The Block Cipher SQUARE

Halil İbrahim Kaplan

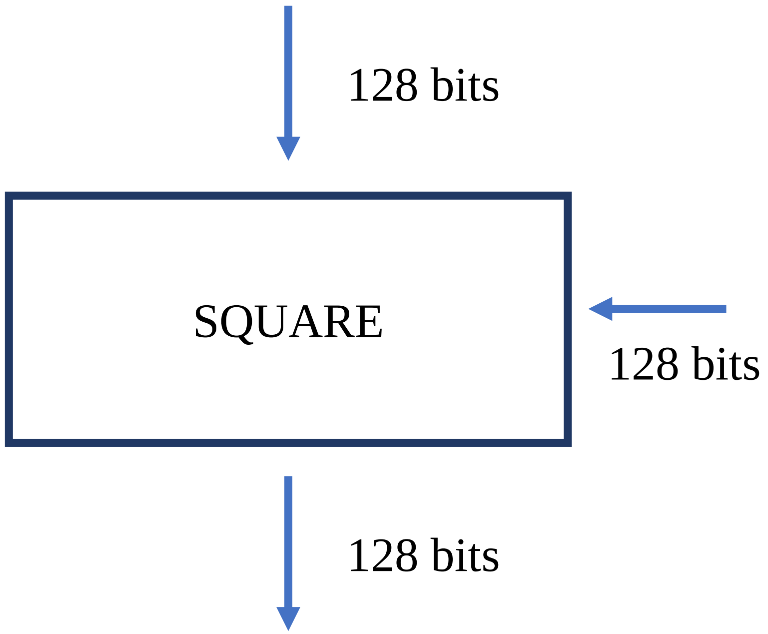
2021

TÜBİTAK BİLGEM

Overview

- **Structure of SQUARE**
- **Wide Trail Design Strategy**
- **The Multiplication Polynomial c(x)**
- **The Nonlinear Substitution γ**
- **Dedicated Attack**

Structure of SQUARE



Input :

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

Rows can be considered as polynomials

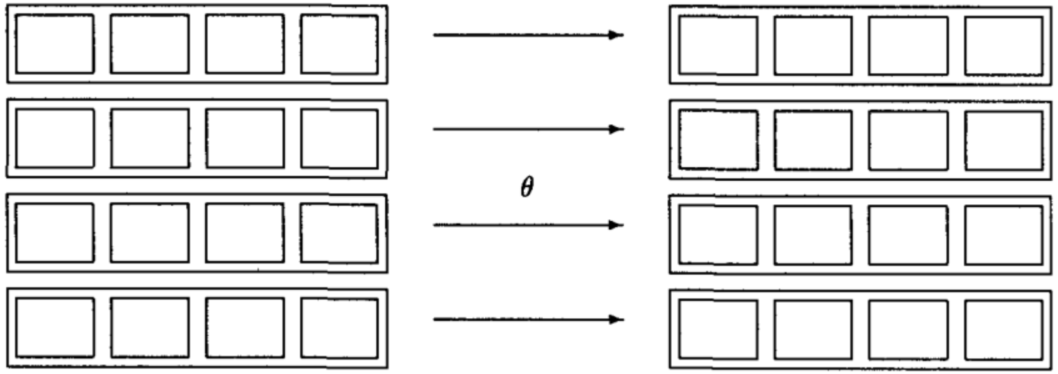
$$\boldsymbol{a_i(x) = a_{i,0} \oplus a_{i,1}\, x \oplus a_{i,2}\, x^2 \oplus a_{i,3}\, x^3}$$

where $a_{i,j}$ *is also polynomial in* GF(2^8)

Structure of SQUARE

1) Linear Transformation θ

Multiplication in GF(2^8) with polynomial x^4+1:



$Define\ c(x) = c_0 \oplus c_1\, x \oplus c_2\, x^2 \oplus c_3\, x^3$

For each $a_i(x) = a_{i,0} \oplus a_{i,1}\, x \oplus a_{i,2}\, x^2 \oplus a_{i,3}\, x^3$

θ defined as :

$$b = \theta(a) \Leftrightarrow b_i(x) = c(x)a_i(x) \bmod 1 \oplus x^4 \qquad \text{for} \qquad 0 \leq i < 4.$$

θ^{-1} defined as :

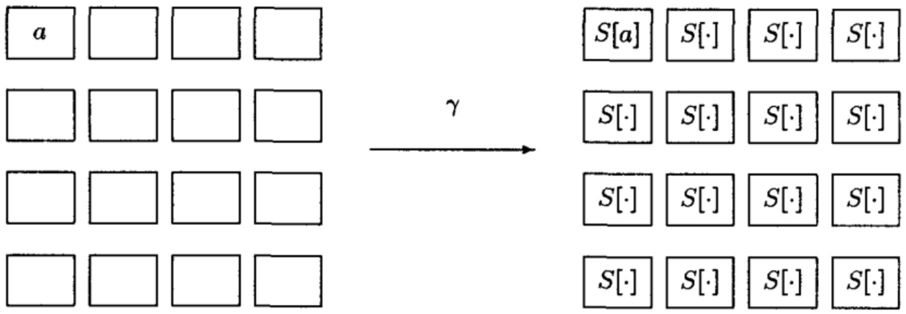
$$d(x)c(x) = 1 \pmod{1 \oplus x^4}$$

2) Nonlinear Transformation γ

Identical for all bytes We have :

$$\gamma : b = \gamma(a) \Leftrightarrow b_{i,j} = S_\gamma(a_{i,j})$$

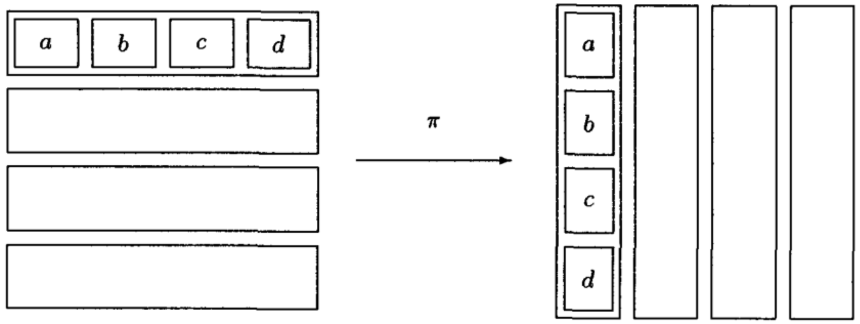
where S_γ is invertable 8 bits S-box



3) Byte Permutation π

Interchanging Rows and Columns

$$\pi : b = \pi(a) \Leftrightarrow b_{i,j} = a_{j,i}$$



$\pi^{-1} = \pi$

4) Bitwise Round Key Addition σ

Bitwise addition of a round key k^t

$$\sigma[k^t] : b = \sigma[k^t](a) \Leftrightarrow b = a \oplus k^t.$$

The inverse of $\sigma[k^t]$ is $\sigma[k^t]$ itself.

5) The Round Key *Evaluation* ψ

$k^0 = Cipher\ key\ K$

Other round keys are derived iteratively by invertible affine transformation ψ

$$\psi : k^t = \psi(k^{t-1})$$

- **6) The Key Evolution ψ**

The key schedule is defined in terms of the rows of the key. We can define a left byte-rotation operation $\mathsf{rotl}(a_i)$ on a row as

$$\mathsf{rotl}[a_{i,0}a_{i,1}a_{i,2}a_{i,3}] = [a_{i,1}a_{i,2}a_{i,3}a_{i,0}]$$

and a right byte rotation $\mathsf{rotr}(a_i)$ as its inverse.

The key schedule iteration transformation $k^{t+1} = \psi(k^t)$ and its inverse are defined by

$$\begin{aligned} k_0^{t+1} &= k_0^t \oplus \mathsf{rotl}(k_3^t) \oplus C_t \\ k_1^{t+1} &= k_1^t \oplus k_0^{t+1} \\ k_2^{t+1} &= k_2^t \oplus k_1^{t+1} \\ k_3^{t+1} &= k_3^t \oplus k_2^{t+1} \end{aligned} \qquad \begin{aligned} \kappa_3^{t+1} &= \kappa_3^t \oplus \kappa_2^t \\ \kappa_2^{t+1} &= \kappa_2^t \oplus \kappa_1^t \\ \kappa_1^{t+1} &= \kappa_1^t \oplus \kappa_0^t \\ \kappa_0^{t+1} &= \kappa_0^t \oplus \mathsf{rotr}(\kappa_3^t) \oplus C_t' \end{aligned}$$

The simplicity of the inverse key schedule is thanks to the fact that θ and ψ commute. The round constants C_t are also defined iteratively. We have $C_0 = 1_{\mathtt{x}}$ and $C_t = 2_{\mathtt{x}} \cdot C_{t-1}$.

$$k^0 = \begin{pmatrix} k_0^0 \\ k_1^0 \\ k_2^0 \\ k_3^0 \end{pmatrix}$$

Structure of SQUARE

In SQUARE

Linear Parts = θ and π

Nonlinear Part = γ

The Cipher SQUARE :

The building blocks are composed into the round transformation denoted by $\rho[k^t]$:

$$\boldsymbol{\rho[k^t] = \sigma[k^t] o\ \pi\ o\ \gamma\ o\ \theta} \quad \textbf{(1)}$$

