



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Computer Security and Cryptography

CS381

来学嘉

计算机科学与工程系 电院3-423室

34205440 1356 4100825 laix@sjtu.edu.cn

2015-03



Organization



- Week 1 to week 16 (2015-03 to 2014-06)
- 东中院-3-102
- Monday 3-4节; week 9-16
- Wednesday 3-4节; week 1-16
- lecture 10 + exercise 40 + **random tests** 40 + other 10
- Ask questions in class – counted as points
- Turn ON your mobile phone (after lecture)
- Slides and papers:
 - <http://202.120.38.185/CS381> <http://yuyu.hk/pages/CS381.html>
 - computer-security
 - http://202.120.38.185/references
- TA: Geshi Huang gracehgs@mail.sjtu.edu.cn
- Send homework to the TA

Rule: do the homework on your own!



Contents

- **Introduction -- What is security?**
- **Cryptography**
 - Classical ciphers
 - Today's ciphers
 - Public-key cryptography
 - Hash functions and MAC
 - Authentication protocols
- **Applications**
 - Digital certificates
 - Secure email
 - Internet security, e-banking
- **Computer and network security**
 - Access control
 - Malware
 - Firewall
- **Examples: Flame, Router, BitCoin ??**



References

- W. Stallings, *Cryptography and network security - principles and practice*, Prentice Hall.
- W. Stallings, 密码学与网络安全：原理与实践（第4版），刘玉珍等译，电子工业出版社，2006
- Lidong Chen, Guang Gong, *Communication and System Security*, CRC Press, 2012.
- A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997, ISBN: 0-8493-8523-7, <http://www.cacr.math.uwaterloo.ca/hac/index.html>
- B. Schneier, *Applied cryptography*. John Wiley & Sons, 1995, 2nd edition.
- 裴定一,徐祥, 信息安全数学基础, ISBN 978-7-115-15662-4, 人民邮电出版社,2007.

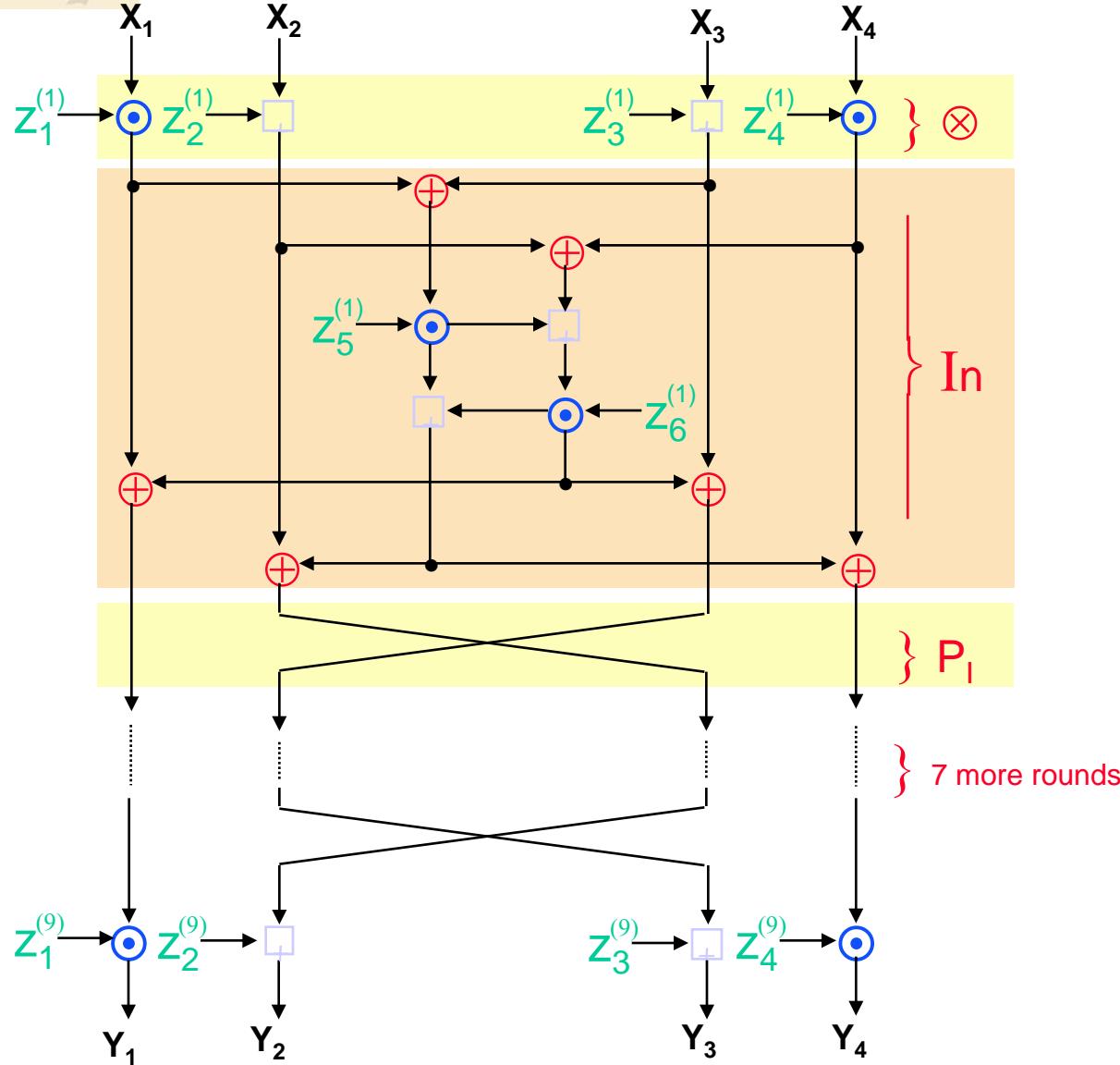


The IDEA cipher

- International Data Encryption Algorithm
- Block length 64-bit, key length 128-bit
- EU Project OASIS (88) (initial)
 - Key length of DES is too short (56 bits)
 - US export restrictions
 - Provable security (crypto is more art than science)
- Lai-Massy, Eurocrypt 90 (PES)
- Lai-Massey-Murphy, Eurocrypt 91 (IPES)
- Naming 92



