Quantum Key Distribution, cellular automata based large S-Box and almost key homomorphic block cipher for long term secret storage

Thomas Prévost, Bruno Martin, Olivier Alibart, Marc Kaplan, Anne Marin

Agenda

01
LONG TERM SECRET STORAGE

02
QUANTUM KEY DISTRIBUTION

03
MULTISS: LONG TERM STORAGE ACROSS MULTIPLE QKD NETWORKS

04
CELLULAR AUTOMATA BASED LARGE S-BOX

05
BLOCK-CIPHER WITH SECURITY LEVEL UPDATE

06
CONCLUSION

LONG TERM SECRET STORAGE

Issue and challenges

Why long term secret storage

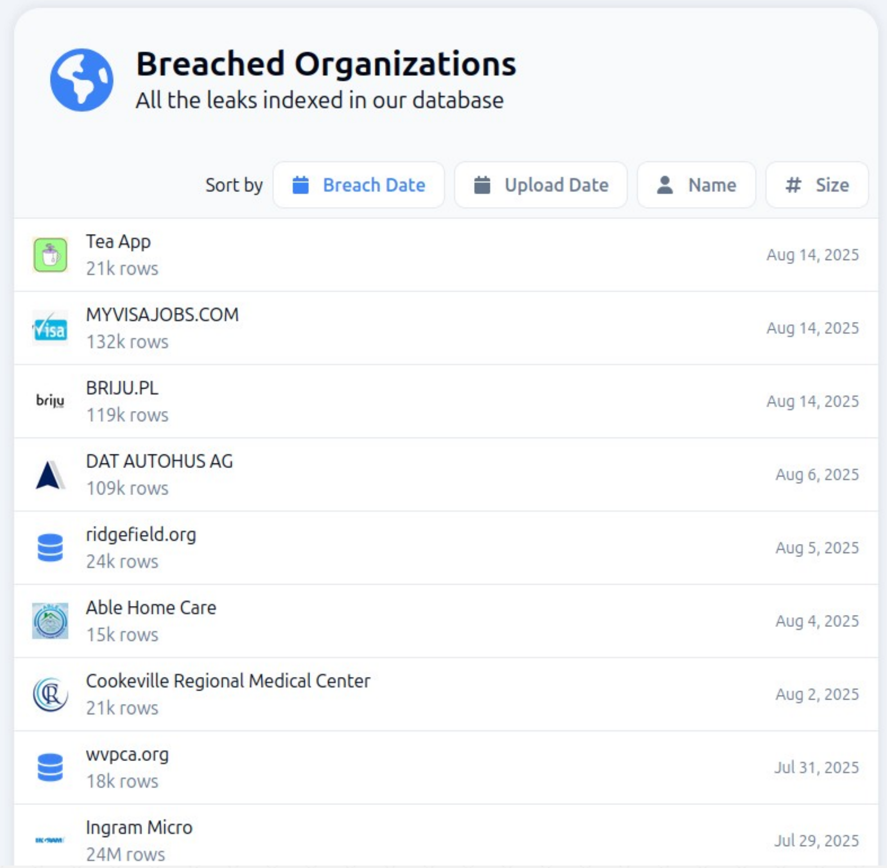
Some data could require confidentiality over decades

Medical data

Trade secrets

State secrets

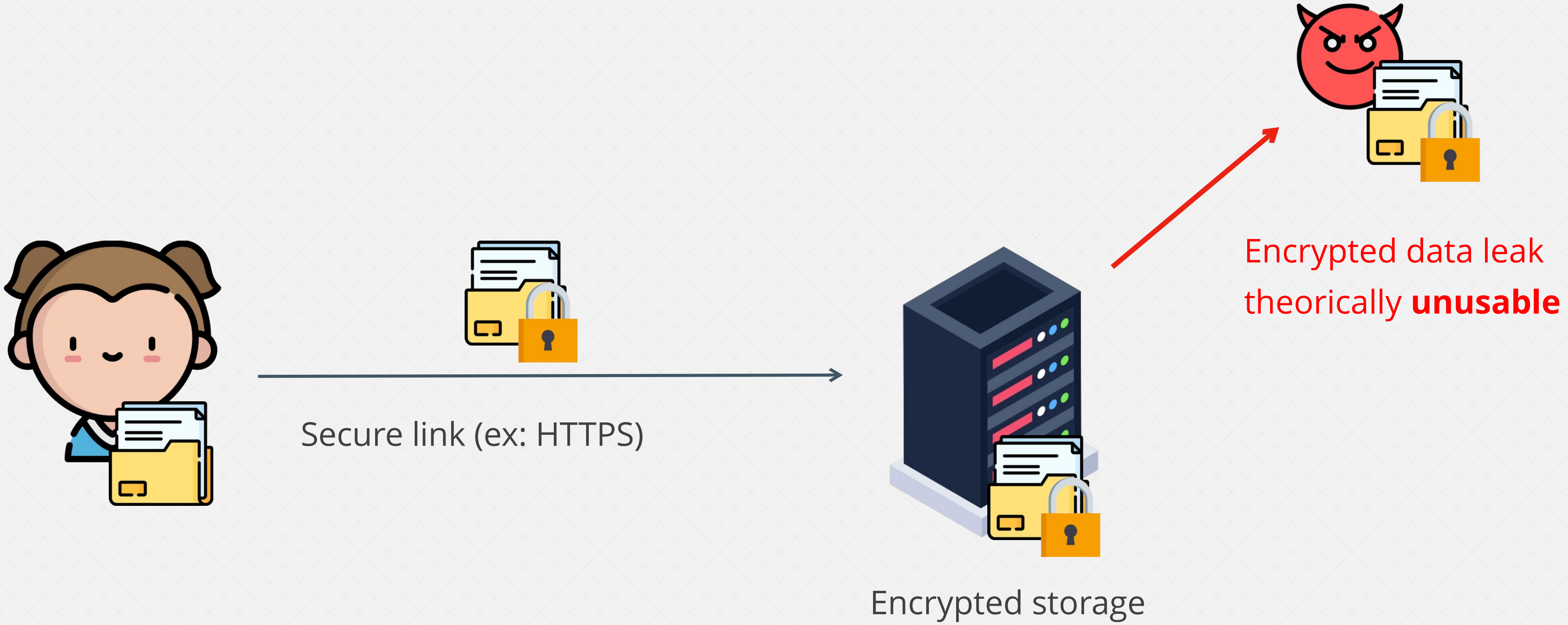
Issues with classical storage



Source: *https://databreach.com/breach*

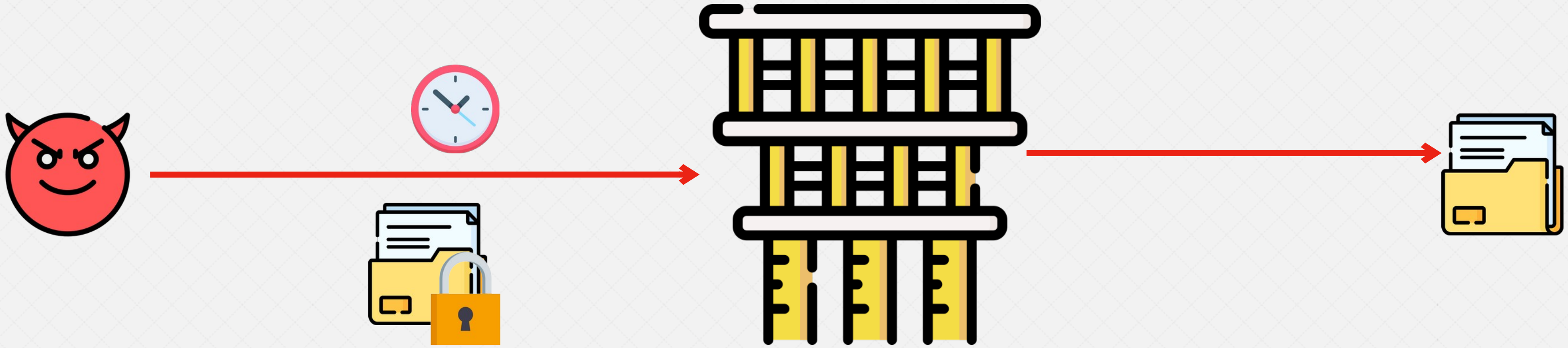
Source: *https://bonjourlafuite.eu.org/*

Solution: cryptography



Long term cryptography ?

