

Lecture 4

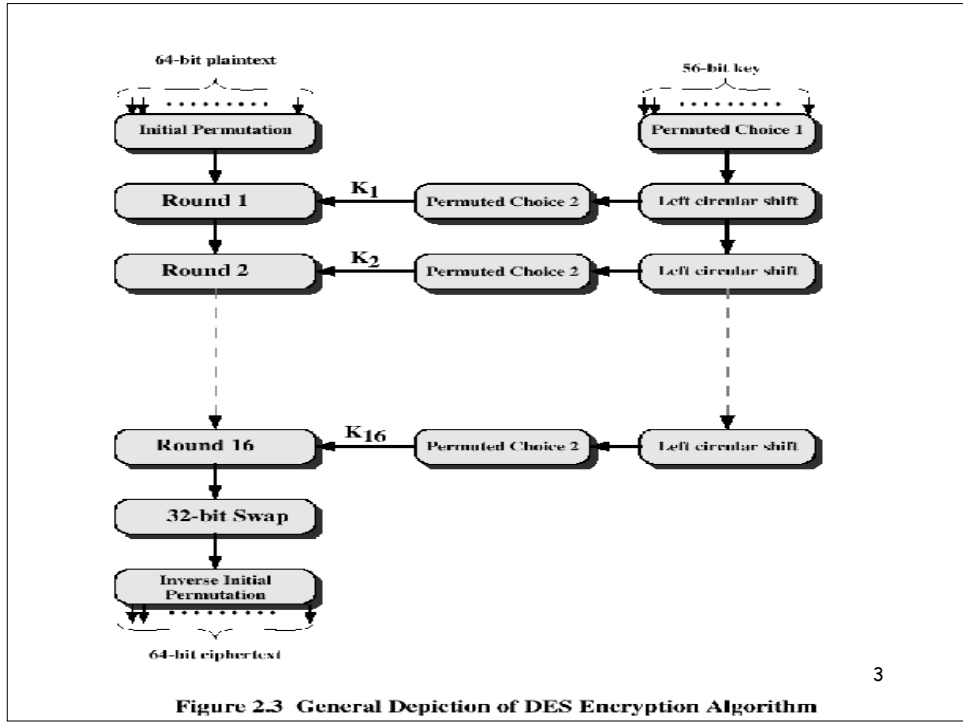
Encryption Continued...

1

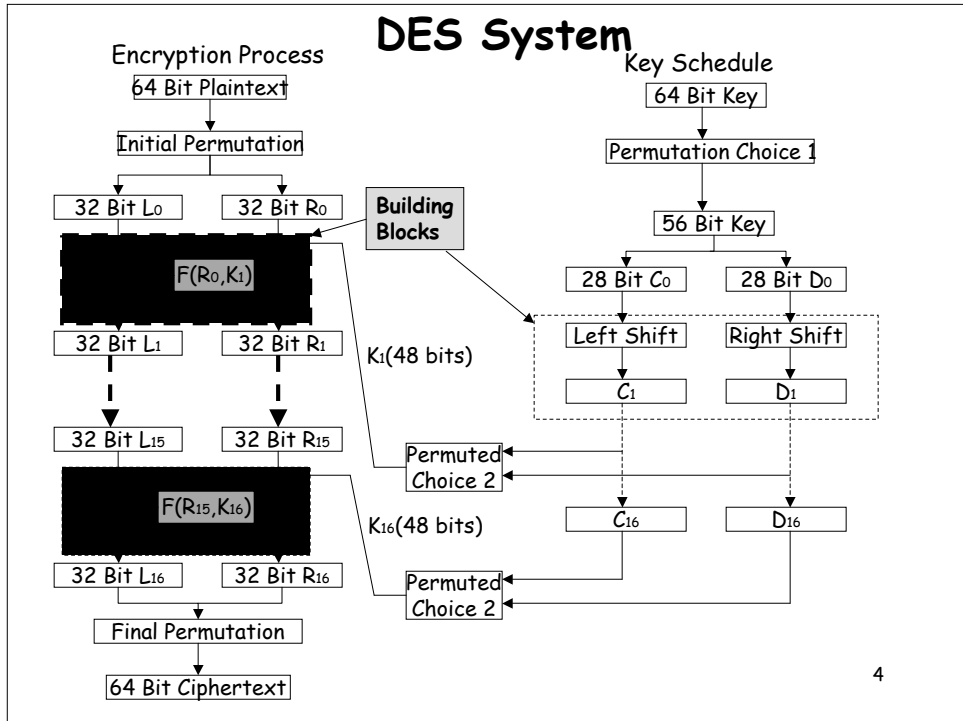
Data Encryption Standard (DES)

