Cryptography CS-215

Instructor   Sergey Abrahamyan

AUA-2023

SAFER+ algorithm history

- SAFER+ algorithm was invented by James Massey, Gurgen Khachatrian and Melsik Kuregian in 1998
- SAFER+ was sponsored by CYLINK corporation and was one of the candidate algorithms for Advanced Encryption Standard (AES)
- SAFER+ was adopted for use in the challenge/response authentication scheme in the Bluetooth protocol in 2000

SAFER+ application in Bluetooth

- Bluetooth server-client challenge response authentication protocol:

  - Server generates a challenge (a random number *r),* sends to client (mobile phone, etc.) and asks client to encrypt it with the key *k* that server and client share.

  - A Client encrypts *r* with the *k* by using SAFER+ encryption algorithm to get $E_k(r)$ and sends it back to Server

  - A Server calculates $D_k(E_k(r))$ and if the result is *r* then server authentication is confirmed.

Design principles for SAFER+

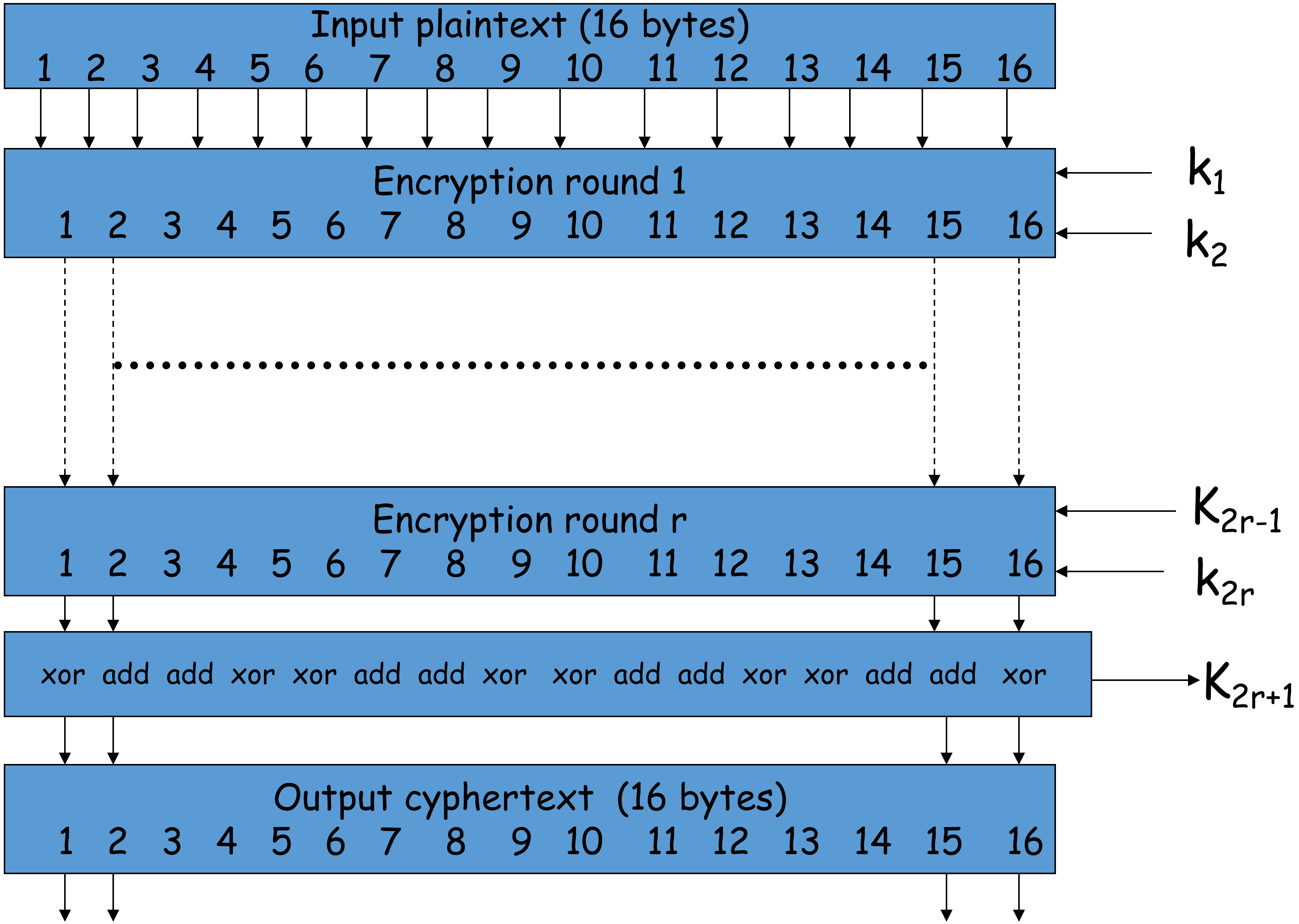
- *Byte orientation* – all operations within encryption and decryption are on bytes
- *Substitution (non-linear)/Linear-Transformation* encrypting structure
- *Use of two additive group operations on bytes* -- takes advantage of their strengths

Design principles for SAFER+

- *Confusion* (Shannon's 1st design principle) *via well-defined nonlinear functions* – no "suspicious-looking" tables
- *Fast-diffusing linear transformation* – good *diffusion* (Shannon's 2nd design principle) achieved with the special convertible binary matrices
- *Number of rounds* – conservatively chosen for security with a margin of safety

- Plaintext 16 bytes
- Ciphertext 16 bytes
- Key length 16 , 24, 32 bytes
- If key length= 16/24/32 then r=8/12/16 rounds

SAFER+ encrypting structure



Input plaintext (16 bytes)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Encryption round 1

