



HAL
open science

How can we ensure that (symmetric) cryptographic primitives are trustworthy?

Léo Perrin

► **To cite this version:**

Léo Perrin. How can we ensure that (symmetric) cryptographic primitives are trustworthy?. Rostock University Seminar, Jun 2023, Rostock, Germany. 2023. hal-04327478

HAL Id: hal-04327478

<https://inria.hal.science/hal-04327478>

Submitted on 6 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

How can we ensure that (symmetric) cryptographic primitives are trustworthy?

Léo Perrin

Based on joint works with Biryukov, Bonnetain, Canteaut,
Duval, Tian and Udovenko

June, 2023
University of Rostock



$\pi' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241, 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).$

How can we go from \uparrow to \downarrow ?

$$\pi : \begin{cases} \mathbb{F}_{2^8} & \rightarrow \mathbb{F}_{2^8} \\ 0 & \mapsto \kappa(0), \\ (\alpha^{2^m+1})^j & \mapsto \kappa(2^m - j), \text{ for } 1 \leq j \leq 2^m - 1, \\ \alpha^{j+(2^m+1)j} & \mapsto \kappa(2^m - i) \oplus (\alpha^{2^m+1})^{s(j)}, \text{ for } 0 < i, 0 \leq j < 2^m - 1. \end{cases}$$

$\pi' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241, 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).$

How can we go from \uparrow to \downarrow ?

$$\pi : \begin{cases} \mathbb{F}_{2^8} & \rightarrow \mathbb{F}_{2^8} \\ 0 & \mapsto \kappa(0), \\ (\alpha^{2^m+1})^j & \mapsto \kappa(2^m - j), \text{ for } 1 \leq j \leq 2^m - 1, \\ \alpha^{j+(2^m+1)j} & \mapsto \kappa(2^m - i) \oplus (\alpha^{2^m+1})^{s(j)}, \text{ for } 0 < i, 0 \leq j < 2^m - 1. \end{cases}$$

Why do we care?

Outline

- 1 What are S-Boxes?
- 2 On Standardization and Trust
- 3 Don't Trust, Verify (and How to do it)
- 4 Conclusion

Outline

- 1** What are S-Boxes?
- 2 On Standardization and Trust
- 3 Don't Trust, Verify (and How to do it)
- 4 Conclusion

Plan of this Section

- 1** What are S-Boxes?
 - Basics of Symmetric Cryptography
 - Block Cipher Design
- 2 On Standardization and Trust
- 3 Don't Trust, Verify (and How to do it)
- 4 Conclusion

Symmetric Cryptography

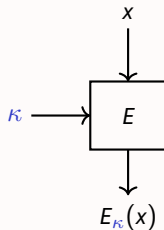
There are many **symmetric** algorithms! Hash functions, MACs...

Symmetric Cryptography

There are many **symmetric** algorithms! Hash functions, MACs...

Definition (Block Cipher)

- Input: n -bit block x
- Parameter: k -bit key κ
- Output: n -bit block $E_{\kappa}(x)$
- Symmetry: E and E^{-1} use the same κ

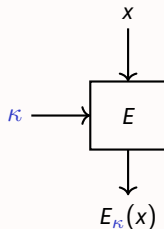


Symmetric Cryptography

There are many **symmetric** algorithms! Hash functions, MACs...

Definition (Block Cipher)

- Input: n -bit block x
- Parameter: k -bit key κ
- Output: n -bit block $E_{\kappa}(x)$
- Symmetry: E and E^{-1} use the same κ



Properties needed:

Diffusion

Confusion

No cryptanalysis!

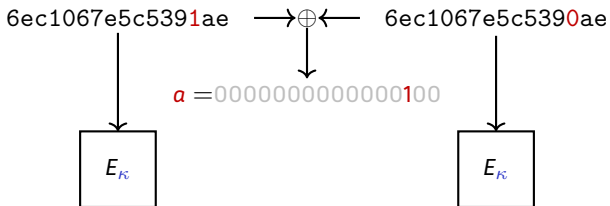
No Cryptanalysis?

Let us look at a typical cryptanalysis technique:
the **differential attack**.