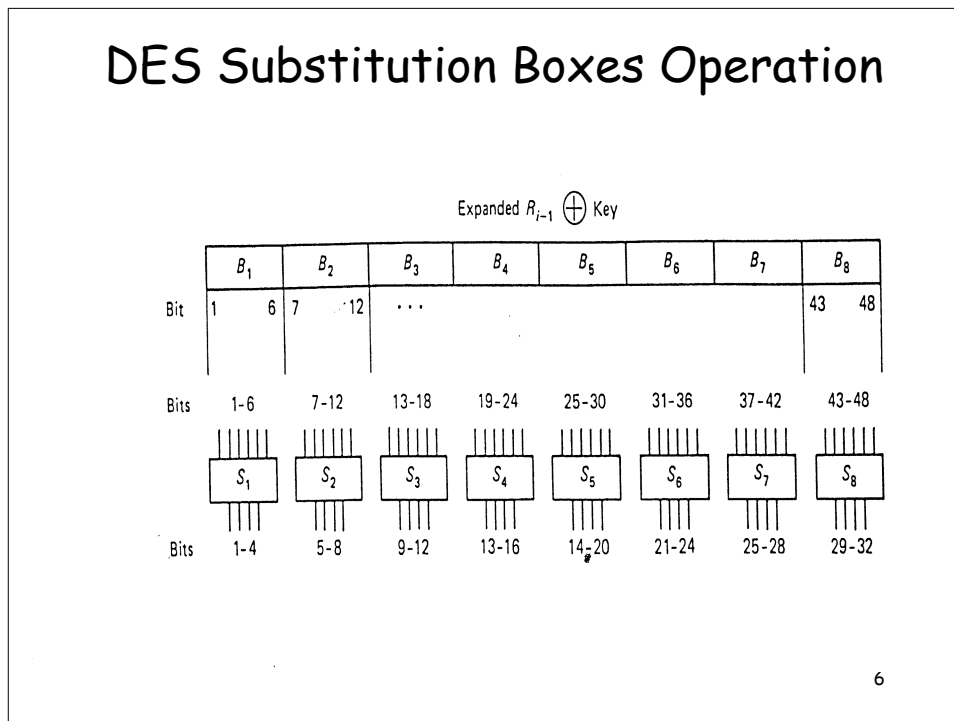
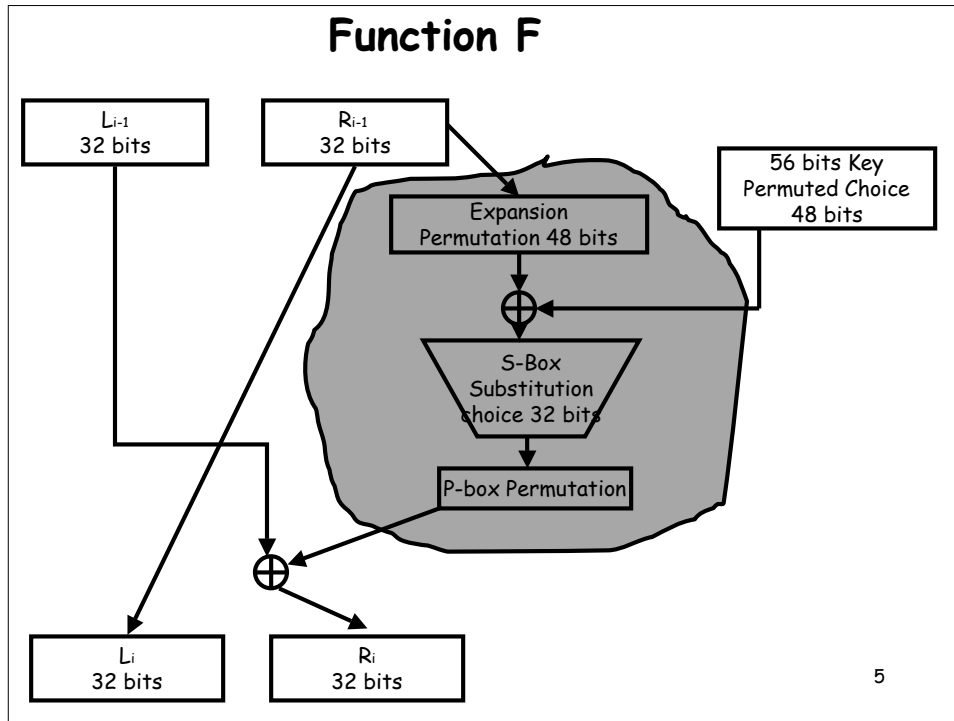
- 64 bit input block
- 64 bit output block
- 16 rounds
- 64 (effective 56) bit key
- Key schedule computed at startup
- Aimed at bulk data
- >16 rounds doesn't help
- >56 bit key doesn't help
- Other S-boxes usually hurt...

