Computer and Network Security: Symmetric Encryption

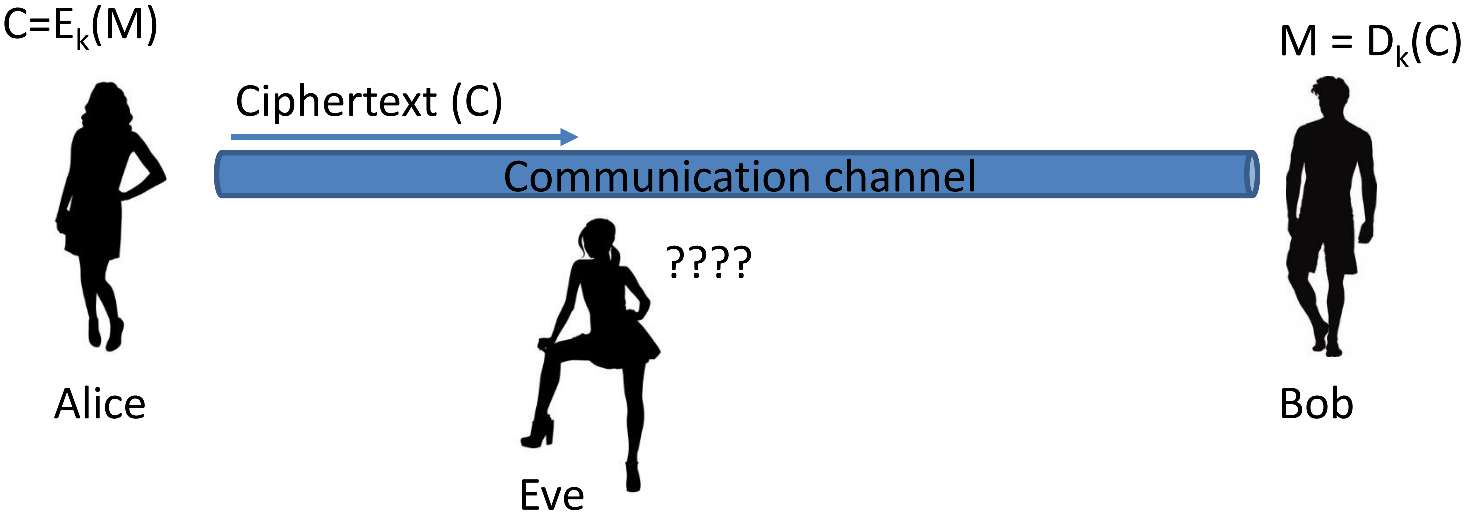
Kameswari Chebrolu

Outline

- **Modern Cryptography**
  - Overview
  - **Confidentiality**
    - Background: Definition, Crypto-analysis, One Time Pads
    - **Symmetric key encryption,** Block modes
    - Asymmetric key encryption
  - Integrity (includes Authentication)
    - Hashes, MAC, Digital signature

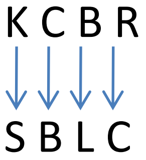
Recap: Symmetric Key Algorithms

- Alice and Bob share a key k (how?)
- Eve does not know the key but knows the encryption/decryption algorithm

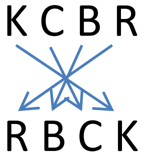


Recap: Confusion and Diffusion

- Confusion: Transform information in plaintext so that it is not easy to extract
  - Hide plaintext symbols
  - Achieved by substitution



- Diffusion: Spread information from a region of plaintext much wider in cipher text
  - Achieved by transposition



- Symmetric ciphers use a combination of both

Types

- Stream Cipher: Operate on a stream of plain/cipher text, one symbol (e.g. byte) at a time
  - E.g. Simple substitution
- Block Cipher: Operate on a block (e.g. 64 or 128 bits) of plain/cipher text
  - E.g. Transposition cipher
- Most modern symmetric algorithms are block ciphers

Pros and Cons

Stream

\+ Fast encryption

\+ Low error propagation

\- Low diffusion

\- Susceptible to tampering (insertion)

Block

\+ High diffusion

\+ Immunity to tampering (easy detection)

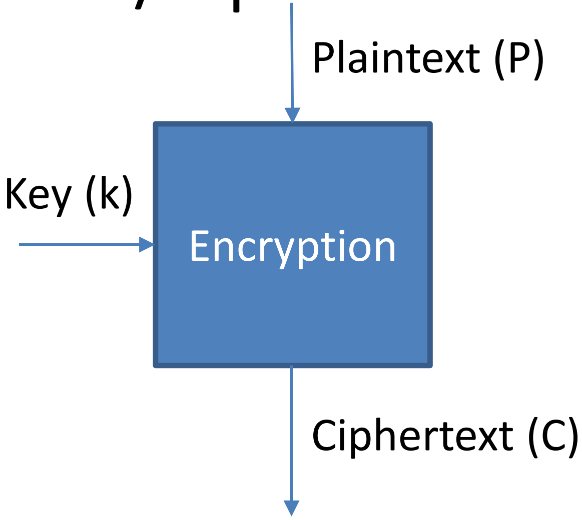
\- Slow encryption

\- Error propagation

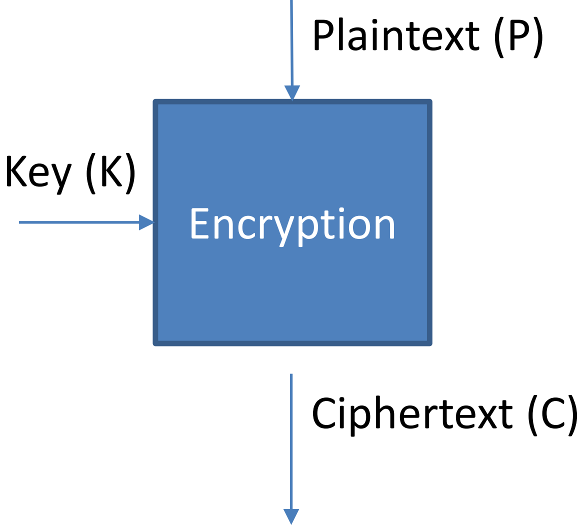
For now focus on Block Ciphers

Block Encryption

- Block length short
  - Can construct a table to decrypt (plaintext / ciphertext pairs)
- Key length short
  - Can search through all keys
- Block/Key length too long
  - Inconvenient, performance penalties



- Block length (n): 64 to 128 bits adequate
- Key length (k): 128 to 256 bits adequate
- DES: |P|=|C|= 64 bits, |K|= 56 bits
- AES: |P|=|C|= 128 bits, |K|= 128, 192 or 256 bits
- What it plain text exceeds block size?
  - Covered in block modes



Property-1

- **Correctness:** For a given key, one-on-one mapping between plaintext and cipher text ($\{0,1\}^n \rightarrow \{0,1,\}^n$)
  - Two or more plaintexts cannot map to same cipher text. Why?
  - Can a plain text map to two or more ciphertext?
    - No. The output space is same as input space.