The IDEA cipher round function



$x_i, y_j, Z_k^{(r)}$: 16-bit subblocks

\oplus : XOR of 16-bit strings

\odot : multiplication mod $2^{16}+1$ of 16-bit integers with $(0\dots 0) \leftrightarrow 2^{16}$

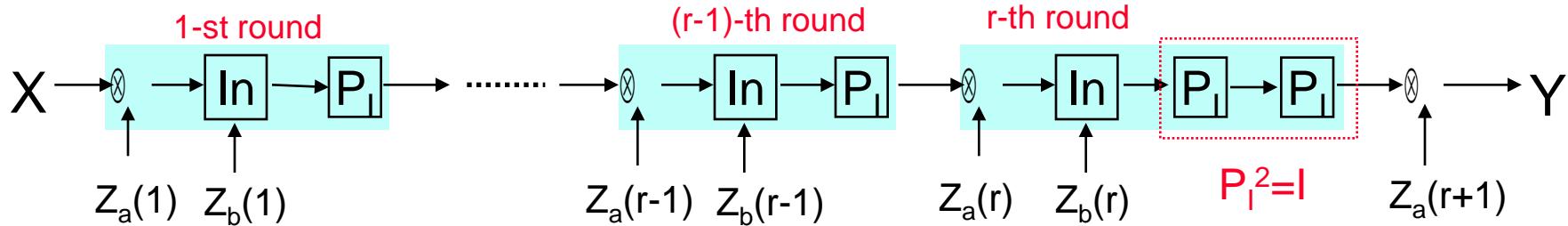
\square : addition mod 2^{16} of 16-bit integers

Eurocrypt'91, Lai, Massey & Murphy:
"Markov ciphers and differential cryptanalysis "

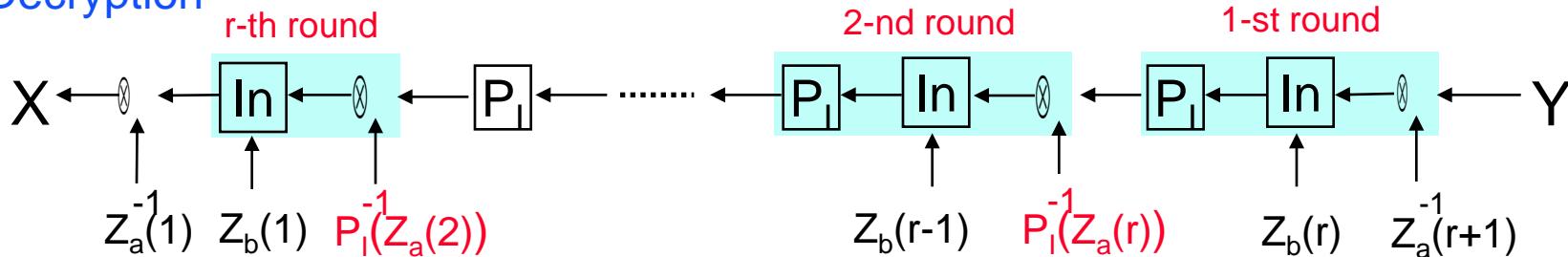


IDEA

Encryption



Decryption



P_I is a homomorphism of the group (F_2^{64}, \otimes) :
 $P_I(\alpha \otimes \beta) = P_I(\alpha) \otimes P_I(\beta)$, $P_I(\alpha^{-1}) = P_I(\alpha)^{-1}$

$$X \otimes Z = (x_1 \odot z_1, x_2 + z_2, x_3 + z_3, x_4 \odot z_4)$$



Key schedule

128-bit key (8 16-bit blocks)

$Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8$

Cyclic-shift to left by 25 bits

$Z_9, Z_{10}, Z_{11}, Z_{12}, Z_{13}, Z_{14}, Z_{15}, Z_{16}$

.....

$Z_{49}, Z_{50}, Z_{51}, Z_{52}$

$Z_1, Z_2, Z_3, Z_4, Z_5, Z_6$
 $Z_7, Z_8, Z_9, Z_{10}, Z_{11}, Z_{12}$
 $Z_{13}, Z_{14}, Z_{15}, Z_{16}, Z_{17}, Z_{18}$
 $Z_{19}, Z_{20}, Z_{21}, Z_{22}, Z_{23}, Z_{24}$
 $Z_{25}, Z_{26}, Z_{27}, Z_{28}, Z_{29}, Z_{30}$
 $Z_{31}, Z_{32}, Z_{33}, Z_{34}, Z_{35}, Z_{36}$
 $Z_{37}, Z_{38}, Z_{39}, Z_{40}, Z_{41}, Z_{42}$
 $Z_{43}, Z_{44}, Z_{45}, Z_{46}, Z_{47}, Z_{48}$
 $Z_{49}, Z_{50}, Z_{51}, Z_{52}$

encryption

$Z_{49}^{-1}, -Z_{50}, -Z_{51}, Z_{52}^{-1}, Z_{47}, Z_{48}$
 $Z_{43}^{-1}, -Z_{45}, -Z_{44}, Z_{46}^{-1}, Z_{41}, Z_{42}$
 $Z_{37}^{-1}, -Z_{39}, -Z_{38}, Z_{40}^{-1}, Z_{35}, Z_{36}$
 $Z_{31}^{-1}, -Z_{33}, -Z_{32}, Z_{34}^{-1}, Z_{29}, Z_{30}$
 $Z_{25}^{-1}, -Z_{27}, -Z_{26}, Z_{28}^{-1}, Z_{23}, Z_{24}$
 $Z_{19}^{-1}, -Z_{21}, -Z_{20}, Z_{22}^{-1}, Z_{17}, Z_{18}$
 $Z_{13}^{-1}, -Z_{15}, -Z_{14}, Z_{16}^{-1}, Z_{11}, Z_{12}$
 $Z_7^{-1}, -Z_9, -Z_8, Z_{10}^{-1}, Z_5, Z_6$
 $Z_1^{-1}, -Z_2, -Z_3, Z_4^{-1}$

