (12, 4) $\begin{pmatrix} 4 & 1 \\ 3 & 2 \end{pmatrix} = (48 + 12 \quad 12 + 8) = (60 \quad 20)\ mod\ 26 = (8 \quad 20)$

Step 3

Convert numeric letters to alphabetic equivalent
Hence, the encryption of plaintext "come" => YEIU

Decryption

Given $d_k(y) = yk^{-1}$

If k = $\begin{pmatrix} 4 & 1 \\ 3 & 2 \end{pmatrix}$, then $k^{-1} = |k|^{-1}\ (Adj\ k)$

The determinant of k = $|k| = 8 - 3 = 5$

$$Adj\ k = inverse\ of\ transpose\ of\ a\ matrice\ k = \begin{pmatrix} 2 & -1 \\ -3 & 4 \end{pmatrix}$$

$k^{-1} = 5^{-1} \begin{pmatrix} 4 & 1 \\ 3 & 2 \end{pmatrix}$

But $5^{-1}\ mod\ 26 = 21$

Therefore $k^{-1} = 21 \begin{pmatrix} 2 & -1 \\ -3 & 4 \end{pmatrix} mod\ 26 = \begin{pmatrix} 42 & -21 \\ -63 & 84 \end{pmatrix}$

But 42 mod 26 = 16
-21 mod 26 = 26 x (-1) + 5 = 5
-63 mod 26 = 26 x (-3) + 15 = 15
84 mod 26 = 6
i.e.

$k^{-1} = \begin{pmatrix} 16 & 5 \\ 15 & 6 \end{pmatrix}$

We may check this using the identity property of matrix $kk^{-1} = I_n$

⇨ $\begin{pmatrix} 4 & 1 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} 16 & 5 \\ 15 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$yk^{-1} = (24 \quad 4) \begin{pmatrix} 16 & 5 \\ 15 & 6 \end{pmatrix} = (2 \quad 14)$ and

$(8 \quad 20) \begin{pmatrix} 16 & 5 \\ 15 & 6 \end{pmatrix} = (12 \quad 4)$

3.1.8 Transposition System

This is a symmetric cipher where a kind of permutation is carried out on the plaintext to derive new mapped characters. A special type of transposition cryptosystem is rail fence cryptosystem. In the rail fence system, the plaintext is documented as a sequence of diagonals. It is transmitted as a sequence of rows. Another complex way of expressing rail fence is to arrange the words in a rectangular block of rows and columns, permute the order of column, and use the order of the column as the key to the rail fence cryptosystem.

Example 1.8

Encrypt the plaintext "let us meet tomorrow after the class" using rail fence transposition cipher.

Solution 1.9

The first technique is given as:

l t s e t o o r w f e t e l s
e u m e t m r o a t r h c a s

The encrypted message is given as

l t s e t o o r w f e t e l s e u m e t m r o a t r h c a s

The second technique:

Step 1

Express the plaintext in a rectangular block of row by row, and read the text of column by column, permute the column and use the order as the key to the cipher.

The order is

3 1 2 4 5 6 7 <= Key

l e t u s m e
e t t o m o r
r o w a f t e
r t h e c l a
s s v w x y z

Step 2

Arrange the ciphertext in ascending order of the key according to the column. For example, the first key is the second column which will be written first, the second key is the third column and will be the second to be written and so on till the last column which is the key 7 in column 7. Therefore,

The ciphertext

=> e t o t s t t w h v l e r r s u o a e w s m f c x m o t l y e r e a z

Where key = 3 1 2 4 5 6 7

Note: The permutation of the key is done in any order depending on the encryptor discretion.

3.2 Security of Symmetric Cryptography

The security of the symmetric cryptosystem is ensured if two conditions are satisfied:

1. There is a need for a strong encryption algorithm or function so that an attacker with access to encryption algorithm or one or more ciphertext will be unable to recover the ciphertext or identify the key. Since the ciphertext may be publicly available, the opponents should be unable to decipher the ciphertext or identify the used key.
2. The sender or encryptor and receiver or decryptor should have exchanged the key in a secret manner and over the secure communication channel. The feature of the symmetric cryptosystem model where the encryption and decryption algorithms are public while the key is secret makes it widely acceptable to the user.

Example:

1. Determine the number of keys in an affine cipher over $\mathbb{Z}_n$ for n = 30

Solution 1.10

30 = $2^1 \, x \, 3^1 \, x \, 5^1$ then

$\emptyset\,(30) = (2-1)(3-1)(5-1) = 1 \, x \, 2 \, x \, 4 = 8$. Then the number of keys

⇨ n x $\emptyset\,(n)$ = 30 x 8 = 240

4.0 Self-Assessment Exercise(s)

1. Given that k = (5,3), the encryption of the word "cyber security science department" using Affine cipher is

   A. NTIXKPXNZKRUTPNRXQNXSXADKULXQU
   B. NTIXJQXNZKRUTPNRXQNXSXADKULXQU
   C. NTIXKPXNZKRUTPNRXQNXSXADKULXRS
   D. MSIXKPXNZKRUTPNRXQNXSXADKULXQU

   The correct answer is A

2. The encryption of the plaintext "let me see you" using the transposition technique with password 12345 is
A. slueetemyeo B. lsueetemyeo C. lseuetemeye D. lsetemueyeo

Correct Answer: B

3. The number of keys in an affine cipher over $\mathbb{Z}_n$ for n = 90 is
A. 240 B.320C. 360 D. 450
Correct Answer: C

5.0 Conclusion

In this unit, you have learnt the nature and structure of communicating data, together with their security features. Data communication has been categorised into plaintext or ciphertext. The message or word in its original form is called plaintext or intelligible message. In this unit, the basics of symmetric ciphers were discussed. The unit categorised symmetric ciphers into substitution and transposition techniques. The unit gave substitution technique as the one that substitutes numeric and alphabetic letters over $\mathbb{Z}_{26}$. Transposition technique, on the other hand, applied permutation on the plaintext to achieve strategic mapping of text.

6.0 Summary

The security of symmetric ciphers depends on the form of encryption algorithm or function and the exchange of key in a secret manner and over the secure communication channel. The essence of data encryption is to convert the data into unintelligible format before sending over an insecure communication channel to prevent any forms of attacks. The encrypted form of plaintext is called ciphertext. Symmetric cryptography concept involves the use of a single key for message encryption and decryption. Substitution and transposition ciphers are two basic symmetric cryptosystem techniques for encryption and decryption of data. Substitution cipher includes shift cipher, affine cipher, vigenere cipher, and hill cipher. The rail fence technique of transposition technique was also discussed. Examples were given for your better understanding. Finally, the privacy and authenticity of communicating data over insecure communication channel were examined and the best ways to ensure the security of data.

7.0 References/Further Reading

Adebayo, O. S. (2018). CSS 216: Cryptography Theory II, Lecture Note, 2018 (Unpublished).

Jaiswal, R. (n.d.). *Modern Cryptography: An overview.* CSE, IIT Del. Retrieved on 25-12-2019 from http://www.cse.iitd .ernet.in/ ~siy 107537/sil765/PDF/1-Provable-Security.pdf

https://www.cs.umd.edu/class/spring2015/cmsc414/writeups/symmetric-key.pdf

William, S. (2014). *Cryptography and Network Security. Principle and Practice.* (6th ed.). Pearson Education, Retrieved on 25-12-2019 from http://www.uoitc.edu.iq/images/documents/informatics-institute/Competitive_exam/Cryptography_and_Network_Security.pdf.

Unit 2: Indistinguishability Under Chosen-Plaintext Attack

Contents

1.0 Introduction

In this unit, you will learn about the attack on a plaintext called chosen-plaintext attack. You will also learn the security against chosen-plaintext attack. That is how to prevent an attack against plaintext. The unit will also discuss the processing of plaintext using the secret key. Finally, the unit will examine the indistinguishability experiment of Chosen Plaintext Attack (CPA). The experiment will demonstrate how an attacker can win or lose, depending on the probabilistic polynomial-time Turing machine.

2.0 Intended Learning Outcomes (ILOs)

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

- describe Chosen Plaintext Attack (CPA)
- process plaintext using the key.

3.0 Main Content

3.1 Chosen Plaintext Attack (CPA)

A chosen-plaintext attack (CPA) is an attack model designed for cryptanalysis. The attack leans on the presumption that the attacker can

obtain the ciphertexts for plaintexts of its choice. This presumption formally allows the interaction of adversary with an encryption oracle, interpreted as a black box. The attack aims primarily to gain information to break the security of the encryption scheme. In this attack, information available to an attacker includes encryption algorithm, available ciphertext, and one or more plaintext-ciphertext pairs formed with the available or cracked secret key.

CPA Attack Model
Ade (sender) and Ola (receiver) had previously agreed on a pre-shared key, k before the start of communication. Ade will encrypt the message x using the key k to get the corresponding ciphertext y. Ade sent the message x to Ola through an insecure communication medium. Ayo (attacker) can observe the ciphertexts through the medium.

3.1.1 Chosen-Plaintext Dictionary Attacks Against Block Ciphers

This is an attack where attacker Ayo constructs a table with the following practices:

a. Ayo finds key k with encryption of k over 0 message given as $(K, \mathcal{E}_k[0]) \forall K$
b. Sort the ciphertext based on the second field
c. Examine the amount of time spent.

In order to attack a new key K (under chosen message attacks)

a. Ayo selects 0 message, obtain the ciphertext C, looks up in the table, and finds the corresponding key K
b. Examine the amount of time spent

3.1.2 Security against Chosen Plaintext Attack

This is an attack model where an adversary knows plaintext-ciphertext pairs. A comprehensive (but partially) conventional technique to examine this knowledge is to observe this as a model in which the adversary able to see encryptions of arbitrary messages of his choice.

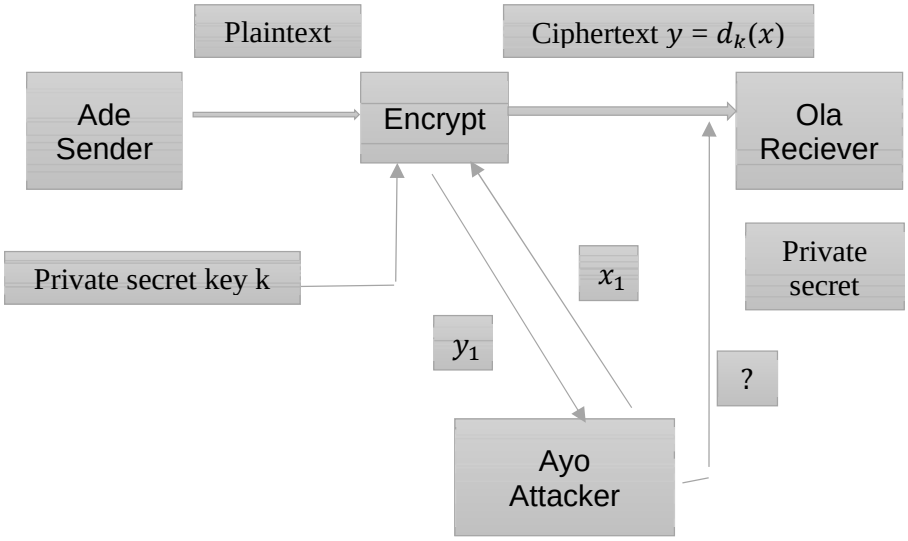

Fig. 1.2 Single Message Security against Chosen-Plaintext Attacks

Fig. 1.2 shows a single message security against Chosen-plaintext attacks where the attacker (Ayo) monitoring the communicating message (x) between Ade (sender) and Ola (receiver). Ayo observed the encrypted message (y) and monitored the pairs x-y over the communicating channel.

3.2 CPA Indistinguishability Experiment

CPA indistinguishability experiment is based on the assumption of computational security. Where an attacker is modelled by a probabilistic polynomial-time Turing machine (PPT), this means the attacker must complete the game and output a "guess" within a polynomial number of time steps. Fig. 1.3 shows an CPA Indistinguishability Experiment model.

The experiment $PrivK^{cpa}_{A,\,II}(n)$ phases

This experiment consists of training, challenge, post-challenge, and response phases.

Training Phase Model
An Attacker T is given Oracle access. Attacker T adaptively submits its query messages. T receives their encryptions.

Challenge Phase:
Attacker T submits two equal length challenge plaintexts, m0 and x_1 and x_2 to the Challenger.

T is free to submit any plaintext of its choice (including the ones already queried during the training phase)

Challenger C obtains the secret key k, for encryption from the conventional cryptosystem.

C plays a random coin to obtain the bit value b of 2 ∈ (0; 1)

C encrypts the corresponding challenge plaintext, say x_b,
sends the ciphertext to T.

Post-Challenge Training Phase
The Oracle access O can still be used by T, even after the challenge

T adaptively submits its query messages
T receives the encryptions of submitted query messages.

Response Phase
T submits its potential guess obtained from encrypted challenge plaintext, in a bit form b^k b^i

T eventually wins the experiment if he guesses correctly but loses if his guess is wrong

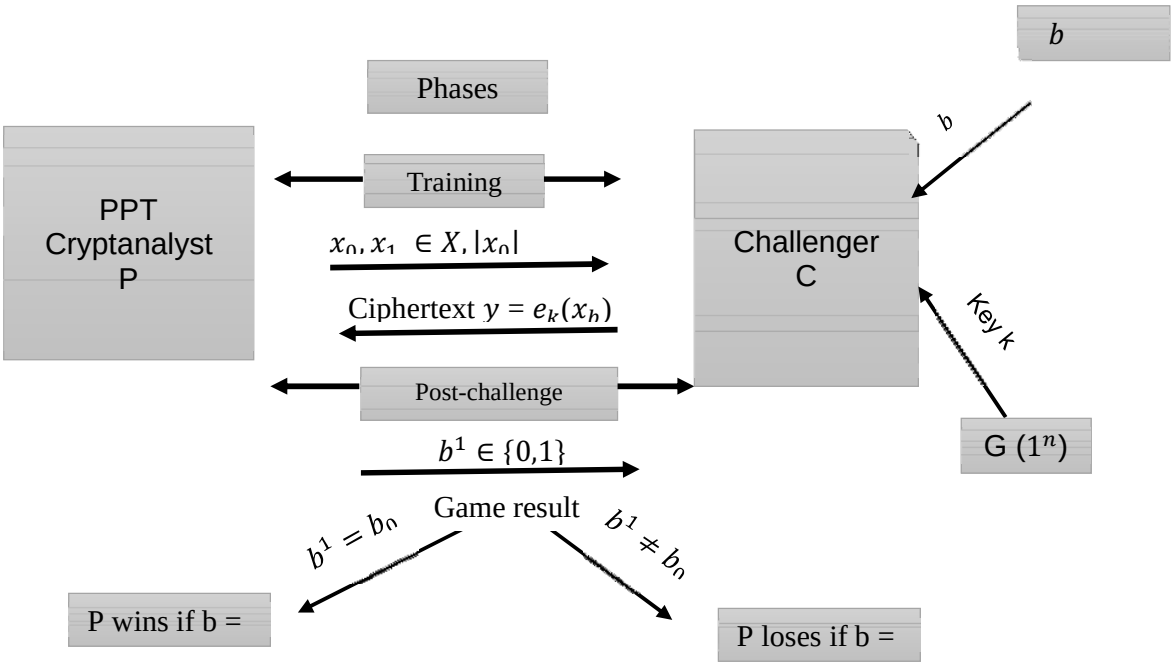

Fig. 1.3 CPA Indistinguishability Experiment

Definition 1.1
A private-key encryption scheme II = (G; $e_k(x_b)$; $d_k(y)$) has indistinguishable encryptions under a chosen-plaintext attack, or is CPA-secure, if for all probabilistic polynomial-time cryptanalyst T there is a negligible function negl(n) such that

$$Pr[PrivK^{cpa}_{A,\Pi(n)} = 1] \leq ½ + negl(n).$$

Examine phases of CPA Indistinguishability.

Solution

There are four phases, namely:

Training Phase Model
An Attacker T is given Oracle access. Attacker T adaptively submits its query messages. T receives their encryptions.

Challenge Phase:
Attacker T submits two equal length challenge plaintexts to the challenger. T submits any plaintext of its choice. Challenger C obtains the secret key k, and plays random coin to obtain the bit value b of 2 ∈ (0; 1). C encrypts the corresponding challenge plaintext and sends the ciphertext to T.

Post-Challenge Training Phase
The Oracle access O can still be used by T even after the challenge. T submits its query messages. T receives the encryptions of submitted query messages.

Response Phase
T submits its potential guess obtained from encrypted challenge plaintext, in a bit form b^k b^i . T wins the experiment if he guesses correctly but loses if his guess is wrong.

4.0 Self-Assessment Exercise(s)

1. The phase of an attack where the attacker receives encryption with given Oracle access is called:

   A. Training Phase B. Challenge Phase C. Response Phase D. Post-challenge phase

   Answer
   A

2. A chosen-plaintext attack (CPA) is an attack model which is
   A. designed for cryptanalysis B. rely on the presumption that the attacker can obtain the ciphertexts for plaintexts of its choice. C. Interpreted as a black box. D. All of the above

   Answer
   D

5.0 Conclusion

In this unit, you have learnt about the attack on a plaintext message called chosen-plaintext attack. You have also learnt the security against chosen-plaintext attack, which is the prevention strategy against plaintext. The unit also discusses the processing of plaintext using the secret key. Finally, the unit examined the indistinguishability experiment of Chosen Plaintext Attack (CPA). The experiment demonstrates how an attacker can win or lose depending on the probabilistic polynomial-time Turing machine.

6.0 Summary

This unit has discussed the chosen-plaintext attack. A chosen-plaintext attack (CPA) is an attack model designed for cryptanalysis. The attack leans on the presumption that the attacker can obtain the ciphertexts for plaintexts of its choice. The unit also x-rayed the security mechanism against this form of attack. It is an attack model where an adversary knows plaintext-ciphertext pairs. Finally, the indistinguishability experiment of Chosen-Plaintext Attack (CPA) was examined. Three phases of indistinguishability experiment were considered: the training, the challenge, and the post-challenge training phase.

7.0 References/Further Reading

Adebayo, O. S. (2019). CSS 723: Modern Cryptography II, Lecture Note, (Unpublished)

Patra, A. (2015). CSA E0 235: Cryptography, (Extra) Lecture 3. Retrieved on March 16, 2015. From https://www.csa. iisc.ac.in/ ~arpita/ Cryptography15.html

Unit 3: Indistinguishability Under Chosen Cipher Text Attack

Contents

1.0 Introduction

In this unit, you will learn how to process the original text called plaintext using a key to produce ciphertext. The unit will examine the concepts of chosen ciphertext attack (CCA) against some cryptosystems and its security. The unit will be concluded by reviewing the CCA experiment which an attacker can perform to gain access to plaintext through chosen-ciphertext attacks. The unit will finally examine the attacker's strategy to carry out his attack.

2.0 Intended Learning Outcomes (ILOs)

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

- describe chosen-ciphertext attack
- describe types of chosen ciphertext attack
- process plaintext using the key, thereby providing ciphertext.

3.0 Main Content

3.1 Chosen Ciphertext Attack (CCA)

This is an attack where an adversary chooses ciphertexts and gets decrypted plaintext through chosen-ciphertext to exploit properties of some cryptosystems. One of the cryptosystems that are vulnerable to CCA is RSA. CCA exploits the properties of RSA to provide info to help cryptanalysis that can counter with a random pad of plaintext or use Optimal Asymmetric Encryption Padding (OASP). Another Cryptosystem that is vulnerable to CCA is ElGamal cryptosystem.

3.1.1 Type of CCA

1. **CCPreMA (Chosen Ciphertext - preprocessing mode (Lunch-break))**
   This is a CCA attack where the challenge ciphertext is given after the release of control of decryption device by the adversary. This attack represents a good model for membership queries in computational learning.

2. **CCPostMA: (Chosen Ciphertext - postprocessing mode)**
   A chosen ciphertext attack where challenge ciphertext is known at the occurrence of attack but in which an attack cannot be submitted.

3.1.2 Elgamal Cryptosystem and Chosen-Ciphertext Attacks

Elgamal is one of the cryptosystems which is not secure against chosen-ciphertext attacks.

Assuming Ade wants to prevent attacker Ayo from decrypting a ciphertext (C, D), but may allow him to decrypt a different ciphertext, then

Ade

a. Compute $(C`, D`) = (C, k D) \bmod p$
b. If Ade get
$m' = Dec (C', D')$,
c. then compute the m, the product of the inverse of key k and ciphertext m, then

$m = (k)^{-1} m' \bmod p$

d. Note: $m = Dec (C, D)$; therefore, Ayo wins.

Since Elgamal cryptosystem is used as a hybrid scheme in practice, then we can say this is not a problem in practice.

From the description of Chosen-Ciphertext Attack (CCA) above, how can you mitigate the attack?

3.2 Security against Chosen Ciphertext Attack (CCA)

In order to ensure the resistance of cryptosystems to CCA, the following steps are to be taken while encrypting the plaintext.

a. Redundancy should be added to the encrypted data. This redundancy will make it challenging to obtain flippant ciphertexts.
b. A consistence-checking routine should be added to the encryption scheme
c. Ola should only decrypt ciphertext if it passes the consistency checks. This is achievable using more than one private keys.

3.3 Chosen Ciphertext Attack Experiment

This experiment is defined over private-key encryption scheme as $\pi = (G, \mathcal{E}, \mathfrak{D})$, any given attack A, and any given value n for the security parameter, where G is generation function, $\mathcal{E}$ is encryption function, and $\mathfrak{D}$ is decryption function.

The CCA indistinguishability experiment $PrivK^{cca}_{A,\Pi}(n)$ is given as:

1. A key k is generated by running $G(1^n)$.
2. The generated 1^n is given to adversary A plus oracle access to $\mathcal{E}_k(.)$ and $\mathfrak{D}_k(.)$. The results are a pair of messages $m0, m1 \in M$ of equal length.
3. A random bit $b \leftarrow \{0,1\}$ is chosen. A challenge ciphertext $C \leftarrow \mathcal{E}_k(m_j)$ is computed and given to attacker A.
4. The attacker A continuously having oracle access to $\mathcal{E}_k(.)$ and $\mathfrak{D}_k(.)$ but not allowed to query the $\mathfrak{D}_k(.)$ on the challenge ciphertext. Finally, attacker A produces a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. Then we write $PrivK^{eav}_{A,\pi}(n) = 1$ if the output = 1 and in this case, we say that A *succeeded*.

Definition:
A private-key encryption scheme $\pi = (G, \mathcal{E}, \mathfrak{D})$ had indistinguishable encryption under a chosen-ciphertext attack if for all probabilistic polynomial-time attackers A there exists a negligible function negl such that $Pr[PrivK^{cca}_{A,\Pi(n)} = 1] \leq ½ + negl(n)$, where the probability is taken over the random coins used by A, as well as the random coins used by the

experiment (for choosing the key, the random bit b, and any random coins used in the encryption process)

Note: The attack has unlimited access to the decryption oracle except for the fact that it was restricted not to request decryption of the challenge ciphertext.
It is, however, allowed to come closer as it desires.

That is, if a scheme has indistinguishable encryption under a chosen-ciphertext attack, then it has indistinguishable multiple encryptions under a chosen-ciphertext attack.

Proposition:

Virtually all the cryptosystems are not CCA-secure.
Consider this construction where encryption is given as

$\mathcal{E}_k(m) = \langle t, H_k(t) \oplus m \rangle = u$

An attacker A experimenting using CCA indistinguishability can select $m_0 = 0^n$ and $m_1 = 1^n$. When an attacker receives $c = \langle t, u \rangle$, he can toss u_1 bit of u to obtain the decryption of the ciphertext c_1. Since $c_1 = c$ this is allowed, and the decryption oracle replies with either 10^{n-1} (where b = 0) or 01^{n-1} (where b = 1).

3.3.1 Attacker's Attack

Given $c_1, c_2$to be ciphertexts observed by an attacker, and given m_1, m_2 represent plaintext to be encrypted. Then

$m_2 = \langle H^{-1}(c_2) \oplus c_1 \rangle$, k is the secret k and block m_2 ends in b times, i.e. $0 \times b \ldots 0 \times b$.