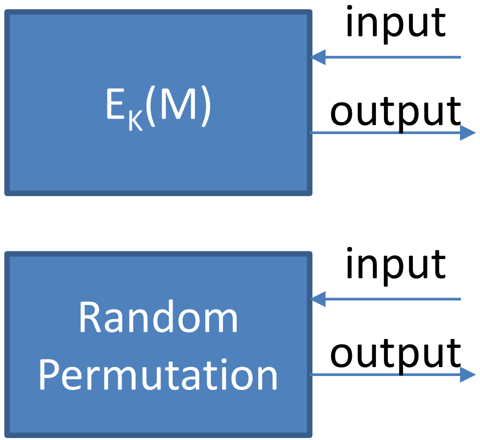
Property-2

- **Efficiency:** Both encryption and decryption should be fast (polynomial time)

Property-3

- **Secure:** Encryption should look as if mapping between input and output generated by a random permutation

Two unknown boxes



?????
Which one is Encrypting ?

Can give input to each box an examine output

Must guess which is the encryption box?

Pr[winning] <= ½ + negligible e

- Each output should have about half the bits as 1
- A particular bit in a large set of outputs should be 1 half the time
- Similar inputs should produce uncorrelated outputs
  - 1 bit change in input, half of output bits changed
- How does one achieve this?

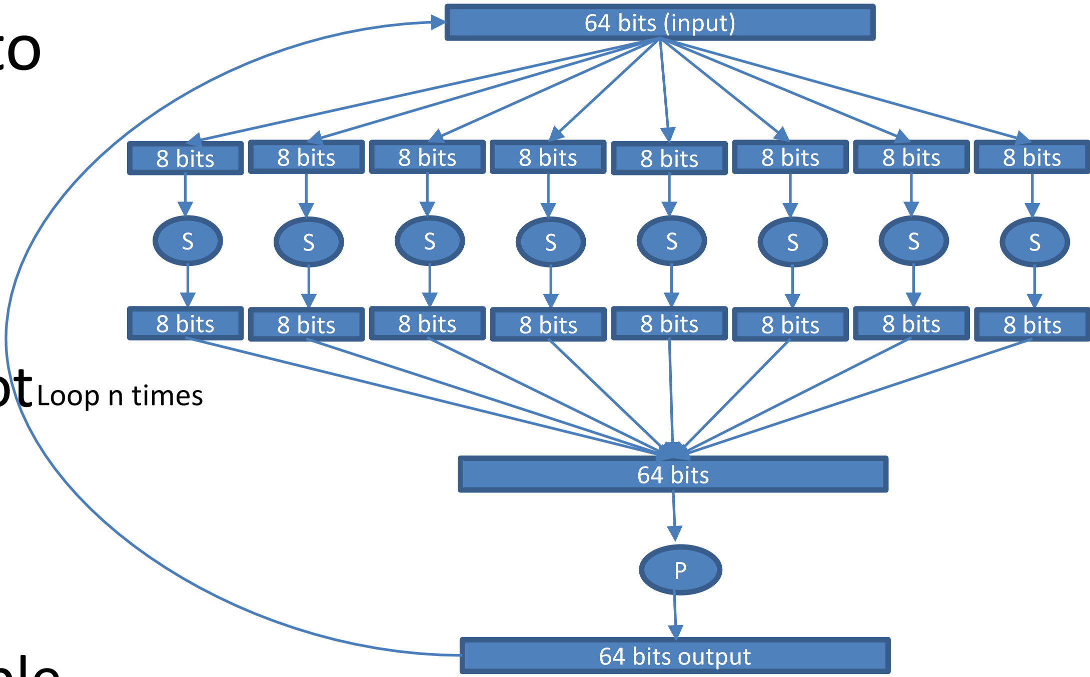
Naive Approach

Suppose input block was 64 bits,

- 2^{64} input values map to 2^{64} output values (one-to-one)
- 2^{64} ! Mappings possible → astronomical for brute force --> good security
- Table: ~ (2^{64} * 64 bits = 2^{70} bits) → key
  - – Not practical (how to convey? how to store?)

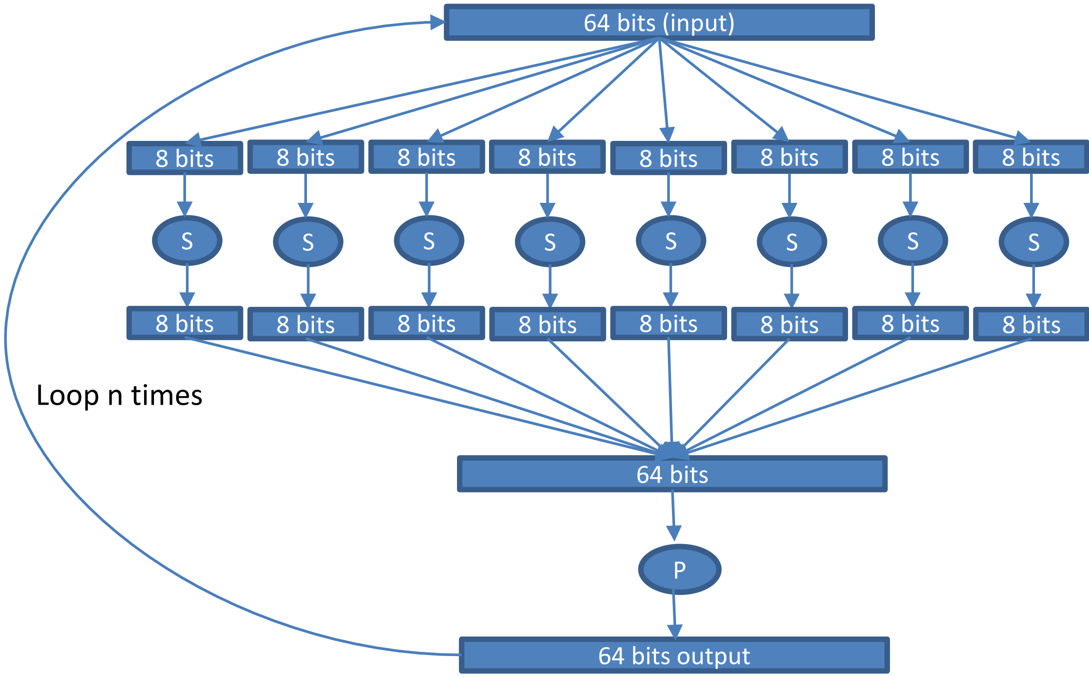
Practical Implementation

- Divide the 64 bit input into smaller chunks of 8 bits
- **Confusion** via S-box: implements substitution
  - Substitution table may not be a function of key (e.g. DES, AES)
  - Popular standards use functions in place of a table
  - Table: 8 * 2^8 bits (more manageable)