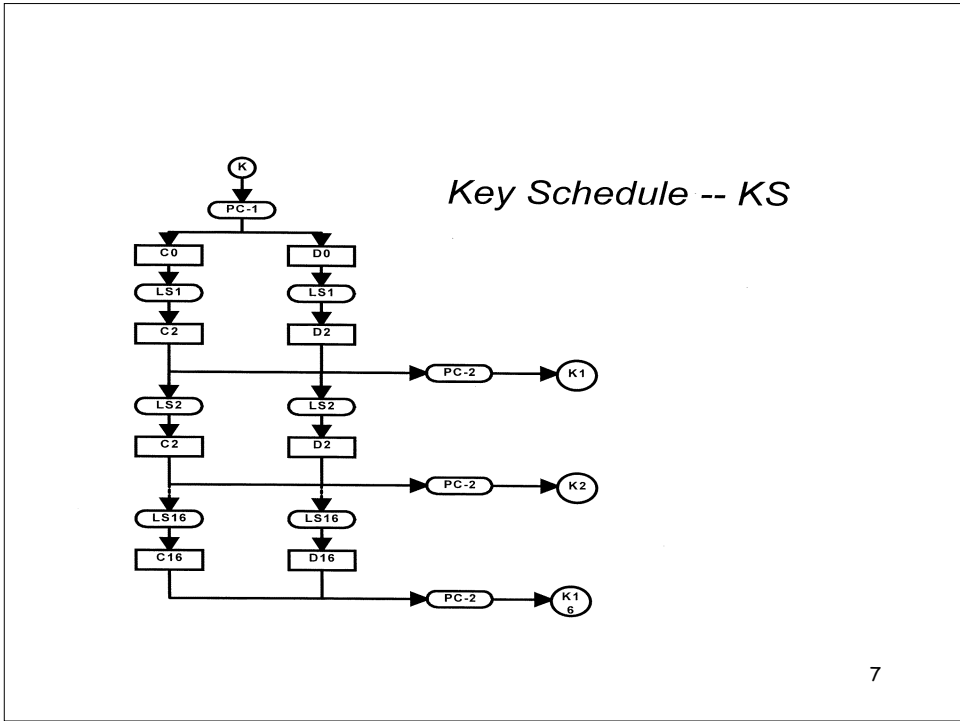
2



3







7

Key schedule of shifts

Iteration(i)	No. of shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Key permutation PC-1

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

Key permutation PC-2

14	17	11	24	1	5
3	28	15	6	20	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	54
46	42	50	36	29	32

Operation Tables of DES
(Key Schedule, PC-1, PC-2)

8

Operation Tables of DES (IP, IP⁻¹, E and P)

Initial Permutation (IP)

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

Bit-Selection Table E

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

Inverse Initial Permutation (IP⁻¹)

