Some attacks against block ciphers

Christina Boura

École de printemps en codage et cryptographie
May 19, 2016

Outline

Statistical attacks

Statistical attacks exploit relations that hold with a certain probability only.

- Rely on the existence of a distinguisher.

A distinguisher $\mathcal{D}$ for a block cipher $(E_k)_k$ is an algorithm taking N pairs (x_i, y_i), $1 \leq i \leq N$ and returning 0 or 1.

Goal: Decide if the N pairs are input-output pairs of the target block cipher or not:

- 1: If the (x_i, y_i) are input-output pairs of E_k for some key k.
- 0: If the (x_i, y_i) are input-output pairs of a random permutation.

Advantage of the distinguisher

- Let p be the probability that the algorithm returns 1 (the N pairs come from the target block cipher).
- Let p' be the probability that the algorithm returns 0 (the N pairs come from a random permutation).

The capacity to distinguish the target block cipher from a random permutation is measured as

$$|p - p'|$$

and is called advantage.

Consequences of a distinguisher

- The existence of a distinguisher with a non-negligeable advantage is an undesirable property for a block cipher.
- However, this **does not** always guarantee that once the distinguisher is discovered, the secret key will be recovered.

But: For iterated ciphers

$$E_k = F_{k_r} \circ F_{k_{r-1}} \circ \cdots \circ F_{k_1}$$

a distinguisher for the reduced cipher

$$G_k = F_{k_{r-1}} \circ \cdots \circ F_{k_1}$$

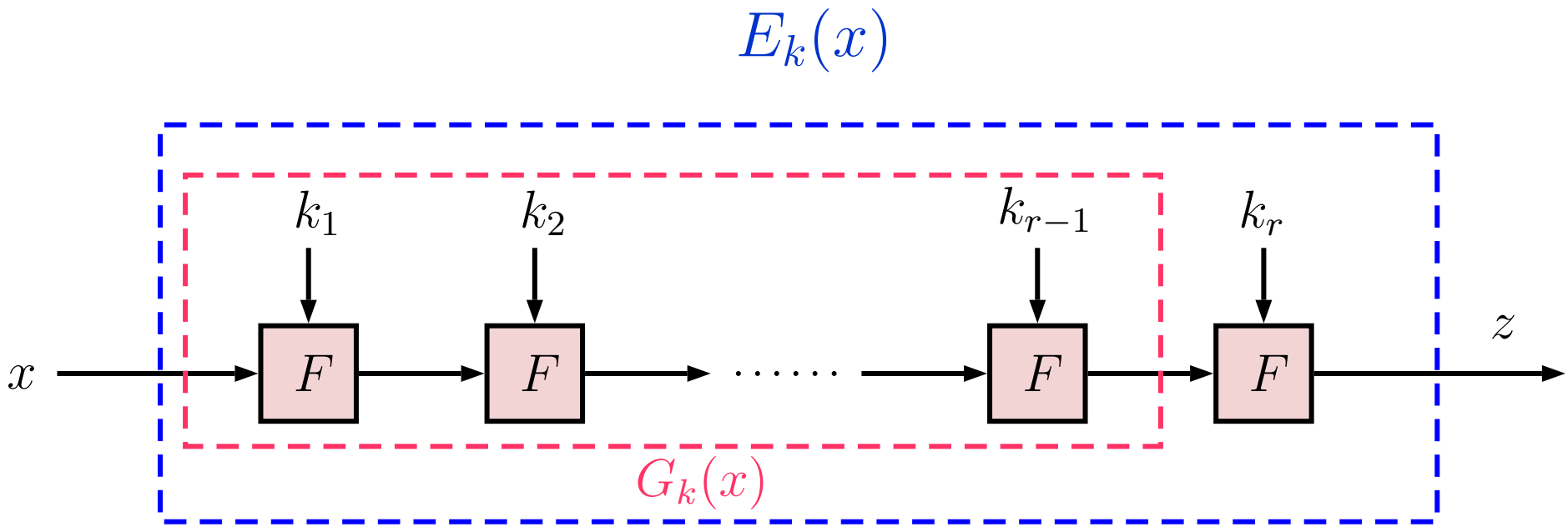
can be a serious thread.

Attack on the last round (I)

If an attacker finds a distinguisher $\mathcal{D}$ for the reduced-round cipher G_k, then he can run a last-round attack.

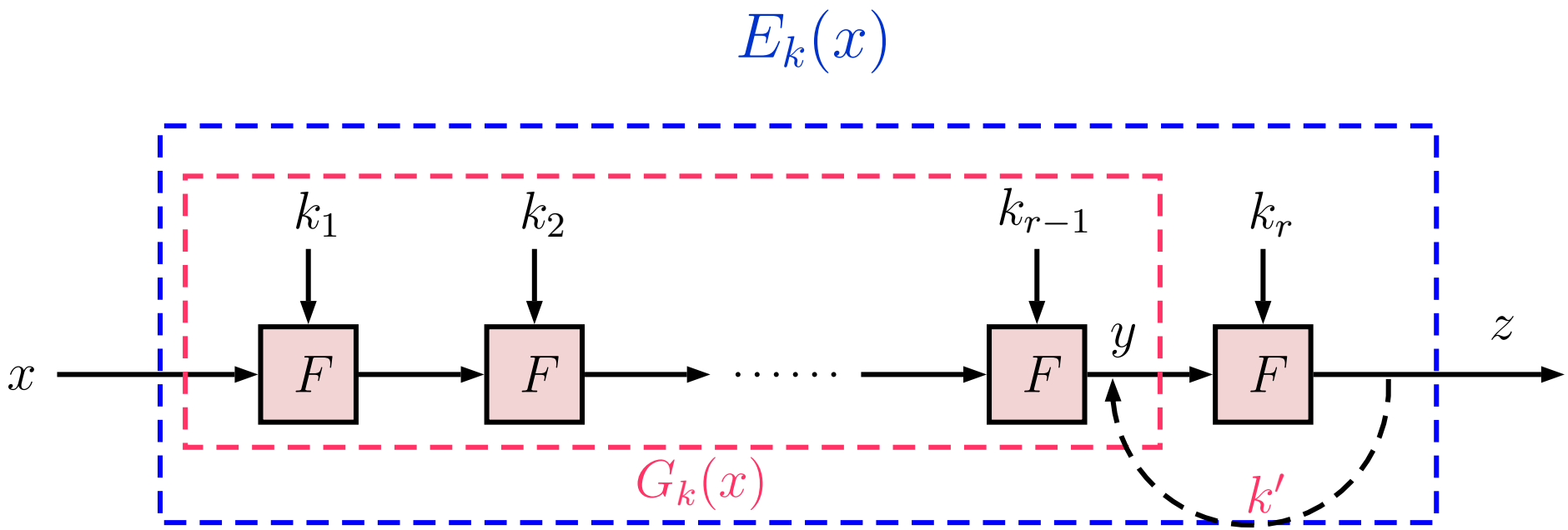
Goal: Recover the last-round subkey k_r.

Attack on the last round (II)



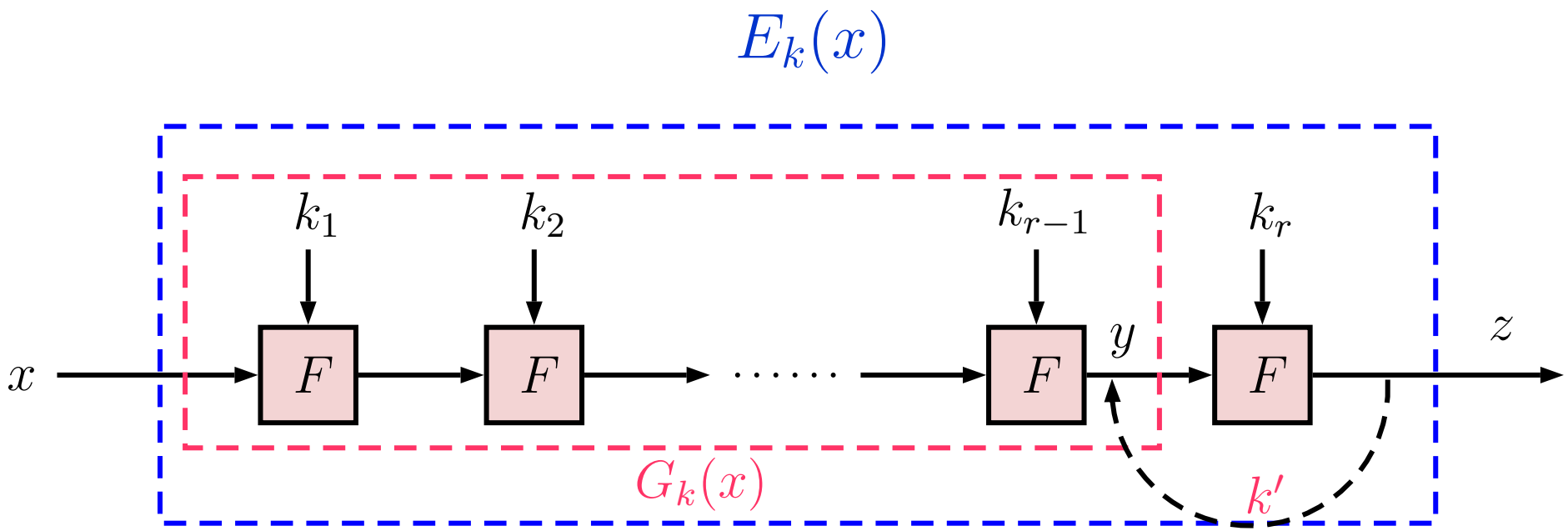
- Collect enough plaintext-ciphertext pairs (x_i, z_i), where $z_i = E_k(x_i)$.

Attack on the last round (II)

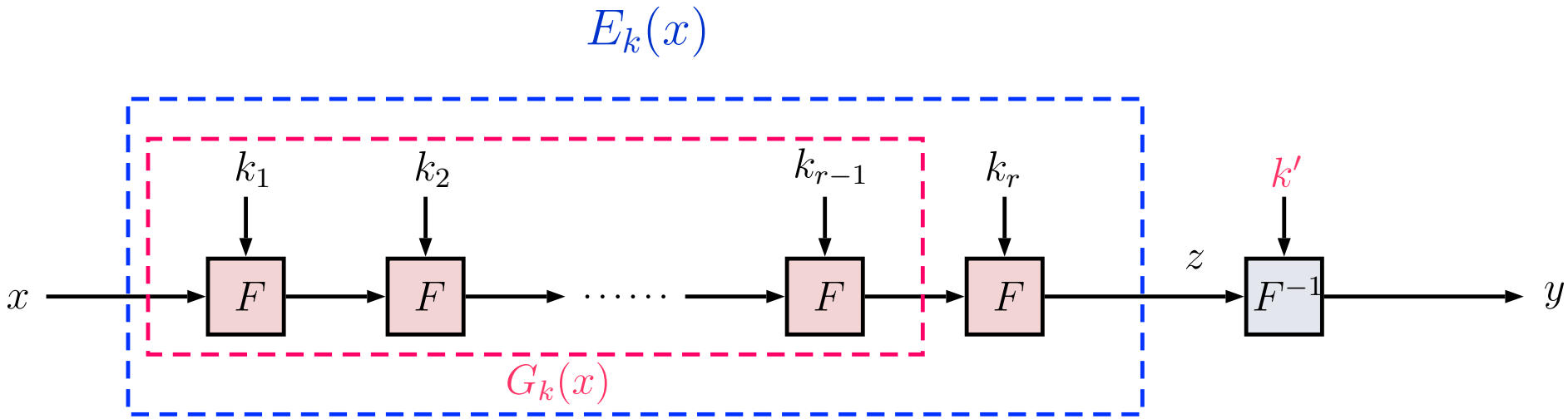


- Collect enough plaintext-ciphertext pairs (x_i, z_i), where $z_i = E_k(x_i)$.
- For all possible values k' compute $y_i = F_{k'}^{-1}(z_i)$

