T-79.4501
Cryptography and Data Security

Lecture 3:

Polynomial arithmetic

- Groups, rings and fields
- Polynomial arithmetic

Block ciphers

- DES
- IDEA
- AES

Stallings: Chapters 3, 4.5, 5

Axioms: Group

Group $(G,*)$: A set G, with operation $*$.

Additive group: "$*$" is addition +

Multiplicative group: "$*$" is multiplication ·

Axiom 1: G is closed under the operation $*$, that is, given $a \in G$ and $b \in G$, then $a*b \in G$.

Axiom 2: Operation $*$ is associative, that is, given $a \in G$, $b \in G$ and $c \in G$, then $(a*b)*c = a*(b*c)$.

Axiom 3: $(G,*)$ has an identity element, that is, an element $e \in G$ such that $a*e = e*a = a$, for all $a \in G$. Then e is denoted by 1 (general and multiplicative case), or by 0 (additive case)

Axiom 4: Every element has an inverse, that is, given $a \in G$ there is a unique $b \in G$ such that $a*b = b*a = e$. Then b is denoted by a^{-1} (general or multiplicative case) or by $-a$ (additive case).

Axioms: Abelian Group

Axiom 5: Group $(G,*)$ is Abelian group (or commutative group) if the operation $*$ is commutative, that is, given $a \in G$ and $b \in G$, then $a*b = b*a$.

Axioms: Ring $(R,+,\cdot)$

A set R with two operations + and · is a ring if the following eight axioms hold:

A1: Axiom 1 for +

A2: Axiom 2 for +

A3: Axiom 3 for +

A4: Axiom 4 for +

A5: Axiom 5 for +

(A1–A5: $(R,+)$ is an Abelian Group)

M1: Axiom 1 for ·

M2: Axiom 2 for ·

M3: Distributive laws hold: given $a \in G, b \in G$ and $c \in G$, then $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(a + b) \cdot c = a \cdot c + b \cdot c$.

Axioms: Commutative Ring and Field

A ring $(R,+,\cdot)$ is said to be commutative if

M4: Axiom 5 for multiplication holds

A commutative ring $(F,+,\cdot)$ is a field if :

M5: Axiom 3 for · in F-$\{0\}$ holds: $a*1 = 1*a = a$, for all $a \in F$, $a \neq 0$.

M6: Axiom 4 for · in F-$\{0\}$ holds: given $a \in F$, $a \neq 0$, there is a unique $a^{-1} \in F$ such that $a* a^{-1} = a^{-1}*a = 1$.

If $(F,+,\cdot)$ is a field, then $F^* = F$-$\{0\}$ with multiplication is a group.

Example: p prime, then $Z_p = \{a \mid 0 \leq a < p\}$ with modulo p addition and multiplication is a field and $(Z_p{}^*, \cdot)$ is a group.

Binary arithmetic:

> There are exactly 10 types of people, those who understand binary arithmetic and those who don’t.
>
> (Doug Stinson’s home page)

Polynomial Arithmetic

- Modular arithmetic with polynomials
- We limit to the case where polynomials have binary coefficients, that is, 1+1 = 0, and + is the same as -.

Example:

$$(x^2 + x + 1)(x^3 + x + 1) =$$

$$x^5 + x^3 + x^2 + x^4 + x^2 + x + x^3 + x + 1 =$$

$$x^5 + x = x \cdot (x^4 + 1) = x \cdot x = x^2 (\mathrm{mod}(x^4 + x + 1))$$

Computation $\mathrm{mod}(x^4 + x + 1)$ means that everywhere we take $x^4 + x + 1 = 0$, for example, we can take $x^4 + 1 = x$.

Galois Field

Given a binary polynomial $f(x)$ of degree n, consider a set of binary polynomials with degree less than n. This set has 2^n polynomials. With polynomial arithmetic modulo $f(x)$ this set is a ring.

Fact: If $f(x)$ is irreducible, then this set with 2-ary (binary) polynomial arithmetic is a field denoted by $\mathrm{GF}(2^n)$.

In particular, every nonzero polynomial has a multiplicative inverse modulo $f(x)$. We can compute a multiplicative inverse of a polynomial using the Extended Euclidean Algorithm.

The next slide presents the Extended Euclidean Algorithm for integers. It works exactly the same way for polynomials.

Extended Euclidean Algorithm for integers and computing a modular inverse

Fact: Given two positive integers a and b there exist integers u and v such that

$$u \cdot a + v \cdot b = \gcd(a,b)$$

In particular, if $\gcd(a,b)$ =1, there exist positive integers u and v such that

$$u \cdot a = 1 \pmod{b}, \text{ and } v \cdot b = 1 \pmod{a}.$$

The integers u and v can be computed using the Extended Euclidean Algorithm, which iteratively finds integers r_i , u_i and v_i such that

$$r_0 = b, \quad r_1 = a; \quad u_0 = 0, \quad u_1 = 1; \; v_0 = 1, \; v_1 = 0$$

and for $i = 2,3,\dots$ we compute q_i such that

$$r_{i-2} = q_i \cdot r_{i-1} + r_i \text{, where } 0 \le r_i < r_{i-1} .$$

We set: $u_i = u_{i-2} - q_i \cdot u_{i-1}$ and $v_i = v_{i-2} - q_i \cdot v_{i-1}$. Then $r_i = u_i \cdot a + v_i \cdot b$.

Let n be the index for which $r_n > 0$ and $r_{n+1} = 0$. Then

$$r_n = \gcd(a,b) \text{ and } u_n = u \text{ and } v_n = v.$$

Extended Euclidean Algorithm: Example

gcd(595,408) = 17 = $u \times 595 + v \times 408$

i	q_i	r_i	u_i	v_i
0	-	595	1	0
1	-	408	0	1
2	1	187	1	-1
3	2	34	-2	3
4	5	17	11	-16

Extended Euclidean Algorithm: Examples

gcd(595,408) = 17 = 11×595 + (-16)×408

= -397×595 + 579×408

We get 11×595 = 17 (mod 408)

and 579×408 = 17 (mod 595)

If gcd(a,b) = 1, this algorithm gives modular inverses.

Example: 557×797 = 1 (mod 1047) that is

557 = 797^{-1} (mod 1047)

If gcd(a,b) = 1, the integers a and b are said to be coprime.

Extended Euclidean Algorithm for polynomials

Example

Example: Compute the multiplicative inverse of x^2 modulo $x^4 + x + 1$

i	q_i	r_i	u_i	v_i
0		$x^4 + x + 1$	0	1
1		x^2	1	0
2	x^2	$x + 1$	x^2	1
3	x	x	$x^3 + 1$	x
4	1	1	$x^3 + x^2 + 1$	$x + 1$

Extended Euclidean Algorithm for polynomials

Example cont'd

So we get

$$u_4 \cdot x^2 + v_4 \cdot (x^4 + x + 1) = (x^3 + x^2 + 1)x^2 + (x + 1)(x^4 + x + 1) = 1 = r_4$$

from where the multiplicative inverse of x^2 mod $x^4 + x + 1$ is equal to $x^3 + x^2 + 1$.

Motivation for polynomial arithmetic:

- uses all n-bit numbers (not just those less than some prime p)
- provides uniform distribution of the multiplication result

Example: Modulo 2^3 arithmetic compared to $GF(2^3)$ arithmetic (multiplication).