SQUARE is defined as eight rounds proceeded by a key addition $\sigma[k^0]$ and by θ^{-1}:

$$\textbf{SQUARE [k]} = \boldsymbol{\rho[k^8]}\ \mathbf{o}\ \boldsymbol{\rho[k^7]}\ \mathbf{o}\ \boldsymbol{\rho[k^6]}\ \mathbf{o}\ \boldsymbol{\rho[k^5]}\ \mathbf{o}\ \boldsymbol{\rho[k^4]}\ \mathbf{o}\ \boldsymbol{\rho[k^3]}\ \mathbf{o}\ \boldsymbol{\rho[k^2]}\ \mathbf{o}\ \boldsymbol{\rho[k^1]}\ \mathbf{o}\ \boldsymbol{\sigma[k^0]}\ \mathbf{o}\ \boldsymbol{\theta^{-1}}\ \textbf{(2)}$$

The Cipher SQUARE :

$$\text{SQUARE [k]} = \rho[k^8] \text{ o } \rho[k^7] \text{ o } \rho[k^6] \text{ o } \rho[k^5] \text{ o } \rho[k^4] \text{ o } \rho[k^3] \text{ o } \rho[k^2] \text{ o } \rho[k^1] \text{ o } \sigma[k^0] \text{ o } \theta^{-1}$$

$\boldsymbol{\theta^{-1}}$ can be discarded by omitting $\boldsymbol{\theta}$ in the first round and applying $\boldsymbol{\sigma[\theta(k^0)]}$ instead of $\boldsymbol{\sigma[k^0]}$:

$$
\begin{aligned}
\rho[k^1] \text{ o } \sigma[k^0] \text{ o } \theta^{-1} &= \sigma[k^1] o\ \pi\ o\ \gamma\ o\ \theta \text{ o } \sigma[k^0] \text{ o } \theta^{-1}(\text{a}) \\
&= \sigma[k^1] o\ \pi\ o\ \gamma\ o\ \theta \text{ o } \sigma[k^0] \text{ o } d * a \\
&= \sigma[k^1] o\ \pi\ o\ \gamma\ o\ \theta \text{ o } k^0 \oplus d * \text{a}) \\
&= \sigma[k^1] o\ \pi\ o\ \gamma\ o \text{ c} * (\ k^0 \oplus d * \text{a}) \\
&= \sigma[k^1] o\ \pi\ o\ \gamma\ o \text{ c} * k^0 \oplus c * d * \text{a}) \\
&= \sigma[k^1] o\ \pi\ o\ \gamma\ o \text{ c} * k^0 \oplus \text{a} \\
&= \sigma[k^1] o\ \pi\ o\ \gamma\ o\ \sigma[\theta(k^0)]\ (\text{a})
\end{aligned}
$$

Wide Trail Design Strategy

1) Choose an S-box where the maximum difference propagation probability and the maximum input-output correlation are as small as possible.

Gives two criteria for the selection of the S_γ

2) Choose the linear part (θ and π) in such a way that there are no linear trails with few active S-boxes.

Gives hint to how to select c(x)

The Multiplication Polynomial c(x)

Branch Number :

Let F be a linear transformation acting on byte vectors

Byte weight : number of nonzero bytes in the vector denoted by W(a)

$$B\,(F) = \min_{a \neq 0}(\mathrm{W}(a) + \mathrm{W}(F(a))).$$

The Branch Number of a linear transformation is a measure of its diffusion power

The Multiplication Polynomial c(x)

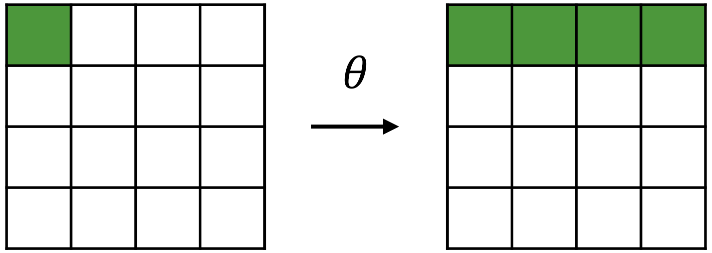
Branch Number :

$$B(F) = \min_{a\neq 0}(W(a)+W(F(a))).$$

Upper bound for the branch number is 5.

The coefficients of c(x) have been chosen in such a way that the upper bound is reached.

A non-zero byte is called an active byte. For θ it can be seen that if a state is applied with a single active byte, the output can have at most 4 active bytes, as θ acts on the rows independently.



If the branch number is 5, a difference in 1 input (or output) byte propagates to all 4 output (or input) bytes, a 2-byte input (or output) difference to at least 3 output (or input) bytes.

$$c(x) = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

$\boldsymbol{Define\ \lambda =}$ Highest occuring correlation probability between any pair of linear combination of input bits and linear combinations of output bits.

Exor table of γ : $E_{ij} = \#\{x | S(x) \oplus S(x \oplus i) = j\}$

Define δ $= \dfrac{max_{i,j}\{E_{ij}\}}{2^8}$

We will Show 3 alternative choice of S-box :

The Nonlinear Substitution γ

$\lambda = Linear$
$\delta = Differential$

1) Explicit Construction

$$Select\ the\ mapping\ x \longrightarrow x^{-1}\ over\ GF(2^8)\ with\ \delta = 2^{-6}\ \ , \lambda = 2^{-3}$$

Problem : Mapping has very simple description in $GF(2^8)$ like in other components.

This may enable cryptanalytic attacks based on the algebraic manipulation of equations to derive key information.

By choosing a **different basis** for the definition of γ and θ we can prevent that the round transformation has a simple description in any basis of $GF(2^8)$

2) Modification

$$Select\ the\ mapping\ x \longrightarrow x^{-1}\ over\ GF(2^8)\ and\ \boldsymbol{modify\ it}\ to\ prevent\ simple\ algebraic\ description$$

Problem : λ *and*/or δ *will increase*

Consider mapping as a look-up table swap some pairs

300.000 variants	4 entry swapped	8 entry swapped
	Increases λ *to* 9 x 2^{-6}	Increases λ *to* 9 x 2^{-6} Increases δ *to* 6 x 2^{-8}

The Nonlinear Substitution γ

$\lambda = Linear$
$\delta = Differential$

3) Random Search

1.5 million samples with m = 8 and measured at the same time δ and λ.

The results are given in table 2. The S-boxes with the highest resistance against both linear and differential cryptanalysis have

$\delta = 10 * 2^{-8}$ and $\lambda = 15 * 2^{-6}$

λ	δ						
	$8 \cdot 2^{-8}$	$10 \cdot 2^{-8}$	$12 \cdot 2^{-8}$	$14 \cdot 2^{-8}$	$16 \cdot 2^{-8}$	$18 \cdot 2^{-8}$	$20 \cdot 2^{-8}$
15×2^{-6}	0	$\boxed{0.07}$	0.07	0.006	0.0001	0	0
16×2^{-6}	0.0003	4.77	5.58	0.58	0.04	0.002	0
17×2^{-6}	0.002	15.63	20.55	2.24	0.15	0.007	0.0004
18×2^{-6}	0.0002	12.21	17.17	1.96	0.13	0.007	0.0005
19×2^{-6}	0.0004	4.91	7.31	0.87	0.05	0.003	0
20×2^{-6}	0	1.52	2.34	0.28	0.02	0.001	0
21×2^{-6}	0	0.41	0.64	0.08	0.004	0.001	0

Table 2. Maximum input-output correlation and difference propagation probability of randomly generated nonlinear permutations. The entries denote the percentage of the generated mappings that have the indicated λ and δ.

Our Choice :

$Mapping\ x \longrightarrow x^{-1}\ over\ GF(2^8)\ with\ \delta = 2^{-6}$, $\lambda = 2^{-3}$ because of its **optimal values.**

Differential trial probability < 2^{-150} << Critical noise value 2^{-127}

Linear trial probability < 2^{-75} << Critical noise value 2^{-64}

Therefore,

Resistance against LC and DC **six round** may seem sufficient.

Dedicated Attack

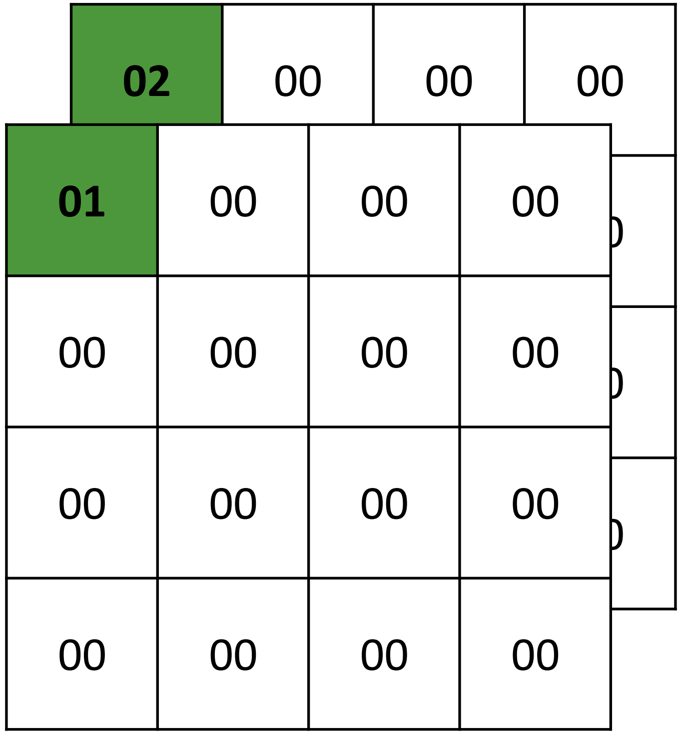
$Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t] \circ \pi \circ \gamma \circ \theta$

- Chosen plaintext attack

- Let Λ be the set of 256 plaintexts that are all different in some of the (16) state bytes (the active) and all equal in the other state bytes (the passive)