Assuming $c_1' = c_1$ except for the last c byte. The decryption of the ciphertext $c_1', c_2\ is\ m_1', m_2'$ where

$$m_2' = \langle H^{-1}(c_2) \oplus c_1' \rangle$$

Note:

$m_2' = m_2$ except for changes in its final byte

In-Text Question(s)

Examine two types of CCA attacks

Answer

1. **CCPreMA (Chosen Ciphertext - preprocessing mode (Lunch-break))**
   This is a CCA attack where the challenge ciphertext is given after the release of control of decryption device by the adversary.

2. **CCPostMA: (Chosen Ciphertext - postprocessing mode)**
   A chosen ciphertext attack where challenge ciphertext is known at the occurrence of attack but in which attack cannot be submitted.

4.0 Self-Assessment Exercise(s)

1. The following are the vulnerable cryptosystems to CCA exploits except:
   A. Elgamal
   B. RSA
   C. DSA
   D. AES

   The correct answer is D.

2. The property of RSA to provide info to help cryptanalysis exploits is:
   A. Random pad of plaintext
   B. Optimal Asymmetric Encryption Padding (OASP)
   C. CCA
   D. RSA

   The correct answer is C.

5.0 Conclusion

In this unit, you have learnt how to process the original text or plaintext using the key to produce ciphertext. The unit also examines the concepts of chosen ciphertext attack (CCA) against some cryptosystems and its security. The unit concluded by reviewing the CCA experiment which attacker can perform to gain access to plaintext through chosen-ciphertext attacks and the attacker's strategy to carry out his attack.

6.0 Summary

This unit has explained how to process the original text or plaintext using the key to produce ciphertext. The unit also discussed the concepts of chosen ciphertext attack (CCA) against some cryptosystems and its security. The unit rounded up by conducting the CCA experiment which an attacker can perform to gain access to plaintext through chosen-ciphertext attacks and the attacker's strategy to carry out his attack. This chosen-ciphertext attack (CCA) experiment is defined over private-key encryption scheme as $\pi = (G, \mathcal{E}, \mathfrak{D})$, any given attack A, and any given value n for the security parameter, where G is generation function, $\mathcal{E}$ is encryption function, and $\mathfrak{D}$ is decryption function.

7.0 References/Further Reading

Bad, N. (2016). *Chosen-Ciphertext Attacks. Foundations of Cryptography.* Computer Science Department, Wellesley College. Retrieved from http://cs.wellesley.edu/~cs310/lectures/10_CCA_slides_handouts.pdf

Module 2: Asymmetric Cryptography

Unit 1 Asymmetric Encryption Scheme
Unit 2 Problem with Deterministic Encryption
Unit 3 RSA Cryptosystem
Unit 4 Probabilistic Public Key Encryption

Unit 1 Asymmetric Encryption Scheme

Contents

1.0 Introduction

In this unit, you will learn the asymmetric encryption scheme of encrypting and decrypting messages using two independent keys called private and public keys. The private key in asymmetric is used to encrypt the plaintext while the public is made public and used to decrypt the ciphertext (encrypted message). The unit will examine the differences between the symmetric and asymmetric cryptography. The unit will also discuss some useful concepts necessary for the comprehension of asymmetric encryption, which includes the prime number, modular arithmetic, Fermat's and Euler's theorems. In the unit, public-key cryptosystem such as the *ElGamal Cryptosystem* that is based on the Discrete Logarithm problem

and RSA will be discussed. Examples will be given, and solutions to the problems shall be provided.

2.0 Intended Learning Outcomes (ILOs)

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

- define asymmetric encryption scheme
- describe public-key cryptosystem
- identify various types of public-key cryptosystems
- analyse a public-key cryptosystem such as the ElGamal Cryptosystem that is based on the Discrete Logarithm problem.

3.0 Main Content

3.1 Asymmetric Encryption Scheme

This is double-key cryptography where the sender and receiver of message use different keys (private and public keys) for encryption and decryption respectively. In this cryptosystem, both encryption and decryption functions are computed using the same techniques but different keys. Virtually (but not all) most of the cryptosystems are stream cipher oriented, where data are encrypted at once as they arrive, within the encryption algorithm. That is, the data do not store in any location before encryption. This solves the problem of a breach in the security of data that are stored in a memory location. The data are stored to wait for other data to be encrypted. Asymmetric encryption systems are modern cryptosystems which include RSA and ElGamal cryptosystems. Fig. 2.1 is a structure of asymmetric cryptography model.

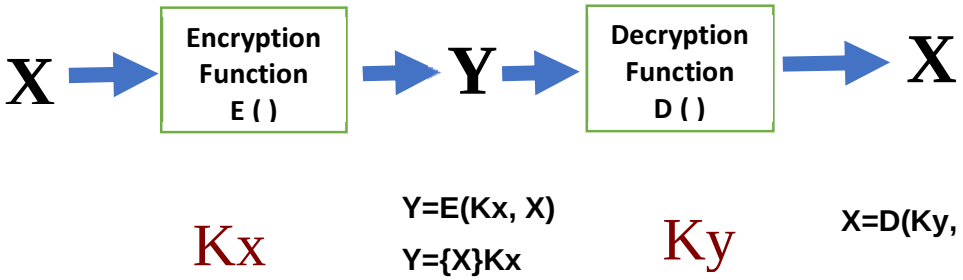

Fig. 2.1: Asymmetric Cryptography Model

From the description of symmetric and asymmetric encryption above, some factors differentiate them. What are these factors?

Difference between Symmetric and Asymmetric Cryptography
The symmetric and asymmetric cryptographies are compared in terms of numbers of keys, the encryption algorithm (reverse o same), below:

The secret key for symmetric is given as Kx = Ky = K while Kx and Ky are different in asymmetric. Also, in terms of encryption and decryption functions, the decryption function D( ) in symmetric cryptography is the reverse of encryption algorithm E( ). In contrast, the encryption and decryption functions in asymmetric are obtained using the same computation.

3.2 Asymmetric or Public-key Cryptography Concepts

A vast number of theories are prerequisites for a better understanding of public-key cryptosystems. These theories of numbers are being used in the design and implementation of asymmetric or public-key cryptosystems. The theories include prime numbers, Fermat's theorem, Euler's theorem, a test of primality, Chinese remainder theorem and discrete logarithm problems.

3.1.1 Prime Numbers

The integer $Q > 1$ is b prime number if and only if $Q's$ divisor is +1 or -1 and = Q. Any integer $b > 1$ can be factored as $b = Q_1^b, Q_2^b, \ldots Q_n^b$

where $Q_1 < Q_2 < \cdots Q_n$ are prime numbers and b is the integer.

Example 2.1

21 = 3 x 7

33 = 3 x 11;

$1617 = 3\,x\,7^2\,x\,11$

In another form, the integer P can be represented with its power such that

$$b = \prod_{p \in P} P^{c_p}$$

Note: each $c_p \geq 0$

For example, the value of b from the following integers can be represented as:

For integer 21, $c = 1, c_7 = 1$

For integer 33 $c_3 = 1, c_{11} = 1$

3.2.1.1 Modular Arithmetics

Given b as an integer and n a positive integer, $b \bmod n$ is a remainder when b is divided by n.

Example 2.2

$11 \bmod 3 = 2$

$$-11 \bmod 3 = 3 \times (-4) + 1 = 1$$

3.2.1.2 Divisors

We posited that a nonzero c divides b if $b = nc$ for some n, where b, c and n are integers. For example, c divides b if there is no value remain (remainders).

Observation:

If b/1, then b = ± 1

If b/c, and c/b, then b = $\pm$ c
Any $c \neq 0$ divides 0

If $\frac{c}{j}$ and $\frac{b}{k}$, then $\frac{b}{lj + mh} \forall$ arbitraryl and m

If $b \equiv 0 \bmod n$, then n/a

3.2.1.3 Properties of Modulo Operation

1. $b \equiv c \bmod n$ if $n/(b - c)$
2. $b \equiv c \bmod n$ implies $c \equiv b \bmod n$)
3. $b \equiv c \bmod n$ and $c \equiv d \bmod n$ implies $b \equiv d \bmod n$
4. If $n/(b - c)$, then $(b - c) = kn$ for some k. Therefore, b = c + kn

Example 2.3

27 = 6 (mod 7) because 27 – 6 = 21 = 7 x 3

-13 = 3 (mod 8) because -13-3 = -16 = 8 x (-2)

-13 mod 8 = 8 x -2 + 3 = 3

3.2.1.4 Modular Arithmetic Operations

1. $[(b \bmod n) + (c \bmod)] \bmod n = (b + c) \bmod n$
2. $[(b \bmod n) - (c \bmod)] \bmod n = (b - c) \bmod n$

3. $[(b \ mod \ n) \times (c \ mod)] mod \ n = (b \times c) \ mod \ n$

Example 2.4

Given b = 13 mod 8 = 5, c = 17 mod 8 = 1. Find

(i) (b+c) mod n
(ii) (b-c) mod n
(iii) (b x c) mod n

Solution 2.1

(i) (b+c) mod n = (13+17) mod 8 = 30 mod 8 = 6
OR (5+1)mod 8 = 6

(ii) (b-c) mod n = (13-17) mod 8 = -4 mod 8 = 8 x (-1) + 4 = 4
OR (5-1)mod 8 = 4

(iii) (b x c) mod n = (13 x 7) mod 8 = 221 mod 8 = 5
OR (5 x 1) mod 8 = 5

3.2.1.5 Greatest Common Divisor (GCD)

The greatest common divisor (gcd) of integer b and c (gcd(b,c)) is an integer d that divides both b and c without remainder and that the divisor of b and c is a divisor of d.

Observation:

1. gcd(b,c) = gcd(b, -c) = gcd(-b,c) = gcd(-b,-c).
   generally, gcd(b,c) = gcd(|b|,|c|)

Example 2.5

Find the greatest common divisor of 30 and 12

Solution 2.2

gcd(30,12) =>

30 = 1,2,3,5,6,10,15,30 (the factors of 30 with no remainder)

12 = 1,2,3,4,6,12 (the factors of 12 with no remainder)

gcd(30,12) = 6

2. gcd(b,c) = 1 if b and c are relatively prime.

Example 2.6

Find gcd (8,15)

Solution 2.3

8 = 1, 2, 4, 8
15 = 1, 3,5,15

gcd (8,15) = 1

3.1.2 Fermat's Theorem

Fermat's theorem plays an essential role in the computation of public-key cryptography

The theorem states:

If Q is a prime b is a positive integer not divisible by Q, then

$b^{P-1} \equiv 1 \ (mod\ Q)$ and

$b^{P} \equiv b \ (mod\ Q)$

Example 2.6

If b = 3, and Q = 7, Show that $b^{P-1} \equiv 1 \ (mod\ Q)$

Solution 2.4

$3^2 = 9 = 2 \ (mod\ 7)$ i.e. $3^{7-1} \equiv 3^6 \equiv 729 (mod\ 7) = 1 \ (mod 7)$

$3^4 = 81 = 4 \ (mod\ 7)$

$3^8 = 6561 = 2 \ (mod\ 7)$

Again, if $b^{P} \equiv b \ (mod\ Q)$, and b = 3, Q = 7, then

$3^7 = 2187 \equiv 3 \ (mod\ 7)$ = a (mod Q)

3.1.3 Euler's Theorem

Another theorem that plays a vital role in public-key cryptography is called Euler's theorem.

Euler's theorem states:

$b^{\emptyset(n)} = 1 \ (mod\ n)$,

$b^{k \times \emptyset(n)+1} = b\ (mod\ n)$, and

$$b^{-1} mod\ n = b^{\emptyset(n)-1}\ (mod\ n)$$

where $\emptyset(n)$ is a Euler's totient function and define as the number of positive integers less than *n* and relatively prime to *n*.
Generally, $\emptyset(1) = 1$, and if q is prime, then $\emptyset(q) = q - 1$

Definition: Given two prime numbers $q\ and\ r \ni\ q\ \neq\ r, then\ n = qr$

and $\emptyset(n) =\ \emptyset(q)\ \times \emptyset(r) = (q-1) \times (r-1)$

Example 2.7

Given b = 6 and n = 35. Show that $b^{\emptyset(n)} = 1\ (mod\ n)$

Solution 2.5

$$\emptyset(n) = (7-1)(5-1) = 24$$

Therefore $6^{24} = 6^{24}(mod\ 35) = 1\ (mod\ 35)$

Example 2.8

Find $20^{62}(mod\ 77)$

Solution 2.5

Let k = 1, then

$$20^{62}(mod\ 77) = (20\ mod\ 77\ \left(20^{\emptyset(77)+1}(mod\ 77)\right) mod 77$$

$$= (20)\left(20(mod\ 77)\right) = 400\ mod\ 77 = 15$$

Example 2.9

Find $8^{-1}(mod\ 77)$

Solution 2.5

$$8^{-1}(mod\ 77) = 8^{\emptyset(77)-1} mod\ 77 = 8^{59}(mod\ 77) = 29\ mod\ 77$$

Note: $\emptyset(77) =\ \emptyset(11)\ \times\ \emptyset(7) = (11-1)(7-1) = 10\ \times 6 = 60$

3.2 Public-key Cryptography Cryptosystems

3.2.1 Discrete Logarithms

The basic fundamentals of public-key cryptosystems are based on **discrete logarithm problem (DLP).** The DLP forms the basis of numerous cryptographic protocols, and the first and best-known DLP is the **Elgamal cryptosystems.**

Discrete Logarithm Problem (DLP)

Given a finite multiplicative group (G), for an element $\propto \in$ G Gravity order n, we define $\langle \propto \rangle = \{\propto^i : 0 \leq i \leq n-1\}$.

Note: $\langle \propto \rangle$ is a subgroup of G, and $\langle \propto \rangle$ is cyclic of order, if G is a finite multiplicative group of a finite field. $\mathbb{Z}_p$ (where P is prime), and $\propto$ be a primitive elt modulo P, then we have $n = |\langle \propto \rangle| = P - 1$.

Again, if $\propto$ is taken as an elt having a prime order q in the multiplicative group $\mathbb{Z}_p$ (where P is prime & P-1 $\equiv 0$ (mdq), then such $\propto$ can be obtained by raising a primitive elt in $\mathbb{Z}_p$ to the $^{P-1}/_{q^{th}}$ power.

DLP Cryptosystem

Given a multiplicative group $(G,\cdot)$, an elt $\propto \in$ G having order n, and an elt $\beta \epsilon \langle \propto \rangle$. Then a unique integer a, $0 \leq 9 \leq$ n-1, such that

$\propto^a = \beta$ is given by;

$a = \log_{\propto} \beta$ where a is called the discrete log of β.

The solution of this a is given by Elgamal public-key cryptosystem.

3.2.2 ElGamal Cryptosystem

ElGamal Public-Key Cryptosystem in $\mathbb{Z}_p$

Let P be a prime such that the Discrete Logarithm problem is $(\mathbb{Z}_p,\cdot)$ is infeasible and let $\propto \in \mathbb{Z}_p$ be a primitive element.

Let P = $\mathbb{Z}_p$, e = $\mathbb{Z}_p \, x \, \mathbb{Z}_p$, and defined.

K = $\{(p, a, \propto, \beta) : \beta \equiv \propto^a \pmod p\}$.

The values P, $\propto$, and β are the public-key, while a is a private key.

For K = $(p, a, \propto, \beta)$, and for a (secret) random number K $\in \mathbb{Z}_{p-1}$, defined.

$e_k(x, k) = (y_1, y_2)$,

Where

$y_1 = \propto^k \bmod p$ And $y_2 = x\beta^k \bmod p$

For $y_1, y_2 \in \mathbb{Z}_p$, define

$$d_k(y_1, y_2) = y_2 (y_1^a)^{-1} \bmod p$$

Example 1.9

Suppose Ade uses an Elgamal scheme with P = 7879, $\propto$ (a primitive elt module P) = 2, a = 75. Encrypt and decrypt a message x = 52 using a random integer k = 43.

Solution 1.11

Given P = 7879, $\propto$ = 2, a = 75, x = 52, k = 43,

$$\beta = \propto^a \bmod p = 2^{75} \bmod 7879 = 4379$$

Ade computes, to send message x

$$y_1 = 2^{43} \text{ mode } 7879 = 3422 \text{ and}$$

$$y_2 = x\beta^k \bmod p = 52 \text{ x } 4379^{43} \bmod 7879 = 733$$

The ciphertext y = (3422, 733),
Ola computes, to decrypt *y*

$$d_k(y_1, y_2) = x = y_2 (y_1^a)^{-1} \bmod p \implies x = 733 (3422^{75})^{-1} \bmod 7879 = \underline{\underline{52}}$$

Note: The basic condition for Elgama cryptosystem to be secured is that the Discrete Logarithm problem is $\mathbb{Z}_p$ is infeasible, i.e. the value a = $log_{\propto}^{\beta}$ must be incomputable. To avoid known attacks, P should have at least 300 digits, and p-1 should have at least one larger prime factor.

3.2.3 RSA Cryptosystem

RSA is the first public-key cryptosystem having full-scaled features of encryption and decryption. RSA became popular ought to the fact that some previous cryptosystem such as Diffie Hellman can only exchange random bits and not other specific information. This is an example of public-key cryptosystem discussed in detail in unit three of this module.

 Discussion

Need for Encryption

1. Imagine the bank accounts of important people like president Muhammadu Buhari and Governor of Central bank of Nigeria, Godwin Emefiele is public?

4.0 Self-Assessment Exercise(s)

1. Given b = 11 mod 8 = 3, c = 7 mod 8 = 7. Find the following:

   (i) (b+c) mod n

   A. 1 B. 2 C. 3 D. 4

   Correct Answer is B

   (ii) (b-c) mod n

   A. 2 B.6 C. 4 D. None of the above

   Correct Answer is C

   (iii) (b x c) mod n

   A. 7 B. 2 C. 3 D. 5

   Correct Answer is D

2. Suppose Ade uses an Elgamal scheme with P = 2579, $\propto$ (a primitive elt module P) = 2, a = 765. The encryption of a message x = 1259 using a random integer k = 853 is:

   A. (415, 839) B. (435, 859) C. (435, 759) D. (405, 859)

   Correct Answer is B

3. One of the following best describes the Asymmetric Encryption Scheme:
   A. This is single-key cryptography where the sender and receiver of message use the same key for encryption and decryption.

   B. This is double-key cryptography where the sender and receiver of message use different keys for encryption and decryption, respectively.

   C. This is single-key cryptography where various keys for encryption and decryption were used by the sender and receiver of message respectively.
   D. This is double-key steganography where the sender and receiver of message use the same key for encrypting and decrypting message respectively.

5.0 Conclusion

In this unit, you have learnt how to prevent data attack using the asymmetric scheme of encryption and decryption. This scheme encrypts and decrypts messages using two independent keys called private and public keys. The private key in asymmetric is used to encrypt the plaintext while the public key is used to decrypt the ciphertext (encrypted message). The unit also examined the differences between the symmetric and asymmetric cryptography. In the unit, you have also learnt some useful concepts necessary for the comprehension of asymmetric encryption such as prime number, modular arithmetic, Fermat's and Euler's theorems. In the unit, public-key cryptosystem such as the ElGamal Cryptosystem that is based on the discrete logarithm problem and RSA were discussed. Examples of these algorithms were given, and solutions to the problems were provided.

6.0 Summary

This unit has x-rayed the concepts of asymmetric encryption scheme of encrypting and decrypting messages using two separate keys called private and public keys. The private key in asymmetric is used for plaintext encryption while the public key is used for ciphertext decryption. The unit will examine the differences between the symmetric and asymmetric cryptography. The unit also examined some useful concepts of asymmetric encryption of public-key cryptosystem for better understanding. *ElGamal Cryptosystem* that is based on the Discrete Logarithm problem and RSA were discussed with adequate examples and solutions to the problems.

7.0 Reference/Further Reading

Adebayo, O. S. (2018). CSS 312: Public Cryptography, Lecture Note, (Unpublished)

Douglas, R. S. (1995). *Cryptography: Theory and Practice*, CRC Press, Boca Raton. Retrieved on 25-12-2019 from http://www.ksom.res.in/files/RCCT-2014-III-CM/CryptographyTheoryandpractice(3ed).pdf.

Jaiswal, R. (n.d.). *Modern Cryptography: An overview*. Retrieved on 25-12-2019 from https://www.cse.iitd.ernet.in/~siy107537/sil765/PDF/1-Provable-Security.pdf.

https://www.cs.umd.edu/class/spring2015/cmsc414/writeups/symmetric-key.pdf

Unit 2: Problem with Deterministic Encryption

Contents

1.0 Introduction

In this unit, you will learn the concept of deterministic encryption. It consists of the meaning, types and problems associated with deterministic encryption. The unit will also examine the difference between the deterministic and probabilistic encryption. Deterministic encryption is public-key encryption which usually produces a single ciphertext over some time, irrespective of a number of plaintext and key used. Probabilistic encryption, on the other hand, allows encryption to be produced using some form of chances.

2.0 Intended Learning Outcomes (ILOs)

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

- describe deterministic encryption
- identify different types of deterministic encryption
- differentiate between the types of deterministic encryption
- analyse the problem with deterministic encryption.

3.0 Main Content

3.1 Deterministic Encryption

Encryption is the conversion of information from a readable (intelligible) format to a non-readable (unintelligible) format. Deterministic encryption is a type of public-key encryption where a given plaintext and encryption key can produce the same ciphertext over a specific time. For example, if a given plaintext "STOP" yields ciphertext "20812182" under certain deterministic encryption, then the plaintext will always yield the same ciphertext. Unlike deterministic encryption, probabilistic encryption always gives a chance for different result by producing different ciphertext, using the same plaintext and key over a number of times.

The main aim of deterministic encryption is to address the challenge of filtering prevention in probabilistic encryption. In order to allow the use of filtering while data is encrypted, patterns are allowed in the data specification. The use of a filter with encrypted data in deterministic encryption is possible with the use of static initialization vector that matches an encrypted data with a specific field input. The static initialization vector is unique for a given field and helps to retrieve the ciphertext that represents a piece of data. Fig. 2.2 is an example of deterministic encryption using type of disease.

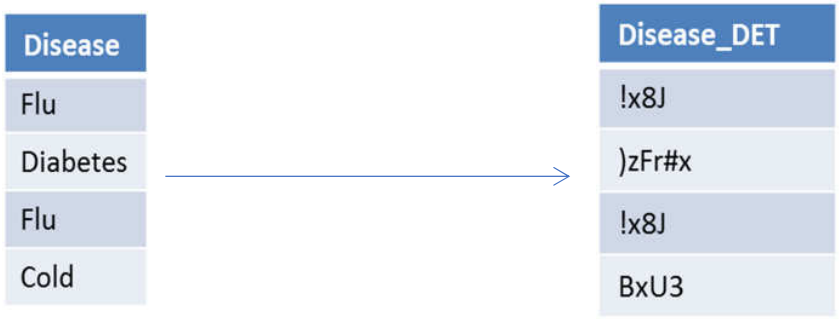

Fig. 2.2 Deterministic Encryption Using Disease Types

Source: Arvind Arasu, Ken Eguro, Ravi Ramamurthy, Raghav Kaushik. Querying Encrypted Data. Microsoft Research

Examine two types of deterministic encryption?

3.1.1 Types of Deterministic Encryption

1. Case Sensitive
   This deterministic encryption allows the filtering of data on the basis of case-sensitive. For example, `ADE` will be considered different from `Ade` and each of them will be identified using different ciphertext strings.

2. Case Insensitive
   This deterministic encryption technique allows data filtering without the use of case sensitivity. For example, `ADE` will be considered the same value with `Ade` and the same ciphertext value will be assigned for both records by the encryption scheme.

3.1.2 Problem with Deterministic Encryption

The decryption algorithm D is usually deterministic in nature while the encryption of public-key cryptosystem is usually probabilistic/randomised/non-deterministic or deterministic. The deterministic decryption takes as input, the private key dk and a ciphertext c. It produces an output message m or a special symbol $\perp$ as a failure.

This can be represented using

$m = D_{dk}(c)$ which also implies

$$m = D(c, dk)$$

$D_{dk}\left(E_{pk}(m)\right) = m \; \forall \; m \; \in M$, except with negligible probability over key

$(p_k, s_k) \leftarrow G\ (1^n)$

The basic problem associated with deterministic encryption is security. That is the deterministic encryption not CPA-secured because secure encryption requires some randomness features.

