M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

Course Name: Cryptography and Network Security

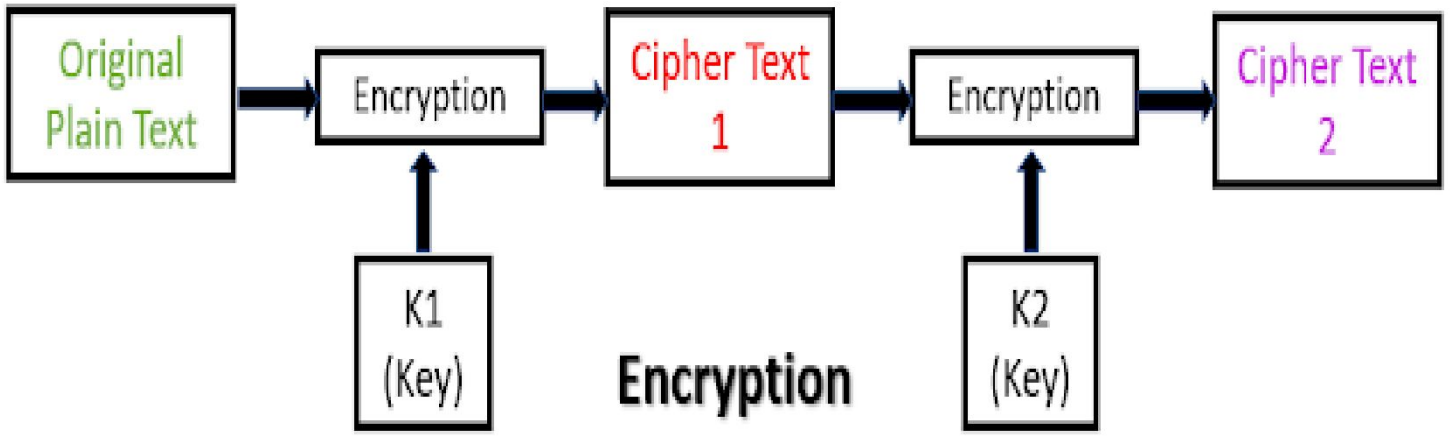
Course Code – CSE555

Credits - 3:0:0

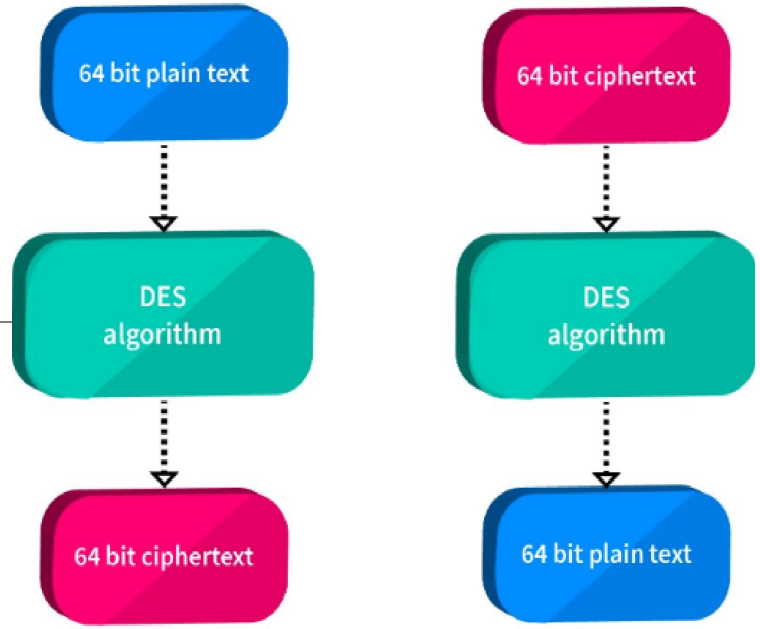
Term: Oct 2024 – Jan 2025

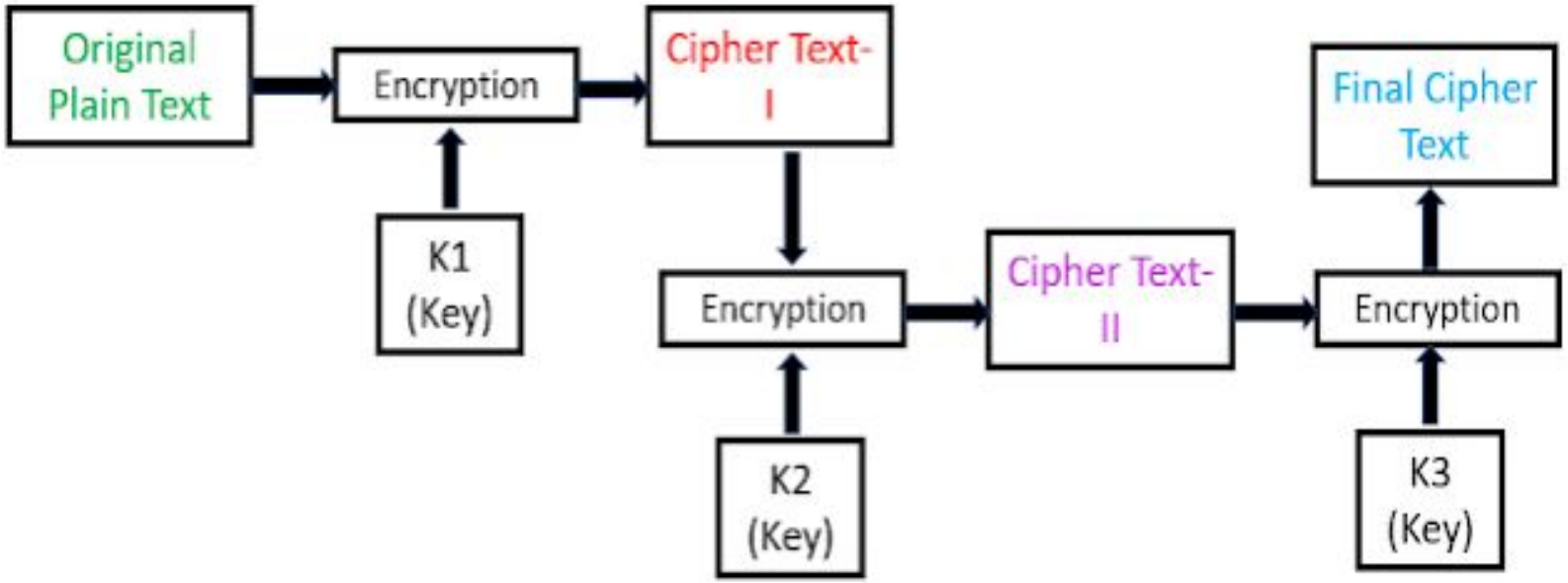
UNIT -3

Prepared by: Dr. Sangeetha. V
Associate Professor


Original Plain Text
Encryption
Cipher Text 1
Encryption
Cipher Text 2
K1 (Key)
K2 (Key)

Encryption


64 bit plain text
DES algorithm
64 bit ciphertext
Encryption
64 bit ciphertext
DES algorithm
64 bit plain text
Decryption

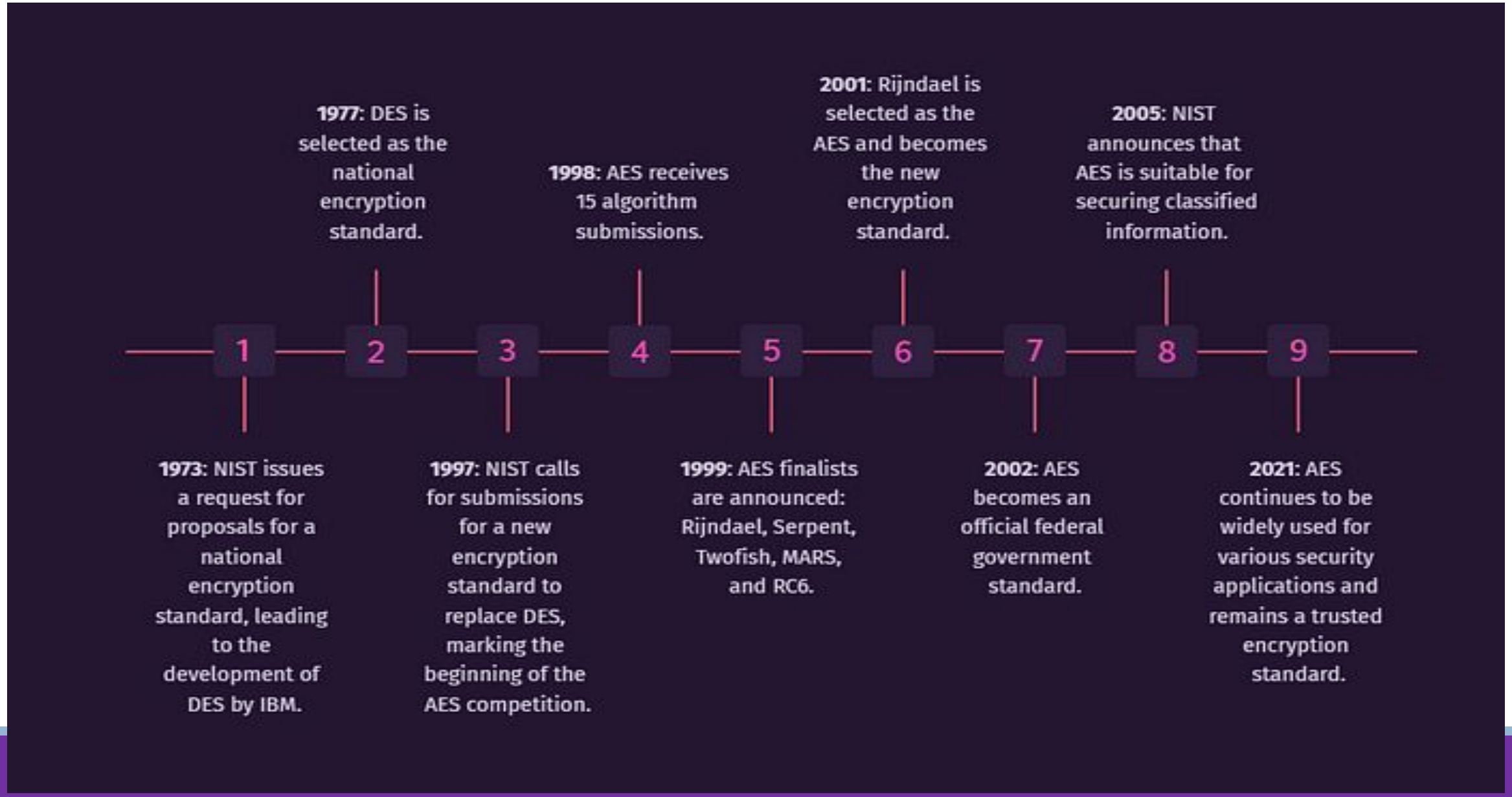

Original Plain Text
Encryption
Cipher Text-I
K1 (Key)
Encryption
Cipher Text-II
Encryption
Final Cipher Text
K2 (Key)
K3 (Key)

Why Was the AES Encryption Algorithm necessary?

Chronology of DES Cracking	
Broken for the first time	1997
Broken in 56 hours	1998
Broken in 22 hours and 15 minutes	1999
Capable of broken in 5 minutes	2021

Factors	AES	3DES	DES
Key Length	128, 192 or 256 bits	(k1, k2 and k3) 168 bits (k1 and k2 are same) 112 bits	56-bit
Cipher type	Symmetric block cipher	Symmetric block cipher	Symmetric block cipher
Block size	128,192, or 256 bits	64 bits	64 bits
Developed	2000	1978	1977
Cryptanalysis resistance	Strong against differential, truncated differential, linear, interpolation and square attacks	Vulnerable to differential; Brute Force Attacker could analyze plain text using differential cryptanalysis	Vulnerable to differential and linear cryptanalysis; weak substitution tables
Security	Considered Secure	One Only weakness, which exists in DES	Proven inadequate
Possible Keys	2^{128}, 2^{192}, or 2^{256}	2^{112} or 2^{168}	2^{56}
Possible ASCII printable character keys	95^{16}, 95^{24} or 95^{52}	95^{14} or 95^{21}	95^{7}
Time Required to check all possible keys at 50 billion keys per second**	For a 128-bit key: $5x10^{21}$ years	For a 112-bit key: 800 days	For a 56-bit key: 400 days

RAMAIAH
Institute of Technology



RAMAIAH Institute of Technology

Unit III(Text1)

Advanced Encryption Standard(AES): (Chapter 7)

- Introduction
- Transformations
- Key Expansion
- The AES Ciphers
- Examples
- Analysis of AES.

Asymmetric Key Cryptography: (Chapter 10)

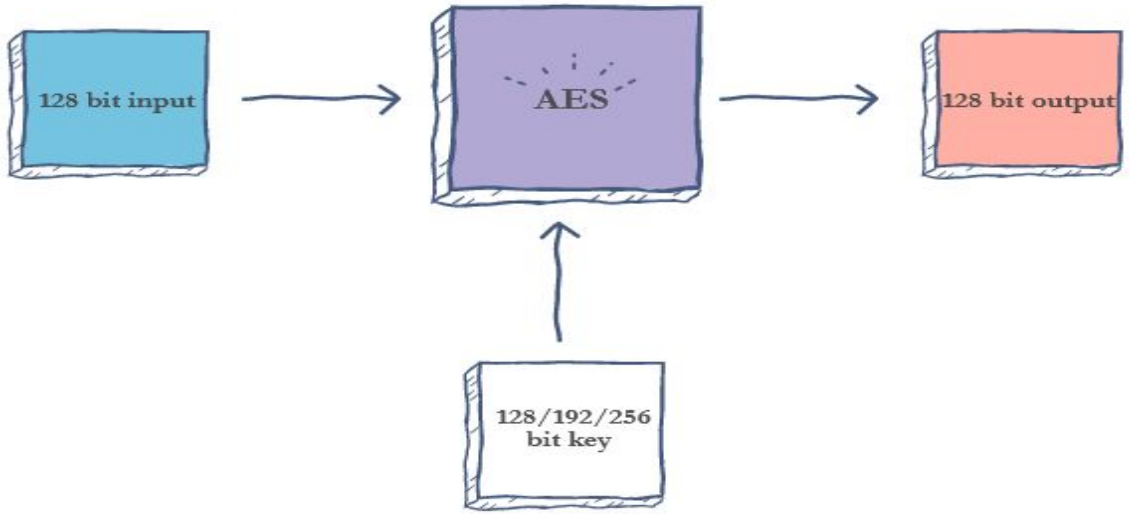
- Introduction
- RSA Cryptosystem
- Rabin Cryptosystem
- Elgamal Cryptosystem

RAMAIAH Institute of Technology

Introduction

Advanced Encryption Standard (AES) features are as follows:

- Symmetric key symmetric block cipher
- 128-bit block data, 128/192/256-bit keys
- Stronger and faster than Triple-DES



	DES	AES
Developed	1977	2000
Cipher Type	Symmetric block cipher	Symmetric block cipher
Block size	64 bits	128 bits
Key length	56 bits	128/192/256 bits
Security	Rendered insecure	Considered secure

Applications

- Data on storage media, including hard drives and USB drives.
- Electronic communication apps.
- Programming libraries.
- Internet browsers
- File and disk compression.
- Databases
- Login credentials including passwords

Introduction

- AES is an iterative rather than Feistel cipher.
- It is based on 'substitution–permutation network'.
- It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).
- AES performs all its computations on bytes rather than bits.

Introduction

The Advanced Encryption Standard (AES) published by the National Institute of Standards and Technology (NIST) in December 2001.

- History
- Criteria
- Rounds
- Data Units
- Structure of Each Round

Introduction

History

- In February 2001, NIST announced that a draft of the Federal Information Processing Standard (FIPS) was available for public review and comment.
- Finally, AES was published as FIPS 197 in the Federal Register in December 2001.

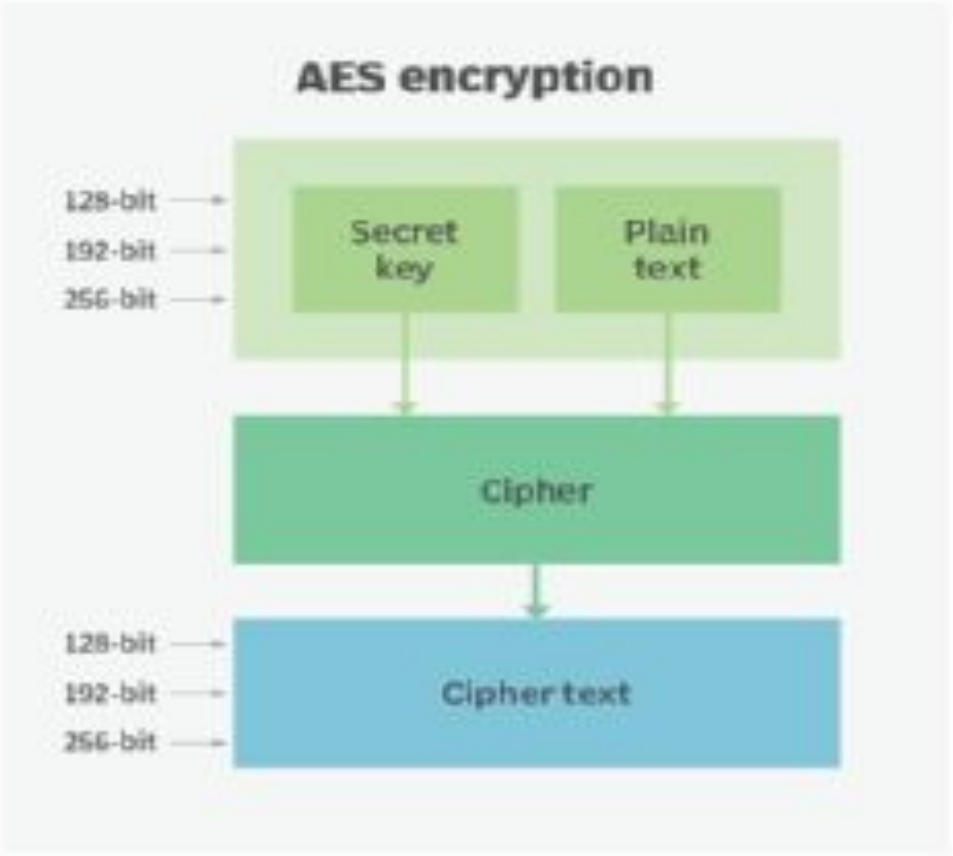
Introduction

Criteria

The criteria defined by NIST for selecting AES fall into three areas:

1. **Security** -128bit key
2. **Cost**-Computational efficiency and storage requirements
3. **Implementation** -Flexibility(platform independent)

DES encryption vs. AES encryption

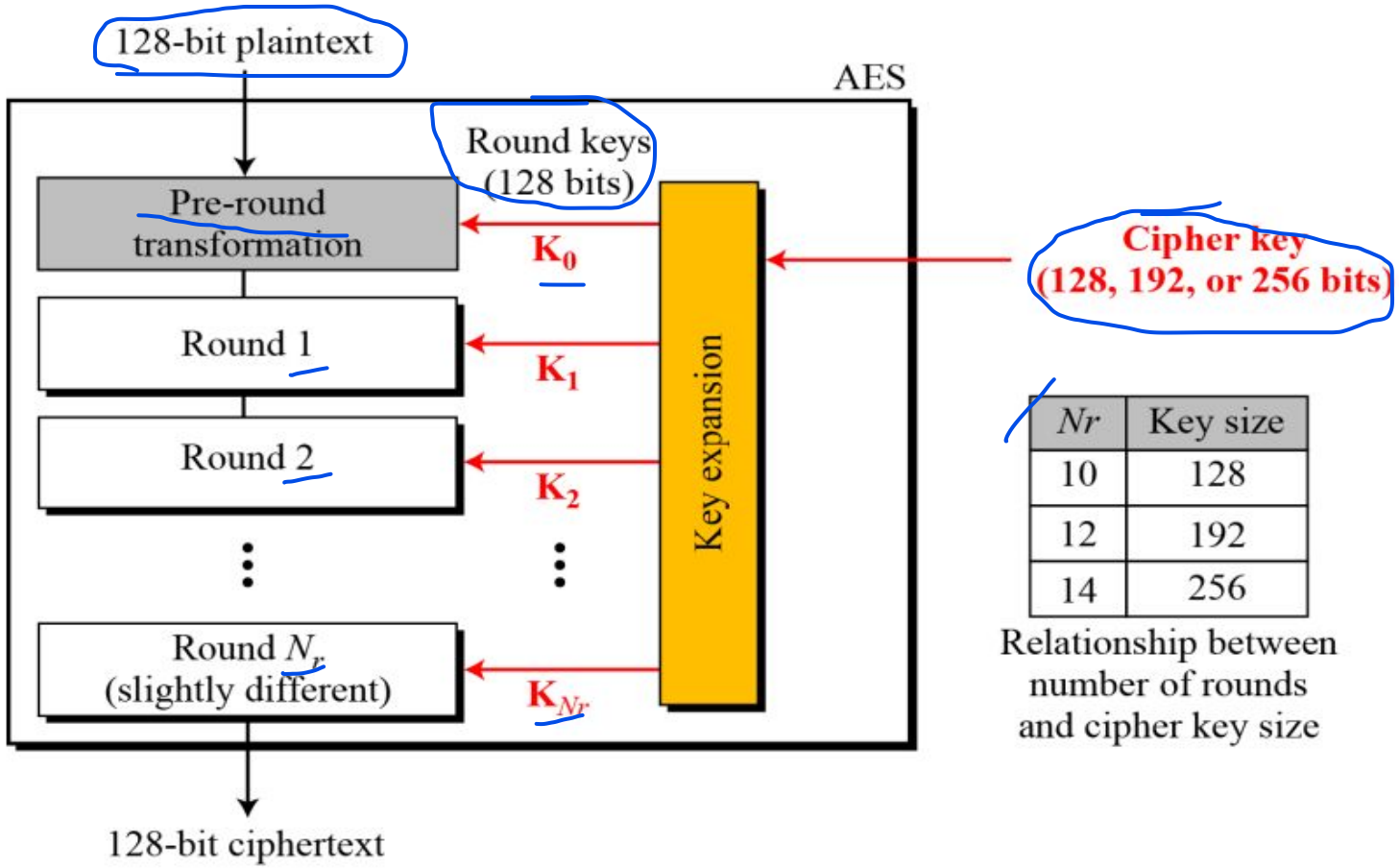


Introduction

Rounds

The number of round key is one more than the number of rounds

Number of Round keys = Nr+1



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Introduction

Rounds

- AES is a non Feistel cipher that encrypts and decrypts a block of 128bit data.
- It uses 10,12 or 14 rounds
- Cipher keysize-128/192/256bits

Introduction

Rounds

AES-128

AES-192

AES-256

1. AES-128 uses a 128-bit key length to encrypt and decrypt message blocks.
2. AES-192 uses a 192-bit key length to encrypt and decrypt message blocks.
3. AES-256 uses a 256-bit key length to encrypt and decrypt message blocks.

But roundkey is always 128bits

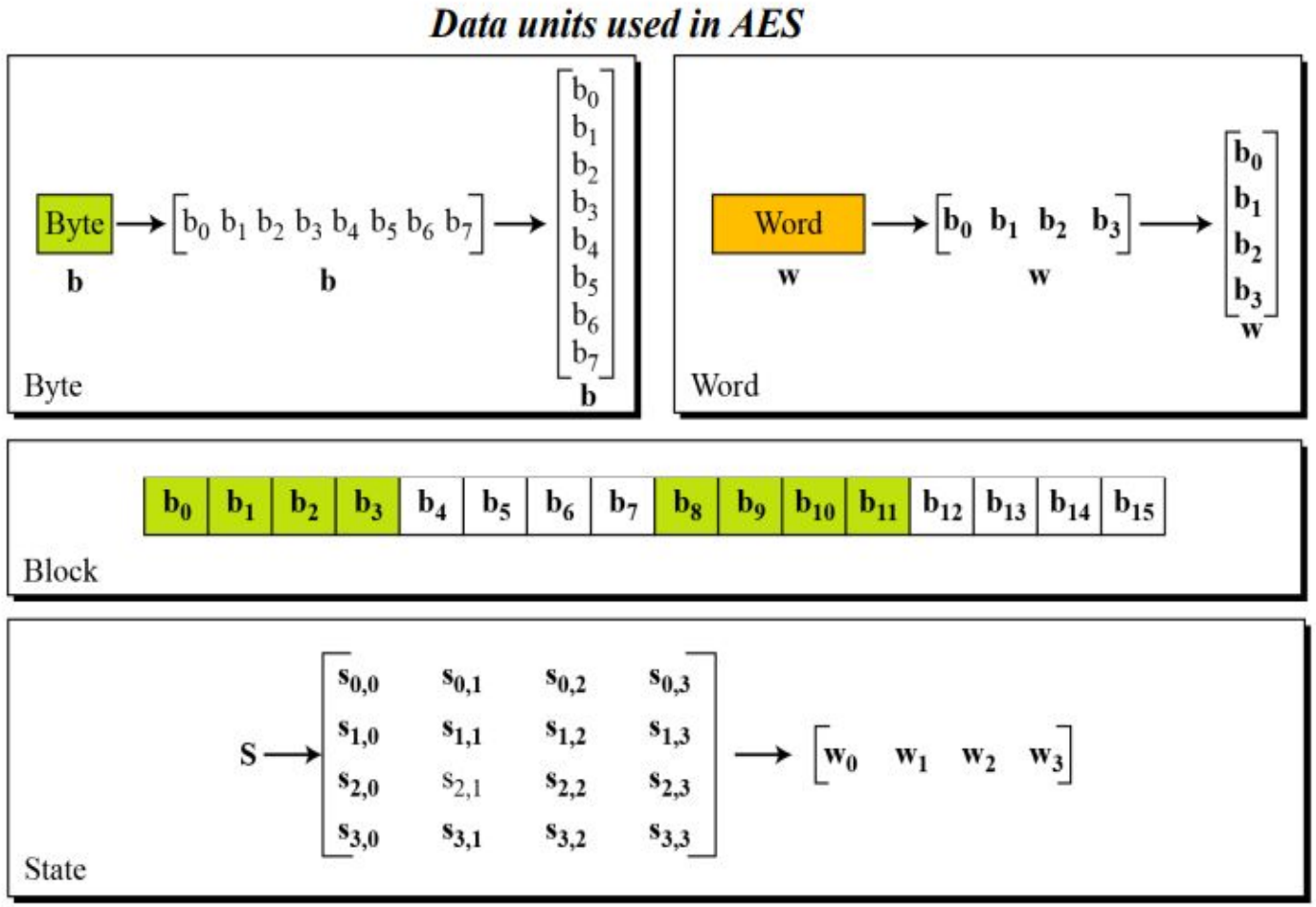
RAMAIAH Institute of Technology

Introduction

Data Units

AES uses 5 units of measurement to refer to data

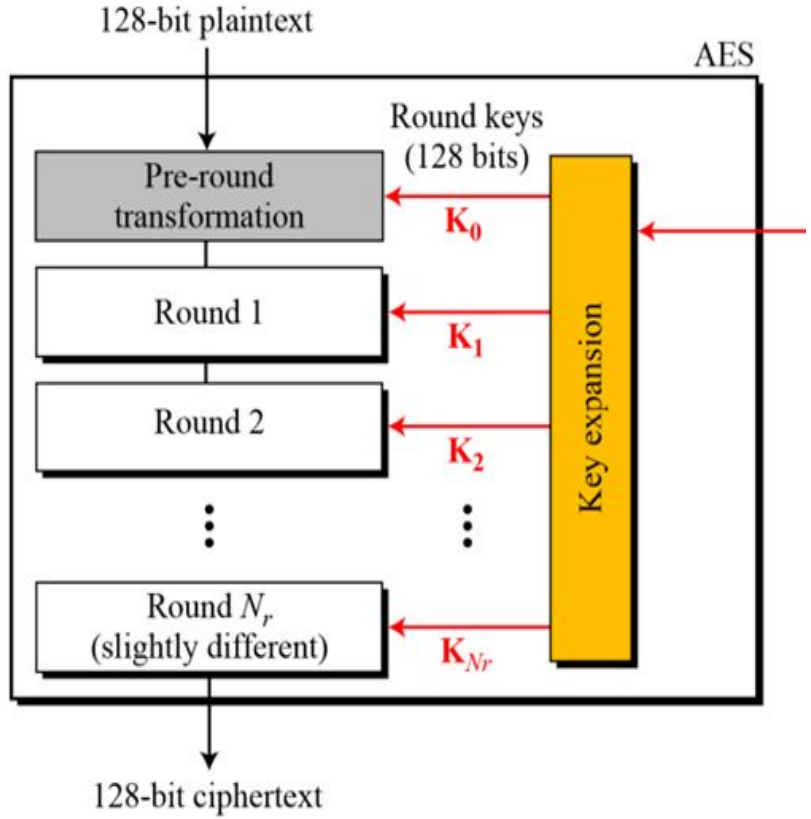
1. Bits
2. Bytes
3. Words
4. Blocks
5. State



Introduction

Data Units – States

- AES uses several rounds
- Each round is made of several stages – Data blocks are transferred from one stage to other
- At the beginning and end of cipher –data blocks
- Before and after each stage – data block is referred as state

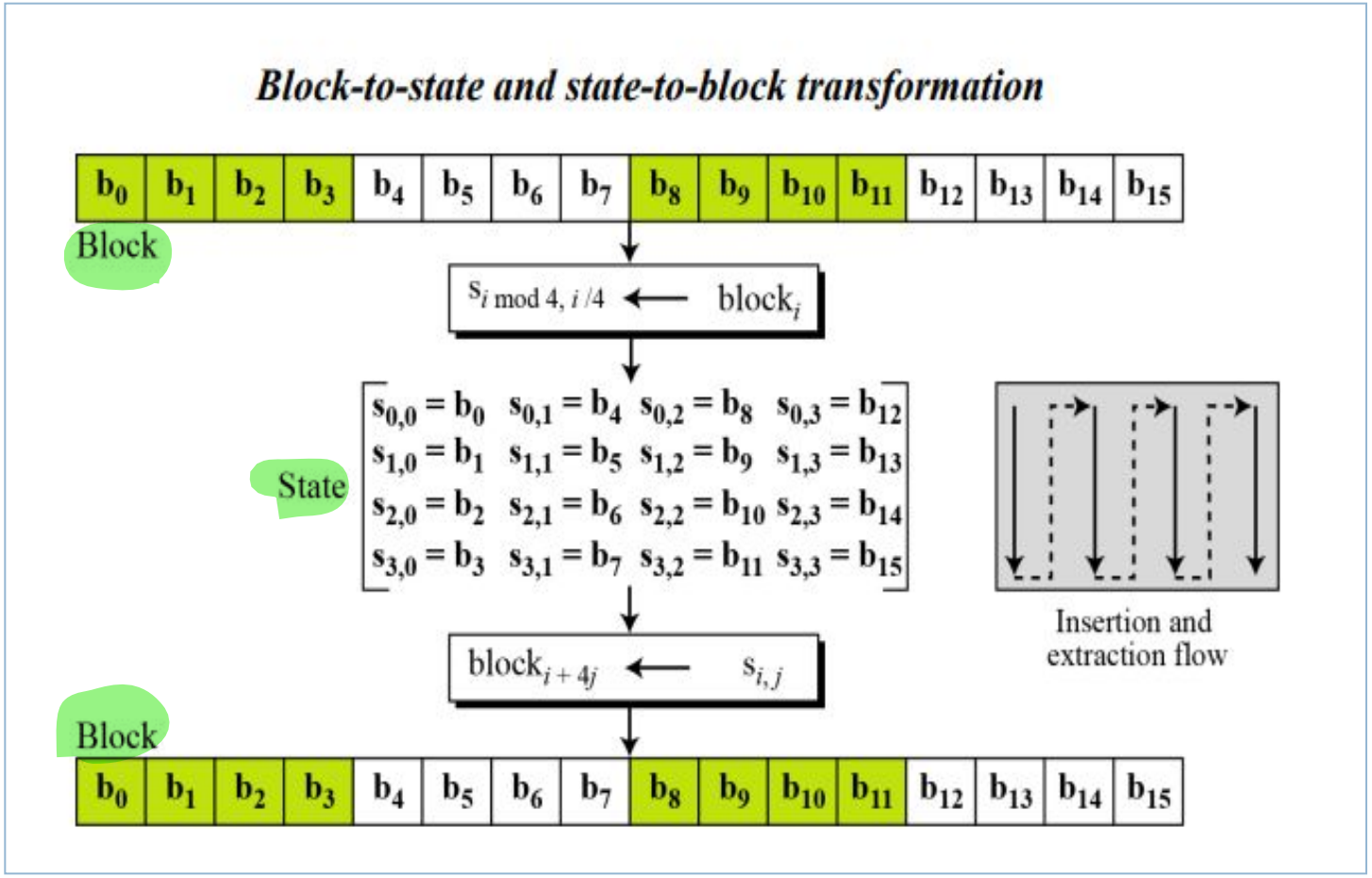


Introduction

Data Units – States

- S – State
- T- temporary state
- States are made up of 16bytes
- Matrix (4x4)