- Let λ be the set of indices of the active bytes. Then we have :

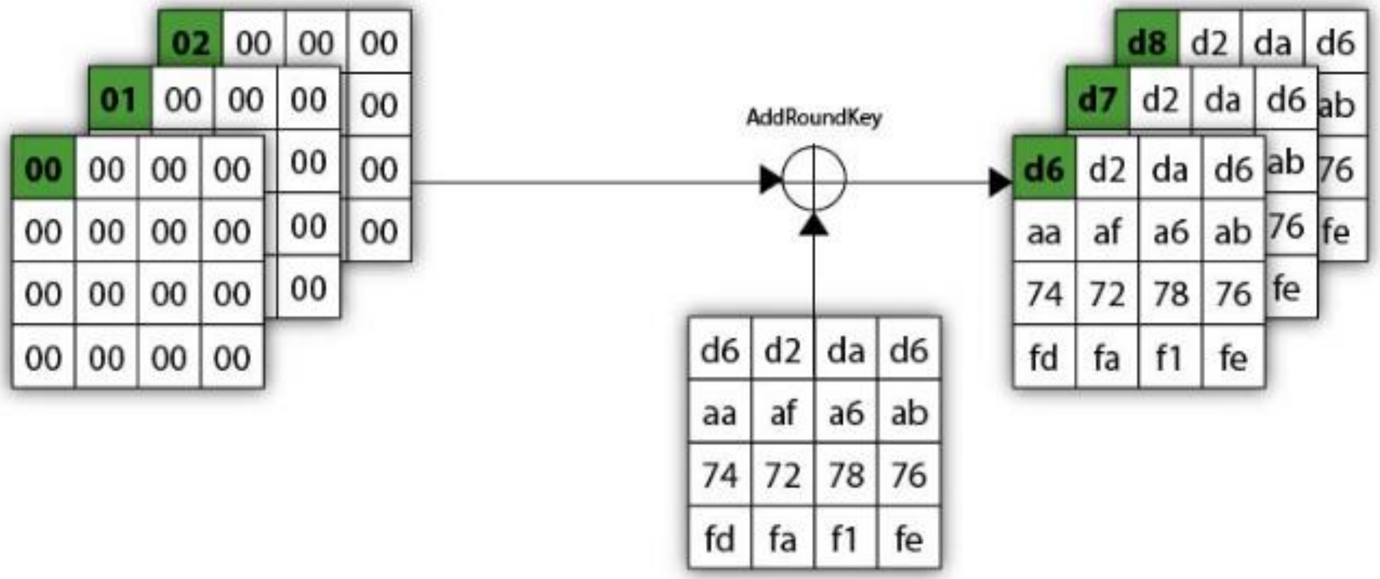
$$\forall x, y \in \Lambda : \begin{cases} x_{i,j} \neq y_{i,j} \text{ for } (i,j) \in \lambda \\ x_{i,j} = y_{i,j} \text{ for } (i,j) \notin \lambda \end{cases}$$



Green = Active

White = Passive

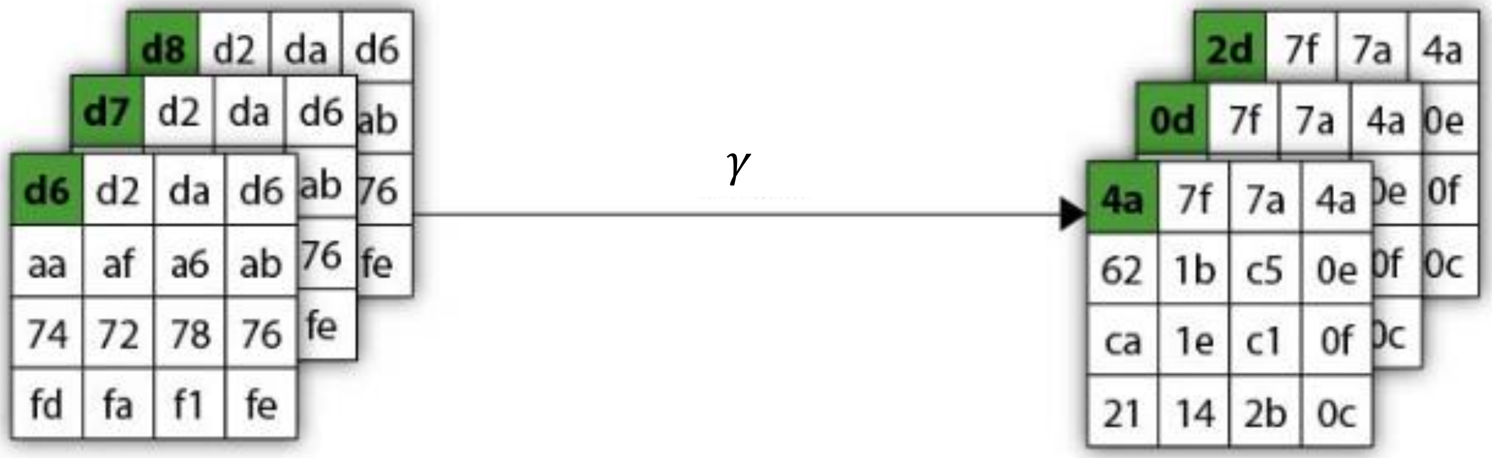
Applying $\sigma[k^t]$ *on a set* Λ **results** Λ *set* with same λ



Dedicated Attack

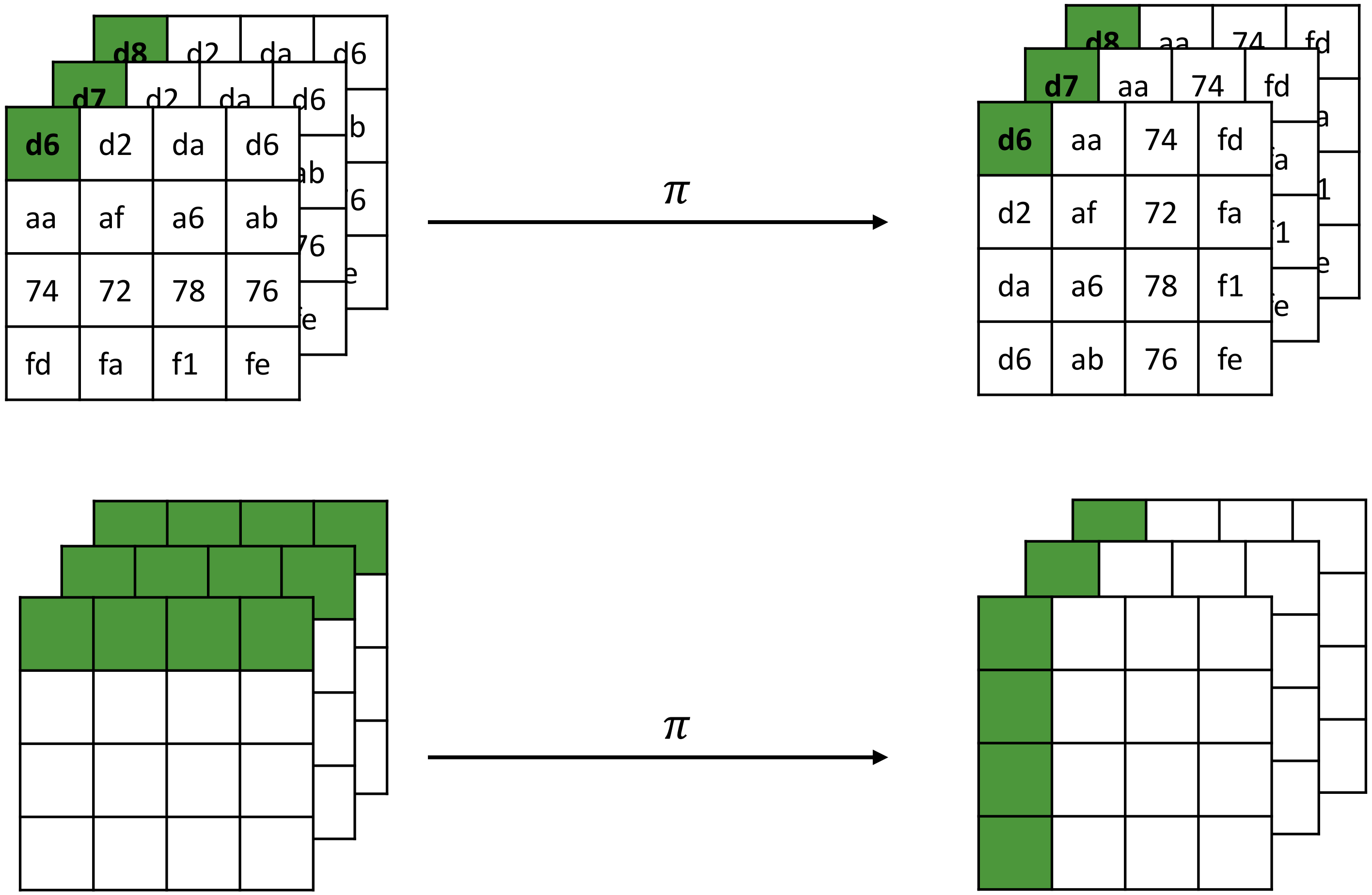
$\boldsymbol{Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t] o\ \pi\ o\ \gamma\ o\ \theta}$

Applying γ *on a set* Λ ***results*** Λ *set* with same λ



$$Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t]\, o\ \pi\ o\ \gamma\ o\ \theta$$