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

k_1

k_2

Encryption round r

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

K_{2r-1}

k_{2r}

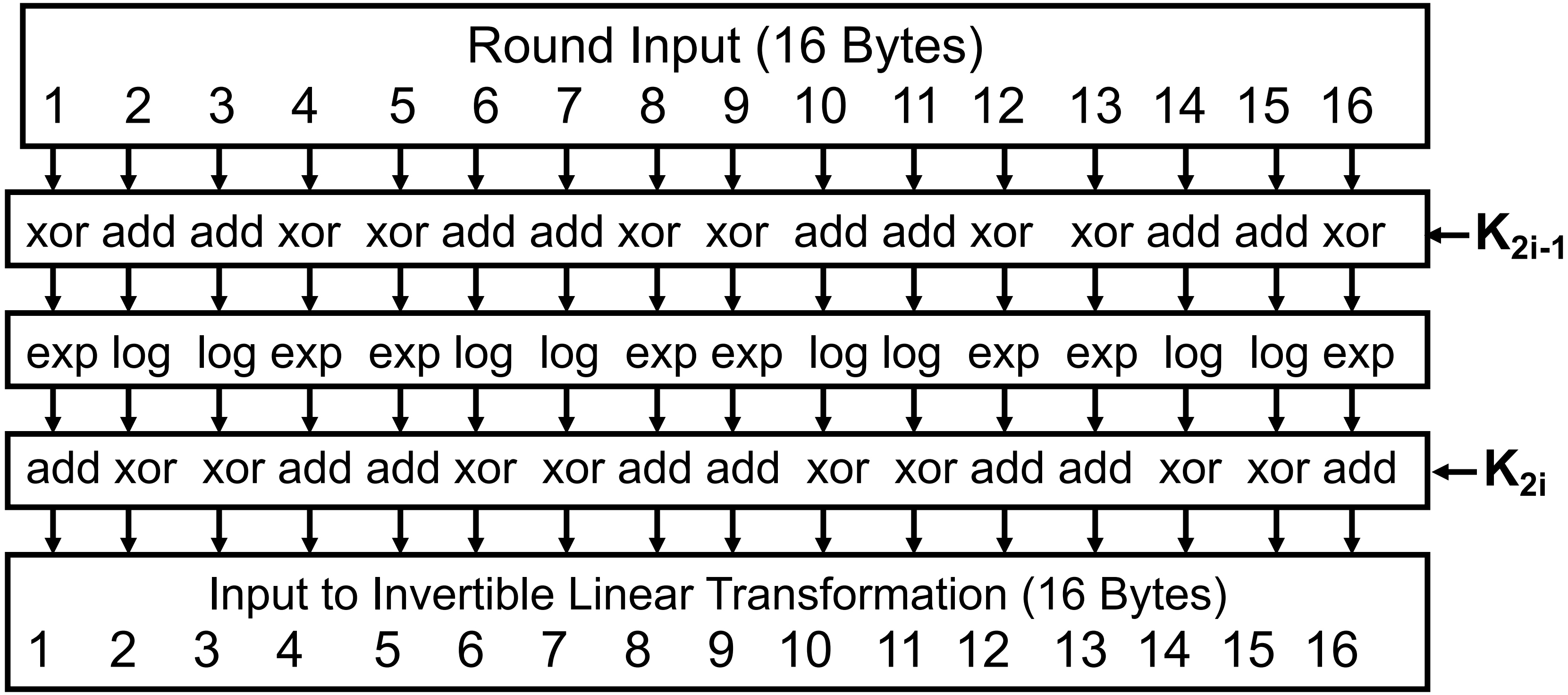
xor add add xor xor add add xor xor add add xor xor add add xor

K_{2r+1}

Output cyphertext (16 bytes)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Non-linear part of SAFER+ round