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

Permutation P

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

9

DES: Modes of Operation

- Electronic code-book (ECB)
 - $C_i = E(K, P_i)$
 - Local error, permutation attack, parallel encr.
- Chained block cipher (CBC)
 - $C_i = E(K, C_{i-1} \text{ XOR } P_i)$
 - Need IV, error causes 2-block loss, no parallel enc.
- Output feedback (OFB)
 - $V_i = E(K, V_{i-1}) \quad C_i = P_i \text{ XOR } V_i$
 - Stream cipher, local error, pre-computation
- Cipher feedback (CFB)
 - $C_i = P_i \text{ XOR } E(K, C_{i-1})$
 - OFB/CFB - encrypt only!
 - Plaintext dependence, avalanche effect, parallel decryption.
- Message Auth Code (MAC)
 - P_1, \dots, P_n, C_n (CBC)

10

CBC Mode

- Cipher Block Chaining Mode (CBC)
 - Input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block.
 - Repeating pattern of 64-bits are not exposed
 - Block rearrangement made difficult!

$$C_i = E_k[C_{i-1} \oplus P_i]$$

$$D_K[C_i] = D_K[E_K(C_{i-1} \oplus P_i)]$$

$$D_K[C_i] = (C_{i-1} \oplus P_i)$$

$$C_{i-1} \oplus D_K[C_i] = C_{i-1} \oplus C_{i-1} \oplus P_i = P_i$$

11

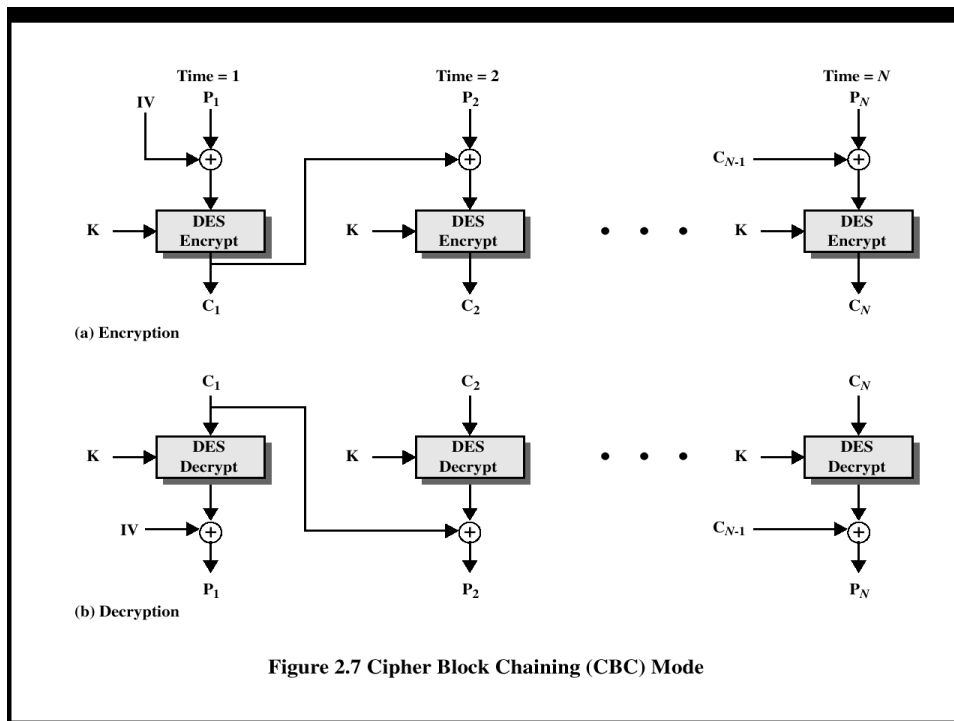


Figure 2.7 Cipher Block Chaining (CBC) Mode

Breaking DES (Cryptanalysis)

Differential Cryptanalysis

- ❖ Looks for correlations in $f()$ -function input and output

Linear cryptanalysis

- ❖ Looks for correlations between key and cipher input and output

Related-key cryptanalysis

- ❖ Looks for correlations between key changes and cipher input/output

Differential cryptanalysis discovered in 1990; virtually all block ciphers from before that time are vulnerable...