decryption



subkey bits

Dependency of subkey bits on the master key bits of IDEA.
i-th round

	$Z_1^{(i)}$	$Z_2^{(i)}$	$Z_3^{(i)}$	$Z_4^{(i)}$	$Z_5^{(i)}$	$Z_6^{(i)}$
1	0–15	16–31	32–47	48–63	64–79	80–95
2	96–111	112–127	25–40	41–56	57–72	73–88
3	89–104	105–120	121–8	9–24	50–65	66–81
4	82–97	98–113	114–1	2–17	18–33	34–49
5	75–90	91–106	107–122	123–10	11–26	27–42
6	43–58	59–74	100–115	116–3	4–19	20–35
7	36–51	52–67	68–83	84–99	125–12	13–28
8	29–44	45–60	61–76	77–92	93–108	109–124
O	22–37	38–53	54–69	70–85		



Group operations

- Design basis: mixing different group operations.
- For both confusion and diffusion
- Having “one-time-pad” security
- Object: n-bit blocks ($n=8, 16, 32$)
- Available: XOR, Add mod 2^n ,
- Integer multiplication: available for most CPU, require Z_p^* , P prime.
- Multiplication mod 2^n+1 is invertible if $n=1,2,4,8,16$ (Fermat primes)
- It is unknown if other Fermat prime exists
- IDEA can have block size of 4, 8, 16, 32, 64 bits (unfortunately not 128).



multiplication

- Example n=2, $Z_5^* = \{1, 2, 3, 4\} \leftrightarrow \{1, 2, 3, 0\} = F_2^2$
- $\{ (00), (01), (10), (11) \} \leftrightarrow \{ 4, 1, 2, 3 \}$, $4=100$
- $2 \odot 3 = 1$, $2 \odot 2 = 0$

$$0 \odot 2 = (4 \times 2 \bmod 5) = (-1 \times 2 \bmod 5) = 3$$

⊕	0	1	2	3
0	1	0	3	2
1	0	1	2	3
2	3	2	0	1
3	2	3	1	0



Efficient computation of Θ

For $n=16$, directly compute $ab \bmod 65537$ is expensive (division).

Low-high algorithm

- $ab \bmod 2^n + 1 =$
$$(ab \bmod 2^n) - (ab \text{ div } 2^n) \quad \text{if } (ab \bmod 2^n) \geq (ab \text{ div } 2^n)$$
$$(ab \bmod 2^n) - (ab \text{ div } 2^n) + 2^n + 1 \quad \text{if } (ab \bmod 2^n) < (ab \text{ div } 2^n)$$
- where $ab \text{ div } 2^n$ is the quotient when ab is divided by 2^n
 - $ab \bmod 2^n$ corresponds to the **lower** n bits of ab
 - $ab \text{ div } 2^n$ is the **higher** n bits of ab
- Because $ab = q(2^n + 1) + r = q2^n + (q+r) = (q+1)2^n + (q+r-2^n)$
- Example: $4 \cdot 8 \bmod 17 = (32 \bmod 17) = (\textcolor{blue}{0010}, \textcolor{red}{0000}) \bmod 17$
$$= (32 \bmod 16) - (32 \text{ div } 16) + 17 = (\textcolor{red}{0000}) - (\textcolor{blue}{0010}) + 17 = 15$$

Exp and log table look-up: $x \cdot y = g^{\log(x) + \log(y)}$

For $n=16$, size of table is $2 \cdot 65536$ bytes

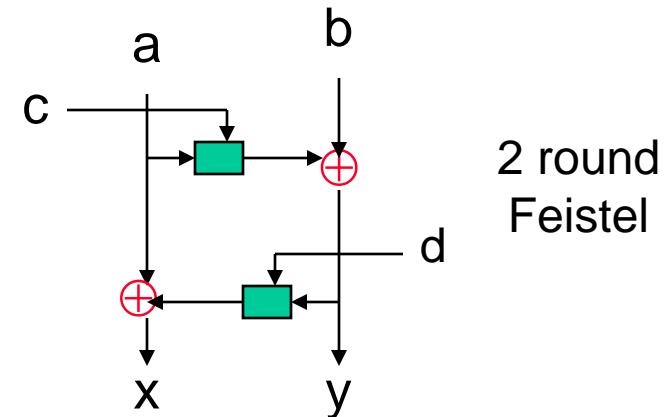
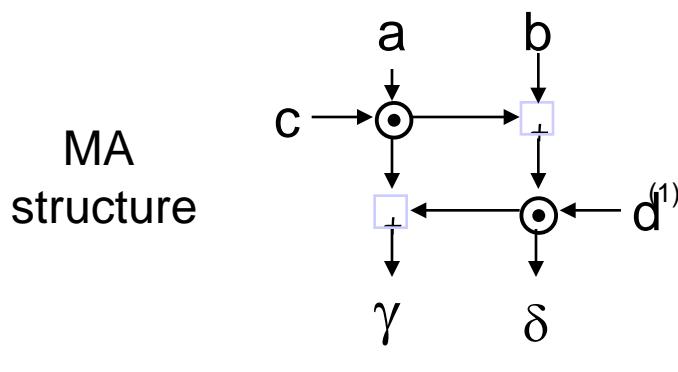


properties

- 3 group operations on 16-bit blocks
- Incompatible: non-associative, non-distributive
- Non-isotopic:
 - Isotopic: exist f,g,h , s.t., $f(a^*b)=g(a)\#h(b)$
- Confusion
 - Interaction of 3 operations
 - Consecutive operations are different
- Diffusion
 - MA structure, In
 - Complete in 1 round (each input-bit influences every output bit)