Round Input (16 Bytes)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

xor add add xor xor add add xor xor add add xor xor add add xor ← K_{2i-1}

exp log log exp exp log log exp exp log log exp exp log log exp

add xor xor add add xor xor add add xor xor add add xor xor add ← K_{2i}

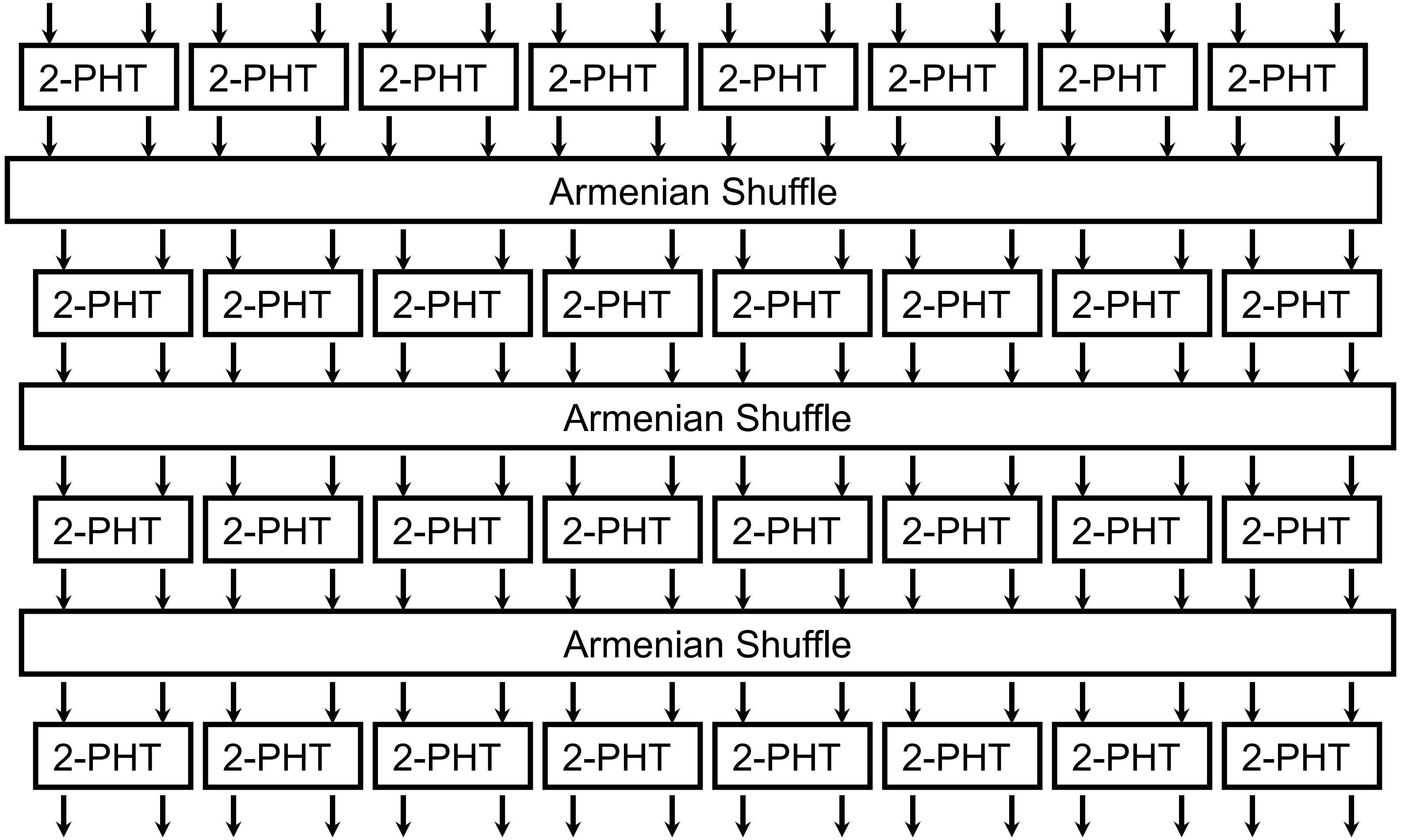
Input to Invertible Linear Transformation (16 Bytes)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Building blocks of SAFER+

- Safer+ has two nonlinear byte to byte transformation tables. One table denoted by *EXP* is based on exponentiation function $45^X \equiv Y \bmod 257$ where *X* and *Y* are any numbers between *0* and *255.* The second one denoted by *LOG* is based on logarithm function $\log_{45}(X) \equiv Y \bmod 257$
- There are two operations between bytes *X* and *Y* , one is *XOR* operation i.e. $X \oplus Y$ and another one is *mod 256* ***sum*** between *X* and *Y* denoted by *add*

The SAFER+ linear transformation:


2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT
Armenian Shuffle
2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT
Armenian Shuffle
2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT
Armenian Shuffle
2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT 2-PHT

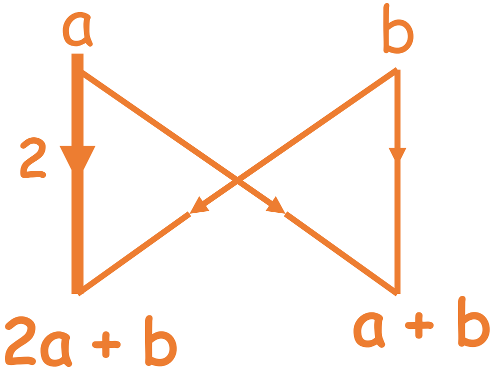
where the *"Armenian Shuffle"* is the coordinate permutation:

9 12 13 16 3 2 7 6 11 10 15 14 1 8 5 4

Building blocks of SAFER+

The matrix, *butterfly*, and inverse matrix of the *2-PHT (pseudo- Hadamard matrix)*