...except DES. IBM (and the NSA) knew about it 15 years earlier

13

Breaking DES (Cryptanalysis)

DES Key size = 56 bits

- Brute force = 2^{55} attempts
- Differential cryptanalysis = 2^{47} attempts
- Linear cryptanalysis = 2^{43} attempts

- Longer than 56 bit keys don't make it any stronger
- More than 16 rounds don't make it any stronger
- DES Key Problems:
 - Weak keys (all 0s, all 1s, a few others)
 - Key size = 56 bits = $8 * 7$ -bit ASCII
 - Alphanumeric-only password converted to uppercase
 $8 * \sim 5$ -bit chars = 40 bits

14

DES Variants

- **3-DES (triple DES)**

$C = E(K1, D(K2, E(K1,P))) \rightarrow 112$ effective key bits

$C = E(K3, D(K2, E(K1,P))) \rightarrow 168$ effective key bits

- **DESx**

$C = K3 \text{ XOR } E(K2, (K1 \text{ XOR } P)) \rightarrow$ seems like 184 key bits

Effective key bits \rightarrow approx. 118

- **Other simple variations:**

$C = K1 \text{ XOR } E(K1', P) \rightarrow$ weak!

$C = E(K1, K1' \text{ XOR } P) \rightarrow ?$

15

DES Variants

Why does n-DES work?

Because, as a function, DES is not a group...

Suppose: $C = \text{DES}(K1, \text{DES}(K2, P))$

**There is no K , such that, for each possible
plaintext P , $\text{DES}(K, P) = C$**

16

DES summary

- Permutation/substitution block cipher
- 64-bit data blocks
- 56-bit keys (8 parity bits)
- 16 rounds (shifts, XORs)
- Key schedule
- S-box selection secret...
- DES "aging"
- 2-DES: rendezvous attack
- 3-DES: 112-bit security?
- DESx : 184-bit security?

17

Other Symmetric Ciphers

Skipjack

- Classified algorithm originally designed for Clipper,
- declassified in 1998
- 32 rounds, breakable with 31 rounds
- 80 bit key, inadequate for long-term security

GOST

- GOST 28147, Russian answer to DES
- 32 rounds, 256 bit key
- Incompletely specified

18

Other Symmetric Ciphers

- IDEA (X. Lai, J. Massey, ETH)
 - Developed as PES (proposed encryption standard),
 - adapted to resist differential cryptanalysis
 - Gained popularity via PGP, 128 bit key
 - Patented (Ascom CH)
- Blowfish (B. Schneier, Counterpane)
 - Optimized for high-speed execution on 32-bit processors
 - 448 bit key, relatively slow key setup
 - Fast for bulk data on most PCs/laptops
 - Easy to implement, runs in ca. 5K of memory

19

Other Symmetric Ciphers

RC4 (Ron's Cipher #4) Stream cipher:

- ❖ Optimized for fast software implementation
- ❖ Character streaming (not bit)
- ❖ 8-bit output
- ❖ Former trade secret of RSADSI,
- ❖ Reverse-engineered and posted to the net in 1994:
- ❖ 2048-bit key
- ❖ Used in many products until recently

20

Other Symmetric Ciphers (RC4)

```
while( length-- )
{
    /* state[0-255] contains key bytes */
    sx = state[ ++x & 0xFF ];
    y += sx & 0xFF 256;
    sy = state[ y ];
    state[ y ] = sx;
    state[ x ] = sy;
    *data++ ^= state[ ( sx+sy ) & 0xFF ];
}
```

Takes about a minute to implement from memory₂₁

Other Symmetric Ciphers

- RC5
 - Suitable for hardware and software
 - Fast, simple
 - Adaptable to processors of different word lengths
 - Variable number of rounds
 - Variable-length key (0-256 bytes)
 - Very low memory requirements
 - High security (no effective attacks, yet...)
 - Data-dependent rotations

Other Symmetric Ciphers

- RC5 single round pseudocode:

$$L \leftarrow L \text{ XOR } R$$
$$L \leftarrow L \lll R$$
$$L \leftarrow L + \text{subkey}[2i]$$
$$R \leftarrow R \text{ XOR } L$$
$$R \leftarrow R \lll L$$
$$R \leftarrow R + \text{subkey}[2i + 1]$$