In $GF(2^n)$ arithmetic, we identify polynomials of degree less than n:

$$a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$$

with bit strings of length n : $(a_{n-1}, a_{n-2}, \ldots, a_1, a_0)$

and further with integers less than 2^n:

$$a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_2 2^2 + a_1 2 + a_0$$

Example: In $GF(2^3)$ arithmetic with polynomial $x^3 + x + 1$ (see next slide) we get:

$4 \cdot 3 = (100) \cdot (011) = x^2 \cdot (x+1) = x^3 + x^2 = (x+1) + x^2 = x^2 + x + 1 = (111) = 7$

Multiplication tables

modulo 8 arithmetic

	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	0	2	4	6
3	3	6	1	4	7	2	5
4	4	0	4	0	4	0	4
5	5	2	7	4	1	6	3
6	6	4	2	0	6	4	2
7	7	6	5	4	3	2	1

$GF(2^3)$ Polynomial arithmetic

	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	3	1	7	6
3	3	6	5	7	4	1	2
4	4	3	7	6	2	5	1
5	5	1	4	2	7	3	6
6	6	7	1	5	3	2	4
7	7	5	2	1	6	4	3

Block ciphers

Confidentiality primitive

- Threat: recover the plaintext from the ciphertext without the knowledge of the key.
- Security goal: protect against this threat.

Plaintext P: strings of bits of fixed length n

Ciphertext C: strings of bits of the same length n

Key K: string of bits of fixed length k

Encryption transformations: For each fixed key the encryption operation E_K is one-to-one (invertible) function from the set of plaintexts to the set of ciphertext. That is, there exist an inverse transformation, decryption transformation D_K such that for each P and K we have: $D_K(E_K(P)) = P$

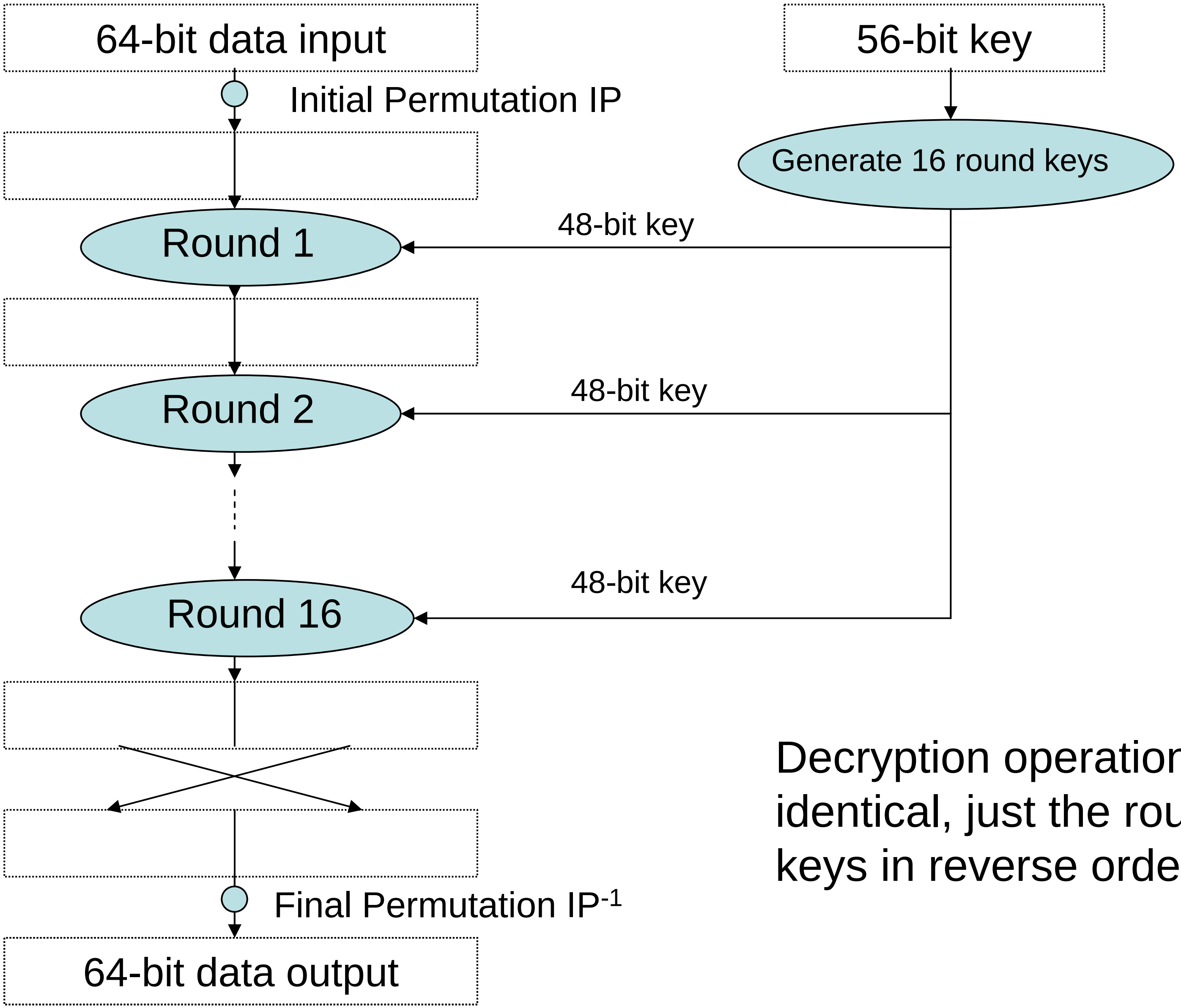
Block ciphers, design principles

- The ultimate design goal of a block cipher is to use the secret key as efficiently as possible.
- Confusion and diffusion (Shannon)
- New design criteria are being discovered as response to new attacks.
- A state-of-the-art block cipher is constructed taking into account all known attacks and design principles.
- But no such block cipher can become provably secure, it may remain open to some new, unforeseen attacks.
- Common constructions with iterated round function
  - Substitution permutation network (SPN)
  - Feistel network

DES Data Encryption Standard 1977 - 2002

- Standard for 25 years
- Finally found to be too small. DES key is only 56 bits, that is, there are about 10^{16} different keys. By manufacturing one million chips, such that, each chip can test one million keys in a second, then one can find the key in about one minute.
- The EFF DES Cracker built in 1998 can search for a key in about 4,5 days. The cost of the machine is $250 000.
- DES has greately contributed to the development of cryptologic research on block ciphers.
- The design was a joint effort by NSA and IBM. The design principles were not published until little-by-little. The complete set of design criteria is still unknown.
- Differential cryptanalysis 1989
- Linear cryptanalysis 1993