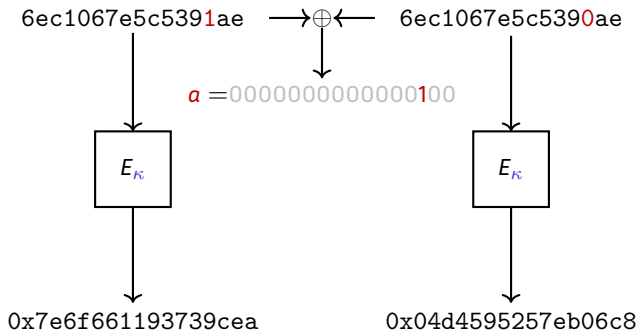
Differential Attacks

$$\begin{array}{ccc} 6ec1067e5c5391ae & \xrightarrow{\oplus} \xleftarrow{} & 6ec1067e5c5390ae \\ & \downarrow & \\ a = 000000000000000100 & & \end{array}$$

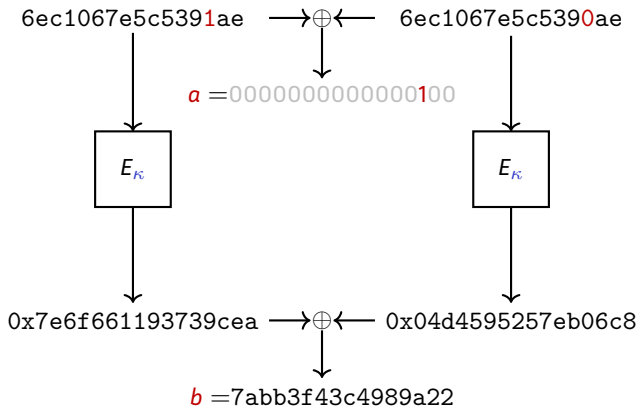
Differential Attacks



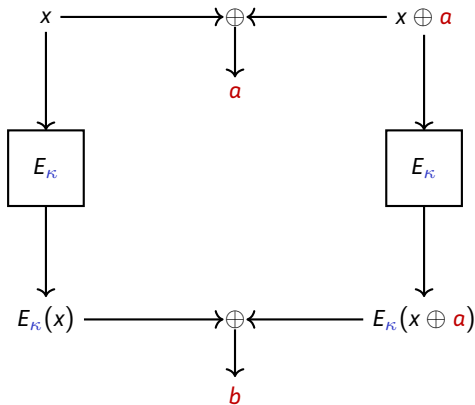
Differential Attacks



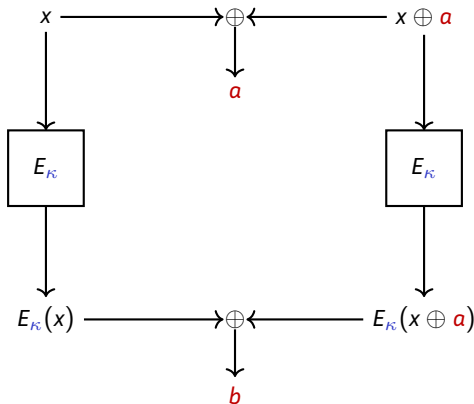
Differential Attacks



Differential Attacks



Differential Attacks



Differential Attack

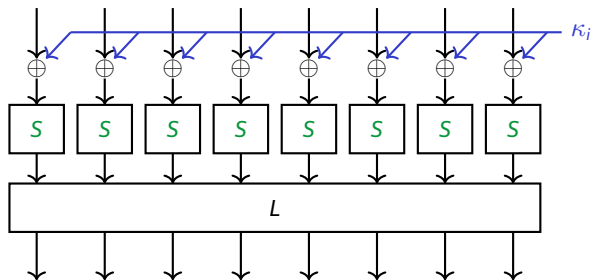
If there are many x such that $E_{\kappa}(x) \oplus E_{\kappa}(x \oplus a) = b$, then the cipher is **not secure**.

Basic Block Cipher Structure

How do we build block ciphers that prevent such attacks (as well as others)?

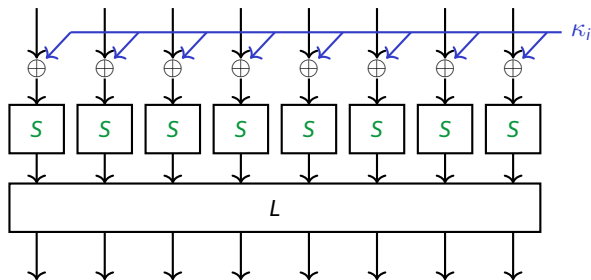
Basic Block Cipher Structure

How do we build block ciphers that prevent such attacks (as well as others)?



Basic Block Cipher Structure

How do we build block ciphers that prevent such attacks (as well as others)?



Substitution-Permutation Network

Such a block cipher iterates the round function above several times. **S** is the Substitution Box (S-Box).

The S-Box (1/2)

π' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241, 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).

The S-Box π of the latest Russian standards, *Kuznyechik (BC)* and *Streebog (HF)*.

The S-Box (2/2)

Importance of the S-Box

If S is such that

$$S(x) \oplus S(x \oplus a) = b$$

does not have many solutions x for all (a, b) then the cipher may be proved secure against differential attacks.

The S-Box (2/2)

Importance of the S-Box

If S is such that

$$S(x) \oplus S(x \oplus a) = b$$

does not have many solutions x for all (a, b) then the cipher may be proved secure against differential attacks.

