CRYPTOLOGY

Godwin S. Monserate

MSIT, CCNA/CCAI

Objectives

- Explain the difference between cryptology, cryptography and cryptanalysis
- Understand the basic principles of cryptography
- Understand the operating principles of the most popular tools in the area of cryptography
- List and explain the major protocols used for secure communications

History and Timeline of Cryptography

- 1,900 BCE - Monumental Hieroglyphs of the Old Kingdom of Egypt
- 1,500 BCE - Mesopotamian Secrets of Pottery
- 800 - Al-Kindi, "The Philosopher of the Arabs“
- 1467 - Leon Battista Alberti, "The Father of Western Cryptography“
- 1586 - The Babington Plot
- 1853 - Vigenère’s autokey cipher and the weaker Vigenère cipher
- 1917 - The Vernam Cipher
- 1923 - The Enigma Rotor Machine
- 1940 - Edgar Allan Poe Cracks the Code!
- 1942 - WW2 Japanese Navy Cryptography
- 1943 - The Colossus Computer
- 1953 - The VIC Cipher
- 1975 - The Data Encryption Standard
- 1976 - Diffie-Hellman key exchange
- 1991 - Phil Zimmermann, PGP (Pretty Good Privacy)
- 2001 - Advanced Encryption Standard
- November 2, 2007 -- NIST hash function competition announced.
- 2009 - Bitcoin network was launched.
- 2010 - The master key for High-bandwidth Digital Content Protection (HDCP) and the private signing key for the Sony PlayStation 3 game console are recovered and published using separate cryptoanalytic attacks. PGP Corp. is acquired by Symantec.
- 2012 - NIST selects the Keccak algorithm as the winner of its SHA-3 hash function competition.

History and Timeline of Cryptography

- 2013 - Edward Snowden discloses a vast trove of classified documents from NSA. See Global surveillance disclosures (2013–present)
- 2013 - Dual_EC_DRBG is discovered to have a NSA backdoor.
- 2013 - NSA publishes Simon and Speck lightweight block ciphers.
- 2014 - The Password Hashing Competition accepts 24 entries.
- 2015 - Year by which NIST suggests that 80-bit keys be phased out.

- References
- http://www.cypher.com.au/crypto_history.htm
- https://www.timetoast.com/timelines/the-history-of-cryptography
- https://en.wikipedia.org/wiki/Timeline_of_cryptography#2000_and_beyond

Terminology

- Cryptography: process of making and using codes to secure transmission of information
- Encryption: converting original message into a form unreadable by unauthorized individuals
- Cryptanalysis: process of obtaining original message from encrypted message without knowing algorithms
- Cryptology: science of encryption; combines cryptography and cryptanalysis

Abash/Atbash Cipher

- Abash Cipher is one of the easiest methods for cryptography and crypto-analysis.
- It was first used for the Jewish language but it can be used for the other languages.
- The way of cryptography is to make the last letter of the language to the first letter.
- The method of cryptography in English:

  Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ

  Cipher: ZYXWVUTSRQPONMLKJIHGFEDCBA
- Example: Plain Text: money

  Cipher Text: nlmvb

Answer this Cipher… using Transposition Cipher Encryption

u	y	h	f		t	i	i	o	n
a	t	t		t		h	k	h	e
e	b		u	c	t	l	s	j	n
a	r	o	o	t	i	n	f	m	
r	s	c	s	a	n	e	a	u	
s	d	u	n	e	c	r	a		i
s		i	y		s	m	t	i	p
h			e	e	n	t	l	t	r
y	c	o		p	t	g	y	r	r
	h		p				a	y	

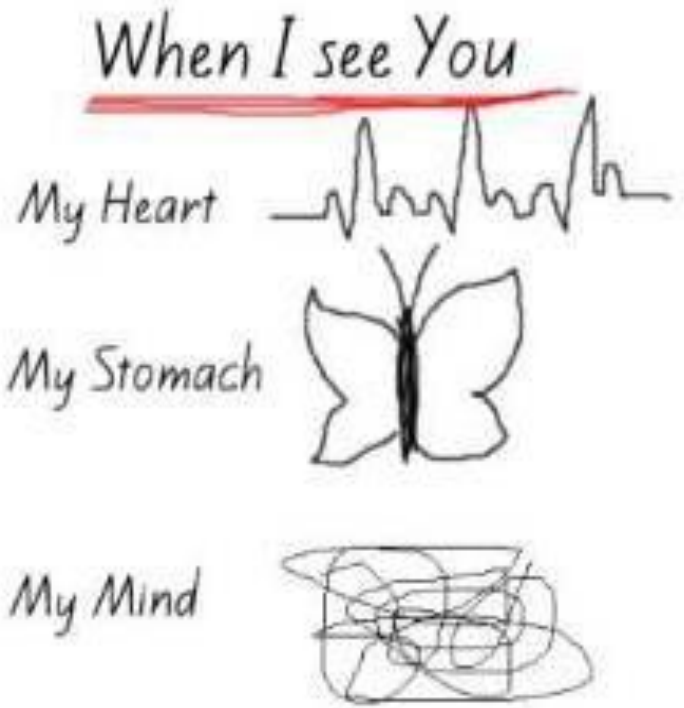
Cryptology Model

BOB

ALICE

Cryptology Model

BOB

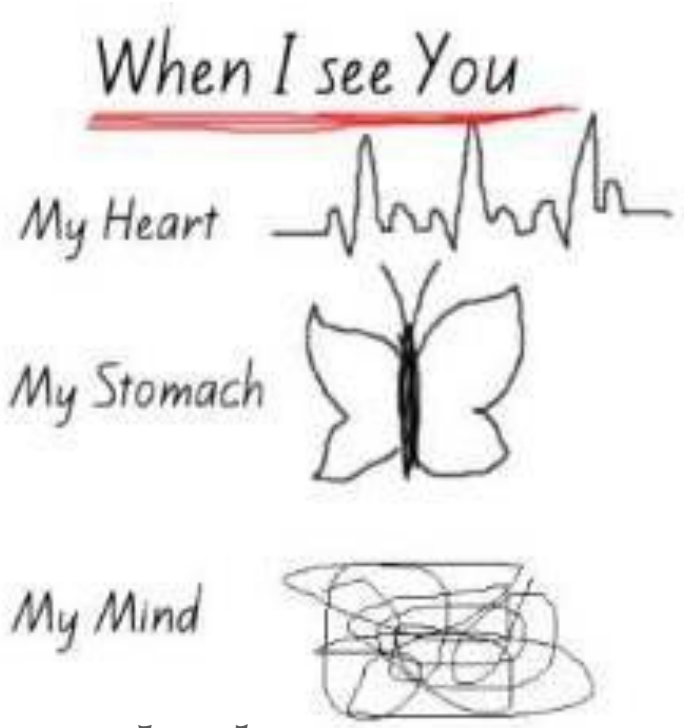


I think I am in love with you

ALICE

Cryptology Model

BOB



I think I am in love with you

ALICE

Cryptology Model

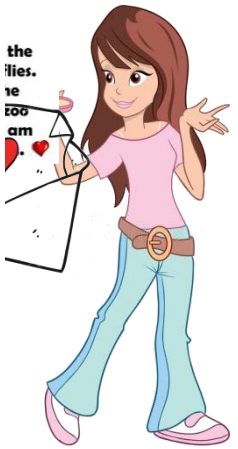
BOB

ALICE

Cryptology Model

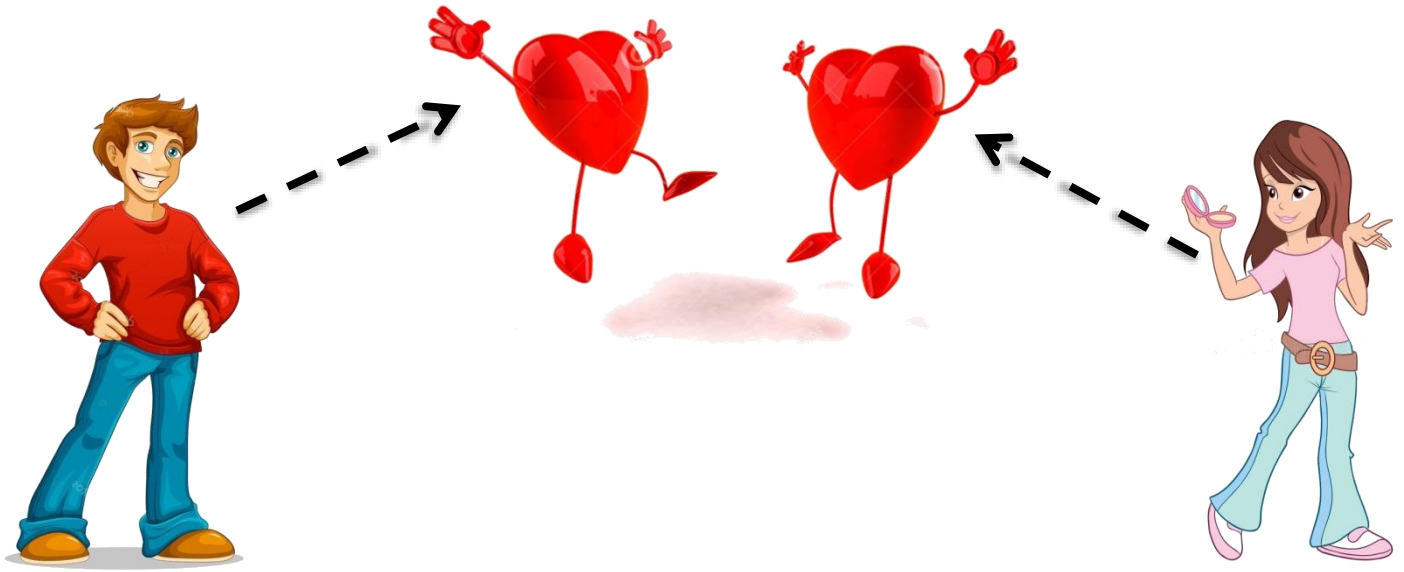
BOB

Forget the butterflies. I feel the whole zoo when I am with you.