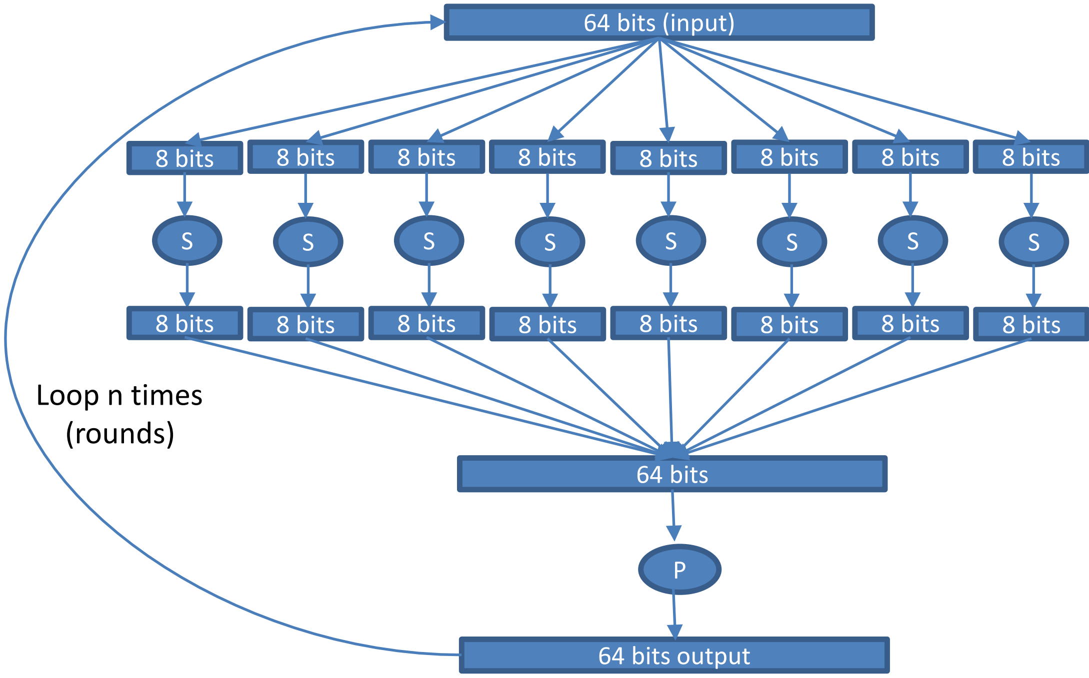
Practical Implementation

- **Diffusion** via P-box: implements Permutation
  - Scrambles the bits
  - Permutation also may not depend on the key
  - Requires 64 $\log_2 64$ bits
- Role of key?
  - Derivative of key often xor'ed with input or output bits of a box



Practical Implementation

- Rounds: Ensure each plaintext input bit affects most ciphertext output bits
  - Diffuses better
  - Only one round: an input bit affects only 8 output bits
  - What is the best number of rounds?



A good implementation

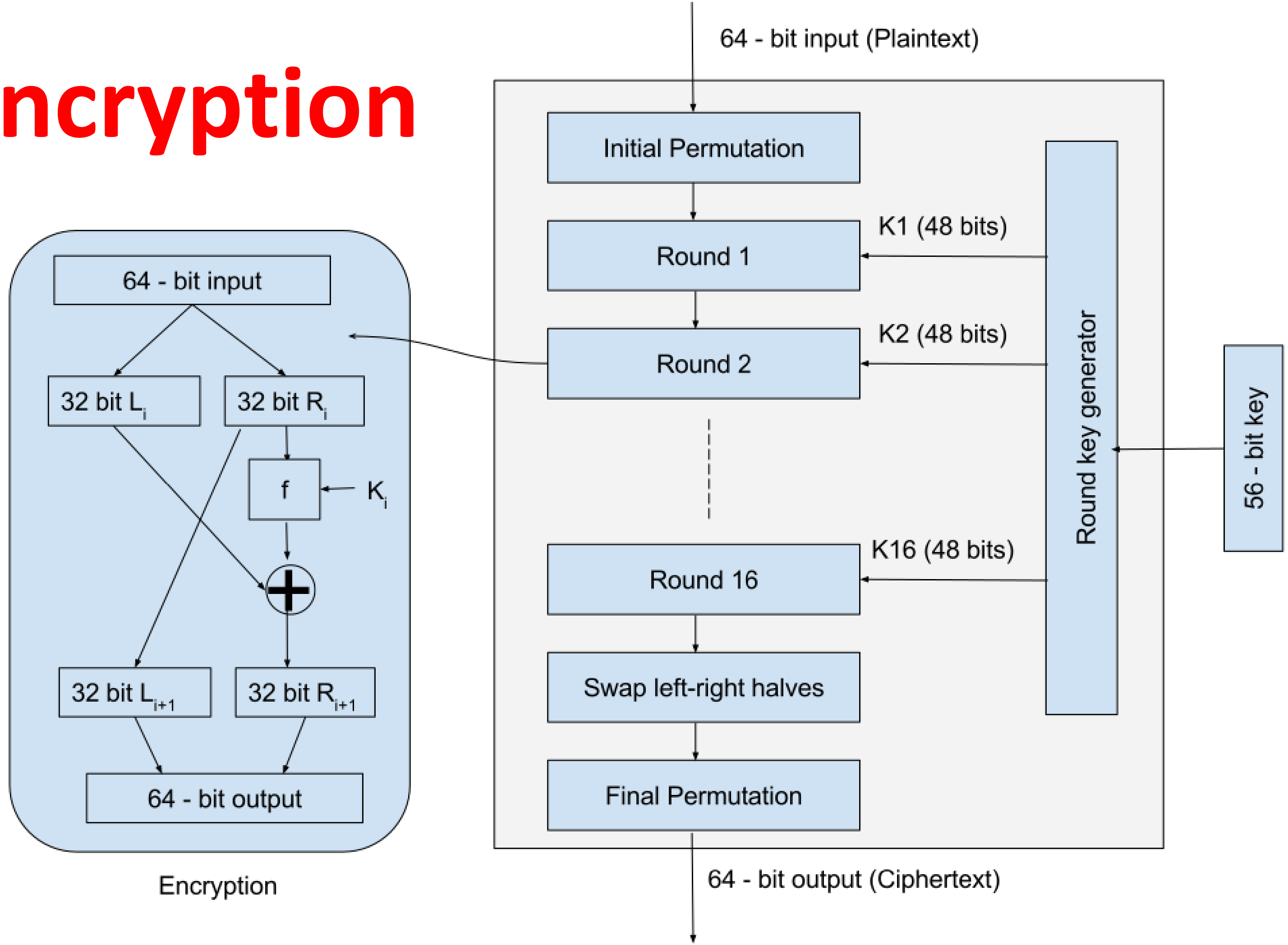
- Confusion via s-box
- Diffusion via p-box
- Many Rounds
- Fast and easy to implement
- Efficient to reverse (decrypt)
  - Same code/hardware for decryption

Data Encryption Standard (DES)

- Designed by IBM, published in 1977 by NIST
- 56-bit key (8 bit parity, controversy NSA), 64-bit input block
- Efficient to implement in hardware, slow in software (likely done on purpose)

- DES challenge: break an encrypted text
  - 1977: Can cost $20 million to break in 12 hours
  - 1997: 96 days; early 1998: 39 days; mid 1998: 56 hours ($250k machine); early 1999: 22 hours
- 3DES more secure
  - ciphertext = $E_{K3}(D_{K2}(E_{K1}(\text{plaintext}))))$
  - Often K3=K1 (112 bits of security, adequate)