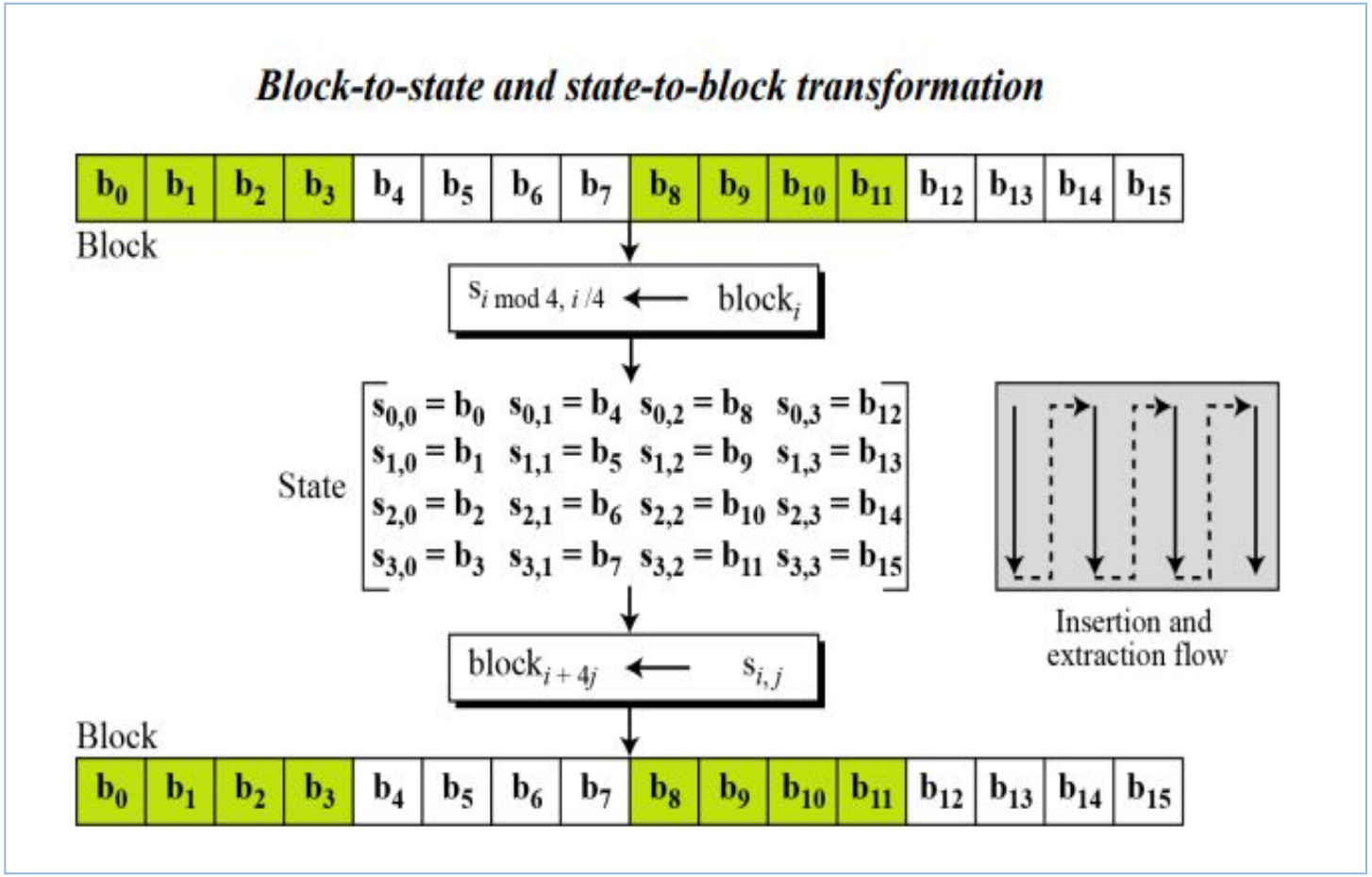
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



Introduction

Data Units – States

- Sate is treated as row matrix(1 x 4) of words



RAMAIAH Institute of Technology

Introduction

Example

Changing plaintext to state

Text	A	E	S	U	S	E	S	A	M	A	T	R	I	X	Z	Z
Hexadecimal	00	04	12	14	12	04	12	00	0C	00	13	11	08	23	19	19

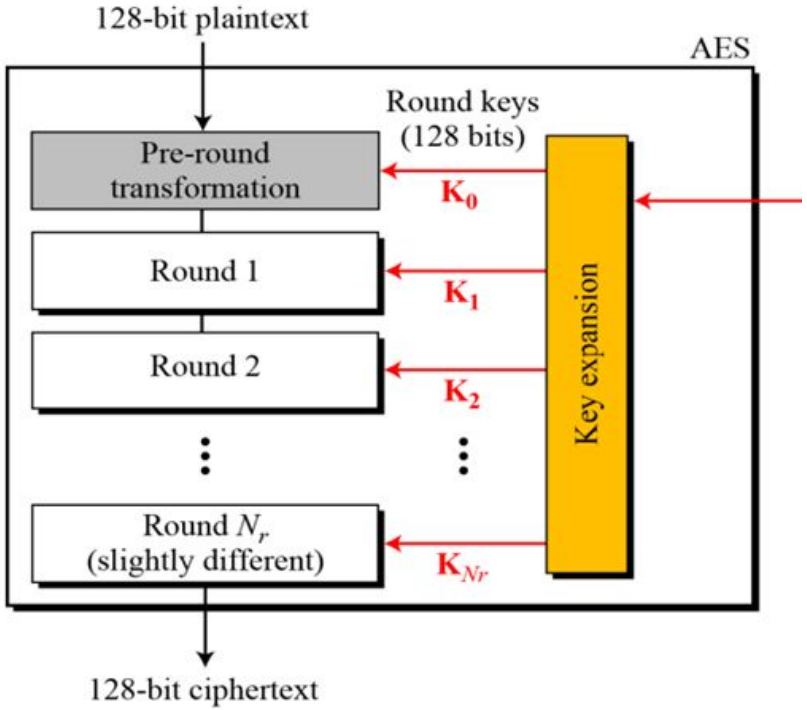
$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$$ State

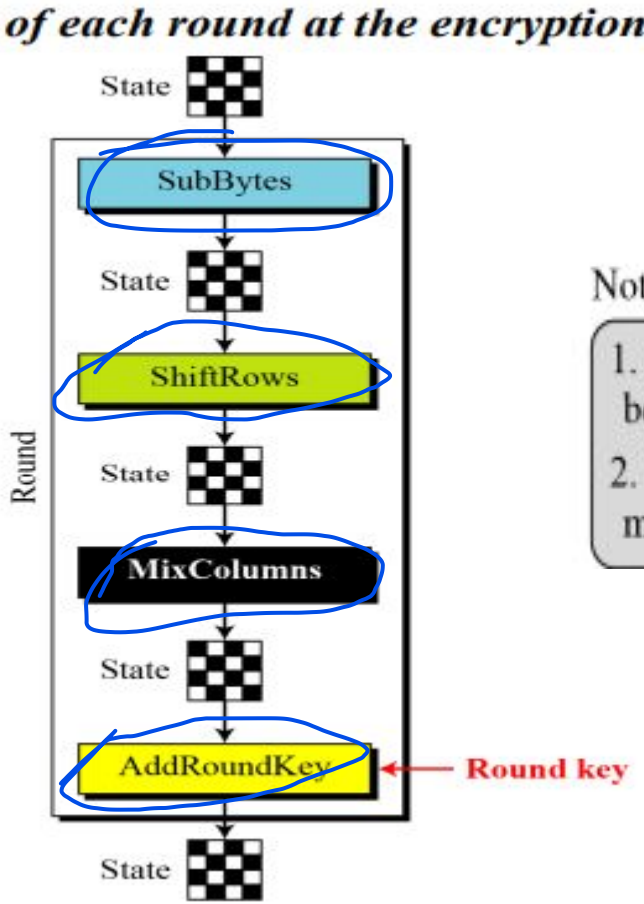
Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Decimal	Hex	Char
0	0	[NULL]
1	1	[START OF HEADING]
2	2	[START OF TEXT]
3	3	[END OF TEXT]
4	4	[END OF TRANSMISSION]
5	5	[ENQUIRY]
6	6	[ACKNOWLEDGE]
7	7	[BELL]
8	8	[BACKSPACE]
9	9	[HORIZONTAL TAB]
10	A	[LINE FEED]
11	B	[VERTICAL TAB]
12	C	[FORM FEED]
13	D	[CARRIAGE RETURN]
14	E	[SHIFT OUT]
15	F	[SHIFT IN]
16	10	[DATA LINK ESCAPE]
17	11	[DEVICE CONTROL 1]
18	12	[DEVICE CONTROL 2]
19	13	[DEVICE CONTROL 3]
20	14	[DEVICE CONTROL 4]
21	15	[NEGATIVE ACKNOWLEDGE]
22	16	[SYNCHRONOUS IDLE]
23	17	[ENG OF TRANS. BLOCK]
24	18	[CANCEL]
25	19	[END OF MEDIUM]
26	1A	[SUBSTITUTE]
27	1B	[ESCAPE]
28	1C	[FILE SEPARATOR]
29	1D	[GROUP SEPARATOR]
30	1E	[RECORD SEPARATOR]
31	1F	[UNIT SEPARATOR]

Introduction

Structure of Round





Notes:

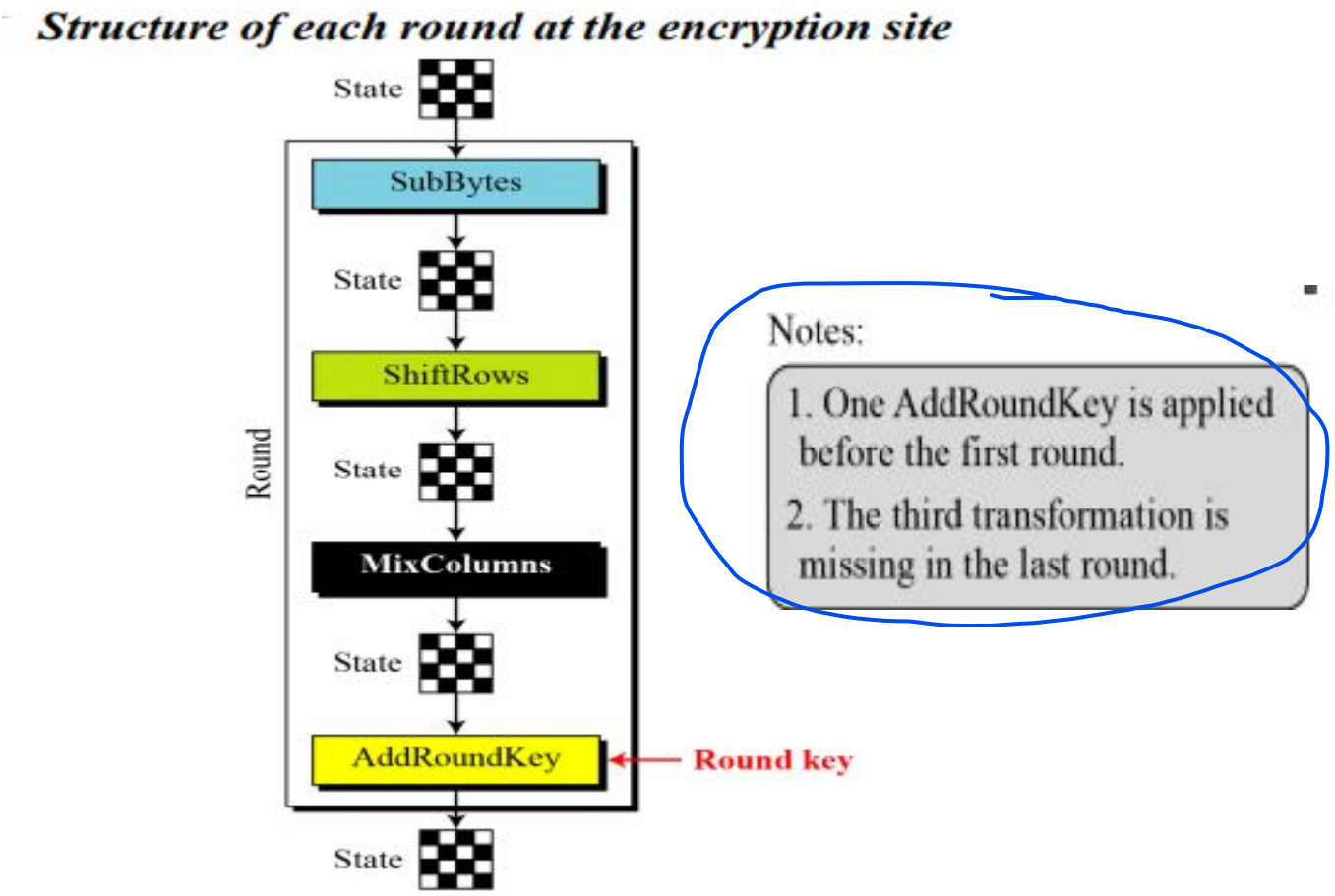
1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

Introduction

Structure of Round

4 Transformations

- SubBytes
- ShiftRows
- MixColumns
- Add Round Keys



Transformation

SubBytes(Substitution)

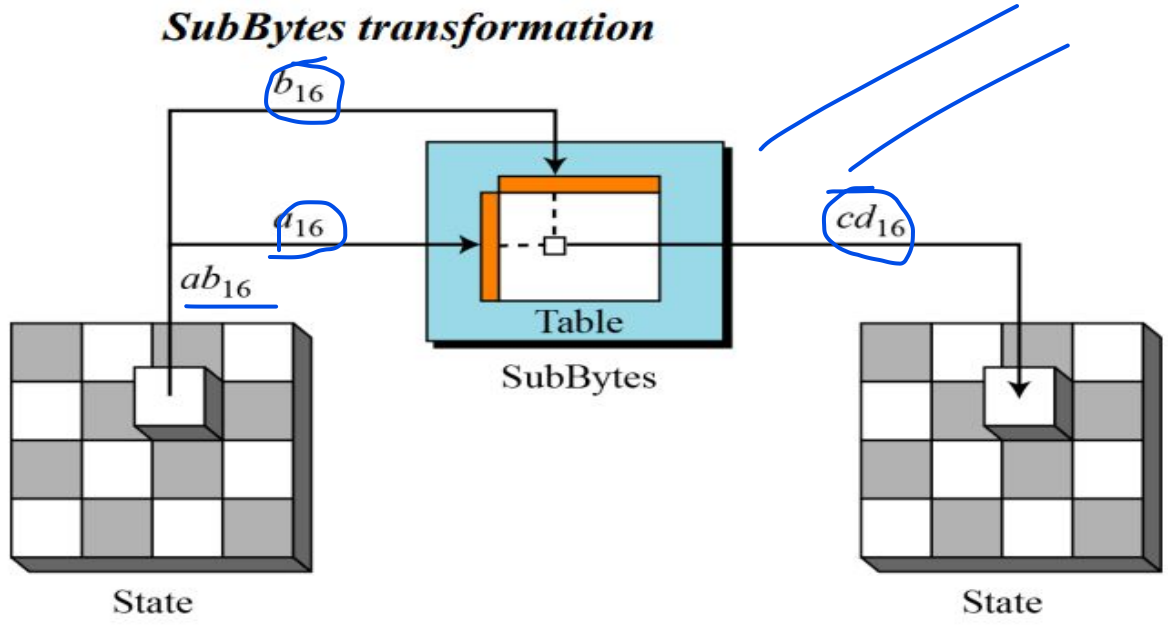
- AES uses substitution mechanism
  1. Substitution done for each byte
  2. Table is used for substitution for each byte
  3. Table Lookup process or mathematical calculation in Galois Field(GF-2^8 field)

RAMAIAH
Institute of Technology

Transformation

SubBytes

- SubBytes is used at the encryption site.
- To substitute a byte, we interpret the byte as two hexadecimal digits.
- Left digit –row
- Right digit -column



RAMAIAH Institute of Technology

Transformation

SubBytes- Transformation table

SubBytes transformation table

	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	CB	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

InvSubBytes-Transformation table

	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

RAMAIAH
Institute of Technology

Transformation

Example 1

$5A_{16} = BE_{16}$

$5B_{16} = 39_{16}$

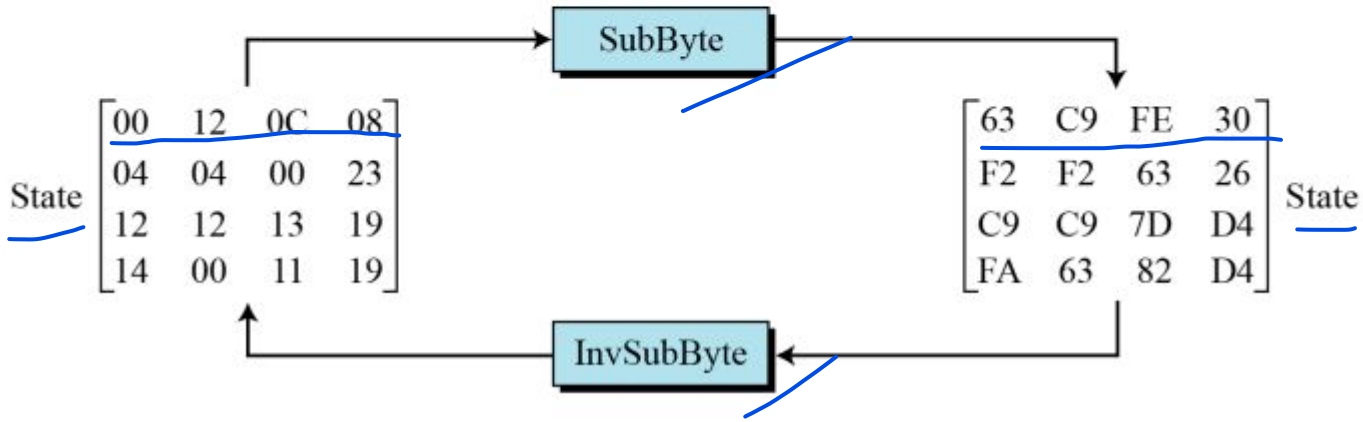
SubBytes transformation table

	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	CB	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

Transformation

Example 2

shows how a state is transformed using the SubBytes transformation. The figure also shows that the InvSubBytes transformation creates the original one. Note that if the two bytes have the same values, their transformation is also the same.



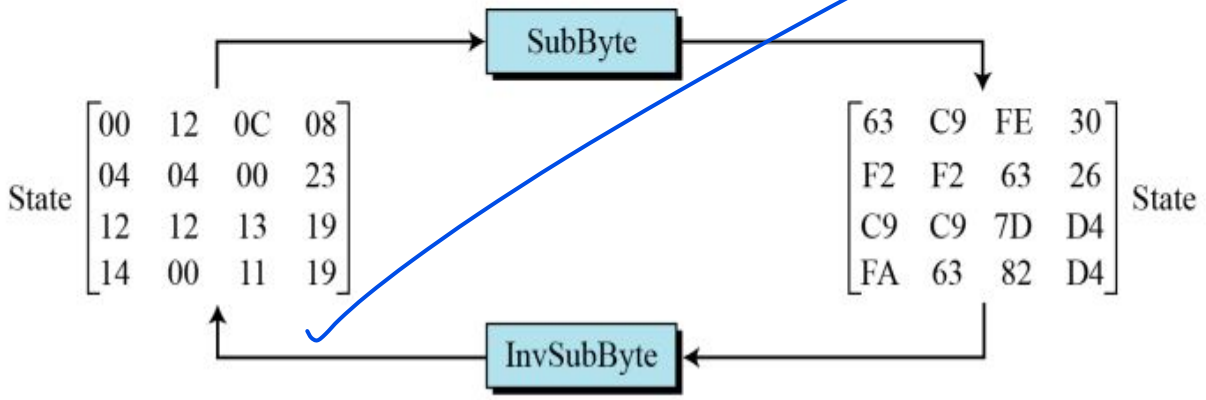
SubBytes transformation table

	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	CB	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

Transformation

Example 2

shows how a state is transformed using the SubBytes transformation. The figure also shows that the InvSubBytes transformation creates the original one. Note that if the two bytes have the same values, their transformation is also the same.



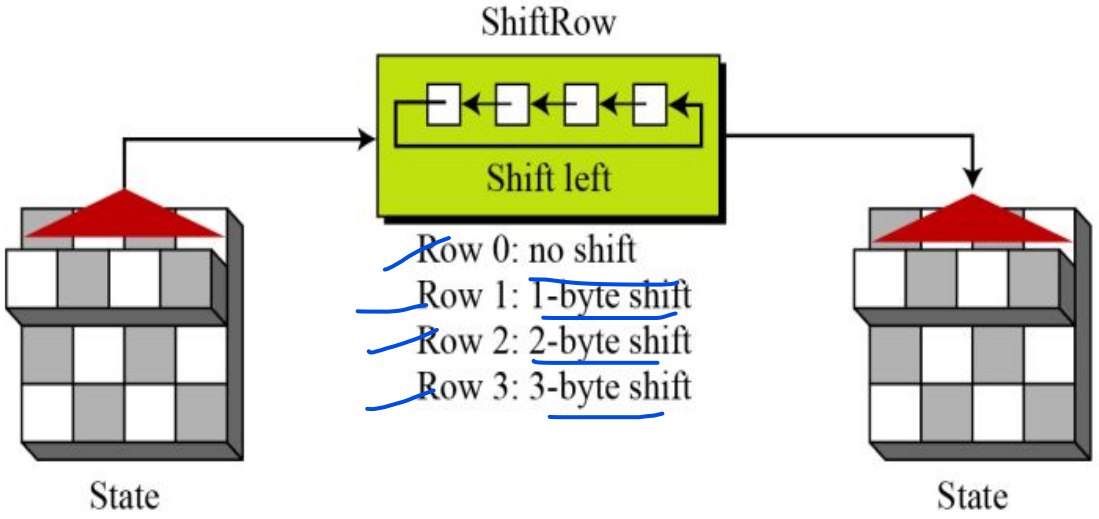
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	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

RAMAIAH
Institute of Technology

Transformation

ShiftRows (Permutation)

- Another transformation found in a round is shifting, which permutes the bytes.
- Each row is shifted a particular number of times.

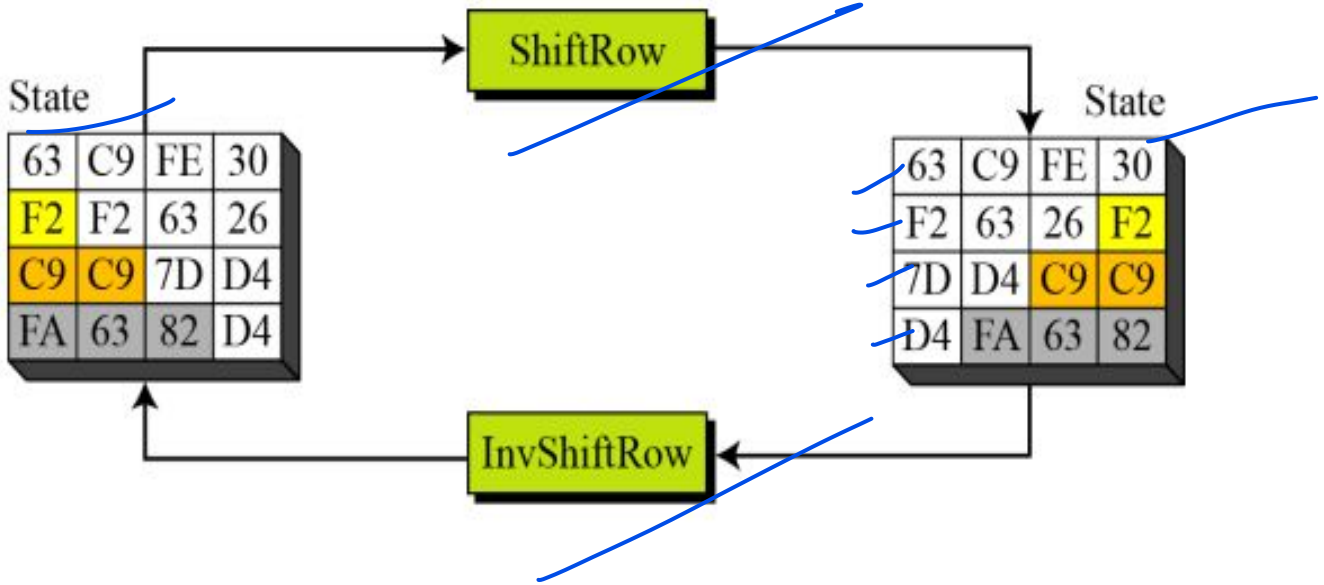


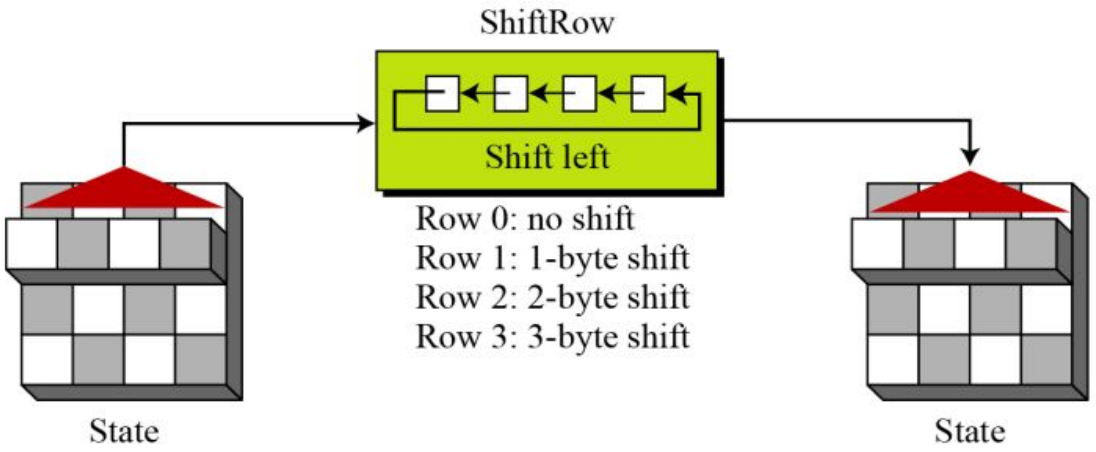
[b0	b1	b2	b3]		[b0	b1	b2	b3]
| b4	b5	b6	b7 |	->	| b5	b6	b7	b4 |
| b8	b9	b10	b11 |		| b10	b11	b8	b9 |
[b12	b13	b14	b15]		[b15	b12	b13	b14]

Transformation

ShiftRows (Permutation)

shows how a state is transformed using ShiftRows transformation. The figure also shows that InvShiftRows transformation creates the original state.

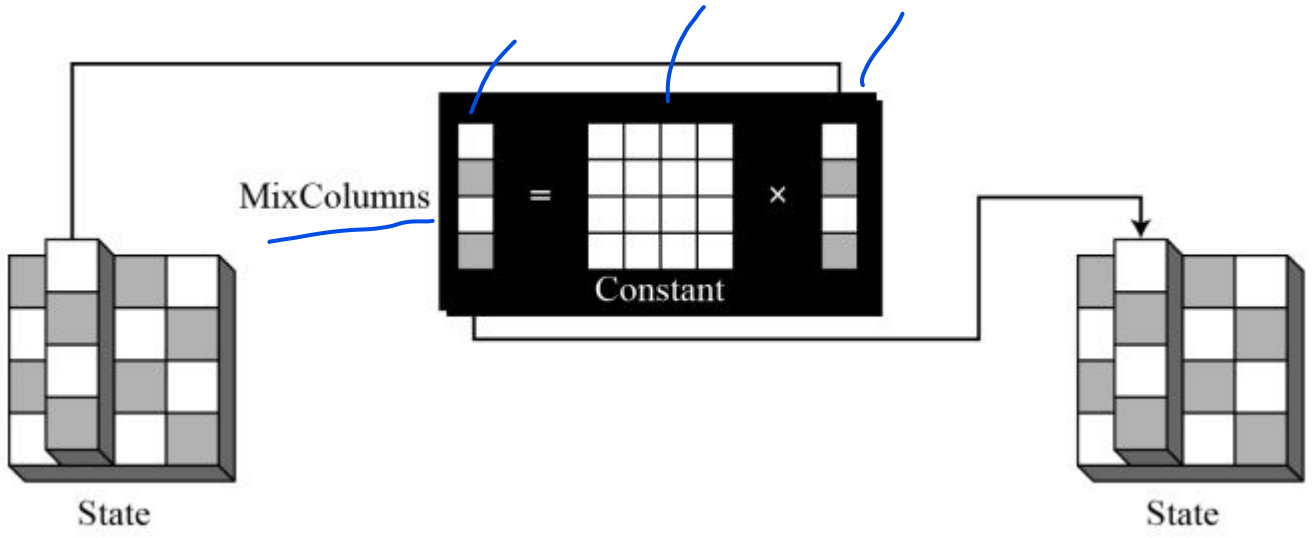




Transformation

MixColumns (Mixing)

- The MixColumns transformation operates at the column level; it transforms each column of the state to a new column.



Transformation

MixColumns (Mixing)

- This step is basically a matrix multiplication.
- Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

Mixing bytes using matrix multiplication

$$\begin{bmatrix} ax + by + cz + dt \\ ex + fy + gz + ht \\ ix + jy + kz + lt \\ mx + ny + oz + pt \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}$$

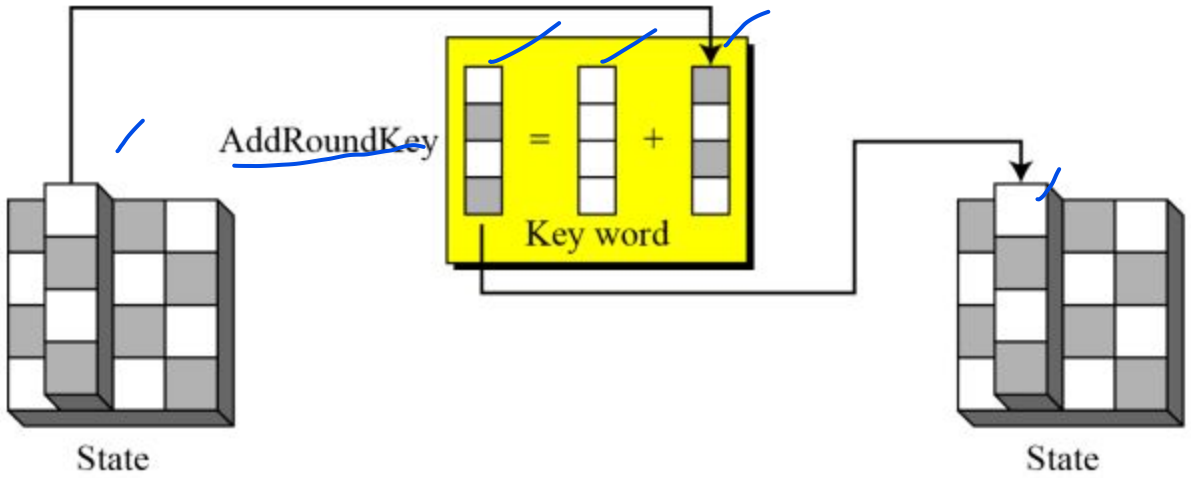
New matrix | Constant matrix | Old matrix

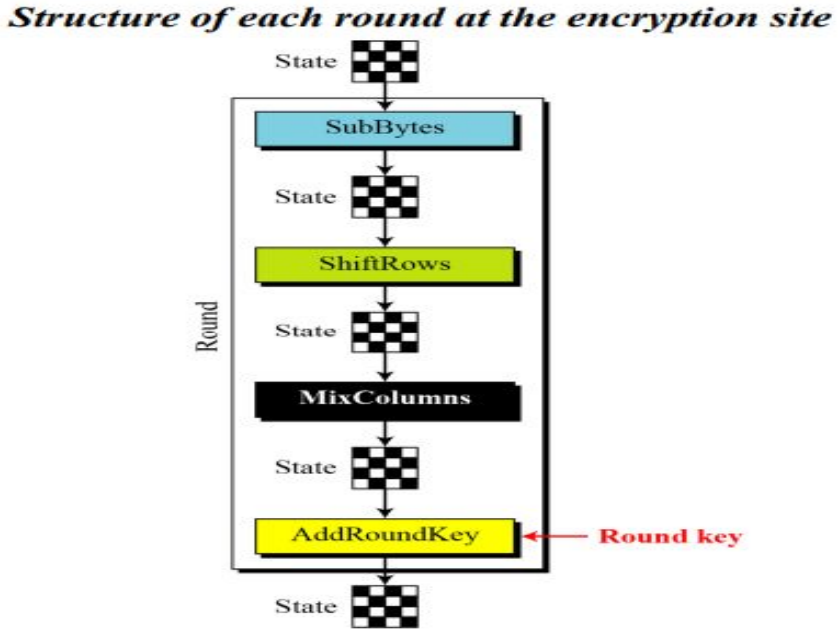
RAMAIAH
Institute of Technology

Transformation

Add Round Keys (Adding)

- AddRoundKey proceeds one column at a time.
- AddRoundKey adds a round key word with each state column matrix; the operation in AddRoundKey is matrix addition.