Increased computing power, advances in cryptanalysis and the arrival of the quantum computer will allow an attacker to break current encryption



«*Harvest now, decrypt later*» attack

What about «post-quantum» cryptograhy?

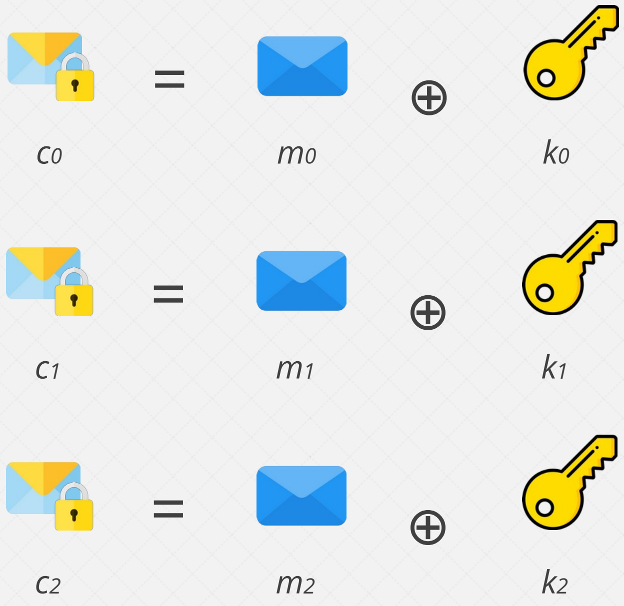
Mathemathical foundations are too recent for «long term security»

Solution: perfect secrecy

*(Loosely speaking, we will interchange the term "perfect secrecy" with "Information Theoretic Security" (**ITS**), although these two terms do not precisely designate the same thing)*

One Time Pad (OTP) : **random, non-reused** key, the **same size as the original message**

Unbreakable even in the long term («it is as hard to break the encryption as guessing the message by chance»)



Drawback: the key is hard to carry between the participants...

We should find a way to exchange the key securely

etc.

QUANTUM KEY DISTRIBUTION (QKD)

Possibilities and limitations

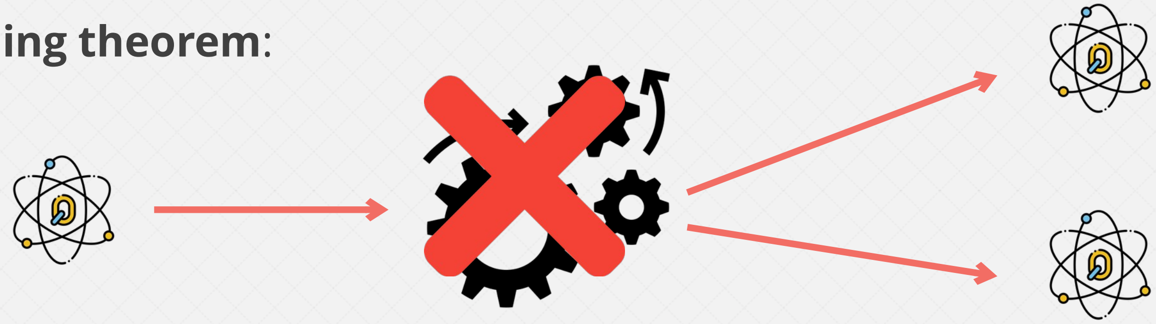
Uncertainty principle

A qubit (like a photon in an optical fiber) exists in a **superposition of states**:

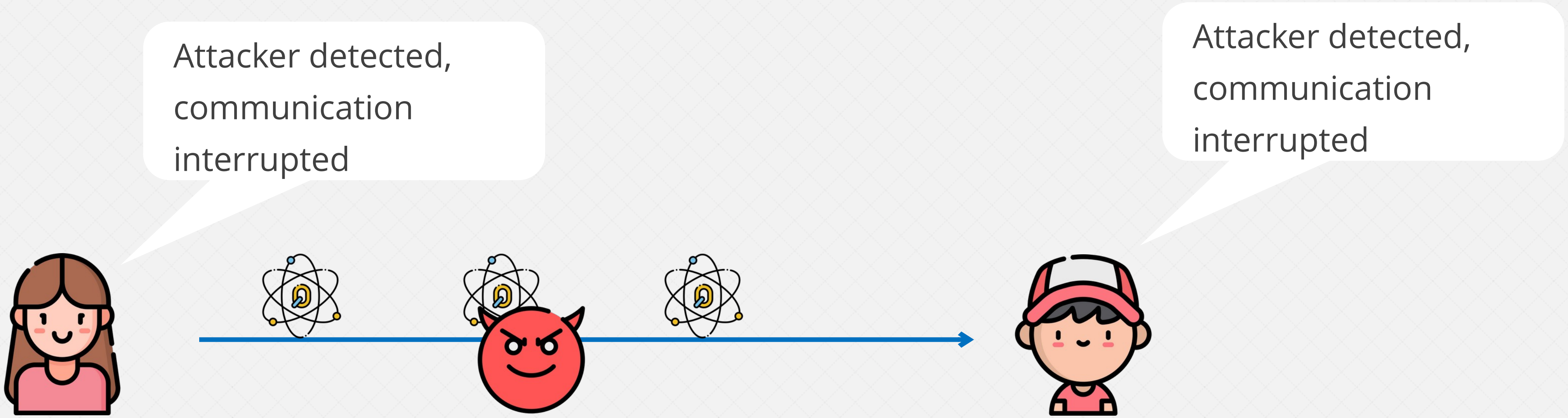
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Reading (ie measuring) the qubit collapse its state randomly:

Consequently: **no-cloning theorem**:



Principles of QKD



Advantages:

- The attacker has no information at all
- Theorically perfect security

Drawbacks:

- Costly hardware (for now)
- Limited geographical reach (few hundreds kms, as no-cloning theorem forbids to use optical amplifiers)

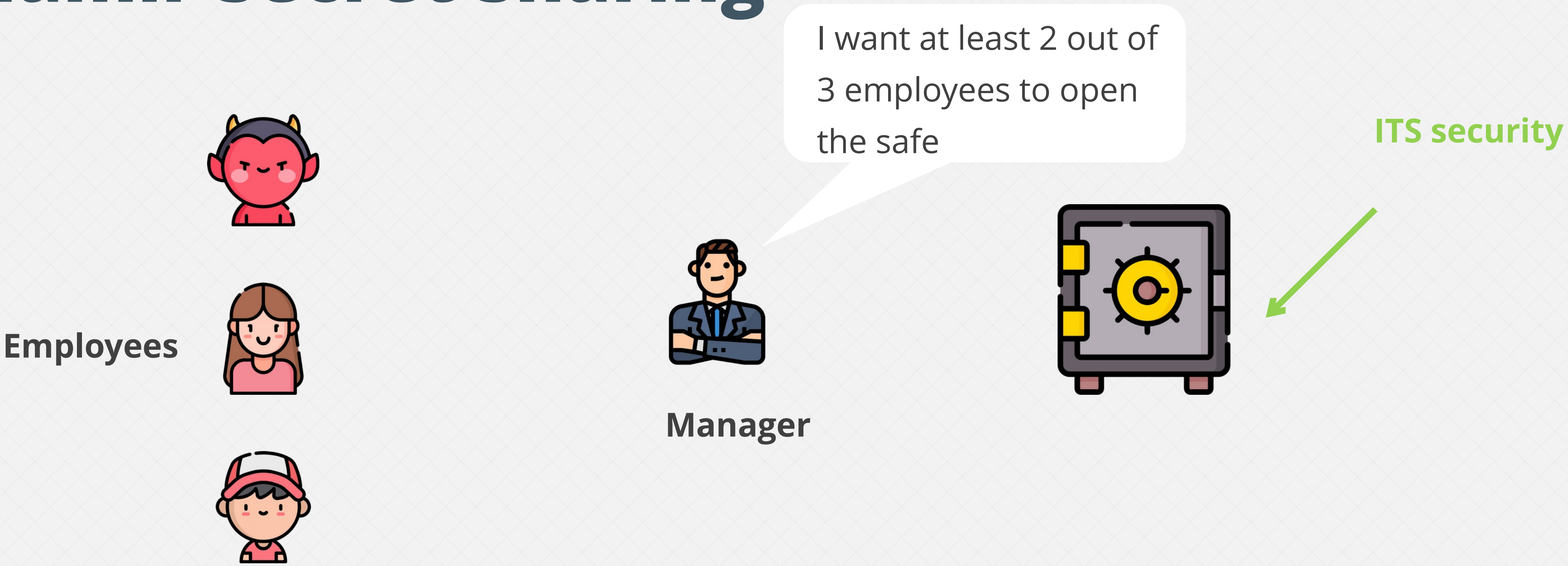
Progressive deployment of metropolitan QKD networks