ALICE

Cryptology Model



Cryptology Model - Adversary

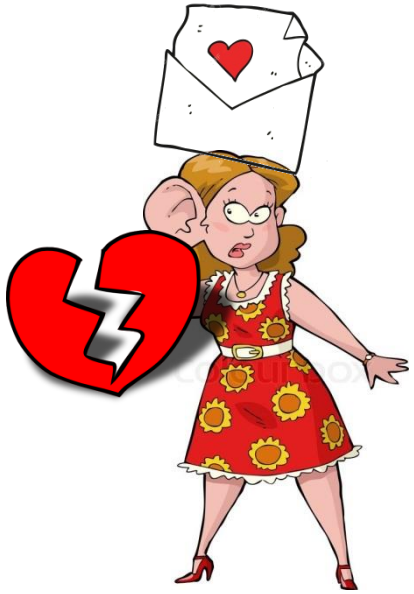
BOB

ALICE

EVE (Eavesdropper)

Cryptology Model

BOB



EVE (Eavesdropper)

ALICE

Cryptology Model

BOB

EVE (Eavesdropper)

ALICE

Cryptology Model

BOB

ALICE

EVE (Eavesdropper)

Cryptology Model – Applying Encryption

ENCRYPTED MESSAGE

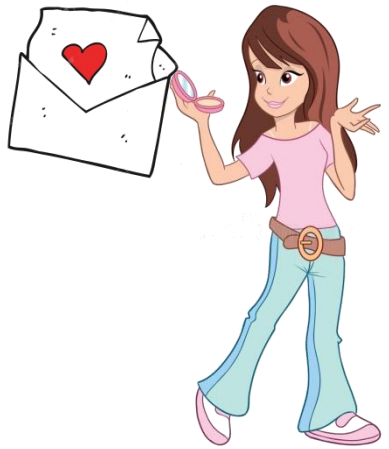
BOB

ALICE

Cryptology Model – Applying Encryption

ENCRYPTED MESSAGE

BOB



ALICE

Cryptology Model – Applying Encryption

ENCRYPTED MESSAGE

BOB

ALICE

EVE (Eavesdropper)

Cryptology Model – Applying Encryption

BOB

ENCRYPTED MESSAGE

EVE (Eavesdropper)

ALICE

Cryptology Model – Applying Encryption

BOB

ENCRYPTED MESSAGE

EVE (Eavesdropper)

ALICE

Cryptology Model – Applying Encryption

DECRYPT THE MESSAGE

BOB

ALICE

EVE (Eavesdropper)

Cryptology Model – Applying Encryption

ENCRYPTED MESSAGE

BOB

ALICE

EVE (Eavesdropper)

Cryptology Model – Applying Encryption

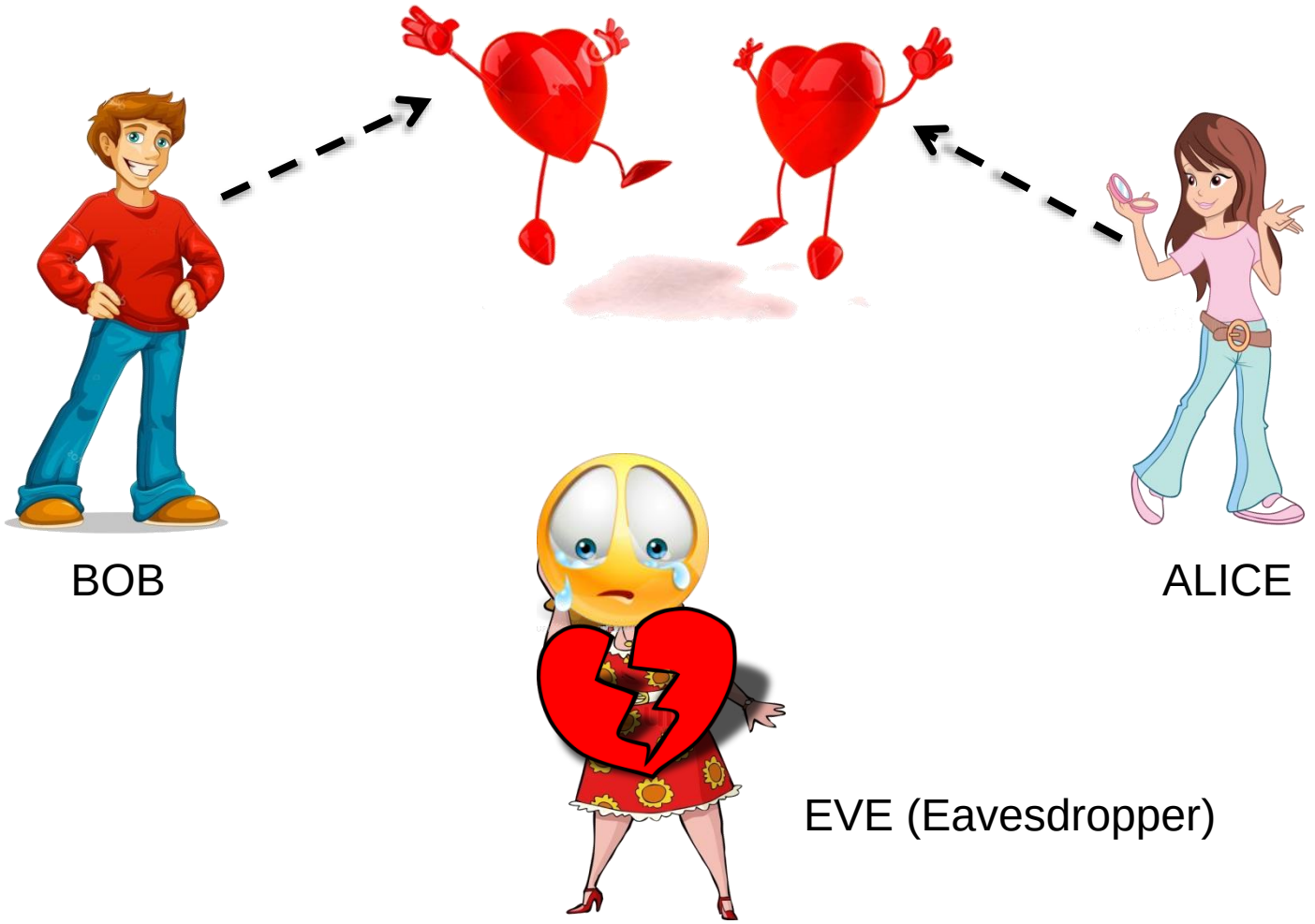
DECRYPT
MESSAGE

BOB

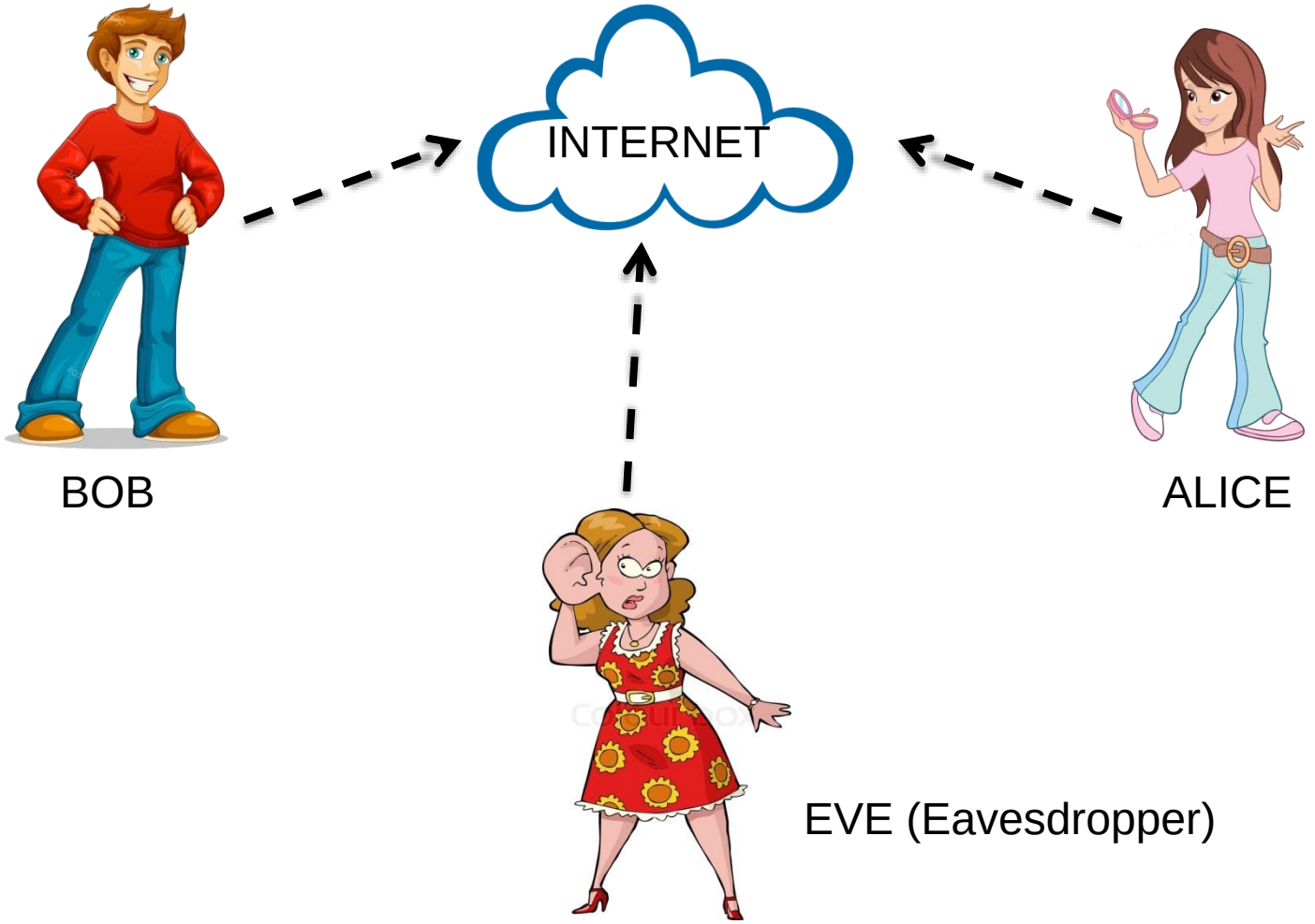
ALICE

EVE (Eavesdropper)

Cryptology Model – Applying Encryption



Cryptology in the Internet



Is there any need for Encryption?