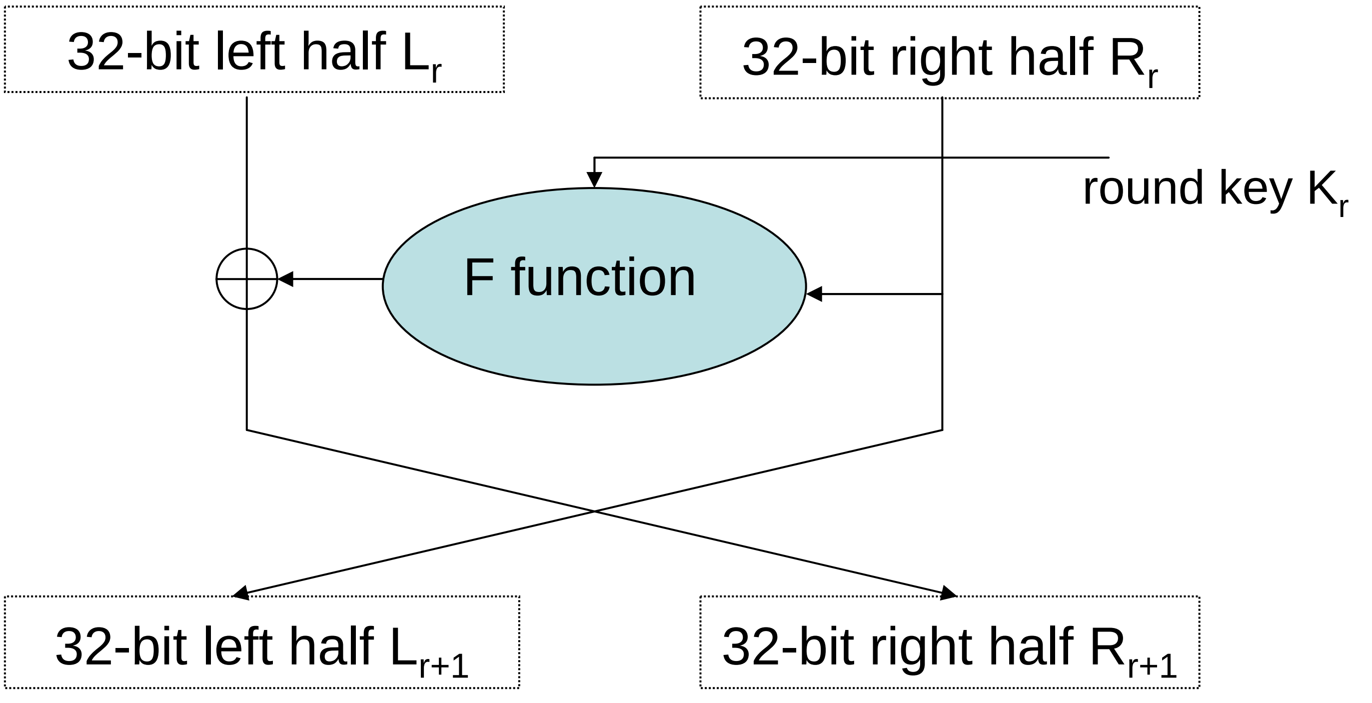
DES encryption operation overview



Decryption operation is identical, just the round keys in reverse order

DES round function

Round function is its own inverse (involution):