$H_2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$



$H_2^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$

(a,b) is a 2-byte input, and *(2a+b,a+b)* is a 2-byte output of *2-PHT*

All operations are modulo 256

Armenian Shuffle

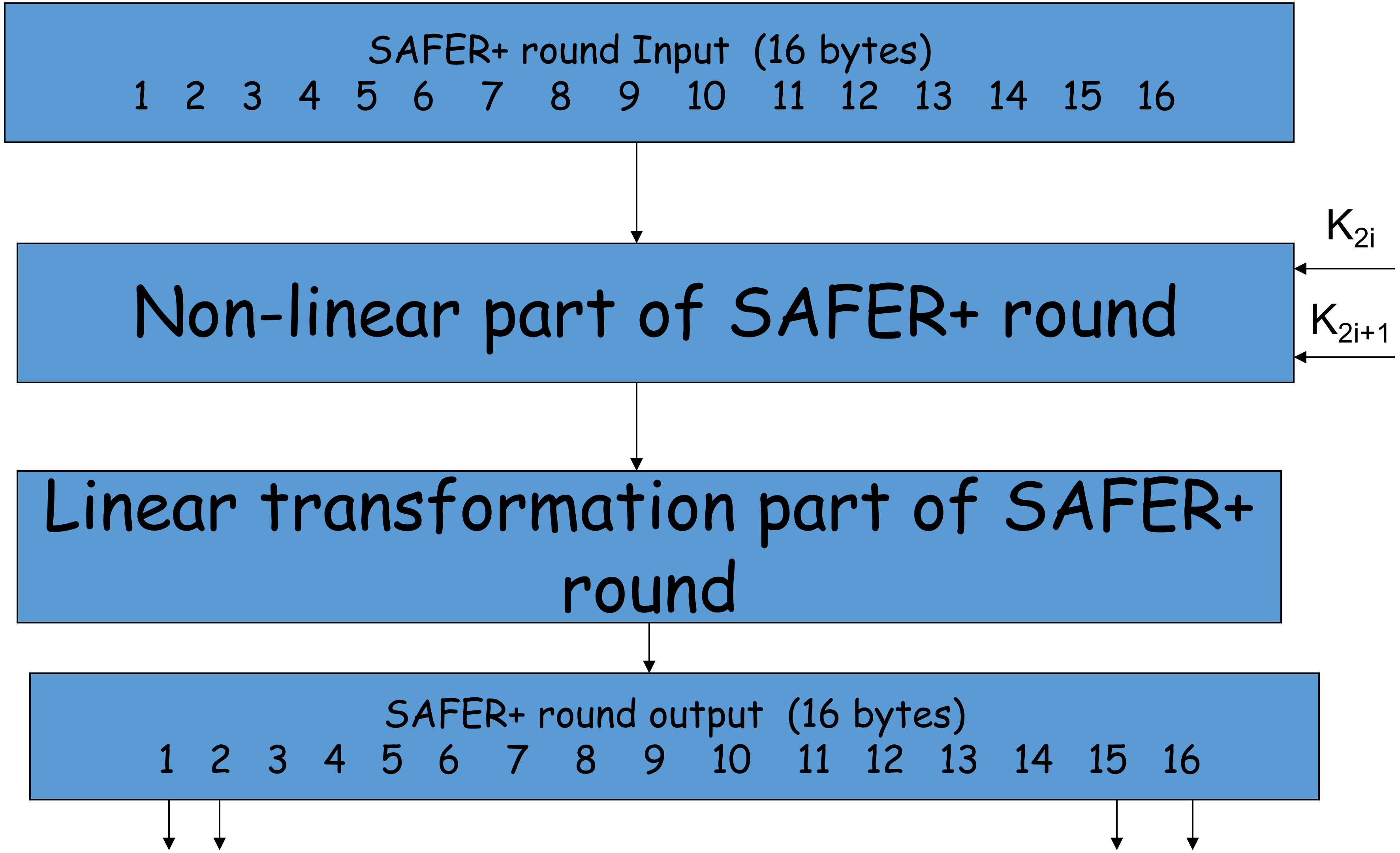
- Is a simple transformation
- $\begin{pmatrix} 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8, & 9, & 10, & 11, & 12, & 13, & 14, & 15, & 16 \\ 9, & 12, & 13, & 16, & 3, & 2, & 7, & 6, & 11, & 10, & 15, & 14, & 1, & 8, & 5, & 4 \end{pmatrix}$

SAFER+ Linear Transformation matrix:

$$M = \begin{bmatrix}
2 & 2 & 1 & 1 & 16 & 8 & 2 & 1 & 4 & 2 & 4 & 2 & 1 & 1 & 4 & 4 \\
1 & 1 & 1 & 1 & 8 & 4 & 2 & 1 & 2 & 1 & 4 & 2 & 1 & 1 & 2 & 2 \\
1 & 1 & 4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 2 & 2 & 1 & 1 \\
1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 1 & 1 & 1 & 1 \\
4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 1 & 1 & 1 & 1 & 2 & 2 \\
2 & 2 & 2 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 2 & 1 & 2 & 2 & 4 & 4 & 1 & 1 \\
1 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 2 & 1 & 1 & 1 & 2 & 2 & 1 & 1 \\
2 & 1 & 16 & 8 & 1 & 1 & 2 & 2 & 1 & 1 & 4 & 4 & 4 & 2 & 4 & 2 \\
2 & 1 & 8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 1 \\
4 & 2 & 4 & 2 & 4 & 4 & 1 & 1 & 2 & 2 & 1 & 1 & 16 & 8 & 2 & 1 \\
2 & 1 & 4 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 8 & 4 & 2 & 1 \\
4 & 2 & 2 & 2 & 1 & 1 & 4 & 4 & 1 & 1 & 4 & 2 & 2 & 1 & 16 & 8 \\
4 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 1 & 2 & 1 & 8 & 4 \\
16 & 8 & 1 & 1 & 2 & 2 & 1 & 1 & 4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 \\
8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 4 & 2
\end{bmatrix}$$

Note that *changing a single input byte is guaranteed to change at least 5 output bytes*

SAFER+ round structure



SAFER+ round Input (16 bytes)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Non-linear part of SAFER+ round

K_{2i}

K_{2i+1}

Linear transformation part of SAFER+ round

SAFER+ round output (16 bytes)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

SAFER+ Linear Transformation matrix:

- It was recently shown that from differential and linear cryptanalysis point of view “Armenian Shuffle” is the best choice in the sense that all other possible shuffles of coordinates will not require less number of rounds for the SAFER+ to be safe.

SAFER+ Linear Transformation example

- Suppose we are changing 2 most significant bits of the first 2 bytes at the input of SAFER+ LTM. Then the resulting output change for LTM will be:

$(2\cdot 128 + 128,\ 2\cdot 128 + 128,\ 128 + 128,\ 128 + 128,$

$16 \cdot 128 + 8 \cdot 128,\ 8 \cdot 128 + 4 \cdot 128,\ 2 \cdot 128 + 2 \cdot 128,$