3.1.3 Deterministic vs Probabilistic Encryption

1. Encryption can be randomised or probabilistic if the same message, same key, running the encryption algorithm yields two different ciphertexts as output. For example, if $Enc_k[m] = (r, PRNG[k||r] \oplus m)$, then the ciphertext c includes two sections, a randomly generated r, and a second part $PRNG[k||r] \oplus m$
2. Decryption is deterministic in the sense that the input plaintext together with key k usually produces single output ciphertext, In other words, the output ciphertext and same key, running the decryption algorithm as many twice as possible always results in the same plaintext

3. Each key induces a one-to-many mapping from plaintext space to ciphertext space. In other words, the ciphertext space must be equal to or larger than plaintext space.

4.0 Self-Assessment Exercise(s)

1. The major problem associated with deterministic encryption is:

   A. Privacy B. Security C. Coercion D. Encryption

   The correct answer is B

2. The major difference between deterministic and non-deterministic encryption is:
   A. Type of ciphertext B. Number of plaintext C. Category of ciphertext D. Number of ciphertext

The correct answer is D

5.0 Conclusion

In this unit, you have learnt the meaning of deterministic encryption. You have also learnt the types and problems associated with deterministic encryption. The unit also examined the difference between the deterministic and probabilistic encryption. Deterministic encryption is defined as public-key encryption which usually produces a single ciphertext over a given time, irrespective of the number of plaintext and key used for encryption. Probabilistic encryption, on the other hand, allows the production of more than one ciphertext with a given plaintext and key k.

6.0 Summary

This unit x-rayed the concept of deterministic encryption as a type of public-key encryption where a given plaintext and encryption key can produce the same ciphertext over a specific period. The unit examined the type and problems associated with deterministic encryption. The two basic deterministic are case sensitive and case insensitive. The unit also examined the difference between the deterministic and probabilistic encryption. The deterministic encryption produces the same type of ciphertext over time using the same key. Conversely, the non-deterministic produces different ciphertext with the same key over some time. In the unit, you are made to understand the deterministic encryption is public-key

encryption which usually produces a single ciphertext over a period of time, irrespective of the number of plaintext and key used while the probabilistic encryption produces more than one ciphertext over time using plaintext and a key.

7.0 References/Further Reading

Arasu, A., Eguro, K., Ramamurthy, R., & Kaushik, R. (2016). "Querying Encrypted Data." *Microsoft Research*. Retrieved on 29-10-2019 from https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/sigmod-tutorial-final.pptx.

David, G. (n.d.). Deterministic vs. Probabilistic Encryption. Retrieved on 18th September 2019 from https://study.com/academy/ lesson/ deterministic-vs-probabilistic-encryption.html#lesson.

Unit 3 RSA Cryptosystem

Contents

1.0 Introduction

In this unit, you will learn the concepts of Rivest-Shamir Adelman cryptosystem as a type of public-key cryptosystem. You will learn the algorithm of RSA as a first public-key cryptosystem. The description of RSA will be given in step by step for a better understanding. The public-key is an asymmetric cryptosystem which uses two keys (private and public keys) for encryption and decryption. This is unlike the symmetric system, which uses a single private key. The unit will also examine the multiplicative properties of RSA. It enables the cryptosystem to compute multiple encryption and decryption using two keys. The security of RSA will be finally examined in the unit, where the situation at which the cryptanalysis of RSA is feasible will be considered. In all the sections, examples will be given, and solutions will be provided for your better comprehension.

2.0 Intended Learning Outcomes (ILOs)

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

- describe the Rivest, Shamir, and Adelman cryptosystem
- discuss and analyse the security of RSA cryptosystem.

3.0 Main Content

3.1 RSA (Rivest-Shamir Adelman) Cryptosystem

3.1.1 The RSA Public-key Cryptosystem

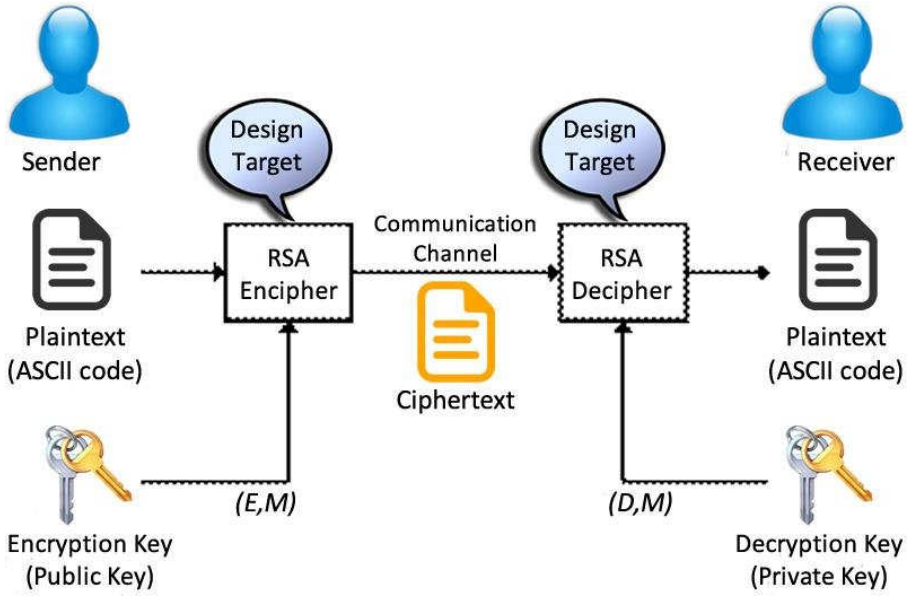

Fig. 2.3: RSA Cryptosystem

RSA is the first public-key cryptosystem having full-scaled features of encryption and decryption. RSA became popular ought to the fact that some previous cryptosystem such as Diffie Hellman can only exchange random bits and not other specific information. Figure 1.1 describes how a sender uses the public key to encrypt a message and send the encrypted message (ciphertext) to the receiver. When the receiver receives the message, he will use the private key to decrypt it back to the (plaintext). Fig. 2.3 shows RSA cryptosystem.

Steps in RSA

Step 1:

Ade and Ola chose two large prime numbers $p \; and \; q$

Step 2:

They compute $n = pq$

Step 3:

They chose an odd number, c such that $1 < c < Q(n)$ which is co-prime to $Q(n)$ (i.e. $c \in \mathbb{Z}^*_{Q(n)}\ c \in \mathbb{Z}$)

Step 4:

From step 3; compute $d = c^{-1} \pmod{Q(n)}$ using extended Euclid's algorithm

Publish (n, c) as the public key and keep d as the secret key.

RSA Encryption Algorithm $e \equiv \mathrm{E}_A\,(m) = m^{CA} \bmod n^A$ that is a message ($0 \leq m \leq n_A$) can be encrypted to use A by algorithm *e.*

RSA Decryption algorithm =>

M = D_A (*e*) = e^{dA} mod n_A

In-Text Question(s)

Example 1.10

- Given that prime number $p\ and\ q$ is given as $p = 13\ and\ q = 17$. 1) Compute the RSA public and private keys (n, c) and d. 2) Use the key (n, c) to encrypt and decrypt message $m = 7$.

Solution 1.12

1)
Given $p = 13\ and\ q = 17$

$$n = pq = 13\ x\ 17 = 221$$

then $Q(n) = Q(p)\ \mathrm{x}\ Q(q) = 12\ x\ 16 = 192$

we choose *c,* an odd; $1 < c < Q(n),\ c \in \mathbb{Z}^*_{Q(n)} \in \mathbb{Z}^*$
i.e. *c* = 11

then $d = c^{-1} \bmod \big(Q(n)\big)$ i.e.

$$d = 11^{-1} \bmod (192) = 35$$

i.e. public key $(n, c) = (221,11)$ while

private key $d = 35$

2) To encrypt and decrypt message $m = 7$ using keys (n, c)

$$e = \mathrm{E}_{(m)}^{A} \equiv m^{cA} \bmod n^{A}$$

$$m = 7,\ c_A = 11,\ n^A = 221,\ d = 35$$

i.e. $m^{cA} \bmod n^A = 7^{11} \bmod 221 = 184.$

Decryption

$$d = 35$$

$$m = D_A\,(e) = e^{dA} \bmod n_A$$

$$184^{35} \bmod 221 = 7$$

3.1.2 RSA Multiplicative Property

Given a public key (n_A, c_A) of user A, and m_1, m_2 then

$$\mathrm{E}_A\,(m_1 \text{ x } m_2) \equiv \mathrm{E}_A\,(m_1) \text{ x } \mathrm{E}_A\,(m_2) \bmod n$$

But

$$\mathrm{E}_A\,(m_1) \equiv {m_1}^{cA} \pmod{n_A}$$

$$\mathrm{E}_A\,(m_2) \equiv {m_2}^{cA} \pmod{n_A}$$

Then

$$\mathrm{E}_A\,(m_1\,,\, m_2) \equiv (m_1\,,\, m_2)^{cA} \equiv {m_1}^{cA},\, {m_2}^{cA}$$

$$\mathrm{E}_A\,(m_1).\mathrm{E}_A\,(m_2) (\bmod n_A)$$

Example 1.11

Given i^{th} user a private key d_i and $n = pq$, a published key c_i such that $\forall_{i,j}\, e_i \neq e_j$ where gcd $(c_i, c_j) = 1$ for user *i* x *j* receive message *m,* then $m^{Ci} \text{ x } m^{Ci\,(y)} \equiv m^{x\,Ci + y\,Cj} = m \bmod n$, given that p=43, q = 59, c=13. Encrypt and decrypt the word "stop".

Solution 1.13

$$N = 43\,X\,59 = 2537$$

$$Q(N) = (43-1)(59-1) = 42\,X\,58 = 2436$$

$$GCD\,(C, Q(N)) = GCD\,(13, 2436) = 1$$

But stop is numerical equivalence of Ansi standard is given as:

18 19 14 15 → STOP

Where **S = 18, T = 19, O = 14, P = 15**

$E = m^{13} MOD\ 2537$

i.e. $1891^{13}\ \text{Mod}\ 2537 = 2081$

and

$1415^{13}\ \text{MOD}\ 2537 = 2182$

THEN ENCRYPTED MESSAGE IS GIVEN AS *E* = 20 81 21 82

Decryption

$$dc = 1\big(\text{MOD}\ Q(n)\big) \Longrightarrow d = c^{-1}\ \big(\text{MOD}\ Q(n)\big)$$

BUT THERE IS AN INTEGER *K* SUCH THAT

$dc = 1 + k\ (p-1)(q-1)$

And

$$e^d = (m^c)^d = m^{cd} = m^{\,1+k(p-1)(q-1)}$$

NOW,

CONSIDER MESSAGE *E* = 20 81 21 82 RECEIVED

$$n = 43\ \text{X}\ 59,\ c = 13$$

THEN

$$d = 13^{-1}\ \text{MOD}\ 2436 = 937$$

Decryption

$$p \Longrightarrow e^d\ \text{MOD}\ Q(n)$$

$$\Longrightarrow 2081^{937}\ \text{MOD}\ 2537 = 1819$$

And

$$\Longrightarrow 2182^{937}\ \text{MOD}\ 2537 = 1415$$

With what we have learnt about RSA cryptosystem, How would you describe the security of RSA?

3.1.3 Security of RSA

The security of RSA is based on the computational difficulty of factoring large integers. That is RSA is secure if the encryption function $e = \mathrm{E}_{(m)}^{A} \equiv m^{cA} \bmod n^{A}$ is one-way, so that it will be computationally infeasible for an opponent to decrypt a corresponding ciphertext.

Therefore, for RSA Cryptosystem to be secure, it is certainly necessary that prime factors *n* = *pq* must be large enough that factoring it will be computationally infeasible. Current factoring algorithms can factor numbers having up to 130 decimal digits. It is, therefore, recommended that primes *p* and *q* having about 100 digits should be chosen where n will have not less than 200 digits. The implementations of RSA on many hardware use a modulus which is 512 bits in length which do not provide good long-term security.

Therefore, the security of RSA depends on the computational complexity of prime factors n over a number of times.

Summarily, the possible techniques of attacking RSA are:

1. Chosen ciphertext attacks - given properties of RSA
2. Brute force key search - infeasible given the size of numbers
3. Mathematical attacks - based on the difficulty of computing ø(n), by factoring modulus n
4. Timing attacks - on the running of decryption

Application Scenario 1
PKI

This scenario consists of two application. The first application is the smartcard-based PKI of a university and the second one is the PKI that is required for the issuance of electronic prescriptions. TU Darmstadt PKI is a smartcard-based PKI. About 30.000 smartcards are given to the students and employees of the university. They contain RSA key pairs. This PKI uses 30.000 smartcards for storing the keys of the end-user. The personalisation of the cards was a time-consuming task. It lasted almost two weeks, although one week was desired. The reason for this is the slow key-generation in software. And the slow transfer of the keys to the card. The registration authority (RA) is installed in a web server. This is because it processes lots of certification requests, especially in the roll-out phase of the PKI. The certification authority (CA) is located in a standard PC. During the roll-out, the production of certificates lasted several days.

Electronic Prescriptions PKI in Germany is the second application which is planned, such that prescriptions are issued electronically. A big, distributed PKI is used. This PKI uses OCSP for verifying the status and the existence of certificates.

4.0 Self-Assessment Exercise(s)

1. RSA can be broken using the possible attacks listed below except:

   A. Brute force key search
   B. Sniffing attacks
   C. Mathematical attacks
   D. Timing attacks

   The correct answer is B.

2. The examples PKI application for RSA key pair are:
   A. Smartcard-based PKI and Electronic Prescriptions PKI
   B. Electronic Prescriptions PKI and RSA PKI
   C. RSA PKI and smart card-based PKI
   D. The smart card-based PKI and TU Darmstadt PKI

   The correct answer is A.

3. This PKI that verify the status and the existence of certificates is called
   A. The smart card-based PKI
   B. TU Darmstadt PKI
   C. Electronic Prescriptions PKI
   D. RSA PKI

   The correct answer is C.

5.0 Conclusion

In this unit, you have learnt the concepts and description of Rivest-Shamir Adelman cryptosystem as the first type of public-key cryptosystem. You also learnt the algorithm of RSA as a first public-key cryptosystem. RSA algorithm was described step-by-step for your better understanding. In addition, public-key cryptosystem was also discussed as an asymmetric cryptosystem which uses two keys for encryption and decryption. This is unlike the symmetric system, which uses a single private key. The unit also examined the multiplicative properties of RSA, in which it can be used to compute multiple encryption and decryption of plaintext using two keys. Finally, the security of RSA was examined where the situation at which the cryptanalysis of RSA is feasible. In all the sections, examples were given, and solutions were provided for better comprehension.

6.0 Summary

This unit examined Rivest-Shamir Adelman cryptosystem as a type of public-key cryptosystem. The unit described the RSA's algorithm in step by step for your better understanding. RSA was discussed as a public key cryptosystem and asymmetric cipher, which uses two keys (private and public key) for encryption and decryption as against the symmetric system, which uses a single private key. You have learnt the multiplicative properties of RSA. How cryptosystem can be used to compute multiple encryption and decryption using two keys. The security of RSA was examined as a situation in which the cryptanalysis of RSA is feasible. Examples were given, and solutions were provided for your appropriate comprehension.

7.0 References/Further Reading

Adebayo, O. S. (2018). *CSS 312: Public Cryptography*. Lecture Note. (Unpublished).

Douglas, R. S. (1995). *Cryptography: Theory and Practice*. Boca Raton: CRC Press. Retrieved on 25-12-2019 from http://www.ksom.res.in/files/RCCT-2014-III-CM/CryptographyTheoryandpractice(3ed).pdf.

Karatsiolis, V., Langer, L., Schmidt, A., Tews, E., & Wiesmaier A. (2010). *Cryptographic Application Scenarios.* Darmstadt Germany. Retrieved on 15th January 2020 from https://www-old.cdc.informatik.tu-darmstadt. de/reports/TR/TI-10-01.Cryptographic Application Scenarios. pdf.

Unit 4: Probabilistic Public Key Encryption

Contents

1.0 Introduction

In this unit, you will learn the probabilistic encryption of public-key encryption using some asymmetric encryption principles. The public-key encryption is divided into deterministic encryption and probabilistic encryption. Probabilistic encryption is a public-key cryptosystem that ensures the information associated with a plaintext is incomputable through ciphertext over a polynomial time. That is, it ensures the security of plaintext at the ciphertext end by preventing an adversary from computing or decrypting the ciphertext. You will also learn the concept of Chinese remainder theorem, and use it to solve two congruent equations to construct keystream z from sender's seed's r. The unit will be concluded with examples for your better understanding.

2.0 Intended Learning Outcomes (ILOs)

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

- describe probabilistic encryption
- identify the challenges associated with probabilistic encryption
- demonstrate the use of substitution-permutation model.

3.0 Main Content

3.1 Probabilistic Encryption

Probabilistic encryption is public-key encryption that gives an element of possibility. That is, using the same number of plaintext and key over a number of times, the ciphertext will always be different. Unlike deterministic encryption, probabilistic encryption always yields different ciphertext using some probabilistic encryptions. Therefore, a plaintext "STOP" will always yield different ciphertext after encryption. Probabilistic encryption aims to ensure the protection of information relating to plaintext at the ciphertext end. That is the information associated with plaintext is incomputable through ciphertext (encrypted message) over a polynomial time. Encryption in a public-key cryptosystem is probabilistic rather than deterministic. It is suitable and realistic for achieving plaintext information protection goal.

Suppose there exist a public key cryptosystem to encrypt a single bit x = 0 or x = 1. The task of determining whether a ciphertext is encryption of 0 or 1 is feasible for an adversary, provided anyone can compute the encryption $e_k(1) and\ e_k(0)$. Also, an adversary can encrypt and hypothesised plaintext to determine the specified value of plaintext.

3.1.1 When to use Probabilistic Encryption

Probabilistic encryption is desirable when there is a need for source information to produce different ciphertext over time. For example, if a certain piece of information about an important personality is to be sent over an insecure communication network; and the information is straight forward information like yes or no, which can easily be guessed by an adversary. Then, probabilistic encryption is desirable to avoid easy guessing through a brute-force attack.

The security of probabilistic encryption is guaranteed by raising the alarm when there is a logical execution of variables in the database. This can also happen when the values of ciphertexts are compared. Therefore, probabilistic encryption did not allow filtering of data since such data has been randomised and became patternless. An example is a case of SQL query where FirstName = `Adebayo` of a record is encrypted using probabilistic encryption, this type of query cannot be executed because the instances of the `Adebayo` are not the same text string.

Probabilistic encryption is, therefore, desirable whenever data in a specific field does not require filtering.

3.1.2 Challenges associated with Probabilistic Encryption

There are challenges associated with block encryption of the message using a randomised encryption scheme. Some of these challenges are:

a. The fact each block of encryption uses k bits of randomness increases the complexity of encryption. For example; the d blocks of encryption require dk bits of randomness.
b. The randomised encryption is expensive to communicate for each block.

3.1.3 Solutions to Control Randomness

a. Creation of an initial state: This involves the use of some randomness otherwise called Nonce or initial virtualisation (IV).
b. The use of the current state to encrypt the current block.
c. There is a need to update the state after each use of the block cipher.

3.1.4 Definition of Probabilistic Public-key Cryptosystem

A probabilistic public-key cryptosystem (PPC) is a six-tuple of $(X,\ Y, \mathcal{K}, \mathcal{E}, \mathfrak{D}, R)$ where

X is the plaintext

Y is the ciphertext or encrypted plaintext

$\mathcal{K}$ is the keyspace

R is the randomiser

For each k in $\mathcal{K}$, encryption $e_k \in \mathcal{E}$ is a public encryption rule and

$e_k \in \mathfrak{D}$ represents secret decryption rule

For PPC, the followings properties are essential:

1. $e_k: P \times R \rightarrow Y$ and $d_k: Y \rightarrow P$. These functions imply $d_k\big(e_k(a,r)\big) = a$

$$\forall\ a \in P, r \in R\ then$$

$$e_k(b,r) \neq e_k(b',r)\ if\ b \neq b'$$

2. If α is a security variable, for fixed $K \in \mathcal{K}$ and for any $b \in P$, the probability distribution

$$P_{(k,b)} is\ defined\ on\ Y$$

$P_{(k,b)}(c)$ is the probability that y is the ciphertext

given key K and plaintext $b\ \forall\, r \in R$

Suppose $b, b' \in P, b \neq b'\ where\ K \in \mathcal{K}$. Then we say

$P_{(k,b)}$ and $P_{(k,b')}$ are not ε-distinguishable

Example 1.12
Given **RSA Generator** with n = 29893. Suppose Ade chooses r = 20749 and wants to encrypt the 10-bit plaintext string plaintext $b = 1101001101$. Encrypt this string.

Since r = QR(n) then s_0= QR(n)
if $i = 0$ then $s_i = s_0$

Solution 1.15

Step 1

Using the **Blum-Blum-Shub Generator**

$$z_i = \left({s_0}^{2^i} mod n\right) mod 2$$

The generator is derived from the simple computation as follows:

Given a seed s_0= QR(n), we compute the sequence $s_1, s_2 \ldots s_j$ by successive squaring modulo n, and then reduce each s_i modulo 2 to obtain z_i

$$1 \leq i \leq j$$

n = 29893
s_0=r = 20749

Table 2.1: Bits Produced by BBS Generator

i	s_i	z_i
0	20749	
1	2015	1
2	24670	0
3	17313	1
4	2858	0
5	7375	1
6	15258	0
7	29773	1
8	14400	0
9	22152	0
10	17509	1

The bit string from seed $s_1 \; to \; s_{10}$ is

1010101001

Given plaintext $b = 1101001101$

$z = 1010101001$

Step 2

To compute ciphertext c, Ade finds the exclusive-or of plaintext b and keystream z, which yield ciphertext c to be sent to Ola

$$b = 1101001101$$
$$z = 1010101001$$
$$c = b \oplus z = 0111100100$$

Note: $b \oplus z$ is = 1 when the mapping bits are different, otherwise = 0

Step 3

Ade computes

$$s_i = {s_0}^{2^i} mod n$$

$$s_{11} = \left({s_{10}}^{2^1} mod \; 29893\right)$$

$$= 17509^2 mod 29893$$

= 12366 and send to Ola

Step 4

Ade computes two primes p and q from $n = pq$, where $p = 167$ and $q = 179$ and

$$t_1 = ((p+1)/4))^{j+1} mod\ p - 1 = (\frac{167+1}{4}))^{10+1} = 42^{11} mod\ 166$$

$$= 126$$

and

$$t_2 = ((q+1)/4))^{j+1} mod\ q - 1 = (\frac{179+1}{4}))^{10+1} = 45^{11} mod\ 178$$

$$= 1$$

and

$$u_1 = ({s_{11}}^{t_1}) mod\ p = 12366^{126} mod\ 167$$

$$= 126$$

$$u_2 = ({s_{11}}^{t_2}) mod\ q = 12366^{1} mod\ 179$$

$$= 15$$

Step 5

Ola solves the congruences $r \in R$, a randomiser:

$$r \equiv u_1\ mod\ p$$
$$r \equiv u_2\ mod\ q$$

=>

$$r \equiv 126\ mod\ 167 \qquad \text{r1}$$

$$r \equiv 15\ mod\ 179 \qquad \text{r2}$$

Using Chinese remainder theorem to solve r1 and r2 simultaneously:

3.2 Chinese Remainder Theorem

The theorem states:

Given two congruences

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

Then

$$X = \sum_{i=1}^{r} a_i M_i y_i \pmod{M}$$

Where $M_i = \frac{M}{mi} =$

$$y_i = M_i^{-1} \pmod{m_i}$$

$a_i = a_1, a_2$

Example 1.13

Given two congruences

$$a \equiv 3 \pmod 4$$
$$a \equiv 4 \pmod 7$$

Find the value of a

Solution 1.16

r = 2, m_1 = 4, m_2 = 7,

M = 4 x 7 = 28

$M_i = \frac{M}{mi} =$

For $i = 1$

$M_1 = \frac{28}{4} = 7$

F*or* $i = 2$

$M_2 = \frac{28}{7} = 4$

i.e. gcd (M_i, m_i) = 1

gcd (7,4) ≡ 1

for $y_i = M_i^{-1} \pmod{m_i}$

$i = 1$

$y_1 = 7^{-1} \bmod 4 \equiv 3$

$i = 2$

$y_2 = 4^{-1} \bmod 7 \equiv 2$

$i = 1,\ a_i = 3$

$i = 2,\ a_i = 4$

But

$= (3 \text{ x } 7 \text{ x } 3) + (4 \text{ x } 4 \text{ x} 2) (\text{mod } 28)$

$= (63 + 32) \bmod 28$

$= 95 \bmod 28$

$= 11$

Therefore,

For

$r \equiv 126\ mod\ 167$ r1

$r \equiv 15\ mod\ 179$ r2

M = 167 x 179 = 29893

r = number of equations in the congruence to be solved = 2 , m_1 = 167, m_2 = 179,

$M_i = \frac{M}{mi} =$

For $i = 1$

$M_1 = \frac{29893}{167} = 179$

F$or\ i = 2$

$M_2 = \frac{29893}{179} = 167$

i.e. gcd (M1, M2) = 1

gcd (179,167) ≡ 1

for $y_i = M_i^{-1} \pmod{m_i}$

$i = 1$

$y_1 = 179^{-1} \bmod 167 \equiv 14$

$i = 2$

$y_2 = 167^{-1} \bmod 179 \equiv 164$

$i = 1,\ a_i = 126$

$i = 2,\ a_i = 15$

But

$$r = \sum_{i=1}^{r} a_i M_i y_i \pmod{M}$$

$= (126 \text{ x } 179 \text{ x } 14) + (15 \text{ x } 167 \text{ x}164) \pmod{29893}$

$= (315756 + 410820) \bmod 29893$

$= 726576 \bmod 29893$

$= 9144$

Application Scenario

Telephony

Consider the telephony application scenario. In general, an A-Field has 64 Bit. However, only 40 of them are used for C-channel messages. A B-Field can have an arbitrary length and can be fully encrypted. In general, the total length of a keystream segment in bits is lf = 40+ lb −cb Where lb is the length of the B-field, and cb is the number of bits used for checksums in the B-field. The most common value for lb is 320; usually, no checksums in the B-field are used, resulting in cb = 0. The maximum length of a keystream is twice the length of the keystream segment. A device which uses all available timeslots on a single frequency (there are only 12 double-slots instead of 24 slots) must be able to encrypt: 12∗100∗lf bits per second. A device which holds a single call (24 timeslots, 2 of them used) needs to be able to encrypt: 2∗100∗lf bits per second. The most common value for lf is 360, the maximum value for lf is 4840, which results in at most 5808000 bits per second which need to be encrypted. At most 2∗24∗100 different frames per second are processed on a channel when half-frames are used.