23

AES:
The Rijndael Block
Cipher

24

Introduction and History

- National Institute of Science and Technology (NIST) regulates standardization in the US
- DES is an aging standard that no longer addresses today's needs for strong encryption
- Triple-DES: Endorsed by NIST as a "de facto" standard
- AES: Advanced Encryption Standard
 - Finalized in 2001
 - Goal is to define the Federal Information Processing Standard (FIPS) by selecting a new encryption algorithm suitable for encrypting government documents
 - Candidate algorithms must be:
 - Symmetric-key ciphers supporting 128, 192, and 256 bit keys
 - Royalty-Free
 - Unclassified (i.e. public domain)
 - Available for worldwide export

25

Introduction and History

- AES Round-3 Finalist Algorithms:
 - MARS
 - Candidate offering from IBM
 - RC6
 - By Ron Rivest of MIT & RSA Labs, creator of the widely used RC4/RC5 algorithm and "R" in RSA
 - Twofish
 - From Counterpane Internet Security, Inc.
 - Serpent
 - by Ross Anderson, Eli Biham and Lars Knudsen
 - Rijndael
 - by Joan Daemen and Vincent Rijmen (KUL, Belgium)

26

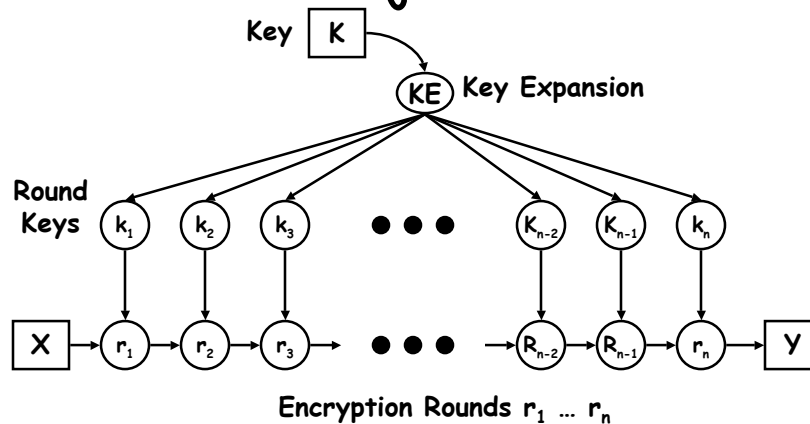
Rijndael

The Winner: Rijndael

- Joan Daemen (of Proton World International) and Vincent Rijmen (of Katholieke Universiteit Leuven).
- pronounced "Rhine-doll"
- Allows only 128, 192, and 256-bit key sizes (unlike the other candidates)
- Variable input block length: 128, 192, or 256 bits. All nine combinations of key-block length possible.
 - A block is the smallest data size the algorithm will encrypt
- Vast speed improvement over DES in both hw and sw implementations
 - 8.416 bytes/sec on a 20MHz 8051
 - 8.8 Mbytes/sec on a 200MHz Pentium Pro