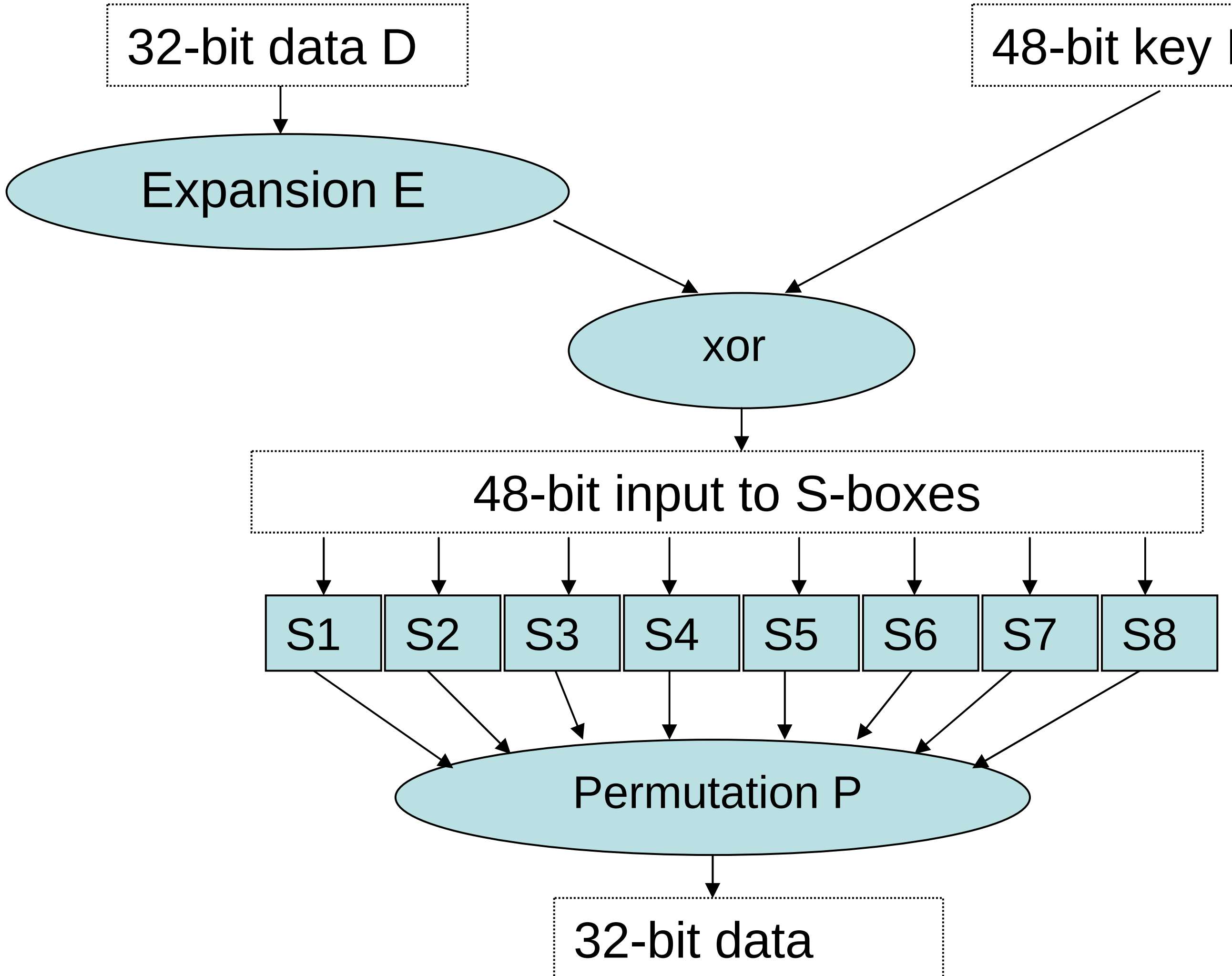


$$L_{r+1} = R_r$$

$$R_{r+1} = L_r \text{ xor } F(R_r, K_r)$$

The F-function of DES

$$F(D;K) = P(S(E(D) \text{ xor } K))$$



The DES S-boxes

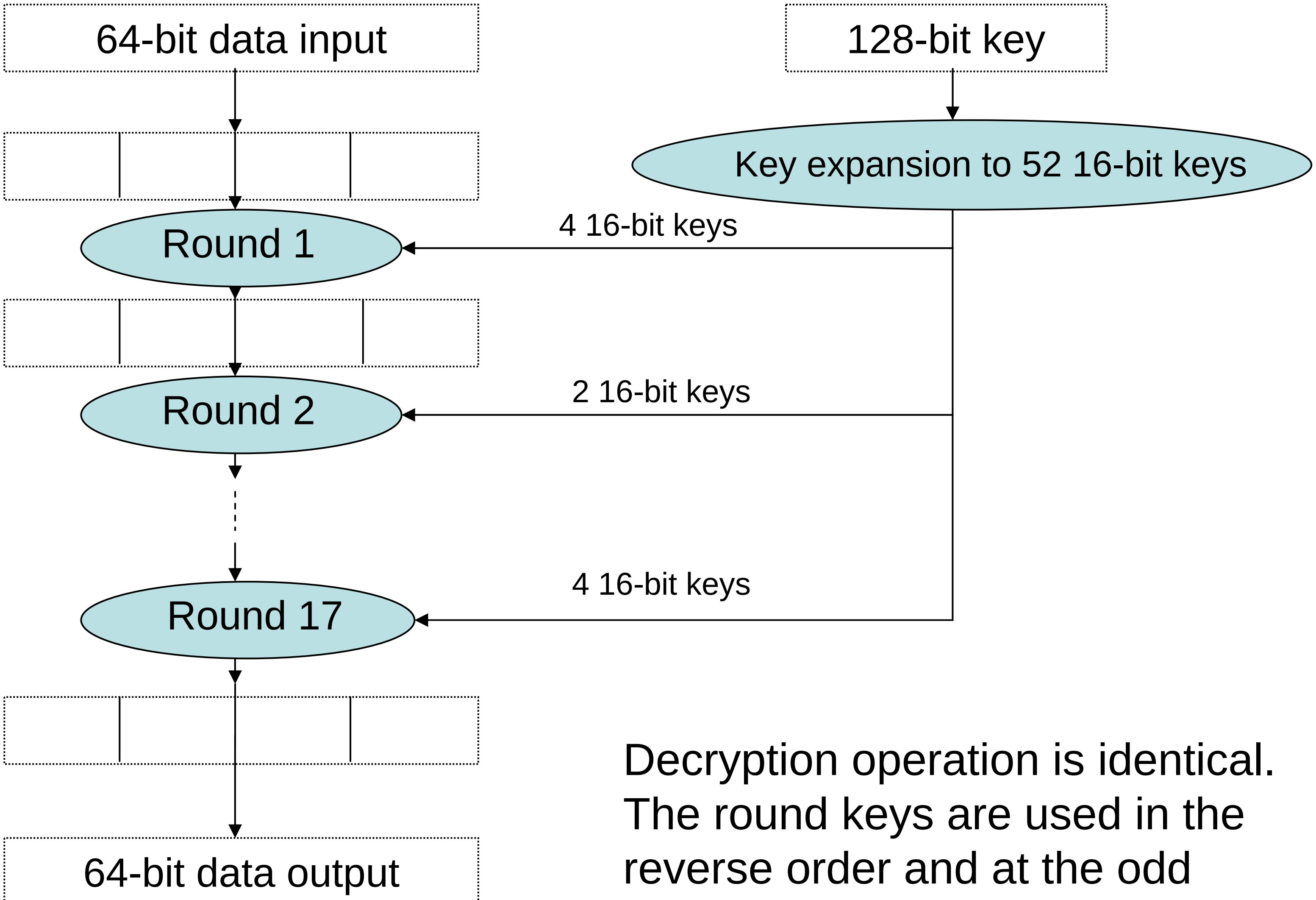
- Small 6-to-4-bit functions
- Given in tables with four rows and 16 columns
- Input data   a1,a2,a3,a4,a5,a6
- The pair of bits a1,a6 point to a row in the S-box
- Given the row, the middle four bits point to a position from where the output data is taken.

Example: S-box S_4

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

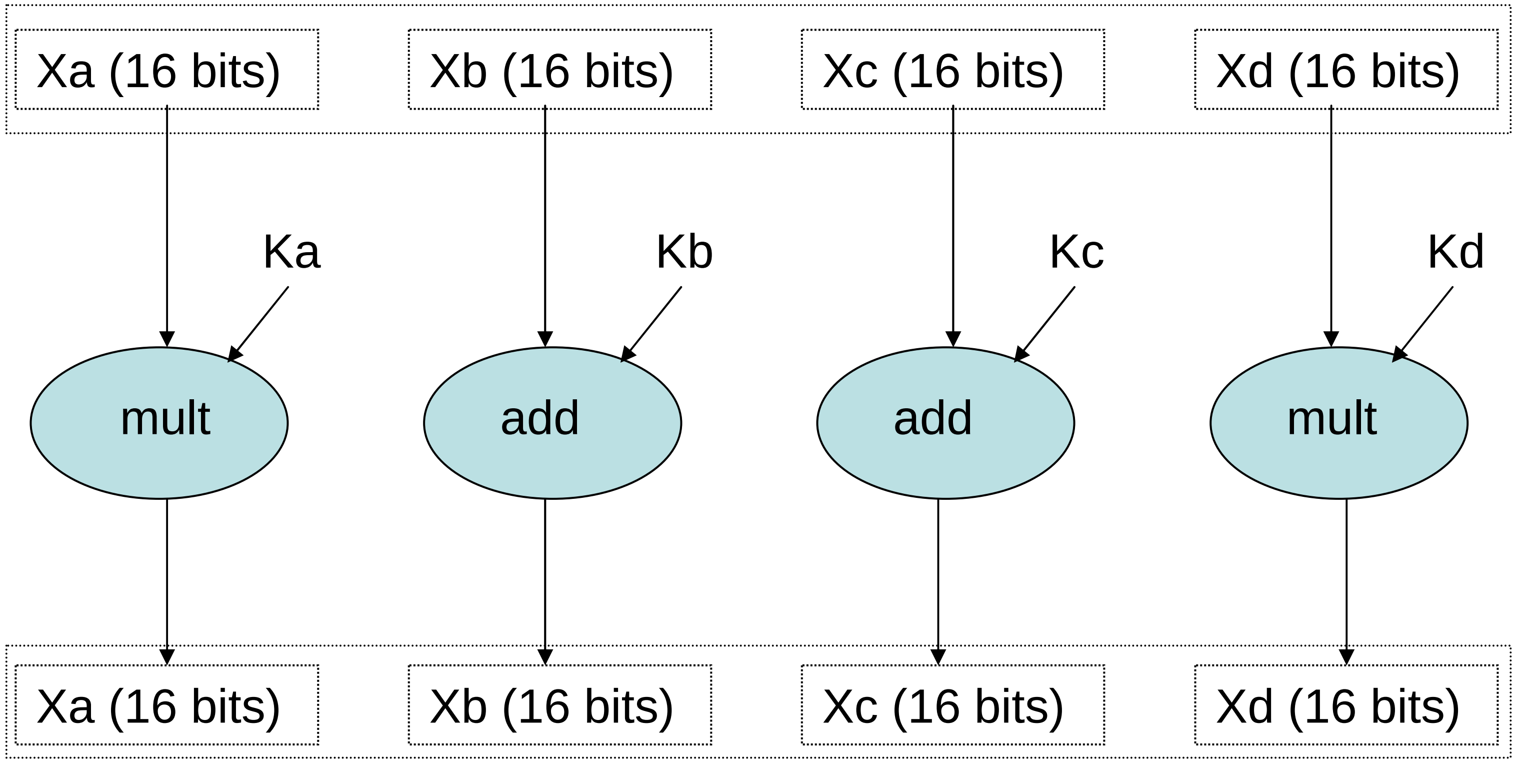
- S-boxes are the only source of nonlinearity in DES. Their nonlinearity properties are extensively studied.