RAMAIAH
Institute of Technology

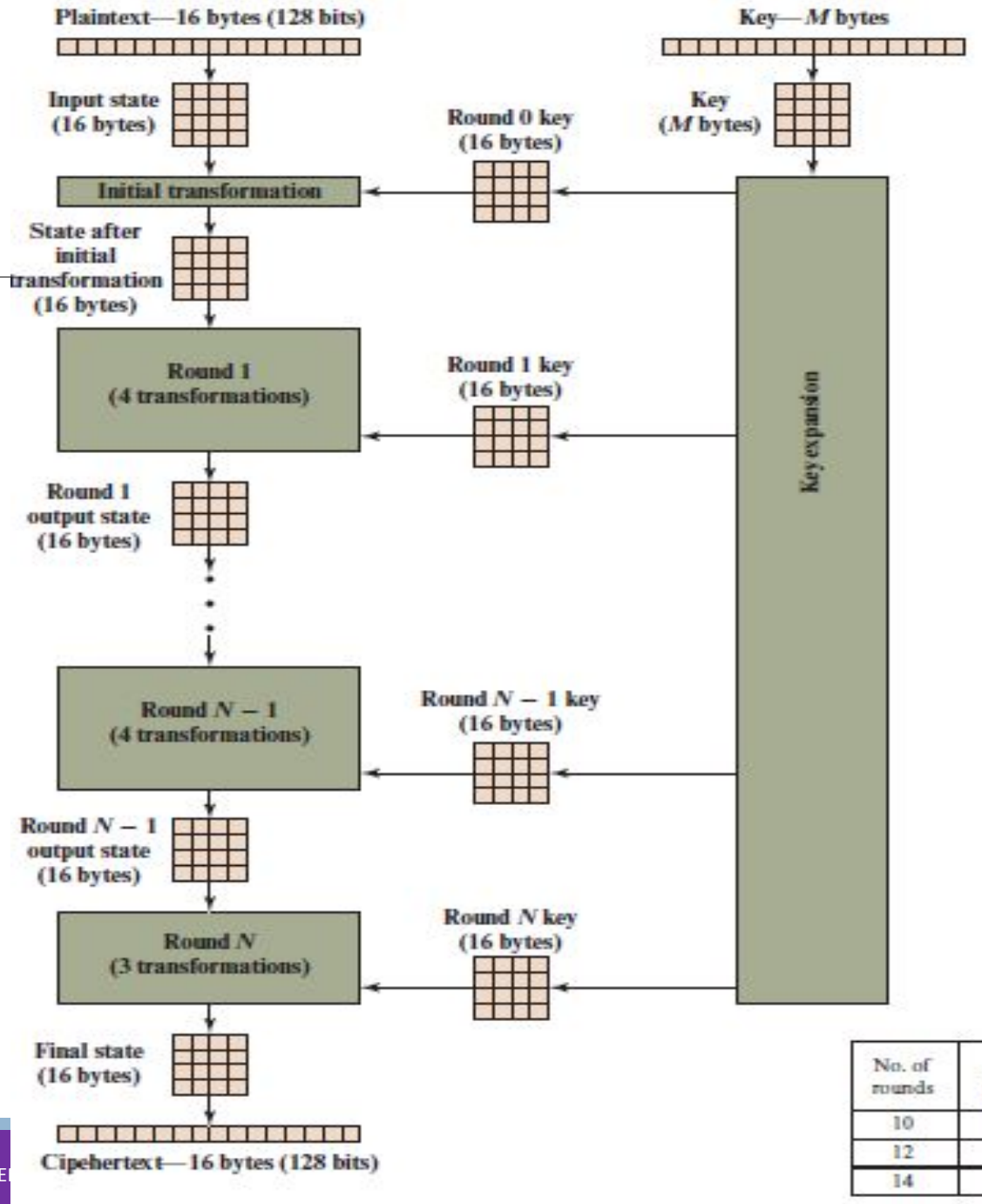
AES Structure

- Figure shows the overall structure of the AES encryption process.

The cipher takes a plaintext block size of 128 bits, or 16 bytes.

The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits).

The algorithm is referred to as AES-128, AES-192, orAES-256, depending on the key length.



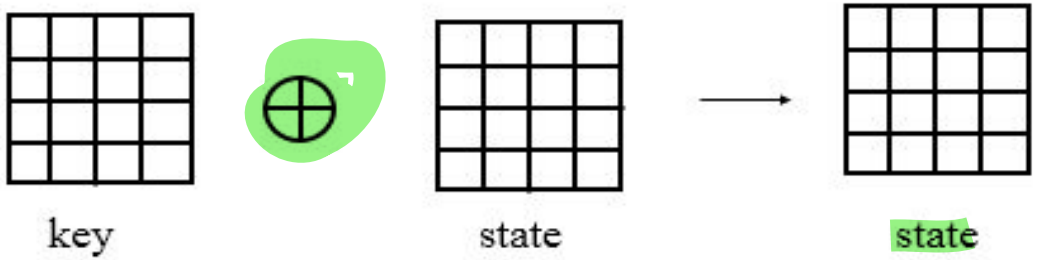
No. of rounds	Key Length (bytes)
10	16
12	24
14	32

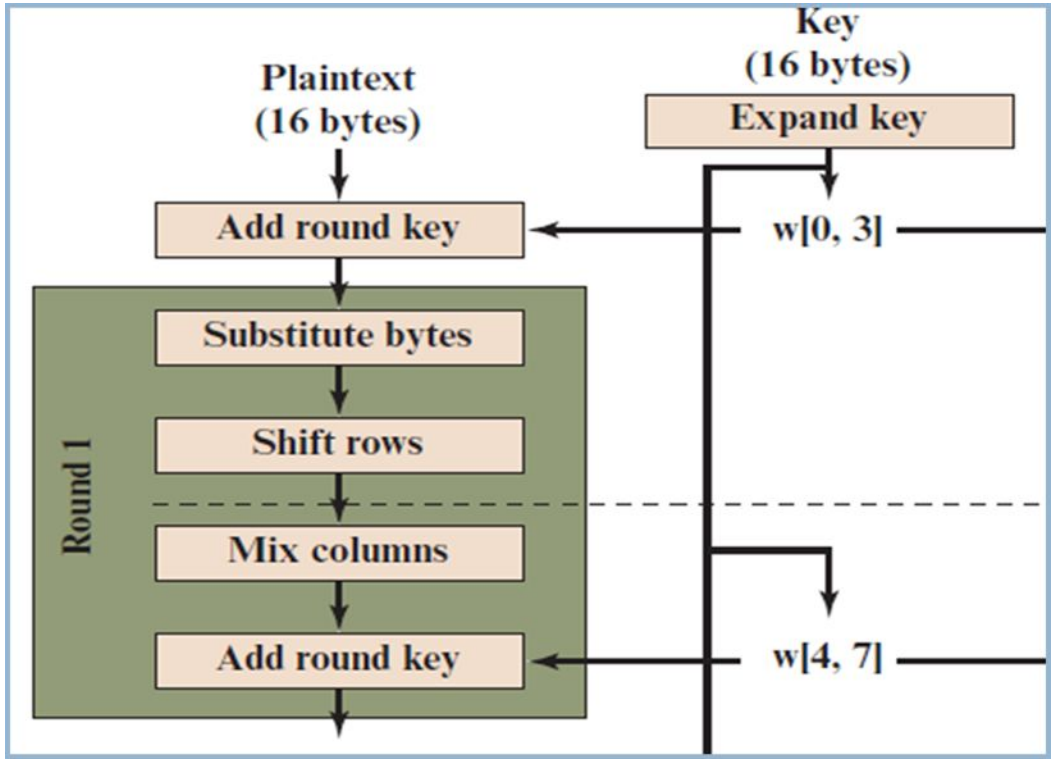
Pre-round transformation

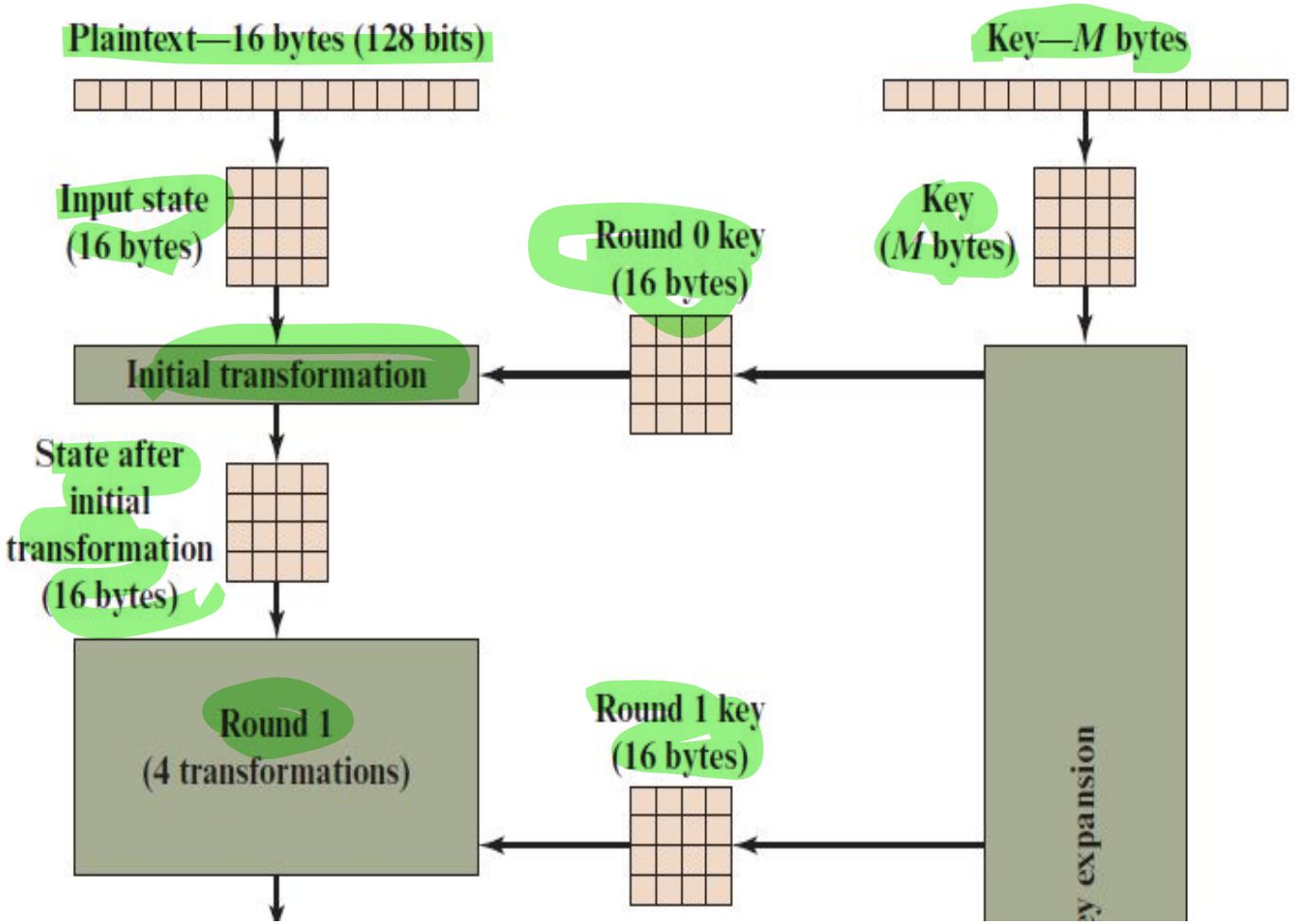
State is represented as follows (16 bytes):

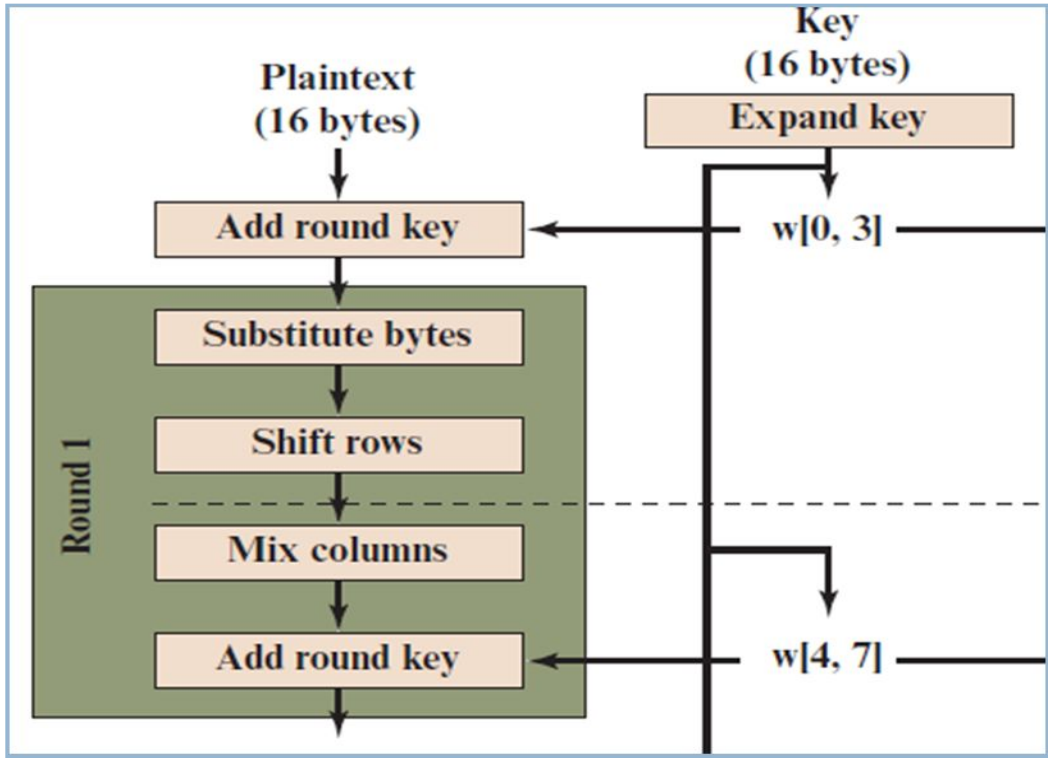
$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

AddRoundKey(State, Key):






Plaintext—16 bytes (128 bits)
Key—M bytes
Input state
(16 bytes)
Key
(M bytes)
Round 0 key
(16 bytes)
Initial transformation
State after
initial
transformation
(16 bytes)
Round 1
(4 transformations)
Round 1 key
(16 bytes)
y expansion


Plaintext
(16 bytes)
Key
(16 bytes)
Expand key
Add round key
w[0, 3]
Substitute bytes
Shift rows
Round 1
Mix columns
Add round key
w[4, 7]

AES Structure

Four different stages are used, one of permutation and three of substitution:

Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block.

ShiftRows: A simple permutation.

MixColumns: A substitution that makes use of arithmetic over GF(28).

AddRoundKey: A simple bitwise XOR of the current block with a portion of the expanded key

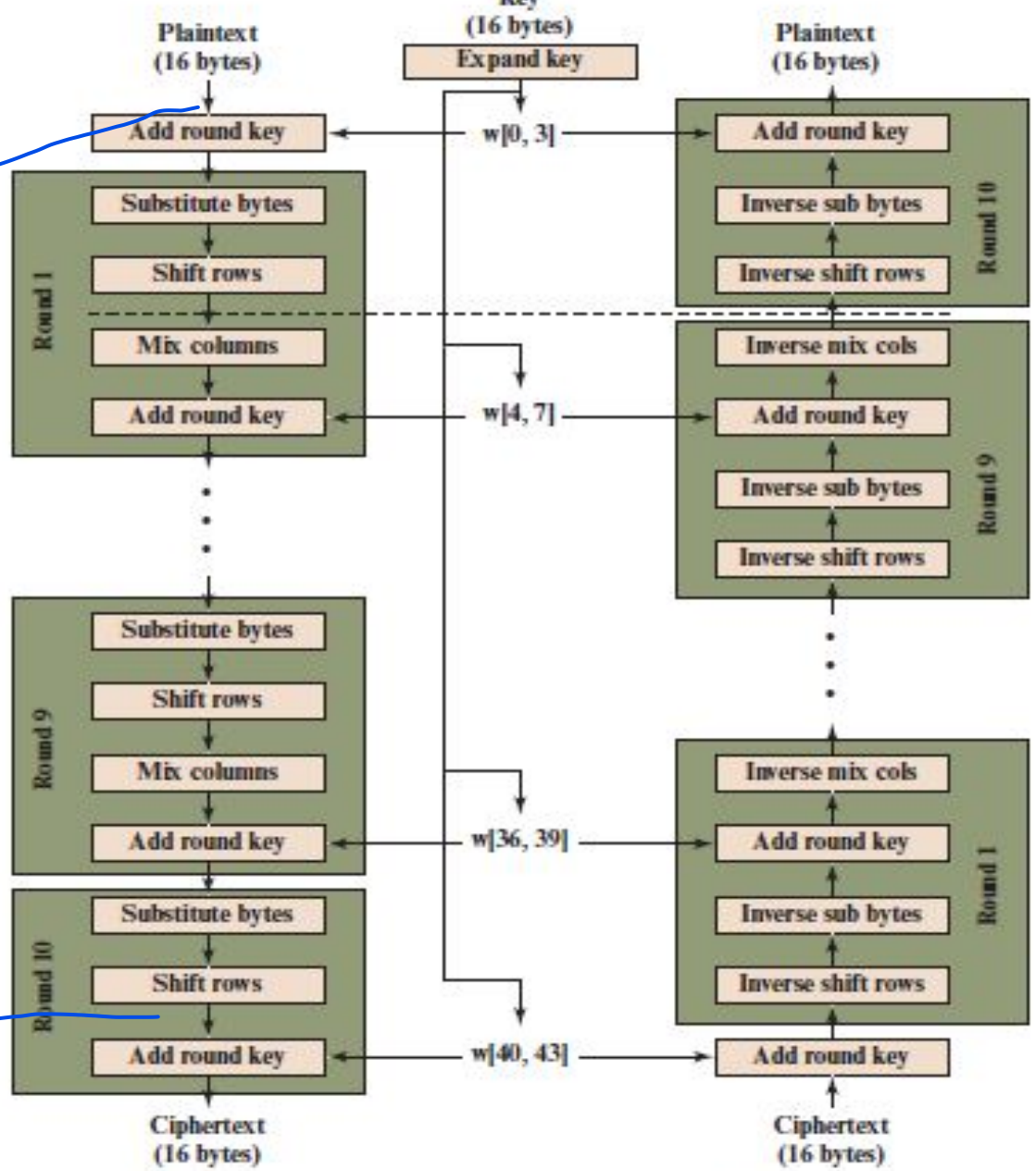


Figure 6.3 AES Encryption and Decryption

	AES-128	AES-192	AES-256
Key Size	128	192	256
Plaintext Size	128	128	128
Number of rounds	10	12	14
Round Key Size	128	128	128

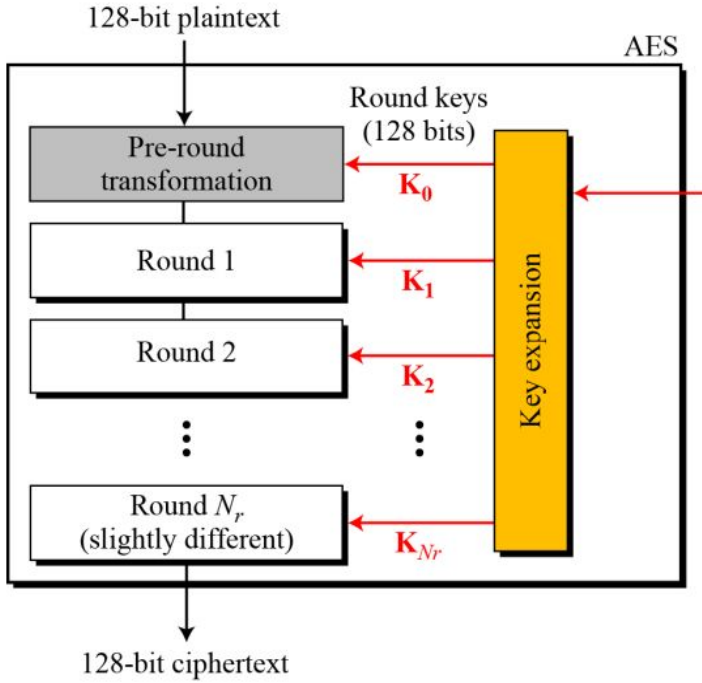
Unit II (Text1)

Advanced Encryption Standard(AES): (Chapter 7)

- Introduction
- Transformations
- Key Expansion
- The AES Ciphers
- Examples
- Analysis of AES.

Key Expansion

- To create round keys for each round, AES uses a key-expansion process.
- If the number of rounds is Nr , the key-expansion routine creates Nr + 1 128-bit round keys from one single 128-bit cipher key

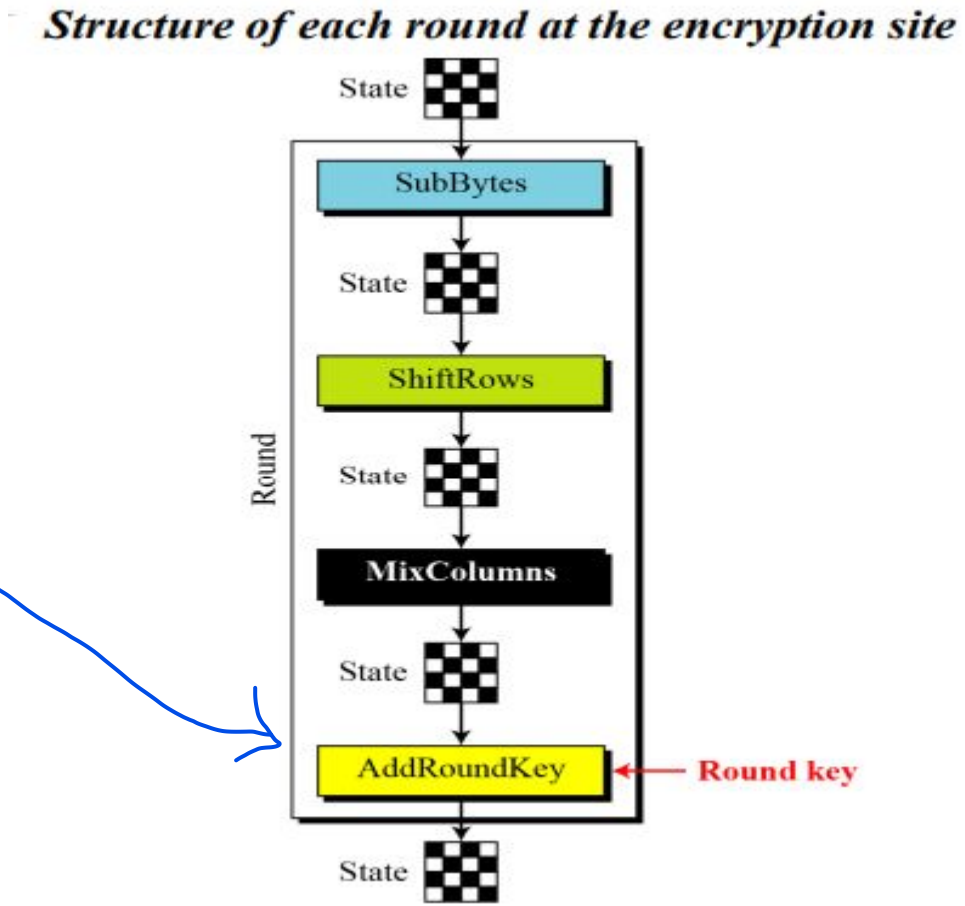


Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Key Expansion

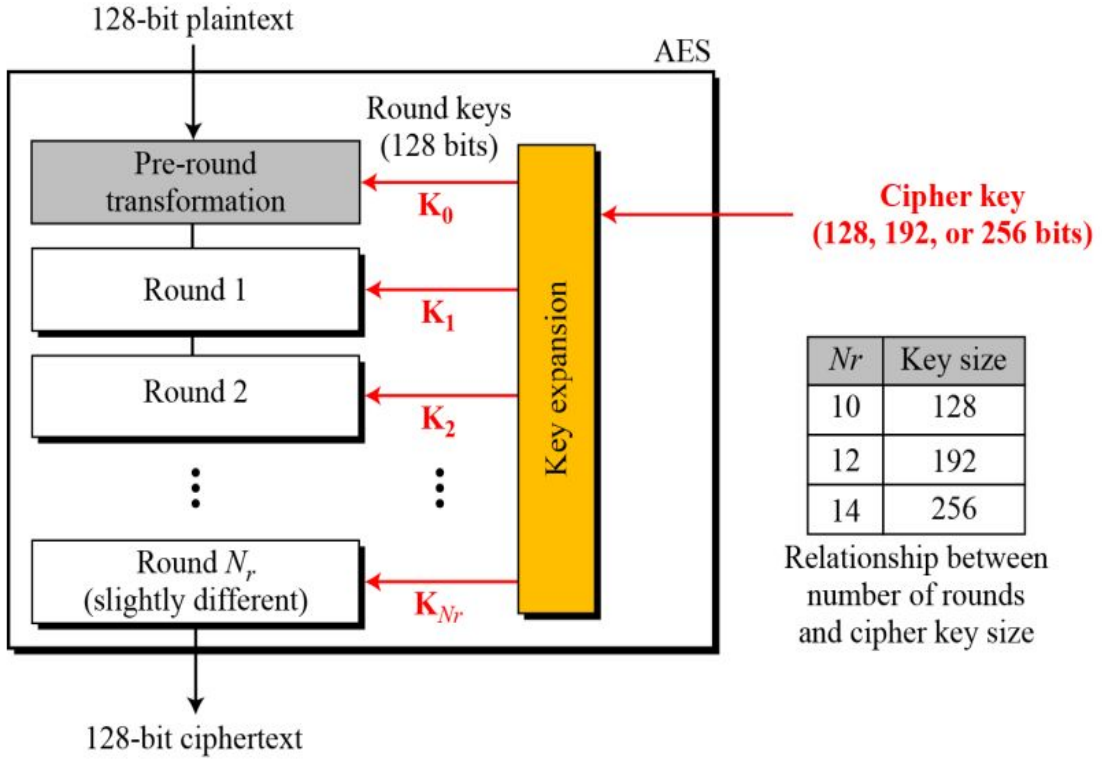
- First round key is used for pre-round transformation
- Remaining all for every round 4th transformation



Key Expansion

Key-expansion creates round key word by word, where a word is an array of 4 bytes.

- Key Expansion in AES-128
- Key Expansion in AES-192 and AES-256
- Key-Expansion Analysis



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Words for each round

Round	*Words*			
Pre-round	w_0	w_1	w_2	w_3
1	w_4	w_5	w_6	w_7
2	w_8	w_9	w_{10}	w_{11}
. . .	. . .			
N_r	w_{4N_r}	w_{4N_r+1}	w_{4N_r+2}	w_{4N_r+3}

RAMAIAH
Institute of Technology

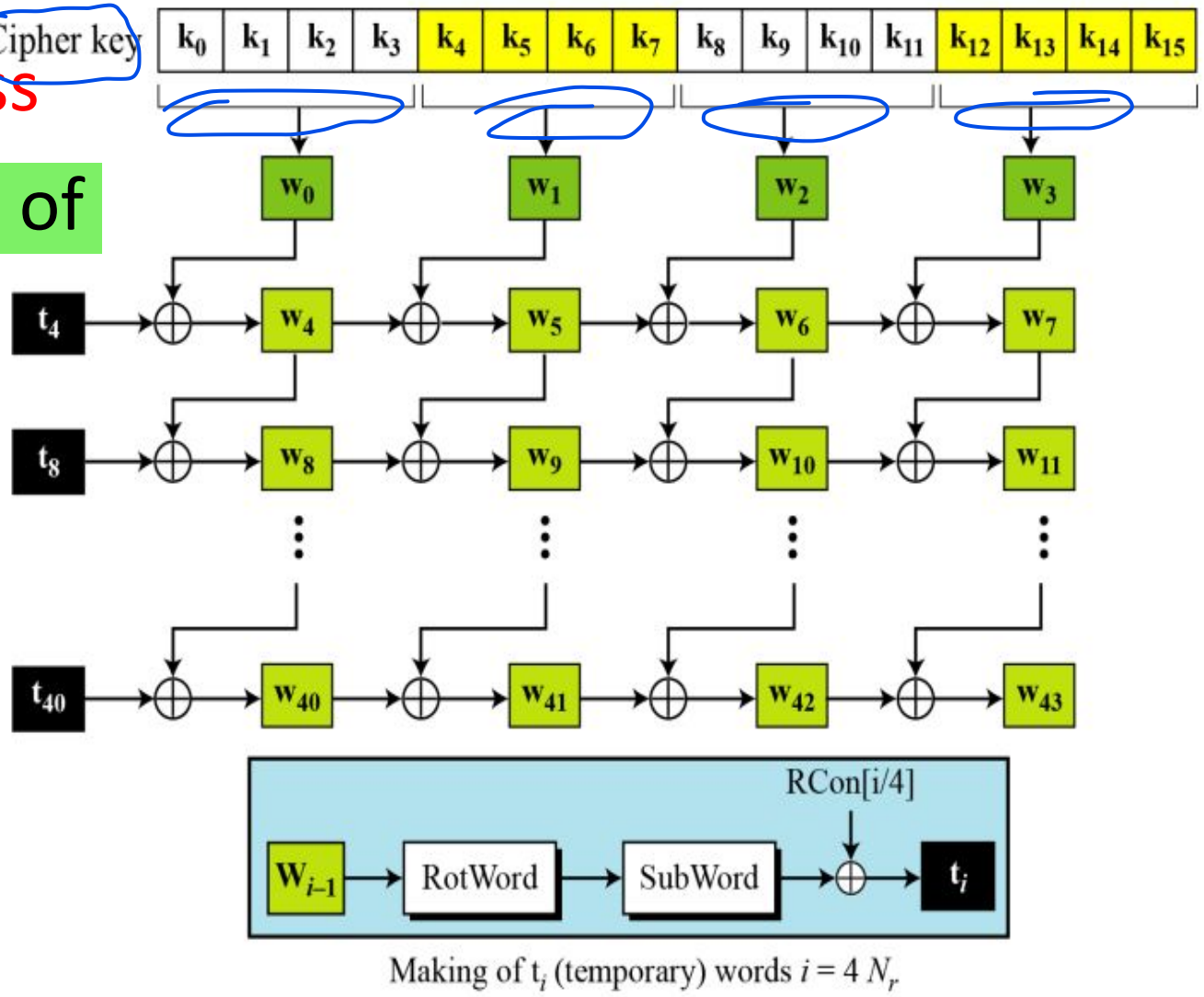
Key Expansion

Key Expansion in AES-128 process

1. Cipher key is an array of 16bytes(k0 to k15)

   The first 4 words(w0,w1,w2,w3) are made from cipher key

   - K0 to k3 -> w0
   - k4 to k7 -> w1
   - K8 to k11 ->w2
   - K12 to k15 ->w3



Making of t_i (temporary) words $i = 4\ N_r$

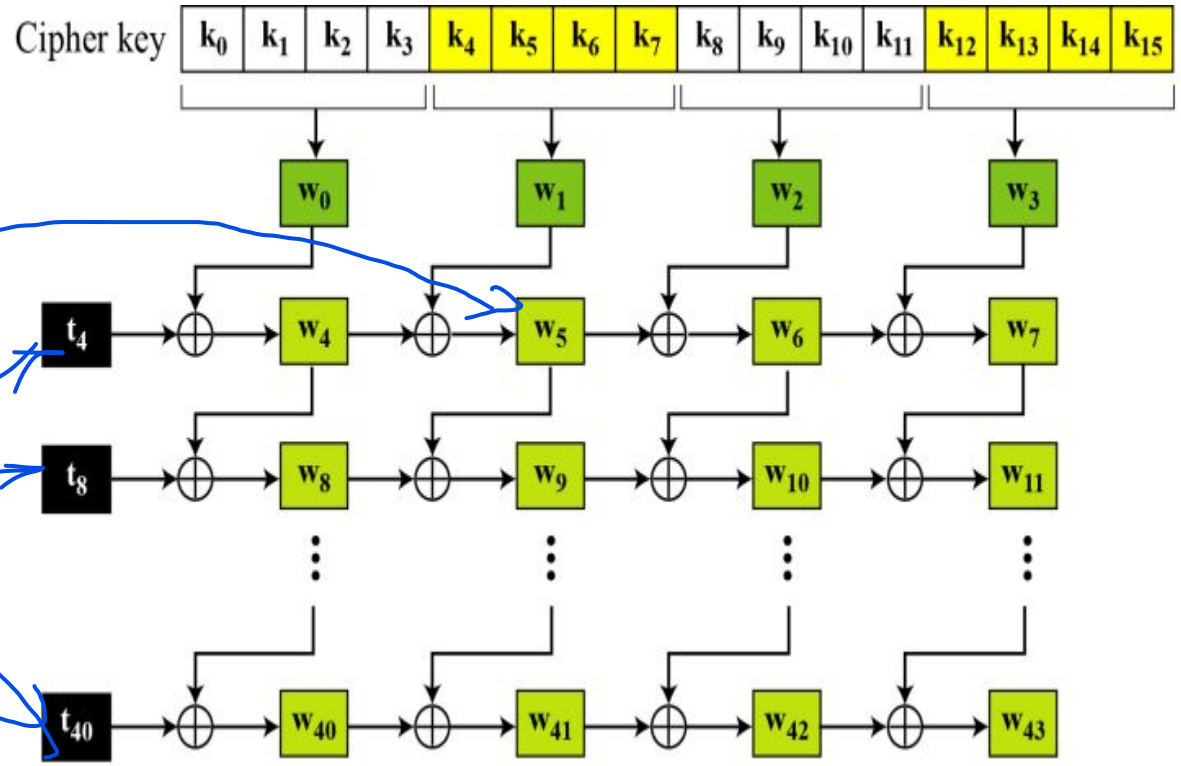
Key Expansion

Key Expansion in AES-128 process

2. The rest of the words (w_i for i =4 to 43) are made as follows

i) if $(i \bmod 4) \neq 0$, $w_i = w_{i-1} \oplus w_{i-4}$

ii) if $(i \bmod 4) = 0$, $w_i = t \oplus w_{i-4}$



Temporary word t

$t_i = subword(Rotword(w_{i-1})) \oplus Rcon_{i/4}$

Key Expansion

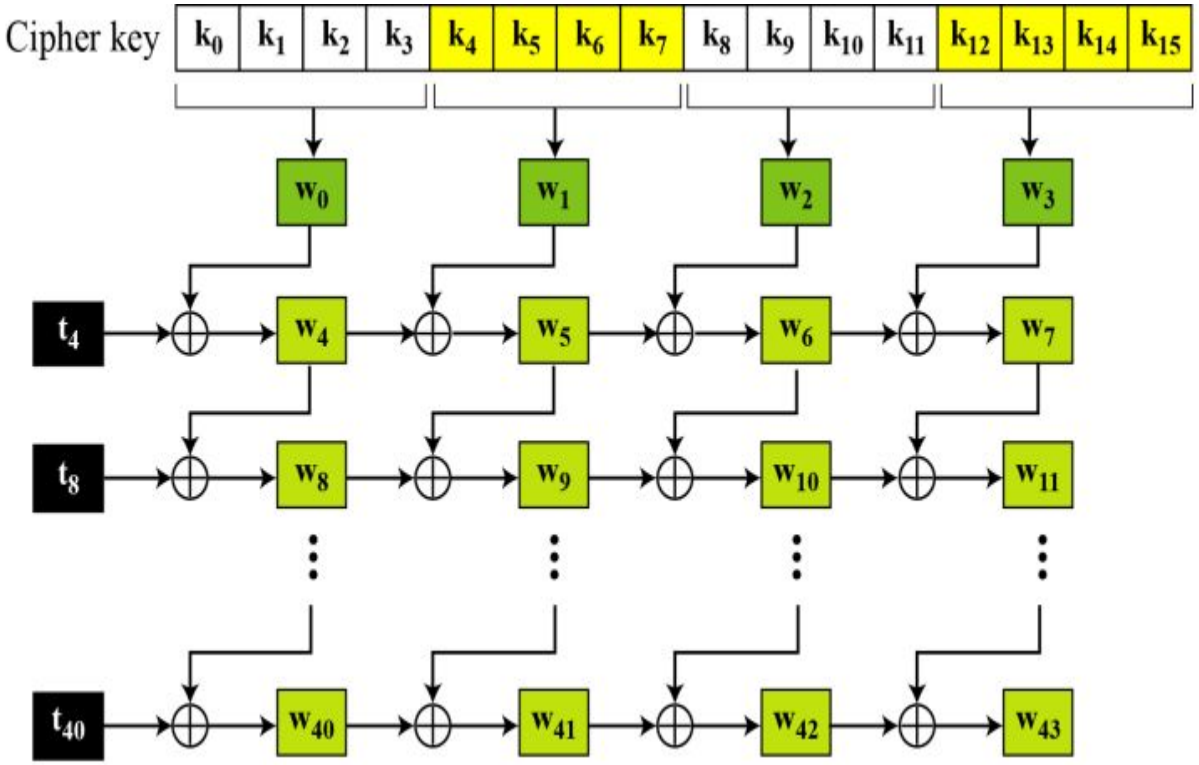
$t_i = subword(Rotword(w_{i-1})) \oplus Rcon_{i/4}$

Rotword : Applied to only one row

Rotate word routine takes a word as an array of 4bytes and shifts each byte to the left with wrapping.