ALICE

Hackers

BOB

Cyberterrorist

Eavesdroppers

Principles of Cryptography

- With emergence of technology, need for encryption in information technology environment greatly increased
- All popular Web browsers use built-in encryption features for secure e-commerce applications

Terminologies

- **Plaintext** – message or information that can be directly read by humans or a machine, ordinary readable text before being encrypted.
- **Ciphertext** is also known as encrypted or encoded information
- **Key** is a variable value that is applied using an algorithm to a string or block of unencrypted text to produce encrypted text, or to decrypt encrypted text.
- **Encryption** is the conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties.
- **Decryption** is the process of taking encoded or encrypted text or other data and converting it back into text that you or the computer can read and understand.

Elements of Cryptosystems

- Cryptosystems typically made up of algorithms, data handling techniques, and procedures
- Substitution cipher: substitute one value for another

m: Plaintext
c: Ciphertext

k_e: Encryptionn Key
E: Encryption Program

k_d: Decryption Key
D: Decryption Program

Cryptosystem

m: Plaintext
c: Ciphertext

k_e: Encryption Key
E: Encryption Program

k_d: Decryption Key
D: Decryption Program

Cryptography And Encryption-Based Solutions

- The notation used to describe the encryption process differs depending on the source.
- The first uses the letters m to represent the original message, C to represent the ending ciphertext, and E to represent the encryption process: E(M) = C.
- This formula represents the application of encryption to a message to create ciphertext. D represents the decryption or deciphering process, thus D[E(m)]=m.
- K is used to represent the key, thus E(m, K) = C, or encrypting the message with the key results in the ciphertext.

Different Types of Ciphers

- **Monoalphabetic Ciphers**
- A monoalphabetic cipher mixes up the characters of the alphabet and uses that same arrangement for the entire message.
- The simple case would be to advance each letter some number of spaces, for example moving 10 letters down the alphabet, **A → L**
- There are only 25 possibilities to check, so this type of cipher is trivial to solve

Caesar Cipher

- The **Caesar Cipher**, also known as a *shift cipher,* is a monoalphabetic cipher and one of the oldest and simplest forms of encrypting a message.
- Probably successful for Caesar because his foes were not able to read Roman and were mostly illiterate.
- It is a type of substitution cipher where each letter in the original message (which in cryptography is called the plaintext) is replaced with a letter corresponding to a certain number of letters shifted up or down in the alphabet.

Monoalphabetic Ciphers Tool

How to use Caesar Cipher Encryption

- For example, here's the Caesar Cipher encryption of a full message, using a right shift of 3.

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
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

Plaintext:
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

Ciphertext:
QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

Caesar Cipher Mathematical form

- The Caesar Cipher can be expressed in a more mathematical form as follows:

$$E_n(x) = (x + n) \ mod \ 26$$

- Decryption of the encrypted text (called the **ciphertext**) would be carried out similarly, subtracting the shift amount.

$$D_n(x) = (x - n) \ mod \ 26$$

Different Types of Ciphers

- **Polyalphabetic Ciphers**
- keep the same cipher alphabet throughout a message
- thwart the efforts of a cryptanalyst from using letter frequency analysis to break the code.
- each letter is now substituted by several different letters within the message.
- It was invented in 1467 along with a cipher disk to make it more user friendly.
- This disk is called a Vigenère cipher disk, even though it was invented by Leon Battista Alberti 56 years before Blaise de Vigenère was born.
- It is made up of two disks with the normal alphabet around the circumference of one and a mixed up alphabet around the smaller disk.

Polyalphabetic Ciphers Tool

Vigenère cipher disk

Jefferson Wheel Cypher

Vigenere Encryption and 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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

Encrypting message

1. Plaintext: **I LOVE YOU ALICE**
2. Create a keyword is **CEBUCITY**. Then, the keyword must be repeated
3. Remove all spaces and punctuation, and dividing the result into 5-letter blocks. As a result, the above plaintext and keyword become the following:

ILOVE YOUAL ICE
CEBUC ITYCE BUC

How to use Vigenere Cipher Encryption

1. To encrypt, pick a letter in the plaintext and its corresponding letter in the keyword
2. use the keyword letter and the plaintext letter as the row index and column index, respectively, and the entry at the row-column intersection is the letter in the ciphertext.
3. For example, plaintext is I, keyword letter is C. Row of I and column of C, intersect at K which is the encrypted result.

Col	i	l	o	v	e	y	o	u	a	l	i	c	e
Row	c	e	b	u	c	i	t	y	c	e	b	u	c
encryption	K	P	P	Q	G	H	G	T	C	P	J	E	M

Vigenere Encryption and 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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

Vigenere Cipher Encryption Output

Message: I love you alice

Cipertext: k ppsg ghs cpjwg

Different Types of Ciphers

- **Transpositions and Grills**
- The beginnings of cryptography can be dated to the use of transpositions of hieroglyphs in the tomb of Khnumhotep II in 1900 BC.
- The transpositions weren't necessarily made to keep the text secret, but to add dignity to the words and make it a form of riddle to keep the text interesting.
- A transposition is simply moving the letters around in a prescribed fashion so the resulting cipher text is mixed up and unintelligible.
- A grill is usually a thin paper or metal sheet with a grid where some of the letter positions or syllables or words are cut out. The grill is a type of cipher machine used to make the transposition cipher easier to use.

Transposition Ciphers Tool

How to use Transposition Cipher Encryption

1. Create a plaintext example:

   Meet at three pm today at the usual location

2. Arrange the plaintext according to 6 columns character and create a keyword eg. ZEBRAS

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

How to use Transposition Cipher Encryption

1. Arrange the plaintext according to 6 columns character using the keyword ZEBRAS
2. Look for Alphabetic sequence of the word ZEBRA e.i. A, B, E, R, S, Z

aeaelo

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

How to use Transposition Cipher Encryption

1. Arrange the plaintext according to 6 columns character and create a keyword eg. ZEBRAS
2. Look for Alphabetic sequence of the word ZEBRA e.i. A, B, E, R, S Z

aeaelo erotat

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

How to use Transposition Cipher Encryption

1. Arrange the plaintext according to 6 columns character and create a keyword eg. ZEBRAS
2. Look for Alphabetic sequence of the word ZEBRA e.i. A, B, E, R, S Z

aeaelo erotat

ehttua

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

How to use Transposition Cipher Encryption

1. Arrange the plaintext according to 6 columns character and create a keyword eg. ZEBRAS
2. Look for Alphabetic sequence of the word ZEBRA e.i. A, B, E, R, S Z

aeaelo erotat
ehttua tedhli

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

How to use Transposition Cipher Encryption

1. Arrange the plaintext according to 6 columns character and create a keyword eg. ZEBRAS
2. Look for Alphabetic sequence of the word ZEBRA e.i. A, B, E, R, S Z

aeaelo erotat
ehttua tedhli
tpyuon

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

How to use Transposition Cipher Encryption

1. Arrange the plaintext according to 6 columns character and create a keyword eg. ZEBRAS
2. Look for Alphabetic sequence of the word ZEBRA e.i. A, B, E, R, S Z

aeaelo erotat
ehttua tedhli
tpyuon mtmasc

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

How to use Transposition Cipher Encryption

1. Look for Alphabetic sequence of the word ZEBRA e.i. A, B, E, R, S Z
2. The result will be…

Z	E	B	R	A	S
m	e	e	t	a	t
t	h	r	e	e	p
m	t	o	d	a	y
a	t	t	h	e	u
s	u	a	l	l	o
c	a	t	i	o	n

Meet at three pm today at the usual location

aeaelo erotat ehttua tedhli tpyuon mtmasc

Symmetrical Stream Cipher

- **Vernam** cipher: developed at AT&T; uses set of characters once per encryption process
- Digital bit-wise XOR
- The Vernam Cipher is based on the principle that each plaintext character from a message is 'mixed' with one character from a key stream.
- If a truly random key stream is used, the result will be a truly 'random' ciphertext which bears no relation to the original plaintext. In that case the cipher is similar to the unbreakable One-Time Pad (OTP).

Vernam Cipher Security

- The above procedure is 100% safe if, and only if, the following conditions are all met:
  1. There are only two copies of the key (OTP),
  2. Both sides of the communications link have the same key (OTP),
  3. The key (OTP) is used only once,
  4. The key (OTP) is destroyed immediately after use,
  5. The key (OTP) contains truly random characters,
  6. The equipment is hack proof or uncompromised,
  7. The key (OTP) was not compromised during transport.

How to use Vernam Cipher Encryption

1. Create a plaintext and group the characters 6 columns, find the value of the text in the alphabet and convert it to binary:

Meet at three pm today at the usual location

Plaintext								Plaintext Value in the Alphabet					
meetat	M	e	e	t	a	t		13	5	5	20	1	20
threep	t	h	r	e	e	p		20	8	18	5	5	16
mtoday	m	t	o	d	a	y		13	20	15	4	1	25
attheus	a	t	t	h	e	u		1	20	20	8	5	21
suallo	s	u	a	l	l	o		19	21	1	12	12	15
cation	c	a	t	i	o	n		3	1	20	9	15	14

How to use Vernam Cipher Encryption

Binary Value of the plaintext

Plaintext Value in Binary Digits					
00001101	00000101	00000101	00010100	00000001	00010100
00010100	00001000	00010010	00000101	00000101	00010000
00001101	00010100	00001111	00000100	00000001	00011001
00000001	00010100	00010100	00001000	00000101	00010101
00010011	00010101	00000001	00001100	00001100	00001111
00000011	00000001	00010100	00001001	00001111	00001110

How to use Vernam Cipher Encryption

Decimal Value of the OTP

Decimal Value of the One Time Pad					
14	25	24	20	7	19
8	6	22	19	14	14
14	14	12	2	2	13
18	3	26	5	24	15
25	9	11	3	18	26
7	6	19	19	25	21

How to use Vernam Cipher Encryption

- Create a key (One Time Pad)

One Time Pad - Key					
00001110	00011001	00011000	00010100	00000111	00010011
00001000	00000110	00010110	00010011	00001110	00001110
00001110	00001110	00001100	00000010	00000010	00001101
00010010	00000011	00011010	00000101	00011000	00001111
00011001	00001001	00001011	00000011	00010010	00011010
00000111	00000110	00010011	00010011	00011001	00010101

Binary XOR Operation

The inputs to a binary **XOR** operation can only be **0** or **1** and the result can only be **0** or **1**

The binary **XOR** operation (also known as the binary **XOR** function) will always produce a **1** output if either of its inputs is **1** and will produce a **0** output if both of its inputs are **0** or **1**.