IDEA encryption operation overview

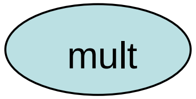


Decryption operation is identical. The round keys are used in the reverse order and at the odd rounds replaced by the inverse values.

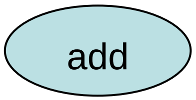
One round of IDEA: odd round



Legend:



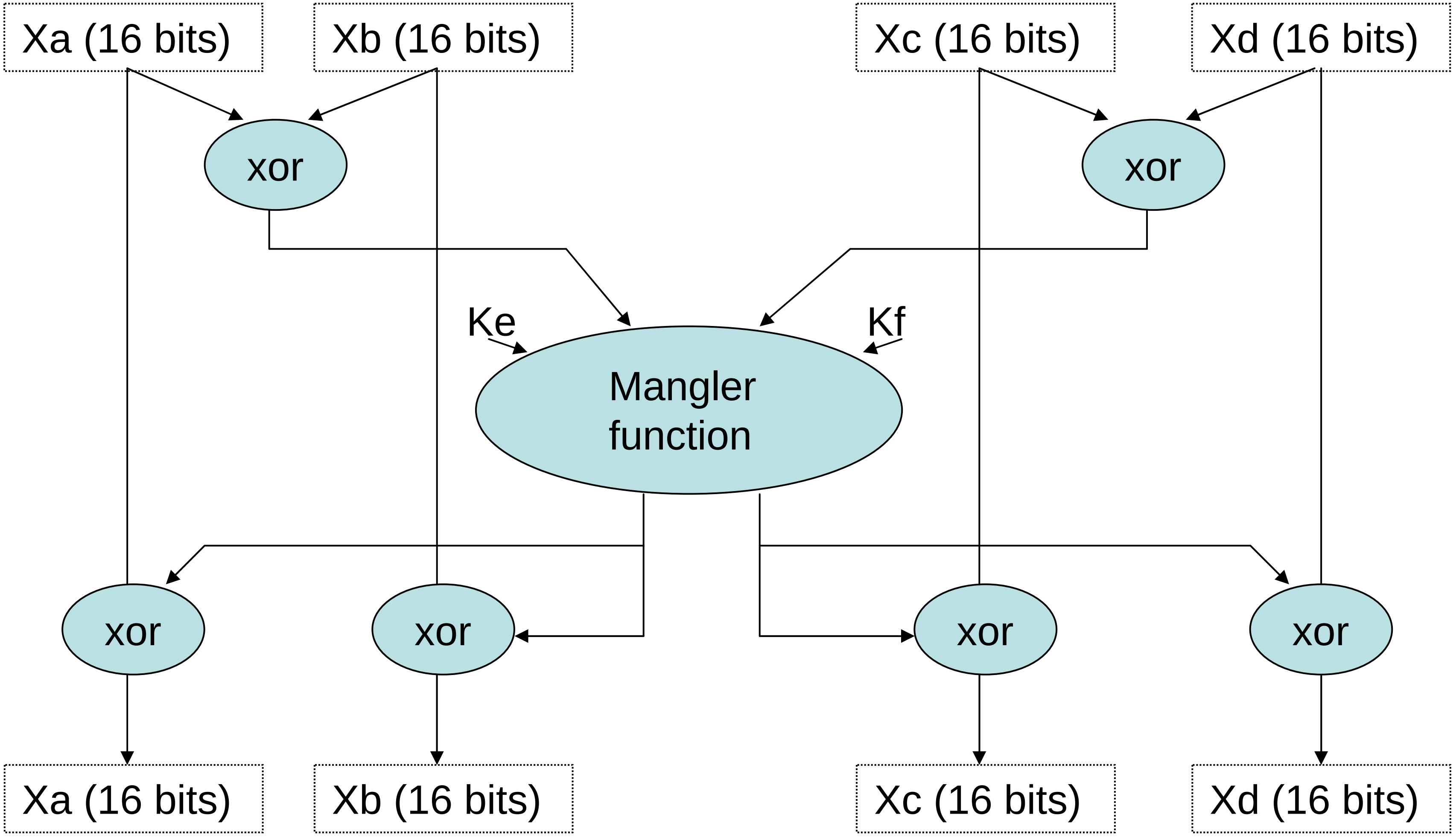
Multiplication modulo $2^{16}+1$, where input 0 is replaced by 2^{16}, and result 2^{16} is encoded as 0



Addition modulo 2^{16}

One round of IDEA: even round

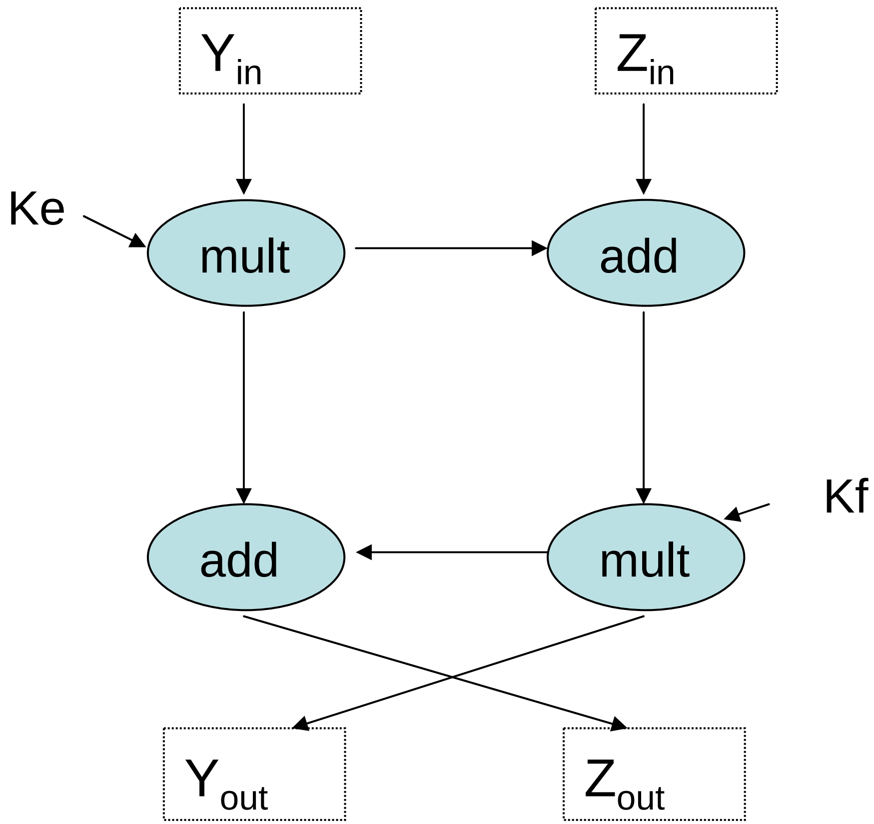
This function is its own inverse!