Subword : Applied to 4 bytes.

Substitute word routine takes each byte in the word and substitute another byte for it.



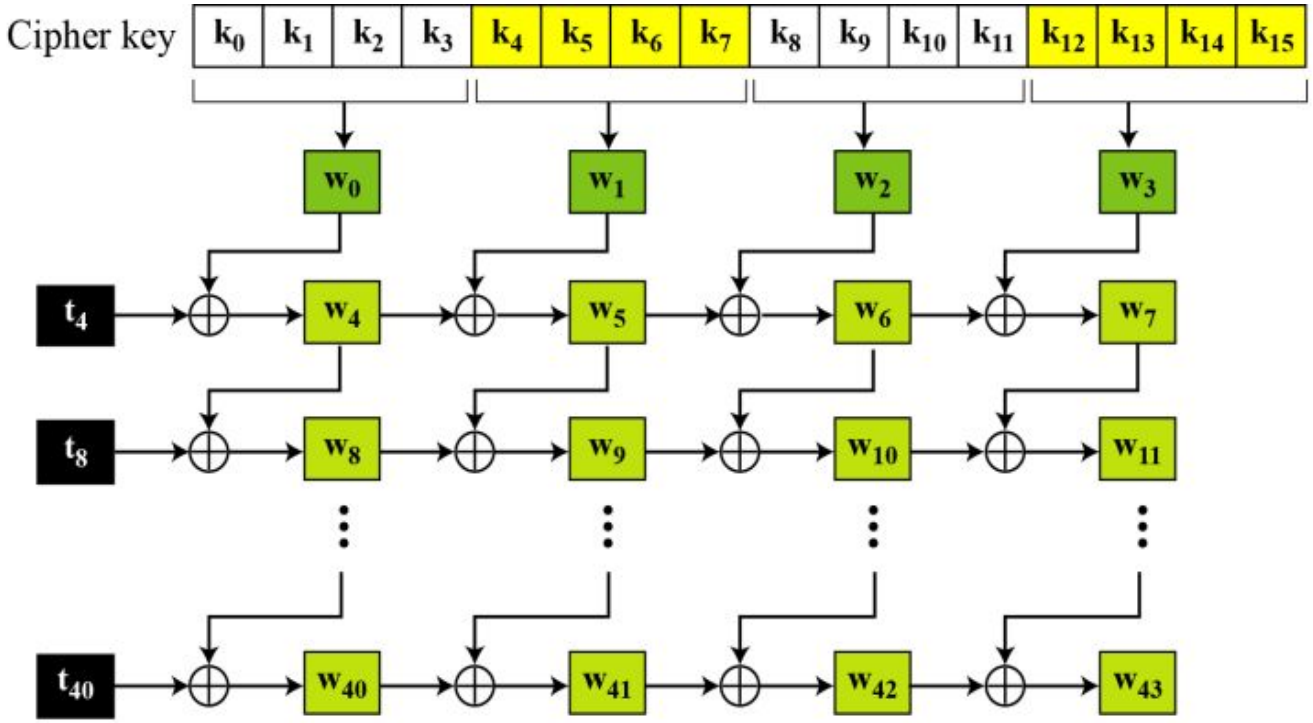
RAMAIAH Institute of Technology

Key Expansion

$t_i = subword(Rotword(w_{i-1})) \oplus Rcon_{i/4}$

Rcon : Round constant is a 4byte value in which the rightmost 3bytes are always zero

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	$(\underline{01}\ 00\ 00\ 00)_{16}$	6	$(\underline{20}\ 00\ 00\ 00)_{16}$
2	$(\underline{02}\ 00\ 00\ 00)_{16}$	7	$(\underline{40}\ 00\ 00\ 00)_{16}$
3	$(\underline{04}\ 00\ 00\ 00)_{16}$	8	$(\underline{80}\ 00\ 00\ 00)_{16}$
4	$(\underline{08}\ 00\ 00\ 00)_{16}$	9	$(\underline{1B}\ 00\ 00\ 00)_{16}$
5	$(\underline{10}\ 00\ 00\ 00)_{16}$	10	$(\underline{36}\ 00\ 00\ 00)_{16}$



RAMAIAH
Institute of Technology

Key Expansion

Key Expansion in AES-128 - Algorithm

KeyExpansion ([key_0 to key_{15}], [$\mathbf{w}_0$ **to** $\mathbf{w}_{43}$])

```
{
    for (i = 0 to 3)
        w_i ← key_4i + key_4i+1 + key_4i+2 + key_4i+3

    for (i = 4 to 43)
    {
        if (i mod 4 ≠ 0)    w_i ← w_i−1 + w_i−4
        else
        {
            t ← SubWord (RotWord (w_i−1)) ⊕ RCon_i/4
            w_i ← t + w_i−4
        }
    }
}
```

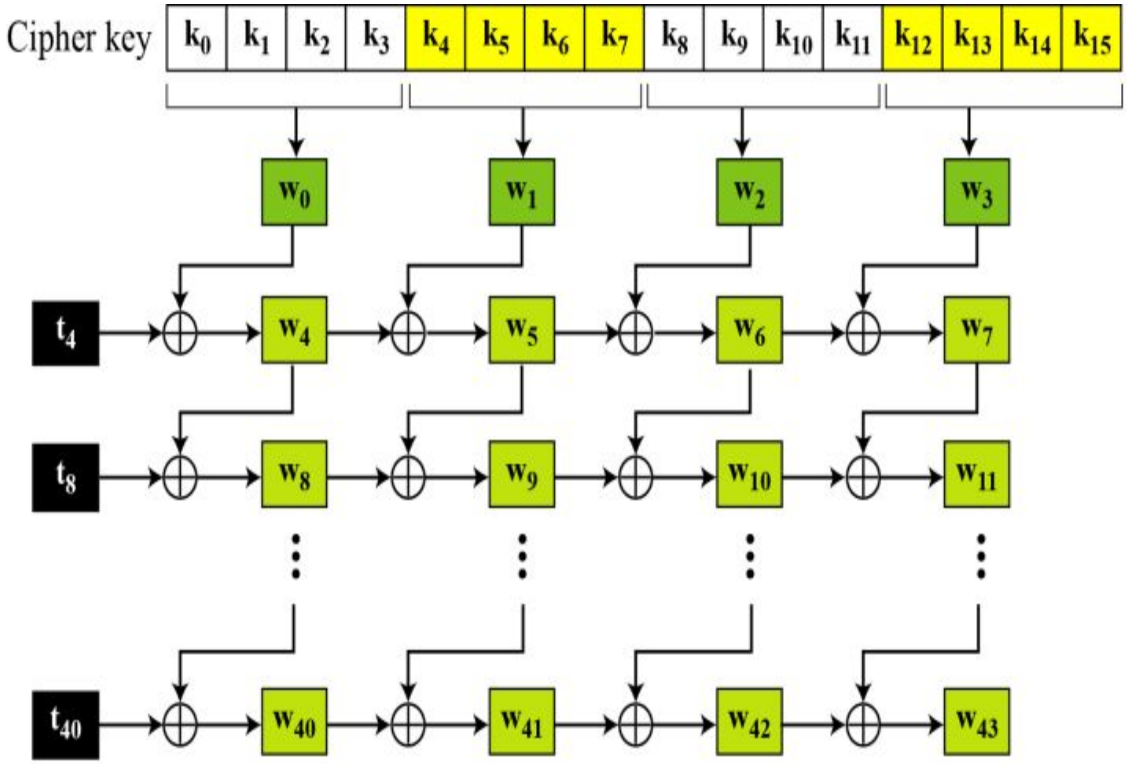
RAMAIAH
Institute of Technology

Key Expansion

Table 7.5 shows how the keys for each round are calculated assuming that the 128-bit cipher key agreed upon by Alice and Bob is (24 75 A2 B3 34 75 56 88 31 E2 12 00 13 AA 54 87)$_{16}$.

Table 7.5 *Key expansion example*

Round	*Values of t's*	*First word in the round*	*Second word in the round*	*Third word in the round*	*Fourth word in the round*
—		w_{00} = 2475A2B3	w_{01} = 34755688	w_{02} = 31E21200	w_{03} = 13AA5487
1	AD20177D	w_{04} = 8955B5CE	w_{05} = BD20E346	w_{06} = 8CC2F146	w_{07} = 9F68A5C1
2	470678DB	w_{08} = CE53CD15	w_{09} = 73732E53	w_{10} = FFB1DF15	w_{11} = 60D97AD4
3	31DA48D0	w_{12} = FF8985C5	w_{13} = 8CFAAB96	w_{14} = 734B7483	w_{15} = 2475A2B3
4	47AB5B7D	w_{16} = B822deb8	w_{17} = 34D8752E	w_{18} = 479301AD	w_{19} = 54010FFA
5	6C762D20	w_{20} = D454F398	w_{21} = E08C86B6	w_{22} = A71F871B	w_{23} = F31E88E1
6	52C4F80D	w_{24} = 86900B95	w_{25} = 661C8D23	w_{26} = C1030A38	w_{27} = 321D82D9
7	E4133523	w_{28} = 62833EB6	w_{29} = 049FB395	w_{30} = C59CB9AD	w_{31} = F7813B74
8	8CE29268	w_{32} = EE61ACDE	w_{33} = EAFE1F4B	w_{34} = 2F62A6E6	w_{35} = D8E39D92
9	0A5E4F61	w_{36} = E43FE3BF	w_{37} = 0EC1FCF4	w_{38} = 21A35A12	w_{39} = F940C780
10	3FC6CD99	w_{40} = DBF92E26	w_{41} = D538D2D2	w_{42} = F49B88C0	w_{43} = 0DDB4F40



RAMAIAH Institute of Technology

Key Expansion

Table 7.5 shows how the keys for each round are calculated assuming that the 128-bit cipher key agreed upon by Alice and Bob is (24 75 A2 B3 34 75 56 88 31 E2 12 00 13 AA 54 87)$_{16}$.

Table 7.5 *Key expansion example*

Round	*Values of t's*	*First word in the round*	*Second word in the round*	*Third word in the round*	*Fourth word in the round*
—		w_{00} = 2475A2B3	w_{01} = 34755688	w_{02} = 31E21200	w_{03} = 13AA5487
1	AD20177D	w_{04} = 8955B5CE	w_{05} = BD20E346	w_{06} = 8CC2F146	w_{07} = 9F68A5C1
2	470678DB	w_{08} = CE53CD15	w_{09} = 73732E53	w_{10} = FFB1DF15	w_{11} = 60D97AD4
3	31DA48D0	w_{12} = FF8985C5	w_{13} = 8CFAAB96	w_{14} = 734B7483	w_{15} = 2475A2B3
4	47AB5B7D	w_{16} = B822deb8	w_{17} = 34D8752E	w_{18} = 479301AD	w_{19} = 54010FFA
5	6C762D20	w_{20} = D454F398	w_{21} = E08C86B6	w_{22} = A71F871B	w_{23} = F31E88E1
6	52C4F80D	w_{24} = 86900B95	w_{25} = 661C8D23	w_{26} = C1030A38	w_{27} = 321D82D9
7	E4133523	w_{28} = 62833EB6	w_{29} = 049FB395	w_{30} = C59CB9AD	w_{31} = F7813B74
8	8CE29268	w_{32} = EE61ACDE	w_{33} = EAFE1F4B	w_{34} = 2F62A6E6	w_{35} = D8E39D92
9	0A5E4F61	w_{36} = E43FE3BF	w_{37} = 0EC1FCF4	w_{38} = 21A35A12	w_{39} = F940C780
10	3FC6CD99	w_{40} = DBF92E26	w_{41} = D538D2D2	w_{42} = F49B88C0	w_{43} = 0DDB4F40

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	(**01** 00 00 00)$_{16}$	6	(**20** 00 00 00)$_{16}$
2	(**02** 00 00 00)$_{16}$	7	(**40** 00 00 00)$_{16}$
3	(**04** 00 00 00)$_{16}$	8	(**80** 00 00 00)$_{16}$
4	(**08** 00 00 00)$_{16}$	9	(**1B** 00 00 00)$_{16}$
5	(**10** 00 00 00)$_{16}$	10	(**36** 00 00 00)$_{16}$

$t_i = \text{subword}(\text{Rotword}(w_{i-1})) \oplus \text{Rcon}_{i/4}$

$t_4 = \text{subword}(\text{Rotword}(w_{4-1})) \oplus \text{Rcon}_{4/4}$

$t_4 = \text{subword}(\text{Rotword}(w_3)) \oplus \text{Rcon}_1$

Rotword(13AA5487) = AA548713

Subword (AA548713) = AC20177D

$t_4 = \text{AC20177D} \oplus \text{Rcon}_1$

$= \text{AC20177D} \oplus \mathbf{01\ 00\ 00\ 00} = \mathbf{AD20177D}$

Key Expansion –AES 192 and AES 256

Key-expansion algorithms in the AES-192 and AES-256 versions are very similar to the key expansion algorithm in AES-128, with the following differences:

1. In AES-192, the words are generated in groups of six instead of four.
   a. The cipher key creates the first six words (w_0 to w_5).
   b. If $i \bmod 6 \neq 0$, $w_i \leftarrow w_{i-1} + w_{i-6}$; otherwise, $w_i \leftarrow t + w_{i-6}$.
2. In AES-256, the words are generated in groups of eight instead of four.
   a. The cipher key creates the first eight words (w_0 to w_7).
   b. If $i \bmod 8 \neq 0$, $w_i \leftarrow w_{i-1} + w_{i-8}$; otherwise, $w_i \leftarrow t + w_{i-8}$.
   c. If $i \bmod 4 = 0$, but $i \bmod 8 \neq 0$, then $w_i = \text{SubWord}(w_{i-1}) + w_{i-8}$.

Key expansion analysis

The key-expansion mechanism in AES has been designed to provide several features that thwart the cryptanalyst.

1. Even if Eve knows only part of the cipher key or the values of the words in some round keys, she still needs to find the rest of the cipher key before she can find all round keys. This is because of the nonlinearity produced by SubWord transformation in the key-expansion process.
2. Two different cipher keys, no matter how similar to each other, produce two expansions that differ in at least a few rounds.
3. Each bit of the cipher key is diffused into several rounds. For example, changing a single bit in the cipher key, will change some bits in several rounds.
4. The use of the constants, the RCons, removes any symmetry that may have been created by the other transformations.
5. There are no serious weak keys in AES, unlike in DES.
6. The key-expansion process can be easily implemented on all platforms.
7. The key-expansion routine can be implemented without storing a single table; all calculations can be done using the $GF(2^8)$ and FG(2) fields.

Unit II (Text1)

Advanced Encryption Standard(AES): (Chapter 7)

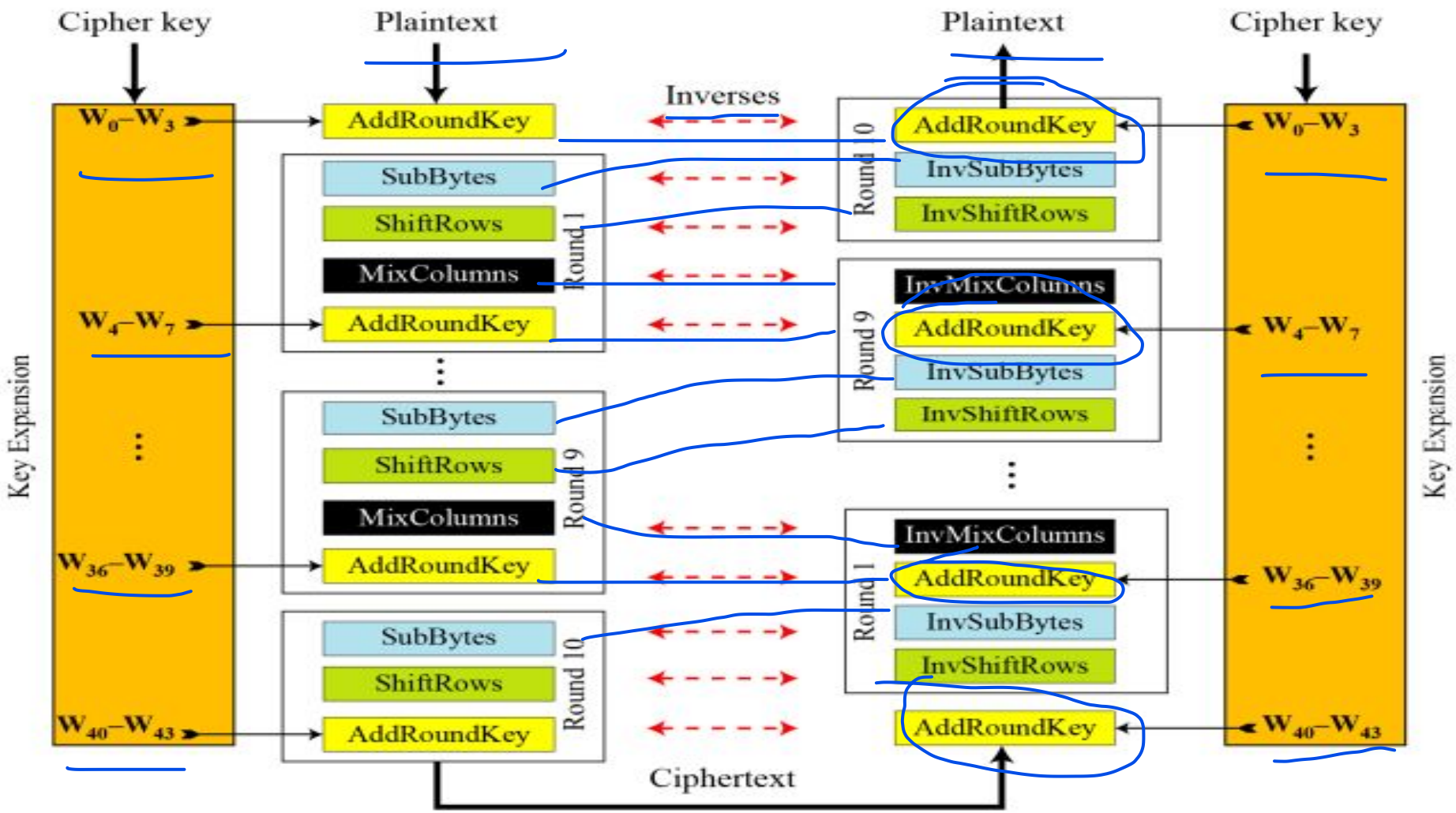
- Introduction
- Transformations
- Key Expansion
- The AES Ciphers
- Examples
- Analysis of AES.

The AES Ciphers

- AES uses four types of transformations for encryption and decryption.
- Encryption algorithm is referred to as the cipher
- Decryption algorithm as the inverse cipher.
- Two different design for implementation
  - Original Design
  - Alternative Design

RAMAIAH
Institute of Technology

The AES Ciphers – Original design



The AES Ciphers –Original design

The code for the AES-128 version of this design is shown in Algorithm

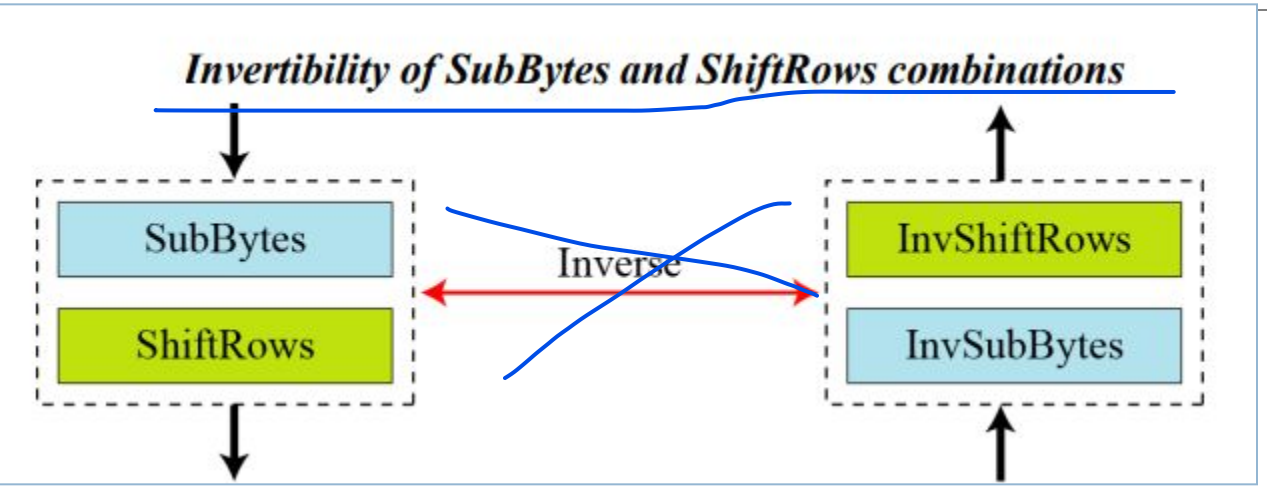
Pseudocode for cipher in the original design

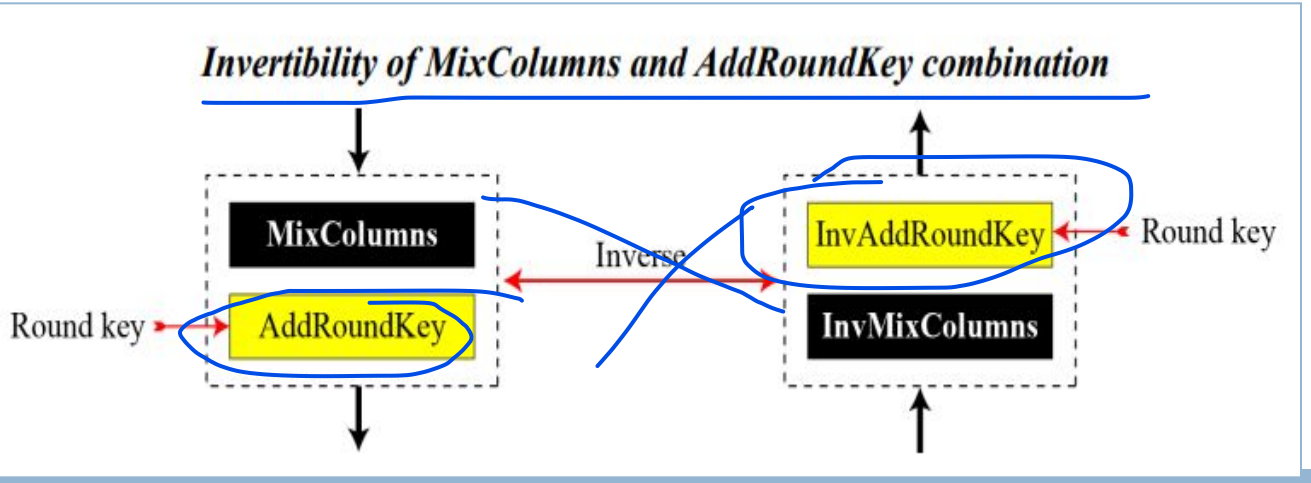
```
Cipher (InBlock [16], OutBlock[16], w[0 … 43])
{
    BlockToState (InBlock, S)

    S ← AddRoundKey (S, w[0…3])
    for (round = 1 to 10)
    {
        S ← SubBytes (S)
        S ← ShiftRows (S)
        if (round ≠ 10)  S ← MixColumns (S)
        S ← AddRoundKey (S, w[4 × round, 4 × round + 3])
    }

    StateToBlock (S, OutBlock);
}
```