4.0 Self-Assessment Exercise(s)

1. Use Chinese remainder to find the value of a in the following congruence:

$$a \equiv 3 \pmod 7$$
$$a \equiv 4 \pmod{11}$$

A. 18 B. -18 C. 15 D. -15

Answer is B

2. The appropriate time to adopt probabilistic encryption is

   A. When there is a need for source information to produce different ciphertext over some time.
   B. When there is a need for source information to produce the same ciphertext over some time.
   C. When there is a need for target information to yield different plaintext over some time
   D. When there is a need for source plaintext to produce different ciphertext over some time.

   Correct Answer is A

5.0 Conclusion

In this unit, you have learnt the probabilistic encryption of the public-key encryption scheme. From the unit, You learnt that public-key encryption is divided into deterministic and probabilistic encryption. The probabilistic encryption is defined as a public-key cryptosystem that ensures the information associated with plaintext is incomputable through ciphertext (encrypted message) over a polynomial time. That is, it provides the security of plaintext at the ciphertext end by preventing an adversary from computing or decrypting the ciphertext. You have also learnt the concept of Chinese remainder theorem and use it to solve two congruent equations to construct keystream z from sender's seed's r. The unit included examples and solutions for better understanding.

6.0 Summary

This unit contains the probabilistic encryption of public-key encryption using some asymmetric encryption fundamentals. The probabilistic encryption is a special type of public-key encryption that ensures the information associated with plaintext is incomputable through ciphertext (encrypted message) over a polynomial time. You have also learnt the concept of Chinese remainder theorem to solve the two congruent equations to construct keystream z from sender's seed's r. The unit was concluded with examples for better understanding.

7.0 References/Further Reading

Adebayo, O. S. (2018). "CSS 312: Public Cryptography," Lecture Note. (Unpublished).

Douglas, R. S. (1995). *Cryptography: Theory and Practice*. Boca Raton: CRC Press.

Karatsiolis, V., Langer, L., Schmidt, A., Tews, E. & Wiesmaier, A. (2010). *Cryptographic Application Scenarios.* Darmstadt, Germany. Retrieved from https://www-old.cdc.informatik.tu-darmstadt.de/reports/TR/TI-10-01.Cryptographic_Application _Scenarios.pdf.

Module 3: Algorithms of Modern Cipher

Unit 1 AES, DES and RSA
Unit 2 RC4 and Key Exchange Management

Unit 1 AES, DES and RSA

Contents

1.0 Introduction

-

This unit introduces you to the concepts of block ciphers and example of the public cryptosystem. The unit discusses two types of block cipher of asymmetric cryptography, namely AES and DES. The unit also discusses a sample of public cryptosystem or asymmetric cryptosystem called RSA. You will learn the algorithms of these cryptosystems and how to solve related problems.

2.0 Intended Learning Outcomes (ILOs)

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

- describe the Advance Encryption Standard
- analyse the Data Encryption Scheme
- explain the Rivest Shamir and Adelman

3.0 Main Content

3.1 AES (Advanced Encryption Standard)

Why was AES established?

3.1.1 Description of AES

The Advanced Encryption Standard (AES) is a block cipher developed by Rijndael. The algorithm was adopted by the National Institute of Standard and Technology (NIST) in January 1997 to replace the previous Data Encryption Standard (DES). AES in block means the number of bytes it encrypts is fixed and can only encrypt blocks of 16 bytes or 128 bits in a memory at a time. If the bytes being processed are larger than 16 bytes, then AES will execute them concurrently. It also supports the key lengths of 128, 192, and 256 bits. AES was accepted finally after its evaluation based on the three criteria: security, cost, and implementation features of its algorithm. AES is an iterative cipher that operates on entire data in every round of operation.

1.1.2 Description of AES

AES is a cipher with a block length 128 bits or 16 bytes

It allows three key lengths to be used namely 16 bytes (128 bits), 24 bytes (192 bits), and 32 bytes (256 bits).

It is an iterated cipher with varying number of round depending on the lengths of keys. For examples, the number of round (N_r) for 16 bytes key is 10, the N_r for 24 bytes key = 12, while N_r = 14 for 32 bytes key.
AES has the capability against any known attack

Note: 1 byte = 8 bits

Operations of AES are four in numbers, can you describe those operations?

1.1.3 Implementation of AES

There are four basic steps in the operation of AES
1. Add Round Key
2. Byte Sub
3. Shift Row
4. Mix Column

Algorithm of AES

Step1:

Given a plaintext p, initialise the State to be p and perform ADD ROUND KEY operation. This x-ors Roundkey with State value p.

Step2:

For each first round $N_r - 1$, perform a **SubBytes** (substitution) operation on p using S-box table

Perform a **ShiftRows** operation on a p

Perform **MixColumns** on p

Perform **AddRoundKey**

Step3:

Perform **SubBytes, ShiftRows, and AddRoundKey**

Step 4:

Define q ciphertext to be **State**

1.1.4 Example of AES

Example 3.1

Given a 128 bits secret key, “That's, my New man”. Encrypt the given plaintext 2191.

Note: Generate and use the first two set of key for the encryption.

Solution 3.1

Step 1:

The first step is to present the key in hexadecimal ASCII equivalent, hence
Table 3.1 Key representation in Hexadecimal

Key	T	h	a	t	s		m	y		N	e	w		m	a	n
He	5	6	6	7	7	2	6	7	2	4	6	7	2	6	6	6

Key in hex: (128 bits): 54 68 61 74 73 20 6D 79 20 4E 65 77 20 6D 61 6E

Plaintext in English: Two One Nine One

Table 3.2 Plaintext translate to Hexadecimal

Plai	T	W	o		O	n	e		N	i	n	e		O	n	e
Hex	5	7	6	2	4	6	6	2	4	6	6	6	2	4	6	6

Plaintext in Hex: 54 77 6F 20 4F 6E 65 20 4E 69 6E 65 20 4F 6E 65

Step 2:

Generate first round key

Key in Hex: 54 68 61 74 73 20 6D 79 20 4E 65 77 20 6D 61 6E

w[0] = (54 68 61 74)
w[1] = (73 20 6D 79)
w[2] = (20 4E 65 77)
w[3] = (20 6D 61 6E)

Starting from w[3] = (20 6D 61 6E)

Perform the following operations:

1. Byte Sub
2. Shift Row
3. Add round constant

1. After byte left shift of w[3]

w[3] = (6D 61 6E 20)

2. Using S-Box table in figure 3.1.1, perform byte substitution

g(w[3]) = (3C, EF, 9F, B7)

3. Add the first round constant (01, 00, 00, 00)

g(w[3]) = (3B EF, 9F, B7)

Then we find w[4] = w[0] ⊕ g(w[3]) = (54 68 61 74) ⊕ (3B EF 9F B7)

We convert Hexadecimal to a binary bit before XOR operation yielding

(54 68 61 74) = 0 1 01 0 1 0 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 1 0 0
⊕
(3B EF 9F B7)= 0 0 1 1 1 0 1 11 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 0 1 1 1

0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1

= (6F, 87, FE, C3)

Note, the binary of 5 from hexadecimal = 0101 and the bitwise operator XOR is = 1 if the bit is different otherwise it is = 0.

For example:
1 ⊕ 1 = 0

1 ⊕ 0 = 1

0 ⊕ 0 = 0

Apply the three previous three steps to find successive w[n] where n = 5, 6, 7

w[5] = w[1] ⊕ g(w[4]) = (1C, A7, 93, BA)
w[6] = w[2] ⊕ g(w[5]) = (3C, E9, F6, CD)
w[7] = w[3] ⊕ g(w[6]) = (1C, 84, 97, A0)

Note: To use S-Box table, for example for 6D, we will find the element lies where the first column of the table is 6 and where the element in the first row is D. The corresponding element in this is 3C. The first table is the S-box table while the second is inverse S-box table first generated key is given as:

Round 1 Key: = (w[4], w[5], w[6], w[7])

= (6F, 87, FE, C3, 1C, A7, 93, BA, 3C, E9, F6, CD, 1C, 84, 97, A0)

After getting the round one key, we repeat the steps for the second round key generation i. e.

New w[0] = (6F, 87, FE, C3)
w[1] = (1C, A7, 93, BA(
w[2] = (3C, E9, F6, CD)
w[3]= (1C, 84, 97, A0)

Apply the three steps again,

1. Byte Sub
2. Shift Row
3. Add round constant

Note: The second round constant = (02, 00, 00, 00). The round constant is increasingly and generated through the formula below:

RCm(i) = RC[E], 0 x 02, 0 x w)

If i = 1, then RC(m(1) = (0 x 01) = 01
If i = 2, RC(m(2) = (0 x 02 x Rc(Rj-1)) = (0 x 02 x Rc1)) = (0 x 02 x 1)) = 02
i = 3 RC(m(3) = (02 x Rc(2)) = (02 x 02) = 04

i = n RC(m(n) = (02 x RC(n - 1))

Round two (2) Key: K2 =

32 0F 1E 5F 2E A1 8D E5 12 41 7B 28 0E C5 EC 85

Round three (3) Key: K3 =

90 C1 89 F4 BE 61 09 11 AC 20 72 39 A2 E5 9E BC

We have three keys now:

K0 = 54 68 61 74 73 20 6D 79 20 4E 65 77 20 6D 61 6E

K1 = 6F 87 FE C3 1C A7 93 BA 3C E9 F6 CD 1C 84 97 A0

K2 = 32 0F 1E 5F 2E A1 8D E5 12 41 7B 28 0E C5 EC 85

K3 = 90 C1 89 F4 BE 61 09 11 AC 20 72 39 A2 E5 9E BC

Note: We may be asked to generate up to 10 keys for encryption

Encryption using generated three keys

Plaintext = 2 1 9 1

The hex equivalent is

54 77 6F 20 4F 6E 65 20 4E 69 6E 65 20 4F 6E 65

FOR ADD ROUND KEY (ROUND 0), the state matrix is formulated from the plaintext

$$= \begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 66 & 69 & 4F \\ 6F & 65 & 6E & 6E \\ 20 & 20 & 65 & 65 \end{pmatrix}$$

The state matrix is generated for key round 0

$$= \begin{pmatrix} 54 & 73 & 20 & 20 \\ 68 & 20 & 4E & 6D \\ 61 & 6D & 65 & 61 \\ 74 & 79 & 77 & 6E \end{pmatrix}$$

The next step is to XOR Plaintext and Round 0 key

$$= \begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 66 & 69 & 4F \\ 6F & 65 & 6E & 6E \\ 20 & 20 & 65 & 65 \end{pmatrix} \oplus \begin{pmatrix} 54 & 73 & 20 & 20 \\ 68 & 20 & 4E & 6D \\ 61 & 6D & 65 & 61 \\ 74 & 79 & 77 & 6E \end{pmatrix} =$$

= 54 4F 4E 20 77 6E 69 4F 6F 65 6E 6F 20 20 65 67

$\oplus$

54 73 20 20 68 20 4E 6D 61 6D 65 61 74 79 77 6E

Which yields

00 3C 6E 00 1F 4E 27 02 0E 08 0B 0F 54 59 12 0B

The new matrix is

$$\begin{pmatrix} 00 & 3C & 6E & 00 \\ 1F & 4E & 27 & 02 \\ 0E & 08 & 0B & 0F \\ 54 & 59 & 12 & 0B \end{pmatrix} \rightarrow P \text{ implies Current State Matrix}$$

The next step is to perform a Subyte (Substitution) operation using the S-box table

$$\rightarrow \begin{pmatrix} 63 & EB & 9F & 63 \\ C0 & 2F & CC & 77 \\ AB & 30 & 2B & 76 \\ 20 & CB & C9 & 2B \end{pmatrix} \rightarrow C$$

Perform the Shift row operation on C yields

$$\begin{pmatrix} 63 & EB & 9F & 63 \\ 2F & CC & 77 & C0 \\ 2B & 76 & AB & 30 \\ 2B & 20 & CB & C9 \end{pmatrix} \rightarrow D$$

Perform MixColumn operation on D using Galois field constant

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$ to multiply D yielding

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & 63 \\ 2F & CC & 77 & C0 \\ 2B & 76 & AB & 30 \\ 2B & 20 & CB & C9 \end{pmatrix} =$$

In order to perform MixColumn, we multiply the row element in the first matrix with the first column elements in the second matrix yielding

$$\begin{pmatrix} 02*63+03*2F+01*2B+01*2B & 02*EB+03*CC+01*76+01*20 & 02*9F+03*77+01*AB+01*CB & 02*63+03*C0+01*30+01*C9 \\ 01*2F+02*2F+03*2B+01*2B & 01*EB+02*CC+03*76+01*20 & 01*9F+02*77+03*AB+01*CB & 01*EB+02*CC+03*76+01*20 \\ 01*2B+01*2F+02*2B+03*2B & 01*EB+01*EC+02*76+03*20 & 01*9F+01*77+02*AB+03*CB & 01*9F+01*77+02*AB+03*CB \\ 03*63+01*2F+01*2B+02*2B & 03*EB+01*CC+01*76+02*20 & 03*63+01*C0+01*30+02*C9 & 03*63+01*C0+01*30+02*C9 \end{pmatrix}$$

=

$$\begin{pmatrix} B7 & B4 & 9F & 64 \\ 6B & F2 & 77 & C3 \\ 67 & CD & AB & 83 \\ F7 & DC & EB & DC \end{pmatrix} \rightarrow G$$

FOR ADDROUNDKEY ROUND 1

XOR Current state G with Key K1

$$\rightarrow\ G\ \oplus \begin{pmatrix} 6F & 1C & 3C & 1C \\ 87 & A7 & E9 & 84 \\ FE & 93 & F6 & 97 \\ C3 & BA & CD & A0 \end{pmatrix} = \begin{pmatrix} D8 & A8 & E0 & 78 \\ EC & 55 & B5 & 47 \\ 99 & 5E & 15 & 14 \\ 34 & DC & EB & DC \end{pmatrix} \rightarrow\ H$$

H is the new state matrix and AES output after Round 1

FOR ADDROUNDKEY ROUND 2

K2 = 32 0F 1E 5F 2E A1 8D E5 12 41 7B 28 0E C5 EC 85

Step 1: Perform Substitution on new state H using S-box table

$$\begin{pmatrix} D8 & A8 & E0 & 78 \\ EC & 55 & B5 & 47 \\ 99 & 5E & 15 & 14 \\ 34 & DC & EB & DC \end{pmatrix} \rightarrow \quad \begin{pmatrix} 61 & C2 & E1 & BC \\ CE & FC & D5 & A0 \\ EE & 58 & 59 & D4 \\ 68 & 86 & E9 & 86 \end{pmatrix} \rightarrow\ I$$

Step 2: Perform Shift row operation I

$$\begin{pmatrix} 61 & C2 & E1 & BC \\ FC & D5 & A0 & CE \\ 59 & D4 & EE & 58 \\ 86 & E9 & 86 & 68 \end{pmatrix} \rightarrow\ J$$

Step 3: Perform MixColumn on new Matrix J

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} 61 & C2 & E1 & BC \\ FC & D5 & A0 & CE \\ 59 & D4 & EE & 58 \\ 86 & E9 & 86 & 68 \end{pmatrix} = \begin{pmatrix} 02 & AD & 4A & 0A \\ EF & FD & 95 & BB \\ BE & 84 & 17 & 7A \\ 11 & 95 & 61 & 25 \end{pmatrix} \rightarrow\ M$$

Step 4: XOR M with Key K2

$$\begin{pmatrix} 02 & AD & 4A & 0A \\ EF & FD & 95 & BB \\ BE & 84 & 17 & 7A \\ 11 & 95 & 61 & 25 \end{pmatrix} \oplus \begin{pmatrix} 32 & 2E & 12 & 0E \\ 0F & A1 & 41 & C5 \\ 1E & 8D & 7B & EC \\ 5F & E5 & 28 & 85 \end{pmatrix} = \begin{pmatrix} 30 & 83 & 58 & 04 \\ E0 & 56 & D4 & 7E \\ A0 & 09 & 6C & 96 \\ 4E & 70 & 49 & A0 \end{pmatrix} \rightarrow\ L$$

L is an output matrix for round 2

FOR ROUND 3

L is the new state matrix

1) Perform Byte sub on L using S-box

$$\begin{pmatrix} 04 & EC & 6A & F2 \\ E1 & 4A & 48 & F3 \\ E0 & 01 & 50 & 90 \\ 2F & 51 & 3B & E0 \end{pmatrix} \rightarrow L$$

2) Perform Shift Row:

$$\begin{pmatrix} 04 & EC & 6A & F2 \\ 4A & 48 & F3 & E1 \\ 50 & 70 & E0 & 01 \\ E0 & 51 & 3B & 3B \end{pmatrix} \rightarrow R$$

Note: There is no MixColumn operation in the last Round i.e

The Ciphertext is: 04 4A 50 E0 EC 48 70 51 6A F3 E0 3B F2 E1 01 3B

END OF AES

(a) S-box

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

(b) Inverse S-box

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Fig. 3.1.1 a): S-box Table and b) Inverse S-box Table

3.1.5 AES Challenges

Although AES is resistant to forms of cryptographic encryption attacks, however, there was a concern the conventional techniques of linear and differential cryptanalysis can be deployed against it. A square attack which has been used against AES's predecessors could be used against it.

3.2 DES (Data Encryption Standard)

The DES (Data Encryption Standard) algorithm was adopted by NIST in May 1973 and became the most widely used encryption algorithm for security. It is a special form of block and iterated cipher known as Feistel cipher. DES as a block cipher indicates it operates on plaintext blocks of a 64-bits and returns ciphertext blocks of 64-bits. When each of 64-bit block is encrypted separately, the encryption strategy is referred to as Electronic Code Book (ECB) mode. The two additional Data Encryption Standard modes are Chain Block Coding and Cipher Feedback.

3.2.1 Description of DES

It is a 16-round Feistel cipher of 64 block length. It encrypts plaintext of 64 lengths using 56 bits key yielding a ciphertext bitstring of 64 lengths.

It contains a ciphertext bitstring 64 length.
It is symmetric, which is it has only one single for both sender and receiver.

The DES cipher of length O^i is divided into two equal halve of left L^i and right R^i.

The key K is denoted by K^i

Round function h $(L^i, R^i, and\ K^i) = (L^i, R^i)$

The left round lengths are generated using the function $(L^i = R^{i-1})$

The right lengths are generated using $R^i = L^{i-1} \oplus f\ (R^{i-1}, K^i)$
s
Again,

$$L^{i-1} = R^i \oplus f\ (L^i, K^i) \text{ and}$$

$$R^{i-1} = L^i$$

Before the encryption of 16 round of DES, the constant initial permutation function used to apply is denoted as

$$\text{IP}\ (p) = L^0\ R^0$$

$$q = IP^{-1}\ (L^{16}\ R^{16})$$

Where q is the ciphertext produced after applying IP^{-1} permutation function to the binary string $(LR)^{16}$ after 16 round of encryption.

The right and left bitstrings L^i, and R^i are usually swapped before the application of inverse permutation IP^{-1}

Note: The length of messages to be encrypted by DES must be 64 bits. For example, the message "That is my man". This plaintext message is 14 bytes (28 hexadecimal digits) long. Therefore, this message must be padded with some additional bytes at the tail end before the encryption. These extra bytes are discarded once the ciphertext is decrypted.

3.2.2 DES Examples

Example 3.2

1. Given an input 010110. Compute the equivalent outputs of an S-box in DES

Solution 3.2

1. G an input b_j = 010110 then

We denote the binary position as follows:

$b_1 = 0$, $b_2 = 1$, $b_3 = 0$, $b_4 = 1$, $b_5 = 1$, $b_6 = 0$

The algorithm to compute output is given below:

a. Combine the first and the last bit of the given bitstrings to find the row position of the bits in the S-box tables. We shall describe S-boxes in table 3.2.1.

   For example

   $b_1\ b_6$ represents the binary position of a row r of $S_j\ (0 \leq r \leq 3)$

b. The remaining bis $b_2b_3b_4b_5$ determine the binary representation of a column c of $S_j\ (0 \leq c \leq 15)$

c. $b_1b_6 = 00 = 0$. This means the row to be checked in that table is 0

d. $b_2b_3b_4b_5 = 1011 = B = 11$

e. The output $S_j\ (r, c) = O_j$. For example, $S_1\ (0{,}11) = 12$, $S_2\ (0{,}11) = 13$, $S_3\ (0{,}11) = 7$, $S_4\ (0{,}11) = 5$, $S_5\ (0{,}11) = 15$, $S_6\ (0{,}11) = 4$, $S_7\ (0{,}11) = 7$, $S_8\ (0{,}11) = 14$.
   That is from each table $S_1\ to\ S_8$, find the corresponding values in the row r and column c

   The outputs are therefore, $S_1 = 12$, $S_2 = 13$, $S_3 = 7$, $S_4 = 5$, $S_5 = 15$, $S_6 = 4$, $S_7 = 7$, $S_8 = 14$.

2. Given plaintext "Your lip" with Key k = 122346688AABCDD1
Using two new key K1 and K2

Solution

Given plaintext "Your lip"

Key k = 122346688AABCDD1

We want to find $K_1 \; and \; K_2$ for encryption and encrypt the message "Your lip"

Step 1:

Table 3.3: Represent the Plaintext in Hex ASCII Equivalent and Key in Bitstrings

Y	O	u	r		L	i	p
59	6F	75	72	20	6C	69	70

The equivalent Hex value of Plaintext is

59 6F 75 72 20 6C 69 70

Note: The message plaintext should be padded if it is less than 64 bits with additional bytes at the end of the plaintext before the encryption. The additional bits should be discarded after the decryption to yield accurate values.