The mangler function

Y_{out} = (Ke mult Y_{in}) add Z_{in}) mult Kf

Z_{out} = (Ke mult Y_{in}) add Y_{out}



The Security of IDEA

- IDEA has been around almost 15 years
- Designed by Xuejia Lai and Jim Massey
- Its only problem so far is its small block size
- Numerous analysis has been published, but nothing substantial
- It is not available in public domain, except for research purposes
- It is available under licence
- It is widely used, e.g in PGP (see Lecture 11)

AES

AES

- Candidates due June 15, 1998: 21 submissions, 15 met the criteria
- 5 finalists August 1999: MARS, RC6, Rijndael, Serpent, and Twofish, (along with regrets for E2)
- October 3, 2000, NIST announces the winner: Rijndael
- FIPS 197, November 26, 2001

Federal Information Processing Standards Publication 197, ADVANCED ENCRYPTION STANDARD (AES)

Rijndael - Internal Structure

Rijndael is an iterated block cipher with variable length block and variable key size. The number of rounds is defined by the table:

	Nb = 4	Nb = 6	Nb = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

AES

Nb = length of data block in 32-bit words

Nk = length of key in 32-bit words

Rijndael - Internal Structure

- First Initial Round Key Addition
- 9 rounds, numbered 1-9, each consisting of
  - Byte Substitution transformation
  - Shift Row transformation
  - Mix Column transformation
  - Round Key Addition
- A final round (round 10) consisting of
  - Byte Substitution transformation
  - Shift Row transformation
  - Final Round Key Addition

Rijndael - Inverse Structure

ENCRYPT	DECRYPT	→ INV ENCRYPT
Initial Round Key Add	Final Round Key Add →	Inv Initial Round Key Add
Byte Substitution	Inv Shift Row	Inv Byte Substitution
Shift Row	Inv Byte Substitution	Inv Shift Row
Mix Column	Round Key Addition	Inv Mix Column
Round Key Addition	Inv Mix Column	Inv Round Key Addition

… eight more rounds like this

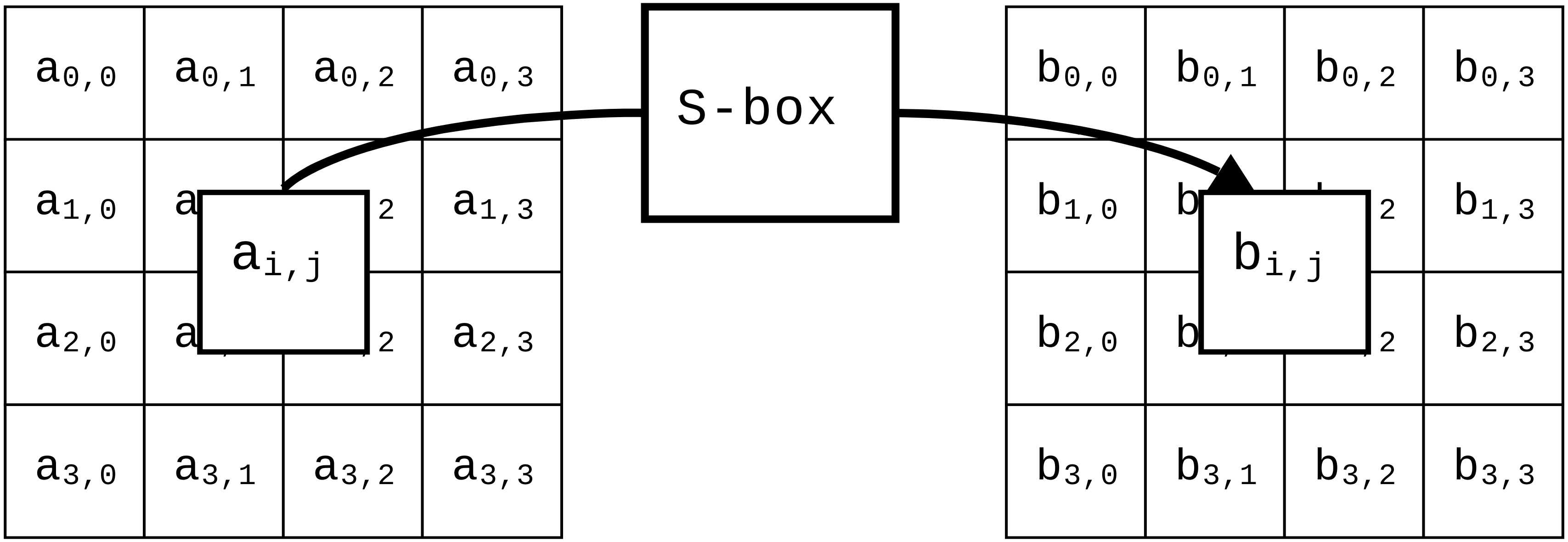
Byte Substitution	Inv Shift Row	Inv Byte Substitution
Shift Row	Inv Byte Substitution	Inv Shift Row
Final Round Key Add	Initial Round Key Add →	Inv Final Round Key Add

Rijndael-128 State and 128 Cipher Key

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

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

Byte Substitution



Rijndael S-box

Sbox[256] = {

```
 99,124,119,123,242,107,111,197, 48,  1,103, 43,254,215,171,118,
202,130,201,125,250, 89, 71,240,173,212,162,175,156,164,114,192,
183,253,147, 38, 54, 63,247,204, 52,165,229,241,113,216, 49, 21,
  4,199, 35,195, 24,150,  5,154,  7, 18,128,226,235, 39,178,117,
  9,131, 44, 26, 27,110, 90,160, 82, 59,214,179, 41,227, 47,132,
 83,209,  0,237, 32,252,177, 91,106,203,190, 57, 74, 76, 88,207,
208,239,170,251, 67, 77, 51,133, 69,249,  2,127, 80, 60,159,168,
 81,163, 64,143,146,157, 56,245,188,182,218, 33, 16,255,243,210,
 96,129, 79,220, 34, 42,144,136, 70,238,184, 20,222, 94, 11,219,
224, 50, 58, 10, 73,  6, 36, 92,194,211,172, 98,145,149,228,121,
231,200, 55,109,141,213, 78,169,108, 86,244,234,101,122,174,  8,
186,120, 37, 46, 28,166,180,198,232,221,116, 31, 75,189,139,138,
112, 62,181,102, 72,  3,246, 14, 97, 53, 87,185,134,193, 29,158,
225,248,152, 17,105,217,142,148,155, 30,135,233,206, 85, 40,223,
140,161,137, 13,191,230, 66,104, 65,153, 45, 15,176, 84,187, 22};
```