Encryption

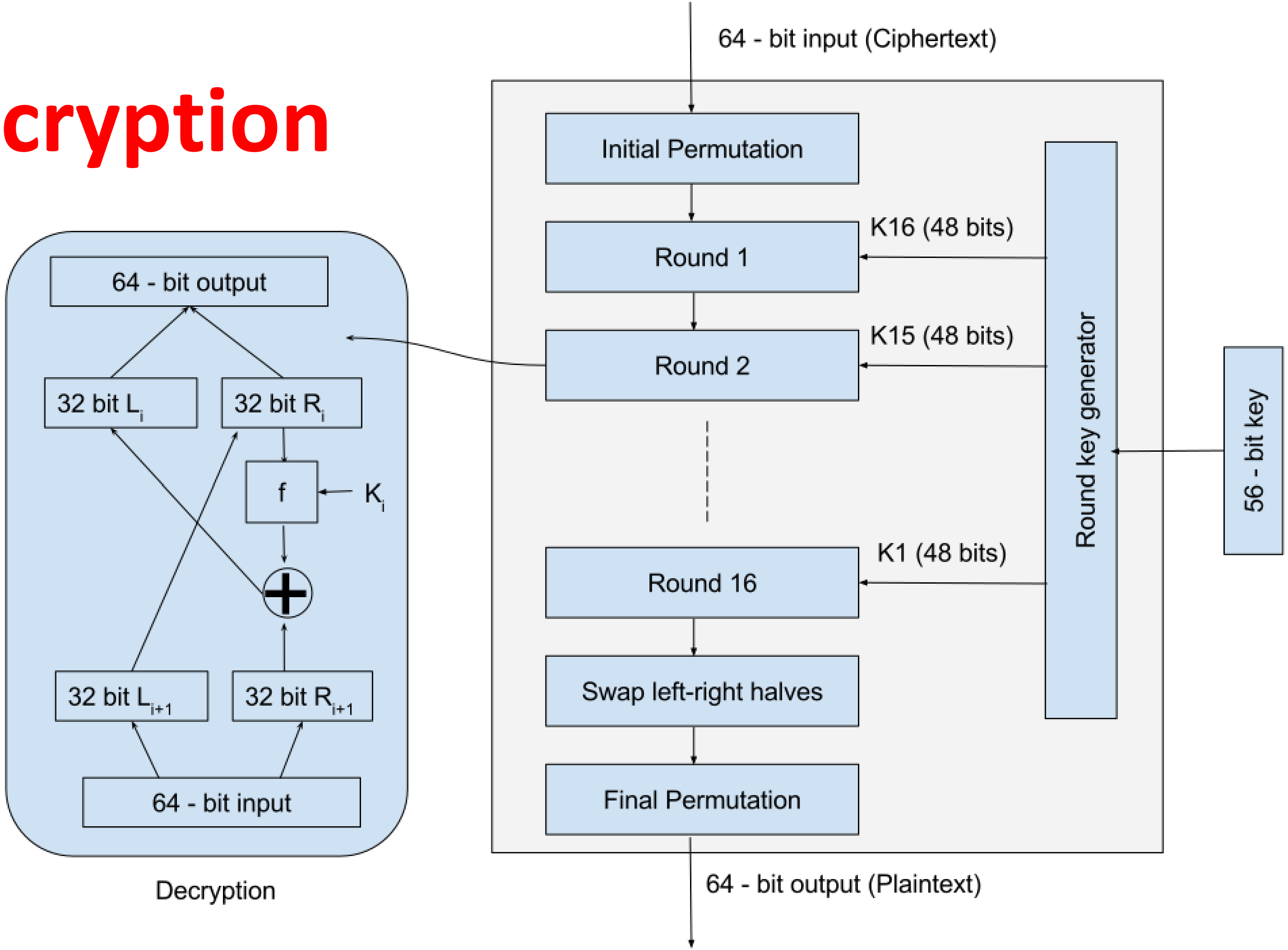


Encryption

F function

- Four operations:
  - Expansion: 32 bits expanded to 48 bits
    - Repeat bits and interchange positions
  - Xor: 48 bits xor’ed with round key
  - Substitution: 48 bits divided into eight 6-bits
    - 8 different S-boxes; output of each is 4 bits (total:32 bits)
    - Non linear operation → important for security
  - Permutation (P-box): 32 bits are permuted
- Number of rounds, S-box, permutations carefully chosen to thwart attacks

Decryption


64 - bit output
32 bit L_i
32 bit R_i
f
K_i
32 bit L_{i+1}
32 bit R_{i+1}
64 - bit input
Decryption
64 - bit input (Ciphertext)
Initial Permutation
Round 1
K16 (48 bits)
Round 2
K15 (48 bits)
Round 16
K1 (48 bits)
Swap left-right halves
Final Permutation
Round key generator
56 - bit key
64 - bit output (Plaintext)

Cryptoanalysis

- Known-plain text attack
  - Bruteforce (few tens of hrs, DES 56 key is too small!)
- Differential cryptoanalysis (chosen plain text attack)
  - Examines the transformation undergone by two related plaintext (many sets) during encryption
  - Based on above, assigns keys probability of being real key
  - Key obtained after 2^{47} examinations for DES (brute-force=2^{56})

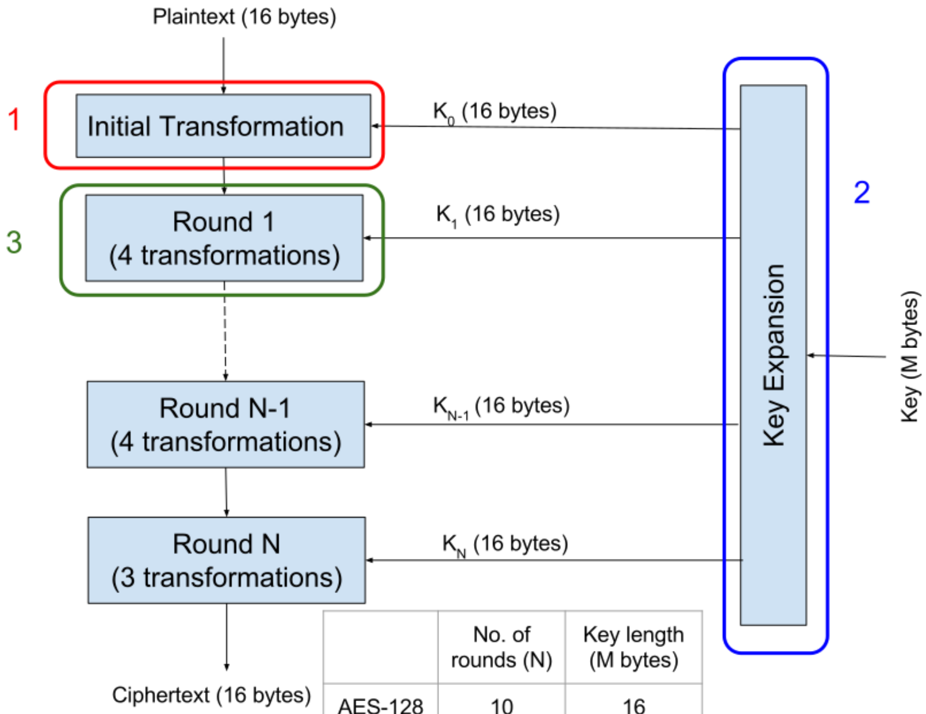
- Linear cryptoanalysis
  - Tries to find approximate linear relations between key bits, plain-text bits and cipher text bits
  - Requires 2^{43} plaintext-cipher text pairs for DES

Advanced Encryption Standard (AES)

- DES key too small (easy to break); 3DES too slow
- NIST wanted a new standard: efficient, flexible, secure, free to implement
  - Contest: 15 entries; winner = Rijndael (Daemen and Rijmen of Belgium); most secure not chosen
  - Standardized Rijndael as AES in 2001
  - Widely supported by all popular OS
- Input block: 128 bits
- Key: 128 or 192 or 256 bits (AES-128, AES-192 and AES-256)