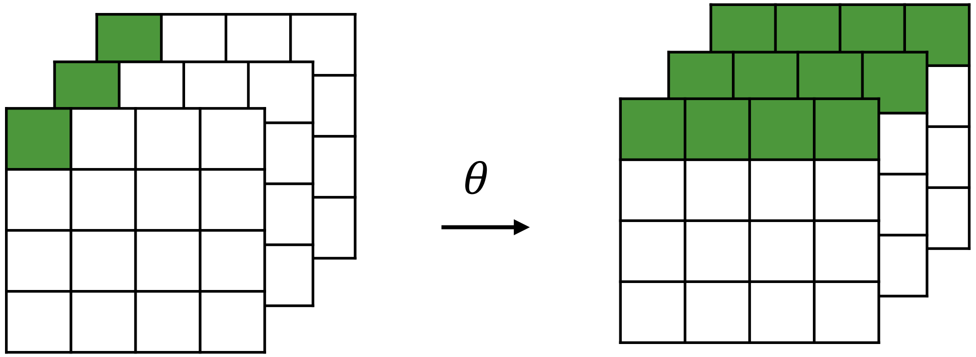
Applying π *on a set* Λ ***results*** Λ *set* which the active bytes are transposed by π

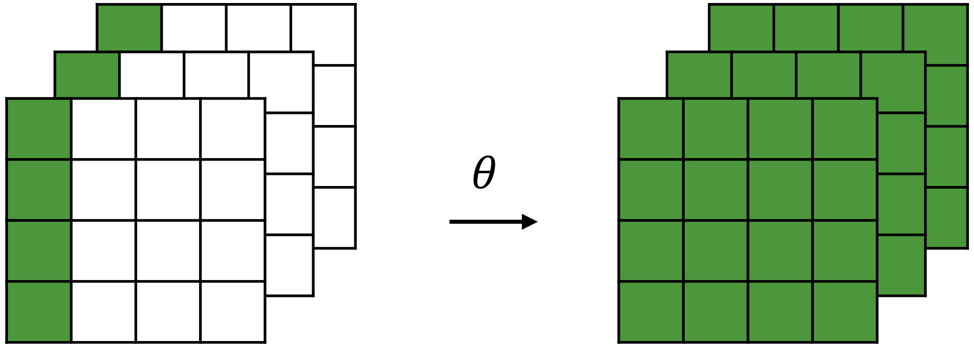


Dedicated Attack

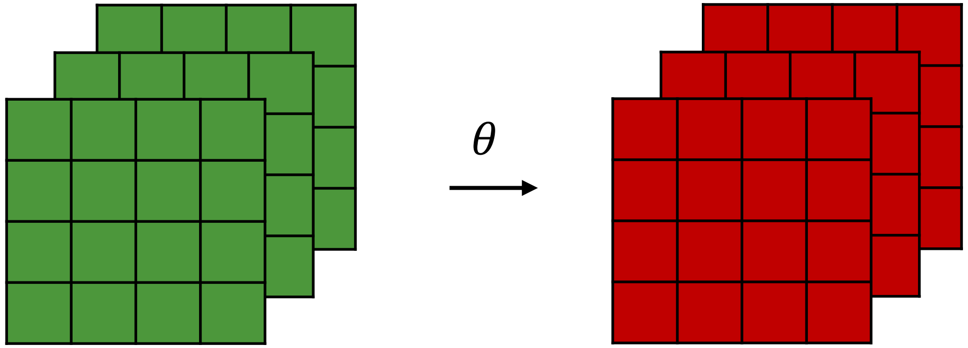
Round of SQUARE $= \boldsymbol{\rho[k^t] = \sigma[k^t] o\, \pi\, o\, \gamma\, o\, \theta}$

Applying θ *on a set* Λ ***does not necessarily result*** *in* Λ *set*





$$\begin{pmatrix} a & p0 & p1 & p2 \\ p3 & p4 & p5 & p6 \\ p7 & p8 & p9 & p10 \\ p11 & p12 & p13 & p14 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 3 \\ 3 & 2 & 1 & 1 \\ 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 2 \end{pmatrix}$$



UNKNOWN

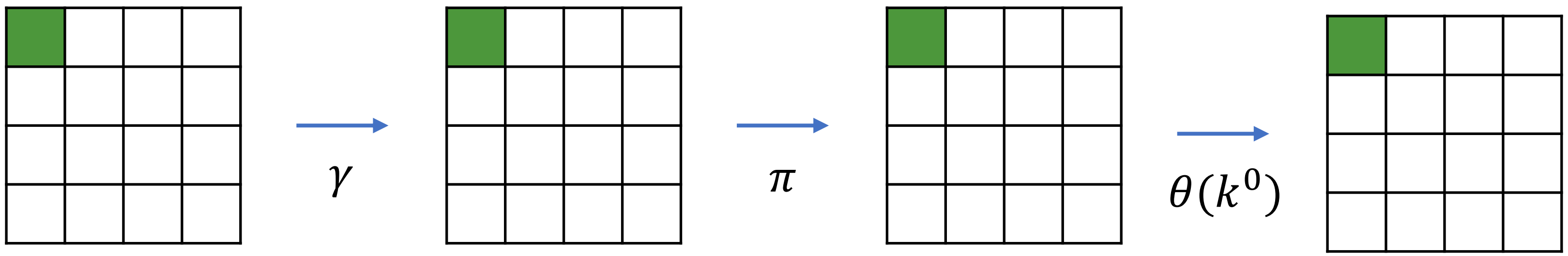
4 Round Attack :

Consider Λ-set with only one byte active. Trace the evolution of the position of the active bytes through 3 rounds :

Round 1 :

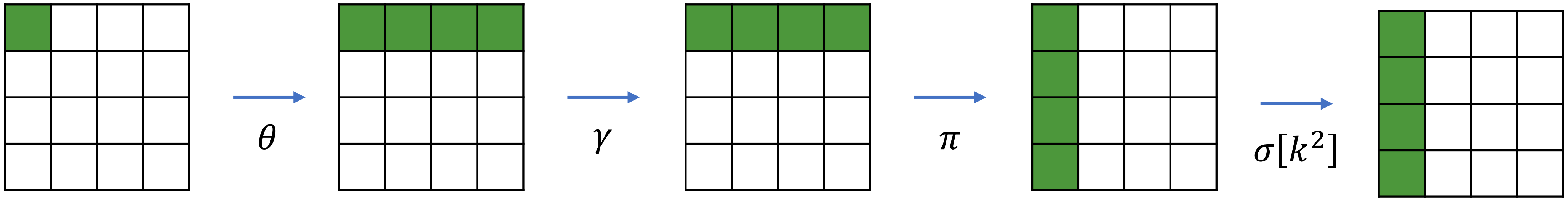
First round does not contain θ.

There is still only one byte active at the beginning of the 2nd round



4 Round Attack :

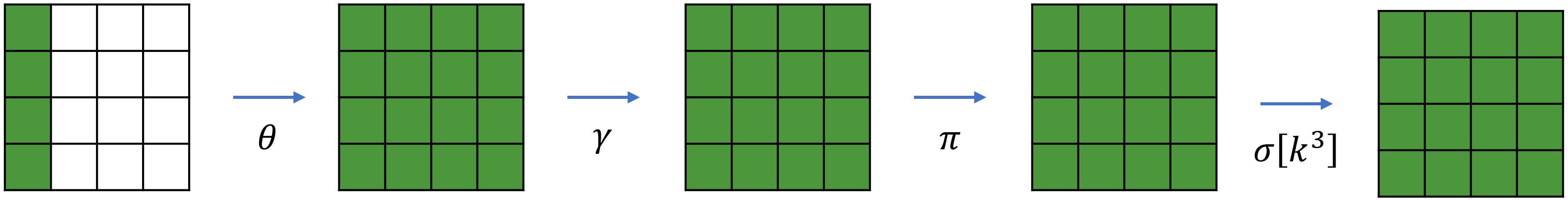
Round 2 :



2nd round converts this to a complete row of active bytes, that is subsequently transformed by π to a complete column.

4 Round Attack :

Round 3 :



θ of the 3rd round converts this to a Λ-set with only active bytes. This is still the case at the input to the 4th round

4 Round Attack :

Since the bytes of the outputs of the 3rd round (denoted by a) range over all possible values and are therefore balanced over the Λ -set, we have

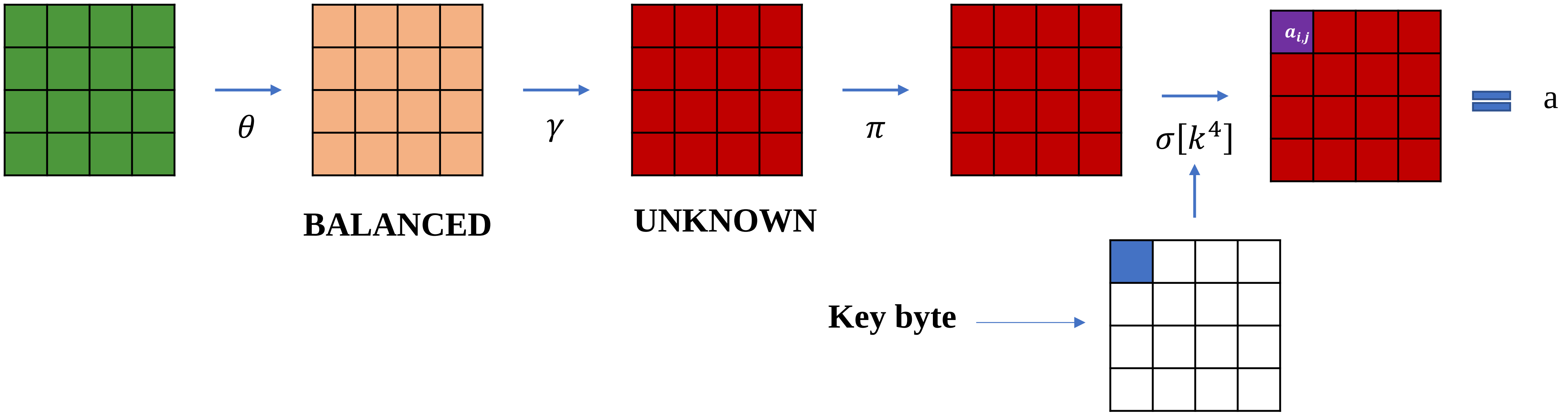
$$\bigoplus_{b=\theta(a),a\in\Lambda} b_{i,j} = \bigoplus_{a\in\Lambda}\bigoplus_{k} c_{j-k}a_{i,k} = \bigoplus_{l} c_l \bigoplus_{a\in\Lambda} a_{i,l+j} = \bigoplus_{l} c_l 0 = 0.$$