Rijndael S-box Design View

Galois field $GF(2^8)$ with polynomial

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

The Rijndael S-box is the composition $f \circ g$ where

$$g(x) = x^{-1},\ x \in GF(2^8),\ x \neq 0, \text{ and}$$

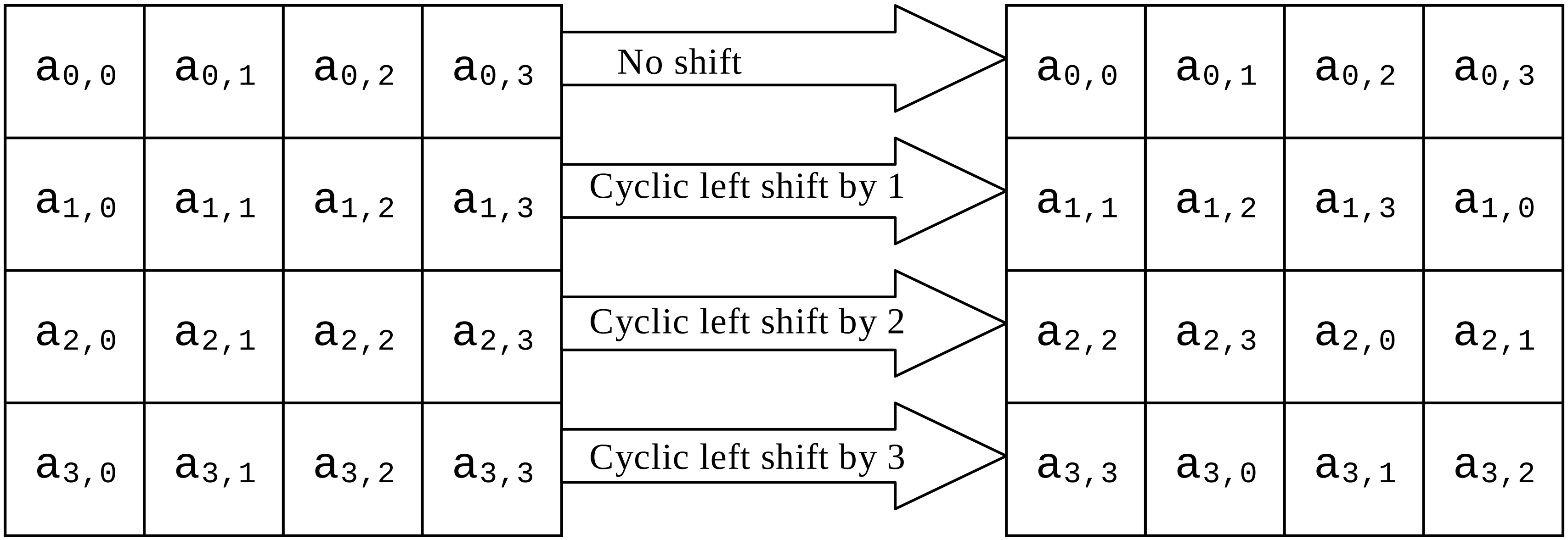
$$g(0) = 0$$

and f is the affine transformation defined by y = f(x)

Inv $(f \circ g) =$
$g \circ (\text{Inv } f)$

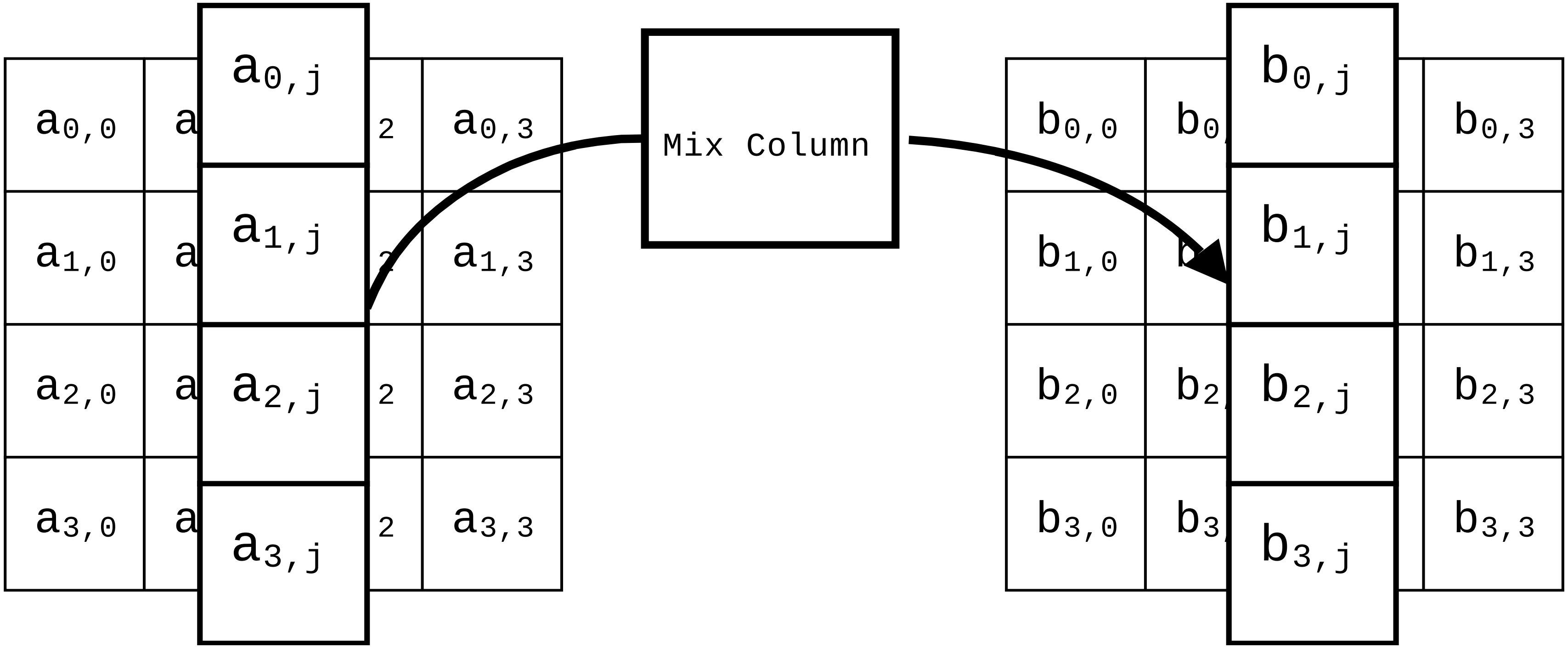
$$\begin{bmatrix} y_o \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\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

Shift Row



a0,0 a0,1 a0,2 a0,3
a1,0 a1,1 a1,2 a1,3
a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3
No shift
Cyclic left shift by 1
Cyclic left shift by 2
Cyclic left shift by 3
a0,0 a0,1 a0,2 a0,3
a1,1 a1,2 a1,3 a1,0
a2,2 a2,3 a2,0 a2,1
a3,3 a3,0 a3,1 a3,2

Mix Column



$a_{0,0}$
$a_{1,0}$
$a_{2,0}$
$a_{3,0}$
$a_{0,j}$
$a_{1,j}$
$a_{2,j}$
$a_{3,j}$
$a_{0,3}$
$a_{1,3}$
$a_{2,3}$
$a_{3,3}$
Mix Column
$b_{0,0}$
$b_{1,0}$
$b_{2,0}$
$b_{3,0}$
$b_{0,j}$
$b_{1,j}$
$b_{2,j}$
$b_{3,j}$
$b_{0,3}$
$b_{1,3}$
$b_{2,3}$
$b_{3,3}$

Mix Column - Implemented

The mix column transformation mixes one column of the state at a time.

Column j:

$$b_{0,j} = T_2(a_{0,j}) \oplus T_3(a_{1,j}) \oplus a_{2,j} \oplus a_{3,j}$$
$$b_{1,j} = a_{0,j} \oplus T_2(a_{1,j}) \oplus T_3(a_{2,j}) \oplus a_{3,j}$$
$$b_{2,j} = a_{0,j} \oplus a_{1,j} \oplus T_2(a_{2,j}) \oplus T_3(a_{3,j})$$
$$b_{3,j} = T_3(a_{0,j}) \oplus a_{1,j} \oplus a_{2,j} \oplus T_2(a_{3,j})$$

where:

$$T_2(a) = 2*a \quad \text{if } a < 128$$
$$T_2(a) = (2*a) \oplus 283 \quad \text{if } a \geq 128$$
$$T_3(a) = T_2(a) \oplus a.$$

Mix Column - Design view

The columns of the State are considered as polynomials over GF(2^8).