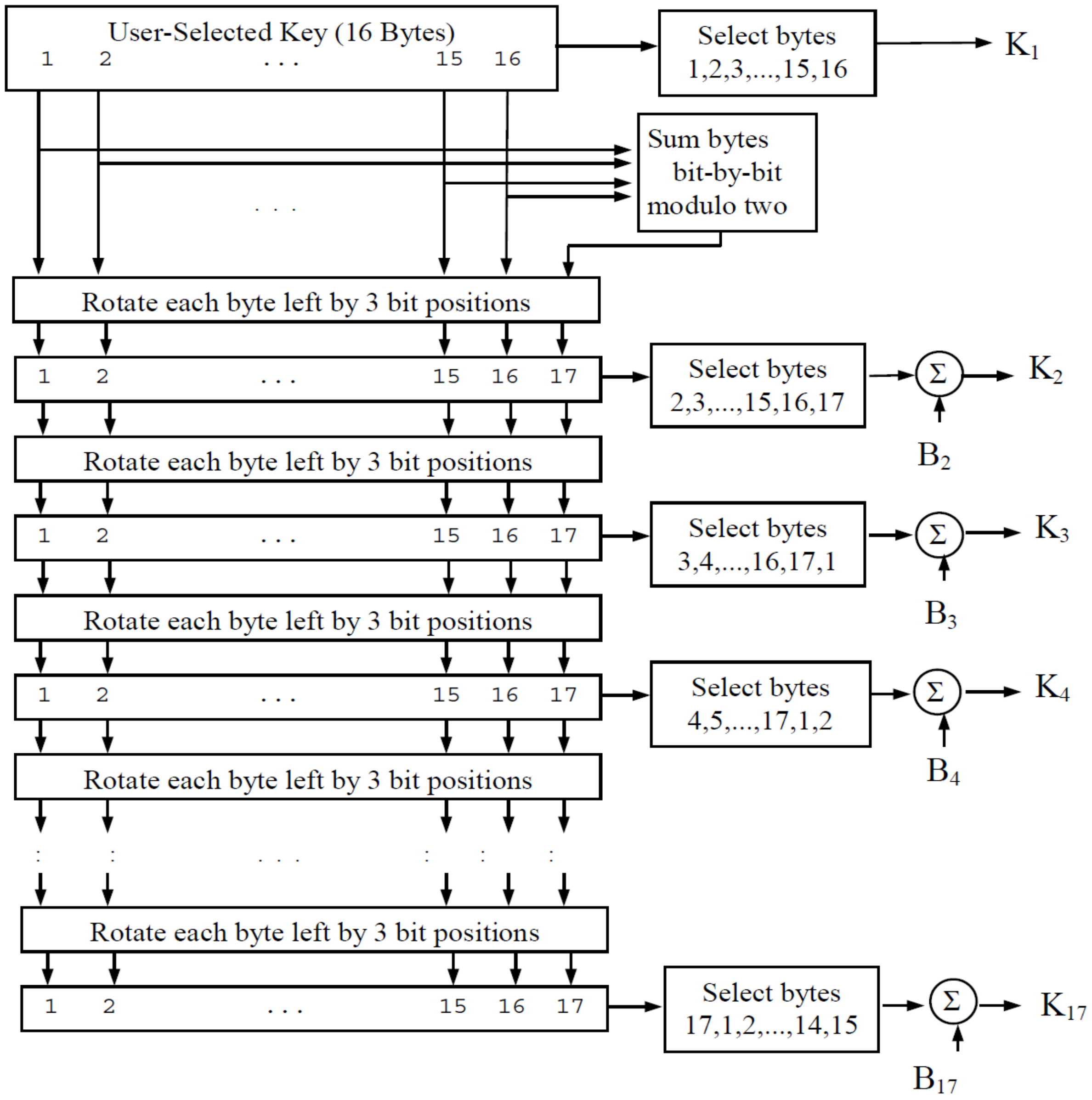
$128 + 128,\ 4 \cdot 128 + 2 \cdot 128,\ 2\cdot 128 + 128,\ 4\cdot 128 + 4\cdot 128,$

$2\cdot 128 + 2\cdot 128,\ 128 + 128,\ 128 + 128,\ 4 \cdot 128 + 2 \cdot 128,$

$4 \cdot 128 + 2 \cdot 128) = (128, 128, 0, 0, 0, 0, 0, 0, 0, 128, 0, 0, 0, 0, 0, 0)$

Note that this is smallest possible change for LTM. For the next round change if we change 1,2 and 10 –th positions by 128 corresponding to the previous change then we will get change at the output for the LTM in 6 positions.

SAFER+ Key Schedule for 128 bit key


User-Selected Key (16 Bytes)
1 2 . . . 15 16
Select bytes 1,2,3,...,15,16
K_1
Sum bytes bit-by-bit modulo two
. . .
Rotate each byte left by 3 bit positions
1 2 . . . 15 16 17
Select bytes 2,3,...,15,16,17
Σ
K_2
B_2
Rotate each byte left by 3 bit positions
1 2 . . . 15 16 17
Select bytes 3,4,...,16,17,1
Σ
K_3
B_3
Rotate each byte left by 3 bit positions
1 2 . . . 15 16 17
Select bytes 4,5,...,17,1,2
Σ
K_4
B_4
Rotate each byte left by 3 bit positions
: : . . . : : :
Rotate each byte left by 3 bit positions
1 2 . . . 15 16 17
Select bytes 17,1,2,...,14,15
Σ
K_{17}
B_{17}