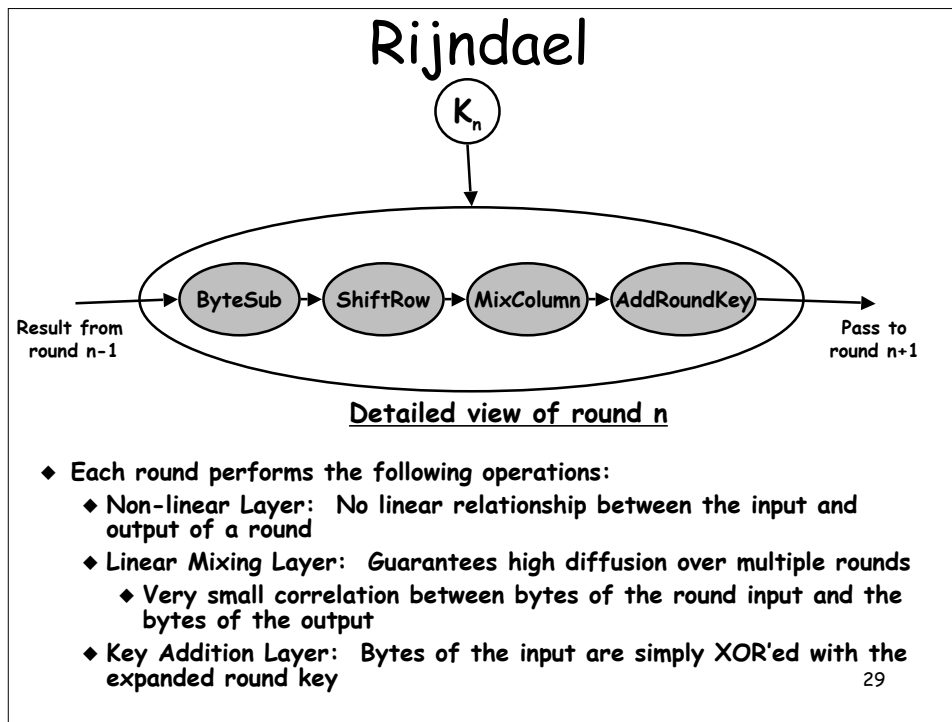
27

Rijndael



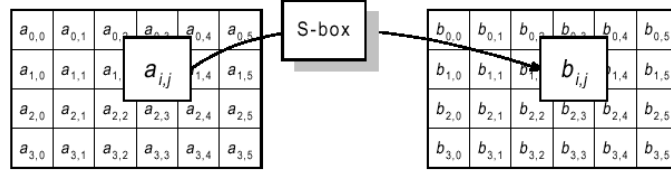
- ◆ Key is expanded to a set of n round keys
- ◆ Input block X undergoes n rounds of operations (each operation is based on value of the n -th round key), until it reaches a final round.
- ◆ Strength of algorithm relies on the fact that it's very difficult to obtain intermediate result (or *state*) of round n from round $n+1$ without the round key.

28



- # Rijndael
- Three layers provide strength against known types of cryptographic attacks: Rijndael provides "full diffusion" after only two rounds
 - Immune to:
 - Linear and differential cryptanalysis
 - Known-key and related-key attacks
 - Square attack
 - Interpolation attacks
 - Weak-keys
 - Rijndael has been shown *secure*:
 - No key-recovery attacks faster than exhaustive search
 - No known symmetry properties in the round mapping
 - No weak keys
 - No related-key attacks → no two keys have a high number of expanded round keys in common
- 30

Rijndael: ByteSub



Each byte at the input of a round undergoes a non-linear byte substitution according to the following transform:

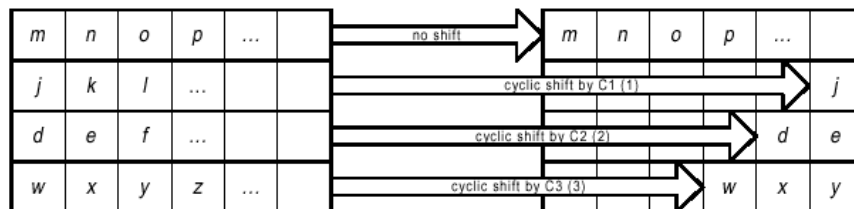
$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Substitution ("S")-box

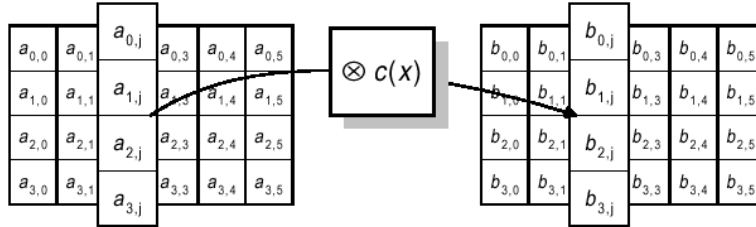
Rijndael: ShiftRow

Nb	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

Depending on the block length, each "row" of the block is cyclically shifted according to the above table



Rijndael: MixColumn



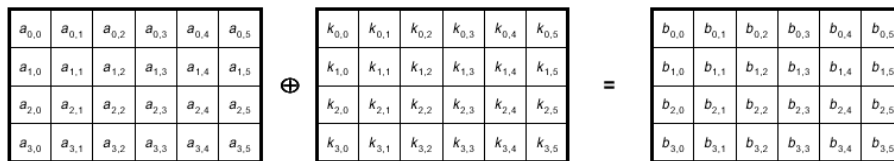
Each column is multiplied by a fixed polynomial
 $C(x) = '03'*X^3 + '01'*X^2 + '01'*X + '02'$