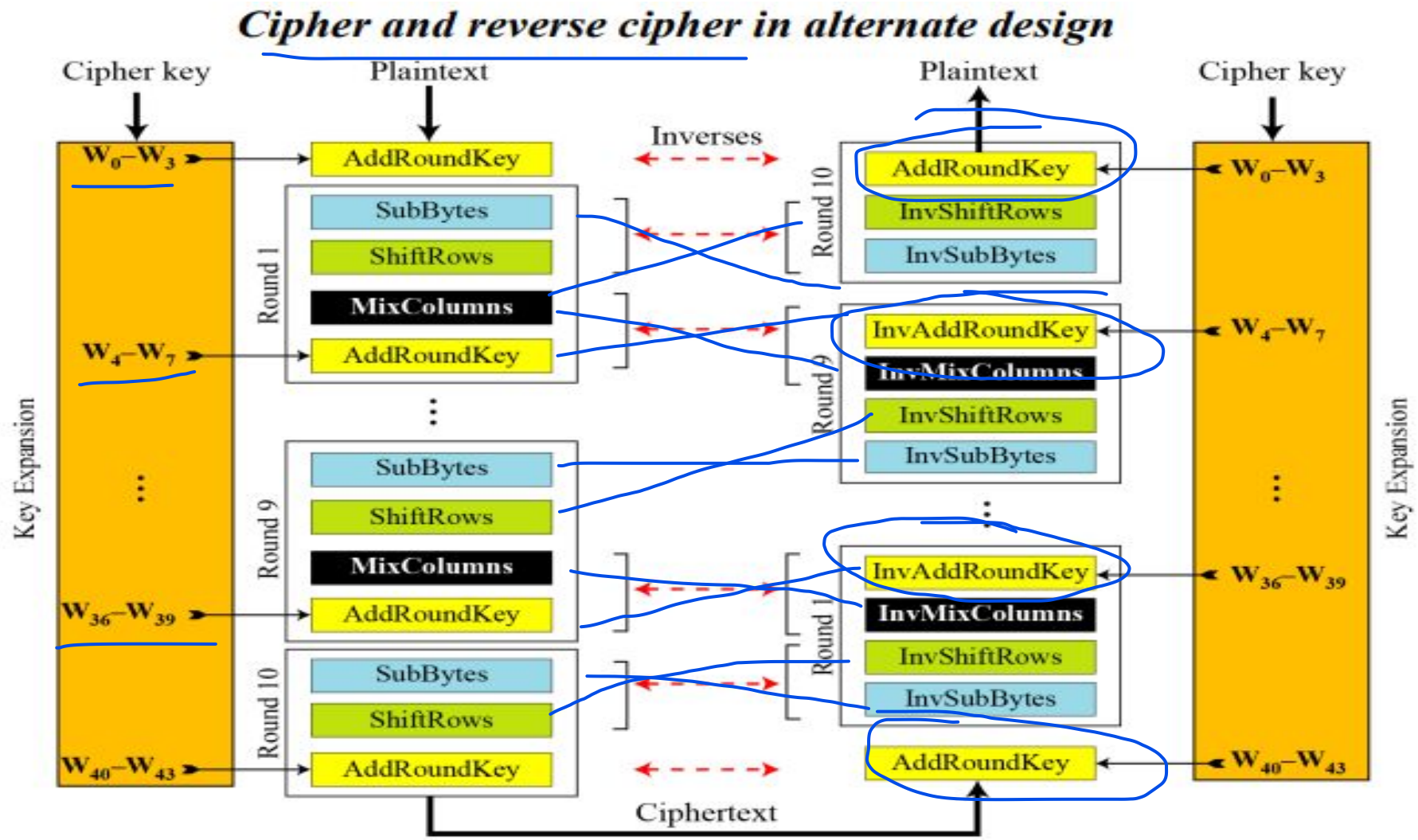
The AES Ciphers –Alternative Design

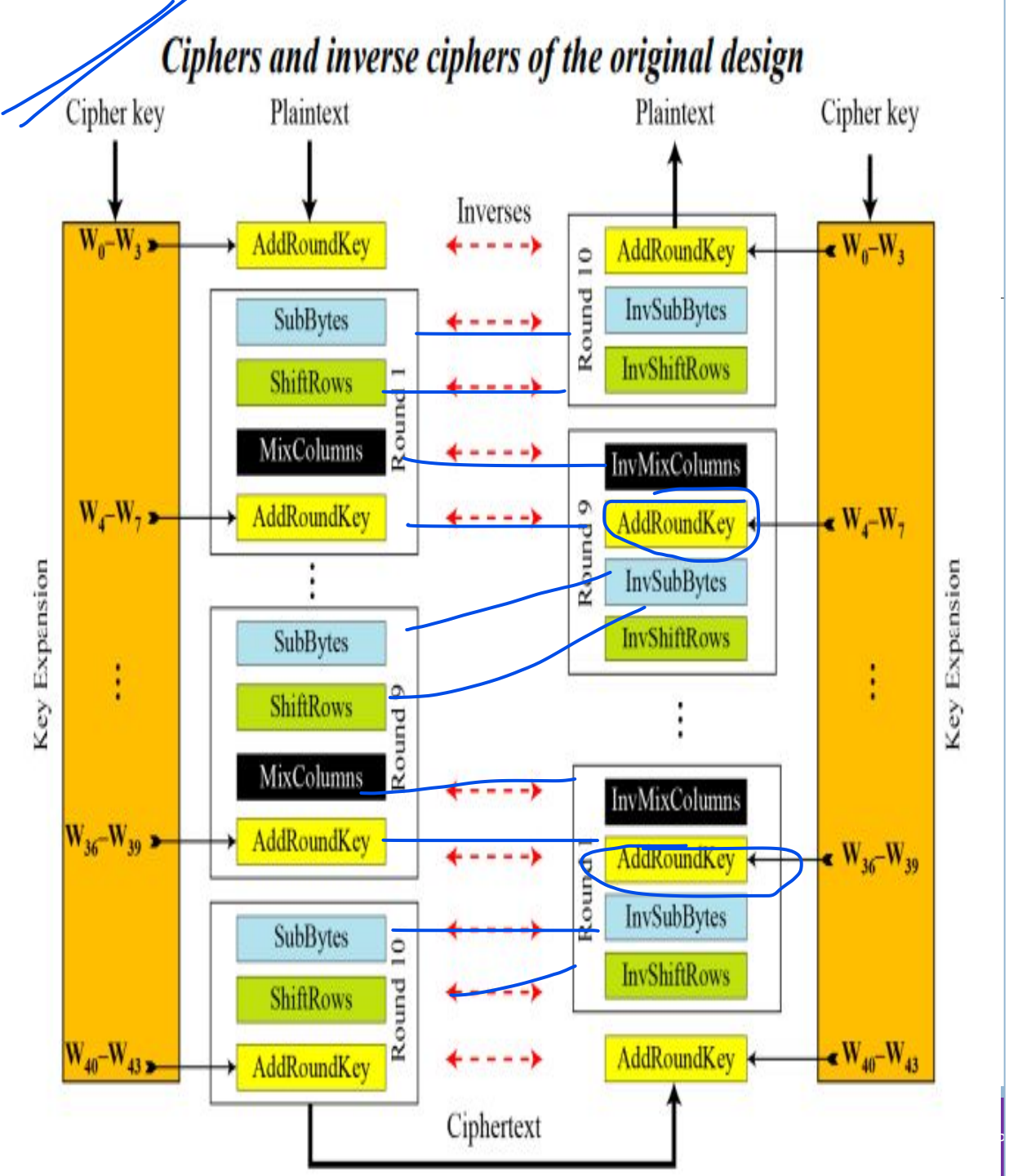


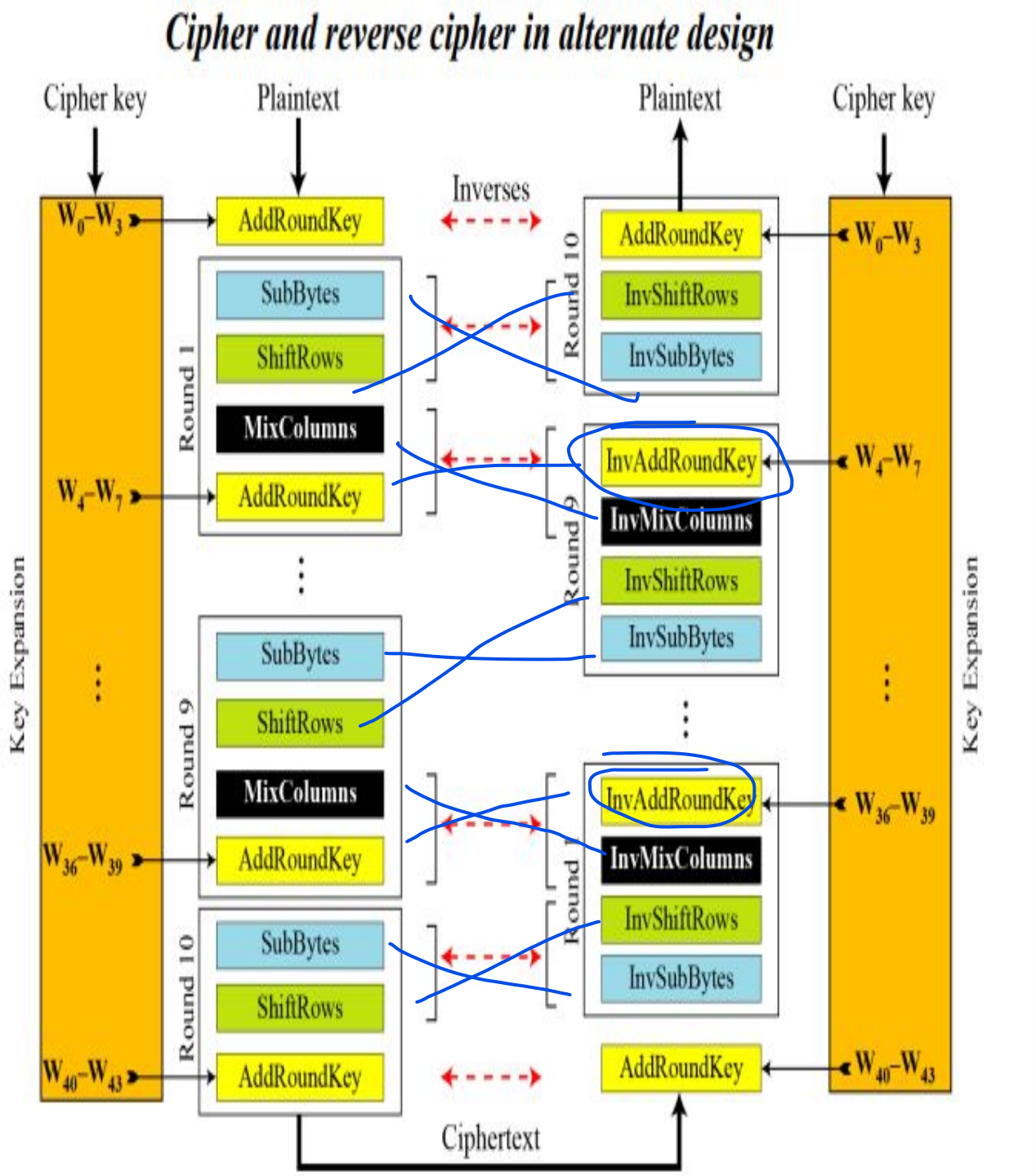


RAMAIAH
Institute of Technology

The AES Ciphers –Alternative Design




Ciphers and inverse ciphers of the original design
Cipher key
Plaintext
Plaintext
Cipher key
Inverses
Key Expansion
Key Expansion
W0–W3
W4–W7
W36–W39
W40–W43
AddRoundKey
SubBytes
ShiftRows
MixColumns
InvMixColumns
InvSubBytes
InvShiftRows
Round 1
Round 9
Round 10
Ciphertext


Cipher and reverse cipher in alternate design
Cipher key
Plaintext
Plaintext
Cipher key
Inverses
Key Expansion
Key Expansion
W0–W3
W4–W7
W36–W39
W40–W43
AddRoundKey
SubBytes
ShiftRows
MixColumns
InvAddRoundKey
InvMixColumns
InvShiftRows
InvSubBytes
Round 1
Round 9
Round 10
Ciphertext

Analysis of AES

This section is a brief review of the three characteristics of AES.

- Security
- Implementation
- Simplicity and Cost

Analysis of AES

Security

- Brute-Force Attack :AES is definitely more secure than DES due to the larger-size key.
- Statistical Attacks :Numerous tests have failed to do statistical analysis of the ciphertext.
- Differential and Linear Attacks :There are no differential and linear attacks on AES as yet.

Analysis of AES

Implementation

AES can be implemented in software, hardware, and firmware.

The implementation can use table lookup process or routines that use a well-defined algebraic structure.

Analysis of AES

Simplicity and Cost

The algorithms used in AES are so simple that they can be easily implemented using cheap processors and a minimum amount of memory.

RAMAIAH
Institute of Technology

Unit III(Text1)

Advanced Encryption Standard(AES): (Chapter 7)

- Introduction
- Transformations
- Key Expansion
- The AES Ciphers
- Examples
- Analysis of AES.

Asymmetric Key Cryptography: (Chapter 10)

- Introduction
- RSA Cryptosystem
- Rabin Cryptosystem
- Elgamal Cryptosystem

Asymmetric Key Cryptography

- Distinguish between two cryptosystems: symmetric-key and asymmetric-key
- Knapsack cryptosystem as one of the first ideas in asymmetric-key cryptography
- RSA cryptosystem
- Rabin cryptosystem
- ElGamal cryptosystem

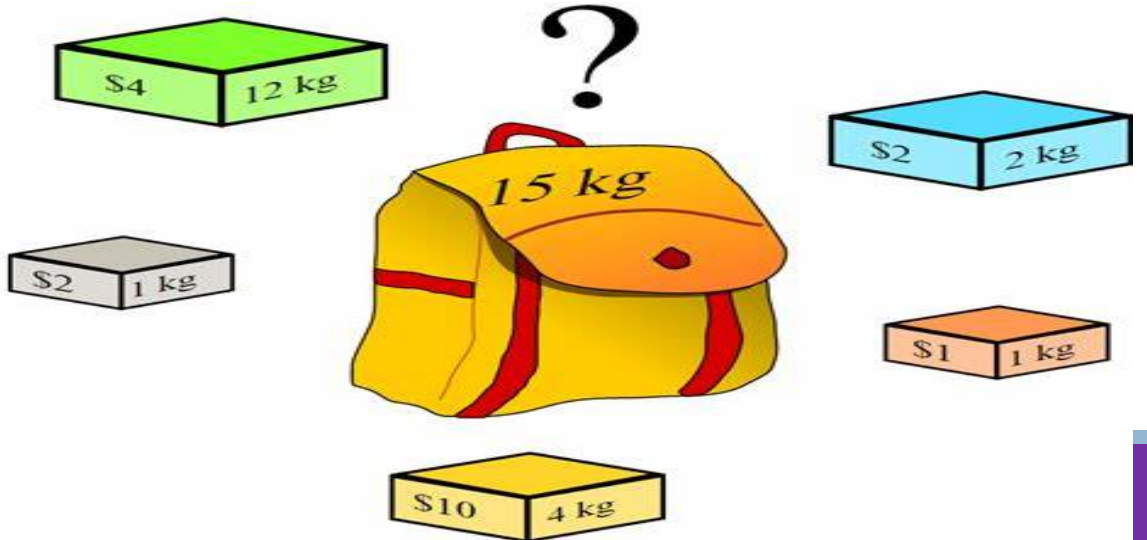
Introduction

- Symmetric and asymmetric-key cryptography will exist in parallel and continue to serve the community.
- The knapsack problem has historical significance in asymmetric cryptography, as it served as the foundation for one of the earliest public-key cryptosystems.

RAMAIAH
Institute of Technology

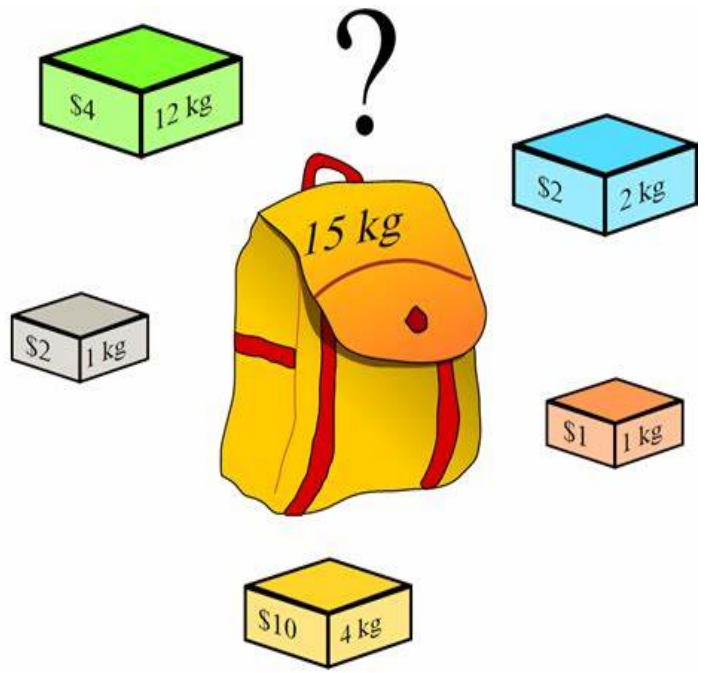
Knapsack Cryptosystem

- The knapsack problem is a well-known combinatorial optimization problem
- It is developed by Ralph Merkle and Mertin Hellman in 1978.
- The Knapsack Cryptosystem is first Public-Key cryptography



Knapsack Cryptosystem

The knapsack algorithm works like this:



Given a set of items, each with a weight and value, determine which items to include in a knapsack so that the total weight doesn't exceed a given limit and the total value is maximized.

RAMAIAH Institute of Technology

Knapsack Cryptosystem

Let us look at an example:

- Imagine you had a set of weights 1, 6, 8, 15 and 24.
- To pack a knapsack weighing 30, you could use weights 1, 6, 8 and 15.

1+6+8+15 = 30

- Represent the weight 30 by the binary code – 1 1 1 1 0
- So, if someone sends the code 30 this can only have come from the plain text 11110 .

Plain text	1 0 0 1 1	1 1 0 1 0	0 1 0 1 1	0 0 0 0 0
Knapsack	1 6 8 15 24	1 6 8 15 24	1 6 8 15 24	1 6 8 15 24
Cipher text	1 + 15 + 24 = 40	1 + 6 + 15 = 22	6 + 15 + 24 = 45	0 = 0

Public key

Knapsack Cryptosystem

- When the Knapsack Algorithm is used in public key cryptography, the idea is to create two different knapsack problems.
- One is easy to solve, the other not.
- Using the easy knapsack, the hard knapsack is derived from it. The hard knapsack becomes the public key.
- The easy knapsack is the private key.
- The public key can be used to encrypt messages, but cannot be used to decrypt messages.
- The private key decrypts the messages.

Knapsack Cryptosystem

Suppose given two k-tuples

$a=[a_1,a_2,.....a_k]$ and $x=[x_1,x_2.....x_k]$

The first tuple is the predefined set

The second tuple in which x_i is only 0 or 1

The sum of elements in the knapsack is

s=knapsackSum(a,x)
x=inv_knapsackSum(s,a)

$s=knapsackSum(a,x) = x_1a_1+x_2a_2 +.....x_ka_k,$

given a and x it is easy to calculate to S, however given the value of s and a it is difficult to compute x

Knapsack Cryptosystem

Superincreasing tuple

- Easy to calculate knapsackSum(a,x) and inv_knapsackSum(s,a) if the k-tuple a is superincreasing
- In superincreasing tuple $a_i \geq a_1 + a_2 + \ldots\ldots\ldots\ldots a_{i-1}$
- Every element except a_1 is greater than or equal to the sum of previous elements.

Knapsack Cryptosystem

- An easy knapsack problem is one in which the weights are in a superincreasing sequence.
- A superincreasing sequence is one in which the next term of the sequence is greater than the sum of all preceding terms.
- For example,
- The set {1, 2, 4, 9, 20, 38} is superincreasing,
- The set {1, 2, 3, 9, 10, 24} is not because 10 < 1+2+3+9.

Knapsack Cryptosystem -Algorithm

```
knapsackSum (x [1 ... k], a [1 ... k])
{
    s ← 0
    for (i = 1 to k)
    {
      s ← s + a_i × x_i
    }
    return s
}
```

Assume that a=[3, 7, 12, 30, 60,115] and s = 82 . Find tuple x using inv_knapsack sum.

Knapsack Cryptosystem

Assume that a=[17,25,46,94,201,400] and x[0,1,1,0,1,0].

Find S using knapsack sum.

i	a_i	x_i	$s \leftarrow s + a_i * x_i$	s
1	17	0	0 + 17 * 0	0
2	25	1	0 + 25 * 1	25
3				
4				
5				
6				

```
knapsackSum (x [1 … k], a [1 … k])
{
    s ← 0
    for (i = 1 to k)
    {
      s ← s + a_i × x_i
    }
    return s
}
```

Knapsack Cryptosystem

Assume that a=[17,25,46,94,201,400] and x[0,1,1,0,1,0].

Find S using knapsack sum.

i	a_i	x_i	$s \leftarrow s + a_i * x_i$	s
1	17	0	0 + 17 * 0	0
2	25	1	0 + 25 * 1	25
3	46	1	25 + 46 * 1	71
4	94	0	71 + 94 * 0	71
5	201	1	71 + 201 * 1	272
6	400	0	272 + 400 * 0	**272**

```
knapsackSum (x [1 … k], a [1 … k])
{
    s ← 0
    for (i = 1 to k)
    {
      s ← s + a_i × x_i
    }
    return s
}
```

Knapsack Cryptosystem

```
inv_knapsackSum (s, a [1 ... k])
{
    for (i = k down to 1)
    {
        if s ≥ a_i
        {
            x_i ← 1
            s ← s − a_i
        }
        else x_i ← 0
    }
    return x [1 ... k]
}
```

RAMAIAH Institute of Technology

Knapsack Cryptosystem

Assume that a=[17,25,46,94,201,400] and s = 272 .

Find tuple x using inv_knapsack sum.

i	a_i	s	$s \geq a_i$	x_i	$s \leftarrow s - a_i \times x_i$
6	400	272	false	$x_6 = 0$	272
5	201	272	true	$x_5 = 1$	71
4				$x_4 =$	
3				$x_3 =$	
2				$x_2 =$	
1				$x_1 =$	

```
inv_knapsackSum (s, a [1 … k])
{
    for (i = k down to 1)
    {
        if s ≥ a_i
        {
            x_i ← 1
            s ← s − a_i
        }
        else x_i ← 0
    }
    return x [1 … k]
}
```

x[0,1,1,0,1,0] = 25 , 46, 201 are in the knapsack

RAMAIAH
Institute of Technology

Knapsack Cryptosystem

Assume that a=[17,25,46,94,201,400] and s = 272 .

Find tuple x using inv_knapsack sum.

i	a_i	s	$s \geq a_i$	x_i	$s \leftarrow s - a_i \times x_i$
6	400	272	false	$x_6 = 0$	272
5	201	272	true	$x_5 = 1$	71
4	94	71	false	$x_4 = 0$	71
3	46	71	true	$x_3 = 1$	25
2	25	25	true	$x_2 = 1$	0
1	17	0	false	$x_1 = 0$	0

```
inv_knapsackSum (s, a [1 … k])
{
    for (i = k down to 1)
    {
        if s ≥ a_i
        {
            x_i ← 1
            s ← s − a_i
        }
        else x_i ← 0
    }
    return x [1 … k]
}
```

x[0,1,1,0,1,0] = 25 , 46, 201 are in the knapsack

Assume that a=[3, 7, 12, 30, 60,115] and s = 82 . Find tuple x using inv_knapsack sum.

Knapsack Cryptosystem

Assume that a=[3, 7, 12, 30, 60,115] and s = 82 .

Find tuple x using inv_knapsack sum.

Knapsack Cryptosystem

Assume that a=[3, 7, 12, 30, 60,115] and s = 82 .

Find tuple x using inv_knapsack sum.

X = [1,1,1,0,1,0]

Knapsack Cryptosystem

Assume that a=[1, 2, 4, 10, 20,40] and s = 11 .

Find tuple x using inv_knapsack sum.

Assume that a=[1, 2, 4, 10, 20,40] and s = 17 .

Find tuple x using inv_knapsack sum.

Assume that a=[1, 2, 4, 10, 20,40] and s = 35 .

Find tuple x using inv_knapsack sum.

Explain Secret communication with Knapsack Cryptosystem.

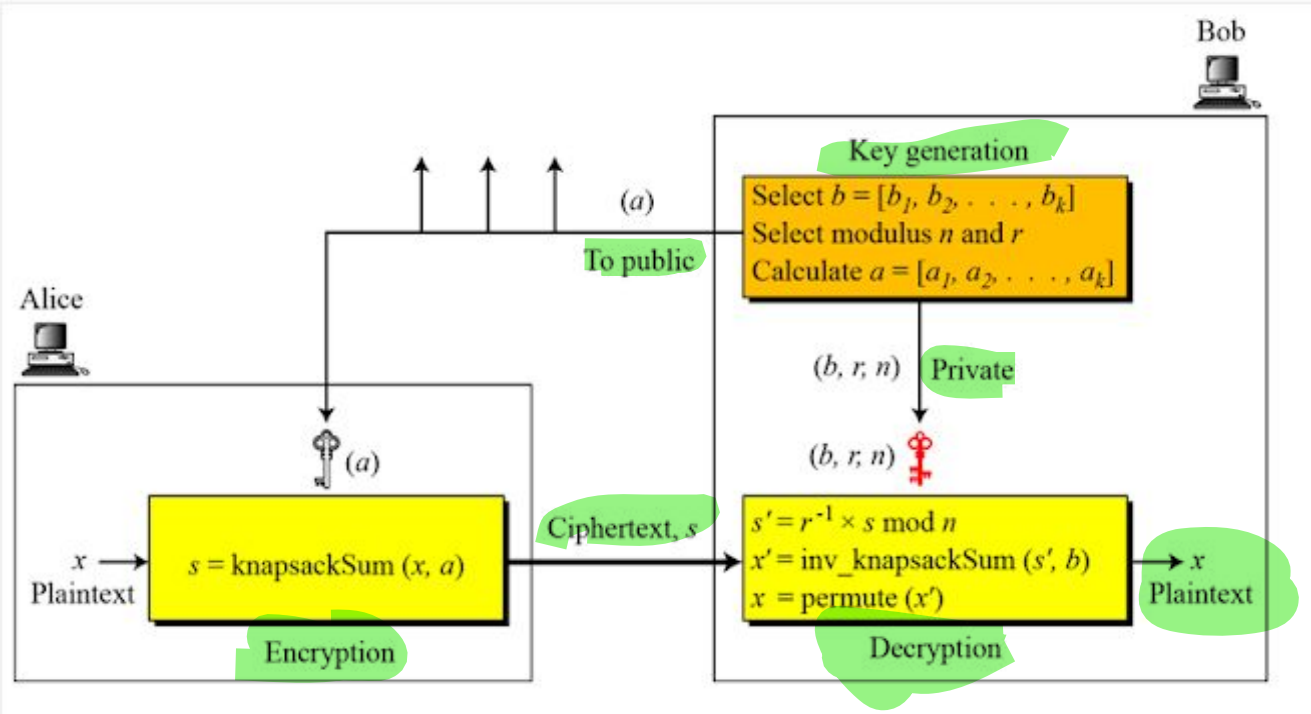
Knapsack Cryptosystem

Secret Communication Process with Knapsack Cryptosystem

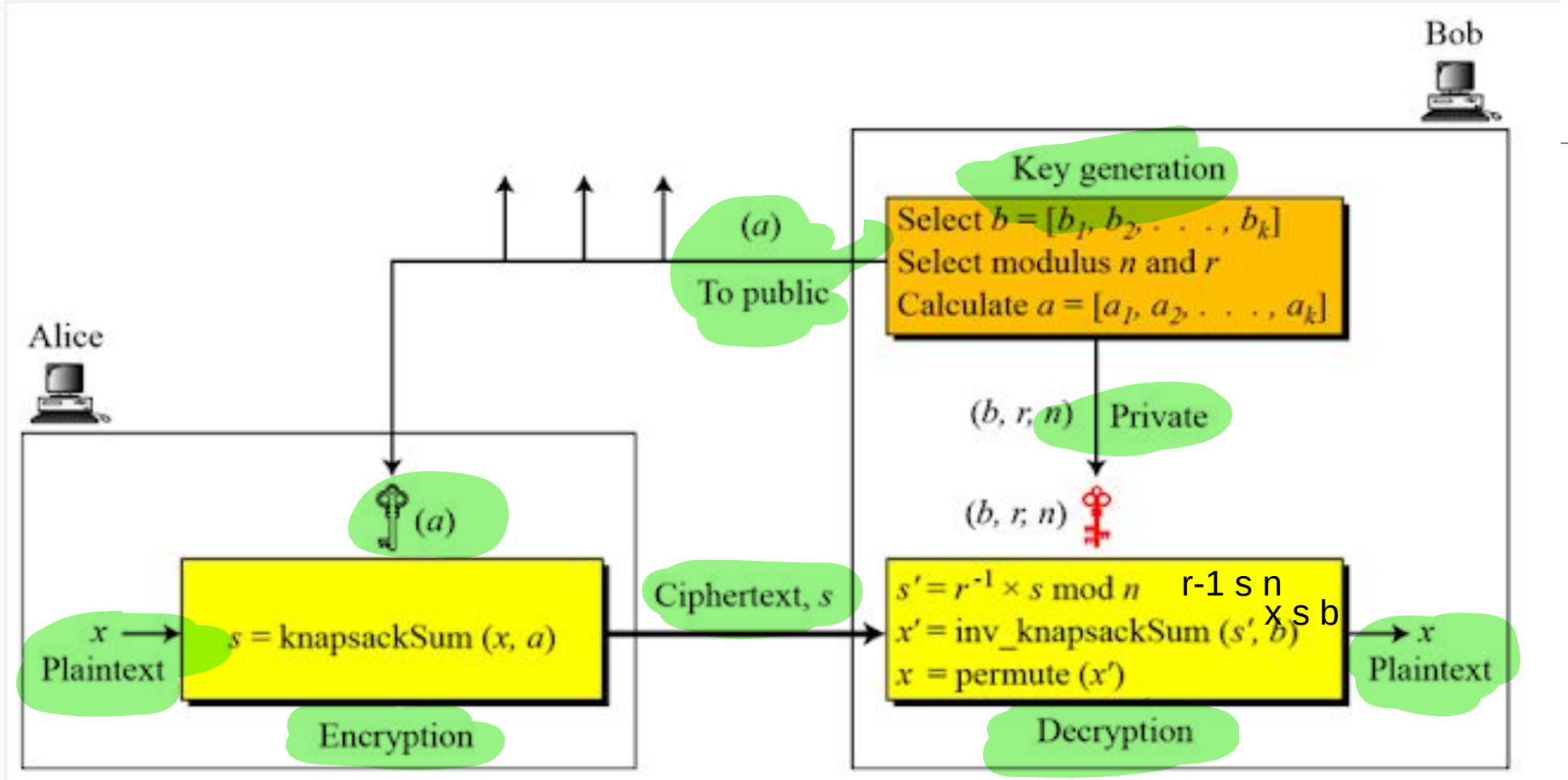
i. Key Generation at Receiver End

ii. Encryption at Sender End

iii. Decryption at Receiver End



Secret Communication with Knapsack Cryptosystem



Secret Communication with Knapsack Cryptosystem

RAMAIAH Institute of Technology

Knapsack Cryptosystem

Key Generation at Receiver End

a. Create a superincreasing k-tuple $b = [b_1, b_2, \ldots, b_k]$ **b={1, 2, 4, 10, 20, 40}.**

b. Choose a modulus n, such that $n > b_1 + b_2 + \cdots + b_k$ for example, n=110

c. Select a random integer r that is relatively prime with n and $1 \le r \le n-1$. for example, r=31

d. Create a temporary k-tuple $t = [t_1, t_2, \ldots, t_k]$ in which $t_i = r \times b_i \bmod n$.

e. Select a permutation of k objects and find a new tuple $a = permute\ (t)$.

f. The public key is the k-tuple a. The private key is n, r, and the k-tuple b.

```
1×31 mod(110) = 31
2×31 mod(110) = 62
4×31 mod(110) = 14
10×31 mod(110) = 90
20×31 mod(110) = 70
40×31 mod(110) = 30
```

Private key → **b={1, 2, 4, 10, 20 40}**

Public key → **a={31, 62, 14, 90, 70, 30}**

RAMAIAH
Institute of Technology

Knapsack Cryptosystem

Encryption at Sender End

Suppose Alice needs to send a message to Bob.

a. Alice converts her message to a k-tuple $x = [x_1, x_2, \ldots, x_k]$ in which x_i is either 0 or 1. The tuple x is the plaintext.

b. Alice uses the *knapsackSum* routine to calculate s. She then sends the value of s as the ciphertext.

x=1001001111001011110

The knapsack contains six weights so we need to split the message into groups of six:

100100

111100

101110

This corresponds to three sets of weights with totals as follows **a={31, 62, 14, 90, 70, 30}**

100100 = 31 + 90 = 121

111100 = 31+62+14+90 = 197

101110 = 31+14+90+70 =205

So the coded cipher message is **s={121, 197, 205}**, will transfer using some channel.

RAMAIAH Institute of Technology

Knapsack Cryptosystem

Decryption at Receiver End

- Receiver receive cipher message **s={121, 197, 205}**, Now the receiver has to decode the message
- The person decoding must know the two numbers n=110 and r= 31
- We need r^{-1}, which is a multiplicative inverse $r^{-1} \bmod n = 71$

121×71 mod(110) = 11 = 1+10=>100100
197×71 mod(110) = 17 = 1+2+4+10=>111100
205×71 mod(110) = 35 = 1+4+10+20=>101110

b={1, 2, 4, 10, 20, 40}.

Bob receives the ciphertext s.

a. Bob calculates $s' = r^{-1} \times s \bmod n$.

b. Bob uses *inv_knapsackSum* to create x'.

c. Bob permutes x' to find x. The tuple x is the recovered plaintext.

Knapsack Cryptosystem

Assume that a=[1, 2, 4, 10, 20,40] and s = 11 .

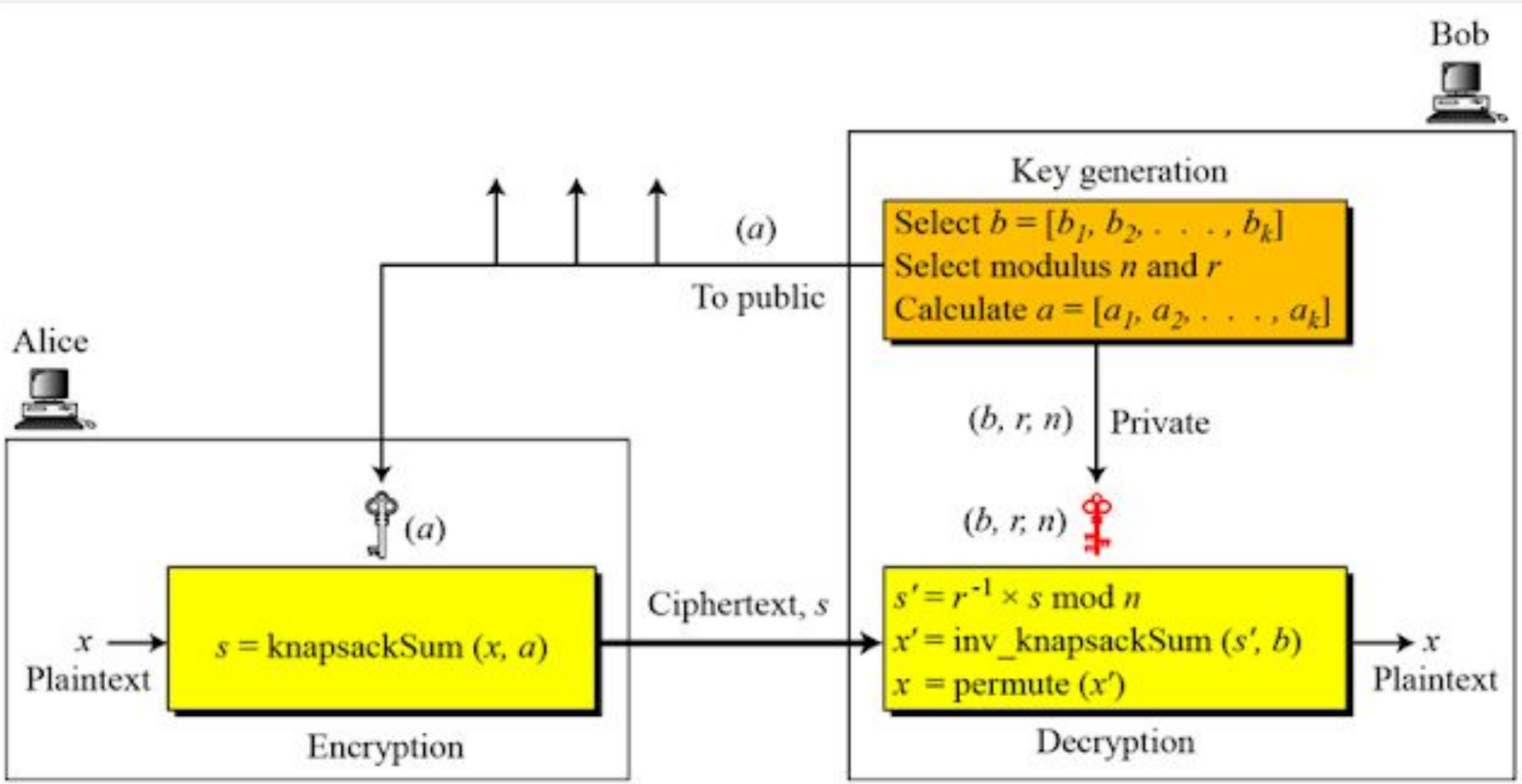
Find tuple x using inv_knapsack sum.