SAFER + Bias vectors

70,151,177,186,163,183,16,10,197,55,179,201,90,40,172,100

236,171,170,198,103,149,88,13,248,154,246,110,102,220,5,61

138,195,216,137,106,233,54,73,67,191,235,212,150,155,104,160

93,87,146,31,213,113,92,187,34,193,190,123,188,153,99,148

42,97,184,52,50,25,253,251,23,64,230,81,29,65,68,143,

221,4,128,222,231,49,214,127,1,162,247,57,218,111,35,202

58,208,28,209,48,62,18,161,205,15,224,168,175,130,89,44

125,173,178,239,194,135,206,117,6,19,2,144,79,46,114,51

192,141,207,169,129,226,196,39,47,108,122,159,82,225,21,56

252,32,66,199,8,228,9,85,94,140,20,118,96,255,223,215

250,11,33,0,26,249,166,185,232,158,98,76,217,145,80,210,

24,180,7,132,234,91,164,200,14,203,72,105,75,78,156,53,

69,77,84,229,37,60,12,74,139,63,204,167,219,107,174,244

45,243,124,109,157,181,38,116,242,147,83,176,240,17,237,131

182,3,22,115,59,30,142,112,189,134,27,71,126,36,86,241

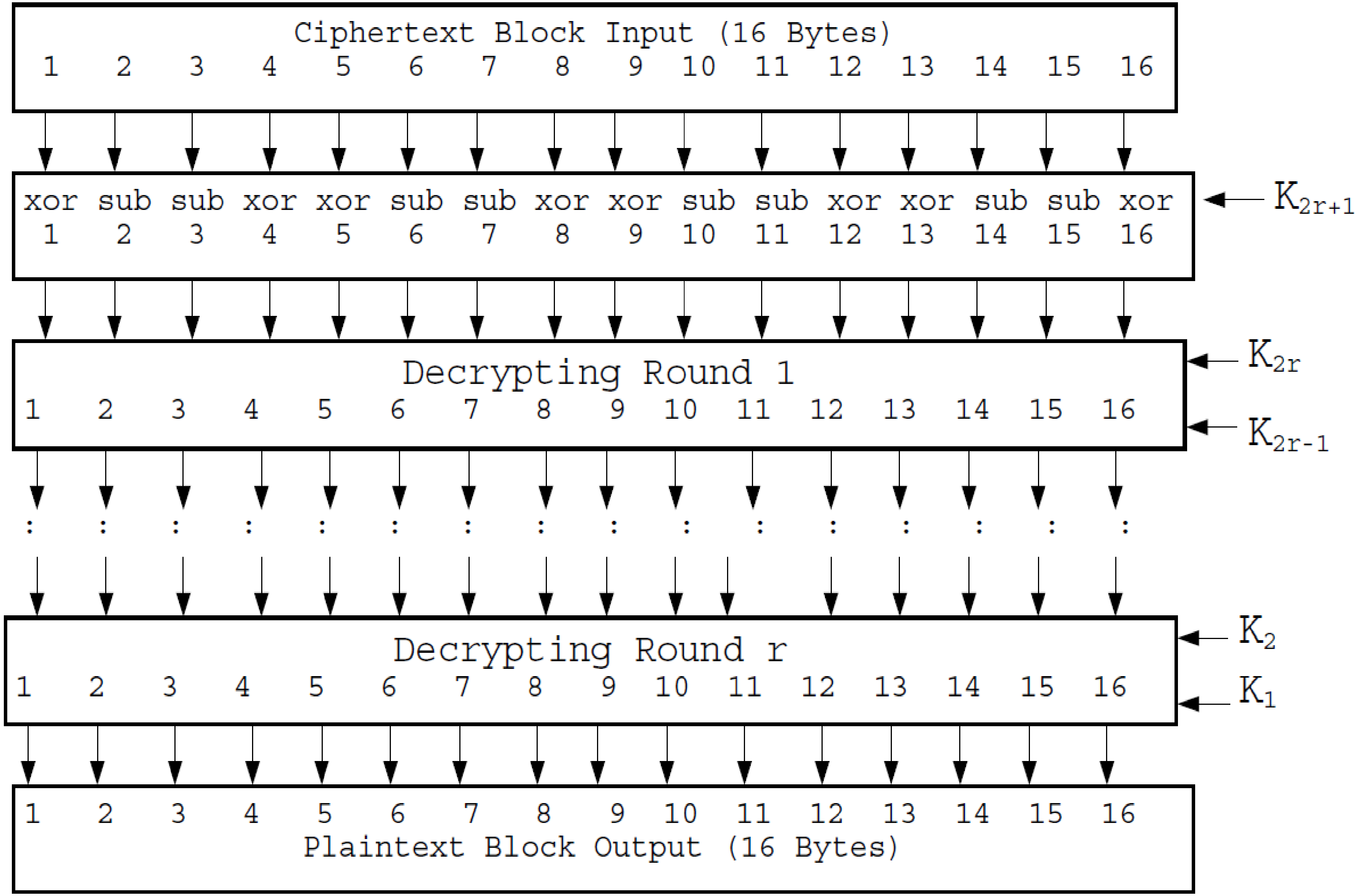
136,70,151,177,186,163,183,16,10,197,55,179,201,90,40,172