MULTISS PROTOCOL

Long term secret storage across multiple QKD networks

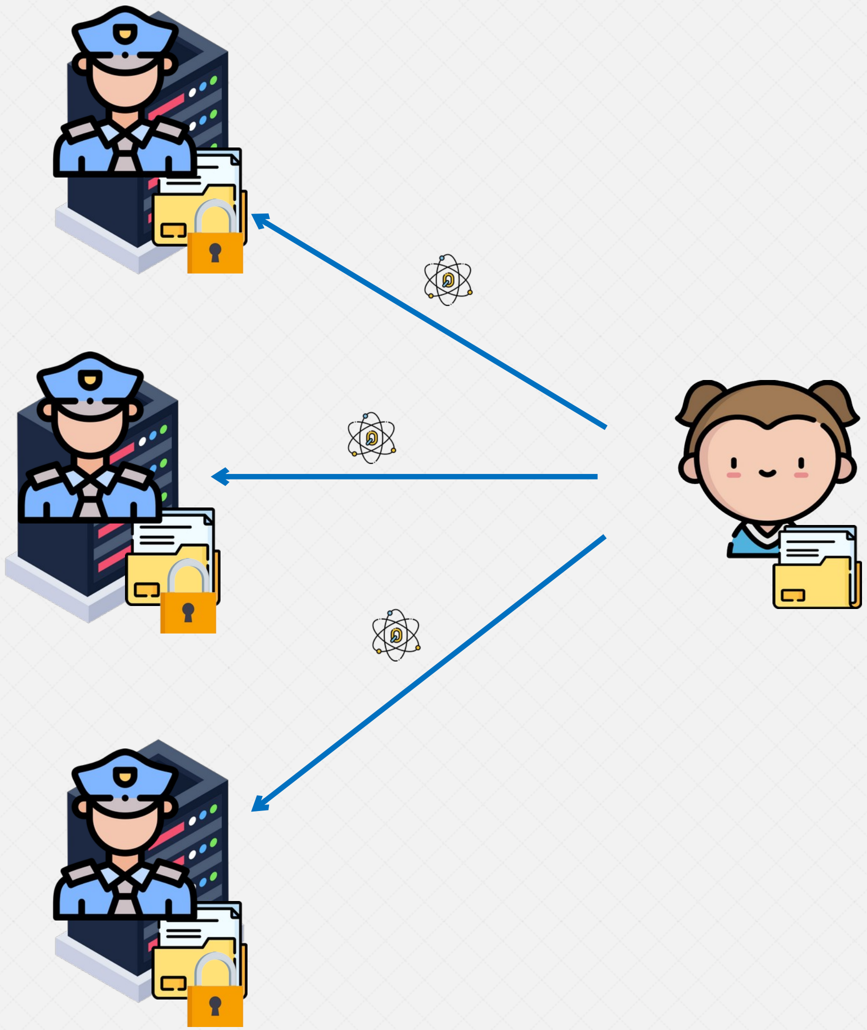
VeriQloud

Shamir secret sharing



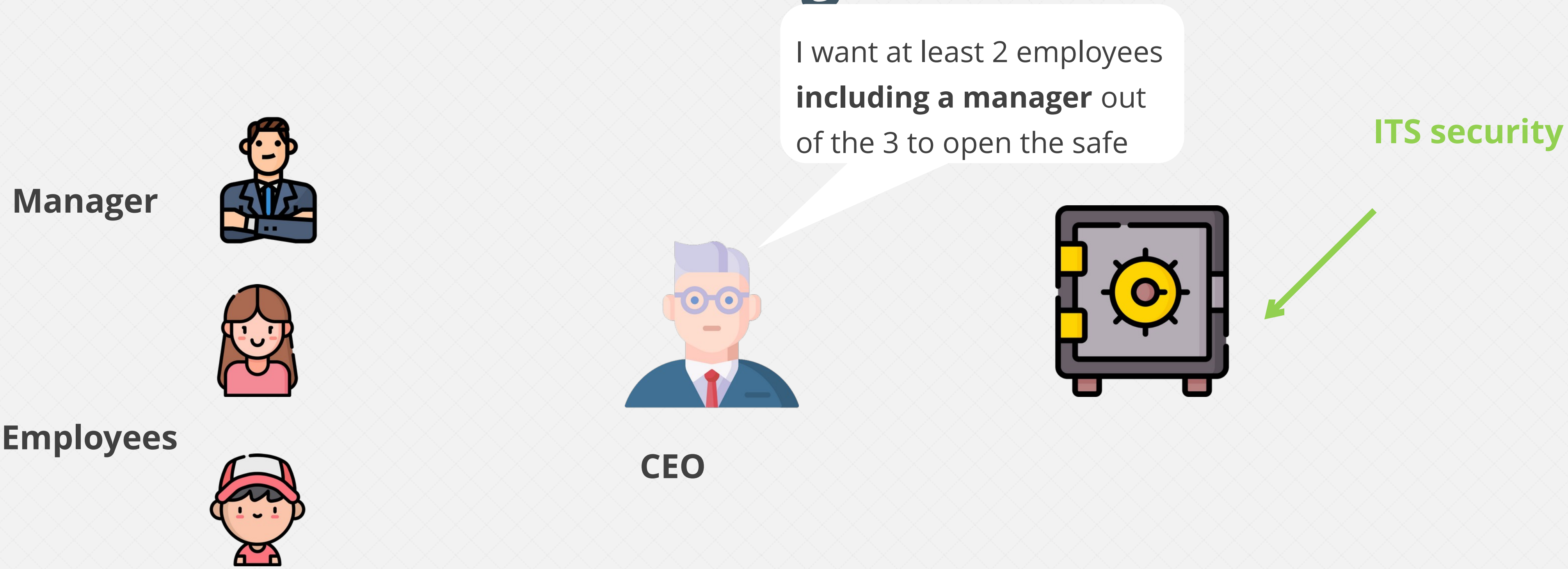
- Generate random polynomial $P \in F_q[x]$, such that deg(P) = threshold - 1 and $P(0)$ = secret
- Distribute to participants $P(1)$ mod q, $P(2)$ mod q ... , q being prime > secret