Key K = 122346688AABCDD1

Table 3.4: Key Representation in Binary

Key Hex	1	2	2	3	4	6
Binary	0001	0010	0010	0011	0100	0110
Key Hex	6	8	8	A	A	B
Binary	0110	1000	1000	1010	1010	1011
Key Hex	C	D	D	1		
Binary	1100	1101	1101	0001		

K = 0001 0010 0010 0011 0100 0110 0110 1000 1000 1010 1010 1011 1100 1101 1101 0001

Step 2:

Create 16 subkeys, each of which is 48 bits long using table $P_{CD} - 1$

$P_{CD} - 1$ **Table**

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

In other to create 16 subkeys using Pc-1 table, the 64-bit key is permuted according to the following:

The elements (bit) in the key K are numbered 1 to 64. Since the first value in the table is "57", it means that the 57th bit of the given key **K** replaces the first bit of the permuted key **K**+. The 49th bit of the given key replaces the second bit of the permuted key. The 12th bit of the given key replaces the second to the last bit of the permuted key while the 4th bit of the given key is the last bit of the permuted key. Observe: only 56 bits of the original key appear in the permuted key.

After applying table $P_{CD} - 1$ on the key K, K becomes

$K - P_{CD} - 1$ = 1111 0000 1100 1100 0010 1010 0000 0011 0111 0100 0100 0111 1000 0001

Step 3:

Split $K - P_{CD} - 1$ into 2 halves $C_0 \, and \, D_0$ yielding:

C_0 = 1111 0000 1100 1100 0010 1010 0000
D_0 = 0011 0111 0100 0100 0111 1000 0001

Step 4:

Using C_0 and D_0, create sixteen blocks $C_n \, and \, D_n$, where $1 \leq n \leq 16$ with each pair of blocks $C_n \, and \, D_n$ forms from the previous pairs $C_{n-1} \, and \, D_{n-1}$ respectively.

Note: if n = 1, 2, 9, 16, then shift is 1 bit to the left else it is 2 bits shift.

Therefore,

C_1 = 111 0000 1100 1100 0010 1010 00001
D_1 = 011 0111 0100 0100 0111 1000 00010

C_2 = 1100 0011 0011 0000 1010 1000 0001
D_2 = 1101 1101 0001 0001 1110 0000 0100

C_3 = 0000 1100 1100 0010 1010 0000 1111
D_3 = 0111 0100 0100 0111 1000 0001 0011

C_4 = 0011 0011 0000 1010 1000 0011 1100
D_4 = 1101 0001 0001 1110 0000 0100 1101

C_5 = 1100 1100 0010 1010 0000 1111 0000
D_5 = 0100 0100 0111 1000 0001 0011 0111

We stop at $C_5\ and\ D_5$ since we only need K1 and K2 for encryption and Kn is given as:

$$K_n = P_{CD} - 2\ (C_n D_n)$$

K1 = $P_{CD} - 2\ (C_1 D_1)$, K2 = $P_{CD} - 2\ (C_2 D_2)$, K3 = $P_{CD} - 2\ (C_3 D_3)$.

$P_{CD} - 2$ Table

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Therefore,

K1 = 0000 1011 0000 0010 0100 0111 1011 1110 0111 0000 1010 0000

K2 = 0110 1001 0010 0110 0101 1001 0000 1011 0110 1000 1010 0111

K3 = 0100 0101 1101 0100 1000 1000 0110 0110 0100 1001 1001 0001

Table 3.2.1: S1 to S8 S-box Tables

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15 1 8 14 6 11 3 4 9 7 2 13 12 0 5 10
3 13 4 7 15 2 8 14 12 0 1 10 6 9 11 5
0 14 7 11 10 4 13 1 5 8 12 6 9 3 2 15
13 8 10 1 3 15 4 2 11 6 7 12 0 5 14 9

S3

10 0 9 14 6 3 15 5 1 13 12 7 11 4 2 8
13 7 0 9 3 4 6 10 2 8 5 14 12 11 15 1
13 6 4 9 8 15 3 0 11 1 2 12 5 10 14 7
1 10 13 0 6 9 8 7 4 15 14 3 11 5 2 12

S4

7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15
13 8 11 5 6 15 0 3 4 7 2 12 1 10 14 9
10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4
3 15 0 6 10 1 13 8 9 4 5 11 12 7 2 14

S5

2 12 4 1 7 10 11 6 8 5 3 15 13 0 14 9
14 11 2 12 4 7 13 1 5 0 15 10 3 9 8 6
4 2 1 11 10 13 7 8 15 9 12 5 6 3 0 14
11 8 12 7 1 14 2 13 6 15 0 9 10 4 5 3

S6

12 1 10 15 9 2 6 8 0 13 3 4 14 7 5 11
10 15 4 2 7 12 9 5 6 1 13 14 0 11 3 8
9 14 15 5 2 8 12 3 7 0 4 10 1 13 11 6
4 3 2 12 9 5 15 10 11 14 1 7 6 0 8 13

S7

4 11 2 14 15 0 8 13 3 12 9 7 5 10 6 1
13 0 11 7 4 9 1 10 14 3 5 12 2 15 8 6
1 4 11 13 12 3 7 14 10 15 6 8 0 5 9 2
6 11 13 8 1 4 10 7 9 5 0 15 14 2 3 12

S8

13 2 8 4 6 15 11 1 10 9 3 14 5 0 12 7
1 15 13 8 10 3 7 4 12 5 6 11 0 14 9 2
7 11 4 1 9 12 14 2 0 6 10 13 15 3 5 8
2 1 14 7 4 10 8 13 15 12 9 0 3 5 6 11

Step 5:

Perform an initial permutation IP of the 64 bits of the plaintext

T = "Your lip" = 596f7572206c6970

T = 0101 1001 0110 1111 0111 0101 0111 0010 0010 0000 0110 1100 0110 1001 0111 0000

Using IP table:

IP Table

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Table 3.5: Plaintext Permutation Using IP(T) Table

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
	0	1	0	1	1	0	0	1	0	1	1	0	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	0	1	0
IP(T)	1	1	1	0	1	1	1	1	1	0	0	0	1	1	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	1	1
	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
	0	0	1	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1	0	1	0	0	1	0	1	1	1	0	0	0	0
IP(T)	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1	0	0	0	1	1	0	0	0	0	1	0	1	0

IP (T) = 1 1 1 0 1 1 1 1 1 0 0 0 1 1 0 1 0 0 1 0 0 1 1 0 0 1 0 0 0 0 1 1 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 1 0

Again, divide IP(T) into L_0 *and* R_0 yielding

L_0 = 1110 1111 1000 1101 0010 0110 0100 0011
R_0 = 0000 0000 1111 1110 0110 0011 0000 1010

But $L_n = R_{n-1}$ and

$$R_n = L_{n-1} + f(R_{n-1}, K_n)$$

Therefore $L_1 = R_0$

Using the E-Table Selection table on R_n

E-BIT-SELECTION TABLE

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

As discussed in previous P tables, the first two bits of E(R_{n-1}) are the bits in positions 32, 1 of R_{n-1} while the last 3 bits of E(R_{n-1}) are the bits in positions 31, 32, and 1.
Again in the *f* calculation, XOR the output E(R_{n-1}) with the key K_n *and find* K_n + E(R_{n-1}).

For K_1 , $E(R_0)$, we have

K1 = 0000 1011 0000 0010 0100 0111 1011 1110 0111 0000 1010 0000

R_0 = 0000 0000 1111 1110 0110 0011 0000 1010
$E(R_0)$ = 000000 000001 011111 111110 001100 000110 100001 010100

Table 3.6: R_0 Permutation Using E-selection Bit Table

	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16
	1	1	1	1	1	1	1	0
	17	18	19	20	21	22	23	24
	0	1	1	0	0	0	1	1
	25	26	27	28	29	30	31	32
	0	0	0	0	1	0	1	0

$E(R_0)$ =

K1 = 0000 1011 0000 0010 0100 0111 1011 1110 0111 0000 1010 0000
$E(R_0)$ = 000000 000001 011111 111110 001100 000110 100001 010100

K1+$E(R_0)$

Table 3.7: K1+$E(R_0)$ Using X-OR Operation

K1	0	0	0	0	1	0	1	1	0	0	0	0
$E(R_0)$	0	0	0	0	0	0	0	0	0	0	0	1
K1+$E(R_0)$	0	0	0	0	1	0	1	1	0	0	0	1
K1	0	0	1	0	0	1	0	0	0	1	1	1
$E(R_0)$	0	1	1	1	1	1	1	1	1	1	1	0

K1+$E(R_0)$	0	1	0	1	1	0	0	0	1	0	1	1
K1	1	0	1	1	1	1	1	0	0	1	1	1
$E(R_0)$	0	0	1	1	0	0	0	0	0	1	1	0
K1+$E(R_0)$	1	0	0	0	1	1	1	1	0	1	0	0
K1	0	0	0	0	1	0	1	0	0	0	0	0
$E(R_0)$	1	0	0	0	0	1	0	1	0	1	0	0
K1+$E(R_0)$	1	0	0	0	1	1	1	1	0	1	0	0

K1+$E(R_0)$ = 0000 1011 0001 0101 1000 1011 1000 1111 0100 1000 1111 0100

Using S-box technique in example 3.2,
L_0 = 1110 1111 1000 1101 0010 0110 0100 0011
$f(R_0, K_1)$ =>
K1+$E(R_0)$ = 000010 110001 010110 001011 100011 110100 100011 110100

B1 B2 B3 B4 B5 B6 B7 B8

For B1 = 000010

$b_1 = 0,\ b_2 = 0\ , b_3 = 0\ , b_4 = 0, b_5 = 1, b_6 = 0$

f. $b_1 b_6 = 00 = 0$. Row r = 0

$b_2 b_3 b_4 b_5 = 0001 = B = 1$

$S_1\ (0,1) = 4$ = 0100

For B2,

$S_2\ (3,8) = 11 =$ 1011

For B3

$S_3\ (0,11) = 7 =$ 0111

B4

$S_4\ (1,5) = 15 =$ 1111

B5

$S_5\ (3,1) = 8 =$ 1000

B6

$S_6\ (2,10) = 4 =$ 0100

B7

$S_7\ (3,1) = 11 =$ 1011

B8

$S_8\ (2,10) = 10 =$ 1010

Therefore,

$S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8$ = 0100 1011 0111 1111 1000 0100 1011 1010

L_0 = 1110 1111 1000 1101 0010 0110 0100 0011

The final stage is to calculate function *f* through permutation P of the S-box output to obtain the final value *f*:

The permutation P, according to the table P defined in the following table and produced 32-bit output from a 32-bit input by permuting the input block bits.

Therefore:

$f(R_0, K_1)$ = P (0100 1011 0111 1111 1000 0100 1011 1010) =

P

	16	7	20	21			
	29	12	28	17			
	1	15	23	26			
	5	18	31	10			
	2	8	24	14			
	32	27	3	9			
	19	13	30	6			
	22	11	4	25			

	1	2	3	4	5	6	7	8
	0	1	0	0	1	0	1	1
	9	10	11	12	13	14	15	16
	0	1	1	1	1	1	1	1
	17	18	19	20	21	22	23	24
	1	0	0	0	0	1	0	0
	25	26	27	28	29	30	31	32
	1	0	1	1	1	0	1	0

yields

$f(R_0, K_1)$ = P (0100 1011 0111 1111 1000 0100 1011 1010) =

1100 1111 0100 1011 1100 0100 0100 1101

$R_1 = L_0 + f(R_0 , K_1)$

L_0	1	1	1	0	1	1	1	1	1	0	0	0
$f(R_0 , K_1)$	1	1	0	0	1	1	1	1	0	1	0	0
R_1	0	0	1	0	0	0	0	0	1	1	0	0
L_0	1	0	1	1	1	1	0	0	0	1	0	0
$f(R_0 , K_1)$	1	1	1	1	1	0	0	0	0	1	0	0
R_1	0	1	0	0	0	1	0	0	0	0	0	0
L_0	0	1	0	0	1	1	0	1				
$f(R_0 , K_1)$	1	0	1	1	1	0	1	0				
R_1	1	1	1	1	0	1	1	1				

R_1 = 0010 0000 11000 1000 1000 0001 1110 111

In the round 2, $L_2 = R_1$, which is the immediate calculated block, and then we must calculate $R_2 = L_1 + f(R_1, K_2)$, and so on for n rounds depending on

the number of keys to be generated. At the end of the L_n and R_n. We then *reverse* the order of the two blocks into the 64-bit block

$$L_nR_n$$

$$L_2 = R_2 - 1 = R_1$$

$$R_2 = L_1 + f(R_1, K_2)$$

$R_1 = L_2$

Now $f(R_1, K_2) = (f(K_2 + E(R_1)$

	1	2	3	4	5	6	7	8
R_1	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16
R_1	1	1	1	1	1	1	1	0
	17	18	19	20	21	22	23	24
R_1	0	1	1	0	0	0	1	1
	25	26	27	28	29	30	31	32
R_1	0	0	0	0	1	0	1	0

$E(R_1)$ = 000000 000001 011111 111100 001100 000110 100001 010100
K2 = 0110 1001 0010 0110 0101 1001 0000 1011 0110 1000 1010 0111

Now, $(f(K_2 + E(R_1)$ =>

K2	0	1	1	0	1	0	0	1	0	0	1	0
$E(R_1)$	0	0	0	0	0	0	0	0	0	0	0	1
K1+$E(R_1)$	0	1	1	0	1	0	0	1	0	0	1	1
K2	0	1	1	0	0	1	0	1	1	0	0	1
$E(R_1)$	0	1	1	1	1	1	1	1	1	1	0	0
K1+$E(R_1)$	0	0	0	1	1	0	1	0	0	1	0	1
K2	0	0	0	0	1	0	1	1	0	1	1	0
$E(R_1)$	0	0	1	1	0	0	0	0	0	1	1	0
K1+$E(R_1)$	0	0	1	1	1	0	1	1	0	0	0	0
K2	1	0	0	0	1	0	1	0	0	1	1	1
$E(R_1)$	1	0	0	0	0	1	0	1	0	1	0	0
K1+$E(R_1)$	0	0	0	0	1	1	1	1	0	0	1	1

$((K_2 + E(R_1)$ = 0110 1001 0011 0001 1010 0101 0011 1011 0000 0000 1111 0011

But we apply the S-Box table to get $(f(K_2 + E(R_1)$ = >

Therefore $(f(K_2 + E(R_1)$ =>

$(f(K_2 + E(R_1)$ = 011010 010011 000110 100101 001110 110000 000011 110011

B1 B2 B3 B4 B5 B6 B7 B8

For B1 = 011010

$b_1 = 0,\ b_2 = 1\ , b_3 = 1\ , b_4 = 0, b_5 = 1, b_6 = 0$

$b_1 b_6 = 00 = 0$. Row r = 0

$b_2 b_3 b_4 b_5 = 1101 = C = 13$

$S_1\ (0,13) = 9$ = 1001

B2

$S_2\ (1,9) =$ = 0000

B3

$S_3\ (0,3) = 14 =$ 1110

$S_4\ (3,2) = 0\ =$ 0000

$S_5\ (0,7) = 6 =$ 0110

$S_6\ (2,8) = 7 =$ 0111

$S_7\ (1,1) = 0 =$ 0000

$S_8\ (3,9) = 12 =$ 1100

Therefore,

$S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8$ = 1001 0000 1110 0000 0110 0111 0000 1100

$f(K_2 + E(R_1)$ = 1001 0000 1110 0000 0110 0111 0000 1100

Now apply P table on $f(K_2 + E(R_1)$ yielding

	1	2	3	4	5	6	7	8
$f(K_2 + E(R_1)$	1	0	0	1	0	0	0	0
	9	10	11	12	13	14	15	16
$f(K_2 + E(R_1)$	1	1	1	0	0	0	0	0
	17	18	19	20	21	22	23	24
$f(K_2 + E(R_1)$	0	0	0	0	0	1	1	0
	25	26	27	28	29	30	31	32
$f(K_2 + E(R_1)$	0	0	0	0	1	1	0	0

$f(K_2 + E(R_1)$ = 0000 1000 1010 0001 0000 0001 0010 1110

$L_1 = R_0$ = 0000 0000 1111 1110 0110 0011 0000 1010

But $R_2 = L_1 + f(R_1, K_2)$ gives

L_1	0	0	0	0	0	0	0	0	1	1	1	1
f (R_1, K_2)	0	0	0	0	1	0	0	0	1	0	1	0
R_2	0	0	0	0	1	0	0	0	0	1	0	1
L_1	1	1	1	0	0	1	1	0	0	0	1	1
f (R_1 , K_2)	0	0	0	1	0	0	0	0	0	0	0	1
R_2	1	1	1	1	0	1	1	0	0	0	1	0
L_1	0	0	0	0	1	0	1	0				
f (R_1 , K_2)	0	0	1	0	1	1	1	0				
R_2	0	0	1	0	0	1	0	0				

R_2= 0000 1000 0101 1111 0110 0010 0010 0100, where
$L_2 = R_1$ = 0010 0000 1100 0100 0100 0000 1111 0111

Therefore,

$R_2L_2 =$ 0000 1000 0101 1111 0110 0010 0010 0100 0010 0000 1100 0100 0100 0000 1111
0111

Applying IP^{-1} on R_2L_2 yields

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

	1	2	3	4	5	6	7	8	9	10	11	12
R_2L_2	0	0	0	0	1	0	0	0	0	1	0	1
	13	14	15	16	17	18	19	20	21	22	23	24
R_2L_2	1	1	1	1	0	1	1	0	0	0	1	0
	25	26	27	28	29	30	31	32	33	34	35	36
R_2L_2	0	0	1	0	0	1	0	0	0	0	1	0
	37	38	39	40	41	42	43	44	45	46	47	48
R_2L_2	0	0	0	0	1	1	0	0	0	1	0	0
	49	50	51	52	53	54	55	56	57	58	59	60
R_2L_2	0	1	0	0	0	0	0	0	1	1	1	1
	61	62	63	64								
R_2L_2	0	1	1	1								

$IP^{-1}(R_2L_2)$ = 00010010 00010110 00110011 01010000 00010010 10000111 00111110 00100010

$IP^{-1}(R_2L_2)$s a ciphertext which is in hexadecimal

C = 1216335012873E22

3.2.3 DES Issues

DES is susceptible to a brute-force attack. This implies after many trials of up to 2^56 possible keys; the ciphertext could be decrypted into an intelligible plaintext message.

3.3 RSA (Rivest – Shamir - Adleman)

RSA is the first public-key cryptosystem having full-scaled features. RSA became popular ought to the fact that some previous cryptosystem such as Diffie Hellman can only exchange random bits and not other specific information. RSA is discussed elaborately in module two.

Application Scenario 5

Examples of block ciphers: AES, DES

block size	key size
3DES	64	168
AES	128	128, 192, or 256

3DES is slower and less secure than AES, but still very useful. AES remains one of the best ciphers.

Block ciphers shortcomings

Problem 1

Suppose Ade often sends certain messages to Ola and each time he sends these message m, Oye will see the same ciphertext E(k,m). To see this ciphertext overtime may be enough for Ola to guess something about Ade and Ola's communication. Suppose that every time Oye sees ciphertext c, the value of stock appreciates the next day, and every time he sees ciphertext c', the value of a stock depreciates. This invariably gives Oye some important information, even without obtaining the messages being sent. This problem is as a result of the non-randomness of the message being sent. This problem can be solved by introducing "initialisation vectors" into block cipher.

Problem 2
Block cipher only operates over small blocks of data. For example, AES's block cipher alone can only encrypt messages up to 128 bits long. This cipher encrypts messages in multiple "modes" by breaking an arbitrarily sized input into segments of length equal to the block size.

4.0 Self-Assessment Exercise(s)

1. Given plaintext "Our case"

Key k = 132246688AABCDD1

The encryption of the plaintext message "Our case" using key K is:

A. 659AB7876A8CDC49
B. 689BA7876A8CDC49
C. 689AB7876A8DCC49
D. 689AB7876A8CDC49

Correct Answer is D

2. DES is susceptible to what type of attack?

A. Brute-force attack
B. Dictionary attack
C. MITM attack
D. Malware attack

Assignment:
Examine what should be done if the length of the message to be encrypted by DES or AES is not up to 64 bits length.

5.0 Conclusion

In this unit, you have learnt some block cipher like Advanced encryption standard (AES) and Data encryption standard (DES). The unit also introduced you to RSA cryptosystem. The benefit of AES and DES as block ciphers were examined over stream ciphers. The algorithms, features, challenges, and merits of these ciphers over other ciphers. The unit examined some examples of problems related to AES and DES and encrypted plaintext using these algorithms.

6.0 Summary

This unit discussed the concepts of block ciphers and gave examples of the public cryptosystem. In addition, it examined two types of block cipher of asymmetric cryptography, namely AES and DES. It also introduced a sample of public cryptosystem or asymmetric cryptosystem called RSA. Examples were given with solutions to illustrate the concepts of AES and DES implementation.

7.0 References/Further Reading

Adebayo, O. S. (2019). "CSS 723: Modern Cryptography II," Lecture Note, (Unpublished)

Douglas, R. S. (1995). *Cryptography: Theory and Practice*. Boca Raton: CRC Press. Retrieved on 25-12-2019 from http://www.ksom. res.in/files/RCCT-2014-III-CM/CryptographyTheoryandpractice(3ed).pdf.

Grabbe, J. O. (n.d.) The DES Algorithm Illustrated. Retrieved from http://www.notesale.co.uk/more-info/78947/The-DES-Algorithm-Illustrated-by-J.-Orlin-Grabbe.

https://www.cs.umd.edu/class/spring2015/cmsc414/writeups/symmetric-key.pdf

Unit 2: RC4 and Key Exchange Management

Contents

1.0 Introduction

In this unit, you will learn the principle and process of the stream cipher. You will also learn the management of encryption secret key exchange techniques. The unit shall discuss RC4 as a variable key size type of stream cipher designed with byte-oriented fundamentals. The unit shall also examine the features, process, and security implications of RC4. Examples of how a key can be generated and distributed for encryption of messages over an insecure communication channel will be given for your adequate understanding.

2.0 Intended Learning Outcomes (ILOs)

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

- define Rivest cipher 4 (RC4)
- describe the processes of stream cipher and keys.
- differentiate between Key Agreement and Key Distribution.

3.0 Main Content

3.1 RC4 (Rivest Cipher 4)

RC4 is a variable key-size stream cipher designed with byte-oriented fundamental. It is designed by Ron Rivest primarily to ensure the security of RSA. The RC4 algorithm is consists of two main components viz: the pseudo-generation algorithm (PRGA) and key scheduling algorithm (KSA). Its internal permutation architecture contains 8-bit words. The secret key accepted by RC4 is between 5 and 32 bytes length. This initial key generates the expanded key K of length 256 bytes using repetition approach.

Given the length L byte of the secret key K, where $5 \leq l \leq 32$ bytes, the expanded key then generated with repetitive technique $K(m) = K\ (m\ mod\ N)$ where $0 \leq m \leq N-1$. The initial permutation generated by the key scheduling algorithm (KSA) serves as the input to PRGA. The PRGA generates the remaining keys for its operation. Fig. 3.1 displays the structure of RC4 encryption algorithm.

3.1.1 Description of RC4

RC4 consists of two algorithms, namely KSA and PRGA, as presented below. The KSA generates the initial input while PRGA generates the mainstream keys for the entire operation. Figure 4.1 describes the process of Rivest Cipher 4.

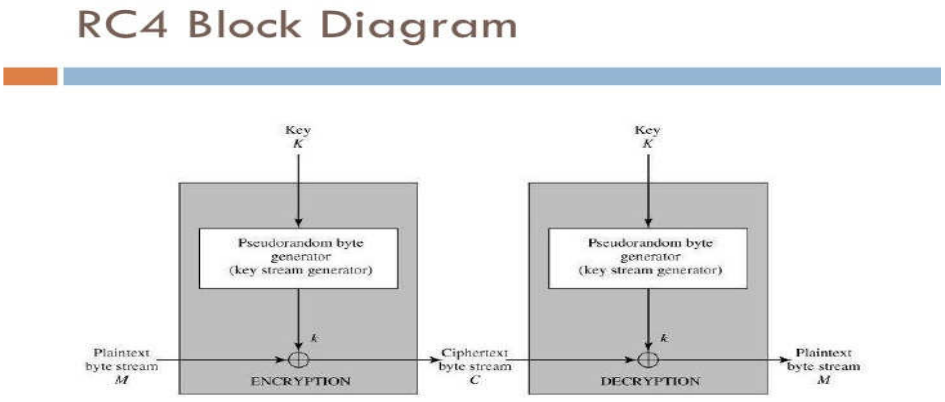

Fig 3.1: RC4 Encryption Algorithm

RC4 KSA Algorithm

It has 256 rounds of bytes generation

$m = 0$

$n = 0$

$K(m) = K\ (m\ mod\ N)$

$n = n + R(m) + S\ (m)$

Swap $S(m)\ with\ S(n)$

$m = m + 1$

RC4 PRGA

$m = 0$

$n = 0$

$m = m + 1$

$n = n + S\ (m)$

Swap $S(i)\ with\ S(j)$

A = S $[S(m) + S(n)]$

For all rounds $rn\ where\ n = 1, 2, 3, \ldots$ of RC4, the output O_r is denoted using $m_r\ and\ n_r$

The byte extraction output $a_r = S_r\ (m_r) + S_r\ (n_r)$

S_0^K is used to represent the initial permutation of the key scheduling algorithm while S_0 stands for PRGA initial permutation.