Assume that a=[1, 2, 4, 10, 20,40] and s = 17 .

Find tuple x using inv_knapsack sum.

Assume that a=[1, 2, 4, 10, 20,40] and s = 35 .

Find tuple x using inv_knapsack sum.



Secret Communication with Knapsack Cryptosystem

1. Key generation:
   a. Bob creates the superincreasing tuple b = [7, 11, 19, 39, 79, 157, 313].
   b. Bob chooses the modulus n = 900 and r = 37, and [4 2 5 3 1 7 6] as permutation table.
   c. Bob now calculates the tuple t = [259, 407, 703, 543, 223, 409, 781].
   d. Bob calculates the tuple a = permute (t) = [543, 407, 223, 703, 259, 781, 409].
   e. Bob publicly announces a; he keeps n, r, and b secret.

2. Suppose Alice wants to send a single character "g" to Bob.
   a. She uses the 7-bit ASCII representation of "g", $(1100111)_2$, and creates the tuple x = [1, 1, 0, 0, 1, 1, 1]. This is the plaintext.
   b. Alice calculates $s = knapsackSum\ (a, x) = 2165$. This is the ciphertext sent to Bob.

3. Bob can decrypt the ciphertext, $s = 2165$.
   a. Bob calculates $s' = s \times r^{-1} \bmod n = 2165 \times 37^{-1} \bmod 900 = 527$.
   b. Bob calculates $x' = Inv\_knapsackSum\ (s', b)$ = [1, 1, 0, 1, 0, 1, 1].
   c. Bob calculates $x = permute\ (x')$ = [1, 1, 0, 0, 1, 1, 1]. He interprets the string $(1100111)_2$ as the character "g".

DES/AES/RSA

TABLE I. COMPARATIVE ANALYSIS BETWEEN AES, DES AND RSA

Features	DES	AES	RSA
Developed	1977	2000	1977
Key Length	56 bits	128,192,256 bits	More than 1024 bits
Cipher Type	Symmetric block cipher	Symmetric block cipher	Asymmetric block cipher
Block size	64 bits	128 bits	Minimum 512 bits
Security	Not secure enough	Excellent secured	Least secure
Hardware & Software Implementation	Better in hardware than software	Better in both	Not efficient
Encryption and Decryption	Moderate	Faster	Slower

RSA Cryptosystem

- RSA cryptosystem, named after those who invented it in 1978: Ron Rivest, Adi Shamir, and Leonard Adleman.
- The RSA algorithm is an asymmetric cryptography algorithm; uses a public key and a private key (i.e two different, mathematically linked keys).

Key size	Key strength
512 bits	Low-strength key
1024 bits	Medium-strength key
2048 bits	High-strength key
4096 bits	Very high-strength key

RSA Cryptosystem

- The length of the plain text is less than or equal to the key length (Bytes)
- RSA is only able to encrypt data to a maximum amount equal to key size (2048 bits = 256 bytes), minus any padding and header data (11 bytes for padding).
- The padding standards we generally use are NoPPadding, OAEPPadding, PKCS1Padding, etc., among which the padding suggested by PKCS#1 occupies 11 bytes.

RSA Cryptosystem

Length of ciphertext

The length of the ciphertext is the bit length of the key

RSA Cryptosystem

Applicaitons

1. Securing communication on the internet
2. WhatsApp, Telegram, Signal and other messaging services.
3. Securing online Payment
4. Secure Web Browsing

RSA Cryptosystem

Characteristics of RSA

It is a public key encryption technique.

It is safe for exchange of data over internet.

It maintains confidentiality of the data.

RSA has high toughness as breaking into the keys by interceptors is very difficult.

RSA Cryptosystem

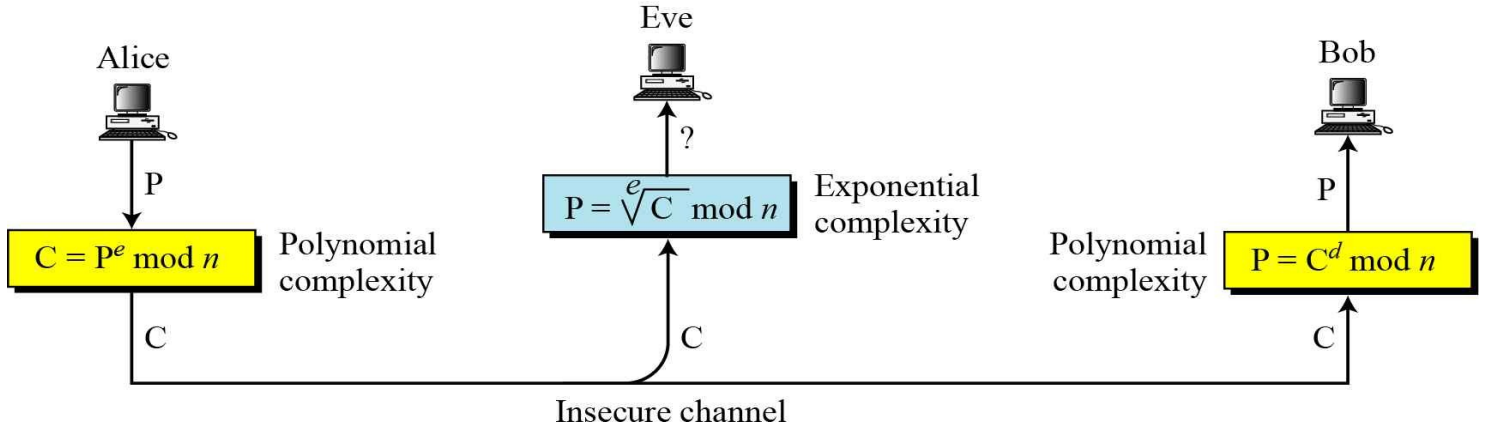
Advantages of RSA

- It is very easy to implement RSA algorithm.
- RSA algorithm is safe and secure for transmitting confidential data.
- Cracking RSA algorithm is very difficult as it involves complex mathematics.
- Sharing public key to users is easy.

Introduction

Complexity of operations in RSA

- P is Plaintext and C is Ciphertext
- RSA uses two exponents e and d, where e is public and d is private.
- Alice uses $C = P^e \bmod n$ to create Ciphertext
- Bob uses $P = C^d \bmod n$ to revert Plaintext



RSA uses modular exponentiation for encryption/decryption; To attack it, Eve needs to calculate $\sqrt[e]{C} \bmod n$.

Introduction

Procedure used in RSA

i. Key Generation
ii. Encryption/Decryption Function

Write RSA Algorithm.

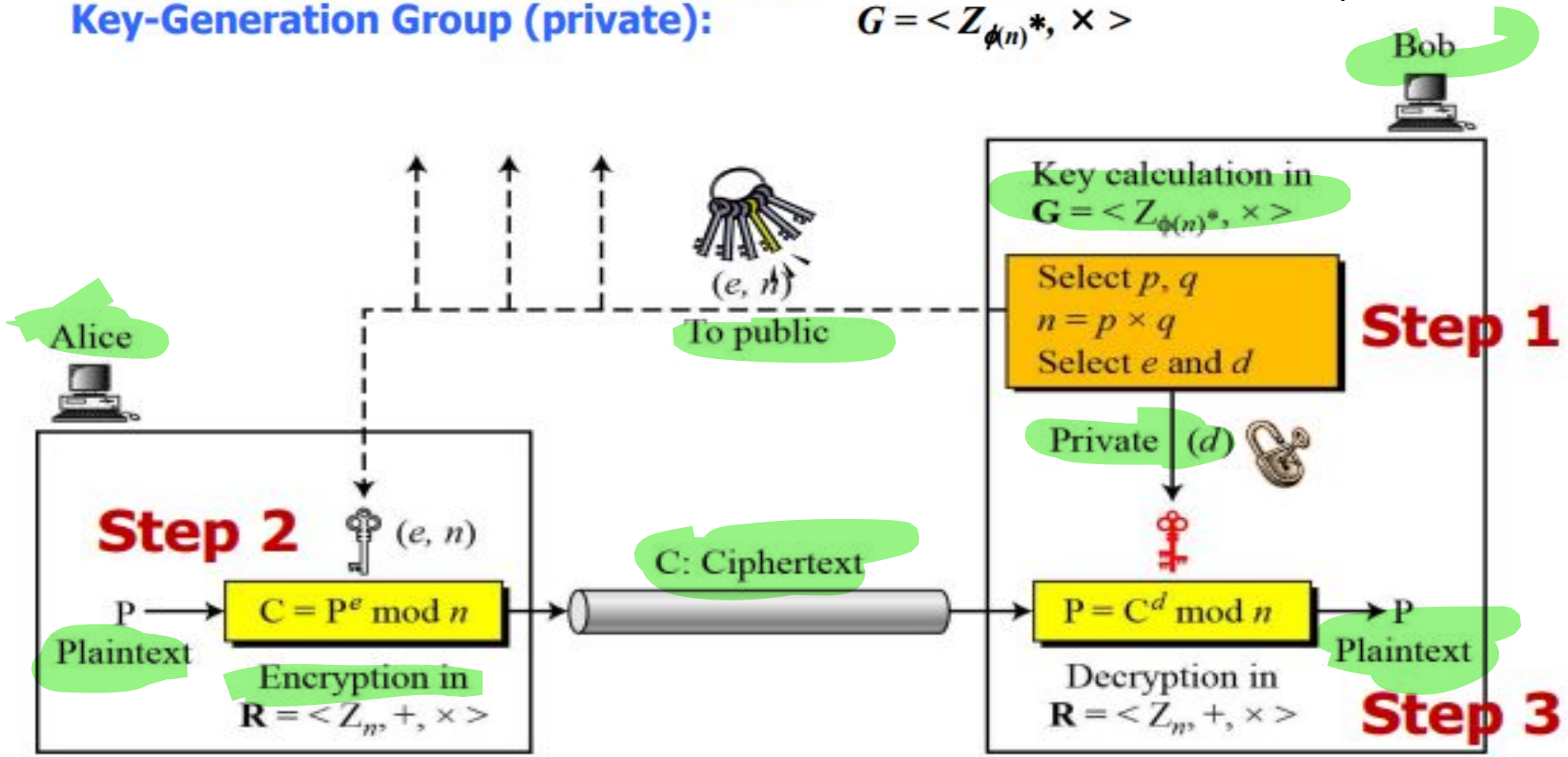
Introduction

RSA uses two algebraic structures :

Encryption/Decryption Ring (public): $R = < Z_n, +, \times >$

Key-Generation Group (private): $G = < Z_{\phi(n)}*, \times >$

In RSA, the tuple (e, n) is the public key; the integer d is the private key.



Write RSA Algorithm.

Introduction

In RSA, p and q must be at least 512 bits; n must be at least 1024 bits.

RSA_Key_Generation

{

Select two large primes p and q such that $p \neq q$.

$n \leftarrow p \times q$

$\phi(n) \leftarrow (p-1) \times (q-1)$

Select e such that $1 < e < \phi(n)$ and e is coprime to $\phi(n)$

$d \leftarrow e^{-1} \bmod \phi(n)$ // d is inverse of e modulo $\phi(n)$

Public_key $\leftarrow$ (e, n) // To be announced publicly

Private_key $\leftarrow d$ // To be kept secret

return Public_key and Private_key

}

RSA_Encryption (P, e, n) // P is the plaintext in Z_n and P < n

{

// Calculation of ($P^e \bmod n$)

C $\leftarrow$ **Fast_Exponentiation** (P, e, n)

return C

}

RSA_Decryption (C, d, n) //C is the ciphertext in Z_n

{

// Calculation of ($C^d \bmod n$)

P $\leftarrow$ **Fast_Exponentiation** (C, d, n)

return P

}

RAMAIAH Institute of Technology

Introduction

Key Generation

- Choose two large prime numbers (p and q)
- Calculate n = p*q and

$$\Phi(n) = (p-1)(q-1)$$

- Choose a number e , where $1 < e < \Phi(n)$

  e is coprime to $\Phi(n)$, $\gcd(e, \Phi(n)) = 1$
- Calculate $d = e^{-1} \bmod n$, or $de \bmod \Phi(n) = 1$
- Public key pair as (e,n)

In RSA, the tuple (e, n) is the public key; the integer d is the private key.

Private key is d

Encryption/Decryption Function

- If the plaintext is P, ciphertext

  $C = P^e \bmod n$
- If the ciphertext is C, plaintext

  $P = C^d \bmod n$

Introduction

Key Generation

- Choose two large prime numbers (p and q)
- Calculate n = p*q and

$$\Phi(n) = (p-1)(q-1)$$

- Choose a number e , where $1 < e < \Phi(n)$

  e is coprime to $\Phi(n)$, $\gcd(e, \Phi(n)) = 1$
- Calculate $d = e^{-1} \bmod n$, or $de \bmod \Phi(n) = 1$
- Public key pair as (e,n)
- Private key is d

Choose p = 3 and q = 11

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

Compute $\varphi(n) = (p - 1) * (q - 1) = 2 * 10 = 20$

Choose e = ? □ 3

gcd(e, Φ(n)) =1
gcd(2 ,20) = 2
gcd(3 ,20) = 1

Introduction

Key Generation

- Choose two large prime numbers (p and q)
- Calculate n = p*q and

  $\Phi(n) = (p-1)(q-1)$
- Choose a number e , where $1 < e < \Phi(n)$

  e is coprime to $\Phi(n)$, gcd(e, $\Phi(n)$) =1
- Calculate $d = e^{-1} \bmod n$, or $de \bmod \Phi(n) = 1$
- Public key pair as (e,n)
- Private key is d

Let e = 3

d e mod Φ(n) = 1
(1 * 3) mod 20 = 3
(2 * 3) mod 20 = 6
(3 * 3) mod 20 = 9
(4* 3) mod 20 = 12
(5 * 3) mod 20 = 15
(6 * 3) mod 20 = 8
(7 * 3) mod 20 = 1

d = 7

Public key is (e, n) => (3, 33)
Private key is d => 7

Introduction

Encryption/Decryption Function

- If the plaintext is P, ciphertext

  $C = P^e \bmod n$

- If the ciphertext is C, plaintext

  $P = C^d \bmod n$

The encryption of P = 2

$C = 2^3 \bmod 33$

$= 8 \bmod 33$?8

The decryption of C =

$P = 8^7 \bmod 33$

$= 2$

Introduction

Key Generation

- Choose two large prime numbers (p and q)
- Calculate n = p*q and

  $\Phi(n) = (p-1)(q-1)$
- Choose a number e , where $1 < e < \Phi(n)$

  e is coprime to $\Phi(n)$, gcd(e, $\Phi(n)$) =1
- Calculate $d = e^{-1} \bmod n$, or de mod $\Phi(n) = 1$
- Public key pair as (e,n)
- Private key is d

Choose p = 3 and q = 11

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

Compute $\varphi(n) = (p - 1) * (q - 1) = 2 * 10 = 20$

Choose e = ?
gcd(e, $\Phi(n)$) =1
gcd(2 ,20) = 2
gcd(3 ,20) = 1
gcd(4 ,20) = 1
gcd(5 ,20) = 1
gcd(6 ,20) = 1
gcd(7 ,20) = 1

Let e = 7

Introduction

Key Generation

- Choose two large prime numbers (p and q)
- Calculate n = p*q and

$$\Phi(n) = (p-1)(q-1)$$

- Choose a number e , where $1 < e < \Phi(n)$

  e is coprime to $\Phi(n)$, gcd(e, $\Phi(n)$) =1
- Calculate d = $e^{-1} \bmod n$, or $de \bmod \Phi(n) = 1$
- Public key pair as (e,n)
- Private key is d

Let e = 7

d e mod Φ(n) = 1
(1 * 7) mod 20 = 7
(2 * 7) mod 20 = 14
(3 * 7) mod 20 = 1

d=3

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

Private key is d => 3

Introduction

Encryption/Decryption Function

- If the plaintext is P, ciphertext

  $C = P^e \bmod n$

- If the ciphertext is C, plaintext

  $P = C^d \bmod n$

The encryption of P = 2

$C = 2^7 \bmod 33$

$= 29$

The decryption of C

$P = 29^3 \bmod 33$

$= 2$

Introduction

Encryption/Decryption Function

- If the plaintext is P, ciphertext

  $C = P^e \bmod n$

- If the ciphertext is C, plaintext

  $P = C^d \bmod n$

The encryption of P = 4

$C = 4^3 \bmod 15$

$= 64 \bmod 15 = 4$

The decryption of C =

$P = 4^3 \bmod 15$

$= 4$

Introduction

Encryption/Decryption Function

- If the plaintext is P, ciphertext

  $C = P^e \bmod n$

- If the ciphertext is C, plaintext

  $P = C^d \bmod n$

Encrypt the word: “dog”

e=7,d=3

word	m	P^e	$C=P^e \bmod n$
d	4	16384	16
o	15	170859375	27
g	7	823543	28

Introduction

Encryption/Decryption Function

- If the plaintext is P, ciphertext

  $C = P^e \bmod n$

- If the ciphertext is C, plaintext

  $P = C^d \bmod n$

Decrypt the code 16 27 28

e=7,d=3

C	C^d	$C^d \bmod n$	
16	4096	4	D
27	19863	15	O
28	21952	7	G

Introduction

Key Generation

- Choose two large prime numbers (p and q)
- Calculate n = p*q and

  $\Phi(n) = (p-1)(q-1)$
- Choose a number e , where $1 < e < \Phi(n)$

  e is coprime to $\Phi(n)$, gcd(e, $\Phi(n)$) =1
- Calculate $d = e^{-1} \bmod n$, or $de \bmod \Phi(n) = 1$
- Public key pair as (e,n)
- Private key is d

Choose p = 3 and q = 5

Compute n = p * q = 3 * 5= 15

Compute φ(n) = (p - 1) * (q - 1) = 2 * 4 = 8

Let e = 3

d = (3 * 3) mod 8 □ 9 mod 8 = 1

Public key is (e, n) => (3, 15)
Private key is d => 3

Example: p = 3 and q = 11

Suppose the message is ‘HELP’, assigned the numbers 2, 3, 4 and 5, respectively

Perform Encryption and Decryption

Example: p = 7 and q = 11,e=13 and d=37

Suppose the message is ‘5’ and ’63’,

Perform Encryption and Decryption

Example: p = 7 and q = 11

Suppose the message is ‘5’,

Perform Encryption and Decryption

C=26 and C= 28

Example

Example 10.5 **Step 1 :** Bob chooses 7 and 11 as p and q and calculates $n = 77$. The value of $\phi(n) = (7 - 1)(11 - 1)$ or 60. Now he chooses two exponents, e and d, from Z_{60}^*. If he chooses e to be 13, then d is 37. Note that $e \times d \bmod 60 = 1$ (they are inverses of each.) **Step 2 :** Now imagine that Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5.

Plaintext: 5 $C = 5^{13} = 26 \bmod 77$ Ciphertext: 26

Step 3 : Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:

Ciphertext: 26 $P = 26^{37} = 5 \bmod 77$ Plaintext: 5

Example 10.6 **Step 2 :** Now assume that another person, John, wants to send a message to Bob. John can use the same public key announced by Bob (probably on his website), 13; John's plaintext is 63. John calculates the following:

Plaintext: 63 $C = 63^{13} = 28 \bmod 77$ Ciphertext: 28

Step 3 : Bob receives the ciphertext 28 and uses his private key 37 to decipher the ciphertext:

Ciphertext: 28 $P = 28^{37} = 63 \bmod 77$ Plaintext: 63

Example: p = 397 and q = 402,e=343 and d=12007

Suppose the message is ‘NO’

Perform Encryption and Decryption

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

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

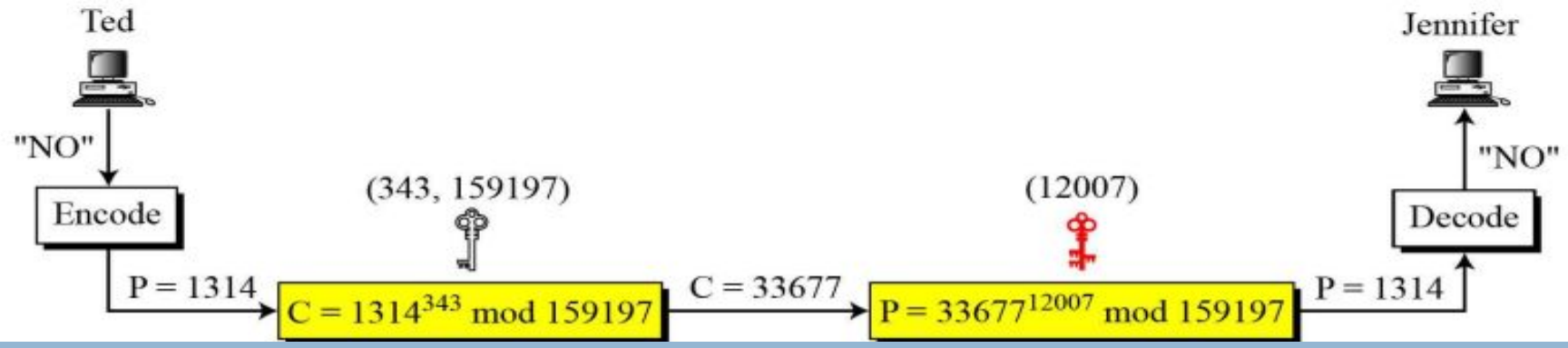
Example

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

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

Example 10.7 **Step 1 :** Jennifer creates a pair of keys for herself. She chooses $p = 397$ and $q = 401$. She calculates $n = 159197$. She then calculates $\phi(n) = 158400$. She then chooses $e = 343$ and $d = 12007$. Show how Ted can send a message to Jennifer if he knows e and n.

Step 2 : Suppose Ted wants to send the message "NO" to Jennifer. He changes each character to a number (from 00 to 25), with each character coded as two digits. He then concatenates the two coded characters and gets a four-digit number. The plaintext is 1314.



$72^{24} \bmod 131$

Binary of exponent 24 is 1 1 0 0 0

b = 72, q=131

Assume d=1	1	1	0	0	0
d	1	72	29	55	12
$d^2 \bmod q$	1	75	55	12	13 (Ans)
d b mod q	72	29			

$2^{97} \bmod 131$

Binary of exponent 97 is 1 1 0 0 0 0 1

b = 2, q=131

Assume d=1	1	1	0	0	0	0	1
d	1	2	8	64	35	46	20
$d^2 \bmod q$	1	4	64	35	46	20	7
d b mod q	2	8					14(Ans)

Example

Bob chooses 7 and 11 as p and q and calculates n value. Find the value of ϕ(n). Now choose the two exponents e and d. Now assume that Alice wants to send the plain text 5 to Bob. Find the cipher text and decrypt it on receiving side to get plaintext using RSA algorithm.

Example

Show the steps of RSA Algorithm. If the RSA public key is (31, 3599), what is the corresponding private key?

Example

Show the steps of RSA Algorithm. If the RSA public key is (31, 3599), what is the corresponding private key?

e=31 and n=3599

p=59 and q=61

phi(n)= 3480

d*e=1 mod 3480

d= 3031

Private key = 3031

Example

Bob chooses 13 and 11 as p and q and calculates n value. Find the value of ϕ(n). Find the two exponents e and d. Now assume that Alice wants to send the plain text 13 to Bob. Find the cipher text and decrypt it on receiving side to get plaintext using RSA algorithm.

Example

Bob chooses 61 and 53 as p and q and calculates n value. Find the value of ϕ(n). Let e= 17, Find the exponents d. Now assume that Alice wants to send the plain text 65 to Bob. Find the cipher text and decrypt it on receiving side to get plaintext using RSA algorithm.

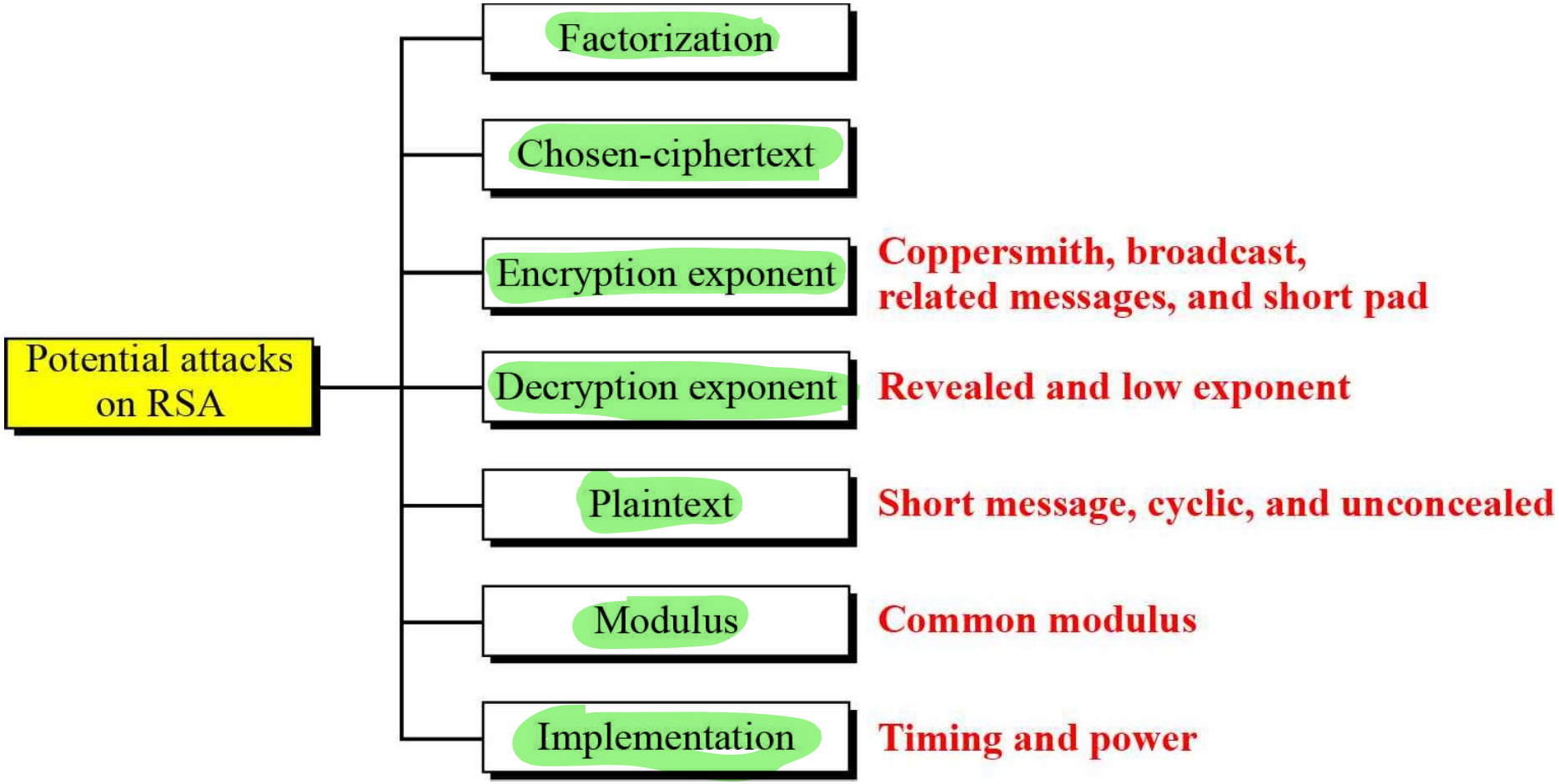
Example

In a RSA cryptosystem a particular A uses two prime numbers p = 13 and q =17 to generate her public and private keys. If the public key of A is 35. Then the private key of A is _____.

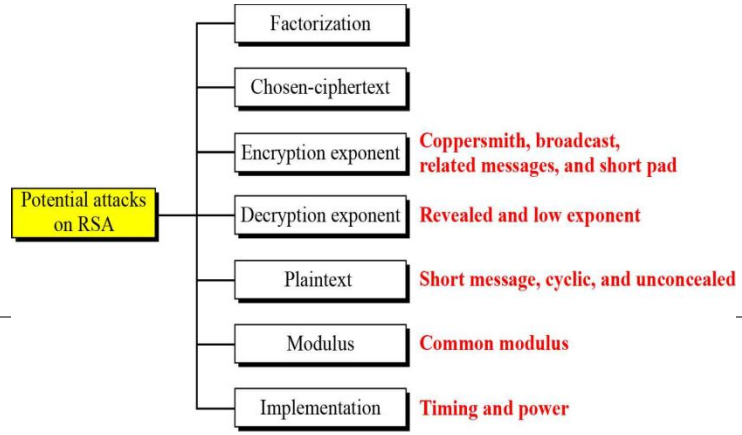
(A) 11
(B) 13
(C) 16
(D) 17

Describe the taxonomy of potential attacks on RSA.

Attacks on RSA



Attacks on RSA



Factoring attacks

Factoring is the act of splitting an integer into a set of smaller integers (factors) which, when multiplied together, form the original integer.

The factoring problem is to find 3 and 5 when given 15.

Factoring an RSA would allow an attacker to figure out the private key(e)

Attacks on RSA

Factoring attacks

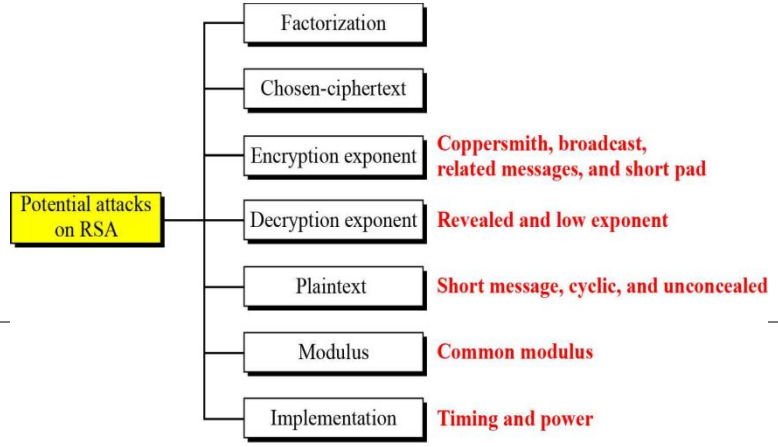
This is **the attack that attempts to find the key through the solving of the very large prime number factor problem**.

If attacker will able to know P and Q using N, then he could find out value of private key.

This can be failed when N contains atleast 300 longer digits in decimal terms, attacker will not able to find.