OR

$$\begin{aligned}\bigoplus_{b=\theta(a),a\in\Lambda} b_{i,j} &= \bigoplus_{a\in\Lambda}\left(2a_{i,j} \oplus 3a_{i+1,j} \oplus a_{i+2,j} \oplus a_{i+3,j}\right)\\ &= 2\bigoplus_{a\in\Lambda} a_{i,j} \oplus 3\bigoplus_{a\in\Lambda} a_{i+1,j} \oplus \bigoplus_{a\in\Lambda} a_{i+2,j} \oplus \bigoplus_{a\in\Lambda} a_{i+3,j}\\ &= 0 \oplus 0 \oplus 0 \oplus 0 = 0\end{aligned}$$

Hence, the bytes of the output of θ of the fourth round are balanced

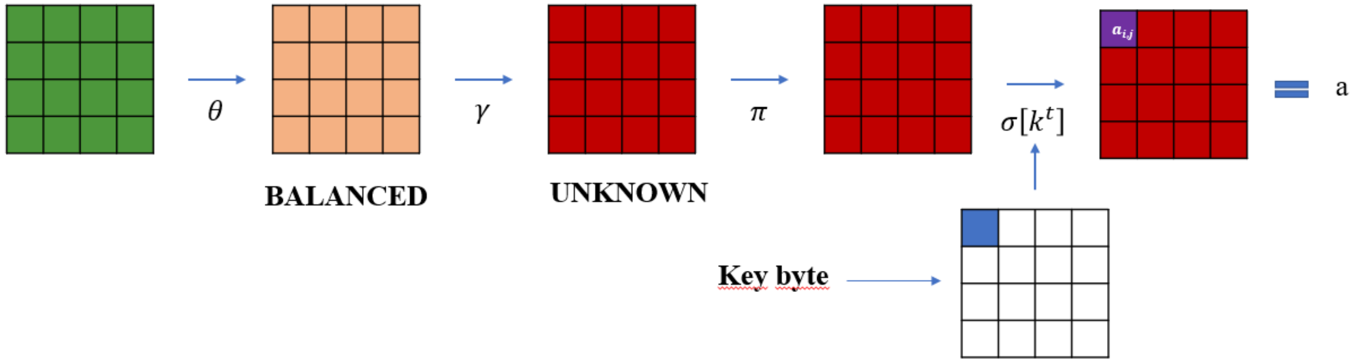
Round 4 :



An output byte of the 4th round (denoted by a here) can be expressed as a function of the intermediate state b

$$a_{i,j} = S_\gamma[b_{j,i}] \oplus k^4_{i,j}.$$

Round 4 :