If we call the inputs **A** and **B** and the output **C** we can show the **XOR** function as:

A		B		C
0	XOR	0	->	0
0	XOR	1	->	1
1	XOR	0	->	1
1	XOR	1	->	0

How to use Vernam Cipher Encryption

- Compute for the Cyphertext using XOR

Result of XOR (Plaintext, OTP) value in Binary digits					
00000011	00011100	00011101	00000000	00000110	00000111
00011100	00001110	00000100	00010110	00001011	00011110
00000011	00011010	00000011	00000110	00000011	00010100
00010011	00010111	00000100	00001101	00011101	00011010
00001010	00011100	00001010	00001111	00011110	00010101
00000100	00000111	00000111	00011010	00010110	00011011

- 00001101
- 00001110
- 00000011 = 3
- 00000101
- 00011001
- 00011100

How to use Vernam Cipher Encryption

- Convert the binary value into decimal, use the result and look for the value in the alphabet table

Decimal Value of the XOR(Plaintext, OTP)					
3	28	29	0	6	7
28	14	4	22	11	30
3	26	3	6	3	20
19	23	4	13	29	26
10	28	10	15	30	21
4	7	7	26	22	27

How to use Vernam Cipher Encryption

- Convert the binary value into decimal, use the result and look for the value in the alphabet table

Alphabet Value of the XOR Result					
C	B	C	Z	F	G
B	N	D	V	K	D
C	Z	C	F	C	T
S	W	D	M	C	Z
J	B	J	O	D	U
D	G	G	Z	V	A

How to use Vernam Cipher Encryption

- The result of converting the decimal value to alphabetic characters

- Plaintext
  Meet at three pm today at the usual location

- Ciphertext
  CBCZ FG BNDVK DC ZCFCT SW DMC ZJBJO DUDGGZVA

Cryptographic Algorithms

- Often grouped into two broad categories, symmetric and asymmetric; today's popular cryptosystems use hybrid combination of symmetric and asymmetric algorithms
- Symmetric and asymmetric algorithms distinguished by types of keys used for encryption and decryption operations

Cryptographic Algorithms (continued)

- Symmetric encryption: uses same “secret key” to encipher and decipher message
  - Encryption methods can be extremely efficient, requiring minimal processing
  - Both sender and receiver must possess encryption key
  - If either copy of key is compromised, an intermediate can decrypt and read messages
- The problem is that both the sender and the receiver must own the encryption key.
  - If either copy of the key is compromised, an intermediate can decrypt and read the messages.
  - One of the challenges of symmetric key encryption is getting a copy of the key to the receiver, a process that must be conducted out-of-band to avoid interception.

Cryptographic Algorithms (continued)

- There are a number of popular symmetric encryption cryptosystems.
- Data Encryption Standard (DES): one of most popular symmetric encryption cryptosystems. developed in 1977 by IBM and based on the Data Encryption Algorithm (DEA)
- On May 15, 1973, during the reign of Richard Nixon, the National Bureau of Standards (NBS) published a notice in the Federal Register soliciting proposals for cryptographic algorithms to protect data during transmission and storage.
- Adopted by NIST in 1976 as federal standard for encrypting non-classified information

Cryptographic Algorithms (continued)

- DEA uses a 64-bit block size and a 56-bit key. The algorithm begins by adding parity bits to the key (resulting in 64 bits) and then applies the key in 16 rounds of XOR, substitution, and transposition operations.
- With a 56 bit key, the algorithm has 256 possible keys to choose from (over 72 quadrillion).
- DES was cracked in 1997 when Rivest-Shamir-Aldeman (RSA) put a bounty on the algorithm.

DES Encryption

- This is the encrypted form of
- **M** = 0123456789ABCDEF
- **C** = 85E813540F0AB405

How DES Work?

1. **Step 1: Create 16 subkeys, each of which is 48-bits long.** Create a key.

   K = 133457799BBCDFF1

2. Convert the key into Binary (ASCII ➔ Hex ➔ Binary)

   00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

3. Create blocks of 64 bits

8 Bits									8 Bits									8 Bits									8 Bits							
0	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
0	0	0	1	0	0	1	1		0	0	1	1	0	1	0	0		0	1	0	1	0	1	1	1		0	1	1	1	1	0	0	1

8 Bits									8 Bits									8 Bits									8 Bits							
32	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
1	0	0	1	1	0	1	1		1	0	1	1	1	1	0	0		1	1	0	1	1	1	1	1		1	1	1	1	0	0	0	1

How DES Work?

a. Change the order using the permutation table – these is a randomly generated numbers from 0 - 55

PC – 1 (56 bits)						
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

How DES Work?

b. Result after changing the order of the original text using the random permutation table

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

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
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
1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	0	0	0

How DES Work?

c. Next, split this key into left and right halves, L0 and R0, where each half has 28 bits.

LO	1	1	1	1	0	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	1	1	1
RO	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1

How DES Work?

d. With L0 and R0 Refined, we now to create sixteen blocks Ln and Rn, 1<=n<=16. Each pair of blocks Ln and Rn is formed from the previous pair Ln-1 and Rn-1, respectively, for n = 1, 2, ..., 16

Iteration Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of Left Shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

How DES Work?

Example: From original pair, L_0 and R_0 we obtain:

L_0 = 11110000110011001010101011111
R_0 = 01010101011001100111100011111

L_0 = 11110000110011001010101011111
R_0 = 01010101011001100111100011111

How DES Work?

Example: From original pair, L_0 and R_0 we obtain:

L_0 = 11110000110011001010101011111

R_0 = 01010101011001100111100011111

How DES Work?

Example: From original pair, L_0 and R_0 we obtain:

L_1 = 11100001100110010101010111111
R_1 = 10101010110011001111100011110

How DES Work?

Example: Output from the shifted pair L_0 and R_0 we obtain:

L_1 = 111000011001100101010101111 1

R_1 = 101010110011001111100011110

How DES Work?

Example: From shifted pair, L_1 and R_1 we obtain L_2 and R_2

L_1 = 1110000110011001010101011111

R_1 = 1010101011001100111100011110

How DES Work?

Example: From original pair, L_0 and R_0 we obtain:

L_2 = 11000011001100101010101111111
R_2 = 01010101100110011110001111101

How DES Work?

Example: From original pair, L_0 and R_0 we obtain:

L_2 = 11000011001100101010101111111
R_2 = 01010101100110011110000111101

How DES Work?

Example: From original pair, L_0 and R_0 we obtain L_1 and R_1 and L_1 and R_1 using 1 shift

L_0 = 11110000110011001010101011111

R_0 = 01010101011001100111100011111

L_1 = 11100001100110010101010111111

R_1 = 10101010110011001111000111110

L_2 = 11000011001100101010101111111

R_2 = 01010101100110011110001111101

How DES Work?

Basing on the shifting table, L3 and R3 are obtained from L2 and R2, respectively, by **two left shifts**, and L16 and R16 are obtained from L15 and R15, respectively, by one left shift.

In all cases, by a single left shift is meant a rotation of the bits one place to the left, so that after one left shift the bits in the 28 positions are the bits that were previously in positions 2, 3,..., 28, 1.

How DES Work?

Example: From shifted pair, L_2 and R_2 we obtain:

L_2 = 11000011001100101010101111111
R_2 = 01010101100110011110000111101

How DES Work?

Example: From shifted pair, L_2 and R_2 we obtain:

L_2 = 11100001100110010101010111111
R_2 = 01010101100110011110000111101

How DES Work?

Example: From shifted pair, L_2 and R_2 we obtain:

L_3 = 00001100110010101010111111111
R_3 = 01010110011001111000111101101

How DES Work?

Example: From shifted pair, L_2 and R_2 we obtain:

L_3 = 00001100110010101010111111111
R_3 = 01010110011001111100011110101

How DES Work?

Example: From shifted pair, L_3 and R_3 we obtain:

L_3 = 00001100110010101010111111111
R_3 = 01010110011001111000111101 01

How DES Work?

Example: From shifted pair, L_3 and R_3 we obtain:

L_4 = 00110011001010101011111111100
R_4 = 01011001100111100011110101010

How DES Work?

This will go on until it satisfies the switch table

Iteration Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of Left Shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

How DES Work?

L_5 = 11001100101010101111111110000
R_5 = 01100110011110001111010101010

L_6 = 00110010101010111111111000011
R_6 = 10011001111000111101010101010

L_7 = 11001010101011111111100001100
R_7 = 01100111100011110101010101010

L_8 = 00101010101111111110000110011
R_8 = 10011110001111010101011001100

How DES Work?

L_9 and L_9 Goes back to 1 - shift

L_9 = 01010101011111111000011100110
R_9 = 00111100011110101010101110011

L_{10} and L_{10} goes back again to 2 - shifts

L_{10} = 01010101111111100001100111001
R_{10} = 11110001111010101010111001100

How DES Work?

L_{12} = 01011111111000011001100110010101
R_{12} = 00011110101010101100110011001111

L_{13} = 01111111100001100110011001010101
R_{13} = 01111010101010110011001111100

L_{14} = 11111110000110011001010101010101
R_{14} = 11101010101011001100111100001

L_{15} = 11111000011001100101010101010111
R_{15} = 10101010101100110011111000111

How DES Work?