3.1.2 RC4 Pseudocode

RC4 encryption algorithm includes the following orders of execution

a. Acquire the data to be encrypted
b. Selected key
c. Establish two string arrays
d. Initialize KSA array with 0 to 255 numbers
e. Fill the PRGA array with the selected key
f. Randomize the first array depending on the array of the key.
g. Randomize the first array within to generate the final keystream
h. XOR the final keystream with the data to be encrypted to give ciphertext.

3.1.3 Features of RC4

RC4 is a symmetric stream cipher.
It has a variable key length of between 5 to 256 bytes.
It has quick execution in software
It is widely adopted in secured communications such as in the encryption of traffic to and from secure web sites using the SSL protocol.

3.1.4 Security of RC4

There are two basic attacks in which RC4 is susceptible to:

1. Initial output bias
   This problem is associated with the generation of the initial key for the RC4 algorithm input. That is, the distribution on each of 256 bytes is not uniform such that with the encryption of a single plaintext below 2^{30} random keys, the probability of recovering the first 128 bytes, is very high. This attack is common on the web where the cookie is being embedded in the first few bytes of a message.

2. Bias in the random stream generator
   This type of attack is associated with WEP encryption, which uses RC4 stream cipher. The objective of WEP is to enable members of the wireless network shares a long term key. The WEP is incorporated with 24-bits IV collisions. However, this IV collisions can easily be detected by an eavesdropper.

3.2 Principle of Stream Cipher

A stream cipher encrypts a byte of plaintext at a time. That is, it does not store data in a disk or memory unlike block cipher for later execution. In the structure of stream cipher, a key is an input into a pseudorandom bit generator that generate a stream of random 8-bit numbers. Using the bitwise exclusive-OR (XOR) operator, the output of a generator known as keystream combines one byte at a time with plaintext stream. Fig. 3.2 and 3.3 depict the structure and model of stream cipher respectively.

Example of exclusive OR operation

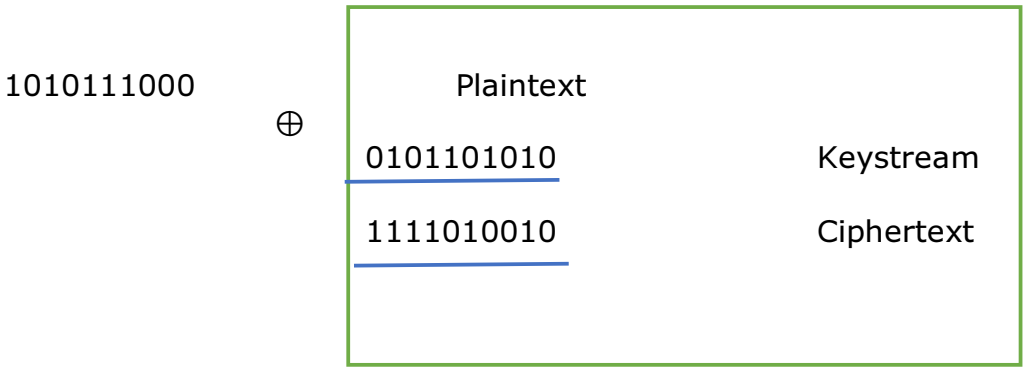

Fig 3.2 Stream Cipher Structure

3.2.1 Process of Stream Cipher

A stream cipher is using One Time Pad (OTP), which has a truly random bit-string length as the plaintext. It uses bitwise exclusive XOR operation on plaintext to generate the ciphertext.

A stream cipher can be defined as a deterministic algorithm with an encryption function.

$f: \{0,1\}^m \times \{0,1\}^n \rightarrow \{0,1\}^j$ maps with a set of key K,
$K = K_q, q = 1,2, \ldots m$ and

$n - bit$ public initialiser $I = I_q, q = 1,2, \ldots n$ to $n - bit$
keystream $Z = Z_q, q = 1,2, \ldots j$

The keystream Z_q is then bitwise XOR with plaintext P

$P = P_q, q = 1,2, \ldots j$ to yield ciphertext C

$C = C_q, q = 1,2, \ldots j$

Practically, the length j of a generated keystream is greater than the length m of a secret key K.

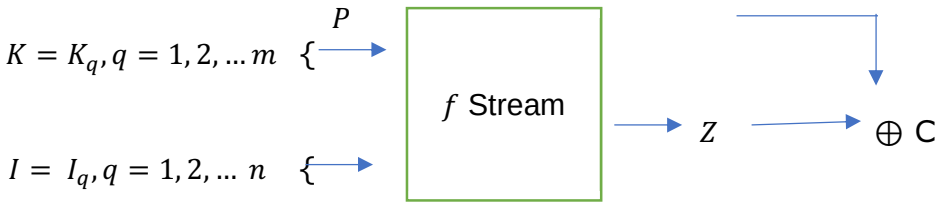

Fig 3.3 Key-I Stream Cipher Model

3.3 Key Exchange Management

The benefits of public-key cryptography like RSA and ElGamal are enormous in terms of security since their keys are public and there is no need for exchange of key. However, public-key cryptography spends longer time for encryption than private cryptography like DES and AES. Therefore, it is a conventional practice to use the private key or asymmetric cryptography which uses the private key to encrypt long messages. The key exchange management involves key distribution and key agreement.

3.3.1 Key Agreement

This key exchange mechanism is where two or more entities mutually create and agree on a secrete key through communication over a public

channel. In this key agreement system, the input function or variable provided by the parties determine the value of key use for encryption.

3.3.2 Key Distribution

This is a key exchange mechanism where one party (sender) selects a secret key and transmits to another (receiver(s)).

Attacker's objective

1. To deceive sender S and receiver R to accept an invalid key as valid.
2. To ensure S and R agree they have exchanged the key when they have not.

3.3.2.1 Key Pre-distribution Model

The model of key exchange is developed considering the number of users on an insecure network, with the trusted party responsible for the verification of user's identity, as well as the selection, and transmission of agreed key. There is a need to guard against the passive and active attackers (say Ayo) since the network is invariably insecure. The activities of attackers are numerous. Ayo may want to intercept and modify the communicating message, keep the message for reuse, impersonate the users on the network among other attacks.

Let the number of users on an insecure network = n

Verifying Trusted Authority = VTA

Key predistribution is between pair (S, R) users

VTA selects a key $K_{R,S} = K_{S,R}$ randomly

VTA sends securely to users $S\ and\ R$

User stores keys $(K_1, K_1 \ldots K_n) = n - 1$ keys

VTA generates and transmits a sum of $\binom{n}{2}$

3.3.2.2 Key Distribution (Blom's Scheme)

Given a network user of n numbers
Let selects key from a finite field $\mathbb{Z}_p$ where $p \geq n$

Let k be an integer $\ni\ n - 2 \leq k \leq 1$, k is the largest value for unconditional security

VTA transmits $k + 1 \in \mathbb{Z}_p$

Each user computes a key $K_{R,S} = K_{S,R}$ randomly

VTA sends to users $R\ and\ S$ through a secure medium

When k = 1, VTA transmits two variables of $\mathbb{Z}_p$ to R and S over a secure channel

Third-party V cannot determine the communication between $K_{R,S}$ if V ≠ R, S

Blom's Example

Example 3.3

Given a three insecure communication network users R, S, and V, p = 19 with public variables $w_R = 14$, $w_S = 9$, $w_V = 1$. If VTA selects a = 10, b = 9, c = 4. Compute the keys for user R, S and V.

Solution 3.3

We compute the polynomial function $f\ and\ g$

$f\ (i,j) = a + b(i + j) + 2ij\ mod\ p$ and

$$gr(i) = b + (bc)i\ mod\ p$$

$$gs(i) = (a - b) + (c^2)i\ mod\ p$$

$$gv(i) = (a + b) + (bc)i\ mod\ p$$

If key k = 1, then

$K_{R,S} = gr(w_S) = b + (bc)(w_S)$ = 9 + (9 * 4)*9 mod 19 = 10

$K_{S,V} = gs(w_V) = (a - b) + (c^2)i = (10 - 9) + (4^2) * 1\ mod 19 =$ 17

$K_{V,S =} gv(w_S) = (a + b) + (bc)i = (10 + 9) + (10 * 9) * 9\ mod\ 19$ = 19 + 90*9 mod 19 = 12

Therefore, the keys for user R, S, and V are:

$K_{R,S}$ = 10
$K_{S,V}$ = 17
$K_{V,S =}$ 12

respectively

3.3.2.3 The Diffie Hellman Key Exchange Protocol

This is one of the key exchange protocol that adopts the asymmetric key fundamentals to avoid the physical transfer of key between communicating parties.

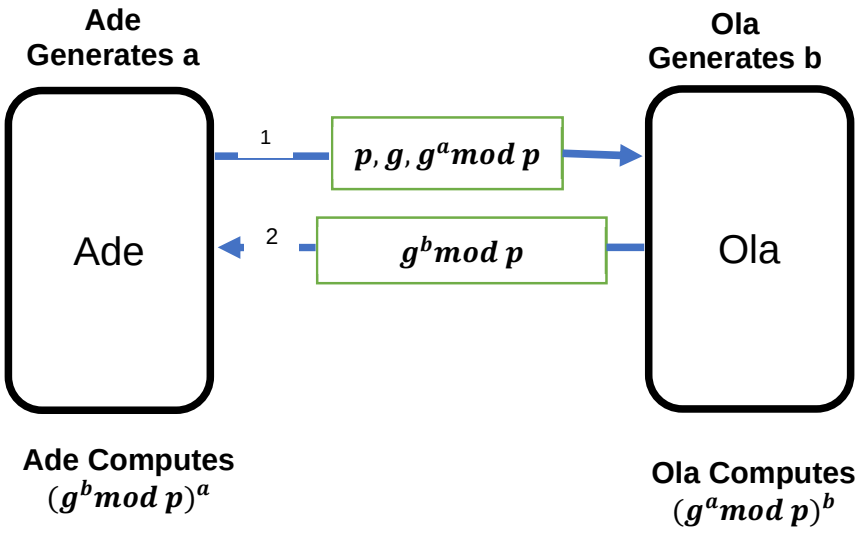

Fig 3.4: Diffie Hellman Key Exchange

Algorithms

Let Fq be a finite field in $R1$ and also an element $g \in F_g$ which generate a group of a large order. To simplify the algorithm, consider the case g field F_p where P is a large prime of several hundreds of digits. Ade and Ola may test the primality of P, using the primality algorithm with the complete factorisation of P -1. A similar description is provided in figure 3.4.

D-H step

Ade and Ola agree on a large prime p and a primitive integer of p; i.e. g mop p

Ade and Ola distribute the two prime numbers $p\ and\ g$ as public – key which could be known to third parties.

Ade picks a secret integer a unknown to anyone; Ola picks an integer b.

They both compute the following and exchange them

$$A \equiv g^a \pmod{p}; \text{and}$$
$$B \equiv g^b \pmod{p}$$

They both compute the positive of A & B with their secret keys i.e.

$$A^i \equiv B^a \pmod{p} \text{ and}$$
$$B^i = A^b \pmod{p}$$

i.e. $A^1 \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g)^a \equiv A^b = B^i$

Example 3.4
Given that Ade and Ola use a Prime P = 841, a primitive root g = 517. Ade chooses secret key a = 357 and Ola chooses a secret key b = 981. Compute their exchange keys.

Solution 3.4
Give P = 841, g = 517
Ade chooses a=357 and compute

$$A = 517^{357} \, (mod \, 357) = 639$$

Ola chooses b=981 and compute

$$B = 517^{981} \, (mod \, 841) = 401$$

So, A = 639, and B = 401.
Then $A^1 = B^1 = 401^{357}$ mod 841 = 639^{981} (mod 841) = 175
= $g^{ab} = 517^{357x981}$ (mod p) = 175

Note: The values in the above example are small and cannot guarantee the security of their message because the third party can easily compute all possible powers of 517 modulo 841. Therefore, a very large p which has approximately 1000 bits (1000, p ≈ 2^{1000}) and a primitive element g of p such that g ≈ p/2 should be used.

3.3.2.3.1 Security of Diffie-Hellman

The Security of D-H lies on the degree of difficulty of computing Diffie-Hellman Problem (DHP), that is the problem of computing the value of g^{ab} (mod p) from the known values. g^a (mod p) & g^b (mod p). So, if James can intercept The values A= g^b and B = g^b, then it is easy to compute their share key g^{ab}.

3.3.2.4 Diffie-Hellman Key Predistribution

This key pre-distribution is a modification of a famous Diffie-Hellman key exchange protocol, which is computationally secure as long as the related discrete logarithm problem is intractable.

Description of Diffie-Hellman Key Pre-distribution (Algorithm)

Let $\mathbb{Z}_p$ be defined over a finite group where p is a prime

Let a be a primitive element of $\mathbb{Z}_p$

p and i are public elements

$ID(R)$ implies information identifier for individual user R

Secrete exponent for the individual user c_R, $0 \leq c_R \leq p-2$,

Public value $d_R = i^{c_R} \, mod \, p$

Verification trusted authority has verification function = Ver_{VTA} and

signing algorithm = Sig_{VTA}

User R related information need be authenticated using an authenticated certificate which is issued and signed by the VTA.

Therefore, individual user R have a certificate

$C\,(R) = (ID\,(R), c_R, \, Sig_{VTA}\,(ID\,(R), d_R))$

The common key is therefore generated from R and S as:

$K_{R,S} = i^{c_R c_S} \, mod \, p$

Diffie-Hellman Key Predistribution Example

Example 3.5

Given p, a prime = 23305, i, a primitive root modulo p = 2, assuming user Rimi (R) selects $c_R = 2374$, Sanni (c_S) $= 1898$. Compute the common key for user Rimi (R) and Sanni (S).

Solution 3.5

R computes

$d_R = i^{c_R} \, mod \, p = 2^{2374,} mod \, 23305 = 1464$

S then computes

$d_S = i^{c_S} \, mod \, p = 2^{1898} mod \, 23305 = 15349$

R computes

$K_{R,S} = i^{c_R c_S} \, mod \, p = (i^{c_S})^{c_R} \, mod \, p = (d_S)^{c_R} \, mod \, p = 15349^{2374} \, mod \, 23305 = 14401$

S then compute

$K_{R,S} = (i^{c_R})^{c_S} \, mod \, p = (d_R)^{c_S} \, mod \, p =$

$1464^{1898}\ mod\ 23305 = 14401$

Therefore, the key is 14401

Assignment

Ade and Ola agree on a key k based on a block cipher.
Ade wants to send message m; she computes C = E(k,m) and sends C.
Ola obtains C and computes D(k,C) = m to obtain the original message m.

1. What is kept secret?

   Answer

   The ONLY thing about a block cipher that is kept secret is the shared key k.

2. What is publicly known

   Answer

   The encryption and decryption algorithms are publicly available.

4.0 Self-Assessment Exercise(s)

1. The objectives of network adversary include the following except
   A. To deceive sender A and receiver O to accept an invalid key as valid.
   B. To ensure A and O agree they have exchanged the key when they have not.
   C. To encourage sender S and receiver R to accept an invalid key as valid.
   D. To ensure A and O disagree they have exchanged the key; when they have not.

   The correct answer is D.

Suppose a prime number, p = 23305, a primitive root i = 4, assuming user Rimi (R) selects $c_R = 1372$, Sanni $(c_S) = 1436$. The common key for user Rimi (R) and Sanni (S) is:

E. 31441
F. 3441
G. 34141

H. 3414

The correct answer is B.

2. The correct order of execution RC4 encryption algorithm is

A. Acquire the data to be encrypted, selected key, Establish two string arrays, Initialise KSA array with 0 to 255 numbers, Fill the PRGA array with the selected key.
B. Acquire the data to be encrypted, selected key, Establish two string arrays, Fill the PRGA array with the selected key, Initialise KSA array with 0 to 255 numbers.
C. Acquire the data to be encrypted, selected key, Fill the PRGA array with the selected key, Initialise KSA array with 0 to 255 numbers, establish two string arrays.
D. Initialise KSA array with 0 to 255 numbers, Acquire the data to be encrypted, selected key, Establish two string arrays, Fill the PRGA array with the selected key.

The correct answer is A.

5.0 Conclusion

In this unit, you have learnt the process and principle of the stream cipher. You have also learnt how the model of stream cipher influences its security and speed of the operation. In the unit, we also discussed the key management and key distribution strategies, RC4 as a variable key size type of stream cipher designed with byte-oriented fundamental. The unit x-rayed stream cipher as symmetric cryptography which encrypts the message in 8-bits of length. The unit also gave a difference between the stream and block cipher, including their benefits over each other. Besides, you have learnt the challenges of RC4 and its security implication. The unit concluded with examples of how a key can be generated and distributed for encryption of messages over an insecure communication channel.

6.0 Summary

This unit is dedicated to the stream cipher and RC4 type of stream cipher. RC4 is a variable key-size stream cipher designed with byte-oriented fundamental. RC4 algorithm consists of two main components viz: the pseudo-generation algorithm (PRGA) and key scheduling algorithm (KSA). The two basic attacks in which RC4 is susceptible to Initial output bias and Bias in the random stream generator. The key exchange management

involves key distribution and key agreement. The Security of D-H lies on the degree of difficulty of computing Diffie-Hellman Problem (DHP).

7.0 References/Further Reading

Douglas, S. (1995). Cryptography: Theory and Practice. CRC Press LLC.ISBN: 0849385210. Retrieved on 25-12-2019 from http://index-of.es/Cryptography/Cryptography%20Theory%20And%20Practice%20-%20Douglas%20Stinson.pdf.

https://www.cs.umd.edu/class/spring2015/cmsc414/writeups/symmetric-key.pdf

Sourav, S. G. (2013). "Analysis and Implementation of RC4 Stream Cipher." A thesis presented to the Indian Statistical Institute in fulfilment of the thesis requirement for the degree of Doctor of Philosophy in Computer Science. Applied Statistics Unit Indian Statistical Institute Kolkata, West Bengal, India, 2013. Retrieved from https://souravsengupta.com/pub/phd_thesis_2013.pdf.

Module 4: Signature Scheme

Unit 1 Security Requirement for Signature Scheme
Unit 2 Digital Signature Algorithm
Unit 3 Secure Hash
Unit 4 Steganography

Unit 1 Security Requirement for Signature Scheme

Contents

1.0 Introduction

In this unit, you will learn the concept of signature schemes and associated algorithms. You will also learn the security requirements for signature schemes and the benefits of this security scheme. The security of any signature scheme is conditional since an adversary can obtain either message or previous signature and use them to compute verification and thereby having a valid signature. Specifically, you will learn the undeniable

and fail-stop signature scheme and their security implications and mechanisms. You will also learn about the attack models and their examples on the signature schemes.

2.0 Intended Learning Outcomes (ILOs)

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

- explain the concepts of signature schemes
- process the treatment of special types of signature schemes such as undeniable and fail-stop signature schemes
- discuss the security necessity of a signature scheme.

3.0 Main Content

3.1 Signature Scheme

A signature scheme is a digital technique of authenticating an electronic document from signer to the verifier. The manual means of authenticating paper document is through comparison of signer signature with the one on the document. However, the problem of authenticating an electronic document arises due to the lack of an electronic signature. This problem was solved using various signature algorithms ranges from El-Gamal signature, Schnorr signature scheme, Digital signature scheme, Elliptic Curve DSA (ECDSA), RSA signature scheme and many more.

3.1.1 Generic Signature Algorithm

A Signature Scheme is a five-tuple (P, A, K, S, V), where each notation is described below.
P: denotes a finite set of possible messages
A: a finite set of signatures
K: the keyspace is a finite set of possible keys

For each $k \in K$, there is a signing algorithm $Sig_k \epsilon S$, and a corresponding verification algorithm $Ver_k \epsilon V$. Each $Sig_k: P \rightarrow A$ and $Ver_k: P \times A \rightarrow (true, false)$ are functions such that the following verification algorithm are satisfied for message $\mathrm{x} \in \mathrm{P}$ and signature $y \in A$ on the message p:

$$Ver\ (x, y) = \begin{cases} true\ if\ y = Sig_k \\ \\ false\ if\ y \neq Sig_k \end{cases}$$

Where a pair (x, y) with $x \in P$ and $y \in S$ is called a signed message

3.1.2 Categories of Signature Schemes

3.1.2.1 Undeniable Signature Scheme

An undeniable signature scheme is a scheme characterized by the feature of duplication prevention. That is, Ola cannot verify the signature of Ade without Ade's knowledge and awareness. Therefore, the signature sign by Ade can only be duplicated and distributed through his awareness; otherwise, it is impossible. In this case, Ade can as well deny the fact of signing his message. For the avoidance of such personal denial, undeniable signature incorporates a protocol called disavowal protocol through which Ade can establish the authenticity of his signature. If Ade has not participated in the disavowal protocol on the signature, then Ade can establish the signature is original and not forged.

3.1.2.2 Description of Undeniable Signature Scheme (USS)

The USS consists of three components:
1. Signing algorithm
2. Verification algorithm
3. Disavowal protocol

3.1.2.3 Undeniable Signature Cryptosystem

1. Signing Algorithm

Given a prime numbers p and q such that m = 2n + 1 and the discrete logarithm problem in $\mathbb{Z}_m$ is intractable.

Note: Intractable problems are problems in which there exist almost no algorithm to solve them.

Also, given $i \epsilon \mathbb{Z}_m^*$ an order p element.

Given 1 ≤ a ≤ n − 1, j = $i^a \, mod \, m$, G implies multiplicative subgroup of Z_m^*

With $m, i, and \, j$ denote public key and *a* as private key.

$\mathrm{K} = (m, i, a, j), \; p \epsilon G$ and $\mathrm{K} = (m, i, a, j)$: $i^a mod \, m$ then

$$q = \; Sig_k(p) = p^a mod \, m$$

2. Verification Protocol

For Ola to verify the signature of Ade with a pair (p, q),

Ola selected $r_1, r_2 \in \mathbb{Z}_m$ randomly, computes $s = q^{r_1} j^{r_2} \, mod \, m$ send it to Ade.

Ade computes $t = s^{a^{-1} \, mod n} \, mod \, m$ and send to Ola.

Ola computes $t = p^{r_1} i^{r_2} \, mod \, m$, and accept q if and only if t value is equivalent to the right-hand side of this equation.

3. Algorithm for Disavowal

Ola selected $r_1, r_2 \in \mathbb{Z}_n^*$ randomly, computes $q^{r_1} j^{r_2} \, mod \, m$ and send to Ade

Ade computes $t = s^{a^{-1} \, mod n} \, mod \, m$ send it to Ola

Ola verifies the signature $t \neq p^{r_1} i^{r_2} \, (mod \, m)$

Ola selects e1, e2 $\in \mathbb{Z}_n^*$ randomly and computes $S = q^{\mathrm{e1}} \, j^{\mathrm{e2}}$mod m and send to Ade

Ade computes $T = S^{a^{-1} \, mod n} \, mod \, m$ and send to Ola

Ola verifies Ade signature by examining $T \neq p^{\mathrm{e1}} i^{\mathrm{e2}} mod \, m$

Ola computes $(ti^{-r_2})^{\mathrm{e1}} \neq (Ti^{-\mathrm{e2}})^{r_1}$

Give five variables of a signature scheme

Answer

P: denotes a finite set of possible messages
A: a finite set of signatures
K: the keyspace is a finite set of possible keys
$Sig_k \in S$: Signing algorithm
$Ver_k \in V$: Verification algorithm

3.1.2.4 Undeniable Signature Example

Given a prime m = 367 with $m = 2n + 1, i = 2, a = 103, j = 229$. If a message p = 186 is signed with assumed signature q = 73. Show that the purported signature from Ade is invalid.