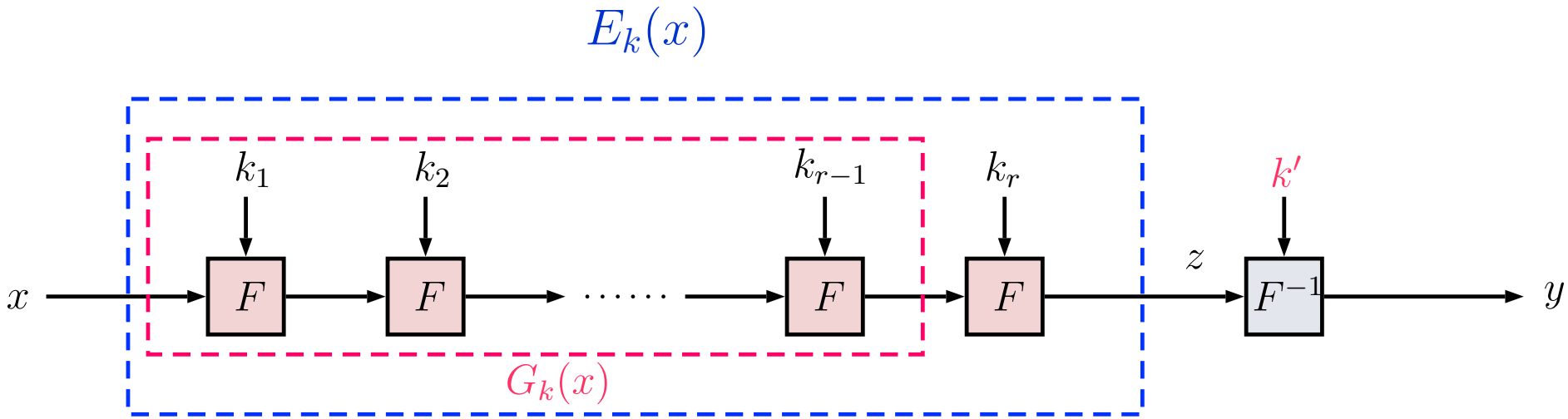
Attack on the last round (III)



Attack on the last round (III)

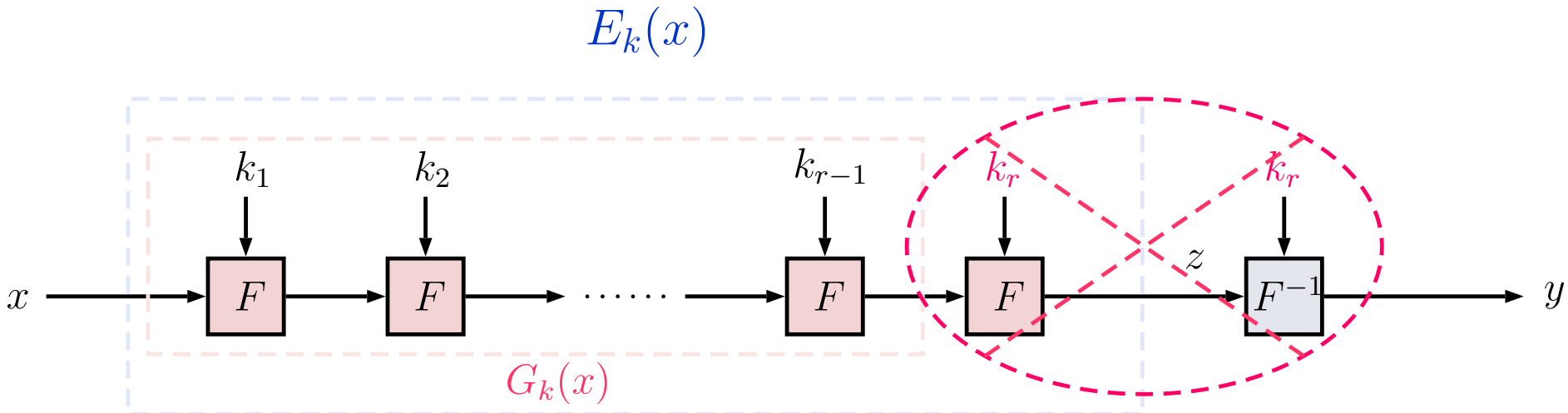


Attack on the last round (III)



- If k' is the right subkey ($k' = k_r$)

Attack on the last round (III)

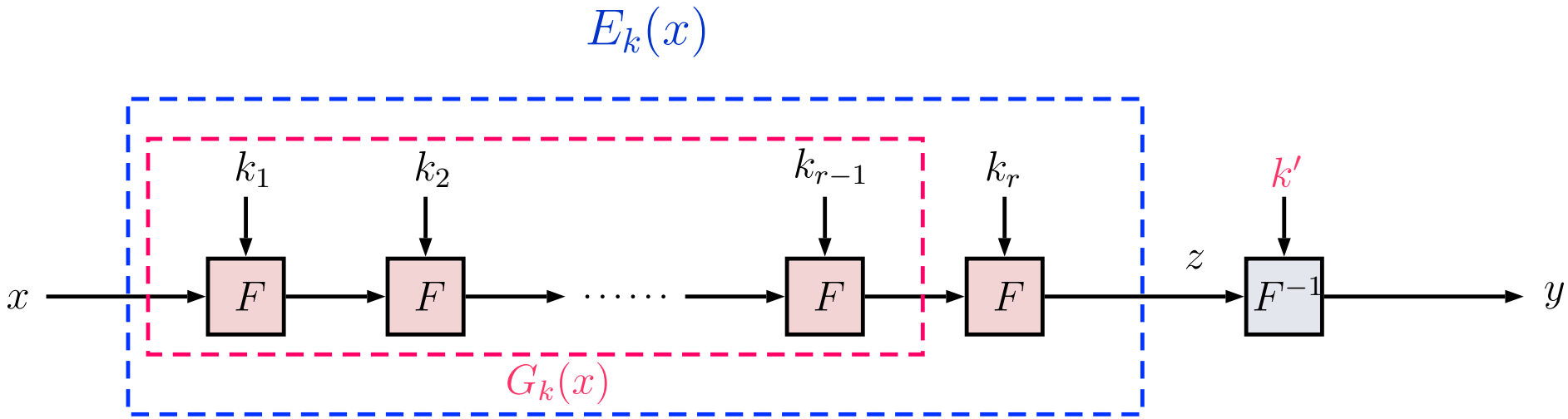


- If k' is the right subkey ($k' = k_r$) :

$$\begin{aligned} P(k') = F_{k'}^{-1} \circ E_k &= F_{k'}^{-1} \circ F_{k_r} \circ F_{k_{r-1}} \circ F_{k_{r-2}} \circ \cdots \circ F_{k_1} \\ &= F_{k_r}^{-1} \circ F_{k_r} \circ F_{k_{r-1}} \circ F_{k_{r-2}} \circ \cdots \circ F_{k_1} \\ &= F_{k_{r-1}} \circ F_{k_{r-2}} \circ \cdots \circ F_{k_1} \\ &= G_k \end{aligned}$$

$P(k')$ belongs to the family of reduced-ciphers.

Attack on the last round (III)



- If k' is a wrong subkey, $P(k')$ is assumed to have the same behaviour as a randomly chosen permutation.
- This assumption is known as the wrong-key randomization hypothesis.

Algorithm

Data: N plaintext-ciphertext couples (x_i, z_i), for $1 \leq i \leq N$
Result: A set of candidate keys for the last-round subkey k_r
for *all possible values* k' *of* k_r **do**
    counter $\leftarrow 0$;
    for $i = 0 \dots N$ **do**
        compute $y_i = F_{k'}^{-1}(z_i)$;
        counter $\leftarrow$ counter $+ \mathcal{D}(x_i, y_i)$;
    end
    if *counter* $\geq \tau$ **then**
        return k' ;
    end
end

- The value τ is a threshold value fixed by the attacker.

Remarks

- As we exhaust all values of the last round subkey, this attack only works in this basic form if the subkeys have a small size (eg. not for AES-128)
- In practice, we only try to recover a small part of the last round key (some bits).
- For the other bits of the subkey, we repeat the attack by modifying the parameters of the attack.

Once the last subkey recovered, how do we proceed next ?

- For some ciphers, once a subkey completely recovered, one can compute back through the key schedule to retrieve the master key.
- If the different subkeys are not related, one can
  - Exhaustively search the remaining key bits
  - Repeat the same attack on the ciphers obtained by successively removing the last round
  - Combine both approaches

Outline

Higher-order derivatives

Let $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$.

Derivative of F at a point $a \in \mathbf{F}_2^n$:

$$D_a F(x) := F(x \oplus a) \oplus F(x), \text{ for every } x \in \mathbf{F}_2^n$$

Xuejia Lai extended this notion in 1994.

Definition[k-th order derivative of F]
For any k-dimensional subspace V of $\mathbf{F}_2^n$, the k-th order derivative of F with respect to V is the function defined by

$$D_V F(x) = D_{a_1} D_{a_2} \dots D_{a_k} F(x) = \bigoplus_{v \in V} F(x + v),$$

for every $x \in \mathbf{F}_2^n$, where $(a_1, \dots, a_k)$ is a basis of V.

Example

Let $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$ and $V = \langle a_1, a_2 \rangle \subset \mathbf{F}_2^n$ of dimension 2.

The 2^{nd}-order derivative of F with respect to V is

$$\begin{aligned} D_V F(x) &= D_{a_1} D_{a_2} F(x) \\ &= D_{a_1}(F(x) + F(x + a_2)) \\ &= F(x) + F(x + a_1) + F(x + a_2) + F(x + a_1 + a_2). \end{aligned}$$

Degree of a derivative

Let $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$ of degree d and $a = (a_1, \ldots, a_n)$. Then,

$$D_a F \leq d - 1.$$

Examples:

- $F(x_1, \ldots, x_n) = x_1$. Then,

$$D_a F(x) = D_a(x_1) = (x_1 \oplus a_1) \oplus x_1 = a_1 \Rightarrow \deg(D_a F) = 0$$

- $F(x_1, \ldots, x_n) = x_1 x_2$. Then,

$$\begin{aligned} D_a F(x) = D_a(x_1 x_2) &= (x_1 \oplus a_1)(x_2 \oplus a_2) \oplus x_1 x_2 \\ &= x_1 x_2 \oplus a_1 x_2 \oplus a_2 x_1 \oplus a_1 a_2 \oplus x_1 x_2 \\ &= a_1 x_2 \oplus a_2 x_1 \oplus a_1 a_2 \Rightarrow \deg(D_a F) = 1 \end{aligned}$$

Important property

Let $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$ of degree d and $a = (a_1, \ldots, a_n)$.