LINCOS protocol



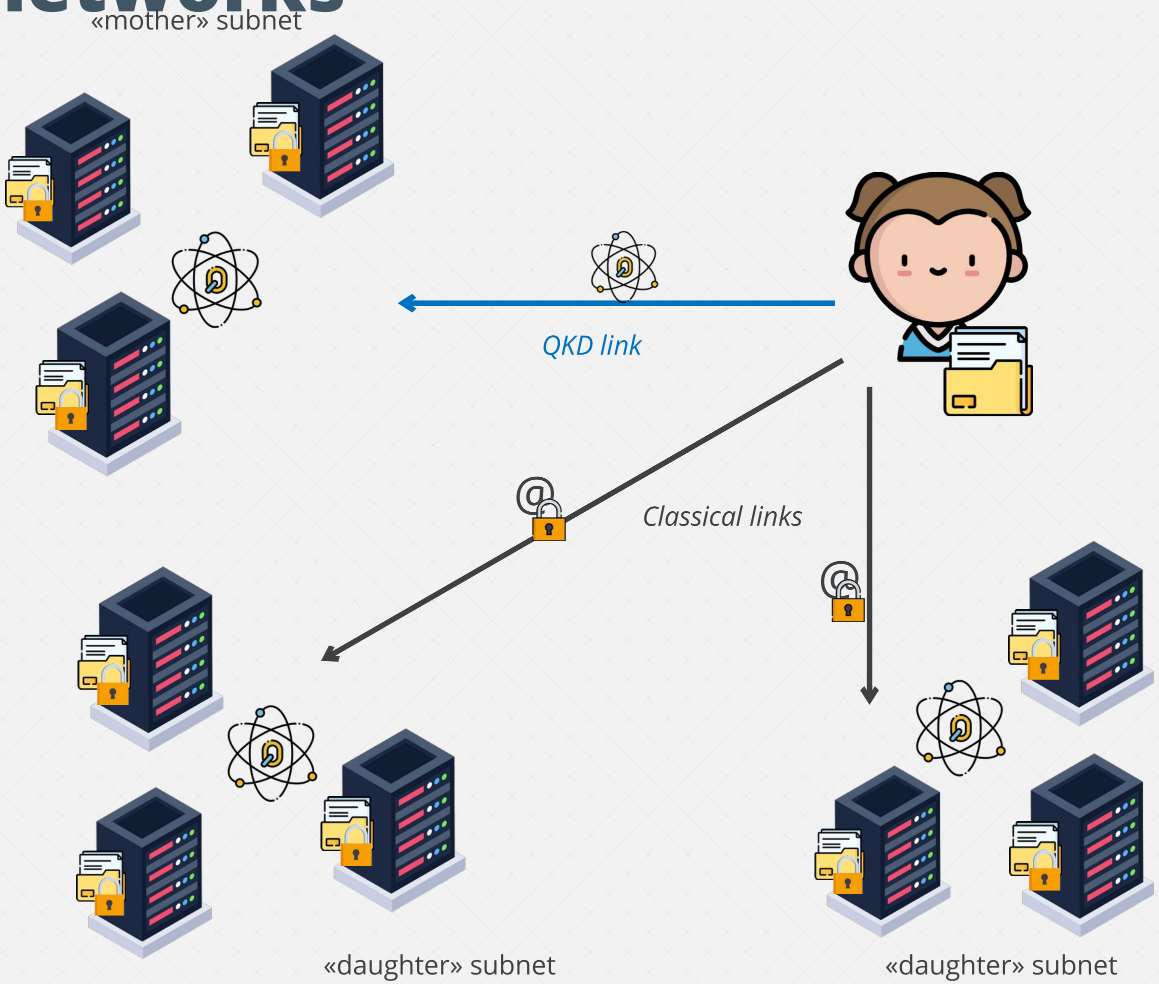
- Existence of a quantum link between the owner of the document and the nodes of the metropolitan network
- Document sharing between the different nodes using the Shamir primitive (perfect theoretical security)
- Vulnerable if the entire metropolitan network is compromised

Hierarchical secret sharing



- The managers are given evaluations *P(1)* mod *q*, *P(2)* mod *q* ...
- The employees are given evaluations of derivative polynomial *P'(1)* mod *q*, *P'(2)* mod *q* ...

MULTISS: Confidential secret storage across multiple QKD networks



- Secret distribution by hierarchical secret sharing (Birkhoff interpolation)
- Primitive shares on «mother subnet», derived on «daughter subnets»

- Retains security properties strictly greater than those of LINCOS
- Ensures security against an adversary who manages to take control of a QKD network

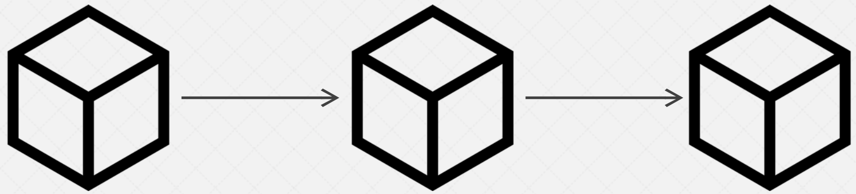
- Currently experimenting between Nice and Paris QKD networks

CELLULAR AUTOMATA BASED LARGE S-BOX

And comparison with AES S-Box

Block cipher encryption

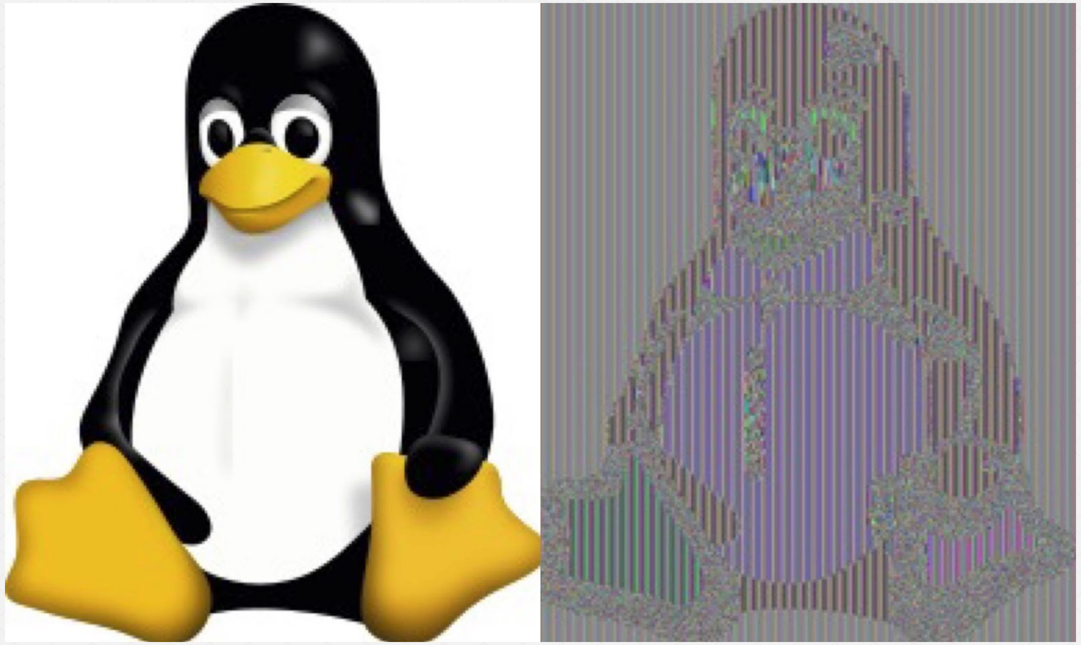
- Commonly used symmetric encryption
- Slicing the message into equal sized blocks



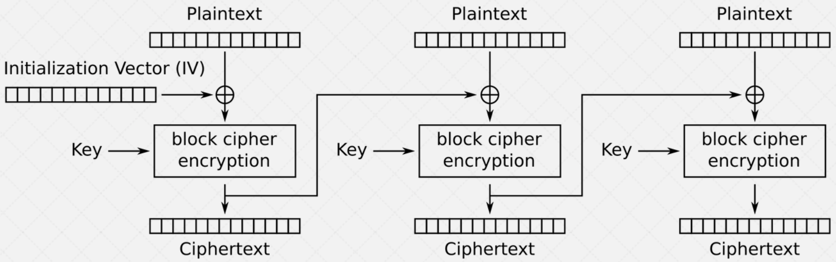
Example: **A**dvanced **E**ncryption **S**tandard (AES),
NIST standardized algorithm for symmetric cryptograohy

Blocks interdependecy

If each block was encrypted independently:

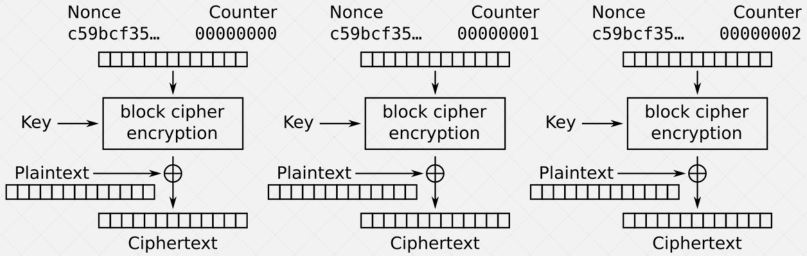


Solution 1: block chaining (CBC): not parallelisable



Cipher Block Chaining (CBC) mode encryption

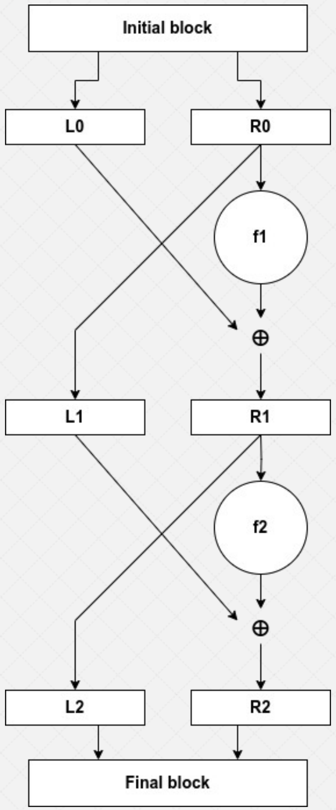
Solution 2: use a counter (GCM, CTR...)



Counter (CTR) mode encryption

Illustration of block encryption structure: Feistel networks

Used in some block cipher algorithms, like Blowfish
(AES uses another similar construction)



With:

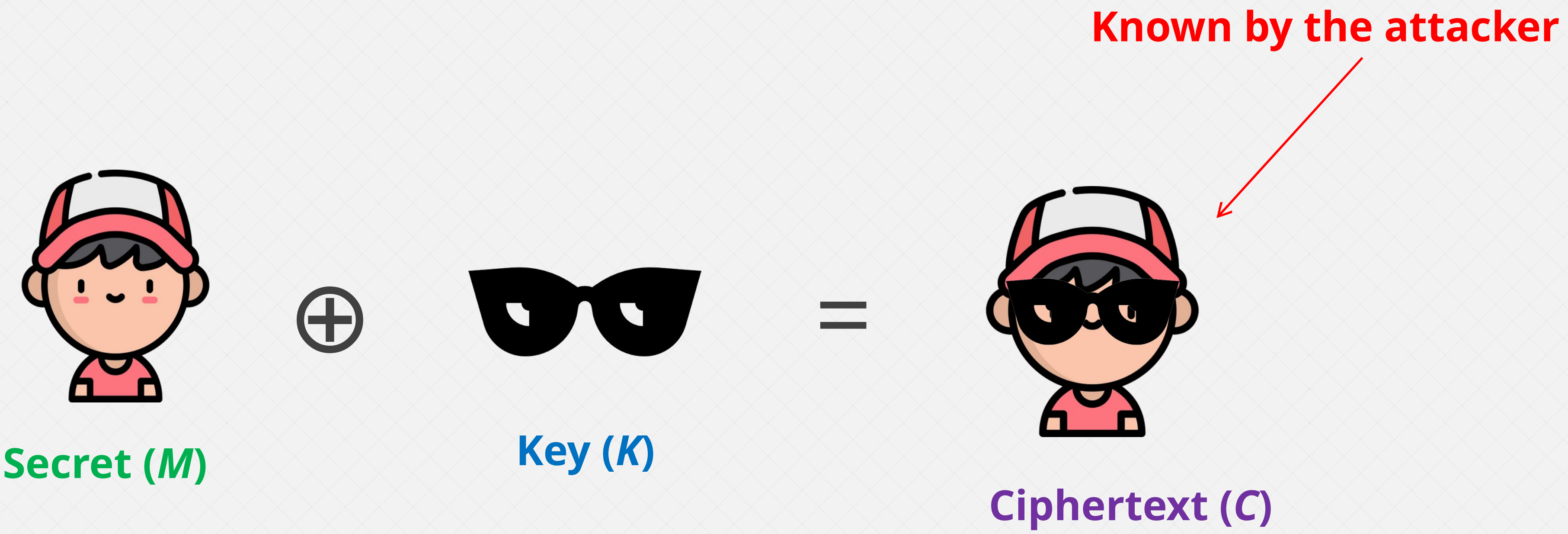
- f_1 and f_2: pseudo-random permutations
- ⊕ XOR operator (exclusive OR)
- Feistel network depth = 2

«pseudo-random» permutation:
Permutation that indistinguishable from a truly random permutation by a «*polynomial time adversary*» (an adversary with a computer with limited computing power)

But what are the subpermutations (f_1, f_2) made of?

Why do we need S-Boxes?

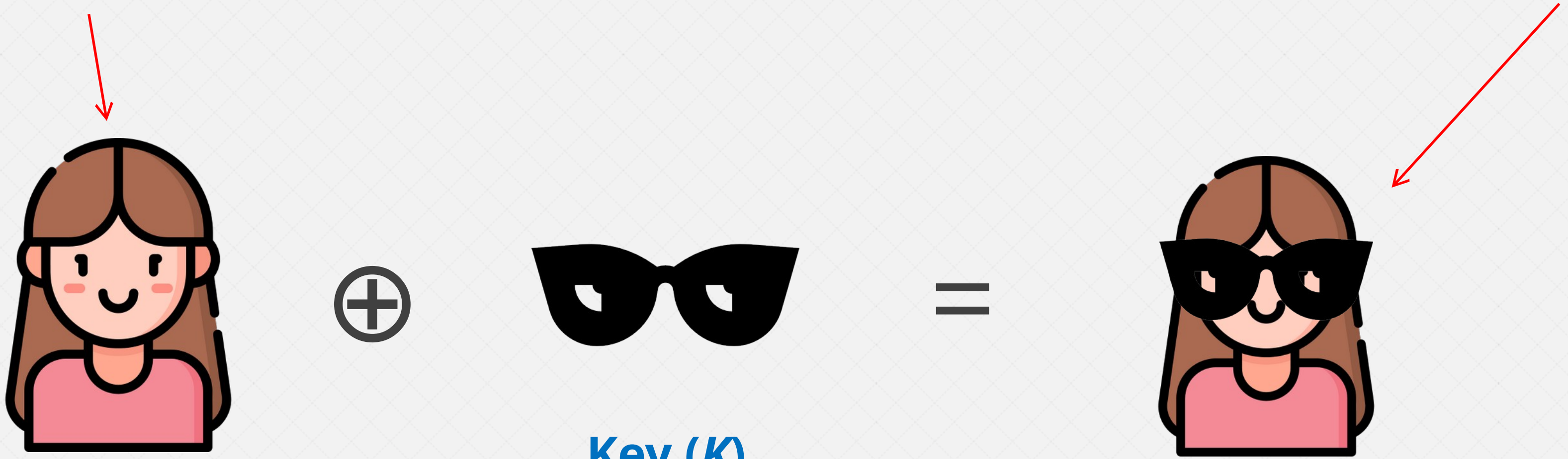
If block cipher was linear:



Why do we need S-Boxes?