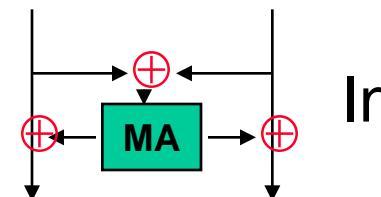


MA and In



MA structure uses the least number of operations (4) to achieve ‘complete diffusion’ – each output depends on every input

Involution **In**: $\text{In}^2 = \text{identity}$



- In can be viewed as 2 round Feistel structure
- Thus, 1 round of IDEA is more than 2 rounds Feistel
- IDEA has 8.5 rounds



Known attacks

Attacks on reduced IDEA (total 8.5 rounds)

rounds	data	process	(memory)	attacks
2.5	2^{10}	2^{106}		differential (Meier 92)
2.5	2	2^{37}		square (Nakahara-Barreto-Preneel 02)
3	2^{22}	2^{50}		linear (Junod, FSE05)
3.5	2^{56}	2^{67}		truncated diff.(Borst-Knudsen-Rijmen 97)
3.5	103	2^{97}		linear (Junod, FSE05)
4	2^{37}	2^{70}		impossible (Biham-Birykov-Shamir 99)
4.5	$\textcolor{blue}{2^{64}}$	2^{112}		impossible differential (Alix-Biham-Shamir 98)
4.5	2^{24}	2^{121}	(2^{64})	collision (Demirci-Ture-Selcuk, SAC03)
5	2^{24}	2^{126}	(2^{64})	collision (Demirci-Ture-Selcuk, SAC03)
5	2^{19}	2^{103}		Biham-Dunkelman-Keller, AC06
6	2^{49}	2^{112}		differential-linear (Sun-Lai, AC09)
6	2	$2^{123.4}$		Meet-in-the-Middle (Keller,Biham,,C11)
8.5	$\textcolor{red}{2^{52}}$	$\textcolor{red}{2^{126.06}}$		biclique(Khovratovich-Lurent-Rechberg,EC12)
Max	2^{64}	2^{127}		



Other issues

- No S-box, so nothing to hide
- Weak-keys:
 - Special value ‘0 (-1)’ and ‘1’ have less confusion and diffusion effect: $0 \oplus x = x$, $0 \otimes x = -x$, $1 \otimes x = x$
 - Linear key schedule
 - Sets of weak keys of size about 2^{51} [Daemen 94], 2^{63} [Hawks 98], 2^{63} [Biryukov 02]
 - Simple fix: XOR a constant to subkeys
- Obtain non-standard but stronger version of IDEA.
- 128-bit version: MESH, IDEA-NXT, new ones?



AES – Advanced Encryption Standard

- Block cipher, **128-bit block**; 128,194,256-bit key
- Fast for SW and 8-bit processor
- **More secure and faster than DES?**
- 1997-04: requirements (128-bit?, free?,...)
- 1997-10: NIST 1-st call
- 1998-08: 1-st AES Conference, Ventura, USA
 - 15 accepted submissions
- 1999-03: 2-nd AES Conference, Rome
- 1999-8: five final candidates
- 2000-03: 3-rd AES Conference, New York
- 2000-10-02: decision -- Rijndael
- 2001-11: published as FIPS PUB 197



AES candidates

- **CAST-256** Entrust Tech. (rep. Carlisle Adams)
- **CRYPTON** Future Systems, Inc. (rep Chae Hoon Lim)
- **DEAL** Richard Outerbridge, Lars Knudsen (attack 2^{70})
- **DFC** CNRS - Ecole Normale Supérieure (rep Serge Vaudenay)
- **E2** NTT - (represented by Masayuki Kanda)
- **FROG** TecApro Int. S.A. (rep Dianelos Georgoudis) - attack (2^{56})
- **HPC** Rich Schroepel [\(???\)](#)
- **LOKI97** Lawrie Brown, Josef Pieprzyk, Jennifer Seberry -Attacks known (2^{56})
- **MAGENTA** Deutsche Telekom (Klaus Huber) [broken](#): trivial chosen plaintext; other 2^{56}
- **MARS** IBM (represented by Nevenko Zunic) [some weakness](#)
- **RC6** RSA Laboratories (rep Matthew Robshaw)
- **RIJNDAEL** **Joan Daemen, Vincent Rijmen**
- **SAFER+** Cylink Corporation (rep Lily Chen)
- **SERPENT** Ross Anderson, Eli Biham, Lars Knudsen
- **TWOFISH** B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson



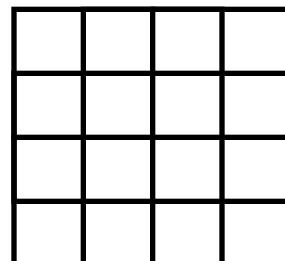
AES parameters

- Number of rounds 10 / 12 / 14
- Keysize: 128/192/256 bit keys

Unit: 32-bit words

Key Length (Nk words)	Block Size (Nb words)	Number of Rounds (Nr)
AES-128	4	4
AES-192	6	4
AES-256	8	4