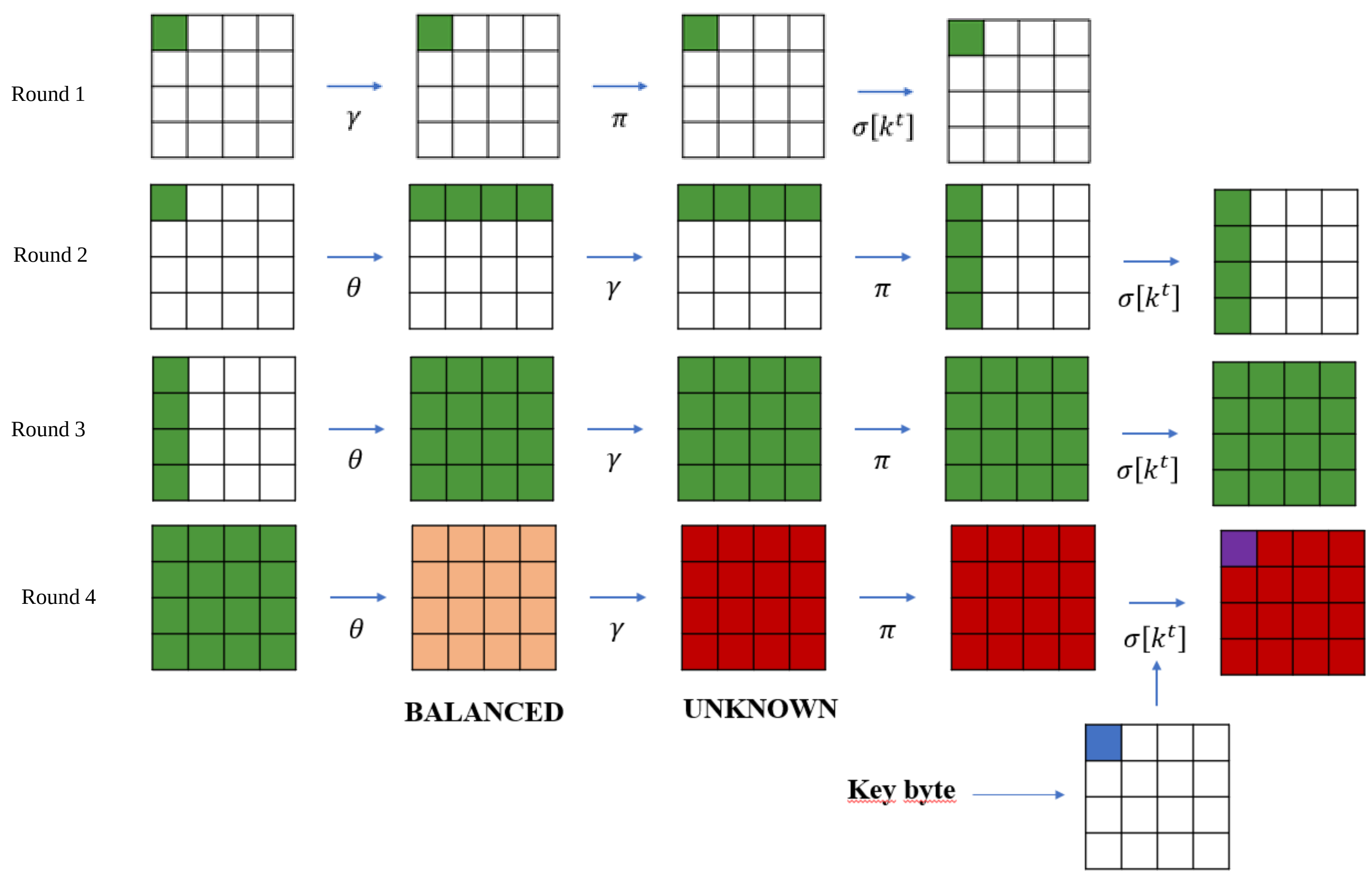


$$a_{i,j} = S_\gamma[b_{j,i}] \oplus k^4_{i,j}.$$

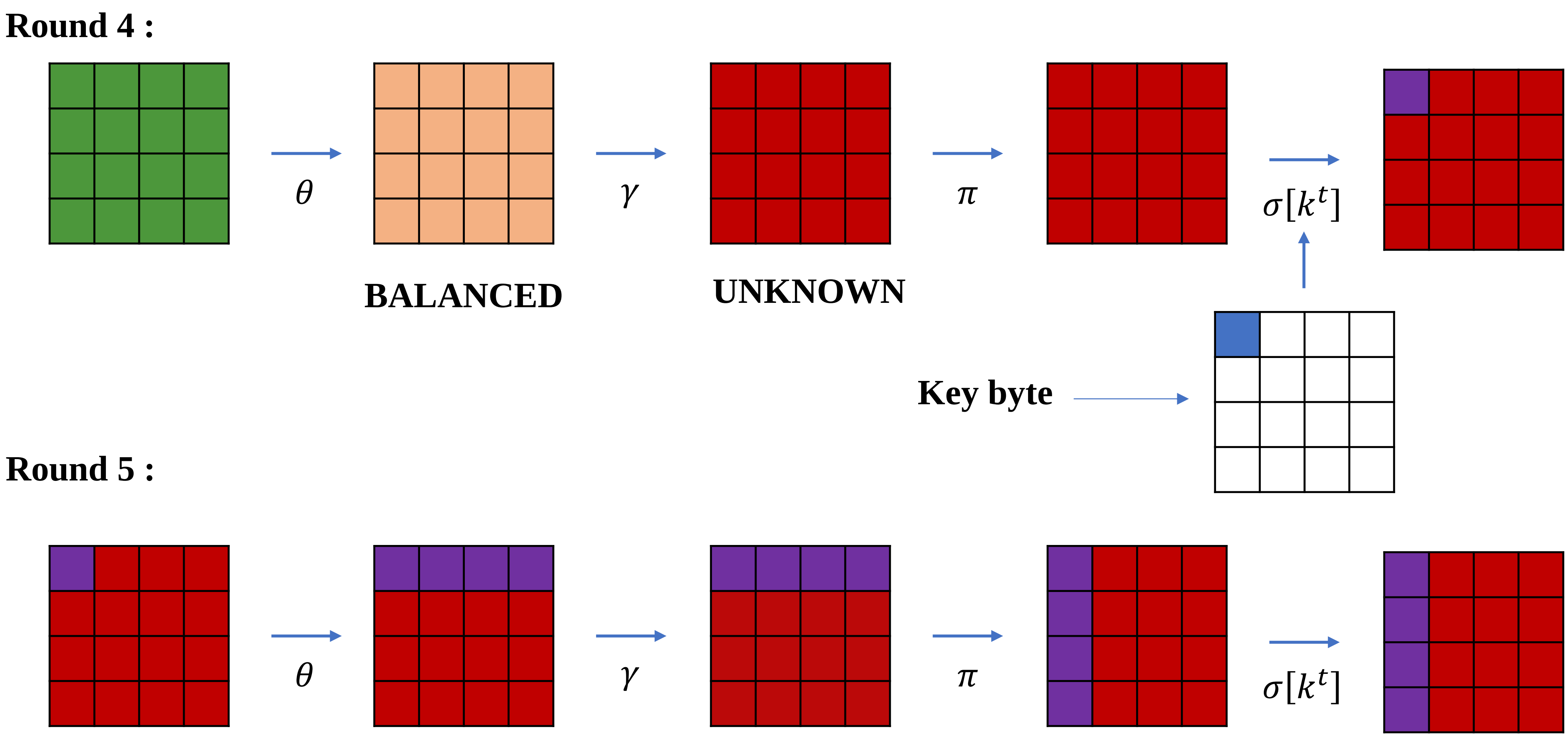
- By assuming a value for $k^4_{i,j}$ the value of $b_{j,i}$ for all elements of the Λ-set can be calculated from the ciphertexts.
- If the values of this byte are not balanced over Λ, the assumed value for the key byte was wrong.
- This is expected to eliminate all but approximately 1 key value. This can be repeated for the other bytes of k^4.
- Two $\Lambda - set$ of 256 chosen plaintexts each are sufficient to uniquely determine the cipher key with an overwhelming probability of success.

Dedicated Attack

$Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t]o\ \pi\ o\ \gamma\ o\ \theta$

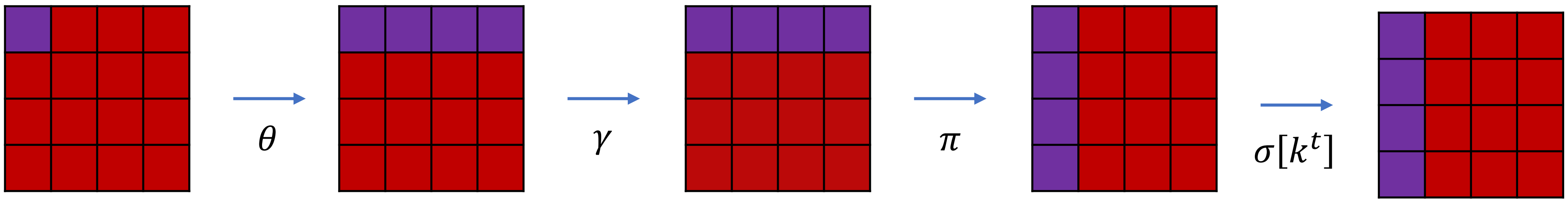


Extension by a round at the end:

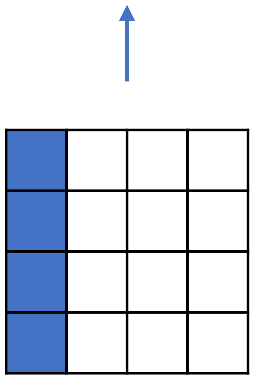


Extension by a round at the end:

Round 5 :



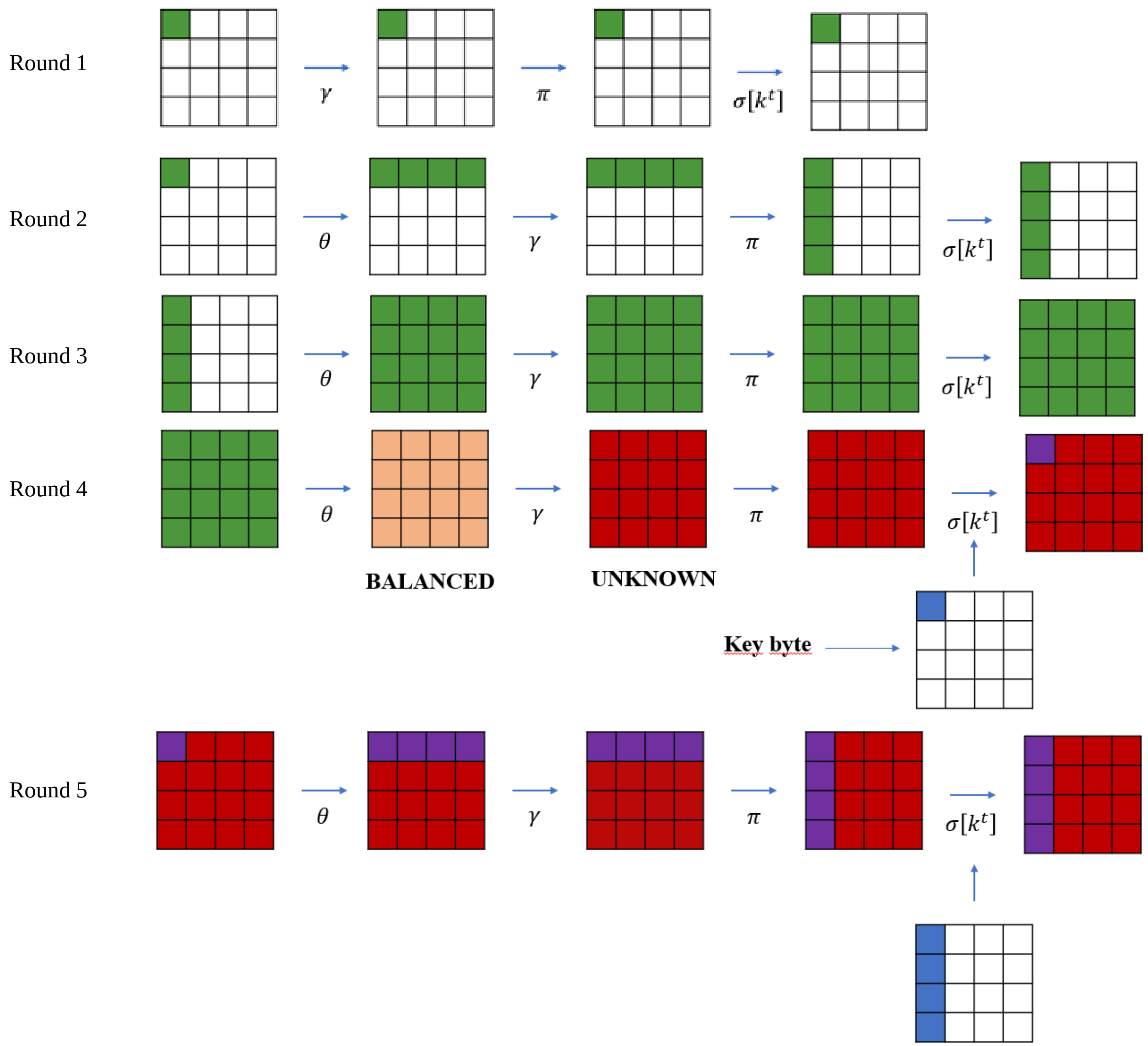
If an additional round is added, we have to calculate the above value of $b_{i,j}$ from the output of the 5th round instead of the 4th round. This can be done by additionally **assuming a value for a set of 4 bytes of the 5th round key.**



As in the case of the 4-round attack, wrong key assumptions are eliminated by verifying that $b_{i,j}$ is not balanced.

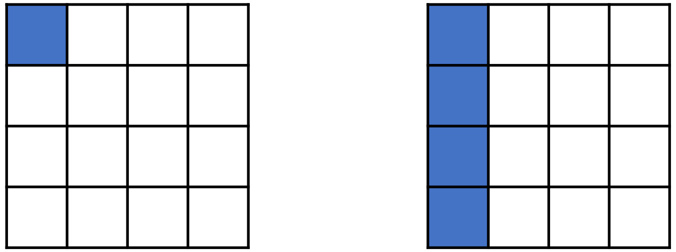
Dedicated Attack

$Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t] o\ \pi\ o\ \gamma\ o\ \theta$



Extension by a round at the end:

- We are guessing total $8 * 5 = 40$ bits.



- So in this 5-round attack 2^{40} key values must be checked.
- This process must be repeated 4 times.
- Since by checking a single Λ-set leaves only 1/256 of the wrong key assumptions as possible candidates, the cipher key can be found with overwhelming probability with only 5 Λ -sets.

Extension by a Round at the Beginning:

IDEA : Choose a set of plaintexts that results in a Λ-set at the output of the 2nd round with a single active byte

Intermediate state after $\boldsymbol{\theta}$ of the 2nd round has only a single active byte $\Rightarrow$ Output of the 2nd round has only a single active byte

This imposes the following conditions on a row of four input bytes of θ of the second round: $\Rightarrow$ One particular linear combination of these bytes must range over all 256 possible values (active) while 3 other particular linear combinations must be constant for all 256 states.