They are multiplied by a fixed polynomial c(x) given by

$$c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$$

The product is reduced modulo $x^4 + 01$.

Matrix form

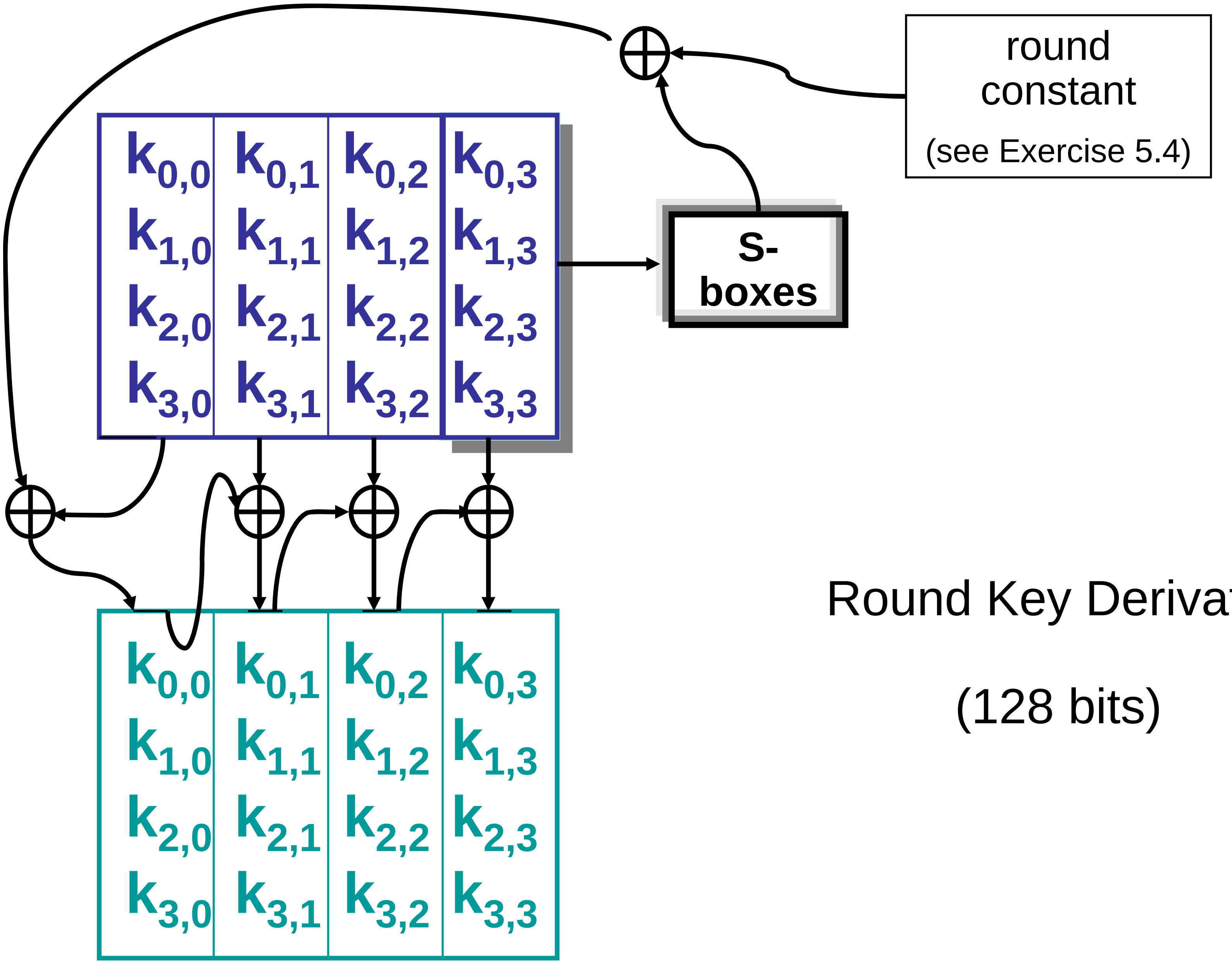
$$\begin{bmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{bmatrix}$$

The Inverse Mix Column polynomial is $c(x)^{-1} \bmod (x^4 + 01) = d(x)$ given by

$$d(x) = 0B \cdot x^3 + 0D \cdot x^2 + 09 \cdot x + 0E$$

Round Key Addition

$$\begin{array}{|c|c|c|c|}\hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline rk_{0,0} & rk_{0,1} & rk_{0,2} & rk_{0,3} \\ \hline rk_{1,0} & rk_{1,1} & rk_{1,2} & rk_{1,3} \\ \hline rk_{2,0} & rk_{2,1} & rk_{2,2} & rk_{2,3} \\ \hline rk_{3,0} & rk_{3,1} & rk_{3,2} & rk_{3,3} \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \\ \hline \end{array}$$



Round Key Derivation

(128 bits)

The Security of AES

- Designed to be resistant against differential and linear cryptanalysis
  - S-boxes optimal
  - Wide Trail Strategy
- Has quite an amazing algebraic structure (see the next slide)
- Algebraic cryptanalysis tried but not yet (!) successful
- Algebraic cryptanalysis: given known plaintext – ciphertext pairs construct algebraic systems of equations, and try to solve them.

Algebraic equations from AES encryption

state $x^{(r)} = (x_{ij}^{(r)}), \quad i, j = 0,1,2,3, \quad r = 1,2,\ldots,10, \quad x_{ij}^{(r)} \in GF(2^8)$

key $k^{(r)} = (k_{ij}^{(r)}), \quad i, j = 0,1,2,3, \quad r = 0,1,2,\ldots,10, \quad k_{ij}^{(r)} \in GF(2^8)$

p plaintext block, c ciphertext block

AES encryption:

$$x^{(1)} = p \oplus k^{(0)}$$

$$x^{(r+1)} = M(S(F(G(x^{(r)}))) \oplus k^{(r)}, \; r = 1,2,\ldots,9$$

$$c = S(F(G(x^{(10)}))) \oplus k^{(10)}$$

where

M, S are linear functions over $GF(2^8)$

$G = (g)$ where $g : GF(2^8) \to GF(2^8), \; g(x) = x^{-1}, \; g(0) = 0$

$F = (f)$ where $f - \lambda_0$ is additive over $GF(2^8)$

Differential and linear cryptanalysis

Differential cryptanalysis (Biham-Shamir 1990)

- Chosen plaintext attack
- A large number of pairs of plaintext blocks are generated. Each pair of plaintext has a fixed difference. Corresponding ciphertexts are computed (using the encryption device with a fixed key as black box).
- Main idea: The statistics of the differences of the data blocks before the last round can be predicted.
- Exhaustive search of the last round key are performed by testing if decryptions with the candidate key of the ciphertext pairs gives results that match with the predicted statistics.

Differential and linear cryptanalysis

Linear cryptanalysis (Matsui 1993)

- Known plaintext attack
- A large number of plaintext blocks and their corresponding ciphertexts are known.
- Main idea: The statistics of a fixed linear combination of the data bits before the last round can be predicted by some fixed linear combination of the plaintext bits.
- Exhaustive search of the last round key are performed by testing if decryptions with the candidate key of the ciphertext blocks gives results that match with the predicted statistics.