Solution

1. Ola randomly selects values $r_1 = 35$, $r_2 = 137$ and computes $s = q^{r_1} j^{r_2} \, mod \, m = \, 73^{35} 229^{137} \, mod \, 367 = 53$
2. Ade computes and responds with $t = s^{a^{-1} \, mod n} \, mod \, m = \, 53^{103^{-1} \, mod n} \, mod 36 = 36 \; i.e \; 103^{-1} \, mod \, 183 = 16$. Note: m = 2n + 1, therefore n = 183.
3. Ola computes $t \; and \; compare \; with \; p^{r_1} i^{r_2} \; (mod \, m) = \, 186^{35} 2^{137} mod \, 367 \, =$ 358.

Since 36 ≠ 358, Ola proceeds to next step

4. Ola selects two values e1= 115, e2 = 3 randomly and computes $S = q^{\mathrm{e}1} \, j^{\mathrm{e}2} = \; mod \, m = \, 73^{115} 229^3 \, mod \, 367 = 350$
5. Ola computes $T = \, S^{a^{-1} \, mod n} \, mod \, m = \, 350^{16} \, mod \, 367 = 274$
6. Ola computes $T \; and \; compare \; with \; p^{\mathrm{e}1} i^{\mathrm{e}2} mod \, m = \, 186^{115} 2^3 \, mod \, 367 = 115$. Since T = 115 $\neq 274$, Ola proceeds to next step
7. Ola computes $(ti^{-r_2})^{\mathrm{e}1} \; and \; compare \; with \; (Ti^{-\mathrm{e}2})^{r_1}$ i.e
8. $(ti^{-r_2})^{\mathrm{e}1} \quad = \left(\left(q^{r_1} j^{r_2}\right)^{a^{-1}} i^{-r_2}\right)\right)^{\mathrm{e}1}$ i.e. $t = s^{a^{-1} \, mod n} \, mod \, m$ and s = $q^{r_1} j^{r_2}$

Therefore, $(ti^{-r_2})^{\mathrm{e}1} = q^{r_1 \, \mathrm{e}1 a^{-1} \, mod n} j^{a^{-1} \, mod n \, \mathrm{e}1 r_2} \; i^{-r_2 \mathrm{e}1} = = q^{r_1 \, \mathrm{e}1 a^{-1} \, mod n} mod \, m$ since $i = j^{a^{-1} \, mod n} = 73^{35 \, x \, 115 \, x \, 103^{-1} \, mod \, 183} mod \, 367 = 238$

Also, $(Ti^{-\mathrm{e}2})^{r_1} = q^{r_1 \, \mathrm{e}1 a^{-1} \, mod n} mod m = 238$

9. Hence, since step 8 = step 9, then it is sufficient for Ola to say the message p purportedly signed by Ade is not valid and therefore, forged.

3.1.2.2 Fail-stop Signature Scheme

A fail-stop algorithm is a one-time algorithm that contains signing, verification, and proof of forgery protocol. It enables the signer to be able to prove his signature has been forged. For example, if Ayo was able to forge the signature of Ade, then with the proof of forgery protocol, Ade will be able to establish his signature by Ayo is indeed forged.

3.1.2.3 Fail-stop signature scheme Components

1. Signing algorithm
2. Verification protocol
3. Proof of forgery

3.1.2.4 Fail-stop signature Cryptosystem

Given prime numbers p and q such that $m = 2n + 1$ and the discrete logarithm problem in $\mathbb{Z}_m$ is intractable.

Also, given $i \, \epsilon \, \mathbb{Z}_m^*$ an order p element of n.

Assuming $1 \leq c_0 \leq n - 1 \; and \; j = \; i^{c_0 mod \, m}$, with $m, n, i, and \; j$ as

public and c_0 as private secret and kept from everyone even signer.

Given a key K of a form $(\delta 1, \delta 2, c1, c2, d1, d2)$

where $1, c2, d1, d2 \in \mathbb{Z}_n$. Again,

$\delta 1 = i^{c1} j^{c2} \ (mod\ m)$

$\delta 2 = i^{d1} j^{d2} \ (mod\ m)$

$\delta 1,\ \delta 1$ are public, where, $c1, c2, d1, d2$ are private keys

For a given key $K = (\delta 1, \delta 2, c1, c2, d1, d2)$ and message $\in Z_n$,

We define signature $Sig_K\ (p) = (q1, q2)$ where

$q1 = c1 + pd1\ mod\ n$ and

$q2 = c2 + pd2\ mod\ n$

For $q1, q2 \in \mathbb{Z}_n \times \mathbb{Z}_n$,

$Ver\ (p, q) = true$ if and only if $\delta 1 \delta 2^p = i^{q1} j^{q2} \ (mod\ m)$

3.1.2.5 Security of Fail-stop Signature (FsS)

To examine the security of this cryptosystem,

two key $K1\ and\ K2$ are defined, where

$K1 = (\delta 1, \delta 2, c1, c2, d1, d2)$ and

$K2 = \delta 1^{|}, \delta 2^{|}, c1^{|}, c2^{|}, d1^{|}, \delta 2^{|}$ then

$K1 = K2$ if $\delta 1 = \delta 1^{|}$ and $\delta 2 = \delta 2^{|}$

To establish the fact $K1 = K2$, we use

Lemma 1.1:

If $K1 = K2\ \ni\ if\ Ver_{K1}(p, q) = true$ then $Ver_{K2}(p, q) = true$

Proof

Given $K1 = (\delta 1, \delta 2, c1, c2, d1, d2)$ and

$$K2 = \ \delta 1, \delta 2, c1^{|}, c2^{|}, d1^{|}, \delta 2^{|}$$

Where $\delta 1 = \ i^{c1} j^{c2} \ (mod\ m) = \ i^{c1^{|}} j^{c2^{|}} \ (mod\ m)$

$\delta 2 = \ i^{d1} j^{d2} \ (mod\ m) = \ i^{d1^{|}} j^{d2^{|}} \ (mod\ m)$

If message p is signed with key K1, yielding signature $q = (q1, q2)$

The verification algorithm $Ver_{K1} = (p, q)$ only depends on $\delta 1, \delta 2, p, q$

Again, Lemma 1.2:

Assuming with key K and $q = Sig_K \ (p)$,
$then\ there\ exist\ q, with\ keys\ k1 = k1\ \ni$

$$q = Sig_{K2} \ (p)$$

Proof

Assuming $(\delta 1, \delta 2)$ is given as public, we wish to find, $c1, c2, d1, d2\ \ni$

$\delta 1 = \ i^{c1} j^{c2} \ (mod\ m)$

$\delta 2 = \ i^{d1} j^{d2} \ (mod\ m)$

$q1 = c1 + pd1\ mod\ n$ and

$q2 = c2 + pd2\ mod\ n$

But since $i \ \in G$ there exists elements $e1, e2, c0 \ \in \ \mathbb{Z}_n \ \ni$

$\delta 1 = \ i^{e1} (mod\ m)$

$\delta 2 = \ i^{e2} (mod\ m)$

$j = \ i^{c0} (mod\ m)$

Hence,

$e1 = c1 + c0c2 \ (mod\ m)$

$e2 = d1 + c0d2 \ (mod\ m)$

$q1 = c1 + pd1\ mod\ n$ and

$q2 = c2 + pd2\ mod\ n.$

3.1.2.6 Fail-stop Signature (FsS) Example

Given m = 2443, i = 2, c0 = 1237, given that Ade is presented with a forged signature (411, 33) over message 2383. Show that the signature is forged or otherwise.

Solution

Compute

Given m = 2443 = 2 x 1221 + 1

i = 2, c0 = 1237, $j = 2^{1237}(mod\ 2443) = 2004$

Note: Ade knows the values of i and j but not c0. If Ade select key with

C1 = 444, c2 = 512, d1 = 392, and d2 = 499 then he computes

$$\delta 1 = i^{c1}j^{c2}\ (mod\ m) = 2^{444}2004^{512}\ (mod\ 2443) = 2202$$

$$\delta 2 = i^{d1}j^{d2}\ (mod\ m) = 2^{39}2004^{499}\ (mod\ 2443) = 1100$$

Compute $\delta 1\delta 2^{p}\ and\ compare\ with\ i^{q1}j^{q2}\ (mod\ m)$

$$\delta 1\delta 2^{p} = i^{q1}j^{q2}\ (mod\ m)$$

$$\delta 1\delta 2^{p} = 2202\ x\ 1100^{2383} = 2118$$

$$i^{q1}j^{q2}\ (mod\ m) = 2^{411}2004^{33}\ (mod\ 2443) = 638$$

Therefore, the signature is forged since

$$\delta 1\delta 2^{p} \neq i^{q1}j^{q2}\ (mod\ m)$$

3.2 Security Requirement for Signature Schemes

The security requirements for a signature are those things necessary for the signature to be secure against forms of attacks. That is, what the signature needs to ensure its security from attacks. In this case, three models which are necessary for the signature's security are discussed.

What are the security requirement models necessary for signature schemes explained above?

3.2.1 Security Requirement Models

1. An attack model
2. The adversary goal
3. Scheme security

An Attack model

This model specifies an available information and resources an attacker possesses to ensure the breach of model's security. The category of attack model includes:

a. Key-only attack

This is an attack model where adversary or third party say Ayo having access to the public key of Ade, for example, if Ayo can access the Ver_K (verification model) of Ade.

b. Chosen message attack:
An attack model where adversary asks for and examines the user's signature on a set of his messages. For example, Ayo examines the list of Ade's signature $q_r = Sig_K(p_r), where\ r = 1, 2, \ldots$ and selects messages $p_r, where\ r = 1, 2, \ldots$

c. Known message attack
An attack model where an adversary say Ayo possesses previously signed Ade's set of messages. The messages, signatures pairs could be $(p_1, q_1),\ (p_2, q_2), \ldots$ In this case, signature $q_r = Sig_K(p_r), where\ r = 1, 2, \ldots$ with p_r selected messages and q_r signatures.

1. The adversary goal

The security of the signature scheme conditionally. This is because an adversary can examine available messages p and test for valid signature $q = Sig_K(p)$. Therefore, it is sufficient for the third party, for instance, Ayo, to forge any signature over some time. The goals are the motives or reasons behind the adversary planned attack. In all these goals, the message p has not been signed previously by the signer. These goals could be of the followings:

a. Total break:
This goal arises as a result of an adversary able to acquire the private key of the signer. For example, if Ayo can guess the private key of Ade correctly, and having possessed the signing function of Ade, then he can easily sign validly any message on his behalf.

b. Existential forgery:
In this goal, an adversary can establish a valid signature for one of the signer's messages p_r. That is, from a set of pair messages p_r, an adversary can establish signatures (p_r, q_r) by constructing $Ver_K\ (p, q) = (true)$.

c. Selective forgery:
With the aid of probability functions, an adversary, for instance, Ayo can compute a signature function on randomly selected messages and able to sign a valid signature of Ade. For example, Ayo can determine from randomly selected message p that signature verification function $Ver_K\ (p, q) = (true)$ using some probability functions.

Examples of Attacks

a. Using an existential forgery in multiplicative RSA, an adversary can establish an authentic signature $q_1, q_2\ mod\ n$ on messages set p_1, p_2, given previously sign signatures $q_1 = Sig_K(p_1)$ and $q_2 = Sig_K(p_2)$, then the $Ver_K\ (p_1, p_2 mod\ n,\ q_1, q_2\ mod\ n\ ) = (true)$.
b. In selective forgery using a selected message, Ayo can find all possible p say $p_1, p_2 \in \mathbb{Z}_n\ \ni p \in P_n mod n$. If Ade gives his signatures q_1, q_2 on the messages p_1, p_2, then $q_1, q_2\ mod\ n$ is the signatures on messages $p_1, p_2\ mod\ n$

eVoting Application Scenarios 7
This scenario depicts the electronic election of the Austrian Student Association of 2009. The Austrian Students Association (¨OH) is the general university students' representative body in Austria. The ¨OH provides students with political and academic representation, information, service, and advice. The ¨OH is a member of the European Students' Union (ESU). The statutes of the ¨OH are regulated in federal law and an ordinance, the "Hochsch¨ulerinnen und Hochsch¨ulerschaftsgesetz". The legal regulation explicitly allows for electronic voting. Every two years, all Austrian students are entitled to elect the representative bodies of the ¨OH. The most recent ¨OH election was held in May 2009.

All students of the 21 Austrian universities matriculated in summer term 2009 were eligible to vote. The students were enabled to cast their vote electronically via the Internet. To this end, they were allowed to vote from their home computers or alternatively from voting computers in official polling stations. Electronic votes could be cast from Monday to Friday. After the electronic election period, a second voting period based on classic paper-based voting took place. Around 2200 students cast their vote over the Internet. Since every voter could cast for several institutions, the number of votes cast was even higher. The deployed software system was the Pnyx.core voting system. The system was implemented at the Austrian

Federal Computing Centre. For identification and authentication, the students used their electronic Austrian Citizen Card (a smartcard) and respective card reader devices which were distributed at no charge. No further registration process was necessary; the electoral roll was generated using information from the university data network.

4.0 Self-Assessment Exercise(s)

1. The followings are common to undeniable signature except:

   A. Duplication prevention feature.
   B. Ola cannot verify the signature of Ade without Ade's knowledge and awareness.
   C. The signature signed by Ade can only be duplicated and distributed through his awareness.
   D. Ola can easily verify the signature of Ade without Ade's knowledge.

   The correct answer is D.

2. The adversary goals of forging signature include the following except:

   a. Total break
   b. Existential forgery
   c. Selective forgery
   d. Alternate forgery

   The correct answer is D.

3. The difference between undeniable and fail-stop signature scheme is

   A. An undeniable signature scheme is characterised with a feature of duplication prevention while the fail-stop signature scheme is a one-time algorithm.
   B. An undeniable signature scheme is a one-time algorithm while the feature of duplication prevention characterises a fail-stop signature scheme.
   C. The undeniable signature scheme is a two-time algorithm while a feature of duplication prevention characterises the fail-stop signature scheme.
   D. The undeniable signature scheme is characterised with the feature of duplication while the fail-stop signature scheme is a one-time algorithm.
   The correct answer is A.

5.0 Conclusion

In this unit, you have learnt about the concepts of a signature scheme. That is, how signer can sign his document or message to ensure the authenticity of the message and how the verifier can authenticate the signature on the message to be a valid signature. You have also learnt the security of a signature scheme and various goals associated with attacks. The attack model specifies the available information possess by an attacker to compromise the validity of the signature.

6.0 Summary

This unit examines the meaning of the signature scheme and its associated security. The signature scheme is a technique to attach a signature to a digitalised document to ensure its authenticity. The scheme consists of signing and verification algorithm. Two basic signature schemes were examined: undeniable signature and fail-stop signature. The undeniable signature is called undeniable because it consists of disavowal protocol which helps a signer to prove that a signature is indeed not originated from him. This signature has three basic algorithms: signing, verification and disavowal protocol. The fail-stop scheme, on the other hand, is characterised by proof of forgery protocol in addition to signing and verification algorithms. The proof of forgery protocol assists the signer to establish a signature is indeed forged by a third party.

7.0 References/Further Reading

Douglas, R. S. (2006). Cryptography Theory and Practice. (3rd ed.). Discrete Mathematics and Its Application. University of Waterloo Ontario, Canada: Chapman and Hall/CRC.

Karatsiolis, V., Langer, L., Schmidt, A., Tews, E. & Wiesmaier, A. (2010). Cryptographic Application Scenarios. Darmstadt Germany. Retrieved on 15th January 2020, from https://www-old.cdc.informatik.tu-darmstadt.de/reports.

Unit 2: Digital Signature Algorithm

Contents

1.0 Introduction

In this unit, you will learn the concepts of a digital signature. The digital signature has to do with the signing and verification of electronic documents. This unit also discusses authentication, authorisation, and identification as related to a digital signature. The unit finally examines the basic components of digital signature and its challenges, including the benefits over the traditional paper signature.

2.0 Intended Learning Outcomes (ILOs)

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

- describe the concepts of digital signature
- verify whether the digital signature is authentic or fake using digital signature algorithms
- distinguish authentication, authorisation, and identification

- manage the special type of signature scheme and its signature algorithm such as undeniable and fail-stop signature schemes.

3.0 Main Content

3.1 Digital Signature

A digital signature is a signature scheme of signing a document stored in electronic form as against the "Conventional" handwritten signature attached to a paper document used to authenticate a person responsible for the signature. A signed message over an insecure Internet can be transmitted over a computer and other communication network systems. In signing an electronic message, an algorithm that is used to sign the message must "bind" the signature to the message; the traditional signature on the other hand attached to the physical data, information or document. The problem of signing an online message is in two categories. The first problem is the problem of signing a document while the second problem is that of verification. A conventional signature could be easily verified by simply compared with the original or authentic signature. For example, a customer paid check could be verified by a cashier by comparing the signature on the check with the original one in the bank for verification. A digital signature, on the other hand, requires a publicly known verification algorithm. Thus a digital signature can be verified by anybody and therefore, requires a secure signature scheme to prevent the possibility of forgeries and repudiation by intruders, as shown in figure 4.1.

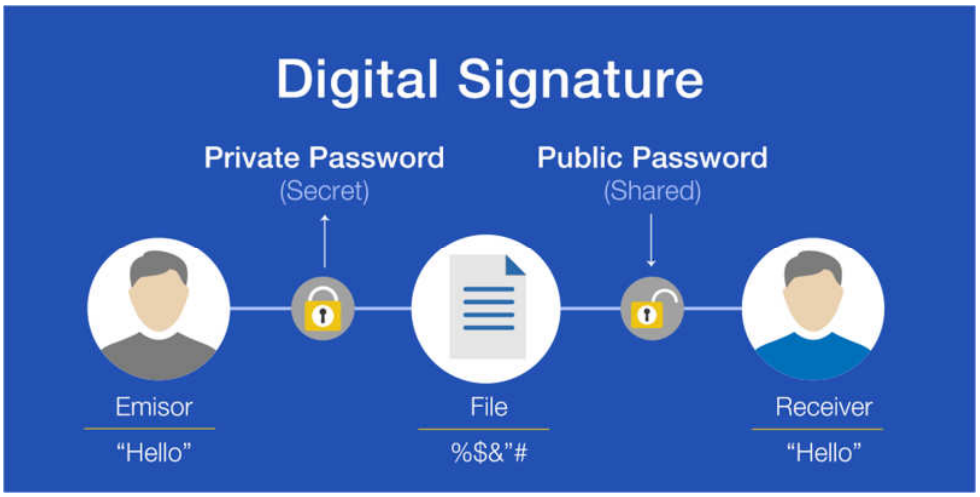

Fig. 4.1: Digital Signature

A digital signature is a cryptographic algorithm used to ensure the authentication, authorisation, and nonrepudiation of electronic messages. The purpose is to provide a digital means of authenticating legitimate user's identity for a given piece of information. The process of signing a document

involves transforming message and secret information into a digital model called a signature. In figure 4.1, The sender encrypts a message with the private key, then send to the receiver. The receiver gets the intended message by decrypting the ciphertext with the public key.

3.1.1 Challenges of Digital Signature

The major challenges associated with the digital signature is its feature of reused. A copy of a digital message is identical to the original and can be easily reused by anybody. For example, if Ade authorised Ola to withdraw a certain amount of money from his account. For prevention of multiple withdrawals of the amount by Ade, the digital message should contain certain information, such as the date, to prevent it from being reused.

3.1.2 Components of Digital Signature

The digital signature scheme has two important components:

a. Signing algorithm (Sig_K)
b. Verification algorithm (Ver_K)

Message x that is signed by Ade using a signature algorithm Sig_K, which depends on his private key can be verified by Ola using a publicly known verification algorithm Ver_K. Consider a pair (x, y) where x is a message and y indicate a signature on a message x, then the resulting verification algorithm Ver_K x = y is true if a message x has been validly signed and y = Ver_K; x is false if x is a forged signature or not previously signed.

3.1.3 Algorithm of DSA

The Digital Signature Algorithm (DSA)(1994) is a modification of the ElGamal Signature Scheme, which incorporates some characteristics of Schnorr Algorithm. The DSA was first published in the Federal Register on May 19, 1994, and was finally adopted as a standard on December 1, 1994. D**SA** modifies the ElGamal Scheme by signing a 160-bit message using a 320-bit signature. The computations are done using a 512-bit modulus *p*. However, due to the criticisms from various quarters over the fixing of prime p value as 512-bits, the NIST altered the description of the standard and various modulus sizes can be used. The first change made is by changing the "-" to a "+" in equation 3 below, so that δ becomes

$$\delta = (x + a\gamma)\, k^{-1} \bmod (p - 1) \qquad (3.1)$$

This changes the verification condition to the following:

$$\alpha^{x}\beta^{\gamma} \equiv \gamma^{\delta} \pmod{p} \qquad (3.2)$$

If $\gcd(x + a\gamma, p - 1) = 1$, then $\delta^{-1} \bmod (p - 1)$ exists, and the equation (3.2) becomes:

$$\alpha^{x\delta^{-1}}\beta\,\gamma^{\delta^{-1}} \equiv \gamma \pmod{p} \qquad (3.3).$$

Also, the message x in DSA needs to be hashed using SHA-1 before it is signed with a 320-bit signature over 160-bit message digest.

3.1.4 A Conventional Description Of Digital Signature

Variables Representation

X is the set of messages which can be signed.

S is a set of elements called *signatures*, possibly binary strings of a fixed length.

S_M represent a *signing transformation* for entity M, is a transformation from the set message X to the signature set S. S_M is secret and will be used to create signatures for messages X.

V_X represent a *verification transformation* for A's signatures. It is a transformation from the set X χ S to the set (*true*; *false). V_M* is used by the receiver to verify signatures from signer M.

3.1.5 Signing Procedure

The *signer* M creates a signature for a message x ϵ X using the following computation:

1. Compute s = S_M(x).
2. Transmit the pair (x; s) where s is the *signature* for message m.

3.1.6 Verification procedure

The verifier Y confirm a signature s on a message m created by signer M. The verifier verifying the signature S_M through the following steps:

1. Obtain the verification function V_M of M.
2. Compute n = V_M (x; s).
3. Accept the signature as signed by M if n = *true*, and reject the signature M if n = *false*.

What does p, q, α, β, and a stand for in DSA?

3.1.7 Example of DSA

Suppose Ade uses DSA with q = 101, p = 7879, α = 170, a = 75 and β = 4567. We wish to determine Ade's signature on a message x such that SHA-1(x) = 52, using a random value k = 49 and find out whether the signature is authentic or forged.

Note: The values p, q, α, and β are public keys while a is private key and $0 \leq a \leq q - 1$

Solution

Given that q = 101, p = 7879, α = 170, β = 4567, a = 75, SHA-1 (x) = 52, k = 43
The first step is to determine the signature of Ade on the message x by computing the following:

$\gamma = \alpha^k \pmod p \bmod q$
$= 170^{43} \bmod 7879 \bmod 101 = 85$
$\delta = (\text{SHA-1}(x) + a\gamma)\, k^{-1} \bmod q \bmod q$
$= (52 + 75 * 85)\, 43^{-1} \bmod 101 \bmod 101 = 79$

The signature (85, 79) of Ade can therefore be verified by computing the following:

$\delta^{-1} = k^{-1} \bmod q = 43^{-1} \bmod 101 = 47$
$e_1 = \text{SHA-1}(x)\, \delta^{-1} \bmod q$
$= 52 * 33 \bmod 101 = 100$
$e_2 = \gamma\delta^{-1} \bmod q$
$= 85 * 33 \bmod 101 = 2805 \bmod 101 = 78$ and
$\text{Ver}_k(x, (\gamma, \delta)) = \text{true} \Leftrightarrow \alpha^{e1}\beta^{e2} \bmod p \bmod q$
$\Rightarrow \quad 170^{100}4567^{78} \bmod 7879 \bmod 101 = 85$
Therefore the signature (85, 79) on message 100 should be accepted by Ade.