Example:

- $F(x_1, \ldots, x_n) = x_1 x_2 \cdots x_d$. Then,

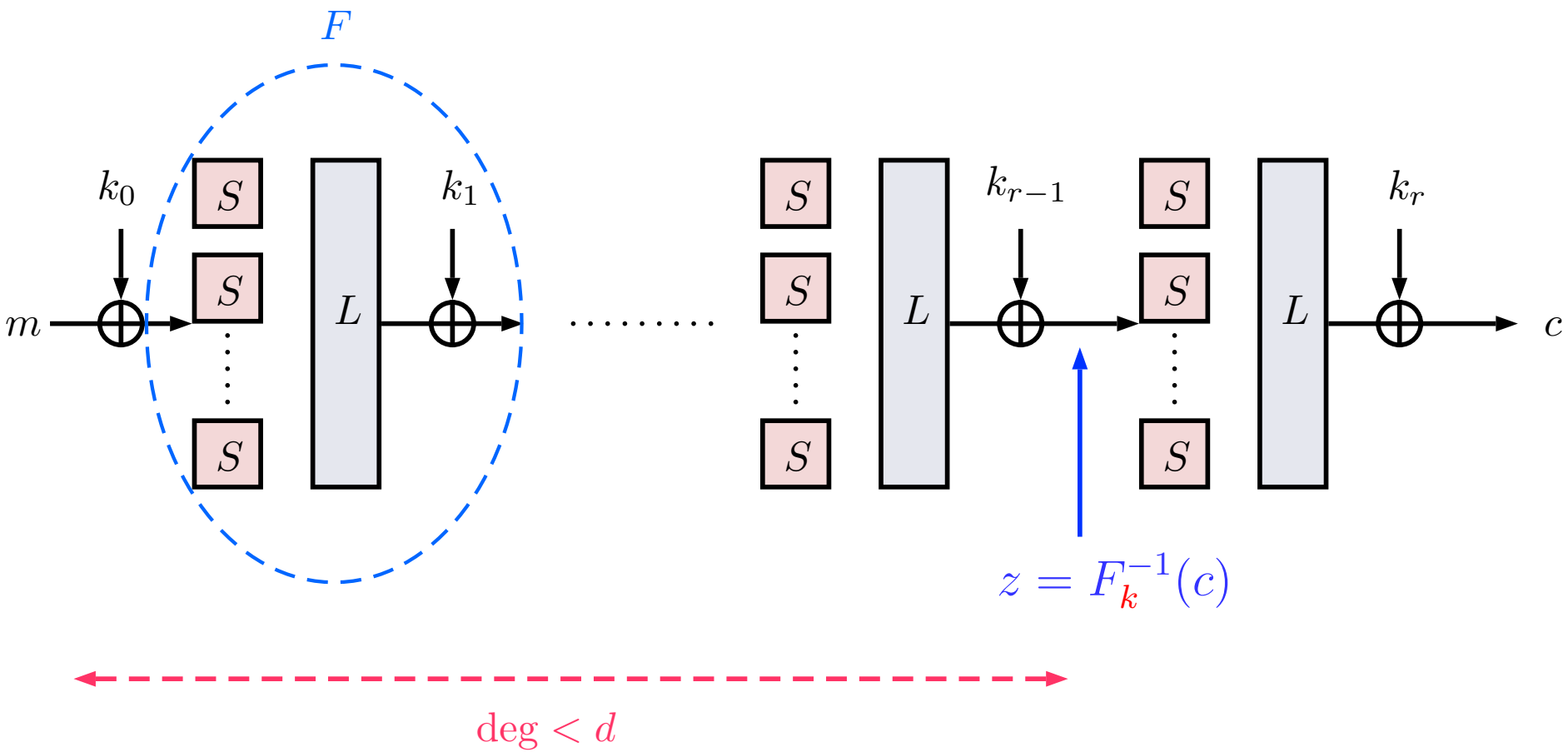
$$\begin{aligned} D_a(x_1 x_2 \cdots x_d) &= (x_1 \oplus a_1)(x_2 \oplus a_2) \ldots (x_d \oplus a_d) \oplus x_1 x_2 \cdots x_d \\ &= x_1 \cdots x_d \oplus \text{ terms of deg } \leq d-1 \oplus x_1 \cdots x_d \\ &\Rightarrow \deg(D_a F) \leq d-1 \end{aligned}$$

Proposition[Lai 94]
For every subspace V with $\dim V > \deg F$,

$$D_V F(x) = \bigoplus_{v \in V} F(x+v) = 0, \text{ for every } x \in \mathbf{F}_2^n.$$

Attack on the last round

Attack based on a low degree.

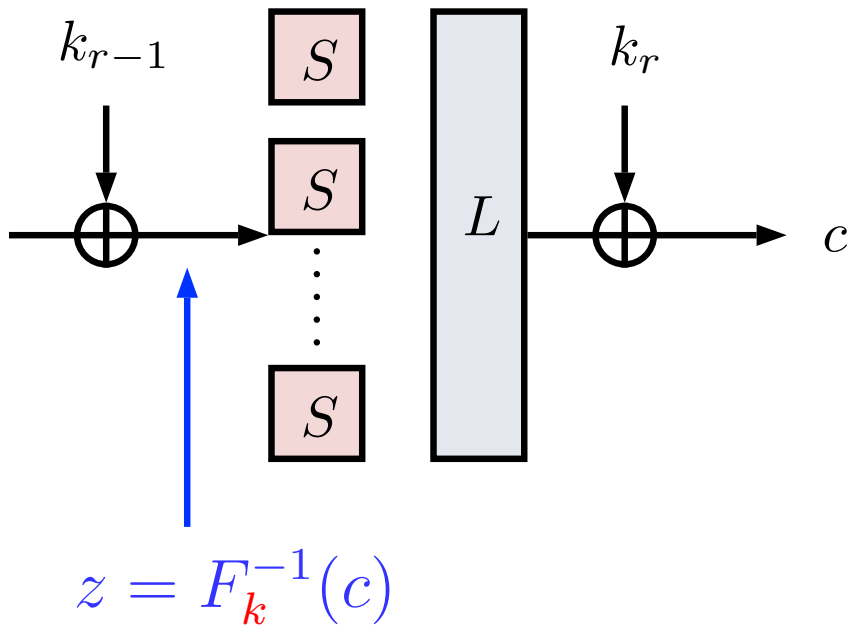


Use higher-order derivatives [Knudsen 94]

For all values of k check whether $m \mapsto z = F_k^{-1}(c)$ has degree $< d$.

How?

Check whether all derivatives of order d are zero.



The attack

Let V be a vector space of dimension d.

Input: Choose 2^d plaintexts of the form $m \oplus v$, $v \in V$ (coset of V) and get the corresponding ciphertexts.

Example $d = 3, m = 0$, $V = \langle v_1, v_2, v_3 \rangle$.

Chosen plaintexts: $0, v_1, v_2, v_3, v_1 \oplus v_2, v_1 \oplus v_3, v_2 \oplus v_3, v_1 \oplus v_2 \oplus v_3$.

If for a key k,

$$\bigoplus_{i=0}^{2^d-1} F_k^{-1}(c_i) \neq 0,$$

we conclude that k is a wrong key.

Number of candidate keys

What is the probability that for a wrong key, $\bigoplus_i F_k^{-1}(c_i) = 0$?
(false alarm probability)

$$P\left(\bigoplus_{i=0}^{2^d-1} F_k^{-1}(c_i) = 0\right) = 2^{-n},$$

where n is the block size.

- As there are 2^κ key candidates (κ is the size of a subkey), around $2^{\kappa-n}$ among them will be proposed as candidates for the right key.

Find the right candidate

How to find the right key among the left candidates ?

- Do an exhaustive search among the remaining candidates or
- Repeat the attack by choosing a different vector space of dimension d.

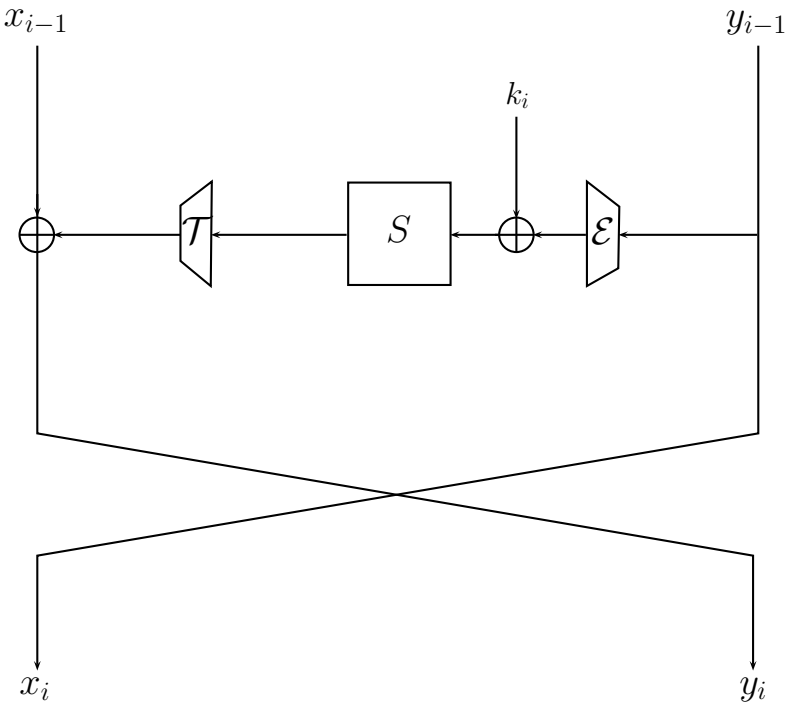
- **Data complexity**: 2^d chosen plaintexts.
- **Time complexity**: $2^d \times 2^\kappa$.

Remark In practice, we recover smaller fragments of the key.

The $\mathcal{KN}$ cipher [Knudsen-Nyberg 95]

6-round Feistel cipher

$\mathcal{E} : \mathbf{F}_2^{32} \to \mathbf{F}_2^{33}$ linear
$\mathcal{T} : \mathbf{F}_2^{33} \to \mathbf{F}_2^{32}$ linear
k_i : 33-bit subkey



$S : \mathbf{F}_{2^{33}} \to \mathbf{F}_{2^{33}}$
with $x \mapsto x^3$

$$\begin{array}{ccc} \mathbf{F}_2^{32} \times \mathbf{F}_2^{32} & \to & \mathbf{F}_2^{32} \times \mathbf{F}_2^{32} \\ (x, y) & \mapsto & (y, x \oplus \mathcal{T} \circ S\left(\mathcal{E}(x) \oplus k_i\right)) \end{array}$$

The role of the function S

- Name initially given to the cipher: CRADIC (Cipher Resistant Against Differential Cryptanalysis).
- The function S plays a crucial role.
- The function $x \mapsto x^3$ on the field $\mathbf{F}_2^{33}$ was chosen.
- This function is known to be resistant against linear and differential attacks.

But, this function is of degree 2.

Higher-order differential attack against $\mathcal{KN}$

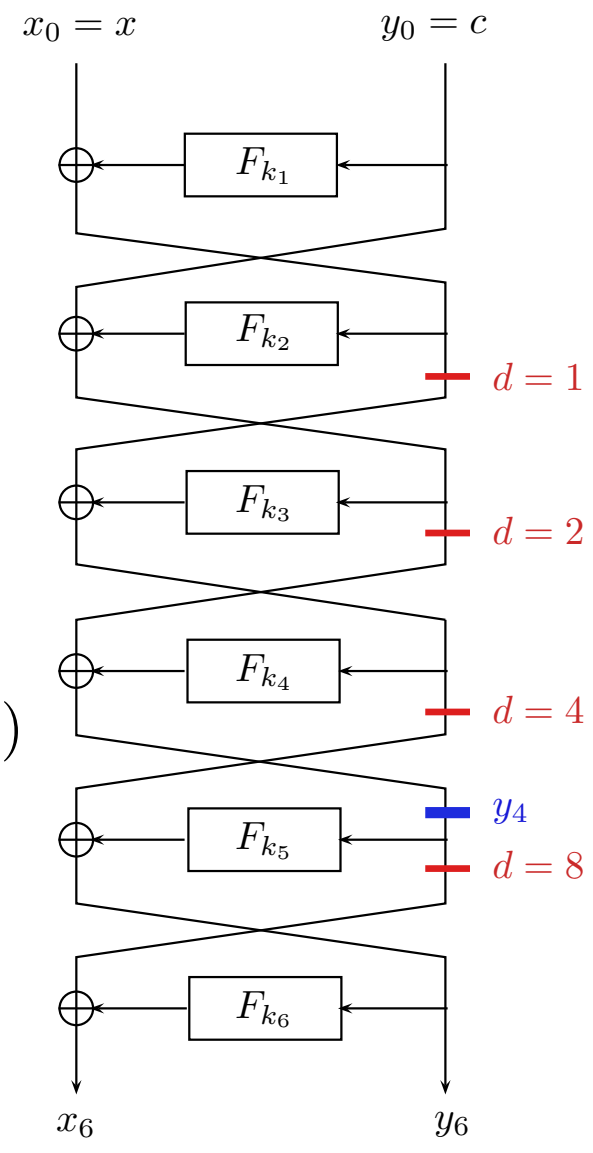
Presented by Jacobsen and Knudsen in 1997.