L_{16} and L_{16} goes back again to 1 - shift

L_{16} = 11110000110011001010101011111
R_{16} = 01010101011001100111100011111

How DES Work?

1. Create a Key of 48 bits by performing DES Algorithm
   a. Change the order using the permutation table – PC 2, a randomly generated numbers from 0 - 47

- We now form the keys K_n, for 1<=n<=16, by applying the following permutation table to each of the concatenated pairs L_nR_n. Each pair has 56 bits, but **PC-2** only uses 48 of these.

How DES Work?

Create a Key of 48 bits

PC – 2 (48 bit)					
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

The first bit of $\boldsymbol{K_n}$ is the 14th bit of $\mathbf{L_n}\boldsymbol{R_n}$, the second bit the 17th, and so on, ending with the 48th bit of $\boldsymbol{K_n}$ being the 32th bit of $\mathbf{L_n}\boldsymbol{R_n}$.

How DES Work?

- Use 48 bit permutation for L_1 and R_1

L_1 = 11100001100110010101011111
R_1 = 10101010110011001111100011110

L1R1 is equals to…
1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

Result for 48-bit permutation K1

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

How? refer to the table for conversion

Permutation Table using 48-bit

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
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
0	0	0	1	1	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	1	1	1	1	1

25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
52	31	37	47	55	30	40	51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32
1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0

Thus the result…

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

How DES Work?

- Use 48 bit permutation for L_1 and R_1

L2R2 is equals to…
110000110011001010101011111101010101100110011110001111 01

L3R3 is equals to…
000011001100101010101111111101010110011001111000111 0101

Result for 48-bit permutation K2 and K3

K_2 = 011110 011010 111011 011001 110110 111100 100111 100101
K_3 = 010101 011111 110010 001010 010000 101100 111110 011001

Permutation Table using 48-bit

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
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
0	1	1	1	1	0	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	1	1	1	1	1	1	0	0	1	0	0	0	1	0	1	0	0

25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
52	31	37	47	55	30	40	51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32
1	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1	1	1	0	0	1	0	1
1	0	0	0	1	1	0	1	1	0	0	1	1	1	1	1	0	0	1	1	0	0	1

Thus the result…

K_2 = 011110 011010 111011 011001 110110 111100 100111 100101

K_3 = 010101 011111 110010 001010 010000 101100 111110 011001

Permutation Table using 48-bit

K_4 = 011100 101010 110111 010110 110110 110011 010100 011101
K_5 = 011111 001110 110000 000111 111010 110101 001110 101000
K_6 = 011000 111010 010100 111110 010100 000111 101100 101111
K_7 = 111011 001000 010010 110111 111101 100001 100010 111100
K_8 = 111101 111000 101000 111010 110000 010011 101111 111011
K_9 = 111000 001101 101111 101011 111011 011110 011110 000001
K_{10} = 101100 011111 001101 000111 101110 100100 011001 001111
K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110
K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001
K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001
K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010
K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010
K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

Summary of Step 1

1. Created 16 subkeys, each of which is 48-bits long.
   a. We created a KEY
   b. Convert the key into binary of 64 bit long
   c. We permutated or changed the order of the 64 bit binary digit using the **PC-1 table**
   d. We shifted the binary according to the shift table
   e. We permutated the shifted key on a 48 bit long key **PC-2 table.**

Step 2: Encode each 64-bit block of data.

1. Create a 64 bits of the message data **M**
   - **M** = 0123456789ABCDEF
   - **M** = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
2. *Use an initial permutation* (**IP)** of the 64 bits of the message data. (use IP table)
3. Perform permutation similar to 56 bit and 48 bit permutation.

64-bit long message in binary

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1

34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
0	0	0	1	0	0	1	1	0	1	0	1	0	1	1	1

50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
1	0	0	1	1	0	1	1	1	1	0	1	1	1	1

Step 2: Encode each 64-bit block of data.

Initial Permutation (IP)							
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

Permutated Message in IP 64-bit table

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4	62
1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
54	46	38	30	22	14	6	64	56	48	40	32	24	16	8	57
1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1

33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	41	33	25	17	9	1	59	51	43	35	27	19	11	3	61
1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
53	45	37	29	21	13	5	63	55	47	39	31	23	15	7
1	1	1	0	0	0	0	1	0	1	0	1	0	1	0

Permutated Message in IP 64-bit table

- **M** = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
- **IP** = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

Step 2: Continuation

- Next divide the permuted block **IP** into a left half L_0 of 32 bits, and a right half R_0 of 32 bits.

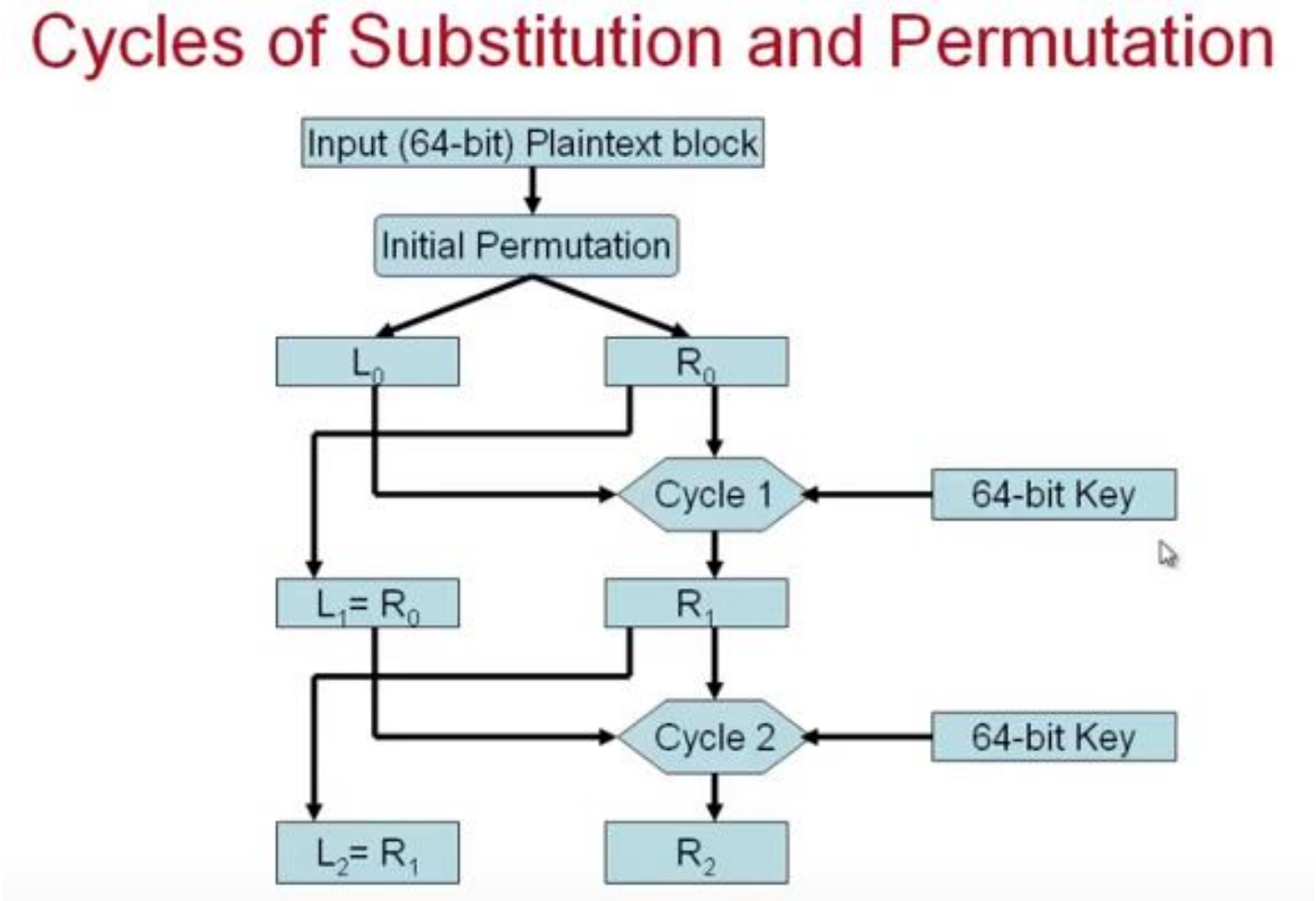
- L_0 = 1100 1100 0000 0000 1100 1100 1111 1111
- R_0 = 1111 0000 1010 1010 1111 0000 1010 1010