AES Encryption

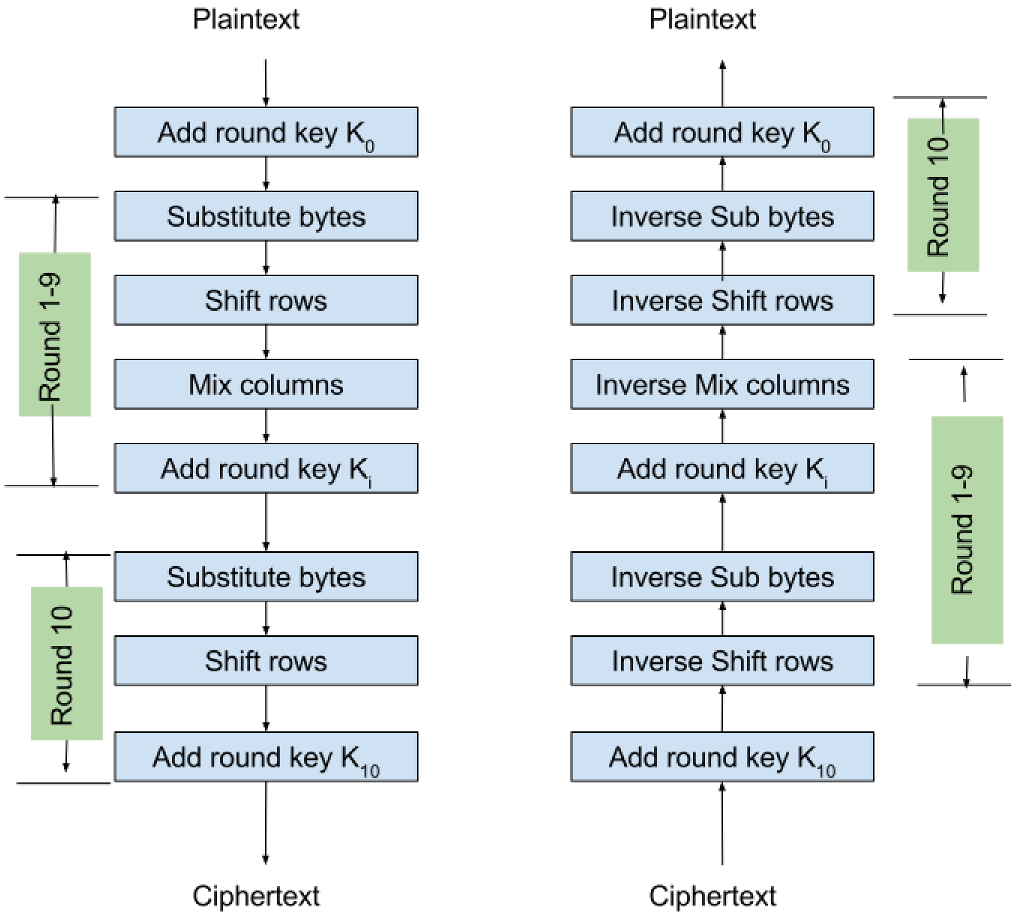
1. Initial Transformation
   – XOR between input state and Round 0 key
2. Key expansion
3. Round Transformations (4)
   – Last Round only 3 transformations (to make encryption/decryption similar in structure)
   – Number of rounds function of key length



	No. of rounds (N)	Key length (M bytes)
AES-128	10	16
AES-192	12	24
AES-256	14	32

AES Process

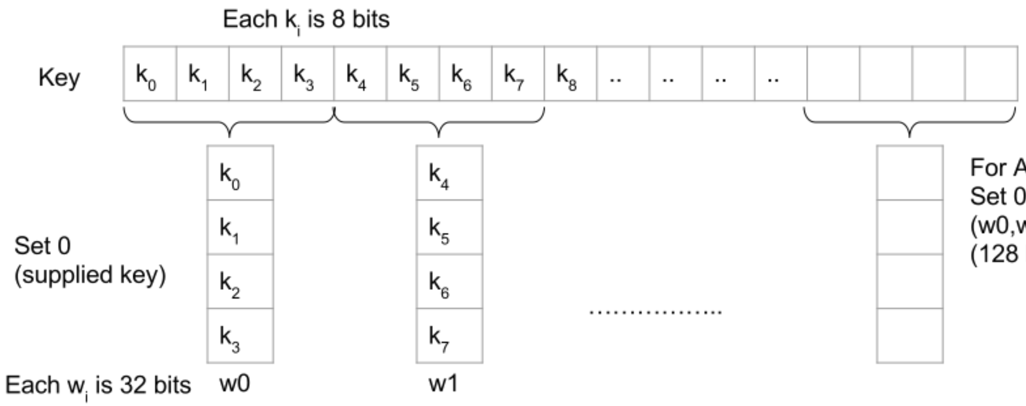
- Round Details
- Encryption and Decryption (for AES 128)



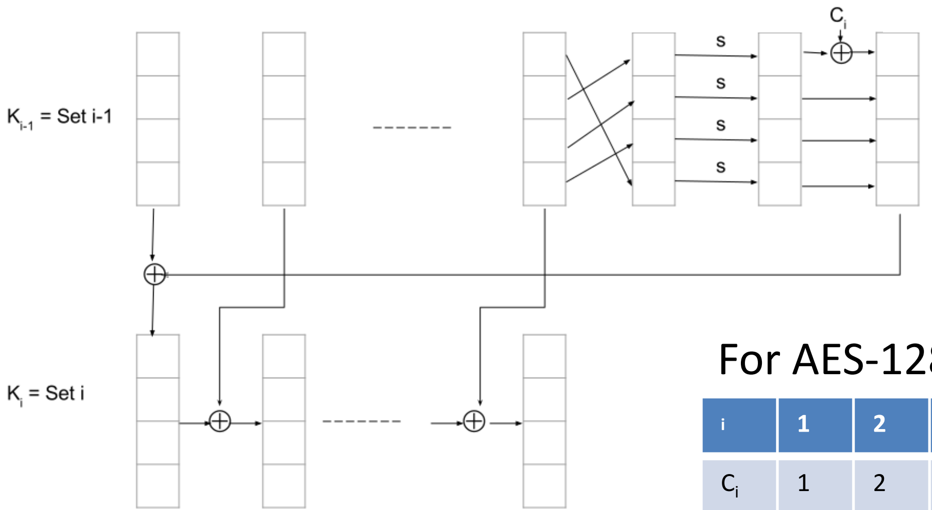
(a) **Encryption** (b) **Decryption**

Key Expansion

- Design Criteria
  - Fast execution
  - Simple design
  - Good Diffusion (from main key to round keys)
  - Resistant to attacks
    - Non-linearity to hinder analysis
    - Can’t recover key or other round keys if you know part of the key or part of the round key



For AES-128
Set 0 (=K_0) has 4 words
(w0,w1,w2 and w3)
(128 bits or 16 bytes)