Calculation of Bias vectors

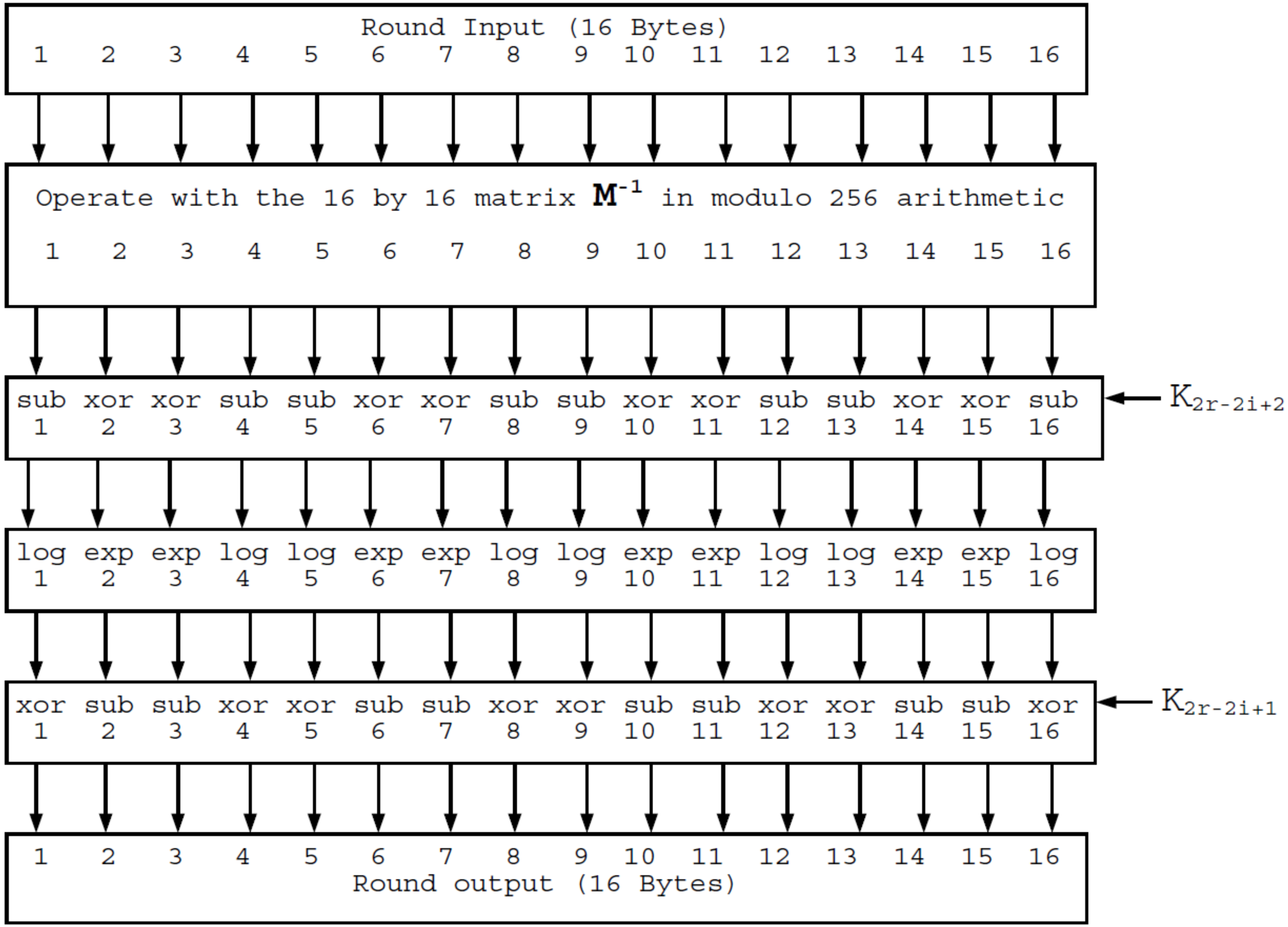
$$B_{i,j} = 45^{(45^{17i+j} \bmod 257)} \bmod 257$$

$B_{i,j}$ denote j-th byte of i-th bias word and *i=2,3….17, j=1,2….16* and $B_{i,j}$ is represented as *0* in the case if expression gives a value of *256*

Decrypting structure of SAFER+ algorthm



Structure of decrypting round



Round Input (16 Bytes)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Operate with the 16 by 16 matrix $\mathbf{M}^{-1}$ in modulo 256 arithmetic
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

sub 1, xor 2, xor 3, sub 4, sub 5, xor 6, xor 7, sub 8, sub 9, xor 10, xor 11, sub 12, sub 13, xor 14, xor 15, sub 16 ← $K_{2r-2i+2}$

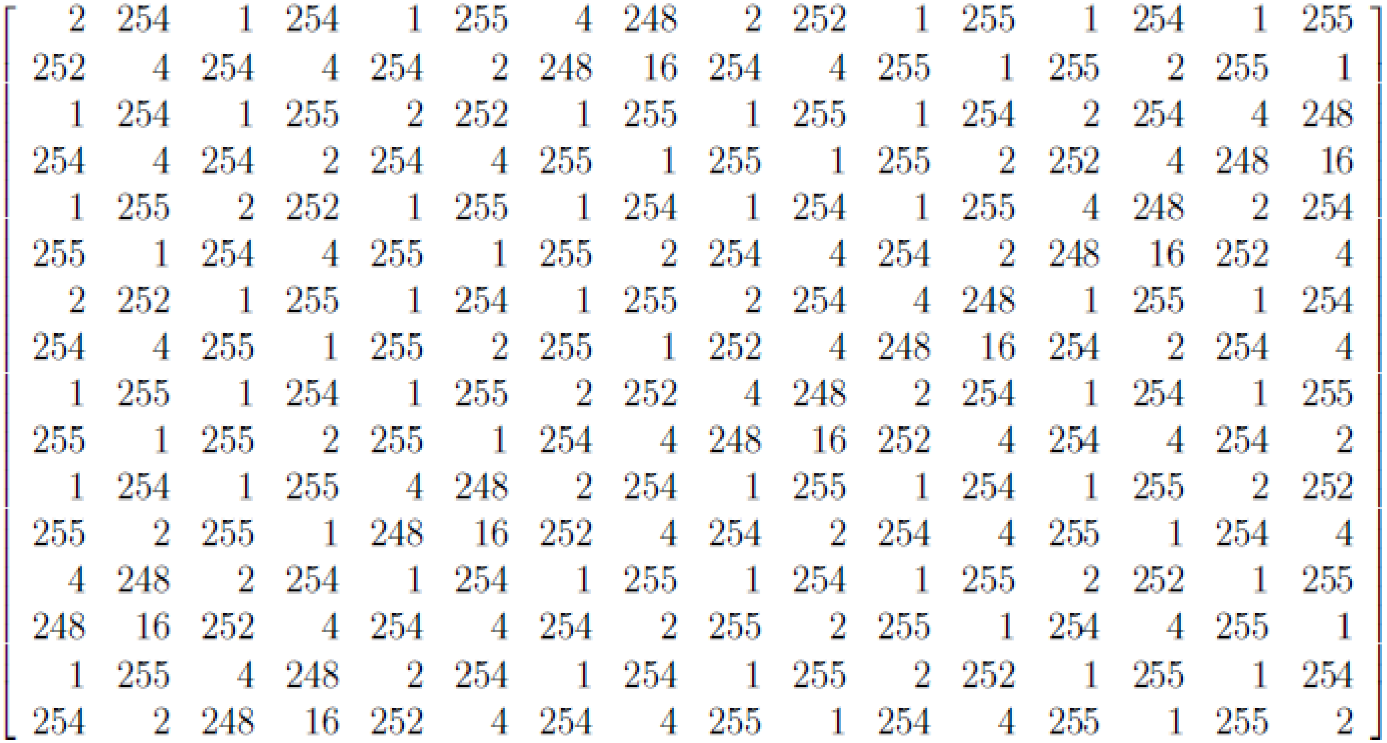
log 1, exp 2, exp 3, log 4, log 5, exp 6, exp 7, log 8, log 9, exp 10, exp 11, log 12, log 13, exp 14, exp 15, log 16