This corresponds to matrix multiplication $b(x) = c(x) \otimes a(x)$:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

33

Rijndael: Key Expansion and Addition



Each word is simply XOR'ed with the expanded round key

Key Expansion algorithm:

```

KeyExpansion(int* Key[4*Nk], int* EKey[Nb*(Nr+1)])
{
    for(i = 0; i < Nk; i++)
        EKey[i] = (Key[4*i], Key[4*i+1], Key[4*i+2], Key[4*i+3]);
    for(i = Nk; i < Nb * (Nr + 1); i++)
    {
        temp = EKey[i - 1];
        if (i % Nk == 0)
            temp = SubByte(RotByte(temp)) ^ Rcon[i / Nk];
        EKey[i] = EKey[i - Nk] ^ temp;
    }
}
    
```

34

Rijndael: Implementations

- Well-suited for software implementations on 8-bit processors (important for "Smart Cards")
 - Atomic operations focus on bytes and nibbles, not 32- or 64-bit integers
 - Layers such as ByteSub can be efficiently implemented using small tables in ROM (e.g. < 256 bytes).
 - No special instructions are required to speed up operation, e.g. barrel rotates
- For 32-bit implementations:
 - An entire round can be implemented via a fast table lookup routine on machines with 32-bit or higher word lengths
 - Considerable parallelism exists in the algorithm
 - Each layer of Rijndael operates in a parallel manner on the bytes of the round state, all four component transforms act on individual parts of the block
 - Although the Key expansion is complicated and cannot benefit much from parallelism, it only needs to be performed *once* until the two parties switch keys.

35

Rijndael: Implementations

- Hardware Implementations
 - Rijndael performs very well in software, but there still exists cases where more performance is required (e.g. server and VPN applications).
 - Multiple S-Box engines, round-key XORs, and byte shifts can all be implemented efficiently in hardware when absolute speed is required
 - Small amount of hardware can vastly speed up 8-bit implementations
- Inverse Cipher
 - Except for the non-linear ByteSub step, each part of Rijndael has a straightforward inverse and the operations simply need to be undone in the reverse order.
 - However, Rijndael was specially written so that the same code that encrypts a block can also decrypt the same block simply by changing certain tables and polynomials for each layer. The rest of the operation remains identical.

36

Conclusions and The Future

- Rijndael is an extremely fast, state-of-the-art, highly secure algorithm
- Amenable to efficient implementation in both hw and sw; requires no special instructions to obtain good performance on any computing platform
- Triple-DES, still highly secure and supported by NIST, is expected to be common for the foreseeable future.

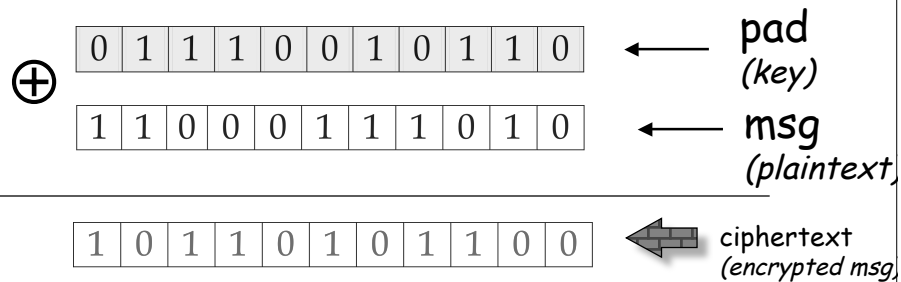
37

Reminder:
World's best cipher!

38

One-time pad

For each character:



39

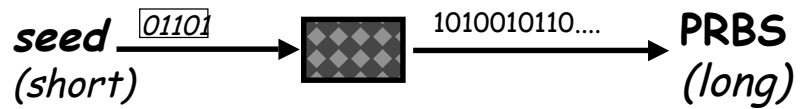
One-time pad (cont.)

- **Symmetric**
- **Pad is selected at random**
- **Pad is as long as plaintext**
- **Perfectly secure, but...**
- **One time only:**
 - so sending the pad is just as hard as sending the msg

40

Pseudo-random OTP

Pseudo-random bit string (PRBS) generator:



PRBS = Hard to guess a bit (after seeing many others)