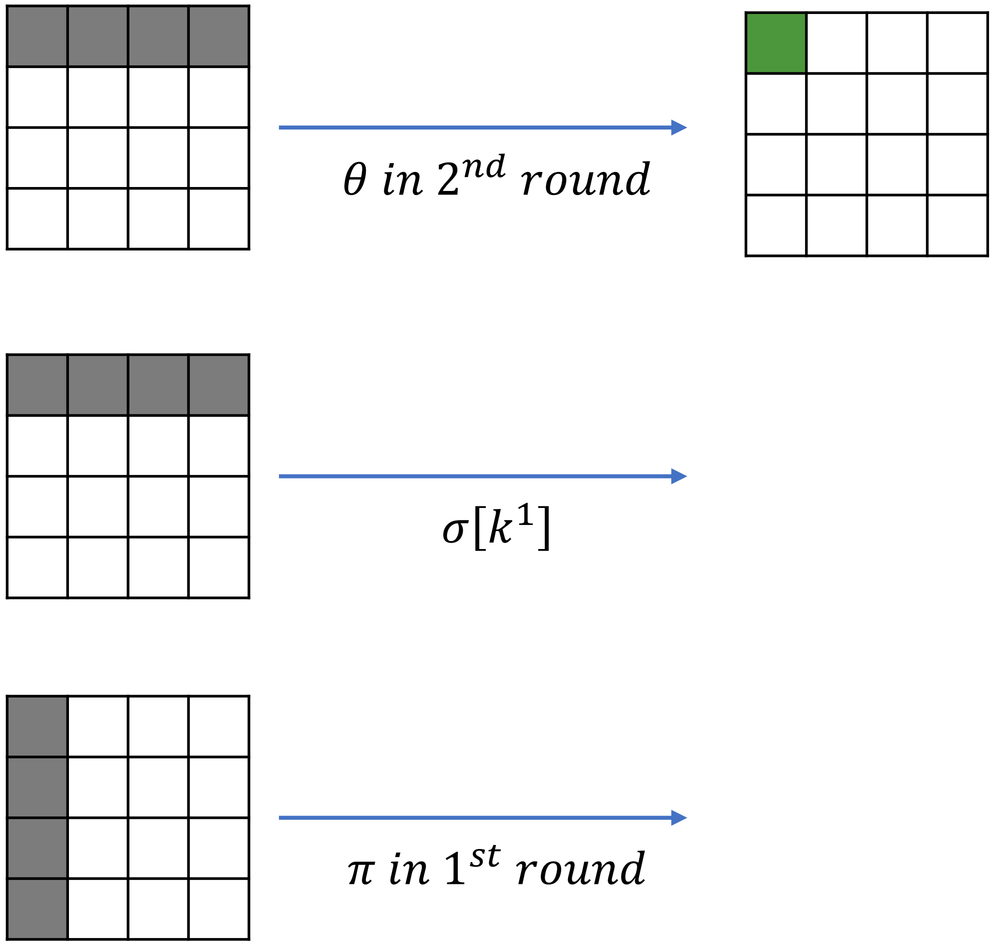
Dedicated Attack

$Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t]o\ \pi\ o\ \gamma\ o\ \theta$

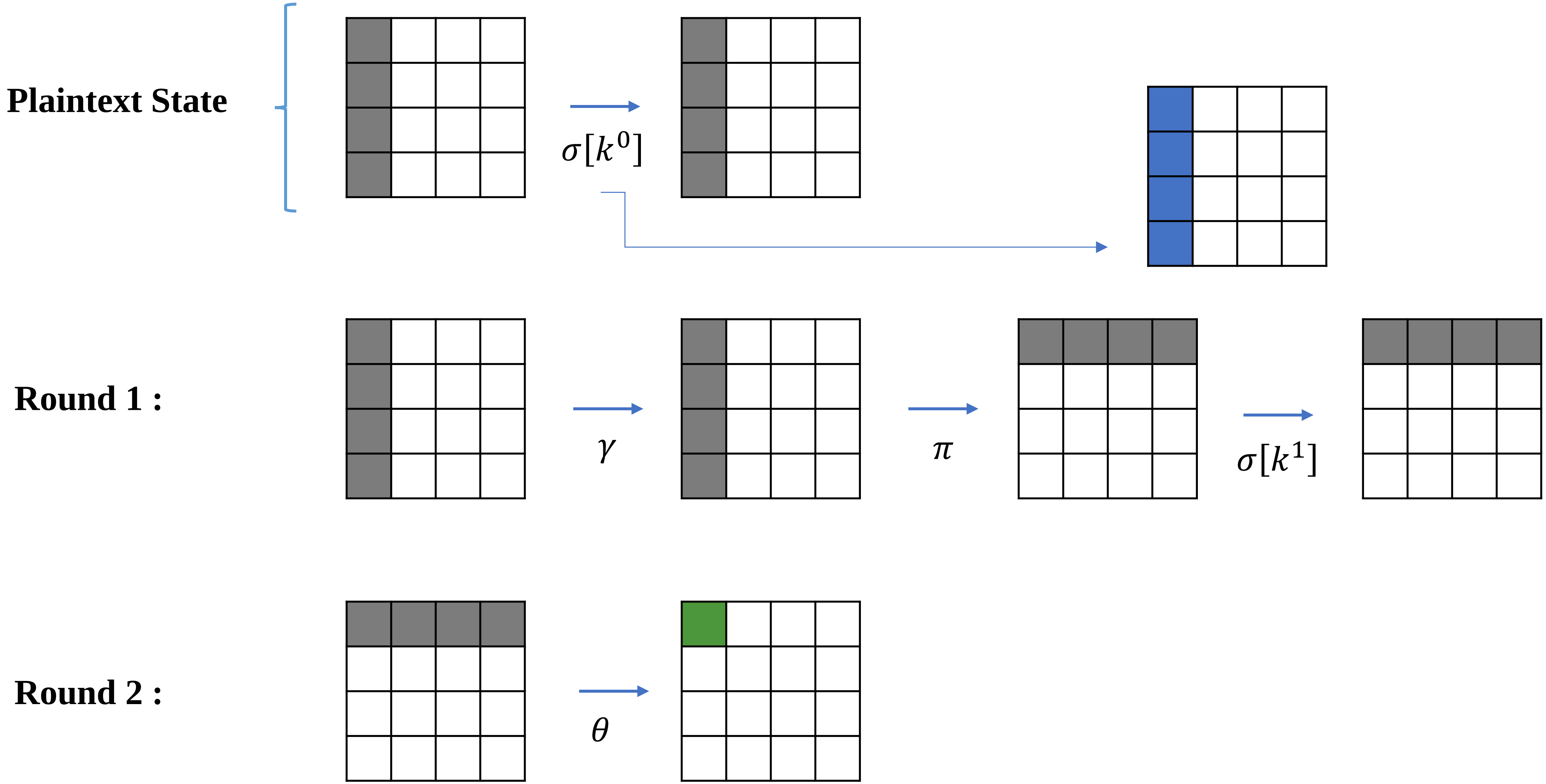
Extension by a Round at the Beginning:

This imposes identical conditions on the bytes in the same row in the input to $\sigma[k^1]$, and consequently on a column of bytes in the input to π of the 1st round.

If the corresponding column of bytes of k^0 is known, these conditions can be converted to conditions on four plaintext bytes.



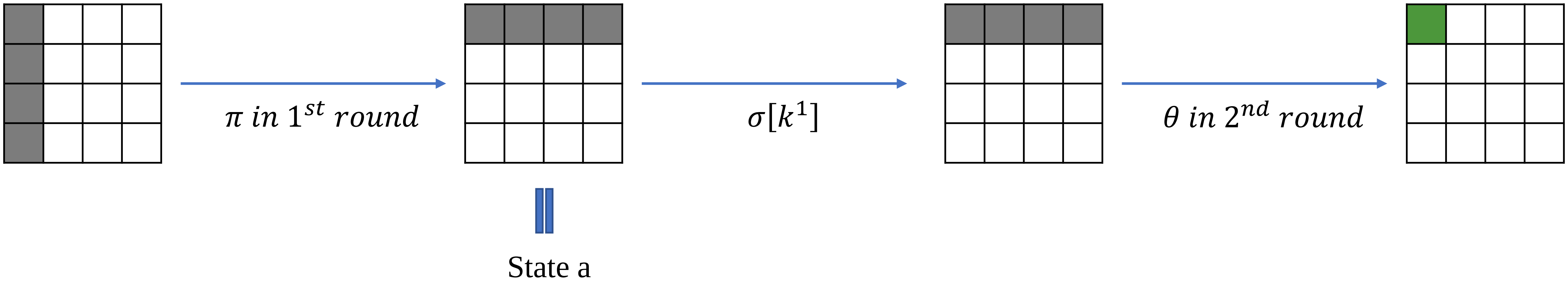
$Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t] \circ \pi \circ \gamma \circ \theta$



Dedicated Attack

$Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t]\, o\ \pi\ o\ \gamma\ o\ \theta$