If block cipher was linear:

Known by the attacker **Known by the attacker**



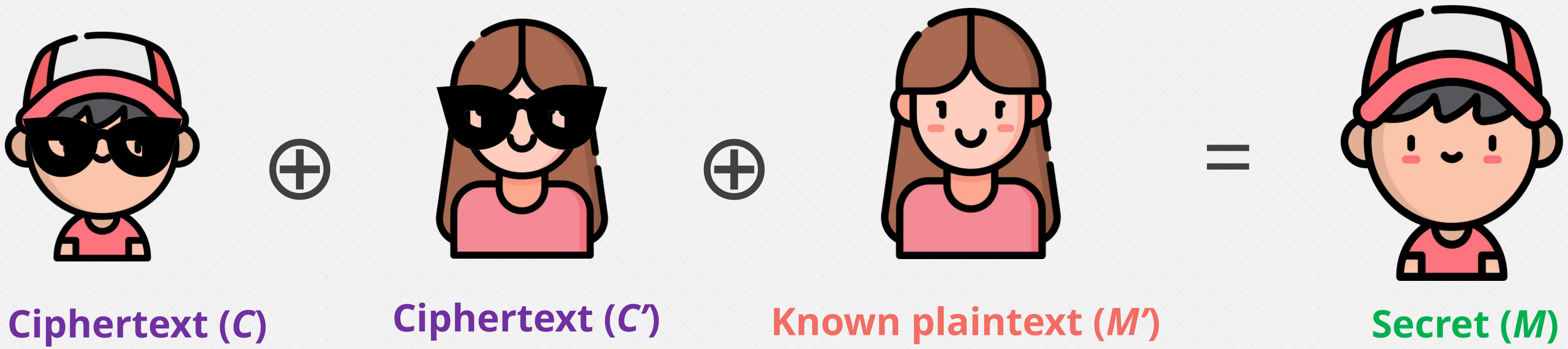
Key (K)

Known plaintext (M') **Ciphertext (C')**

Example of known plaintext: home page of bank website, before filling your credentials

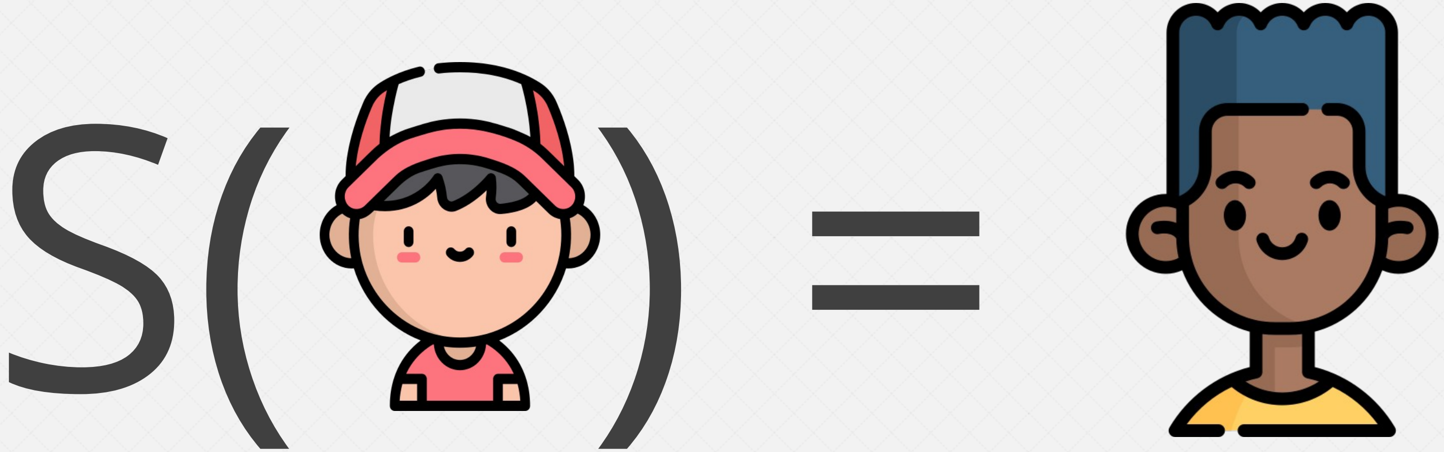
Why do we need S-Boxes?

If block cipher was linear:



This is a **known plaintext attack**

S-Box principle



So a simplified subpermutation round is **the S-Box action combined with a linear operation with the key**

A S-Box is a **public substitution table** that must be as far as possible from a finear function.
As we will see, there are other expected mathematical properties

S-Box example: PRESENT

x	0	1	2	3	4	4	6	7
S(x)	12	5	6	11	9	0	10	13

x	8	9	10	11	12	13	14	15
S(x)	3	14	15	8	4	7	1	2

A S-Box is a **public bijective* function $B^n \rightarrow B^n$ that is as far as possible from a linear function**

**There are non-bijective S-Boxes but this is not what we need here*

Boolean functions

$f(x_1, x_2, \ldots, x_n) = y$, with $x_1, x_2, \ldots, x_n, y \in \boldsymbol{B}$

Algebraic Normal Form (ANF):

$y = x_1*x_2*x_0 \oplus x_2*x_4 \oplus x_5 \oplus 1$

Here deg(f) = 3: size of the largest monomial

Linear function:

if degree = 1 ou degree = 0 (constant function)

There are $2^{\wedge(2^{\wedge}n)}$ possible n-variable Boolean functions

S-Box component functions

For $S(x_1, x_2, \ldots, x_n) = y_1, y_2, \ldots, y_n$, with $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n \in \mathbf{B}$

There are 2^n-1 component Boolean functions of S-Box S:

- $f_1(x_1, x_2, \ldots, x_n) = y_1$
- $f_2(x_1, x_2, \ldots, x_n) = y_2$
- …
- $f_{n+1}(x_1, x_2, \ldots, x_n) = y_1 \oplus y_2$
- …
- $f_{2^\wedge n\text{-}1+1}(x_1, x_2, \ldots, x_n) = y_1 \oplus y_2 \oplus \ldots \oplus y_n$

S-Box component functions

Example:

For *S* defined as:

x	00	01	10	11
$S(x)$	10	00	11	01

We have:

x	$f_1(x) = y_1$
00	1
01	0
10	1
11	0

x	$f_2(x) = y_2$
00	0
01	0
10	1
11	1

x	$f_2(x) = y_1 \oplus y_2$
00	1
01	0
10	0
11	1

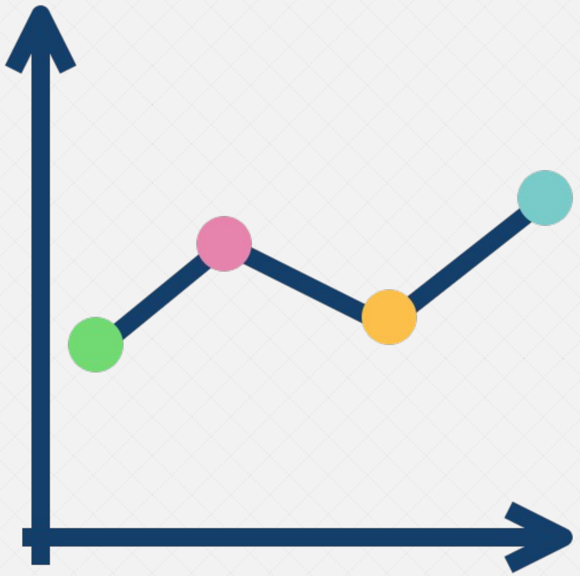
S-Box Mathematical properties

Exhaustive list:

- Min and max algebraic degree
- Algebraic complexity
- Nonlinearity
- Strict Avalanche Criterion (SAC)
- Bit Independence Criterion (BIC)
- Linear Approximation Probability (LAP)
- Differential Approximation Probability (DAP)
- Differential Uniformity (DU)
- Boomerang Uniformity (BU)

Nonlinearity

- For each component function, number of bits that should be switched to have a linear function
- The worst value is the metric