- Exploit the low algebraic degree of the round function.

Input: Plaintexts of the form $(x_0, y_0) \in \mathbf{F}_2^{32} \times \mathbf{F}_2^{32}$, where $y_0 = c$, for some constant c.

4 rounds of encryption

$$
\begin{aligned}
y_0(x) &= c \\
y_1(x) &= x \oplus F_{k_1}(c) := x \oplus c' \\
y_2(x) &= F_{k_2}(x \oplus c') \oplus c \\
y_3(x) &= F_{k_3}(F_{k_2}(x \oplus c') \oplus c) \oplus x \oplus c' \\
y_4(x) &= F_{k_4}(F_{k_3}(F_{k_2}(x \oplus c') \oplus c) \oplus x \oplus c') \\
&+ F_{k_2}(x \oplus c') \oplus c
\end{aligned}
$$



Evaluate the degree of y_4

$$y_4(x) = F_{k_4}(F_{k_3}(F_{k_2}(x \oplus c') \oplus c) \oplus x \oplus c') \oplus F_{k_2}(x \oplus c') \oplus c$$

Obviously, the degree of y_4 is bounded by the degree of

$$G = F_{k_4} \circ F_{k_3} \circ F_{k_2}$$

As $\deg(F_{k_i}) = \deg(S) = 2$, we get that

$$\begin{aligned} \deg(y_4) &\leq \deg(G) \leq \deg(F_{k_4}) \times \deg(F_{k_3}) \times \deg(F_{k_2}) \\ &\leq 2^3 \end{aligned}$$

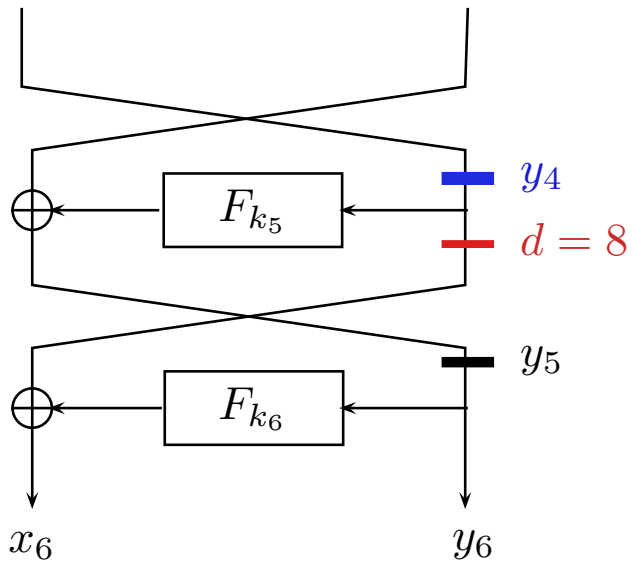
Write down the equations

If V is a subspace of $\mathbf{F}_2^{32}$ with $\dim(V) = 9$, we have:

$$D_V y_4(x) = \bigoplus_{v \in V} y_4(v \oplus x) = 0,$$

for all $x \in \mathbf{F}_2^{32}$. We get now the following equation:

$$\begin{aligned} x_6(x) &= F_{k_6}(y_5(x)) \oplus y_4(x), \\ y_4(x) &= F_{k_6}(y_5(x)) \oplus x_6(x) \end{aligned}$$



Attack equation

$$\bigoplus_{v \in V} F_{k_6}(y_5(v \oplus w)) \oplus \bigoplus_{v \in V} x_6(v \oplus w) = 0.$$

- Recover the key k_6.
- There will be in average $2^{33-32} = 2$ candidate keys for k_6.
- Recover the remaining subkeys by mounting the same attack on the reduced-round cipher.

Outline

Integral attacks - History

- Attack exploiting weaknesses of the non-linear as well as the linear layer of the target cipher.
- In 1997, the SQUARE cipher was presented by Daemen, Knudsen and Rijmen.
- During the design, the authors discover a new chosen-plaintext attack against 6 rounds of the cipher.
- This new attack was named the square attack.
- In the beginning the attack was applied against SPN ciphers.
- Later, Lucks generalizes the attack to other type of ciphers and call it the saturation attack.
- In 2002, Knudsen and Wagner unify the different aspects of these attacks and give them the name integral attacks.

Multisets

Multiset: Every element in the set can appear multiple times.

- An element of a multiset is a pair (value, multiplicity).

Example. $V = \{1, 2, 2, 2, 3, 3, 4\}$, or $V = \{(1, 1), (2, 3), (3, 2), (4, 1)\}$

The attacker studies the propagation of the multiset through the cipher.

Integral over a multiset

Application to word-oriented ciphers.

Notation: w number of words in a plaintext. (**e.g**. AES: 16 words of 8-bits each).

- Choose plaintexts in a way that the multiset in each word verifies a specific property.

Definition. We call integral over a multiset S the sum

$$\sum_{v \in S} v$$

Properties

An attacker tries to predict the values in the integrals after a certain number of rounds.

Distinguish between 3 cases.

(For the examples, the word-size is 3 bits.)

1. $\mathcal{C}$: All w words in the multiset have the same constant value.
   - The multiset $S = \{3, 3, 3, 3, 3, 3, 3, 3\}$ has the property $\mathcal{C}$.
2. $\mathcal{A}$: The w words in the multiset take all possible values.
   - The multiset $S = \{0, 1, 2, 3, 4, 5, 6, 7\}$ has the property $\mathcal{A}$.
3. $\mathcal{B}$: The integral over S is 0.

Example: AES

16 words of 8 bits.

- 2^8 chosen plaintexts m_i of the form

$$(m_i, c, c, c, c, c, c, c, c, c, c, c, c, c, c, c),$$

where $m_i = i$, for $i = 0, \ldots, 255$ and c some constant.

$\mathcal{A}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$

- Analyze how this multiset propagates through the different operations of AES.

Through AddRoundKey

The **same** constant value is XORed to each byte.

Example.

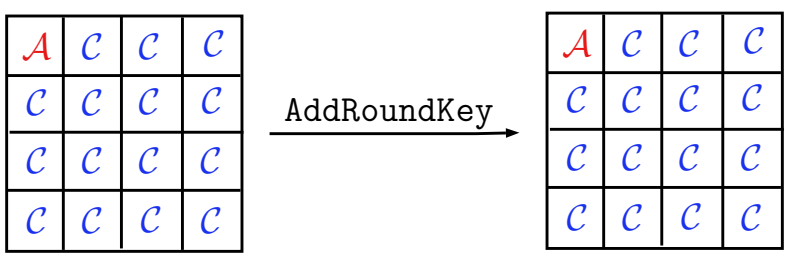
$$(\texttt{0x06}, \ldots, \texttt{0x06}) \rightarrow (\texttt{0x06} \oplus \texttt{0x01}, \ldots, \texttt{0x06} \oplus \texttt{0x01}) = (\texttt{0x07}, \ldots, \texttt{0x07})$$

- $\mathcal{C} \rightarrow \mathcal{C}$

Property. If we XOR the **same** constant value to each different value of a set having $\mathcal{A}$ we get again all possible values in the set.

Example. $S = \{\texttt{0x0},\texttt{0x1},\texttt{0x2},\texttt{0x3}\}$, $k = \texttt{0x2}$, $S \oplus k = \{\texttt{0x2},\texttt{0x3},\texttt{0x0},\texttt{0x1}\}$

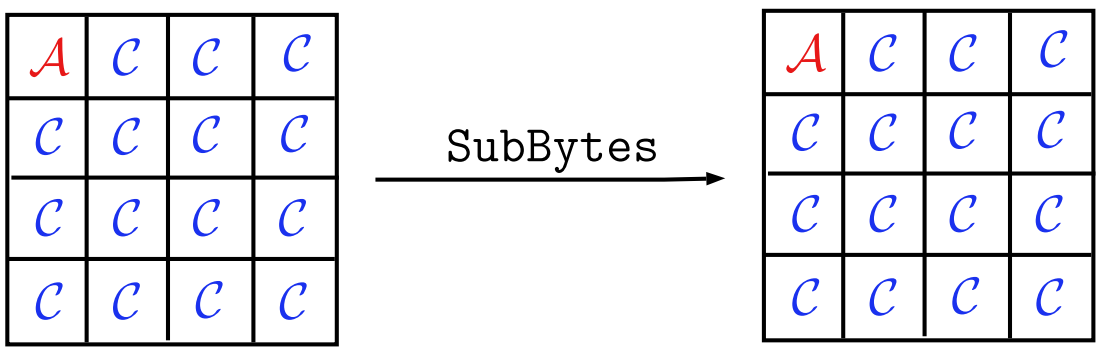
- $\mathcal{A} \rightarrow \mathcal{A}$



Through SubBytes

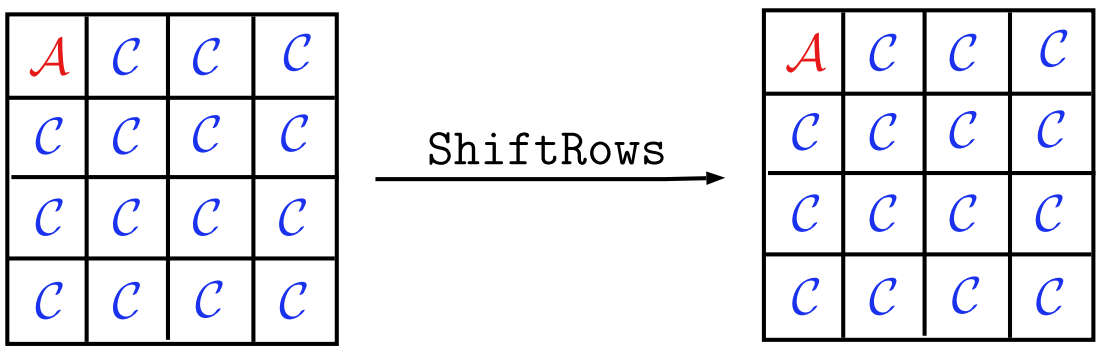
The Sbox S is a permutation.

- If all values of a multiset have the same constant value c, all values will have the same constant value $c' = S(x)$ after SubBytes. $\mathcal{C} \rightarrow \mathcal{C}$
- If the values of a multiset take all possible values, the Sbox will only permute these values. $\mathcal{A} \rightarrow \mathcal{A}$



Through ShiftRows

ShiftRows only permutes the bytes of the state.