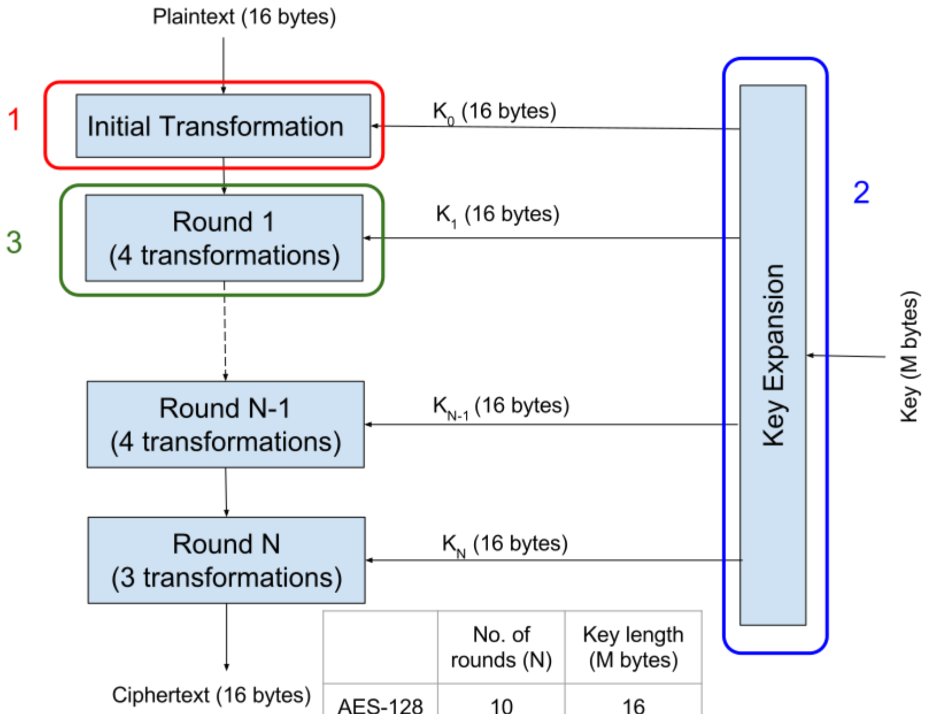


For AES-128

i	1	2	3	4	5	6	7	8	9	10
C_i	1	2	4	8	10	20	40	80	1b	36

AES Encryption

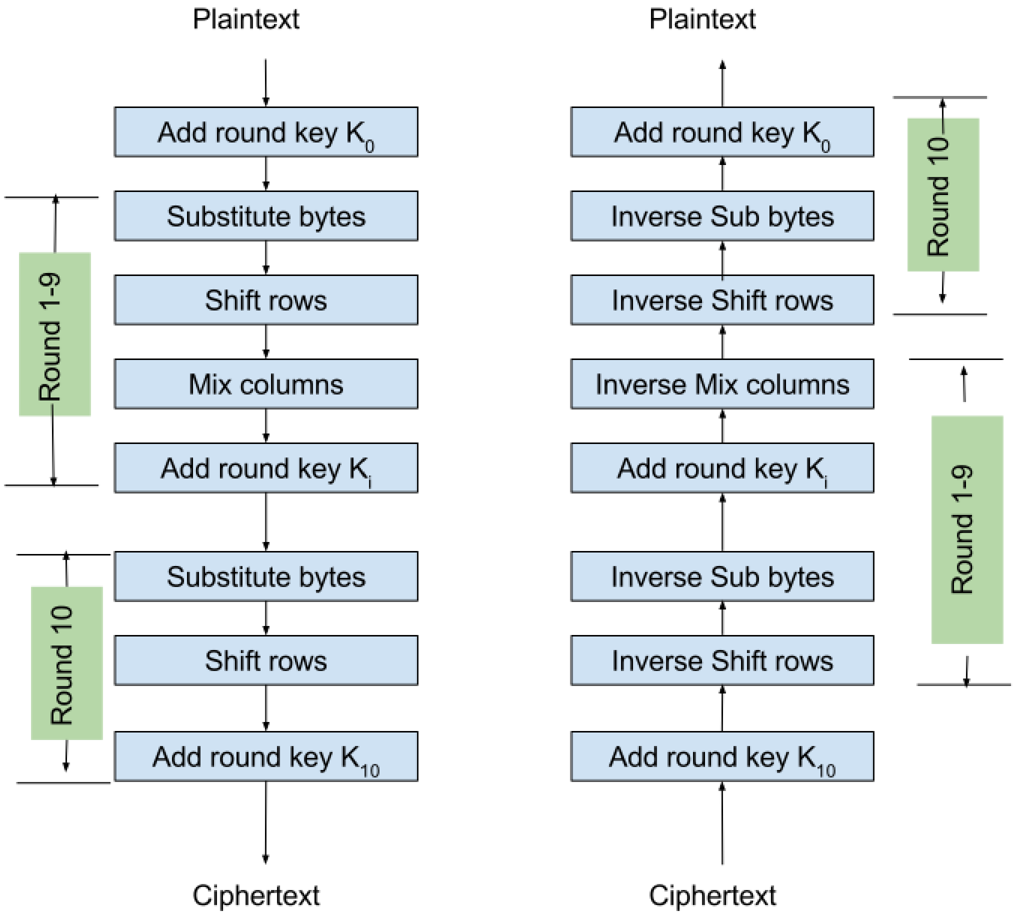
1. ~~Initial Transformation~~
   - ~~XOR between input state and Round 0 key~~
2. ~~Key expansion~~
3. Round Transformations (4)
   - Last Round only 3 transformations (to make encryption/decryption similar in structure)
   - Number of rounds function of key length



	No. of rounds (N)	Key length (M bytes)
AES-128	10	16
AES-192	12	24
AES-256	14	32

AES Process

- Round Details
- Encryption and Decryption (for AES 128)



(a) Encryption **(b) Decryption**

4 Transformations in a Round

- **Substitute Bytes: an S-Box substitution step**
- Shift Rows: A permutation step
- Mix Columns: a matrix multiplication
- Add Round Key: an XOR with a round key derived from the encryption key
- Focus: AES-128

Matrix Representation

- Input Block: 16 bytes

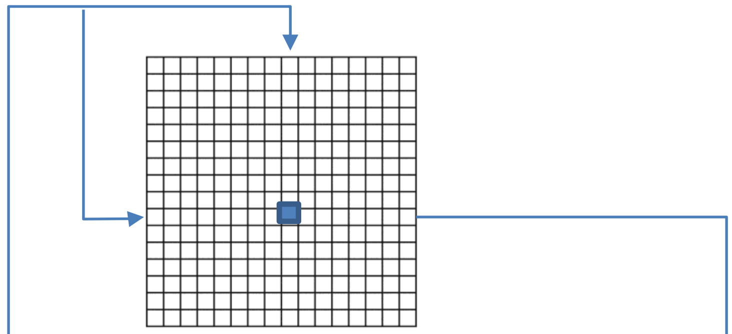
$$(s_{0,0}, s_{1,0}, s_{2,0}, s_{3,0}, s_{0,1}, s_{1,1}, s_{2,1}, s_{3,1}, s_{0,2}, s_{1,2}, s_{2,2}, s_{3,2}, s_{0,3}, s_{1,3}, s_{2,3}, s_{3,3})$$

- Matrix Representation

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

S-Box

- A mathematical operation on 8-bit word in GF(2^8)
- Math ensures resistance to linear and differential crypto-analysis
- Can be implemented via a simple lookup table