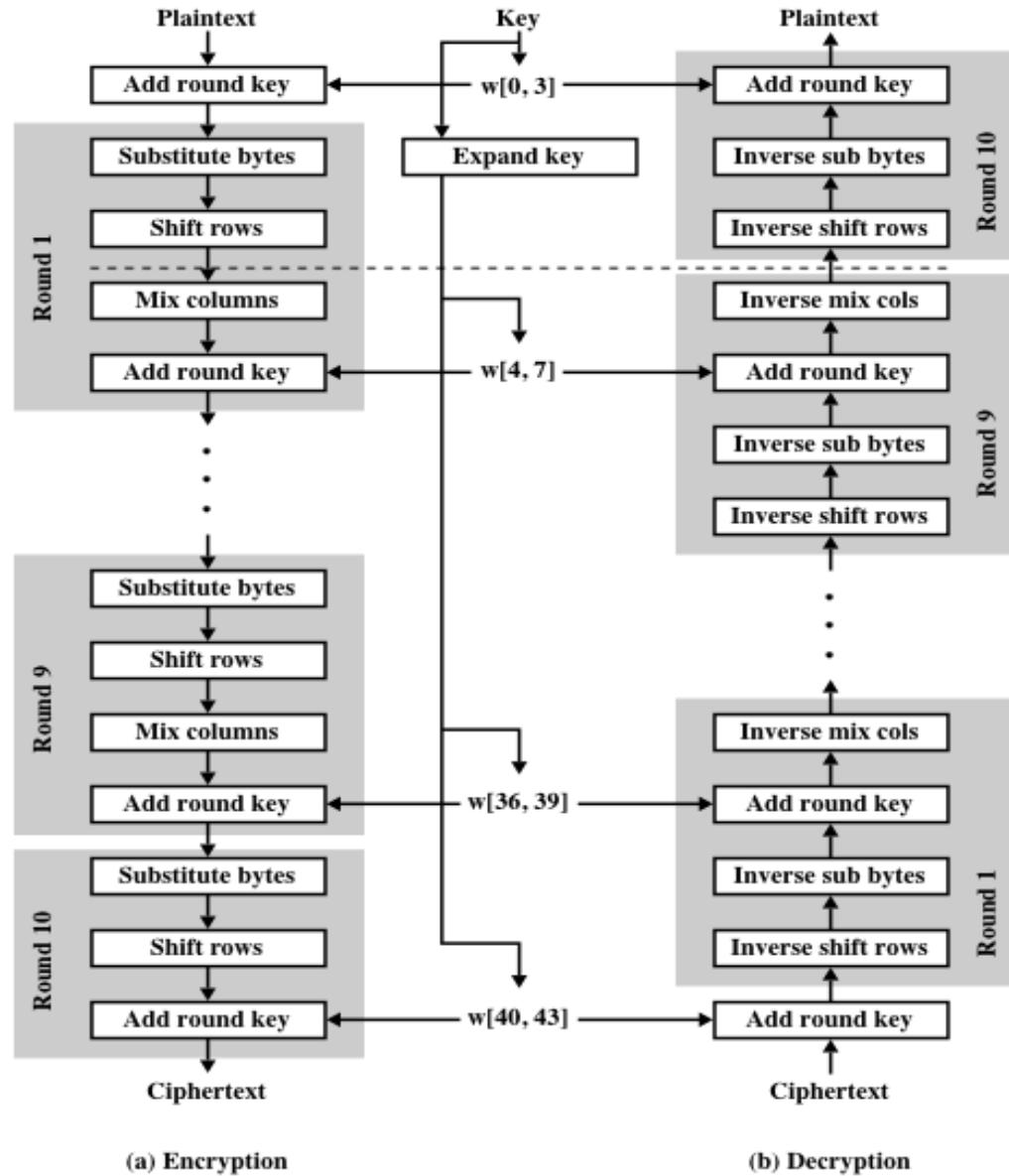
- Text: 128-bit data, represented as 4 by 4 matrix of 8-bit bytes.