Step 2: Continuation

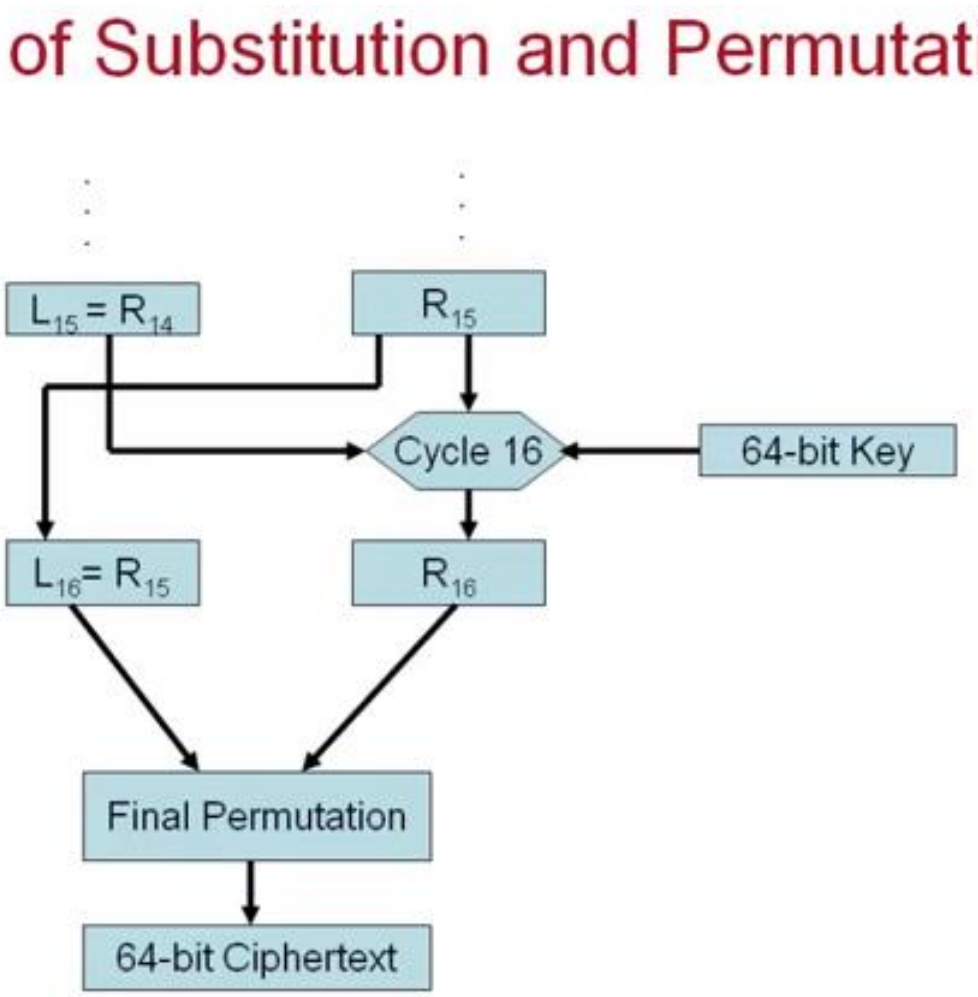
- We now proceed through 16 iterations, for 1<=***n***<=16, using a function ***f*** which operates on two blocks--a data block of 32 bits and a key K_n of 48 bits--to produce a block of 32 bits.
- **Let + denote XOR addition, (bit-by-bit addition modulo 2)**. Then for **n** going from 1 to 16 we calculate

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

Step 2: Continuation



Step 2: Continuation



Final Output

- This is the encrypted form of
- **M** = 0123456789ABCDEF
- **C** = 85E813540F0AB405
- Rivest-Shamir-Aldeman (Cracked DES)

Cryptographic Algorithms (continued)

- Asymmetric Encryption (public key encryption)
  - Uses two different but related keys; either key can encrypt or decrypt message
  - If Key A encrypts message, only Key B can decrypt
  - Highest value when one key serves as private key and the other serves as public key

Examples of Asymmetric Encryption

- Examples of **asymmetric encryption or public key encryption** are **DSA, RSA and PGP**.
- **RSA** is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys. This is also called *public key cryptography*, because one of them can be given to everyone.
- The other key must be kept private. It is based on the fact that finding the factors of an integer is hard (the factoring problem).

Examples of Asymmetric Encryption

- RSA stands for Ron **Rivest**, Adi **Shamir** and Leonard **Adleman**, who first publicly described it in 1978.
- A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key.
- The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message.
- Using the acoustic cryptanalysis, carried out by Daniel Genkin, Adi Shamir (who co-invented RSA), and Eran Tromer, uses what's known as a *side channel attack*.

How does RSA Work?

- **Step 1 –** Choose two prime numbers, Prime1 and Prime2

- Prime1 and Prime2 should be very large prime numbers, at minimum 100 digits long but as larger is more secure and less efficient.
- Prime 1 and Prime2 should not be the same prime number

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,
53, 59, 61, 67, 71, 73, 79, 83, 89, 97;
```

- p = prime 1; q = prime 2; n = p x q

How does RSA Work?

- **Step 1 –** Choose two prime numbers, Prime1 and Prime2

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,
53, 59, 61, 67, 71, 73, 79, 83, 89, 97;
```

p = 2 ; q = 7 ; n = p x q

n = 2 x 7

n = 14

(this number will become the modulo or remainder of the key)

How does RSA Work?

- **Step 2 –** Get the value of *n* (Product of 2 prime numbers

p = 2 ; q = 7 ; *n* = p x q

n = 2 x 7

***n* = 14**

(this number will become the modulo or remainder of the key)

How does RSA Work?

Step 3 – Find the Totient of ProductOfPrimes

- Totient - The **totient** function , also called Euler's **totient** function, is defined as the number of positive integers that are relatively prime to (i.e., do not contain any factor in common with) , where 1 is counted as being relatively prime to all numbers.
- Represented with the symbol Φ (Phi)
- Totient = ΦN

How does RSA Work?

Step 3 – Find the Totient of ProductOfPrimes

- Get the Totient = φ(14)
- Φ – Phi
- Look for the numbers that has a common factor of 1 – 14, and 2 and 7
- Cross out the numbers

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

How does RSA Work?

Step 3 – Find the Totient of ProductOfPrimes

- Cross out the numbers that has a common factor of 1 – 14, and 2 and 7, except 1.
- Remove all even numbers because it has factors of 14, and similar factors of 2 and 7

1

3

5

7

9

11

13

How does RSA Work?

Step 3 – Find the Totient of ProductOfPrimes

- Cross out the numbers that has a common factor of 1 – 14, and 2 and 7
- Remove 7
- That leaves 1, 3, 5, 9, 11, 13 (6 remaining numbers)
- They are called co-prime with 14, since they share no common factors with 14.
- **Totient = ΦN = 6**

1

3

5

9

11

13

How does RSA Work?

Step 3 – Find the Totient of ProductOfPrimes

- Totient = φ(ProductOfPrimes)
- Φ – Phi
- Totient = Φ(14)
- Totient = (Prime1 -1) * (Prime2 – 1)
- Totient = (2-1) * (7– 1)
- Totient = (1) * (6)
- **Totient = ΦN = 6**

How does RSA Work?

Step 4 – Choose a number for *e (Encryption key)*

e – stands for Encryption key and should obey the following rules:

- Should be 1 < e < ΦN
- Should be co-prime with N and ΦN

- 1 < e < ΦN = { 2, 3, 4, 5}
- co-prime with N and ΦN = **5**
- {2,3 4} are factors of 14 and 6 while 5 is not

e = 5 ; encryption key or the lock key$_e$(5,14)

How does RSA Work?

Step 5 – Choose a number for *d (decryption)*

d – stands for Decryption and should obey the following rule

$d = de(\mathrm{mod}\ \Phi N) = 1$

$d = 5d(\mathrm{mod}\ (6)) = 1$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
5	4	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0	5

d = 11 ; $key_d(11,14)$

Encrypting and Decrypting RSA

1. Create a message:

   Meet at three pm

2. Convert the message into its numeric value – using the alphabet table, the following numeric value is identified

m	e	e	t	a	t	t	h	r	e	e	p	m
13	5	5	20	1	20	20	8	18	5	5	16	13

3. Compute for the ciphertext using the formula;

$$e = 5, \text{mod} = 14; c = m^e \pmod{(14)}$$

$$c = 13^5 \bmod 14 ;$$

$$c = 371293 \bmod 14$$

$$c = 13$$

Encrypting and Decrypting RSA

4. Look for the value of the ***c*** in the alphabetic table:

c = 13, c = 3 … c = 13

Iterate until the entire message is converted into a cipher

ciper	13	3	3	6	1	6	6	8	2	3	3	4	13
cipertext	m	c	c	f	a	f	f	h	b	c	c	d	m

Message

Meet at three pm

Cyphertext

Mccfaffhbccdm

Encrypting and Decrypting RSA

1. To decrypt, reverse the process of encryption :

   mccfaffhbccdm

2. Convert the message into its numeric value – using the alphabet table, the following numeric value is identified

Cipertext	m	c	c	f	a	f	f	h	b	c	c	d	m
value	13	3	3	6	1	6	6	8	2	3	3	4	13

3. Compute for the message using the formula;

$$d = 11, \text{mod} = 14; m = c^d \ (\text{mod}\ (14)$$

$$c = 13^{11} \ \text{mod}\ 14 ;$$

$$m = 1792160394037 \ \text{mod}\ 14$$

$$m = 13$$

Encrypting and Decrypting RSA

4. Look for the value of the *m* in the alphabetic table:

m = 13, m = 5 … m = 13

Iterate until the entire message is converted into a cipher

m	13	5	5	20	1	20	20	8	18	5	5	16	13
Ciphertext	m	e	e	t	a	t	t	h	r	e	e	p	m

Cyphertext

Mccfaffhbccdm

Message

Meet at three pm

Encryption Key Power

Number of Bits in Key	Odds of Cracking: 1 in	Estimated Time to Crack*
8	256	.000032 seconds
16	65,536	.008192 seconds
24	16,777,216	2.097 seconds
32	4,294,967,296	8 minutes 56.87 seconds
56	72,057,594,037,927,900	285 years 32 weeks 1 day
64	18,446,744,073,709,600,000	8,090,677,225 years
128	3.40282E+38	5,257,322,061,209,440,000,000 years
256	1.15792E+77	2,753,114,795,116,330,000,000,000,000, 000,000,000,000,000,000,000 years
512	1.3408E+154	608,756,305,260,875,000,000,000,000,000, 000,000,000,000,000,000,000,000,000,000, 000,000,000,000,000,000,000,000,000,000, 000,000,000 years

[NOTE] *Estimated Time to Crack is based on a general-purpose personal computer performing eight million guesses per second.

Hybrid Cryptography Systems

- Except with digital certificates, pure asymmetric key encryption not widely used
- Asymmetric encryption more often used with symmetric key encryption, creating hybrid system
- Diffie-Hellman Key Exchange method: most common hybrid system; provided foundation for subsequent developments in public key encryption

Cryptography Tools

- Public Key Infrastructure (PKI)**:** integrated system of software, encryption methodologies, protocols, legal agreements, and third-party services enabling users to communicate securely
- PKI systems based on public key cryptosystems; include digital certificates and certificate authorities (CAs)

Cryptography Tools (continued)

- PKI protects information assets in several ways:
  - Authentication
  - Integrity
  - Privacy
  - Authorization
  - Nonrepudiation

PKI protects information assets in several ways:

- **Authentication.** Digital certificates in a PKI system permit parties to validate the identity of other of the parties in an Internet transaction.
- **Integrity.** A digital certificate demonstrates that the content signed by the certificate has not been altered while being moved from server to client.