Through MixColumns (I)

- Inputs of the 1^{st} column: $(x_0^i, x_1^i, x_2^i, x_3^i)$, $0 \leq i \leq 255$
- Outputs of the 1^{st} column: $(y_0^i, y_1^i, y_2^i, y_3^i)$, $0 \leq i \leq 255$

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} x_0^i \\ x_1^i \\ x_2^i \\ x_3^i \end{pmatrix} = \begin{pmatrix} y_0^i \\ y_1^i \\ y_2^i \\ y_3^i \end{pmatrix}$$

$$\begin{aligned} y_0^0 &= 02 \cdot x_0^0 + 03 \cdot x_1^0 + 01 \cdot x_2^0 + 01 \cdot x_3^0 \\ y_0^1 &= 02 \cdot x_0^1 + 03 \cdot x_1^1 + 01 \cdot x_2^1 + 01 \cdot x_3^1 \\ &\vdots \quad \ldots \\ y_0^{255} &= 02 \cdot x_0^{255} + 03 \cdot x_1^{255} + 01 \cdot x_2^{255} + 01 \cdot x_3^{255} \end{aligned}$$

Through MixColumns (I)

- Inputs of the 1st column: $(x_0^i, x_1^i, x_2^i, x_3^i)$, $0 \leq i \leq 255$
- Outputs of the 1st column: $(y_0^i, y_1^i, y_2^i, y_3^i)$, $0 \leq i \leq 255$

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} x_0^i \\ x_1^i \\ x_2^i \\ x_3^i \end{pmatrix} = \begin{pmatrix} y_0^i \\ y_1^i \\ y_2^i \\ y_3^i \end{pmatrix}$$

$$\begin{aligned} y_0^0 &= 02 \cdot x_0^0 + 03 \cdot x_1^0 + 01 \cdot x_2^0 + 01 \cdot x_3^0 \\ y_0^1 &= 02 \cdot x_0^1 + 03 \cdot x_1^1 + 01 \cdot x_2^1 + 01 \cdot x_3^1 \\ &\vdots \quad \ldots \\ y_0^{255} &= 02 \cdot x_0^{255} + 03 \cdot x_1^{255} + 01 \cdot x_2^{255} + 01 \cdot x_3^{255} \end{aligned}$$

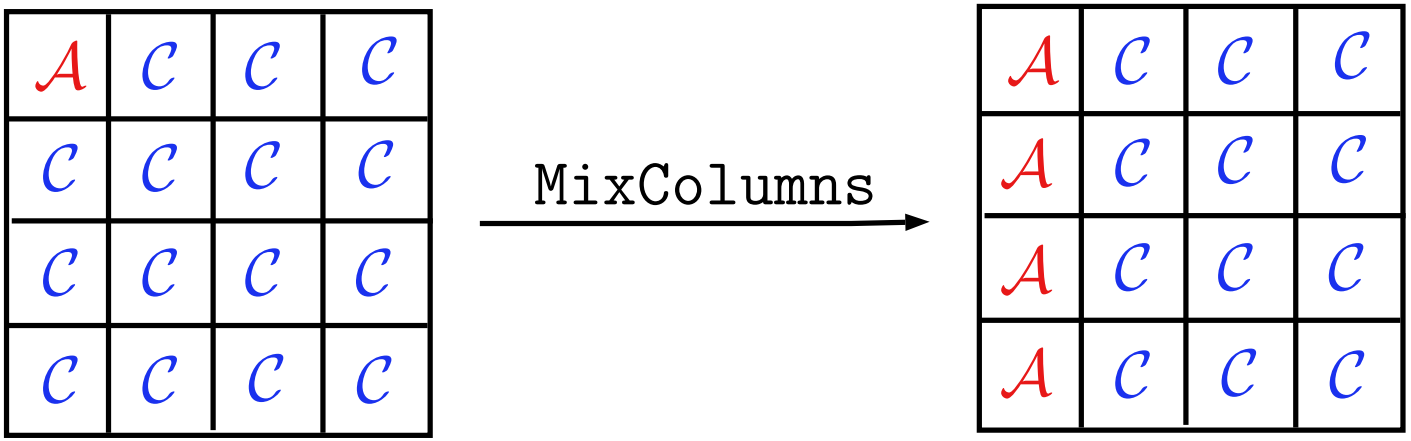
Through MixColumns (I)

- Inputs of the 1st column: $(x_0^i, x_1^i, x_2^i, x_3^i)$, $0 \leq i \leq 255$
- Outputs of the 1st column: $(y_0^i, y_1^i, y_2^i, y_3^i)$, $0 \leq i \leq 255$

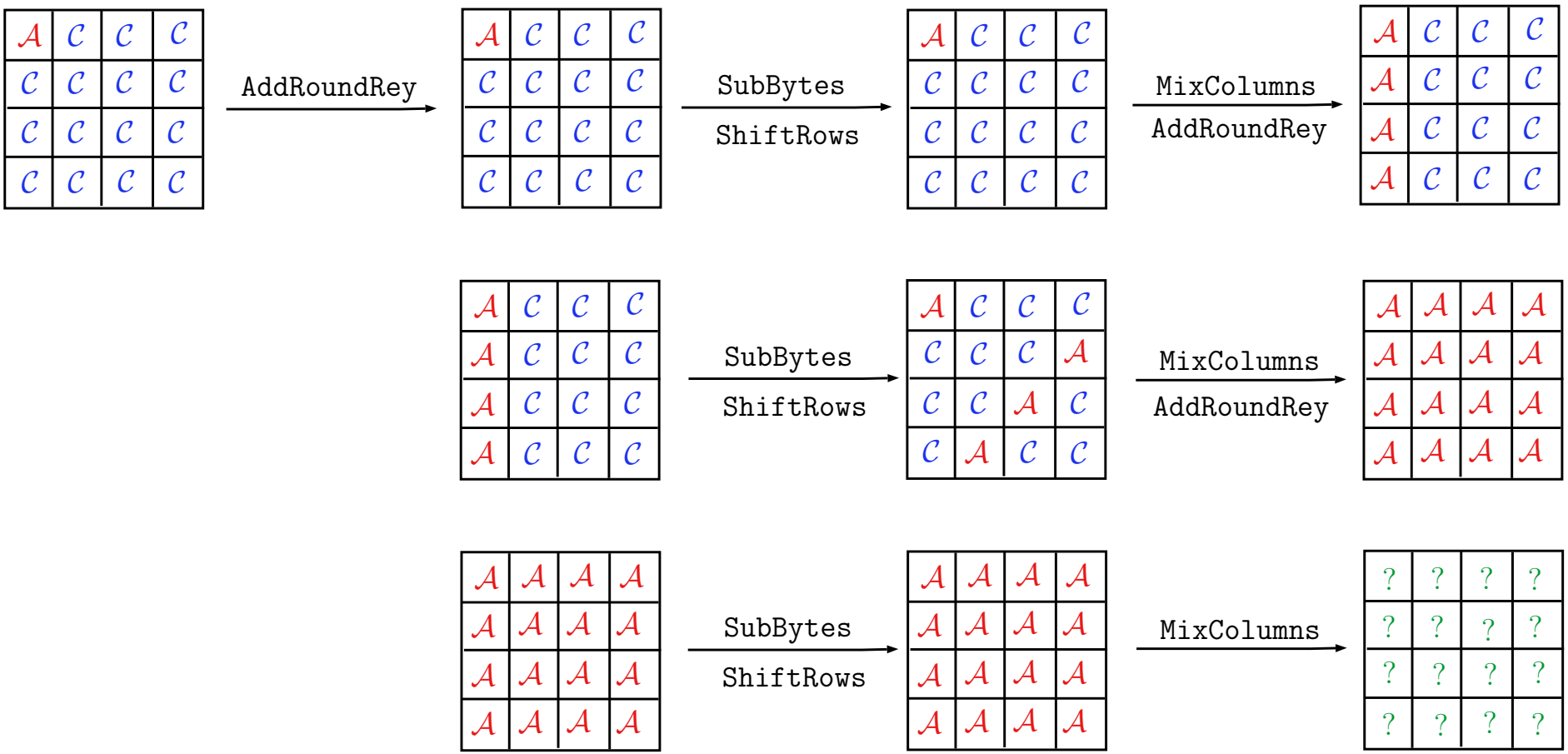
$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} x_0^i \\ x_1^i \\ x_2^i \\ x_3^i \end{pmatrix} = \begin{pmatrix} y_0^i \\ y_1^i \\ y_2^i \\ y_3^i \end{pmatrix}$$

$$\begin{aligned} y_0^0 &= 02 \cdot x_0^0 + c \\ y_0^1 &= 02 \cdot x_0^1 + c \\ &\vdots \quad \ldots \\ y_0^{255} &= 02 \cdot x_0^{255} + c \end{aligned}$$

Through MixColumns (II)



After 3 rounds

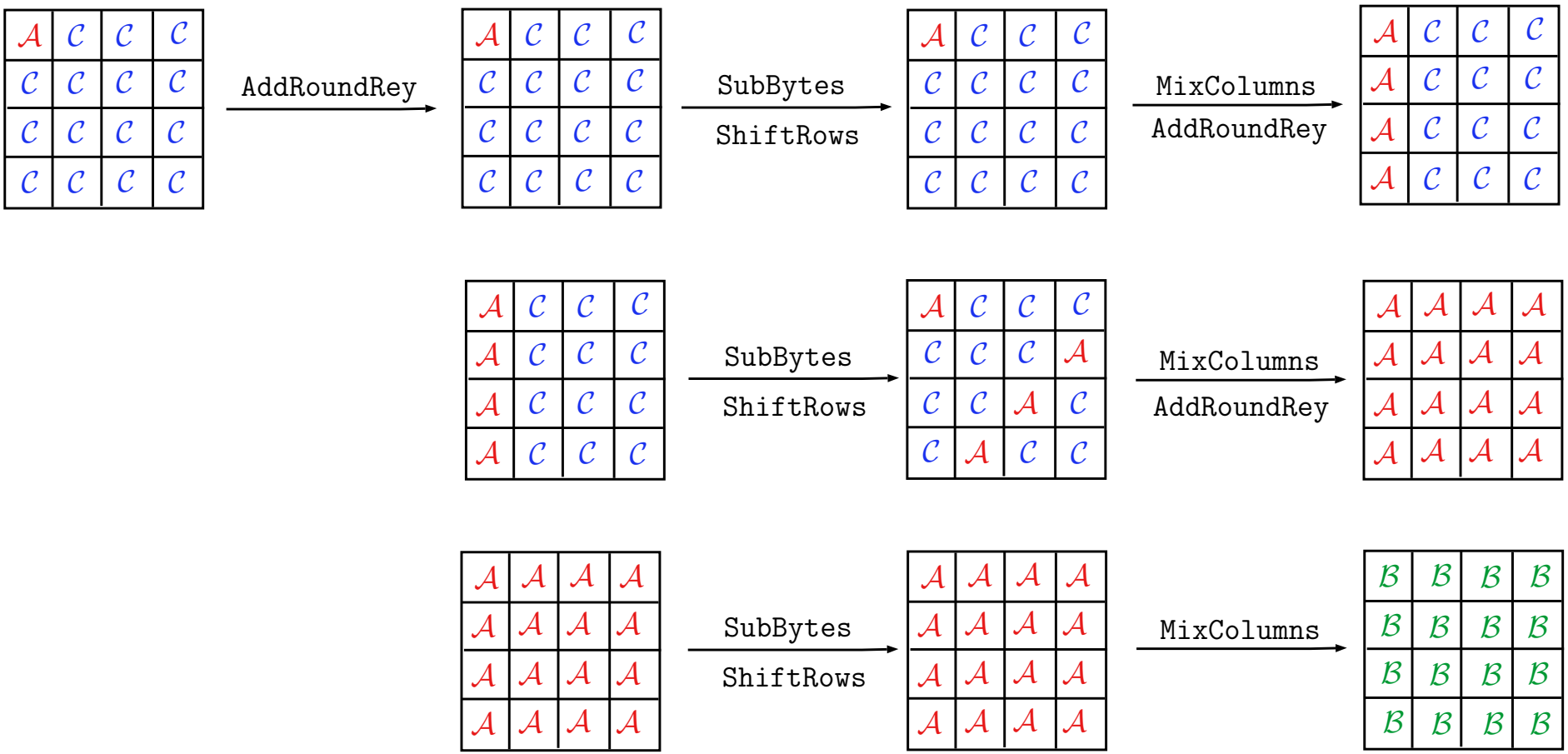


After MixColumns

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} x_0^i \\ x_1^i \\ x_2^i \\ x_3^i \end{pmatrix} = \begin{pmatrix} y_0^i \\ y_1^i \\ y_2^i \\ y_3^i \end{pmatrix}$$