$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \quad \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

B6	90	4F	3C
EA	1D	84	1B
C5	D9	77	81
0B	66	90	2A

4E	60	84	EB
87	A4	5F	AF
A6	35	F5	0C
2B	33	60	E5

Forward S-box

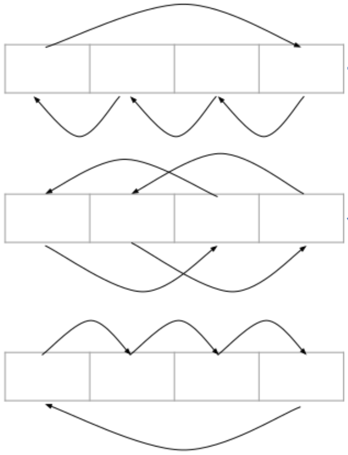
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Transformations in a Round

- Substitute Bytes: an S-Box substitution step
- **Shift Rows: A permutation step**
- Mix Columns: a matrix multiplication
- Add Round Key: an XOR with a round key derived from the encryption key
- Focus: AES-128

Shift Rows
(a simple permutation)

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$



$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{bmatrix}$$

4E	60	84	EB
87	A4	5F	AF
A6	35	F5	0C
2B	33	60	E5

4E	60	84	EB
A4	5F	AF	87
F5	0C	A6	35
E5	2B	33	60

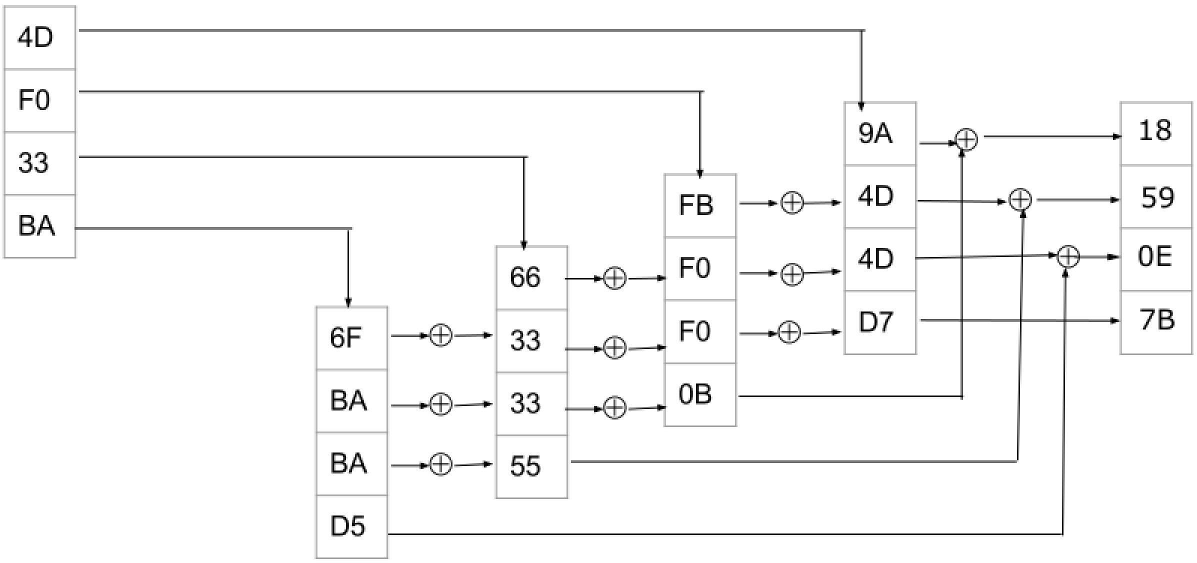
Transformations in a Round

- Substitute Bytes: an S-Box substitution step
- Shift Rows: A permutation step
- **Mix Columns: a matrix multiplication**
- Add Round Key: an XOR with a round key derived from the encryption key
- Focus: AES-128

Mix Columns

- Helps in diffusion
- Replaces a 4 octet column with another 4 octet column
- Involves math; matrix multiplication in $GF(2^8)$
- Can be implemented via a single table and some xors