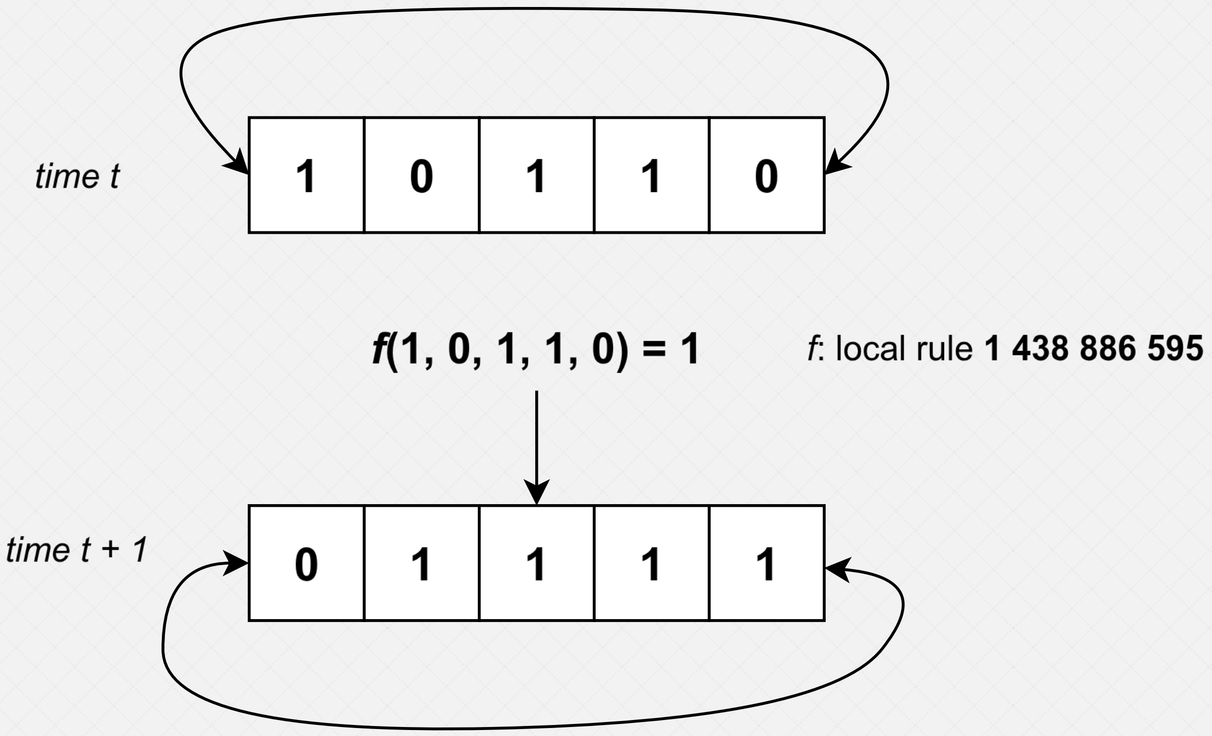
- A **high value enables linear cryptanalysis resistance**

Bit Independence Criterion

- BIC is satisfied when for all input bit *k*, for all output bits *i*, *j*, flipping k^{th} input bit flips i^{th} and j^{th} output bits independently

- The metric is a number between 0 and 1 (closest to satisfy the BIC), **1 the worst and 0 the best**

Uniform cellular automaton



- Ring* of Boolean cells
- At each **discrete** time step, each cell is updated according to its value and the values of its neighbors, according to a weel chosen **local transition function**

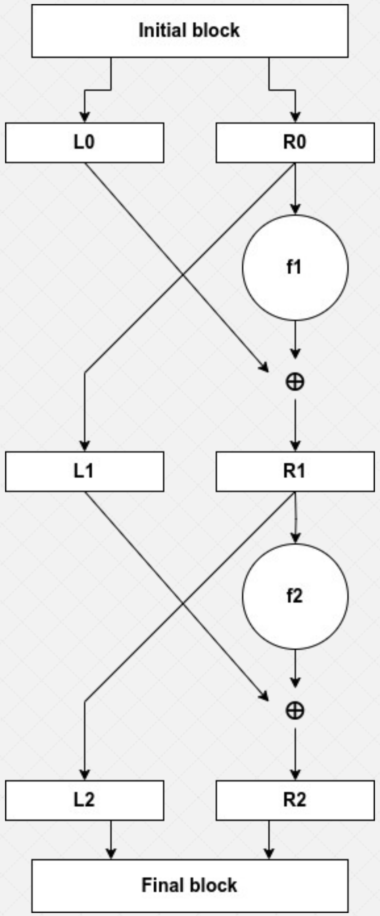
In this specific case

With $f(x) = x_0{*}x_3 \oplus x_1{*}x_3 \oplus x_1 \oplus x_2{*}x_3 \oplus x_2 \oplus x_3{*}x_4 \oplus x_3 \oplus 1$

1 438 886 595 is the **decimal representation** of the truth table

Construction of our 10-bit S-Box

Our S-Box itself is a **sub 10-bit Feistel network**, of depth 11



Empirical construction based on cryptanalysis:

- f_1: affine function: f(x) = 5x+3 mod 31
- f_2 to f_5: 1 generation of our automaton
- f_6: affine function: f(x) = 7x+11 mod 31
- f_7 to f_9: 1 generation of our automaton
- f_{10}: affine function: f(x) = 13x+17 mod 31
- f_{11}: 1 generation of our automaton

Results

Comparison with AES S-Box *(values are normalized to compare a 10-bit S-Box with a 8-bit S-Box)*

Property	Our 10-bit S-Box	8-bit AES S-Box
Min algebraic degree	**8**	7
Max algebraic degree	**9**	7
Algebraic complexity	**1023**	255
Nonlinearity	**434 (= 108.5 * 4)**	112
Strict Avalanche Criterion	**0.44 - 0.5 - 0.57**	0.45 - 0.5 - 0.56

Results

Comparison with AES S-Box *(values are normalized to compare a 10-bit S-Box with a 8-bit S-Box)*

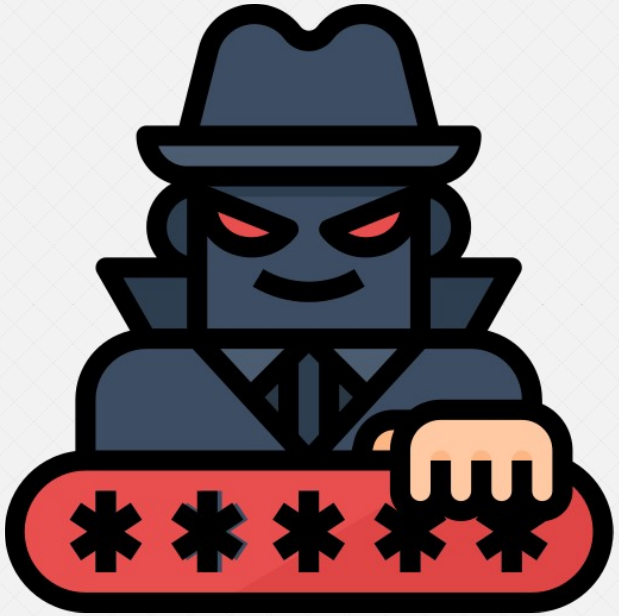
Property	Our 10-bit S-Box	8-bit AES S-Box
Bit Independence Citerion	**0.124**	0.134
Linear Approximation Probability	**9.28%**	6.25%
Differential Approximation Probability	**1.37%**	1.56%
Differential Uniformity	**14 (= 3.5 * 4)**	4
Boomerang Uniformity	**24 (= 6 * 4)**	6

ALMOST KEY-HOMOMORPHIC BLOCK CIPHER

Key rotation and security level update

Security levels

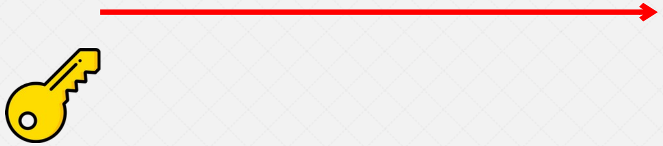
- Indication of the number of operations required to break the cipher (using bruteforce attack)
- 256 bits of security level: $O(2^{256}) \approx 10^{177}$ operations needed to break the cipher
- While 80 bits of security were considered «enough» in the 2000's, the standard today is 256 bits
- For a given ciphertext, we may want to update security level «on-the-fly»...



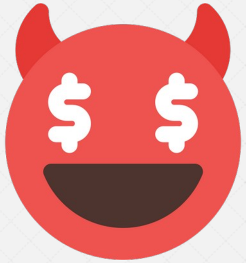
Key rotation ?