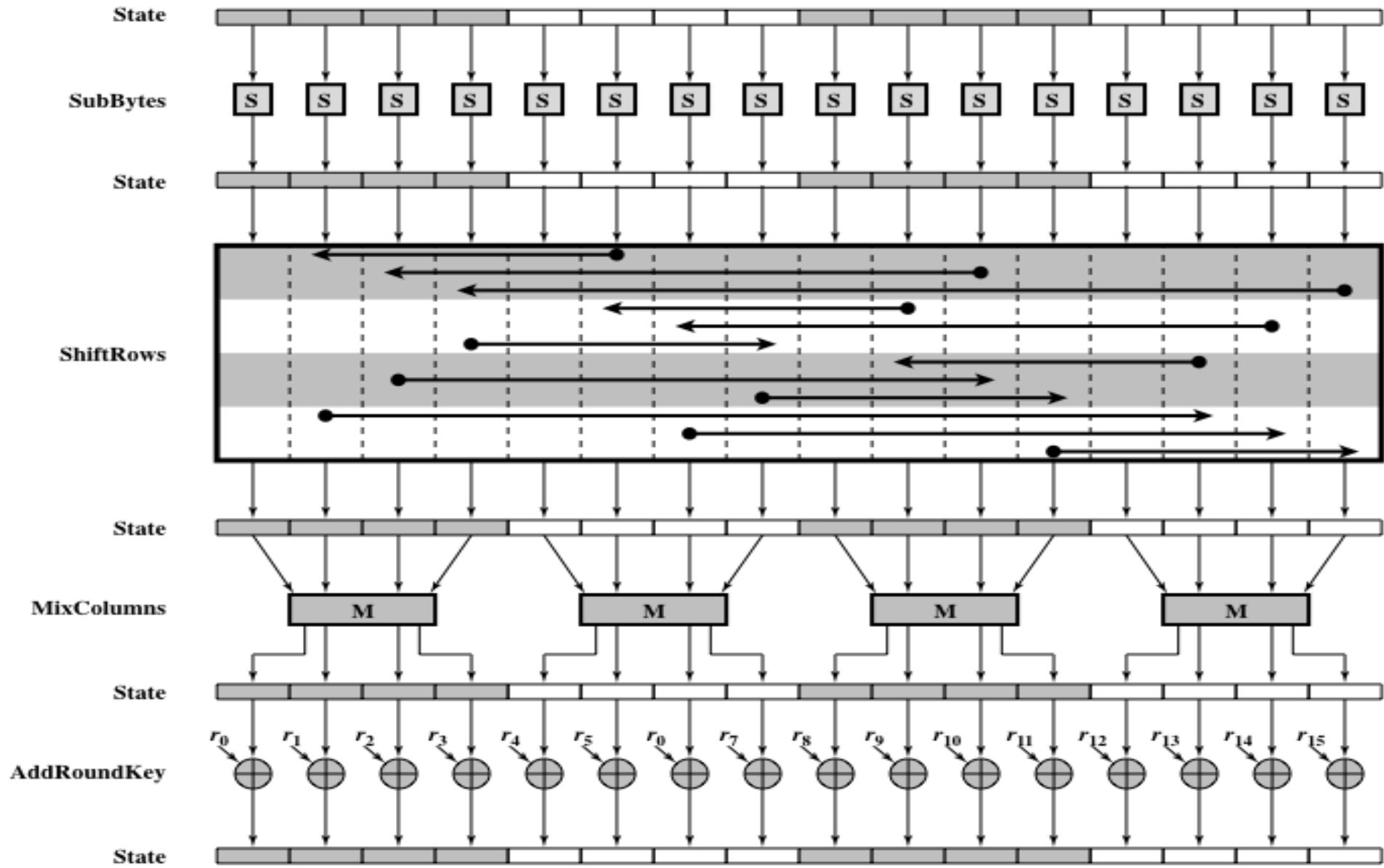
AES Encryption and Decryption

- Add key
- S-box
- Shift row
- Mix column
- Add key-0
- S-box
- Shift row
- Mix column
- Add key-1
- -
- -
- S-box
- Shift row
- Add key-last



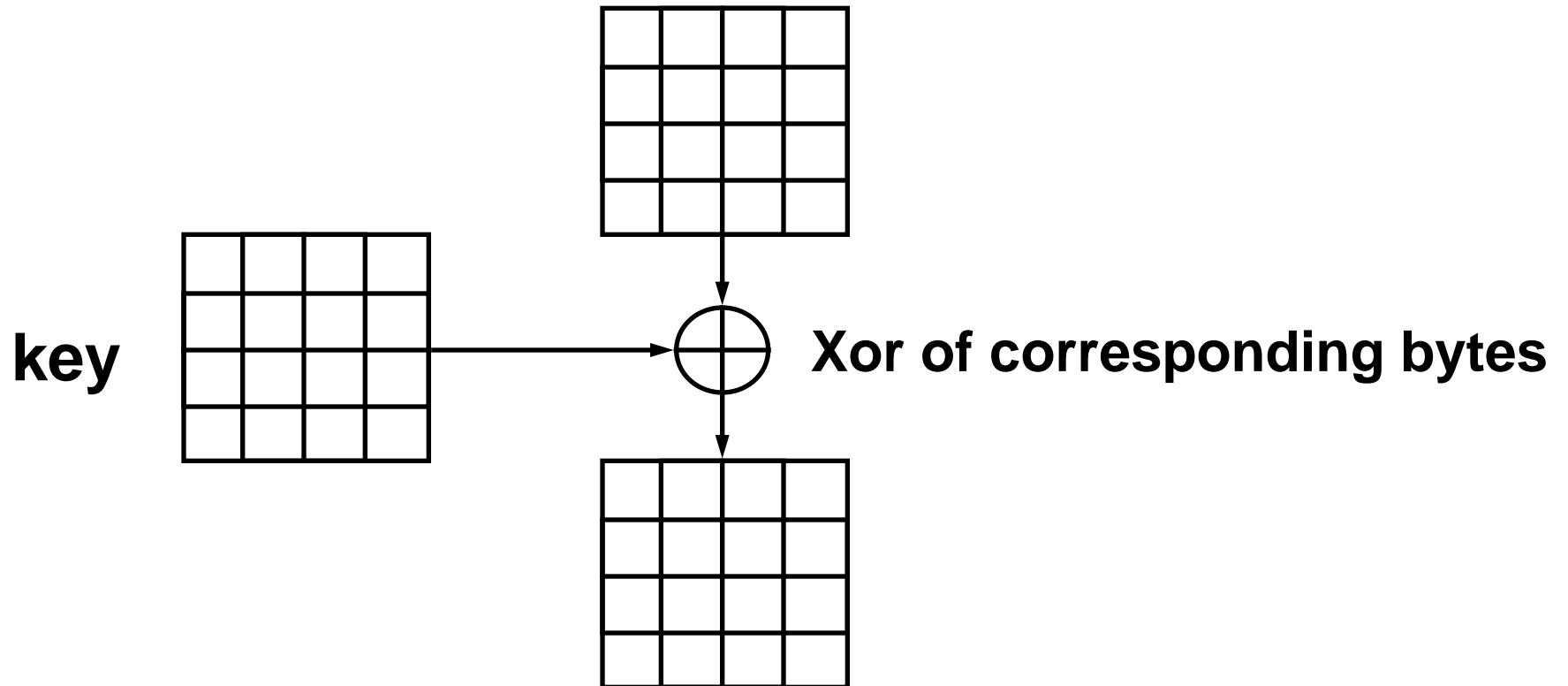


AES Round





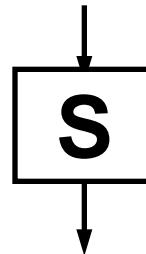
Add key operation





S-box

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}



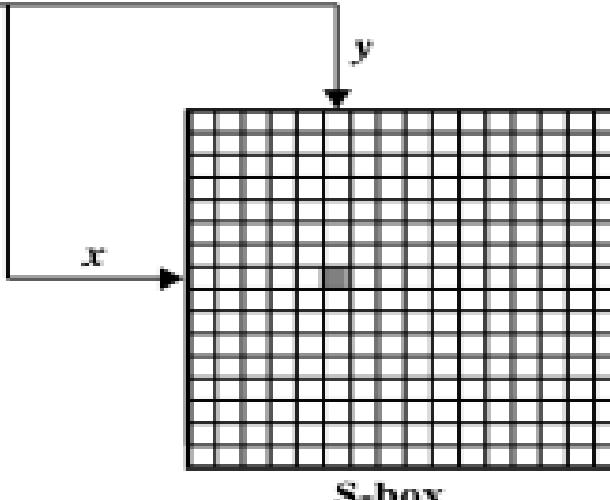
- **8-bit lookup table**
- **16 lookups in parallel**