Mix Column Table



right (low-order) nibble

left (high-order) nibble

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	00 00 00 00	02 01 01 03	04 02 02 06	06 03 03 05	08 04 04 0c	0a 05 05 0f	0c 06 06 0a	0e 07 07 09	10 08 08 18	12 09 09 1b	[illegible] 0a 0a 1e	16 0b 0b 1d	18 0c 0c 14	1a 0d 0d 17	1c 0e 0e 12	1e 0f 0f 11
1	20 10 10 30	22 11 11 33	24 12 12 36	26 13 13 35	28 14 14 3c	2a 15 15 3f	2c 16 16 3a	2e 17 17 39	30 18 18 28	32 19 19 2b	34 1a 1a 2e	36 1b 1b 2d	38 1c 1c 24	3a 1d 1d 27	3c 1e 1e 22	3e 1f 1f 21
2	40 20 20 60	42 21 21 63	44 22 22 66	46 23 23 65	48 24 24 6c	4a 25 25 6f	4c 26 26 6a	4e 27 27 69	50 28 28 78	52 29 29 7b	54 2a 2a 7e	56 2b 2b 7d	58 2c 2c 74	5a 2d 2d 77	5c 2e 2e 72	5e 2f 2f 71
3	60 30 30 50	62 31 31 53	64 32 32 56	66 33 33 55	68 34 34 5c	6a 35 35 5f	6c 36 36 5a	6e 37 37 59	70 38 38 48	72 39 39 4b	74 3a 3a 4e	76 3b 3b 4d	78 3c 3c 44	7a 3d 3d 47	7c 3e 3e 42	7e 3f 3f 41
4	80 40 40 c0	82 41 41 c3	84 42 42 c6	86 43 43 c5	88 44 44 cc	8a 45 45 cf	8c 46 46 ca	8e 47 47 c9	90 48 48 d8	92 49 49 db	94 4a 4a de	96 4b 4b dd	98 4c 4c d4	9a 4d 4d d7	9c 4e 4e d2	9e 4f 4f d1
5	a0 50 50 f0	a2 51 51 f3	a4 52 52 f6	a6 53 53 f5	a8 54 54 fc	aa 55 55 ff	ac 56 56 fa	ae 57 57 f9	b0 58 58 e8	b2 59 59 eb	b4 5a 5a ee	b6 5b 5b ed	b8 5c 5c e4	ba 5d 5d e7	bc 5e 5e e2	be 5f 5f e1
6	c0 60 60 a0	c2 61 61 a3	c4 62 62 a6	c6 63 63 a5	c8 64 64 ac	ca 65 65 af	cc 66 66 aa	ce 67 67 a9	d0 68 68 b8	d2 69 69 bb	d4 6a 6a be	d6 6b 6b bd	d8 6c 6c b4	da 6d 6d b7	dc 6e 6e b2	de 6f 6f b1
7	e0 70 70 90	e2 71 71 93	e4 72 72 96	e6 73 73 95	e8 74 74 9c	ea 75 75 9f	ec 76 76 9a	ee 77 77 99	f0 78 78 88	f2 79 79 8b	f4 7a 7a 8e	f6 7b 7b 8d	f8 7c 7c 84	fa 7d 7d 87	fc 7e 7e 82	fe 7f 7f 81
8	1b 80 80 9b	19 81 81 98	1f 82 82 9d	1d 83 83 9e	13 84 84 97	11 85 85 94	17 86 86 91	15 87 87 92	0b 88 88 83	09 89 89 80	0f 8a 8a 85	0d 8b 8b 86	03 8c 8c 8f	01 8d 8d 8c	07 8e 8e 89	05 8f 8f 8a
9	3b 90 90 ab	39 91 91 a8	3f 92 92 ad	3d 93 93 ae	33 94 94 a7	31 95 95 a4	37 96 96 a1	35 97 97 a2	2b 98 98 b3	29 99 99 b0	2f 9a 9a b5	2d 9b 9b b6	23 9c 9c bf	21 9d 9d bc	27 9e 9e b9	25 9f 9f ba
a	5b a0 a0 fb	59 a1 a1 f8	5f a2 a2 fd	5d a3 a3 fe	53 a4 a4 f7	51 a5 a5 f4	57 a6 a6 f1	55 a7 a7 f2	4b a8 a8 e3	49 a9 a9 e0	4f aa aa [illegible]	4d ab ab e6	43 ac ac ef	41 ad ad ec	47 ae ae e9	45 af af ea
b	7b b0 b0 cb	79 b1 b1 c8	7f b2 b2 cd	7d b3 b3 ce	73 b4 b4 c7	71 b5 b5 c4	77 b6 b6 c1	75 b7 b7 c2	6b b8 b8 d3	69 b9 b9 d0	6f ba ba d5	6d bb bb d6	63 bc bc df	61 bd bd dc	67 be be d9	65 bf bf da
c	9b c0 c0 5b	99 c1 c1 58	9f c2 c2 5d	9d c3 c3 5e	93 c4 c4 57	91 c5 c5 54	97 c6 c6 51	95 c7 c7 52	8b c8 c8 43	89 c9 c9 40	[illegible] ca ca 45	8d cb cb 46	83 cc cc 4f	81 cd cd 4c	87 ce ce 49	85 cf cf 4a
d	bb d0 d0 6b	b9 d1 d1 68	bf d2 d2 6d	bd d3 d3 6e	b3 d4 d4 67	b1 d5 d5 64	b7 d6 d6 61	b5 d7 d7 62	ab d8 d8 73	a9 d9 d9 70	af da da 75	ad db db 76	a3 dc dc 7f	a1 dd dd 7c	a7 de de 79	a5 df df 7a
e	db e0 e0 3b	d9 e1 e1 38	df e2 e2 3d	dd e3 e3 3e	d3 e4 e4 37	d1 e5 e5 34	d7 e6 e6 31	d5 e7 e7 32	cb e8 e8 23	c9 e9 e9 20	cf ea ea 25	cd eb eb 26	c3 ec ec 2f	c1 ed ed 2c	c7 ee ee 29	c5 ef ef 2a
f	fb f0 f0 0b	f9 f1 f1 08	ff f2 f2 0d	fd f3 f3 0e	f3 f4 f4 07	f1 f5 f5 04	f7 f6 f6 01	f5 f7 f7 02	eb f8 f8 13	e9 f9 f9 10	ef fa fa 15	ed fb fb 16	e3 fc fc 1f	e1 fd fd 1c	e7 fe fe 19	e5 ff ff 1a

Transformations in a Round

- Substitute Bytes: an S-Box substitution step
- Shift Rows: A permutation step
- Mix Columns: a matrix multiplication
- **Add Round Key: an XOR with a round key derived from the encryption key**

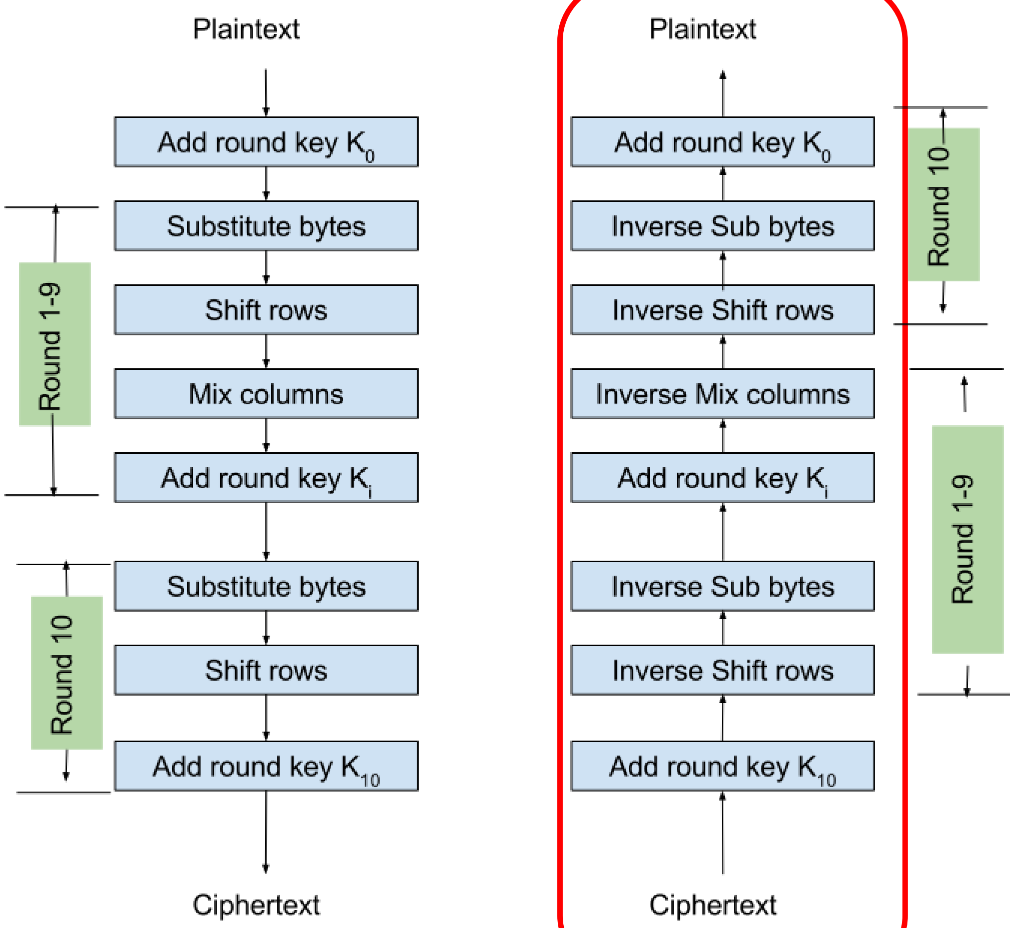
Add Round Key

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \oplus \begin{bmatrix} W_i & W_{i+1} & W_{i+2} & W_{i+3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Decryption

- S-box, Mix-column use a well defined but different table
- Shift-row will shift in the opposite direction
- Use round keys in reverse order

Very efficient implementation responsible for selection



(a) **Encryption** (b) **Decryption**

Cryptoanalysis

- Designed to resist linear/differential crypto-analysis
- Only know practical attack for AES is side-channel attack
- AES can be considered an ideal cipher (for most purposes)

Summary

- Two types of symmetric key encryption: block and stream; focus: block
- Desirable properties; General structure of symmetric block ciphers
- Brief overview of DES
- Detailed overview of AES