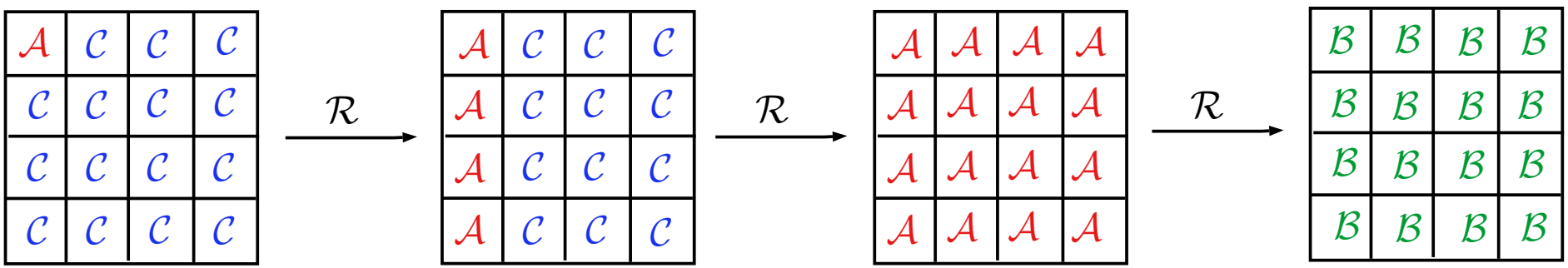
$$\begin{aligned} y_0^0 \oplus \cdots \oplus y_0^{255} &= 02 \cdot x_0^0 \oplus 03 \cdot x_1^0 \oplus 01 \cdot x_2^0 \oplus 01 \cdot x_3^0 \\ &\oplus 02 \cdot x_0^1 \oplus 03 \cdot x_1^1 \oplus 01 \cdot x_2^1 \oplus 01 \cdot x_3^1 \\ &\vdots \\ &\oplus 02 \cdot x_0^{255} \oplus 03 \cdot x_1^{255} \oplus 01 \cdot x_2^{255} \oplus 01 \cdot x_3^{255} \\ &= 02 \cdot \bigoplus_{i=0}^{255} x_0^i \oplus 03 \cdot \bigoplus_{i=0}^{255} x_1^i \oplus 01 \cdot \bigoplus_{i=0}^{255} x_2^i \oplus 01 \cdot \bigoplus_{i=0}^{255} x_3^i \\ &= 02 \cdot 00 \oplus 03 \cdot 00 \oplus 01 \cdot 00 \oplus 01 \cdot 00 \\ &= 00. \end{aligned}$$

After 3 rounds of AES



Distinguishing property for 3 rounds of AES

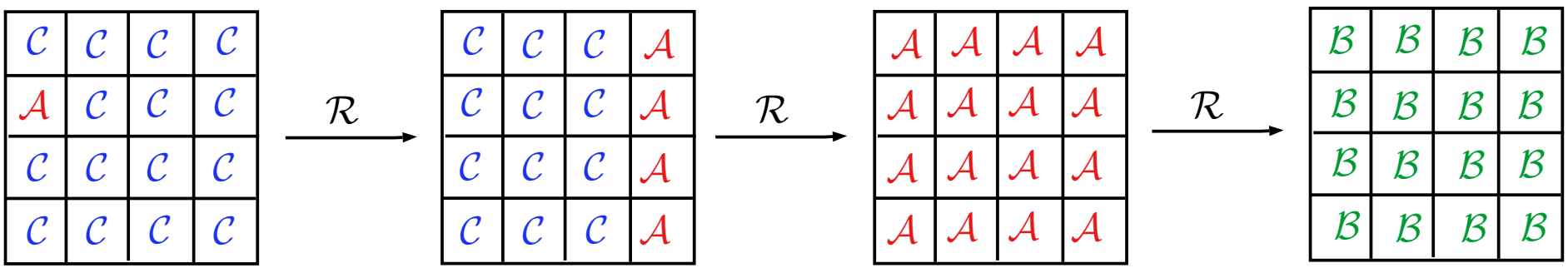
- After the 3rd `MixColumns` **every** byte position will be balanced (XOR of all 256 values in a single byte position is 0).
- Property that holds with probability 1.
- Property independent of the key.



- The byte taking all 256 values (saturated) can be any of the 16 bytes.

Distinguishing property for 3 rounds of AES

- After the 3rd `MixColumns` **every** byte position will be balanced (XOR of all 256 values in a single byte position is 0).
- Property that holds with probability 1.
- Property independent on the key.



- The byte taking all 256 values (saturated) can be any of the 16 bytes.

Attack over AES reduced to 4 rounds

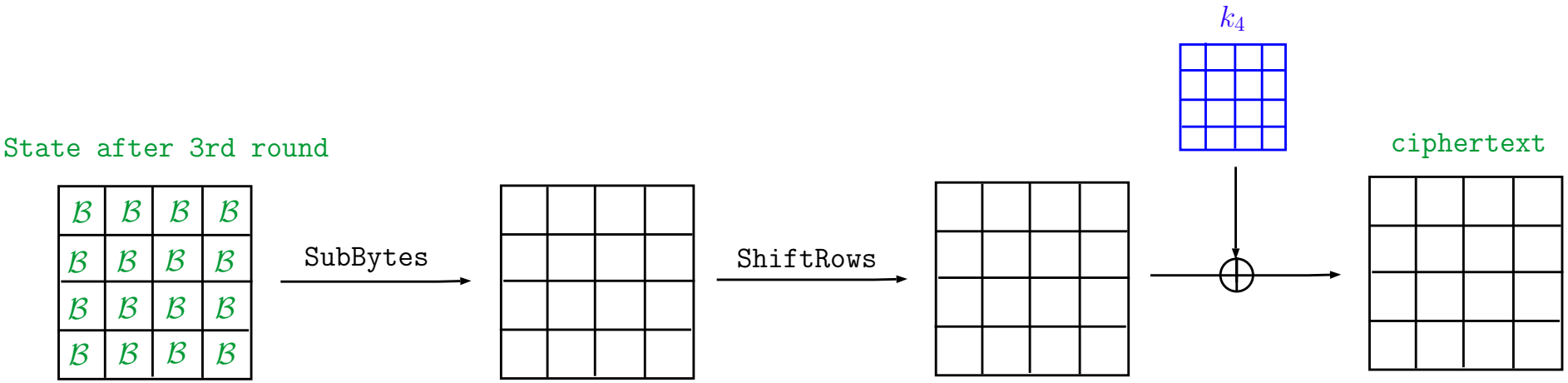
Goal: Recover the subkey k_4 of the 4^{th} round of AES.

Remark No `MixColumns` in the last round.

Input: 256 chosen plaintexts m_i of the form

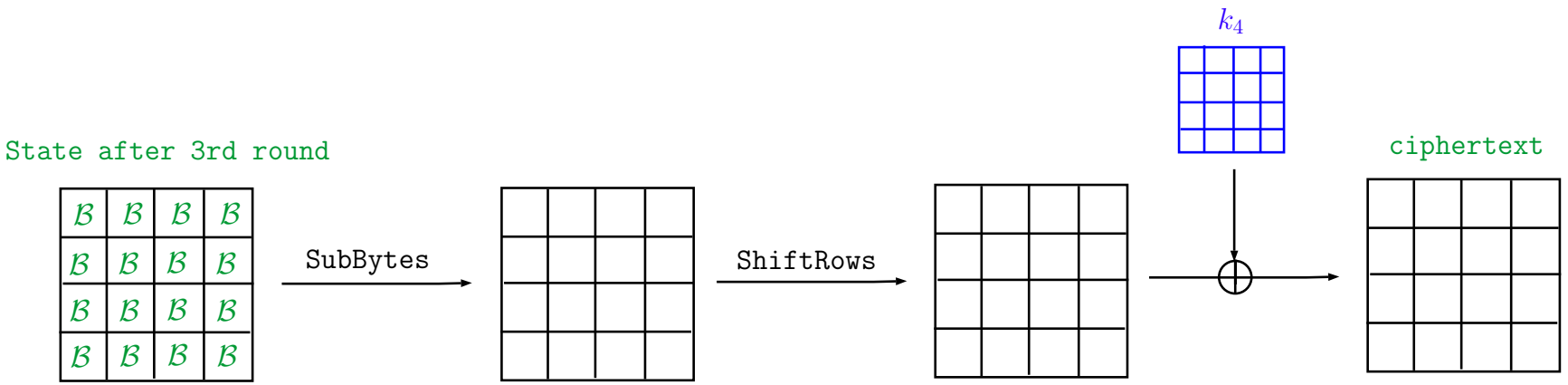
$$(x_i, c, c, c, c, c, c, c, c, c, c, c, c, c, c, c),$$

where $x_i = i$, for $i = 0, \ldots, 255$ and c some constant and the corresponding ciphertexts c_i, $i = 0, \ldots, 255$.



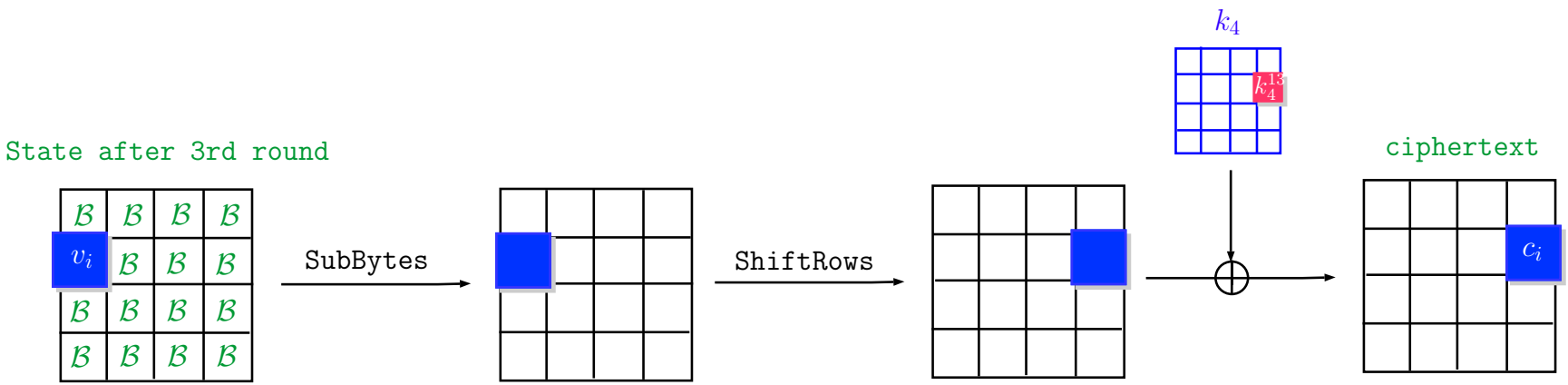
Divide and conquer

- Subkey k_4 is 128-bits long (exhaustive search not possible!).
- Use a divide and conquer strategy and recover the last subkey byte by byte.