xor 1, sub 2, sub 3, xor 4, xor 5, sub 6, sub 7, xor 8, xor 9, sub 10, sub 11, xor 12, xor 13, sub 14, sub 15, xor 16 ← $K_{2r-2i+1}$

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Round output (16 Bytes)

Inverse matrix

- The matrix M^{-1} is the 16x16 matrix



$$\begin{bmatrix}
2 & 254 & 1 & 254 & 1 & 255 & 4 & 248 & 2 & 252 & 1 & 255 & 1 & 254 & 1 & 255 \\
252 & 4 & 254 & 4 & 254 & 2 & 248 & 16 & 254 & 4 & 255 & 1 & 255 & 2 & 255 & 1 \\
1 & 254 & 1 & 255 & 2 & 252 & 1 & 255 & 1 & 255 & 1 & 254 & 2 & 254 & 4 & 248 \\
254 & 4 & 254 & 2 & 254 & 4 & 255 & 1 & 255 & 1 & 255 & 2 & 252 & 4 & 248 & 16 \\
1 & 255 & 2 & 252 & 1 & 255 & 1 & 254 & 1 & 254 & 1 & 255 & 4 & 248 & 2 & 254 \\
255 & 1 & 254 & 4 & 255 & 1 & 255 & 2 & 254 & 4 & 254 & 2 & 248 & 16 & 252 & 4 \\
2 & 252 & 1 & 255 & 1 & 254 & 1 & 255 & 2 & 254 & 4 & 248 & 1 & 255 & 1 & 254 \\
254 & 4 & 255 & 1 & 255 & 2 & 255 & 1 & 252 & 4 & 248 & 16 & 254 & 2 & 254 & 4 \\
1 & 255 & 1 & 254 & 1 & 255 & 2 & 252 & 4 & 248 & 2 & 254 & 1 & 254 & 1 & 255 \\
255 & 1 & 255 & 2 & 255 & 1 & 254 & 4 & 248 & 16 & 252 & 4 & 254 & 4 & 254 & 2 \\
1 & 254 & 1 & 255 & 4 & 248 & 2 & 254 & 1 & 255 & 1 & 254 & 1 & 255 & 2 & 252 \\
255 & 2 & 255 & 1 & 248 & 16 & 252 & 4 & 254 & 2 & 254 & 4 & 255 & 1 & 254 & 4 \\
4 & 248 & 2 & 254 & 1 & 254 & 1 & 255 & 1 & 254 & 1 & 255 & 2 & 252 & 1 & 255 \\
248 & 16 & 252 & 4 & 254 & 4 & 254 & 2 & 255 & 2 & 255 & 1 & 254 & 4 & 255 & 1 \\
1 & 255 & 4 & 248 & 2 & 254 & 1 & 254 & 1 & 255 & 2 & 252 & 1 & 255 & 1 & 254 \\
254 & 2 & 248 & 16 & 252 & 4 & 254 & 4 & 255 & 1 & 254 & 4 & 255 & 1 & 255 & 2
\end{bmatrix}$$

SAFER+ test vectors

K1: 31,29,87,89,178,77,8,242,85,166,186,135,151,117,52,120

P1: 244,15,160,13,16,171,110,151,5,38,161,80,18,75,184,222

C1: 118,163,158,247,108,13,249,229,197,41,30,73,82,160,42,233

K2: 120,229,116,47,13,17,62,220,181,115,51,58,175,193,108,41

P2: 72,79,53,87,226,166,35,114,93,19,147,38,161,153,19,43

C2: 185,48,11,73,95,217,13,73,230,143,169,44,57,145,42,149

K3: 39,222,35,230,230,81,42,122,75,164,235,29,168,208,130,167

P3: 243,133,143,133,57,120,135,96,152,95,81,20,193,112,158,144

C3: 12,245,132,210,26,204,113,135,154,172,201,123,152,176,149,215

Using Block Ciphers as Stream Ciphers

- can use block cipher to generate numbers
- use Counter Mode

  $X_i = E_{Km}[i]$
- use Output Feedback Mode

  $X_i = E_{Km}[X_{i-1}]$
- ANSI X9.17 PRNG
  - uses date-time + seed inputs and 3 triple-DES encryptions to generate new seed & random

ANSI X9.17 PRNG

- R_i - Pseudo-random number produced by the i-th generation stage.
- V_i - Seed value at the beginning of i-th generation stage.
- K_1, K_2 - Tripple DES keys used for each stage
- DT_i - Date/time value at the beginning of i-th generation stage.
- EDE- Triple DES Encrypt-decrypt-Encrypt Then

$$R_i = EDE([K_1, K_2], [V_i \oplus EDE([K_1, K_2], DT_i)])$$

$$V_{i+1} = EDE([K_1, K_2], [R_i \oplus EDE([K_1, K_2], DT_i)])$$

- Thank you