$S(B_{00})$	$S(B_{01})$	$S(B_{02})$	$S(B_{03})$
$S(B_{10})$	$S(B_{11})$	$S(B_{12})$	$S(B_{13})$
$S(B_{20})$	$S(B_{21})$	$S(B_{22})$	$S(B_{23})$
$S(B_{31})$	$S(B_{31})$	$S(B_{32})$	$S(B_{33})$



Use of S-box

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s'_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$



- Simple, **non-linear substitution** of byte
- 16x16-bytes S-box contains permutation of all 256 8-bit values

$s'_{0,0}$	$s'_{0,1}$	$s'_{0,2}$	$s'_{0,3}$
$s'_{1,0}$	$s'_{1,1}$	$s'_{1,2}$	$s'_{1,3}$
$s'_{2,0}$	$s'_{2,1}$	$s'_{2,2}$	$s'_{2,3}$
$s'_{3,0}$	$s'_{3,1}$	$s'_{3,2}$	$s'_{3,3}$

- each byte of state is replaced by byte in matrix
 - left 4 bits -> row
 - right 4 bits -> column

Substitution: two-dimensional table look-up



S-box

$S(x,y)$		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	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

byte {95} is replaced by row 9, column 5 (is {2A})



Inverse S-box

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	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



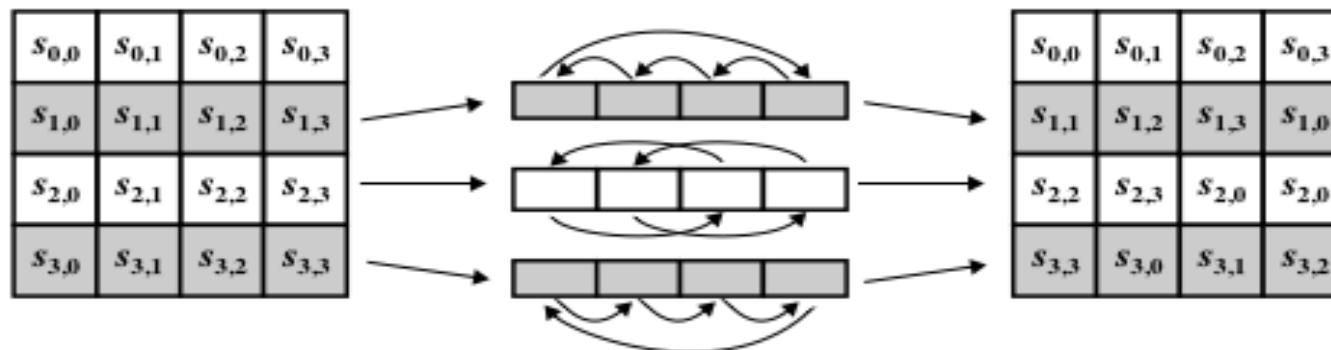
Rationale for S-box Design

- low correlation between input and output bits
- output is no simple function of input
- S-box has no fixed points, i.e., $S(a) \neq a$
- S-box is not self-inverse, i.e., $S(a) \neq \text{Inv}S(a)$
- The mapping $x \rightarrow x^{-1}$ has high non-linear degree and good differential distribution.



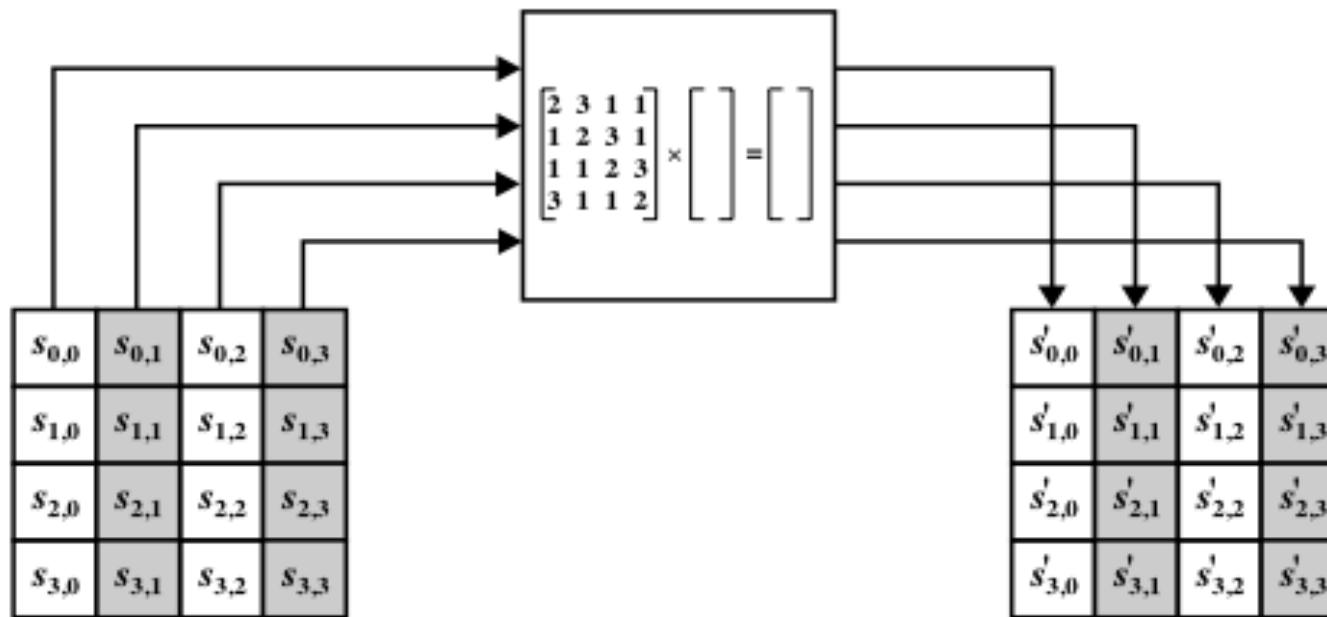
Shift Row Transformation

- a circular byte shift in each row
 - 1st row is unchanged
 - 2nd row does 1 byte circular shift to left
 - 3rd row does 2 byte circular shift to left
 - 4th row does 3 byte circular shift to left
- decrypt does shifts to right
- this step permutes bytes between the columns





Mix Column Transformation





MDS matrix

- A 4×4 matrix over $\text{GF}(2^8)$.
- Matrix is an MDS (Maximum Distance Separable).
- Byte-Hamming weight of input + output is at least 5.