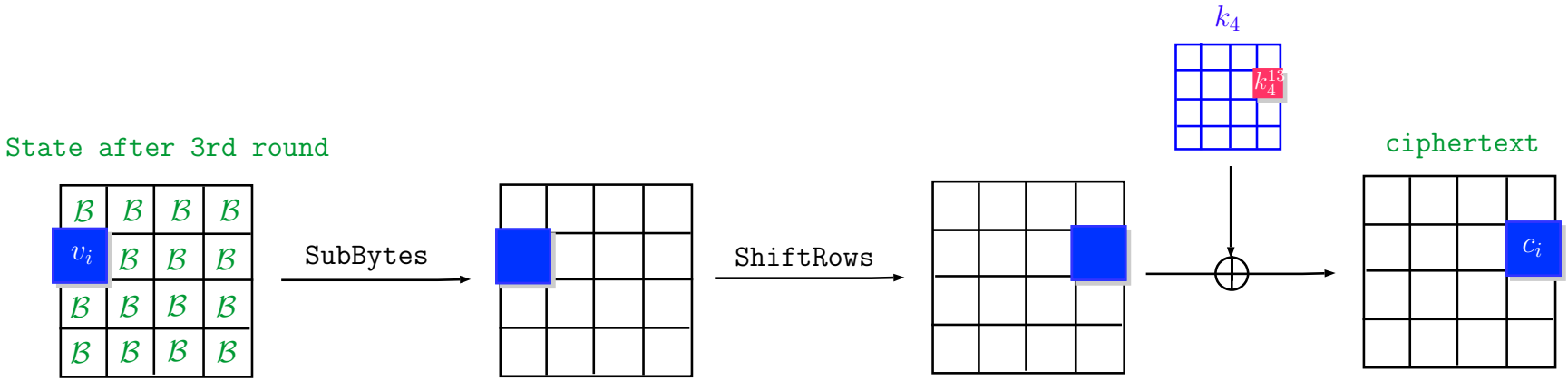
Divide and conquer

- Subkey k_4 is 128-bits long (exhaustive search not possible!).
- Use a divide and conquer strategy and recover the last subkey byte by byte.



$$
\begin{aligned}
c_i &= S(v_i) \oplus k_4^{13} \\
v_i &= S^{-1}(c_i \oplus k_4^{13})
\end{aligned}
$$

Divide and conquer



$$
\begin{aligned}
c_i &= S(v_i) \oplus k_4^{13} \\
v_i &= S^{-1}(c_i \oplus k_4^{13})
\end{aligned}
$$

But, if k_4^{13} is the right value

$$
\bigoplus_{i=0}^{255} v_i = \bigoplus_{i=0}^{255} S^{-1}(c_i \oplus k_4^{13}) = 0
$$

Complexity

- **Data complexity**: 2^8 chosen plaintext-ciphertext pairs (a little bit more to get rid off false alarms)
- **Time complexity**: $\approx 16 \times 2^8 \times 2^8 = 2^{20}$ XOR's.
- Assume that a full encryption is composed of 2^6 similar simple operations. So, time complexity $\approx 2^{14}$ encryptions.

Link with higher-order differential cryptanalysis

- A differential of order d is the sum of 2^d vecteurs of a well-chosen vector space, so it can be seen as an integral.
- Recently, Yosuke Todo extended integral attacks to take in a clearer way the algebraic degree into account. This extension is called the division property.

Outline

Iterated permutations

Most of the symmetric constructions (hash functions, block ciphers) are based on a **permutation iterated a high number of times**.

Important to estimate the **algebraic degree** of such iterated permutations.

Functions with a low degree are vulnerable to:

- Algebraic attacks
- Higher-order differential attacks and distinguishers

A trivial bound

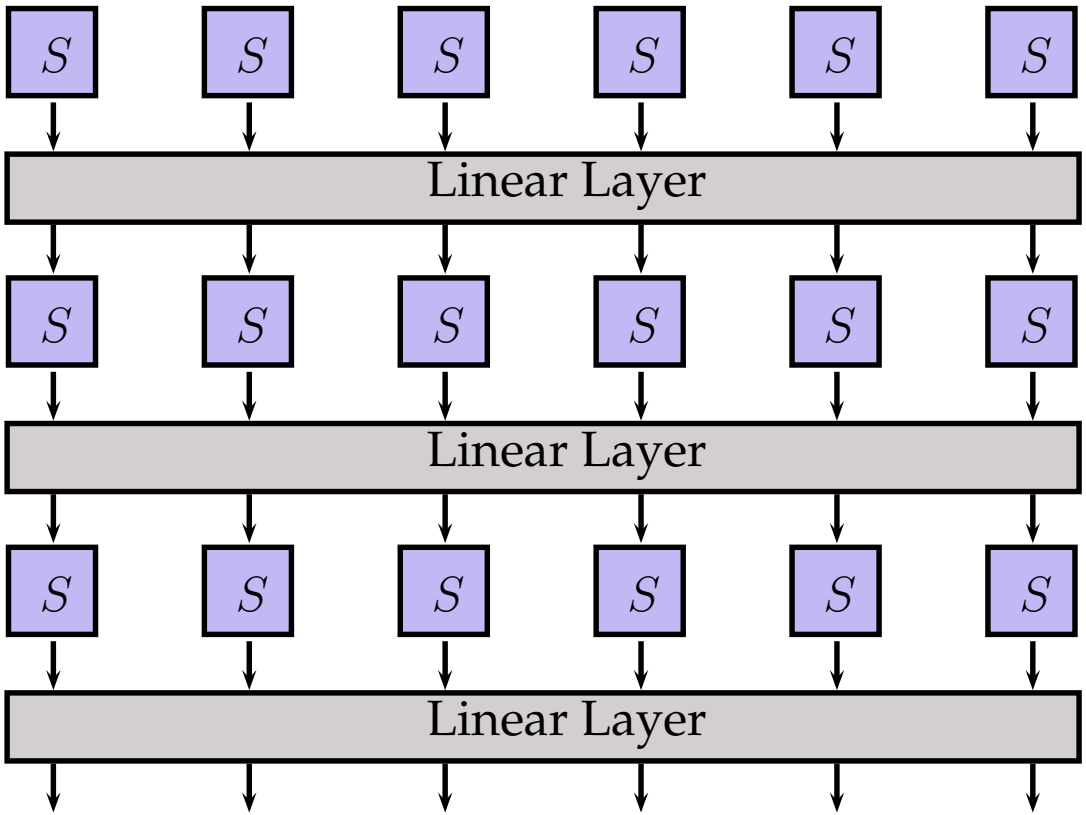
Proposition: Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ and G a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. Then

$$\deg(G \circ F) \leq \deg(G) \deg(F).$$

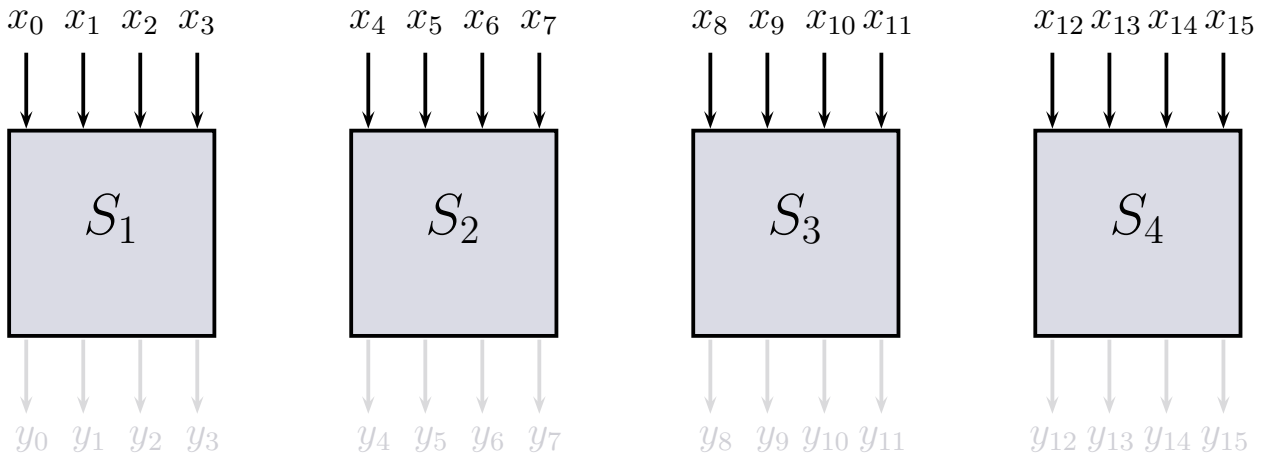
Example: Round function R of AES is of degree 7. Then

$$\deg(R^2) = \deg(R \circ R) \leq 7^2 = 49.$$

Substitution Permutation Networks

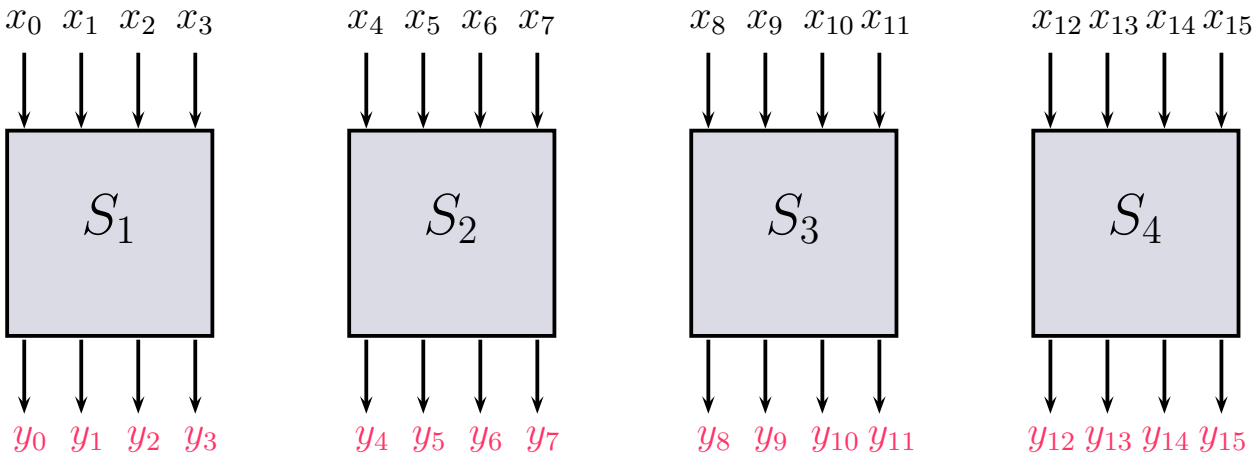


How to estimate the evolution of the degree of such constructions?



After several rounds, all coordinates can be expressed as a sum of monomials.

Each monomial is a **product** of variables in $X = \{x_0, \ldots, x_{15}\}$.



After several rounds, all coordinates can be expressed as a sum of monomials.

Each monomial is a **product** of variables in $Y = \{y_0, \ldots, y_{15}\}$.

The coordinates $y_0 - y_3$ are outputs of the same Sbox (equally for the others).

What is the consequence on the degree of the product ?

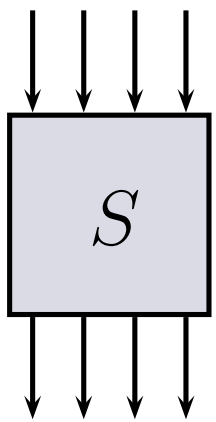
The notion of δ_k

Definition : For a permutation S define $\delta_k(S)$ as the maximum degree of the product of k coordinates of S.