RAMAIAH Institute of Technology

Attacks on RSA



Chosen cipher attack:

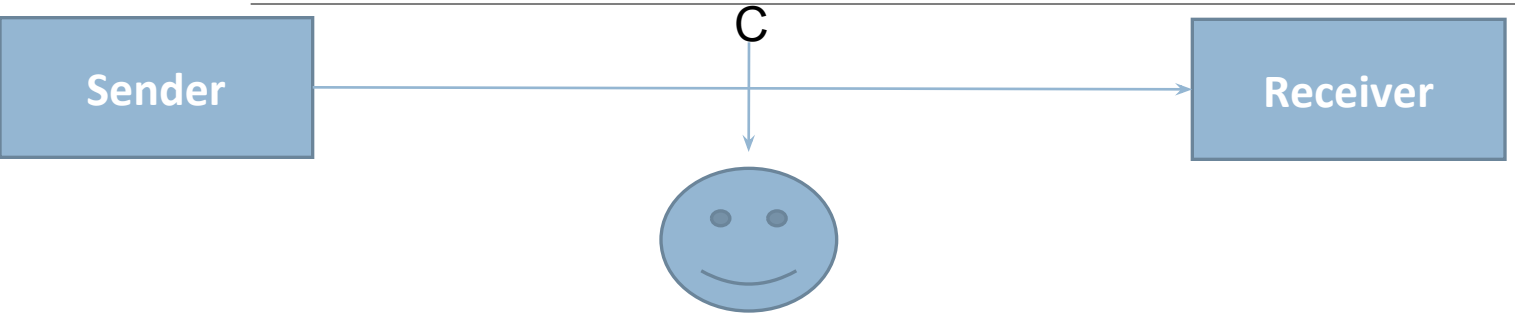
Alice creates ciphertext $C = P^e \bmod n$ and sends C to Bob. Bob will decrypt for eve

Eve intercept C and uses following steps to find P.

a. Eve chooses a random integer X in Z_n^*.
b. Eve calculates $Y = C \times X^e \bmod n$.
c. Eve sends Y to Bob for decryption and get $Z = Y^d \bmod n$; This step is an instance of a chosen-ciphertext attack.
d. Eve can easily find P because

$$Z = Y^d \bmod n = (C \times X^e)^d \bmod n = (C^d \times X^{ed}) \bmod n = (C^d \times X) \bmod n = (P \times X) \bmod n$$
$$Z = (P \times X) \bmod n \quad \rightarrow \quad P = Z \times X^{-1} \bmod n$$

Attacks on RSA



Eve Intercept C
Choose X random integer
Construct New Cipher Y = C X^{e} mod n

Receiver

$Z = Y^{d} \bmod n$

$Z = Y^{d} \bmod N$

$= (C X^{e})^{d} \bmod n$
$= C^{d} X^{e d} \bmod n$
$= C^{d} X \bmod n$
$Z = P X \bmod n$
$P = Z X^{-1} \bmod n$

a. Eve chooses a random integer X in Z_n^*.
b. Eve calculates $Y = C \times X^e \bmod n$.
c. Eve sends Y to Bob for decryption and get $Z = Y^d \bmod n$; This step is an instance of a chosen-ciphertext attack.
d. Eve can easily find P because

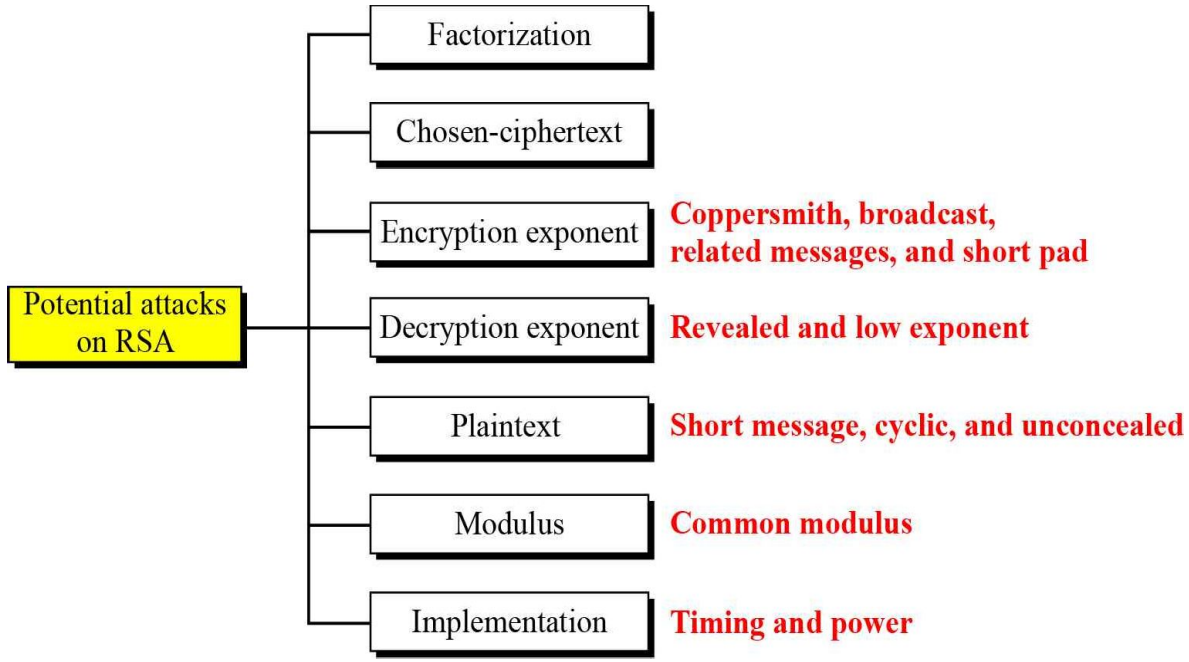
$$Z = Y^d \bmod n = (C \times X^e)^d \bmod n = (C^d \times X^{ed}) \bmod n = (C^d \times X) \bmod n = (P \times X) \bmod n$$
$$Z = (P \times X) \bmod n \rightarrow P = Z \times X^{-1} \bmod n$$

Attacks on RSA

Encryption exponent

Common attack occur when e is low, so use $e = 2^{16} + 1 = 65537$.

- Coppersmith attack
- Broadcast attack
- Related Message attack
- Short pad attack



Attacks on RSA

Coppersmith attack :

The **Coppersmith attack** is a cryptographic attack that exploits certain mathematical weaknesses in the RSA encryption algorithm and related cryptosystems.

It is particularly effective when:

- A message M is partially known.
- The RSA modulus N and the public key e are weakly structured or predictable.

Attacks on RSA

Broadcast attack

Suppose Alice wishes to send same message to three receipents with the same public key exponent e and the moduli n1,n2,n3

$C_1 = P^3 \bmod n_1$ $C_2 = P^3 \bmod n_2$ $C_3 = P^3 \bmod n_3$

$C' = P^3 \bmod n_1 n_2 n_3$.

$P^3 < n_1 n_2 n_3$.

$C' = P^{1/3}$.

- When a small encryption exponent e=3 is used and if M<N^(1/3)
- The cipher text is C=M^e mod N
- Since M<N^(1/3), mod N has no effect
- C=M^e=M^3
- M=cuberoot(C) will give us the message

Attacks on RSA

Related Message attack

- If Alice encrypt two P1 and P2 with e =3 and send C1 and C2 to Bob.
- If P1 and P2 is related by a linear function, then eve can recover P1 and p2 in a feasible computation time.

Attacks on RSA

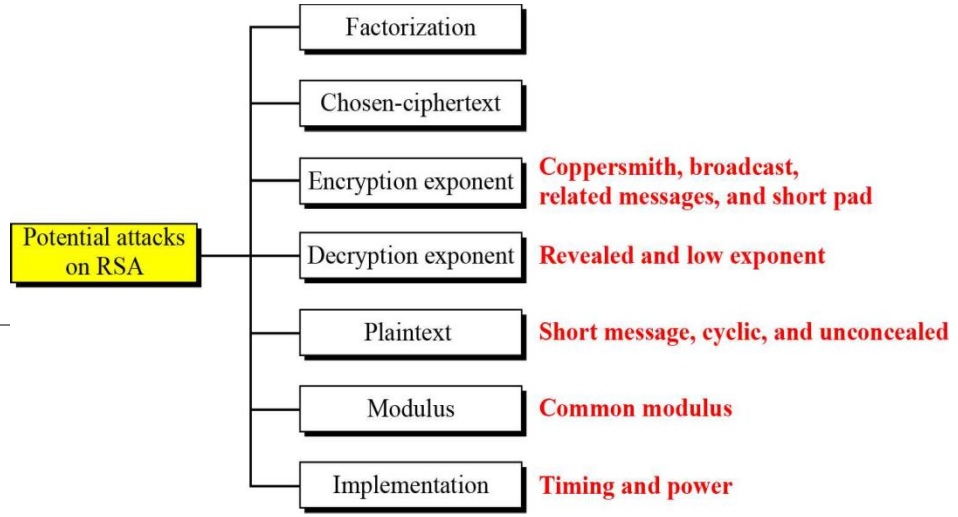
Short pad attack

- Alice has a message M to send to Bob. She pads the message with r1, encrypt and send C1 to Bob. Eve intercept C1 and drops it
- Bob inform Alice that he has not received the message, so Alice pads the message again with r2, encrypt and send to Bob.Eve also will intercept the message.
- Eve now has C1 and C2, knows both belong to same plaintext .
- If r1 and r2 are short, eve may be able to recover M

C 1	M	r1(padding)

C 2	M	r2(padding)

Attacks on RSA



Attacks on Decryption key:

Revealed decryption exponent attack:

If attacker somehow guess decryption key d, cipher text generated by encryption key is in danger, and even future messages are also in danger.

So, it is advised to take fresh values of two prime numbers (i.e; P and Q), N and E.

Low decryption exponent attack:

If we take smaller value of d in RSA this may occur, so to avoid take value of

$d = 2^{16+1}$(atleast).

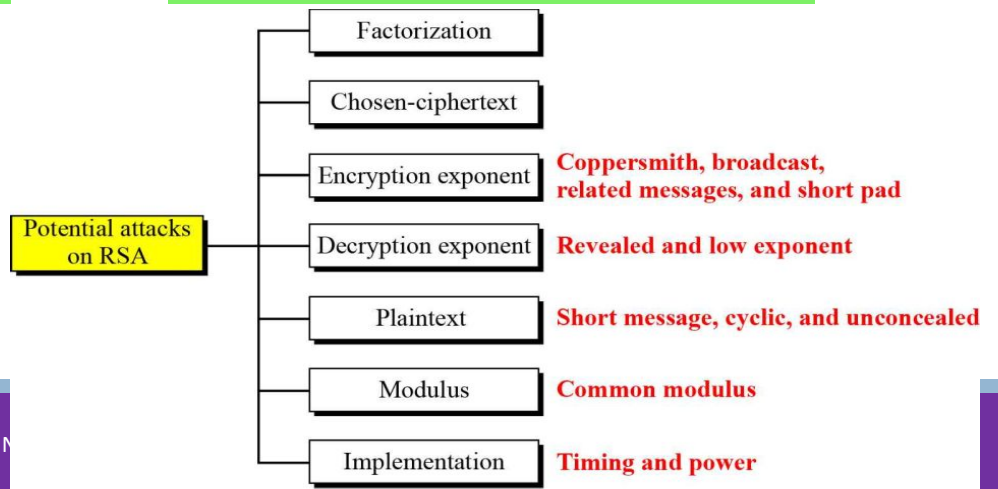
Attacks on RSA

Plain text attacks: It is classified into 3 subcategories:-

Short message attack:

Attacker knows some blocks of plain text. If this assumption is true, the attackers can try encrypting each plain-text block to view if it results into the known cipher-text.

Therefore, it can avoid this short-message attack, it is suggested that it can pad the plain text before encrypting it.



Attacks on RSA

Cycling attack:

Attacker will think that plain text is converted into cipher text using permutation.

Continuous encryption of ciphertext will eventually result in plain text. But attacker does not know the plain text. Hence will keep doing it until gets the ciphertext, goes back one step find the plain text

Intercepted ciphertext: C

$C_1 = C^e \mod n$

$C_2 = C_1^e \mod n$

...

$C_k = C_{k-1}^e \mod n \rightarrow$ If $C_k = C$, stop: the plaintext is $P = C_{k-1}$

Attacks on RSA

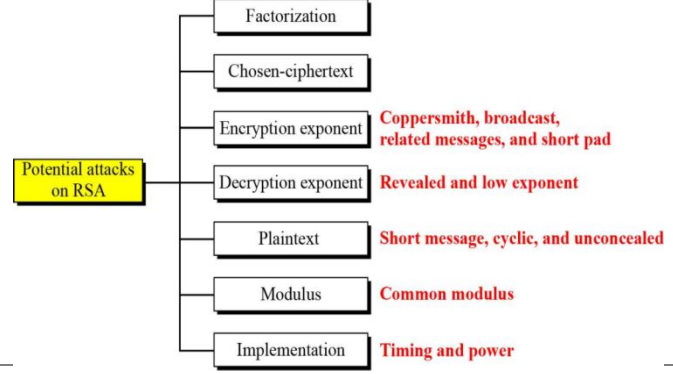
Unconcealed Message attack:

For some plain-text messages, encryption provides cipher-text which is the equal as the original plain-text.

If this appears, the original plain-text message cannot be secret.

Therefore, this attack is known as unconcealed message attack.

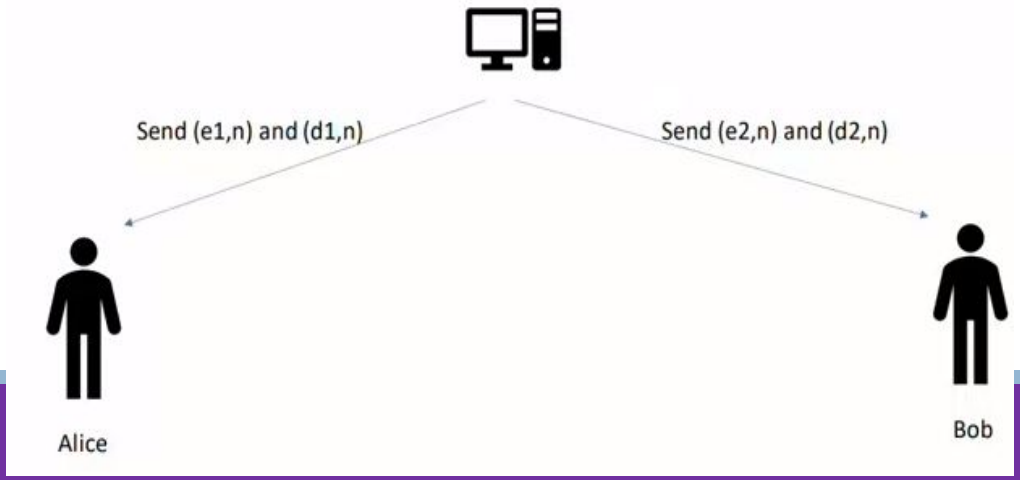
Attacks on RSA



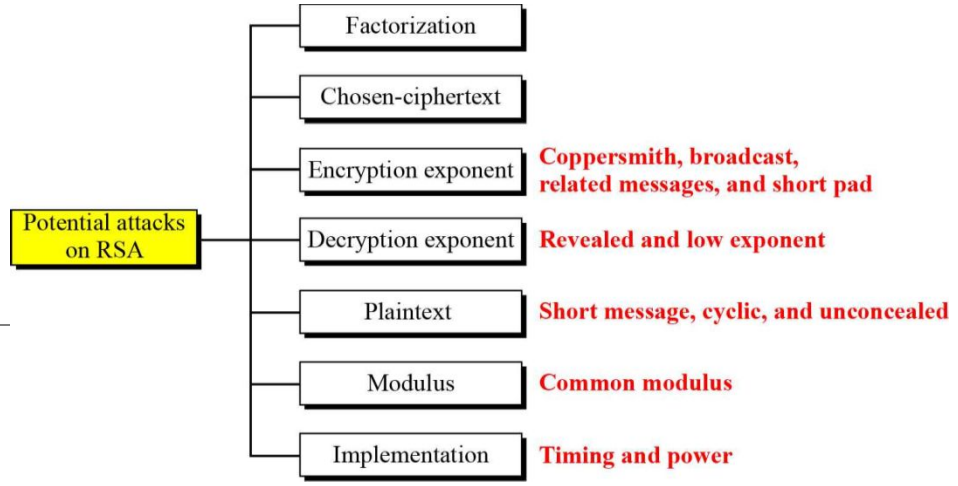
Attacks on the Modulus – Common modulus attack

If a community uses a common modulus n, select p and q, calculate n and Φ(n), and create a pair of exponents(e_i, d_i) for each entity.

The problem is eve can also decrypt the message, if he is a member of the community and assigned a pair of exponent (e_e, d_e)



Attacks on RSA

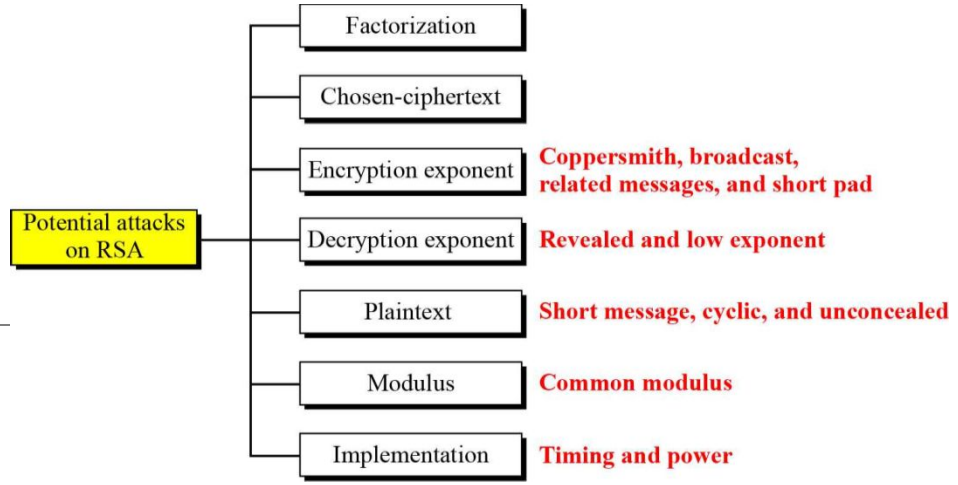


Implementation –Timing attack

- Eve intercept a large number of ciphertext C1,C2… to Cm.
- Eve observe how long it takes for the underlying hardware to calculate a multiplication operation from t1 to tm(t is time required to calculate the multiplication operation)
- The timing difference allows Eve to find the value of bits in d, one by one

RAMAIAH
Institute of Technology

Attacks on RSA

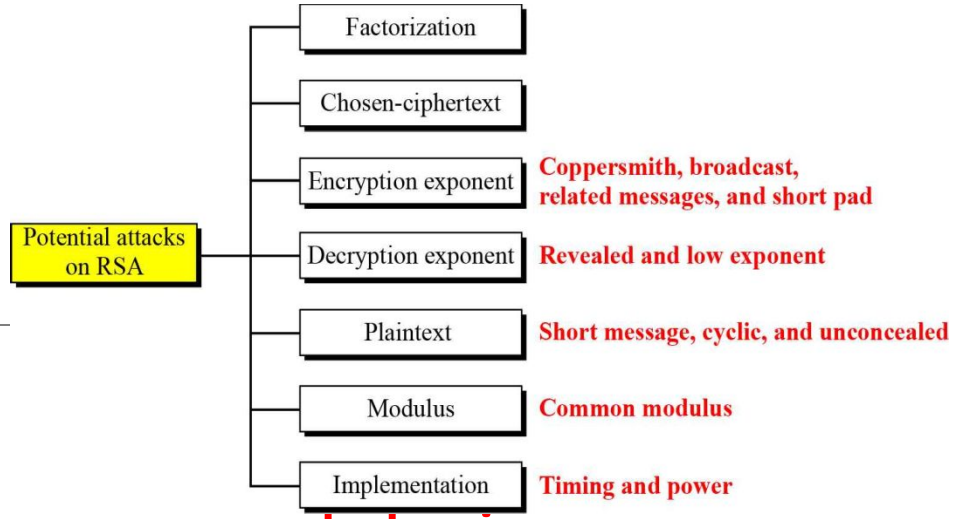


Implementation –Timing attack

There are two methods to thwart timing attack:

1. Add random delays to the exponentiations to make each exponentiation take the same amount of time.
2. Rivest recommended **blinding.** The idea is to multiply the ciphertext by a random number before decryption. The procedure is as follows:
   a. Select a secret random number r between 1 and $(n - 1)$.
   b. Calculate $C_1 = C \times r^e \bmod n$.
   c. Calculate $P_1 = C_1^d \bmod n$.
   d. Calculate $P = P_1 \times r^{-1} \bmod n$.

Attacks on RSA



Implementation – Power attack

Eve can precisely measure the power consumed during decryption, can launch power attack.

Multiplication and squaring consumes more power.

RSA Cryptosystem

- Introduction
- Procedure
- Attacks on RSA
- Optimal Asymmetric Encryption Padding (OAEP)
- Applications

Padding in RSA

- RSA without padding is also called Textbook RSA.
- RSA without padding is insecure.
- With RSA the padding is essential for its core function.
- RSA has a lot of mathematical structure, which leads to weaknesses. Using correct padding prevents those weaknesses.

Padding in RSA

Padding schemes

- PKCS#1 (Public-Key Cryptography Standards)
- Optimal Asymmetric Encryption Padding (OAEP) - It was defined by Bellare and Rogaway, and has been standardized in PKCS#1 v2 and RFC 2437.

Explain optimal asymmetric encryption padding with neat block diagram.

Optimal Asymmetric Encryption Padding (OAEP)

- Short message makes ciphertext vulnerable to short message attacks.
- Adding bogus data(padding) to the message make Eve's job harder, but with additional efforts can still attack the ciphertext.
- The solution is apply a procedure called OAEP.
- A 2048 bit RSA key allows for 256 bytes(2048*8) of which the OAEP padding takes 42 bytes, leaving around 214 bytes for encrypted data.

Key size	Key strength
512 bits	Low-strength key
1024 bits	Medium-strength key
2048 bits	High-strength key
4096 bits	Very high-strength key

Padding in RSA

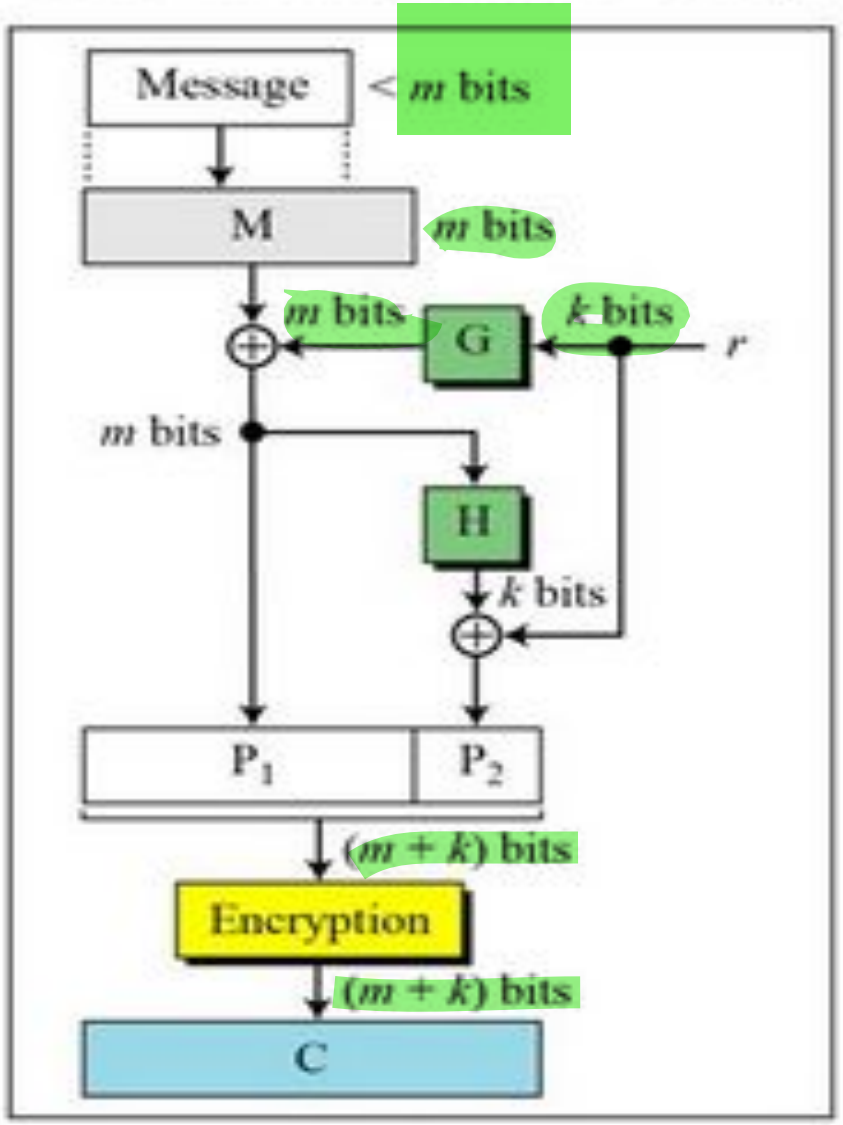
- OAEP uses a Feistel network with a pair of random oracles G and H.
- These operate on the plaintext before it is encrypted.
- Its strengths are that it adds randomness to the process.

g (OAEP)

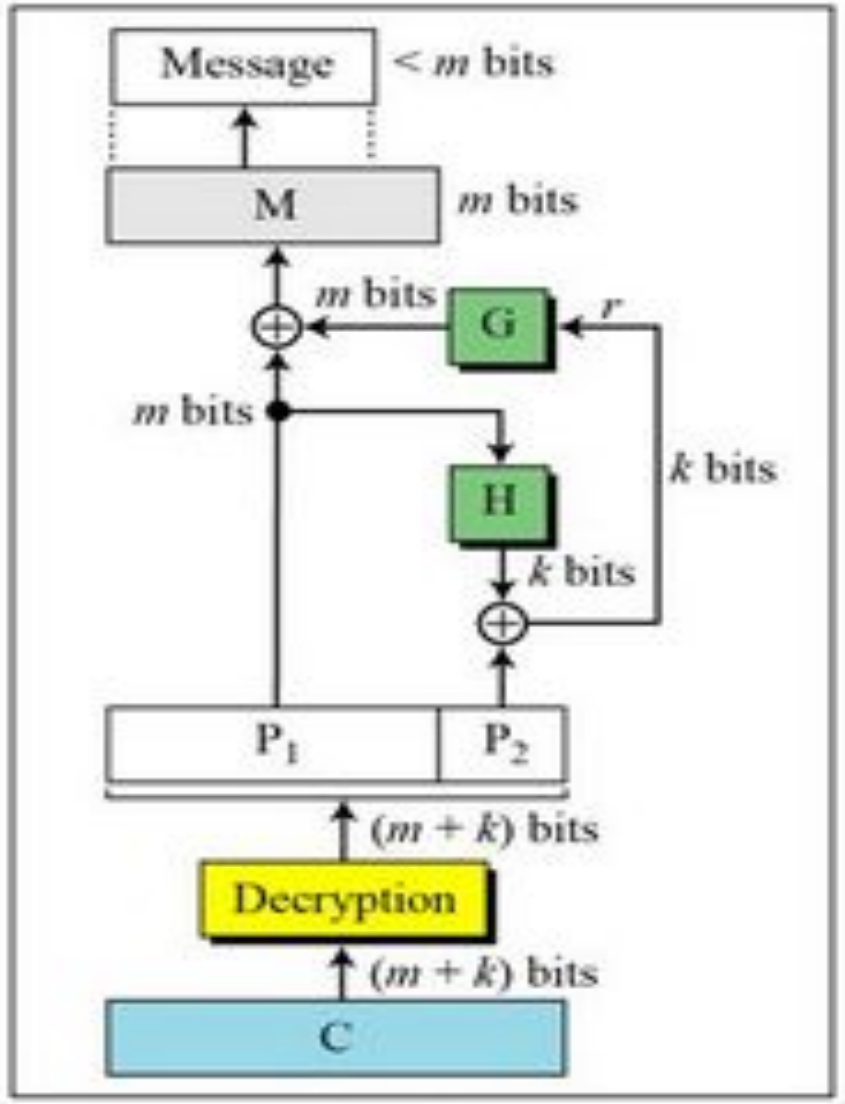
M: Padded message
r: One-time random number

P: Plaintext ($P_1 \| P_2$)
C: Ciphertext

G: Public function (k-bit to m-bit)
H: Public function (m-bit to k-bit)



Sender

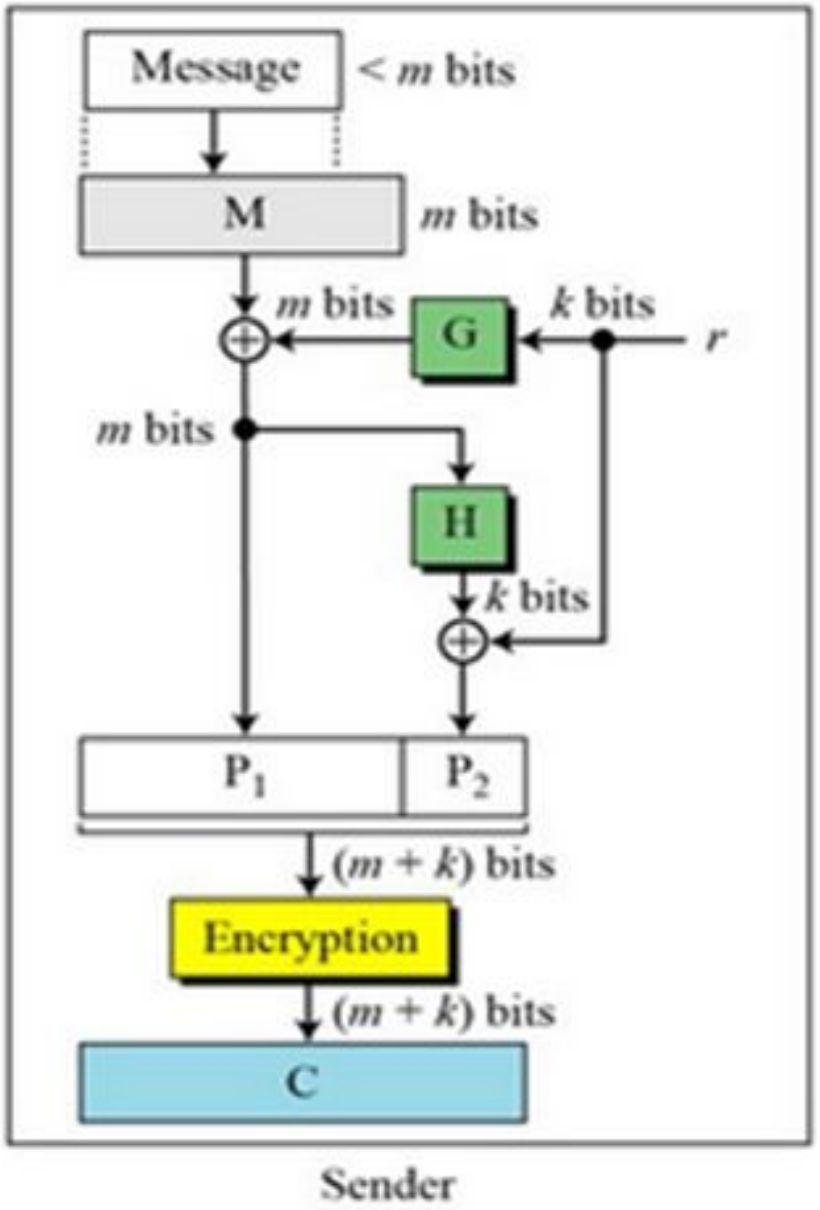


Receiver

Optimal Asymmetric Encryptic

Encryption The following shows the encryption process:

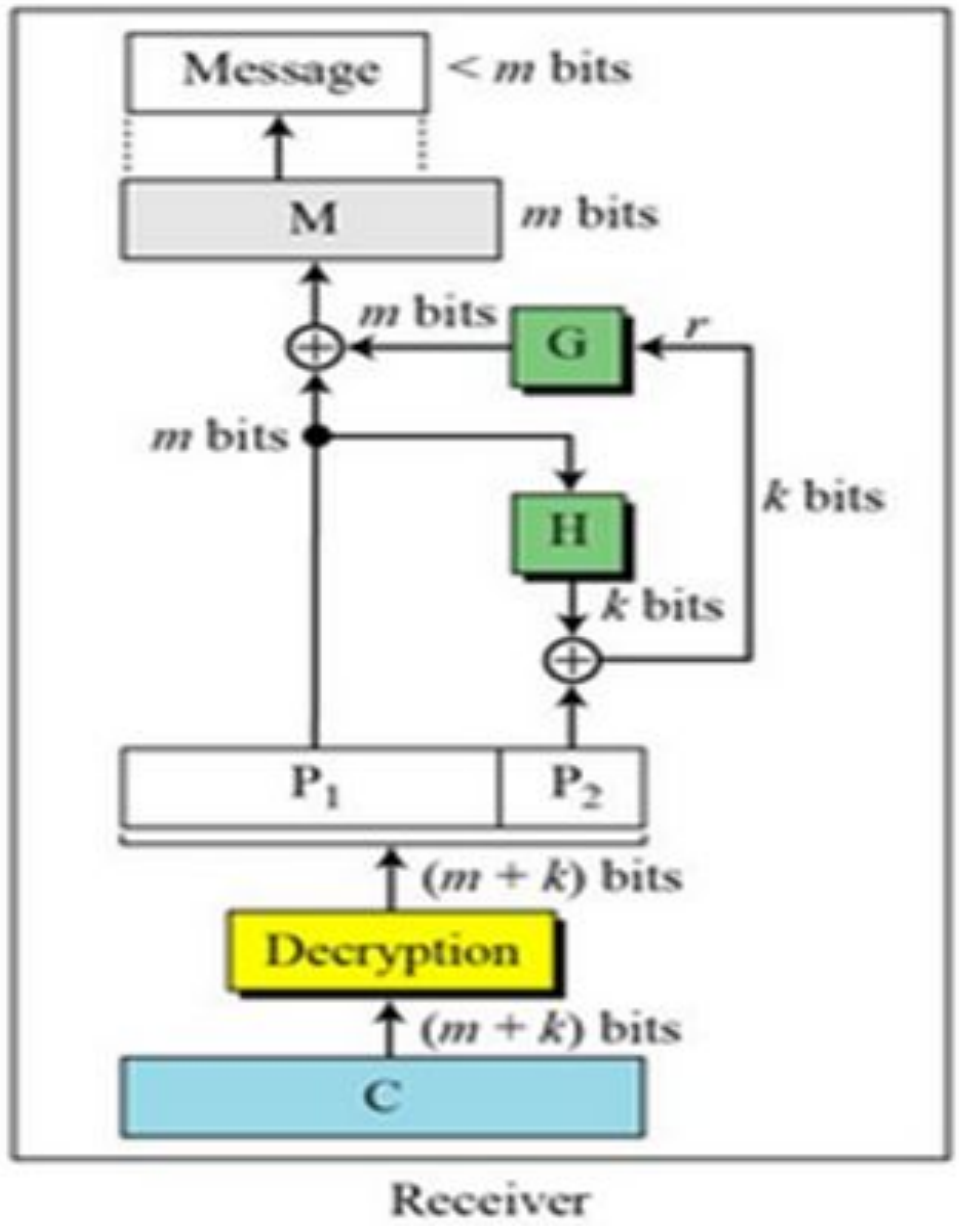
1. Alice pads the message to make an m-bit message, which we call M.
2. Alice chooses a random number r of k bits. Note that r is used only once and is then destroyed.
3. Alice uses a public one-way function, G, that takes an r-bit integer and creates an m-bit integer (m is the size of M, and $r < m$). This is the mask.
4. Alice applies the mask G(r) to create the first part of the plaintext $P_1 = M \oplus G(r)$. P_1 is the masked message.
5. Alice creates the second part of the plaintext as $P_2 = H(P_1) \oplus r$. The function H is another public function that takes an m-bit input and creates an k-bit output. This function can be a *cryptographic hash function* (see Chapter 12). P_2 is used to allow Bob to recreate the mask after decryption.
6. Alice creates $C = P^e = (P_1 \| P_2)^e$ and sends C to Bob.