3.2 Authentication, Authorisation, and Identification

3.2.1 Authentication

Authentication is a security process that allows only legitimate users to access protected resources. These protected resources include computers, database, networks, website, and any other online application. A digital signature is one of the techniques of authenticating electronic documents. There are three techniques to ensure the authentication of electronic resources:

a) What you know, e.g. password, PIN (Personal Identification Number)
b) What you have, e.g. token, card etc
c) Who you are, e.g. biometric

3.2.2 Authorisation

The authorisation, on the other hand, is the validation of authenticated users. It is the process of given access to authentic users. In other words, a user is given access to a system if that user is permitted to access it or deny access if the user was not given permission.

3.2.3 Identification

Identification is authentication of an individual entity that assures one party, using the available evidence of both the identity of a second party and that the second was active at the time the evidence was created or acquired. For example, assuming Ade put a call to Ola on the mobile network. If Ade and Ola know each other, which is the identification, then, voice recognition will provide entity authentication.

Scenario

Message Integrity

Assuming Ade is communicating with Ola on an insecure network. Consider an attacker, Oye, who intercepts and tampers with the messages between Ade and Ola. In this scenario, the goal is to assist Ola to detect any change in Ade's transmitted message. If Ade is sending his message in plaintext, and eavesdropping attack is not considered as a threat, then the problems of confidentiality and integrity must be kept separate for them to apply to a broader set of applications. Message Authentication Codes (MAC) is a general solution to the problems of confidentiality and integrity by combining authenticity and encryption. MAC consists of two algorithms:

1. Ade SIGNING function S
- Inputs: key k, message m
- Outputs: tag t

he then sends m and t to Ola.

2. Ola runs a VERIFICATION function V
- Inputs: key k, message m, tag t
- Outputs: "yes" if Alice sent (m,t), "no" otherwise

4.0 Self-Assessment Exercise(s)

1. The difference between digital and conventional signature is:

A. Digital signature scheme signs a document stored in electronic form while the conventional signature is a handwritten signature attached to a paper document.
B. Conventional signature signs a document stored in electronic form while the digital signature scheme is handwritten signature attached to a paper document.
C. Digital signature and conventional signature scheme sign a document stored in electronic form.
D. Digital signature and conventional signature scheme sign handwritten signature attached to a paper document.

2. Which of the following statement(s) is/are not correct?
   A. Authorisation is validation
   B. Identification is an authentication
   C. Authentication is a security process
   D. Authorisation is authentication

   The correct answer is D.

Assignment

1. MAC provides a general solution to the problems of confidentiality and integrity through what technique?
2. Explain the operation of MAC?

5.0 Conclusion

In this unit, you have learnt how digital signature can be used to sign an electronic document and how the verifier can verify whether the signature is authentic or false using signing and verification algorithm. This unit also has x-rayed the associated challenge of digital signature and its benefits over the conventional technique - the DSA assists in solving the problems of authentication, repudiation, and forgeries. The major challenge of DSA is in the reuse of features where a copy of the digital message is identical to the original and can be easily reused by anybody. The unit also examined the relationship and separation between the three related security mechanisms authentication, authorisation, and identification.

6.0 Summary

A digital signature is a cryptographic algorithm used to ensure the authentication, authorisation, and nonrepudiation of electronic messages. The digital signature allows the signing of electronic documents to curb the menace of document forgery and repudiation. The major challenges associated with the digital signature is its feature of reused. A copy of a digital message is identical to the original and can be reused easily by anybody. Authentication is a security process that allows only legitimate users to access protected resources. Authorisation, on the other hand, is the validation of authenticated users. An authorisation is the process of given access to authentic users. Identification is authentication of an individual entity that assures one party, using the available evidence of both the identity of a second party and that the second was active at the time the evidence was created or acquired. In the following unit, you will learn how hashing can be used to harden the security of data at rest and over an insecure communication network.

7.0 References/Further Reading

Adebayo, O. S., Waziri, V. O., Ojeniyi, J. A. & Bashir. S. A. (2012). "Information Security on the Communication Network in Nigeria Based On Digital Signature." *International Journal of Computer Science and Information Security (IJCSIS), Vol. 10* No. 11, November 2012, ISSN 1947-5500. Retrieved from https://www.academia.edu/12313193/Journal_of_Computer_Science_and_Information_Security__.

Douglas, R. S. (2006). Cryptography Theory and Practice. Third Edition, Discrete Mathematics and Its Application. University of Waterloo Ontario, Canada: Chapman and Hall.

Ronald, L. R. (1996). *Handbook of Applied Cryptography. Massachusetts Institute of Technology*. Retrieved on 30-12-2019 from https://www.academia.edu/36799709/handbook_of_applied_cryptography.

Karatsiolis, V., Langer, L., Schmidt, A., Tews, E., & Wiesmaier, A. (2010). Cryptographic Application Scenarios. Darmstadt Germany. Retrieved on15th January 2020 from https://www-old.cdc.informatik.tu-darmstadt.de/reports/TR/TI-10-01.CryptographicApplication_Scenarios.pdf.

Unit 3: Secure Hash

Contents

1.0 Introduction

In this unit, you will learn how to apply simple strange function from a string of arbitrary length to a string of 160 bit. A cryptography hash function is an algorithm function that has the capability of providing security and data integrity assurance.

2.0 Intended Learning Outcomes (ILOs)

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

- define and describe the hash function
- analyse the security of hash functions
- apply simple strange functions from a string of almost arbitrary length to strings of 160 bit to harden the security and integrity of data.

3.0 Main Content

3.1 Secure Hash

Secure hash is a cryptography function or algorithm that enhance the security and integrity of applying data. In order to use a hash function, a short fingerprint of data is constructed with the hash algorithm to test the validity of such data. The fingerprint is valid only if the data remain the

same; if the data is tampered with, then the fingerprint is no more the same. Irrespective of the storage location of the data (secure or insecure), the fingerprint of the hash function can be reconstructed to determine the authenticity of data. Fig. 4.2 shows the basic hash function model.

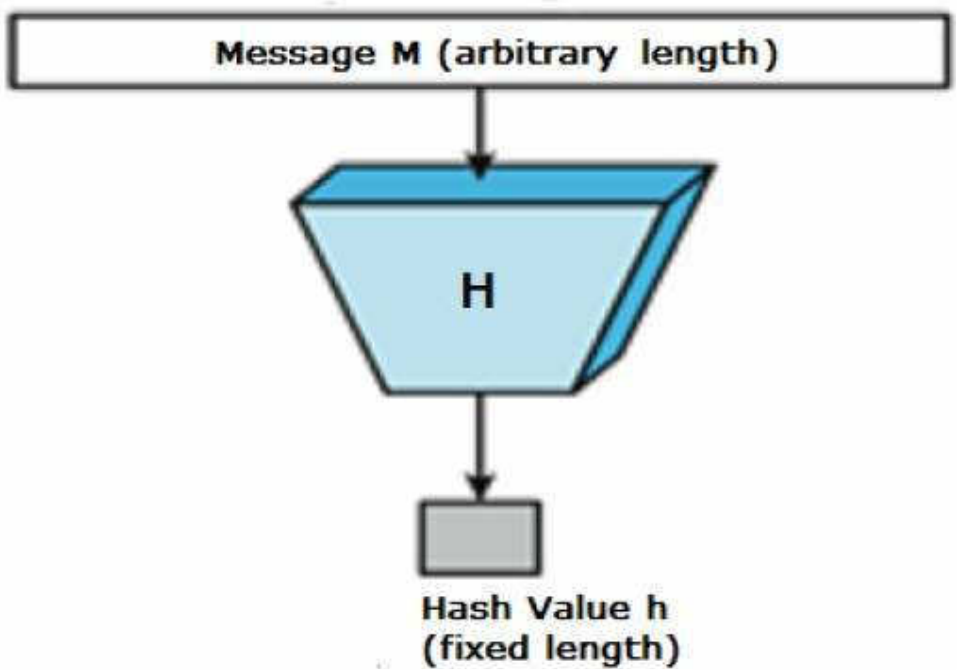

Fig. 4.2: Hash Function

From the definition of secure hash above, can you deduce the essence of hash function?

Examine three variables associated with the hash function.

3.1.1 Description of the Hash Function

Let h be a hash function and let represent some data as p. If p is a binary string of arbitrary length, then the corresponding fingerprint or message digest $q = h(p)$. Conventionally, y is usually a binary string of length 160 bits. Suppose y is stored in a secure location and p is not secure, then if p is altered to $p^{!}$ We can say $p = p^{!}$ .To detect the altered $(p^{!})$, we compute $q^{!} = h\,(p^{!})$ and validate $q^{!} \neq q$. Refer to Figure 1.1 to get in-depth knowledge about the hash function.

Definition 3.1
Suppose Ade and Ola share a secret key K to communicate. This secret key K is used to compute the hash function h_k. For any message p Ade or Ola can compute $q = h_k(p)$ to verify the authenticity of the message pair (p, q). If $q = h_k(p)$, then Ola will be sure the message has not been tampered with.

Definition 3.2
A hash function is a 4-tuple element as given below:

1. ρ represents a set of possible message
2. q represents a finite set of a message digest
3. K represents a finite possible set of keys
4. $p^!$ represents altered message

For every $k \epsilon K$, there exists a hash function $k \epsilon H$, such that for each h_k, $\rho \rightarrow q$

3.2 Security of Hash Functions

Given h: $\rho \rightarrow q$ as a hash function without key k. If $p \epsilon \rho$ and $q = h(p)$ The basic condition for the security of any cryptographic hash function to produce valid variables (p, q) is to first choose variable p and compute $p\, h(q)$. Besides, for any hash function to be secure, it must be associated with three problems which are difficult to be solved.

3.2.1 Hash Category Problems

Problem one (Pr1)
A hash function $h: P \rightarrow Q$ with variable $q \epsilon$ Q
We want to find $p \epsilon \rho \ni h(p) = q$.
The problem one is trying to find out if message p can be found $\ni$ $h(p) = q$ with given possible message-digest q. Provided problem Pr1 is solvable for a given message digest $q \epsilon$ Q then the pairs (p, q) is valid, that is the message p is not altered.

Problem two (Pr2)
A hash function $h: P \rightarrow Q$ with message $p \epsilon P$
We want to find $p^! \epsilon \rho \ni p^! \neq p$ and $h(p^!) = h(p)$.
With a message p problem Pr2 find out if $p^! \neq p$ is solvable $\ni$ $h(p^!) = h(p)$.
If this is possible then $(p^!, h(p))$ is valid.

Problem three (Pr3)
A hash function $h: P \rightarrow Q$ with variable $q \epsilon$ Q

We want to find p, $p^! \epsilon \rho \ni p^! \neq p$ and $h(p^!) = h(p)$.
A solution to this problem produces two pairs (p, q) and $(p^!, q)$.

$q = h(p) = h(p^!)$

4.0 Self-Assessment Exercise(s)

1. A cryptosystem is secure using a secure hash function if
   A. A hash function produces valid variables (p, q), chooses variable p and compute $p\, h\, (q)$.
   B. It is associated with three problems which are difficult to be solved.
   C. A hash function produces valid variables (p, q) chooses variable p and compute q $h\, (p)$.
   D. It is associated with three problems which are simple to be solved.

   The correct answer is A.

2. Hash category problems consist
   A. One problem
   B. Two problems
   C. Three problems
   D. None of the above

   The correct answer is C.

3. Which of the following statement (s) is/are NOT correct
   A. Secure hash is a cryptography function.
   B. Secure hash enhances the security and integrity of applying data.
   C. A short fingerprint of data is constructed with the hash algorithm to test the validity of data.
   D. The fingerprint is valid only if the data is tampered.

   The correct answer is D.

5.0 Conclusion

In this unit, you have learnt how you can ensure the security of communicating a message using a secure hash function. You have also learnt the three categorical problems which must be difficult to solve or insolvable before a secure hash could be considered secure. The concepts of this topic apply to various signature algorithms for ensuring the integrity and authentication of a user.

6.0 Summary

This unit discussed the concepts of secure hash. The security of any cryptographic function depends on the difficulty level of associated secure hash problems. A hash function is to enhance the integrity of data by constructing a fingerprint of such data for identification and validation. A hash function as 4-tuple elements consist ρ which represents a set of the possible message, $p^!$ which represents an altered message, q which represents a finite set of the message digest, and K which represents a finite possible set of keys. There are three categories of secure hash problems viz: problem 1, 2, and 3.

7.0 References/Further Reading

Douglas, R. S. (2006). "Discrete Mathematics and Its Application." In: H. E. Kenneth (Ed.). *Cryptography Theory and Practice.* (3rd ed.). University of Waterloo Ontario, Canada. Chapman & Hall/CRC, Retrieved on 30-12-2019 from https://labs.xjtudlc.com/labs/wldmt1/ reading%20list/books/Network%20security%20and%20crypto/Cryptography%20Theory%20and%20Practice.pdf

https://www.cs.umd.edu/class/spring2015/cmsc414/writeups/symmetric-key.pdf

Unit 4: Steganography

Contents

1.0 Introduction

In this unit, you will learn the meaning of steganography as an alternative means of securing data and information using the hiding technique instead of encryption. You will also learn the features, tools, and techniques to conceal data or file in an image, audio, and video for the prevention of such data. Steganography is an alternative technique of preventing data from attack by hiding such data in an image or other form. The image, audio, or video will be visible while the data within will not be seen. Several tools that can be used to perform steganography ranges from steghide to camouflage. The basic two methods of implementing steganography are Least Significant Bit (LSB) and Discrete cosine transform coefficient technique.

2.0 Intended Learning Outcomes (ILOs)

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

- define steganography
- identify tools for carrying out steganography
- differentiate between steganography vs cryptography
- demonstrate how to conceal a file, image, message and video within another file, message, image and video.

3.0 Main Content

3.1 Steganography

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. The word *steganography* comes from New Latin *steganographia*, which combines the Greek words *steganós* meaning "covered or concealed", and *-graphia* (γραφή) meaning "writing".

The security technique which conceals message either in an image or other medium to makes it look non-existence is referred to as steganography. Figure 4.3 depicts the modes of steganography.

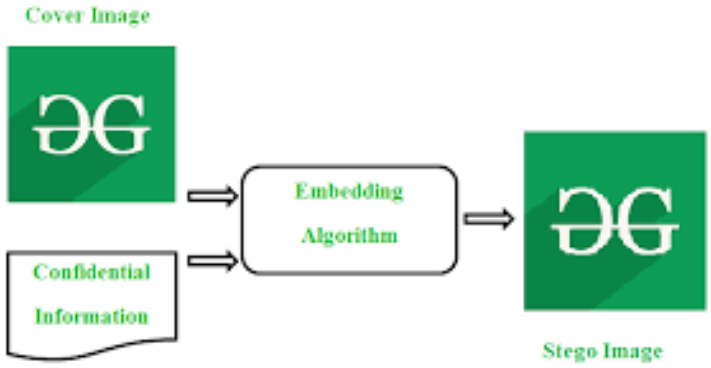

Fig. 4.3: Instance of Steganography

3.1.1 Forms of Steganography

Simple steganography:
This is time-consuming steganography, where the organisation of words clearly define the original message within an insipid text. Figure 4.4 is an example of simple steganography where the representation of initial letters of each word in the entire message clearly defines the hidden message

.

22nd August

Dear Ade

Compliment of the season to all at NOUN. Thanks for your last semester enquiry. Payment will be made soon. Treasury Single Account has made payment easier of recent here. Hopefully you would not relent in your hard work.

Yours truly.

Fig. 4.4: Simple Steganography

Invisible ink
This is the use of different materials for written which does not appear until another material is applied.

Character marking
This is an overwritten of marked or identified text using written material which can only be seen if the content holds close or at adjacent to bright light.

Pin punctures
Small pin punctures on marked texts are ordinarily not visible unless the paper is held up in front of a light.

Typewriter correction ribbon
This is the use of black ribbon between lines type, in which corrected results can only be visible under a strong light.

There are numerous numbers of tools used to carry out steganography. Identify three of those tools.

3.2 Tools for steganography

1. Steghide
   This is an open-source steganography tool that allows hiding of data in an audio file. It runs on a command-line prompt. Provided in figure 4.5 is the Graphical User Interface of steghide.

Fig. 4.5: Steghide

2. OpenPuff
   A steganography tool with the capability to conceal file in images, audios, videos, or flash files; and with various features to protect hidden data from discovery.

3. SSuite Picsel
   This tool allows user to provide carrier image and a key image. The tool uses the key image secretly to extract the hidden text file from the carrier image.

4. Xiao Steganography
   A hybrid tool that conceals file within the image (bmp), and audio (wav) files.

5. Camouflage
   This steganography tool allows the hidden of one file into another and has encryption functionality.

6. Netcross
   This tool uses strict firewall rules to establish covert IP channels across network perimeters. The tool carries data back and forth across the firewall using DNS resolution requests and responses.

3.2 Steganography Implementation

Steganography can be implemented in several ways, among which are:

1. Least Significant Bit (LSB)
   This is a simple technique that changes the last few bits in a byte to encode a message. The steganography technique is relevant when concealing image, where the colours' values of each pixel are represented by eight bits ranging from 0 to 255 in decimal or 00000000 to 11111111 in binary. For example, the change in the last two bits of the black pixel from 00000000 to 00000001 only changes the black value from 255 to 251. This change in colour to an open eye was unnoticeable but allowed encoding of data within the picture. Figure 4.6 shows an example of LSB.

2. Discrete cosine transform coefficient technique
   This approach slightly changes the weights (coefficients) of the cosine waves that are used to reconstruct a JPEG image. This technique might be useful for hidden ASCII text data where a single bit exclusion will completely change the character.

3.3 Basic steps in Steganography

The following steps are necessary to conceal data in a file using Steghide:

1. Install Steghide in Linux operating system using the following command: "apt-get install steghide". After the installation, type "steghide embed -ef *secretFile* -cf *coverFile* -sf *outputFile* -z *compressionLevel*-e *scheme" to embed data into a file.* Use the following argument in the Linux operating system to perform different hiding operations:

a. **-ef:** signifies the path of the file that to be hidden.
b. **-cf** signifies the file the data is embedded. These files are limited to JPEG, BMP, WAV, and AU files.
c. **-sf** signifies output file and its optional argument. The new stenographic file overwrites the cover file if this command is not used.
d. **-z** denotes the compression level between 1 and 9. In order not to compress the stenographic file, then argument **-Z** should be used.
e. **-e** denotes the encryption type. Steghide supports 128-bit AES encryption. Alternatively, use **-e none** to avoid file being automatically encrypted.

2. Use the command "**steghide embed -ef secret.txt -cf StegoDog.jpg -e none –Z**" in a Linux prompt menu to execute Dog image which was being used in figure 3.1 to hide the message

3. Set a password to extract the embedded data
4. Use the command "$ steghide extract -sf stegoFile -xf outputFile" to extract hidden data from stego image.

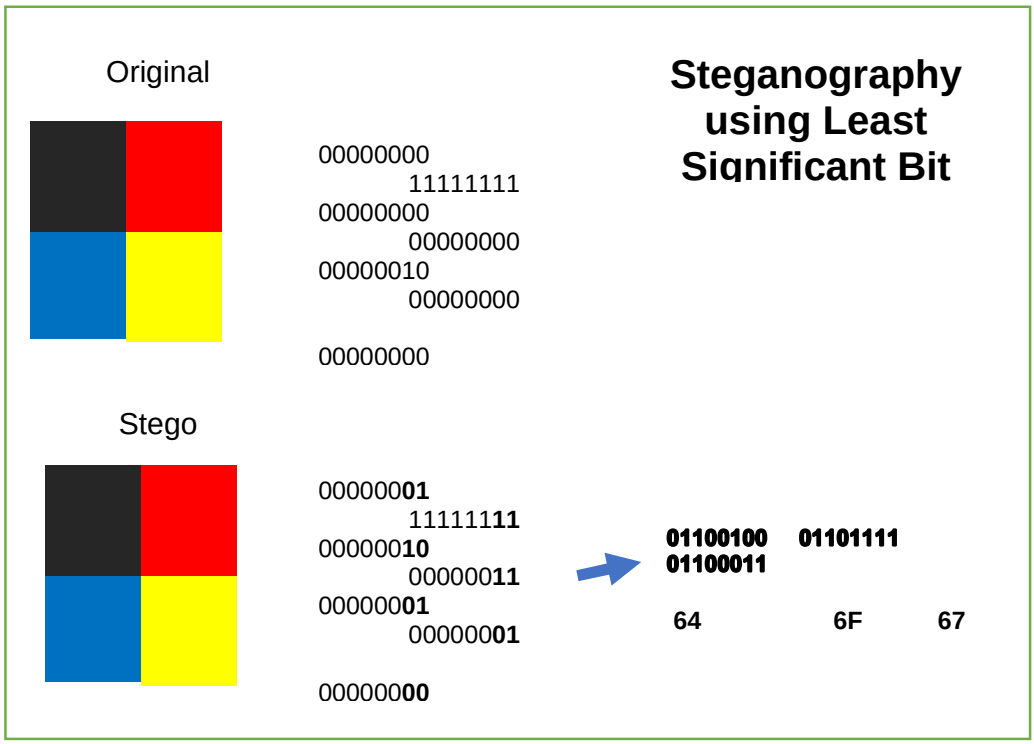

Fig. 4.6. Steganography of 4-pixel images in both colour and binary values

Each binary block represents the corresponding pixel's value

How does steganography differ from cryptography?

3.4 Steganography vs Cryptography

The basic difference is that cryptography technique through text transformation renders the message unintelligible to the users, while steganography hides the message inside an image or in other media to makes it appears non-existence.

3.5.1 Merits of Steganography

1. Steganography is at a higher level of cryptography due to the fact the message doesn't appear, therefore any need of thinking deciphering. This is to say that since the message does not become visible to the world, the attacker does not even know the message is there, unlike cryptography where the message appears unintelligible.
2. In addition, it conceals the identity of communicating parties, i.e. sender and receiver. It is most useful while sending message raise suspicious, for example, in a country where there is no freedom of expression.
3. It can be used as a digital watermark to detect when file or image is stolen.

3.5.2 Demerits of Steganography

The demerits of steganography include:

1. High overhead
2. The danger of steganography compromised may be grave since the message was not encrypted. This problem can be solved by combining cryptography and steganography technique.

4.0 Self-Assessment Exercise(s)

1. Which of the following statement (s) about steganography and cryptography is/are NOT correct?

   A. Both involve the encryption of the message.
   B. Cryptography encrypts the message and send, while steganography encrypts and hide the message.
   C. Steganography only hides messages.
   D. None of the above.

   The correct answer is A.

2. The implementation of steganography includes the followings except

   A. Least Significant Bit (LSB)
   B. Discrete cosine transforms the coefficient technique
   C. Most Significant Bit (LSB)
   D. None of the above

   The correct answer is D.

5.0 Conclusion

In this unit, you have learnt the concepts of steganography and basically how to conceal image in a file, image, and video. You have also learnt various forms, merit and demerit, and tools of steganography. This unit has acquainted you of various skills and techniques of hidden data or file in an image, audio, or video to ensure their security from attacks. The knowledge of steganography will in no doubt assist you in protecting valuable resources of the organisation and private data. This unit discussed the concept of steganography. It also examined the merits and demerits of steganography, In addition, tools of steganography, and the difference between steganography and cryptography were discussed. You will quite agree that the knowledge of steganography will help you to prevent data from been attacked.

6.0 Summary

Steganography is a security technique which conceals message in images or other media to makes it look non-existence. The three basic forms of steganography are Simple steganography, invisible link, and character watermarking. Several tools are available for the implementation of steganography which includes Steghide, OpenPuff, Camouflage, Netcross etc.

7.0 References/Further Reading

http://www.uoitc.edu.iq/images/documents/informaticsinstitute/Competitive_exam/Cryptography_and_Network_Security.pdf

CommonLounge. Retrieved on 22nd August 2019 from https://www.commonlounge.com/discussion/4bc16dbc2c7145ff87ad0f0d5401a242.

"How to Hide Secret Data Inside an Image or Audio File in Seconds."Retrieved on 22nd August 2019 from https://null-byte.wonderhowto.com/how-to/steganography-hide-secret-data-inside-image-audio-file-seconds-0180936.

William S. (2014). *Cryptography and Network Security: Principles and Practice.* (6th ed.). Pearson Education, Inc., ISBN 10: 0-13-335469-5: ISBN 13: 978-0-13-335469-0.