Input weight	Output weight
1	4
2	≥ 3
3	≥ 2
4	≥ 1

- High diffusion – effective against differential and linear attacks



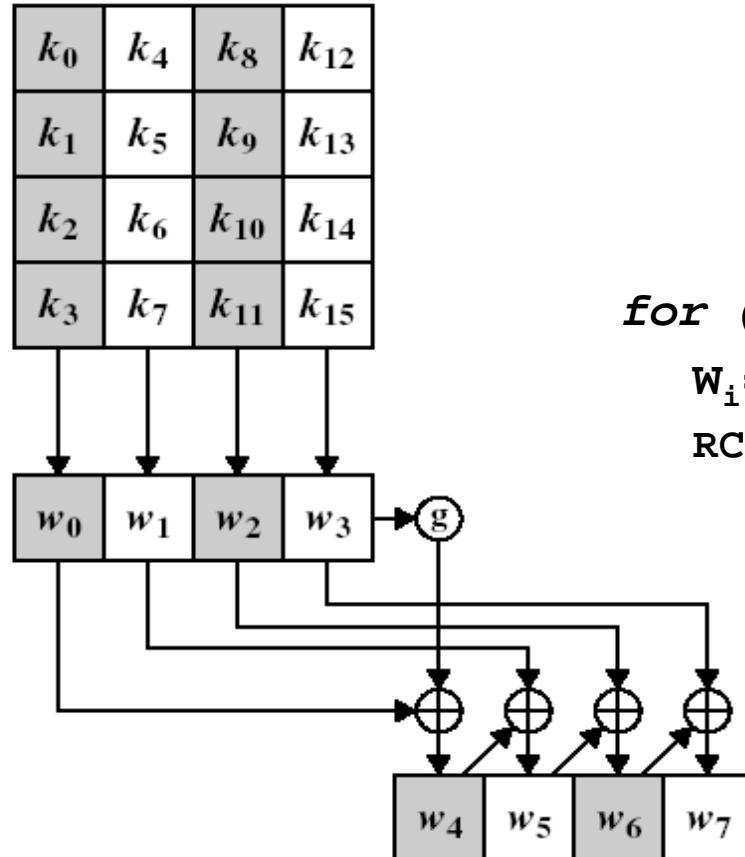
Inverse Mix Column Transformation

- just like Mix Column Transformation
- however, each column is multiplied modulo x^4+1 with fixed polynomial ‘0B’ x^3 + ‘0D’ x^2 + ‘09’ x + ‘0E’
- same as:

$$\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$



AES Key Expansion



128bit = 4word Key \Rightarrow 4*11word subkey
192bit=6word Key \Rightarrow 4*13word subkey
256bit=8word Key \Rightarrow 4*15word subkey

for ($i \bmod 4 = 0$)

$$W_i = W_{i-4} \oplus \text{Sub}(\text{RotWord}(W_{i-1})) \oplus \text{RCON}(i)$$
$$\text{RCON}(i) = 2^{(i-4)/4} = 1, 2, 4, 8, \dots$$

RCON

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Figure 5.6 AES Key Expansion



AES Decryption

Decryption process is different from encryption process

- Inverse S-box.
- Inverse of MDS matrix.
- Modified round keys, or modified operation order.
- Requires extra hardware.

Decryption key

- Cannot directly generate round keys in reverse order.
- Decryption must either store all round keys, or pre-compute the ‘final’ state and work backwards from that.
- Requires extra time from getting key to start of first decryption.



Implementation

- on **8-bit CPU**
 - byte substitution works on bytes using a table of 256 entries
 - shift rows is simple byte shifting
 - add round key works on byte XORs
 - mix columns requires matrix multiply in $\text{GF}(2^8)$ which works on byte values, can be simplified to use a table lookup
- on **32-bit CPU**
 - redefine steps to use 32-bit words
 - can pre-compute 4 tables of 256-words
 - each column in each round can be computed using 4 table lookups + 4 XORs
 - at a cost of 16Kb to store tables
- designers believe this efficient implementation was a key factor in its selection as the AES cipher
- Round function is embedded in new Intel CPU



Security

- Impossible Differential attack on 7-round: $2^{112}, 2^{112}, 2^{117}$
- Related-key attack on full AES [AC09].
- BiClique Attacks on full AES: complexity $2^{\{k-1.3\}}$, for k=128, 192, 256. [AC 2011]
- Algebraic structures: BES, extended to a larger space $\text{GF}(2^8)$, easy to analyze. [Murphy-Robshaw, Crypto02]
- Algebraic attacks [Courteous-Pieprzyk, AC02]: written as an over-defined system of multivariate quadratic equations (MQ), solvable using XSL[Shamir, EC00];
 - claimed to be able to attack BES in about 2^{87} or 2^{100} operations??
 - Algebraic attacks may not work as expected [Cid-Leurent, AC05]
- Linearity and slow diffusion in key schedule



Exercise 5.

- 1. prove the low-high algorithm for computing ◎**
- 2. prove that the In-structure in IDEA is an involution.**

Deadline: before next lecture
Send it to: Geshi Huang
gracehgs@mail.sjtu.edu.cn