Optimal Asymmetric Encryp

Decryption The following shows the decryption process:

1. Bob creates $P = C^d = (P_1 \| P_2)$.
2. Bob first recreates the value of r using $H(P_1) \oplus P_2 = H(P_1) \oplus H(P_1) \oplus r = r$.
3. Bob uses $G(r) \oplus P = G(r) \oplus G(r) \oplus M = M$ to recreate the value of the padded message.
4. After removing the padding from M, Bob finds the original message.



Applications of RSA

- RSA was used with Transport Layer Security (TLS) to secure communications among two individuals.
- Pretty Good Privacy algorithm use RSA
- Virtual Private Networks (VPNs), email services, web browsers
- Bluetooth
- MasterCard, VISA, e-banking
- e-commerce platform

RSA Cryptosystem

Disadvantages of RSA

- It may fail sometimes because for complete encryption both symmetric and asymmetric encryption is required and RSA uses symmetric encryption only.
- It has slow data transfer rate due to large numbers involved.
- It requires third party to verify the reliability of public keys sometimes.
- High processing is required at receiver's end for decryption.

Outline

Encipherment using Modern Symmetric-Key Ciphers: (Text 1: Chapter 8)

- Use of Modern Block Ciphers
- Use of Stream Ciphers
- Other Issues.

Asymmetric Key Cryptography: (Text1: Chapter 10)

- Introduction
- RSA Cryptosystem
- Rabin Cryptosystem
- Elgamal Cryptosystem

Draw the block diagram for encryption, decryption and key generation for Rabin cryptosystem.

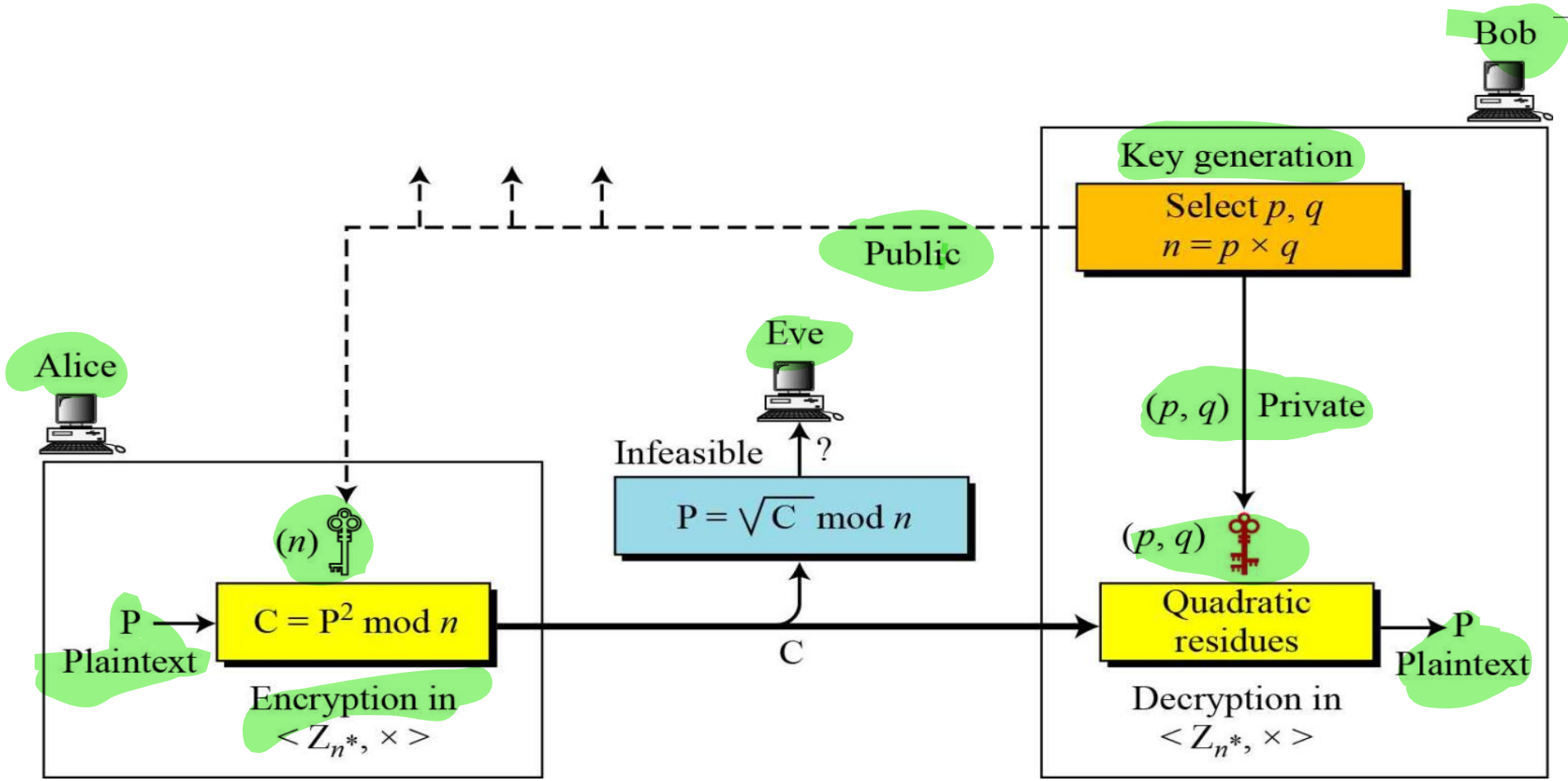
RABIN CRYPTOSYSTEM

- Rabin Cryptosystem is an public-key cryptosystem invented by Michael Rabin.
- In Rabin cryptosystem, value of e = 2 and d= 1/2 is fixed.
- Rabin is based on quadratic congruence

The encryption is $C \equiv P^2 \pmod n$ and the decryption is $P \equiv C^{1/2} \pmod n$.

- Public key is n
- Private key is tuple(p, q)
- Everyone can encrypt using n and only receiver can decrypt using p and q

RABIN CRYPTOSYSTEM



RABIN CRYPTOSYSTEM

Algorithm 10.6 *Key generation for Rabin cryptosystem*

```
Rabin_Key_Generation
{
    Choose two large primes p and q in the form 4k + 3 and p ≠ q.
    n ← p × q
    Public_key ← n                          // To be announced publicly
    Private_key ← (q, n)                    // To be kept secret
    return Public_key and Private_key
}
```

Algorithm 10.7 *Encryption in Rabin cryptosystem*

```
Rabin_Encryption (n, P)          // n is the public key; P is the ciphertext from Z_n*
{
    C ← P^2 mod n                // C is the ciphertext
    return C
}
```

RABIN CRYPTOSYSTEM

Algorithm 10.8 *Decryption in Rabin cryptosystem*

Rabin_Decryption (p, q, C) // C is the ciphertext; p and q are private keys

{

$a_1 \leftarrow +(C^{(p+1)/4}) \bmod p$

$a_2 \leftarrow -(C^{(p+1)/4}) \bmod p$

$b_1 \leftarrow +(C^{(q+1)/4}) \bmod q$

$b_2 \leftarrow -(C^{(q+1)/4}) \bmod q$

// The algorithm for the Chinese remainder algorithm is called four times.

$P_1 \leftarrow$ Chinese_Remainder (a_1, b_1, p, q)

$P_2 \leftarrow$ Chinese_Remainder (a_1, b_2, p, q)

$P_3 \leftarrow$ Chinese_Remainder (a_2, b_1, p, q)

$P_4 \leftarrow$ Chinese_Remainder (a_2, b_2, p, q)

return P_1, P_2, P_3, and P_4

}

RABIN CRYPTOSYSTEM

Step	Alice	Erich	Bob
1			chooses two large random primes, p and q with $p \equiv q \equiv 3 \bmod 4$ and $p \neq q$, keeps them secret, and computes his public key $n = pq$
2		$\Leftarrow n$	
3	encrypts the message m by $c = m^2 \bmod n$		
4		$c \Rightarrow$	
5			decrypts c by computing $m = \sqrt{c} \bmod n$

Chinese Remainder theorem

CRT is used to solve a set of different congruent equations with one variable but different moduli which are relatively prime

$$X \equiv a_1 \pmod{m_1}$$
$$X \equiv a_2 \pmod{m_2}$$
$$\ldots$$
$$X \equiv a_n \pmod{m_n}$$

$$X = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + \ldots + a_n M_n M_n^{-1}) \bmod M$$

CRT states that the above equation have a unique solution of the moduli are relatively prime

Chinese Remainder theorem

$X \equiv 2 \pmod 3$

$X \equiv 3 \pmod 5$

$X \equiv 2 \pmod 7$

$X \equiv a_1 \pmod{m_1}$

$X \equiv a_2 \pmod{m_2}$

$X \equiv a_3 \pmod{m_3}$

$$X = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + a_3 M_3 M_3^{-1}) \bmod M$$

$$X = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + \ldots + a_n M_n M_n^{-1}) \bmod M$$

Given		To Find		
$a_1 = 2$	$m_1 = 3$	M_1	M_1^{-1}	M
$a_2 = 3$	$m_2 = 5$	M_2	M_2^{-1}	
$a_3 = 2$	$m_3 = 7$	M_3	M_3^{-1}	

$M = m_1 \times m_2 \times m_3$

$M = 3 \times 5 \times 7$

$M = 105$

Chinese Remainder theorem

Given		To Find		
$a_1 = 2$	$m_1 = 3$	$M_1 =$	M_1^{-1}	
$a_2 = 3$	$m_2 = 5$	$M_2 =$	M_2^{-1}	$M=105$
$a_3 = 2$	$m_3 = 7$	$M_3 =$	M_3^{-1}	

Chinese Remainder theorem

Given		To Find		
$a_1 = 2$	$m_1 = 3$	$M_1 = 35$	$M_1^{-1} = 2$	M=105
$a_2 = 3$	$m_2 = 5$	$M_2 = 21$	$M_2^{-1} = 1$	
$a_3 = 2$	$m_3 = 7$	$M_3 = 15$	$M_3^{-1} = 1$	

$$X = (a_1M_1 M_1^{-1} + a_2M_2M_2^{-1} + a_3M_3M_3^{-1}) \bmod M$$
$$= (2x35x2 + 3x21x1 + 2x15x1) \bmod 105$$
$$= 233 \bmod 105$$
$$X = 23$$

- $N_1 = \frac{105}{3} = 35,$
- $N_2 = \frac{105}{5} = 21,$
- $N_3 = \frac{105}{7} = 15.$

Mi = M / M i

RAMAIAH
Institute of Technology

RABIN CRYPTOSYSTEM

1. **Bob selects $p = 23$ and $q = 7$. Note that both are congruent to $3 \bmod 4$. p and q are in the form 4k+3 and p not equal to q**
2. **Bob calculates $n = p \times q = 161$.**
3. **Bob announces n publicly; he keeps p and q private.**
4. **Alice wants to send the plaintext $P = 24$.**

   Note that 161 and 24 are relatively prime; 24 is in Z_{161}^*.

   She calculates $C = 24^2 = 93 \bmod 161$,

   and sends the ciphertext 93 to Bob.

RAMAIAH
Institute of Technology

RABIN CRYPTOSYSTEM

5. **Bob receives 93 and calculates four values:**

$a_1 = +(93^{(23+1)/4}) \bmod 23 = +(93^6) \bmod 23 = 1 \bmod 23$

$a_2 = -(93^{(23+1)/4}) \bmod 23 = -(93^6) \bmod 23 = -1 \bmod 23 \equiv -1+23 \bmod 23 = 22 \bmod 23$

$b_1 = +(93^{(7+1)/4}) \bmod 7 = +(93^2) \bmod 7 = 4 \bmod 7$

$b_2 = -(93^{(7+1)/4}) \bmod 7 = -(93^2) \bmod 7 = -4 \bmod 7 \equiv -4+7 \bmod 7 = 3 \bmod 7$

6. **Bob takes four possible answers, (a_1, b_1), (a_1, b_2), (a_2, b_1), and (a_2, b_2), and uses the Chinese remainder theorem to find four possible plaintexts: 116, 24, 137, and 45.**

Note that only the second answer is Alice's plaintext.

RABIN CRYPTOSYSTEM

Algorithm 10.8 *Decryption in Rabin cryptosystem*

Rabin_Decryption (p, q, C) // C is the ciphertext; p and q are private keys

{

$a_1 \leftarrow +(C^{(p+1)/4}) \bmod p$

$a_2 \leftarrow -(C^{(p+1)/4}) \bmod p$

$b_1 \leftarrow +(C^{(q+1)/4}) \bmod q$

$b_2 \leftarrow -(C^{(q+1)/4}) \bmod q$

// The algorithm for the Chinese remainder algorithm is called four times.

$P_1 \leftarrow$ Chinese_Remainder (a_1, b_1, p, q)

$P_2 \leftarrow$ Chinese_Remainder (a_1, b_2, p, q)

$P_3 \leftarrow$ Chinese_Remainder (a_2, b_1, p, q)

$P_4 \leftarrow$ Chinese_Remainder (a_2, b_2, p, q)

return P_1, P_2, P_3, and P_4

}

RABIN CRYPTOSYSTEM

6. Bob takes four possible answers, (a_1, b_1) uses the Chinese remainder theorem

$a_1 = 1 \bmod 23$

$b_1 = 4 \bmod 7$

$X = [1 * 7 (7^{-1} \bmod 23) + 4 * 23 (23^{-1} \bmod 7)] \bmod 161$

$= [1 *7 (10) + 4 *23 (4)]$

$= [70 + 368] \bmod 161$

$= 116$

RABIN CRYPTOSYSTEM

6. Bob takes four possible answers, (a_1, b_2) uses the Chinese remainder theorem

$a_1 = 1 \bmod 23$

$b_1 = 3 \bmod 7$

$X = [1 * 7 (7^{-1} \bmod 23) + 3 * 23 (23^{-1} \bmod 7)] \bmod 161$

$= [1 *7 (10) + 3 *23 (4)]$

$= [70 + 276] \bmod 161$

$= 24$

RABIN CRYPTOSYSTEM

6. **Bob takes four possible answers, (a_1, b_1), (a_1, b_2), (a_2, b_1), and (a_2, b_2), and uses the Chinese remainder theorem to find four possible plaintexts:** 116, 24, 137, **and** 45.

Note that only the second answer is Alice's plaintext.

Security of the RABIN CRYPTOSYSTEM

- The Rabin system is secure as long as p and q are large numbers.
- The complexity of the Rabin system is at the same level as factoring a large number n into its two prime factors p and q. (Rabin system is as secure as RSA)

RABIN CRYPTOSYSTEM

- Suppose Alice wants to send message to Bob
- Bob(Receiver) chooses the prime numbers p = 43 and q = 47

  Note that 43 ≡ 47 ≡ 3 mod 4
  - n = p* q = 2021
- To encrypt the message m = 741, Alice(Sender) computes

  $C = 741^2 \mod 2021 = 549081 \mod 2021 = 1390$

  sends c = 1390 to Bob

Illustrate ElGamal Encryption and decryption algorithm.

ELAGAMAL CRYPTOSYSTEM

- The ElGamal cryptosystem is a public key encryption algorithm invented by Taher Elgamal in 1985 that is based on the Diffie-Hellman key exchange.

- It can be considered the asymmetric algorithm where the encryption and decryption happen by using public and private keys.

Diffie Hellman key exchange

- The Diffie-Hellman key agreement protocol was the first practical method for establishing a shared secret over an unsecured communication channel.
- The point is to agree on a key that two parties can use for a symmetric encryption, in such a way that an eavesdropper cannot obtain the key.

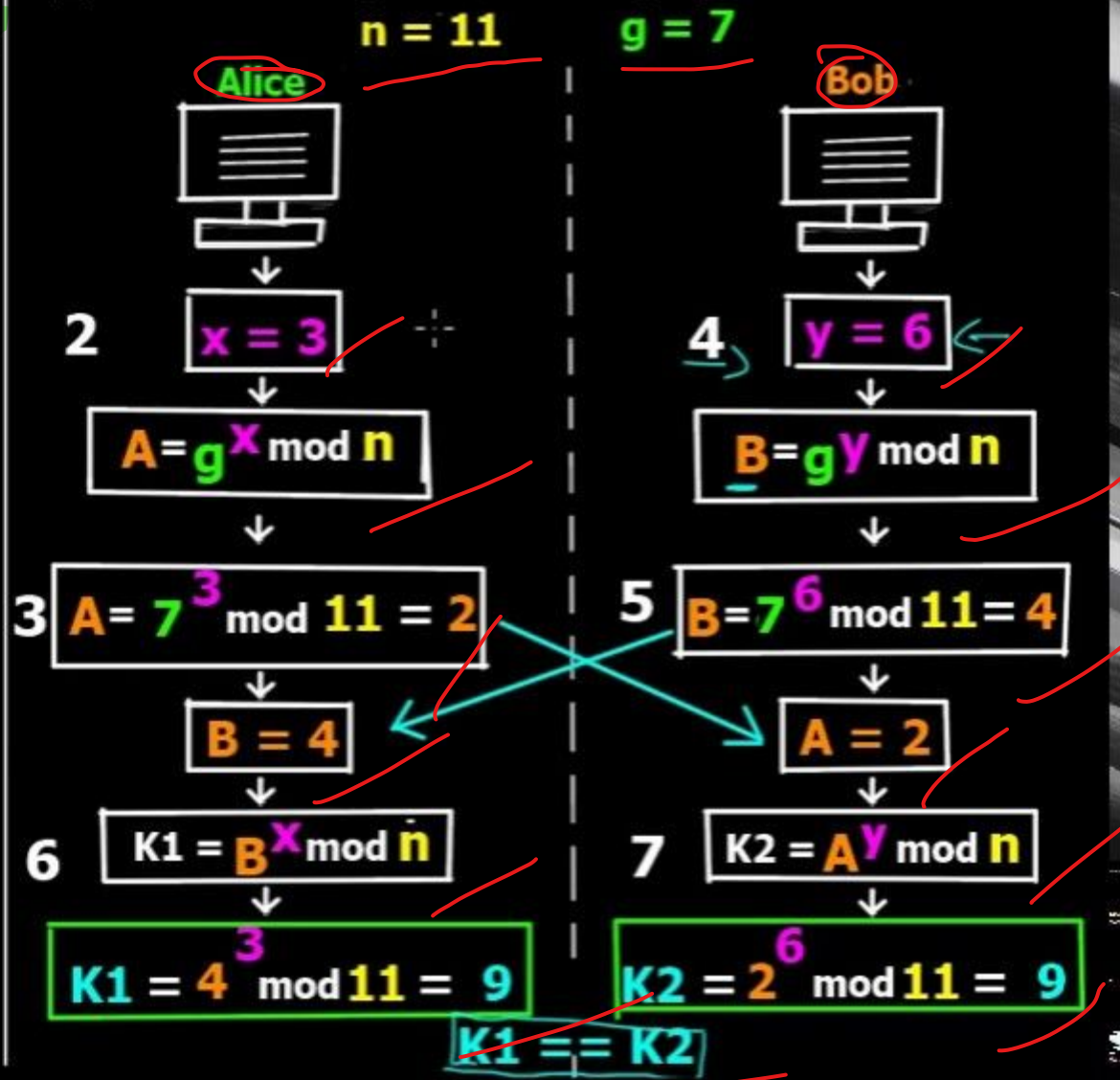
Diffie-Hellman Key Exchange Agreement/Algorithm

Diffie-Hellman Key Exchange/Agreement Algorithm

>> Two parties, can agree on a symmetric key using this technique.
>> This can then be used for encryption/ decryption.
>> This algorithm can be used only for key agreement, but not for encryption or decryption.
>> It is based on mathematical principles.

Algorithm -

1. Firstly Alice & Bob agree upon 2 large prime numbers - n & g
   These 2 numbers need not be secret & can be shared publicly.
2. Alice chooses another large random number x(private to her) & calcuates A such that : $A = g^x \bmod n$
3. Alice sends this to Bob.
4. Bob chooses another large random number y(private to him) & calcuates B such that : $B = g^y \bmod n$
5. Bob sends this to Alice.
6. Alice now computes her secret key K1 as follows:
   $K1 = B^x \bmod n$
7. Bob computes his secret key K2 as follows:
   $K2 = A^y \bmod n$
8. K1 = K2 (key exchange complete)



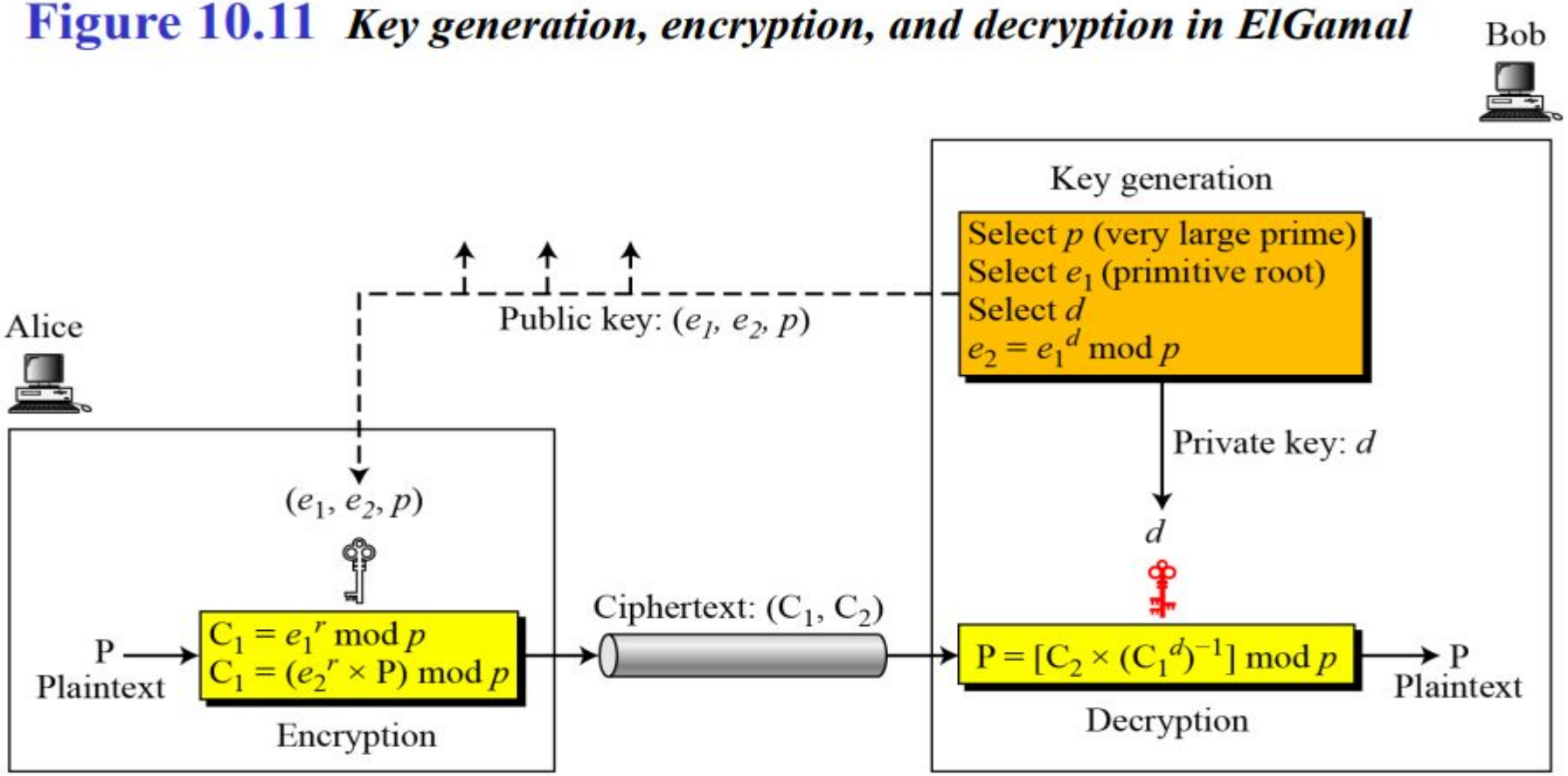
Diffie-Hellman Example

- users Alice & Bob who wish to swap keys:
- agree on prime $q=353$ and $a=3$
- select random secret keys:
  - A chooses $x_A=97$, B chooses $x_B=233$
- compute respective public keys:
  - $y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $y_B = 3^{233} \bmod 353 = 248$ (Bob)
- compute shared session key as:
  - $K_{AB} = y_B^{x_A} \bmod 353 = 248^{97} = 160$ (Alice)
  - $K_{AB} = y_A^{x_B} \bmod 353 = 40^{233} = 160$ (Bob)

Illustrate ElGamal Encryption and decryption algorithm.

ELAGAMAL CRYPTOSYSTEM

Figure 10.11 ***Key generation, encryption, and decryption in ElGamal***



ELAGAMAL CRYPTOSYSTEM

ElGamal_Key_Generation

{

Select a large prime p

Select d to be a member of the group $\mathbf{G} = < \mathbf{Z}_p^*, \times >$ such that $1 \leq d \leq p - 2$

Select e_1 to be a primitive root in the group $\mathbf{G} = < \mathbf{Z}_p^*, \times >$

$e_2 \leftarrow e_1^{\,d} \bmod p$

Public_key $\leftarrow (e_1, e_2, p)$ // To be announced publicly

Private_key $\leftarrow d$ // To be kept secret

return Public_key and Private_key

}

ELAGAMAL CRYPTOSYSTEM

ElGamal_Encryption $(e_1, e_2, p, \mathrm{P})$ // P is the plaintext

{

Select a random integer r in the group $\mathbf{G} = < \mathbf{Z}_p^*, \times >$

$C_1 \leftarrow e_1^{\,r} \bmod p$

$C_2 \leftarrow (\mathrm{P} \times e_2^{\,r}) \bmod p$ // C_1 and C_2 are the ciphertexts

return C_1 and C_2

}

ELAGAMAL CRYPTOSYSTEM

ElGamal_Decryption (d, p, C_1, C_2) // C_1 and C_2 are the ciphertexts

{

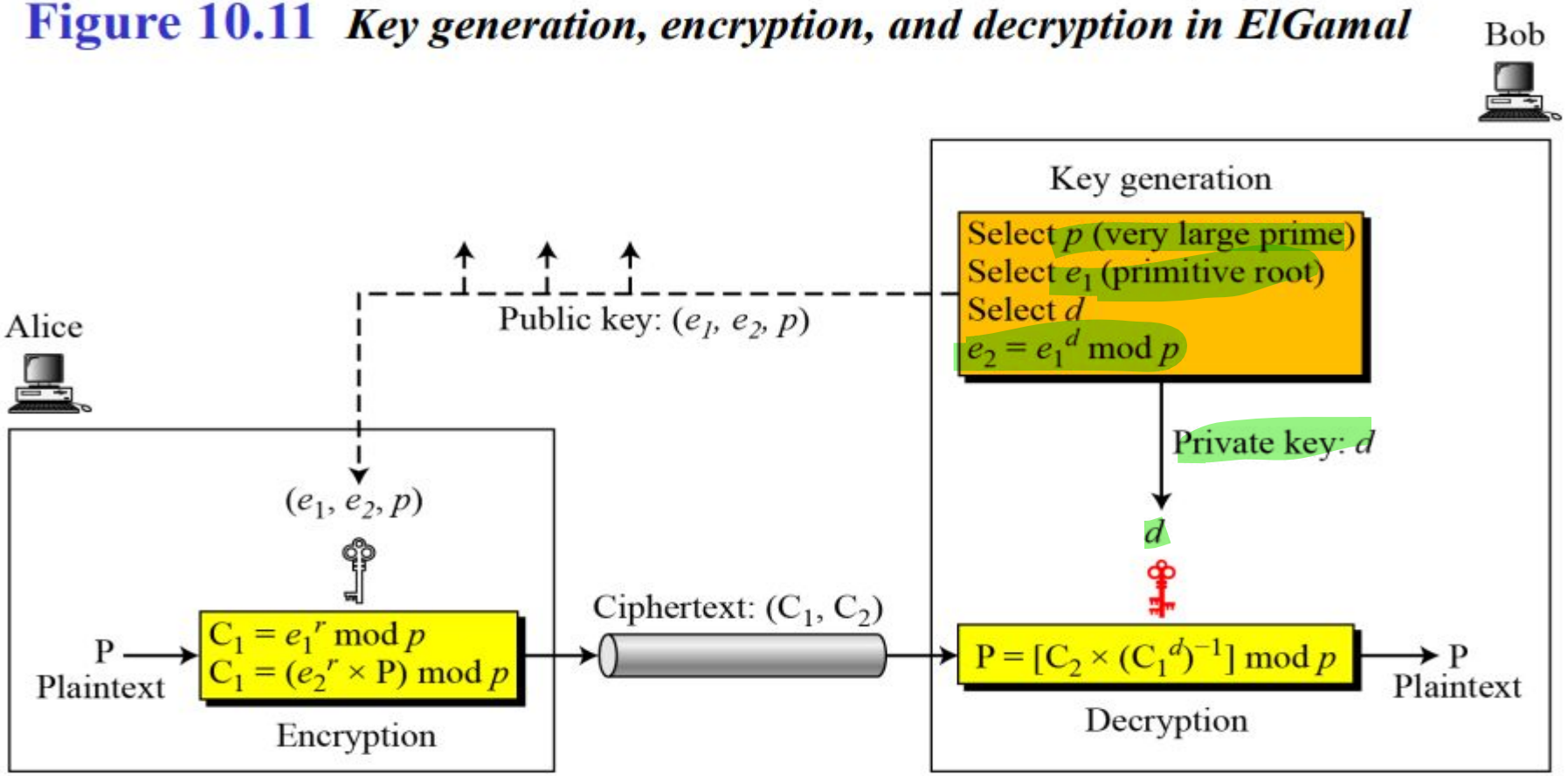
$P \leftarrow [C_2 (C_1^d)^{-1}] \bmod p$ // P is the plaintext

return P

}

ELAGAMAL CRYPTOSYSTEM

Figure 10.11 ***Key generation, encryption, and decryption in ElGamal***



ELAGAMAL CRYPTOSYSTEM

- Bob(receiver) chooses p = 11 and e1 = 2, and d = 3 , e2 = $e1^{d}$ = 23mod 11= 8.
- So the public keys are (2, 8, 11) and the private key is 3.
- Alice(sender)chooses r = 4 and calculates C1 and C2 for the plaintext 7.

Plaintext: 7

$C_1 = e_1{}^r \bmod 11 = 16 \bmod 11 = 5 \bmod 11$

$C_2 = (P \times e_2{}^r) \bmod 11 = (7 \times 4096) \bmod 11 = 6 \bmod 11$

Ciphertext: (5, 6)

ELAGAMAL CRYPTOSYSTEM

Bob receives the ciphertexts (5 and 6) and calculates the plaintext.

$$[C_2 \times (C_1{}^d)^{-1}] \bmod 11 = 6 \times (5^3)^{-1} \bmod 11 = 6 \times 3 \bmod 11 = 7 \bmod 11$$

Plaintext: 7

$= 6 * (125)^{-1} \bmod 11$

$= 6 * 3 \bmod 11$

$= 18 \bmod 11$

$= 7$

ELAGAMAL CRYPTOSYSTEM

- Bob(receiver) chooses p = 19 and e1 = 10, and d = 5 , e2 = ?
- Alice(sender)chooses r = 6 and calculates C1 and C2 for the plaintext 17

ELAGAMAL CRYPTOSYSTEM

Security of Elagamal Cryptosystem

Low Modulus attack

Value of p should be large enough(atleast 1024 bits)

ELAGAMAL CRYPTOSYSTEM

Security of Elagamal Cryptosystem

Known plaintext attack

- If Alice uses the same r to encrypt P and P'.
- Eve discover P' if she knows P.
- Assume $C_2 = P * e_2^r \bmod p$ and $C'_2 = P' * e_2^r \bmod p$
- Eve can find P' using the following steps

1. $(e_2^r) = C_2 \times P^{-1} \bmod p$
2. $P' = C'_2 \times (e_2^r)^{-1} \bmod p$

THANK YOU