PKI protects information assets in several ways:

- **Privacy.** Digital certificates keep information from being intercepted during transmission over the Internet.
- **Authorization.** Digital certificates issued in a PKI environment can replace user IDs and passwords, enhance security, and reduce some of the overhead required for authorization processes and controlling access privileges.

PKI protects information assets in several ways:

- **Nonrepudiation.** Digital certificates can validate actions, making it less likely that customers or partners can later repudiate a digitally signed transaction.

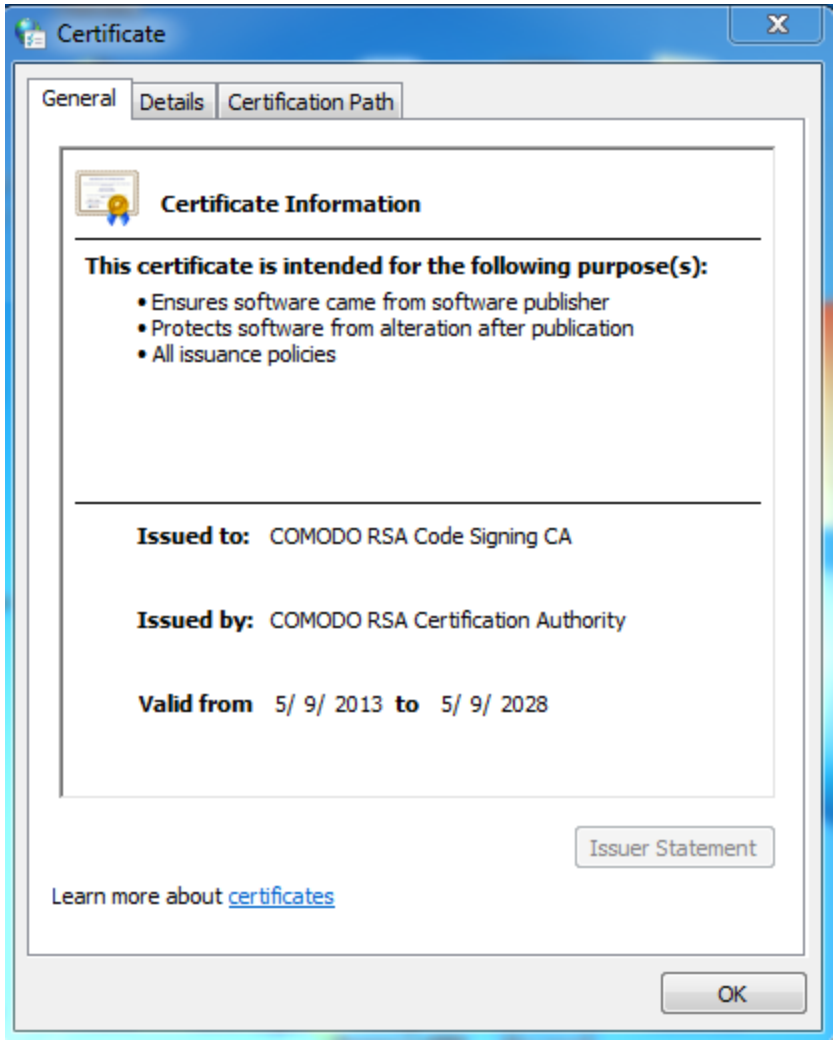
Digital Signatures

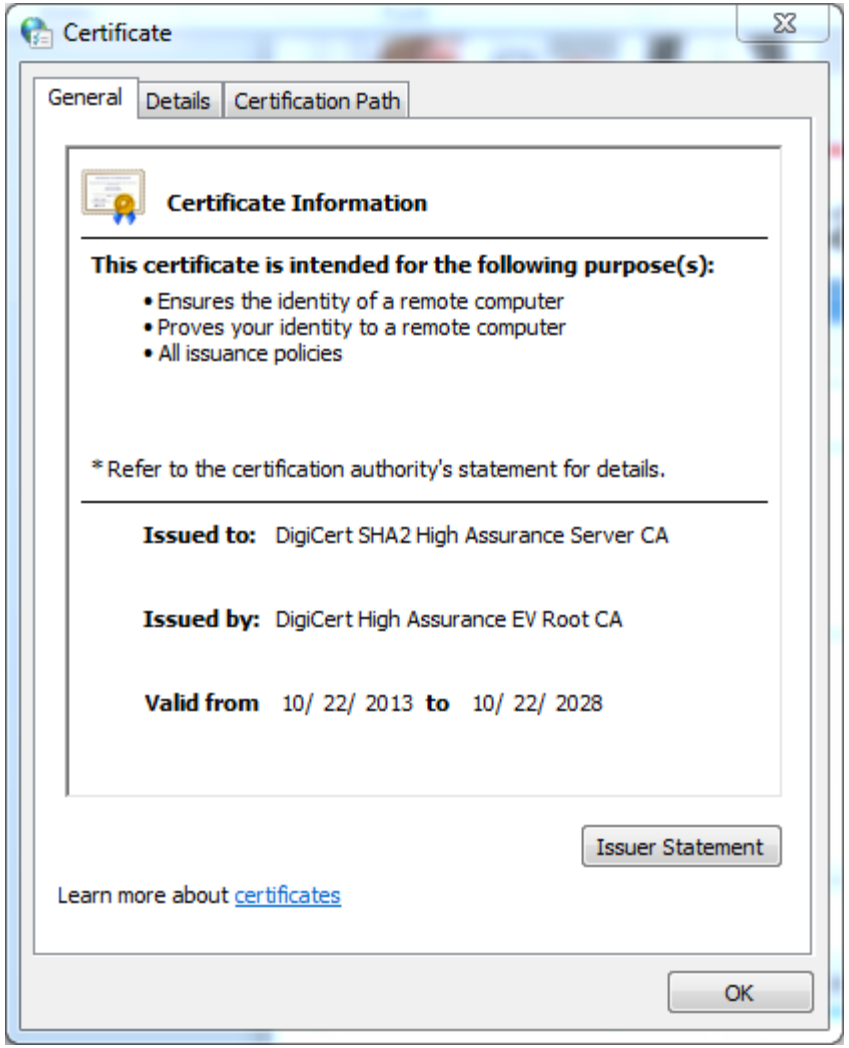
- Encrypted messages that can be mathematically proven to be authentic
- Created in response to rising need to verify information transferred using electronic systems
- Asymmetric encryption processes used to create digital signatures

Digital Certificates

- Electronic document containing key value and identifying information about entity that controls key
- Digital signature attached to certificate’s container file to certify file is from entity it claims to be from

Digital Certificates and Signatures





Steganography

- Process of hiding information; in use for a long time
- Most popular modern version hides information within files appearing to contain digital pictures or other images
- Some applications hide messages in .bmp, .wav, .mp3, and .au files, as well as in unused space on CDs and DVDs

Protocols for Secure Communications

- Secure Socket Layer (SSL) protocol: uses public key encryption to secure channel over public Internet
- Secure Hypertext Transfer Protocol (S-HTTP): extended version of Hypertext Transfer Protocol; provides for encryption of individual messages between client and server across Internet
- S-HTTP is the application of SSL over HTTP; allows encryption of information passing between computers through protected and secure virtual connection
- HTTPS and S-HTTP

Protocols for Secure Communications (continued)

- Securing E-mail with S/MIME, PEM, and PGP
  - Secure Multipurpose Internet Mail Extensions (S/MIME): builds on Multipurpose Internet Mail Extensions (MIME) encoding format by adding encryption and authentication
  - Privacy Enhanced Mail (PEM): proposed as standard to function with public key cryptosystems; uses 3DES symmetric key encryption
  - Pretty Good Privacy (PGP): uses IDEA Cipher for message encoding

Protocols for Secure Communications (continued)

- Securing Web transactions with SET, SSL, and S-HTTP
  - Secure Electronic Transactions (SET): developed by MasterCard and VISA in 1997 to provide protection from electronic payment fraud
  - Uses DES to encrypt credit card information transfers
  - Provides security for both Internet-based credit card transactions and credit card swipe systems in retail stores

Protocols for Secure Communications (continued)

- Securing TCP/IP with IPSec
  - Internet Protocol Security (IPSec): open source protocol to secure communications across any IP-based network
  - IPSec designed to protect data integrity, user confidentiality, and authenticity at IP packet level

Protocols for Secure Communications (continued)

- IPSec combines several different cryptosystems: Diffie-Hellman; public key cryptography; bulk encryption algorithms; digital certificates
- In IPSec, IP layer security obtained by use of application header (AH) protocol or encapsulating security payload (ESP) protocol

Protocols for Secure Communications (continued

- Freeware and low-cost commercial PGP versions are available for many platforms
- PGP security solution provides six services: authentication by digital signatures; message encryption; compression; e-mail compatibility; segmentation; key management

Hashing

Engr. Juliet S. Mendez

MBA, CCNA

Hashing

- Any function that can be used to map data of arbitrary size of data of fixed size.
- Used in checksums, check digits, fingerprints, lossy compression, randomization functions, error-correcting codes and ciphers
- File Verification
- Password Storage
- Database Searching

Popular Cryptographic Hash Functions

- Message Digest 5(MD5) Algorithm
- Secure Hashing Algorithm (SHA)
  - SHA- 1
  - SHA- 256

MD5 Algorithm

- 128 bit hash value
- Suffer from extensive vulnerabilities
- Can still be used as a checksum

SHA -1

- 160 bit hash value
- 40 digits long
- Microsoft, Google, Apple and Mozilla : Stop accepting SHA-1 SSL certificates by 2017
- Used in distributed revision control systems(GIT, Mercurial & Monotone)