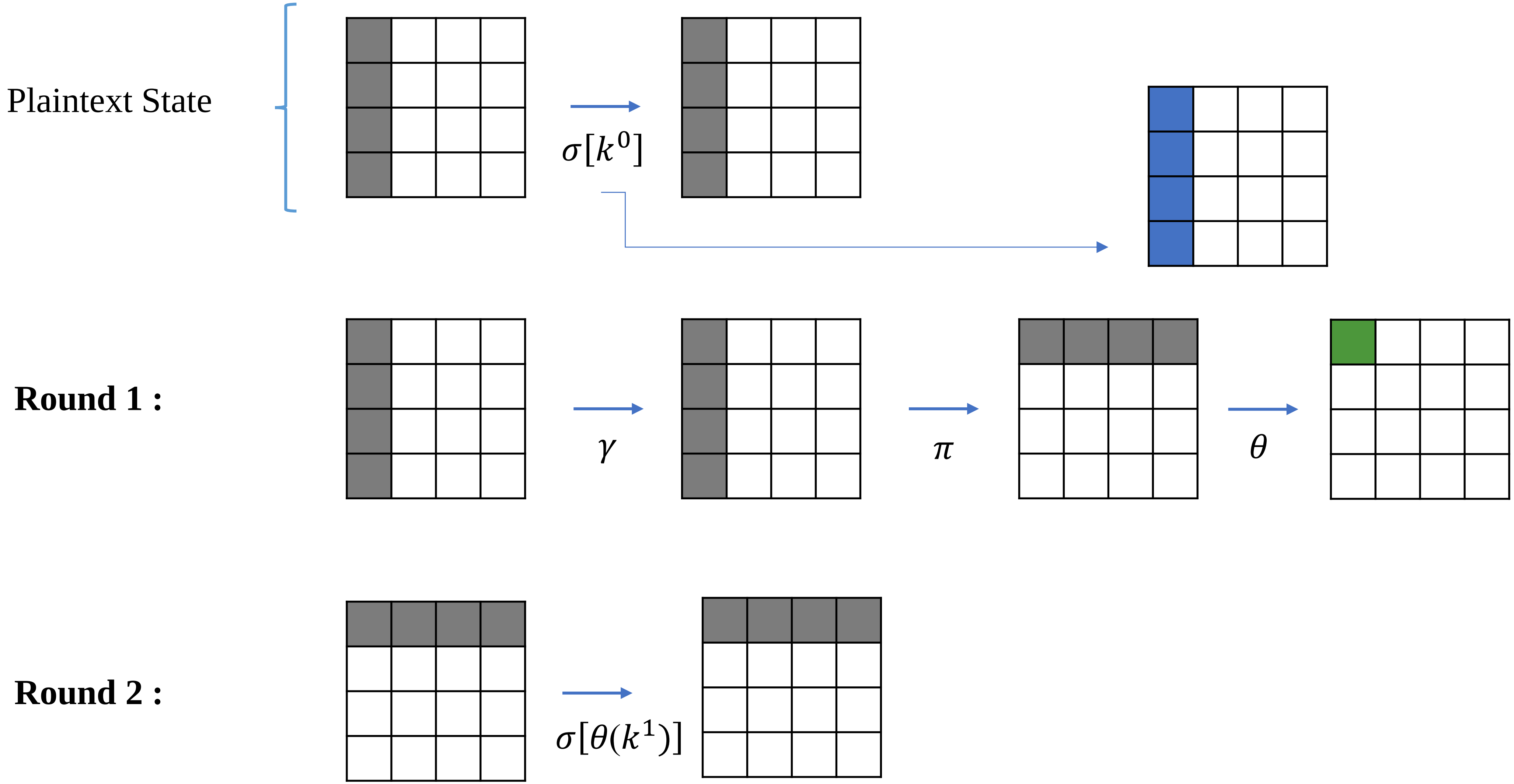
Extension by a Round at the Beginning:



$$\theta(k^1 \oplus a) = \theta(k^1) \oplus \theta(a) = \theta(a) \oplus \theta(k^1)$$

Dedicated Attack

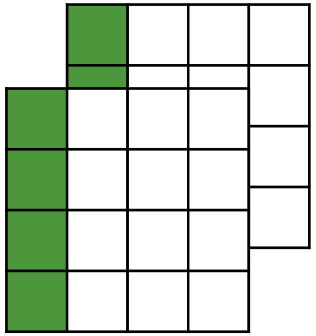
$\boldsymbol{Round\ of\ SQUARE = \rho[k^t] = \sigma[k^t] \circ \pi \circ \gamma \circ \theta}$



Extension by a Round at the Beginning:

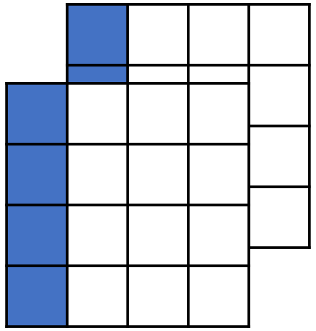
1) Now we consider a set of 2^{32} plaintexts, such that the array of bytes in one column ranges over all possible values and all other bytes are constant.

=



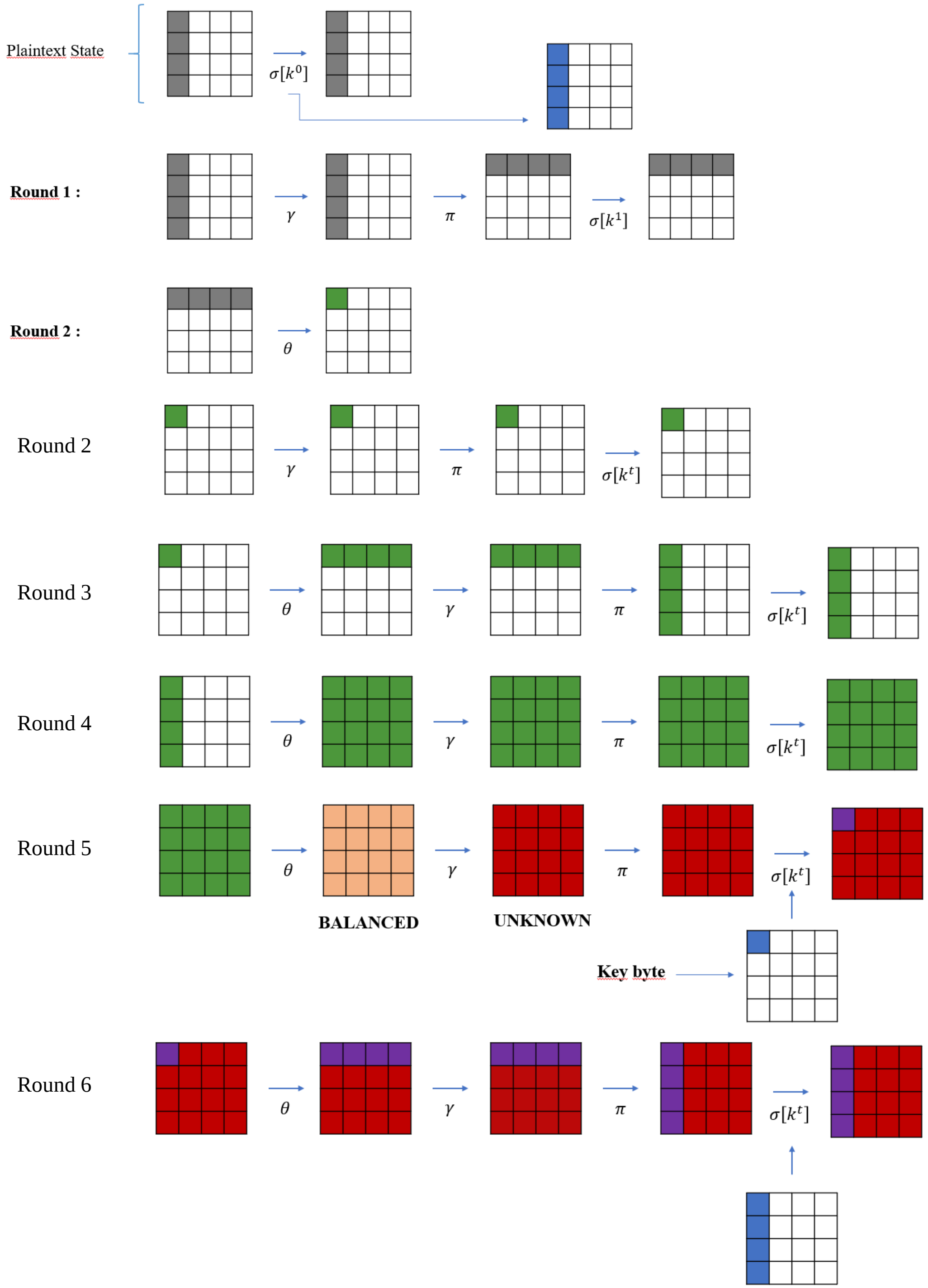
2) Make an assumption for the value of the 4 bytes of the relevant column of k^0

=



3) Select from the set of 2^{32} available plaintexts, a set of 256 plaintexts that obey the indicated conditions.

4) Now the 4-round attack can be performed.



Plaintext State
$\sigma[k^0]$
Round 1 :
γ
π
$\sigma[k^1]$
Round 2 :
θ
Round 2
γ
π
$\sigma[k^t]$
Round 3
θ
γ
π
$\sigma[k^t]$
Round 4
θ
γ
π
$\sigma[k^t]$
Round 5
θ
γ
π
$\sigma[k^t]$
BALANCED
UNKNOWN
Key byte
Round 6
θ
γ
π
$\sigma[k^t]$

Dedicated Attack

Complexity of the Attacks:

	Attack	#Plaintexts	Time	Memory
	4-round	2^9	2^9	small
Extension at end	5-round type 1	2^{11}	2^{40}	small
Extension at beginning	5-round type 2	2^{32}	2^{40}	2^{32}
Both Extensions	6-round	2^{32}	2^{72}	2^{32}

Table 3. Complexities of the attack on SQUARE.

References

1. Daeman, J., Knudsen, L.R., Rijmen, V.: The Block Cipher SQUARE. In: Biham, E. (ed.) FSE’97. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg, 1997.

2. Daemen, J., Rijmen, V.: AES proposal: Rijndael. National Institute of Standards and Technology (1998)