Top secret encrypted document

Fired employee



Encryption key

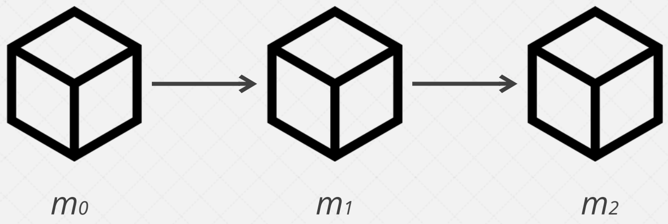


Competitor

Mandatory for a company that manages credit card numbers: **PCI-DSS standard** (Payment Card Industry Data Security Standard)

«Almost Key-Homomorphic» symmetric block cipher

Slicing of message m into equal sized blocks m_0, m_1, m_2 ...



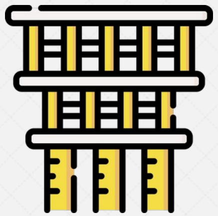
For each block, the cipher c_i is given by:

$$c_i = m_i + a_i\, k + e$$

With a_i **public** random polynomial vector, k **secret key**, e **secret** random error (with small coefficients)

(a_i, k, $e \in \mathbb{Z}q[x] / (x^n + 1)$, q prime)

Supposedly quantum-proof...



Simple key rotation

Knowledge of token Δ is **useless for the adversary**

Token Δ

=

New key k_2

-

Old key k_1

security_level(k_1) = security_level(k_2)

For each block, the new cipher c_i' is given by:

$$c_i' = c_i + a_i \Delta + e'$$

With a_i random **public** vector, Δ **jeton**, e' new **secret** random error *(needed only for indistinguishability between old and new ciphertext)*

We can then decrypt using new key k_2: the ciphertext as been **updated without decryption**

On-the-fly security level update

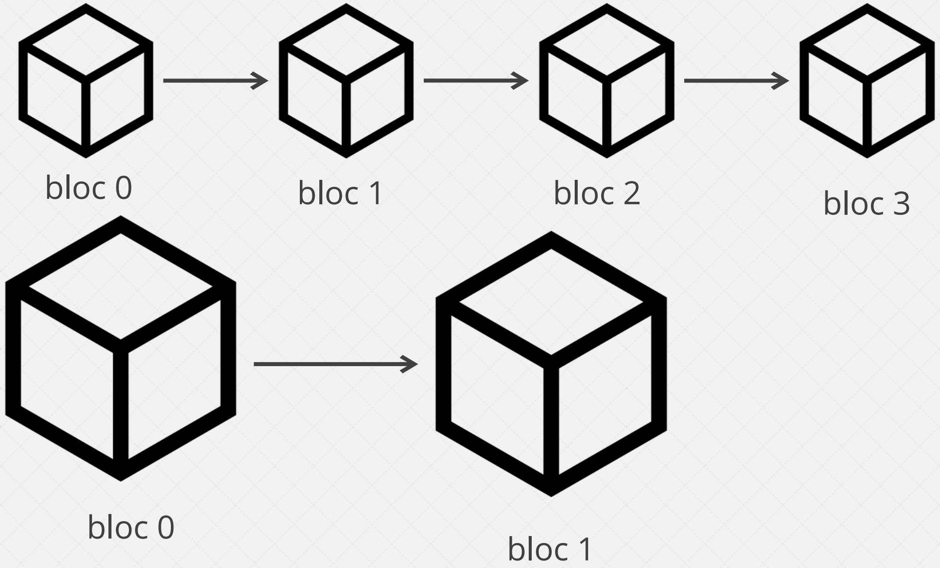
Old block encrypted with key k_1 (security level = n)

k_1: n bits of security

$$c_i = m_i + k_1\, \boldsymbol{a}^{i+1} + e$$

Neighbor blocks merging:

$$c_i' = c_{2i} + X^n\, c_{2i+1}$$



Key rotation toward new key k_2 (security level = $2n$)

k_2: $2n$ bits of security

CONCLUSION

Conclusion

QKD

Ensure perfect secrecy on limited geographical reach

MULTISS

Long term confidential secret storage across multiple remote QKD networks

Strong 10-bit S-Box

Gains time against actual cryptanalysis capabilities

Upgradable security level

Increases ciphertext security against evolving attacker's computing power

THANK YOU

Questions ?

Min and max algebraic degree

Size of the largest monomial of each function:

- If $f1(x_1, x_2, \ldots, x_n) = x_1 * x_2 * x_4 \oplus x_1 * x_2 \oplus x_3$ then $\deg(f1) = 3$
- Largest and lowest degree of each component function

Large values avoid «Low order approximation attack»

Strict avalanche criterion

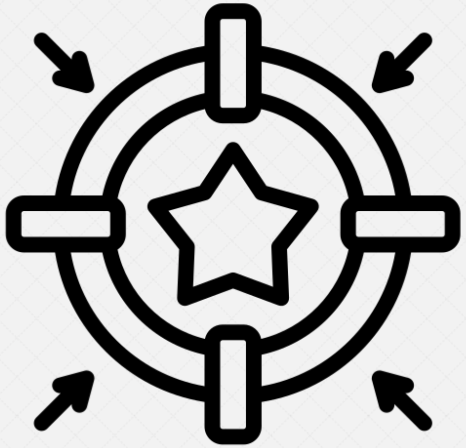
- When an input bit is flipped, 50% of the output bits must be flipped on average
- The ideal value is 50%

We define a table of size *n*n*:

- When the i^{th} input bit is flipped, in which proportion is the j^{th} output bit flipped?

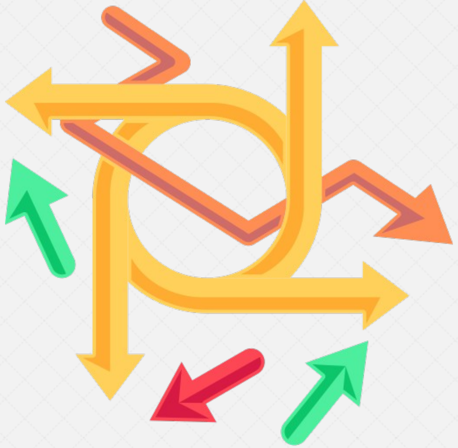
Each table value should be as close as possible of 50%

Differential uniformity



- Gives proximity to a perfectly nonlinear S-Box (impossible for bijectivity)
- For each combination *(a, b)*, differential uniformity table δ gives the number of inputs x such that $S(x) \oplus S(x \oplus a) = b$
- The metric is then $U = \max(\delta)$
- The **lowest value is the best**

Algebraic complexity



Our S-Box is represented over $\mathbb{N}$:

$S(x) = a_0 + a_1 * x + \ldots + a_{(2\hat{}n)-1} * x^{(2\hat{}n)-1}$ mod 2^n avec $x, a_0, a_1, \ldots \in [\![0, 2^n-1]\!]$

Algebraic complexity is the number of monomials in the univariate polynomial

A **large value protects against interpolation attacks**

Linear Approximation probability

- Gives an indication about S-Box resistance against linear cryptanalysis
- Defined as the maximum correlation between $\alpha * x$ et $\beta * S(x)$, pour tout α et $\beta \in [\![1, 2^n]\!]$
- **Lowest value is the best**

Differential Approximation probability

Given by the XOR distribution between input and output

- For each combination *(Δx, Δy)*, differential probability table DP gives the number of inputs *x* such that $S(x) \oplus S(x \oplus \Delta x) = \Delta y$
- So DAP = max(DP)

A **low value ensures resistance** against differential cryptanalysis

Boomerang Uniformity

- Defines S-Box resistance against boomerang attacks (a variant of differential cryptanalysis)
- For each combination *(a, b)*, Boomerang Connectivity Table (BCT) gives the number of inputs *x* such that:

$$S\text{^-}1(S(x) \oplus b) \oplus S\text{^-}1(S(x \oplus a) \oplus b) = a$$

- *BU* = max(BCT)
- The **lowest value is the best** against boomerang attacks