$\rightarrow \delta_1(S) :=$ algebraic degree of S

Example:

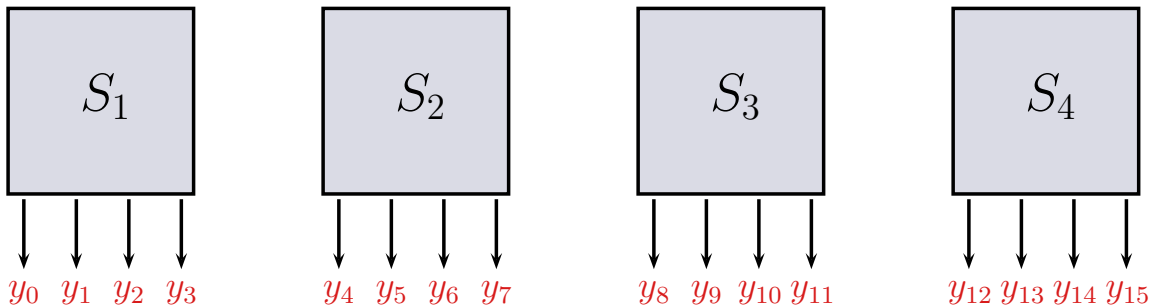
$\deg S = 3$



k	δ_k
1	3
2	3
3	3
4	4

S permutation of $\mathbf{F}_2^n$:
$\delta_k(S) = n$ iff $k = n$.

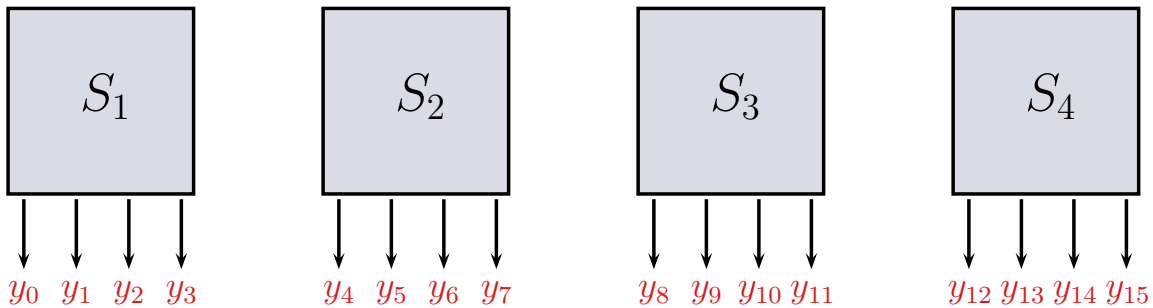
Example: Product of 6 coordinates.



$\pi = y_0y_1y_3y_8y_9y_{10}$.

$$\deg(\pi) \leq \delta_3(S_1) + \delta_3(S_3) = 6.$$

Example: Product of 6 coordinates.

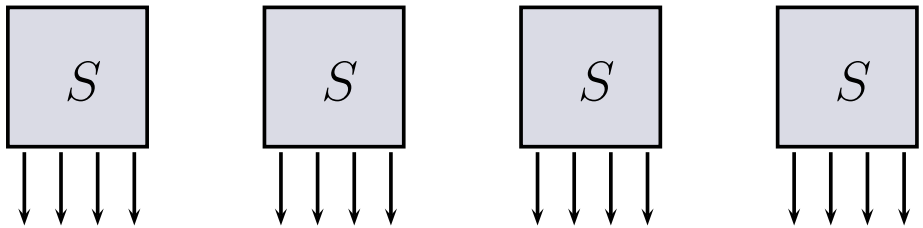


$\pi = y_0y_5y_8y_{10}y_{13}y_{15}$.

$$\deg(\pi) \leq \delta_1(S_1) + \delta_1(S_2) + \delta_2(S_3) + \delta_2(S_4) = 12.$$

The degree of the product is relatively low if many coordinates coming from the same Sbox are involved!

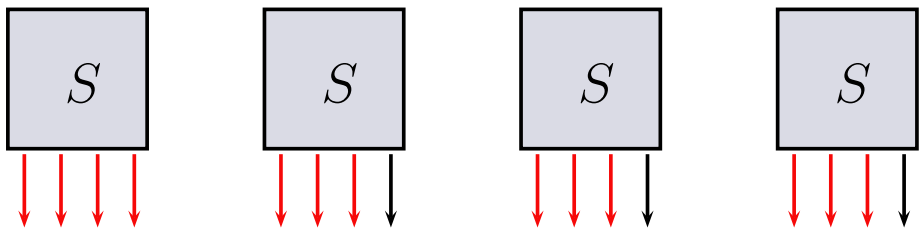
Towards the bound



Find the maximal degree of the product π of d outputs.

$x_i = \#$ Sboxes for which exactly i coordinates are involved in π.

Towards the bound



Find the maximal degree of the product π of d outputs.

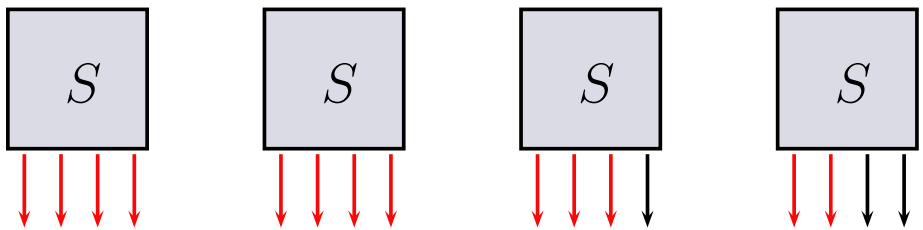
$x_i = \#$ Sboxes for which exactly i coordinates are involved in π.

Example ($d = 13$)

- $x_4 = 1$, $x_3 = 3$:

$$\deg(\pi) \leq \delta_3 x_3 + \delta_4 x_4 = 3 \cdot 3 + 4 \cdot 1 = 13.$$

Towards the bound



Find the maximal degree of the product π of d outputs.

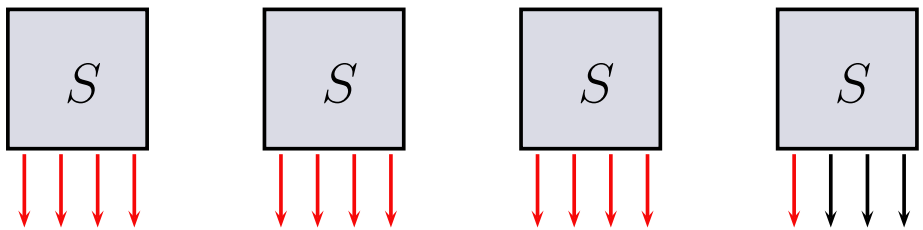
$x_i = \#$ Sboxes for which exactly i coordinates are involved in π.

Example ($d = 13$)

- $x_4 = 2$, $x_3 = 1$, $x_2 = 1$:

$$\deg(\pi) \leq \delta_2 x_2 + \delta_3 x_3 + \delta_4 x_4 = 3 \cdot 1 + 3 \cdot 1 + 4 \cdot 2 = 14.$$

Towards the bound



Find the maximal degree of the product π of d outputs.

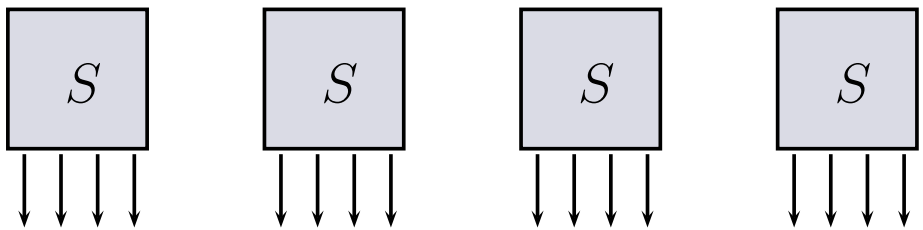
$x_i = \#$ Sboxes for which exactly i coordinates are involved in π.

Example ($d = 13$)

- $x_4 = 3$, $x_1 = 1$:

$$\deg(\pi) \leq \delta_1 x_1 + \delta_4 x_4 = 3 \cdot 1 + 4 \cdot 3 = 15.$$

Towards the bound



Find the maximal degree of the product π of d outputs.

$x_i = \#$ Sboxes for which exactly i coordinates are involved in π.

$$\deg(\pi) \leq \max_{(x_1,x_2,x_3,x_4)} (\delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + \delta_4 x_4)$$

with $x_1 + 2x_2 + 3x_3 + 4x_4 = d$.

d	x_4	x_3	x_2	x_1	$\deg(\pi)$
16	4	-	-	-	16
15	3	1	-	-	15
14	3	-	1	-	15
13	3	-	-	1	15
12	2	1	-	1	14
11	2	-	1	1	14
10	2	-	-	2	14
9	1	1	-	2	13
⋮	⋮	⋮	⋮	⋮	⋮

$$16 - \deg(\pi) \geq \frac{16 - d}{3}$$

d	x_4	x_3	x_2	x_1	$\deg(\pi)$
16	4	-	-	-	16
15	3	1	-	-	15
14	3	-	1	-	15
13	3	-	-	1	15
12	2	1	-	1	14
11	2	-	1	1	14
10	2	-	-	2	14
9	1	1	-	2	13
⋮	⋮	⋮	⋮	⋮	⋮

$$\deg(\pi) \leq 16 - \frac{16 - d}{3}$$

A bound on the degree of SPN constructions

[Boura – Canteaut – De Cannière - 11]

Theorem. Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ corresponding to the parallel application of an Sbox, S, defined over $\mathbf{F}_2^{n_0}$.
Then, for any G from $\mathbf{F}_2^n$ into $\mathbf{F}_2^\ell$, we have

$$\deg(G \circ F) \leq n - \frac{n - \deg G}{\gamma(S)},$$

where

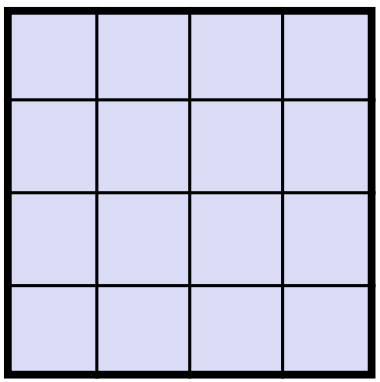
$$\gamma(S) = \max_{1 \leq i \leq n_0 - 1} \frac{n_0 - i}{n_0 - \delta_i}.$$

Application to AES

One round:

$$\texttt{MC} \circ \texttt{SR} \circ \texttt{SB} \circ \texttt{AK}.$$

- AK: AddRoundKey
- SB: SubBytes (Sboxes of degree 7)
- SR: ShiftRows
- MC: MixColumns



The Super Sbox technique

Two rounds:

$$R^2 = \mathtt{MC} \circ \mathtt{SR} \circ \mathtt{SB} \circ \mathtt{AK} \circ \mathtt{MC} \circ \mathtt{SR} \circ \mathtt{SB} \circ \mathtt{AK}.$$

Equivalently:

$$R^2 = \mathtt{MC} \circ \mathtt{SR} \circ \mathtt{SB} \circ \mathtt{AK} \circ \mathtt{MC} \circ \mathtt{SB} \circ \mathtt{SR} \circ \mathtt{AK}.$$

Denote:

$$\mathtt{SuperSbox} = \mathtt{SB} \circ \mathtt{AK} \circ \mathtt{MC} \circ \mathtt{SB}.$$

Then:

$$R^2 = \mathtt{MC} \circ \mathtt{SR} \circ \mathtt{SuperSbox} \circ \mathtt{SR} \circ \mathtt{AK}.$$

Bound on up to 4 rounds

SuperSbox: $\mathbf{F}_2^{32} \to \mathbf{F}_2^{32}$: Two non-linear layers composed of Sboxes of degree 7, separated by a linear layer.

$$\deg(\texttt{SuperSbox}) \leq 32 - \frac{32-7}{7} \leq 28.$$

(Trivial Bound: $\deg(R^2) \leq 7^2 = 49$!!!)

Bound for r rounds:

$$\deg(R^r) = \deg(R^{r-1} \circ R) \leq 128 - \frac{128 - \deg(R^{r-1})}{7}.$$

- $r = 3$: $\deg(R^3) \leq 113$
- $r = 4$: $\deg(R^4) \leq 125$