SHA-2

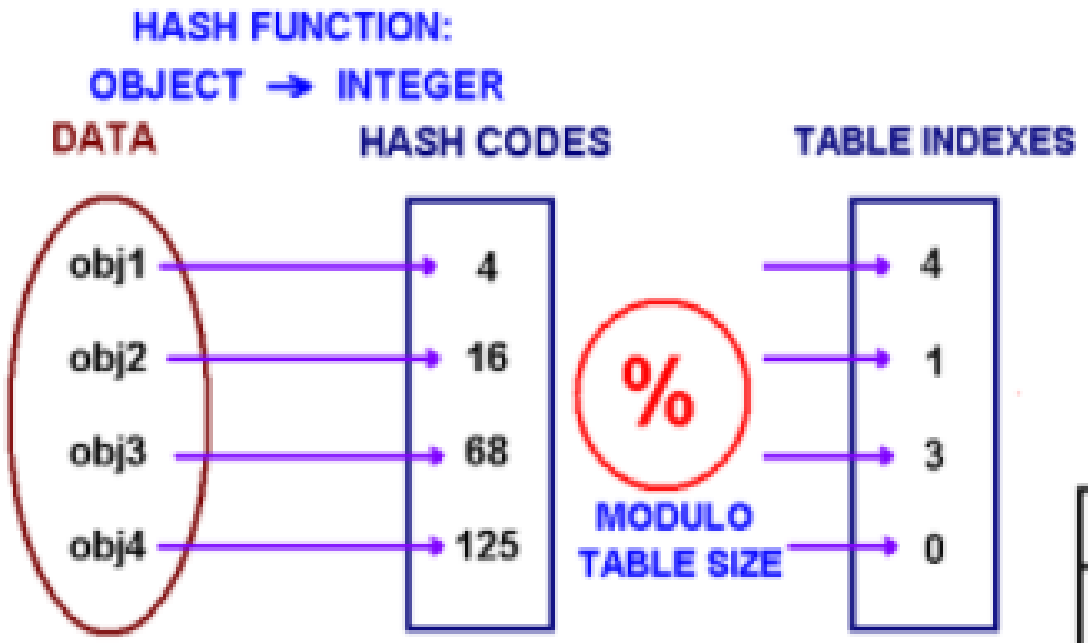
- Consist of six hash functions with hash values
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512

Published in	Year	Attack method	Attack	Variant	Rounds	Complexity
New Collision Attacks Against Up To 24-step SHA-2[32]	2008	Deterministic	Collision	SHA-256	24/64	$2^{28.5}$
				SHA-512	24/80	$2^{32.5}$
Preimages for step-reduced SHA-2[33]	2009	Meet-in-the-middle	Preimage	SHA-256	42/64	$2^{251.7}$
					43/64	$2^{254.9}$
				SHA-512	42/80	$2^{502.3}$
					46/80	$2^{511.5}$
Advanced meet-in-the-middle preimage attacks[34]	2010	Meet-in-the-middle	Preimage	SHA-256	42/64	$2^{248.4}$
				SHA-512	42/80	$2^{494.6}$
Higher-Order Differential Attack on Reduced SHA-256[2]	2011	Differential	Pseudo-collision	SHA-256	46/64	2^{178}
					33/64	2^{46}
Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 family[1]	2011	Biclique	Preimage	SHA-256	45/64	$2^{255.5}$
				SHA-512	50/80	$2^{511.5}$
			Pseudo-preimage	SHA-256	52/64	2^{255}
				SHA-512	57/80	2^{511}
Improving Local Collisions: New Attacks on Reduced SHA-256[35]	2013	Differential	Collision	SHA-256	31/64	$2^{65.5}$
			Pseudo-collision	SHA-256	38/64	2^{37}
Branching Heuristics in Differential Collision Search with Applications to SHA-512[36]	2014	Heuristic differential	Pseudo-collision	SHA-512	38/80	$2^{40.5}$
Analysis of SHA-512/224 and SHA-512/256[37]	2016	Differential	Collision	SHA-256	28/64	practical
				SHA-512	27/80	practical
			Pseudo-collision	SHA-512	39/80	practical

SHA- 3

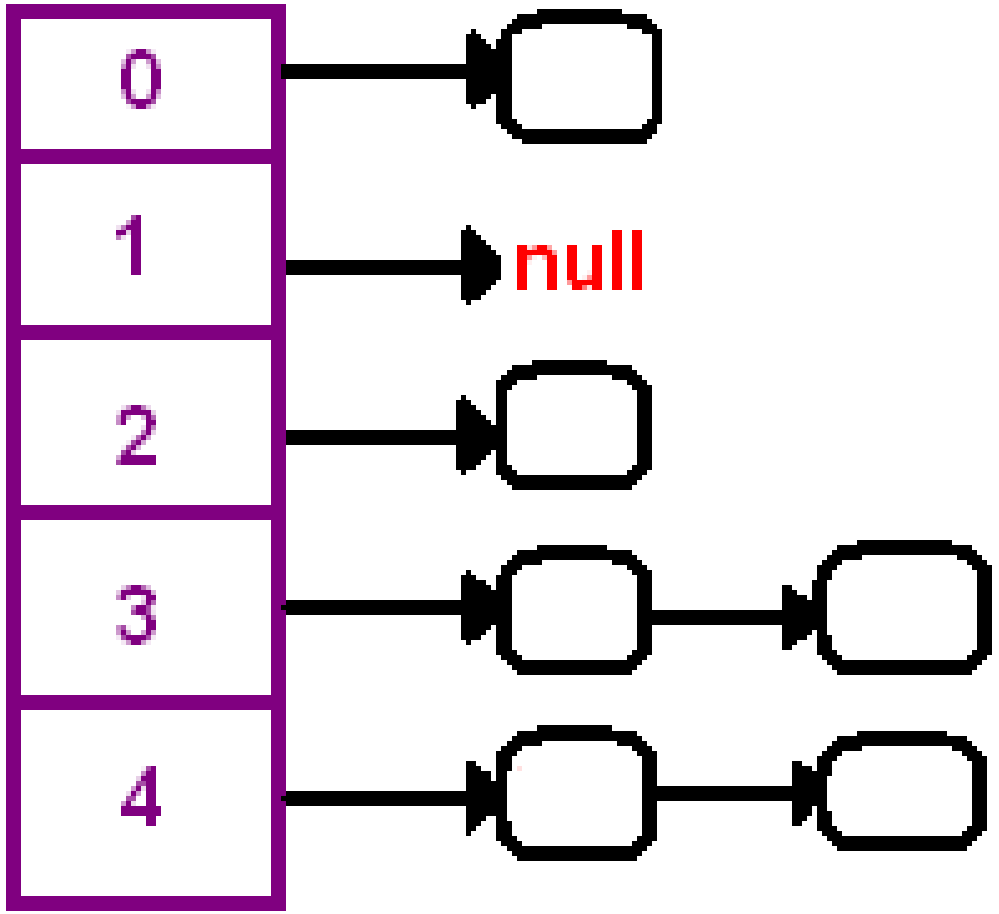
- released on August 5, 2015 by NIST
- Is a subset of Keccak
- Data is absorbed into sponge, result is squeezed out.

Storing Hash Function



HASH TABLE

OBJ4	OBJ2		OBJ3	OBJ1
0	1	2	3	4


0
1
null
2
3
4

JAVA Sample Code

```
17      public static void main(String[] args) {
18          // TODO code application logic here
            Integer obj1 = new Integer(2009);
            String obj2 = new String("ABC");
21          System.out.println("hashCode for an integer is " + obj1.hashCode());
22          System.out.println("hashCode for a string is " + obj2.hashCode());
23
24      }
```

Formula

$$s.charAt(0) * 31^{n-1} + s.charAt(1) * 31^{n-2} + s.charAt(2) * 31^{n-3} + \ldots + s.charAt(n-1)$$

where *s* = string value in ASCII and *n*= length

Sample Computation:

ABC

ABC = ‘A’ * 31^2 + ‘B’ * 31 + ‘C’

ABC = 65 * 31^2 + 66 * 31 + 67

ABC = 64578

Attack on Crytosystems

Attacks on Cryptosystems

- Attempts to gain unauthorized access to secure communications have typically used brute force attacks (ciphertext attacks)

- Attacker may alternatively conduct known-plaintext attack or selected-plaintext attach schemes

Man-in-the-Middle Attack

- Designed to intercept transmission of public key or insert known key structure in place of requested public key
- From victims' perspective, encrypted communication appears to be occurring normally, but in fact attacker receives each encrypted message, decodes, encrypts, and sends to originally intended recipient
- Establishment of public keys with digital signatures can prevent traditional man-in-the-middle attack

Correlation Attacks

- Collection of brute-force methods that attempt to deduce statistical relationships between structure of unknown key and ciphertext
- Differential and linear cryptanalysis have been used to mount successful attacks
- Only defense is selection of strong cryptosystems, thorough key management, and strict adherence to best practices of cryptography in frequency of changing keys

Dictionary Attacks

- Attacker encrypts every word in a dictionary using same cryptosystem used by target
- Dictionary attacks can be successful when the ciphertext consists of relatively few characters (e.g., usernames, passwords)

Timing Attacks

- Attacker eavesdrops during victim’s session; uses statistical analysis of user’s typing patterns and inter-keystroke timings to discern sensitive session information
- Can be used to gain information about encryption key and possibly cryptosystem in use
- Once encryption successfully broken, attacker may launch a replay attack (an attempt to resubmit recording of deciphered authentication to gain entry into secure source

Defending From Attacks

- No matter how sophisticated encryption and cryptosystems have become, if key is discovered, message can be determined
- Key management is not so much management of technology but rather management of people

Video Clip – Cyber Security

- Video Clip showing simple definition about cybersecurity, how does it work and why do we need it.
- Cyber codes

Thank you

Have a nice day!!!

Answer to the Ciphertext

WELCOME TO CEBU
CARAGA STATE UNIVERSITY
CABADBARAN CAMPUS

Summary

- Cryptography and encryption provide sophisticated approach to security
  - Many security-related tools use embedded encryption technologies
  - Encryption converts a message into a form that is unreadable by the unauthorized
- Many tools are available and can be classified as symmetric or asymmetric, each having advantages and special capabilities
- Strength of encryption tool dependent on key size but even more dependent on following good management practices
- Cryptography is used to secure most aspects of Internet and Web uses that require it, drawing on extensive set of protocols and tools designed for that purpose
- Cryptosystems are subject to attack in many ways