In **academic** papers presenting new block ciphers, the choice of S is carefully explained.

S-Box Design

- AES S-Box
- Inverse (other)
- Exponential
- Math (other)
- SPN
- Misty
- Feistel
- Lai-Massey
- Pseudo-random
- Hill climbing
- Unknown

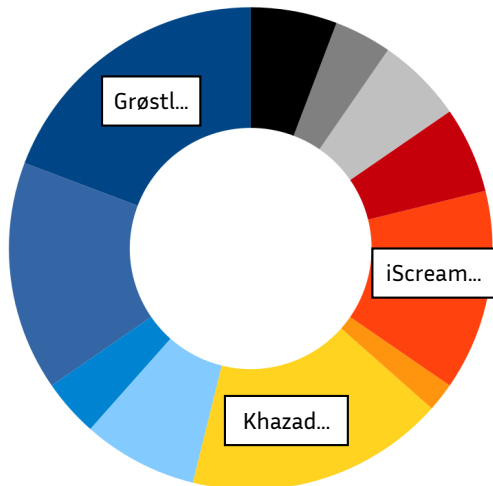
S-Box Design

- AES S-Box
- Inverse (other)
- Exponential
- Math (other)
- SPN
- Misty
- Feistel
- Lai-Massey
- Pseudo-random
- Hill climbing
- Unknown



S-Box Design

- AES S-Box
- Inverse (other)
- Exponential
- Math (other)
- SPN
- Misty
- Feistel
- Lai-Massey
- Pseudo-random
- Hill climbing
- Unknown



Outline

- 1 What are S-Boxes?
- 2 On Standardization and Trust**
- 3 Don't Trust, Verify (and How to do it)
- 4 Conclusion

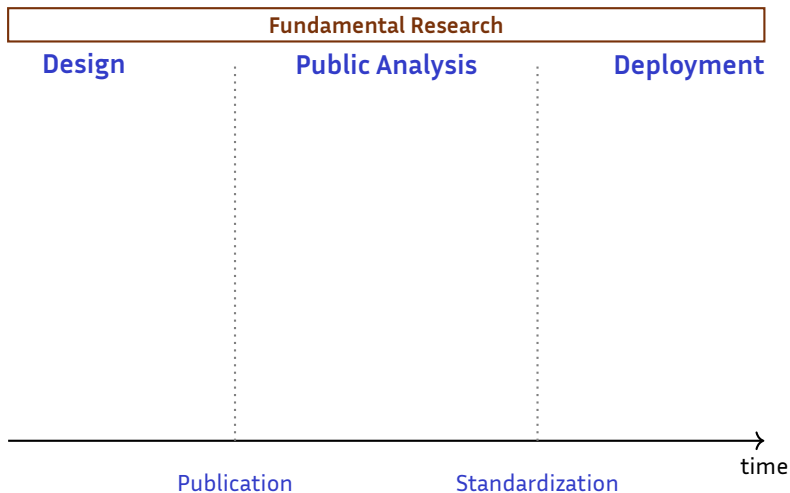
Plan of this Section

- 1 What are S-Boxes?
- 2 **On Standardization and Trust**
 - How Standardization Works
 - How Standardization Fails
- 3 Don't Trust, Verify (and How to do it)
- 4 Conclusion

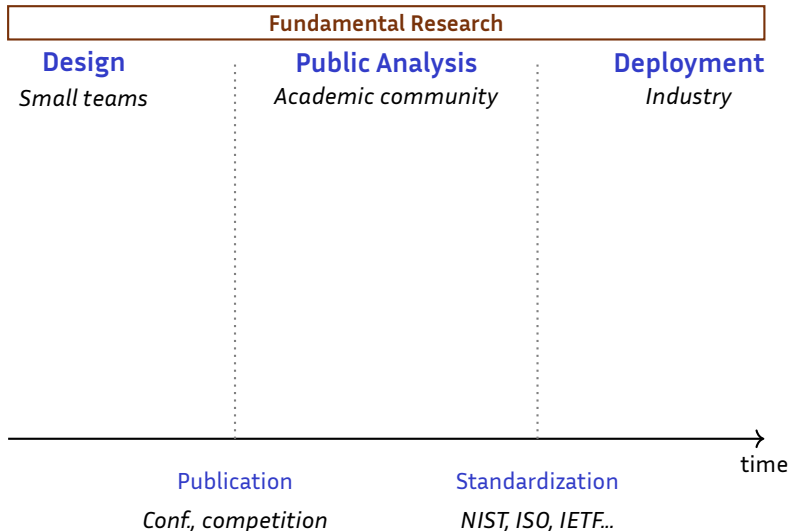
Life Cycle of a Cryptographic Primitive

Fundamental Research

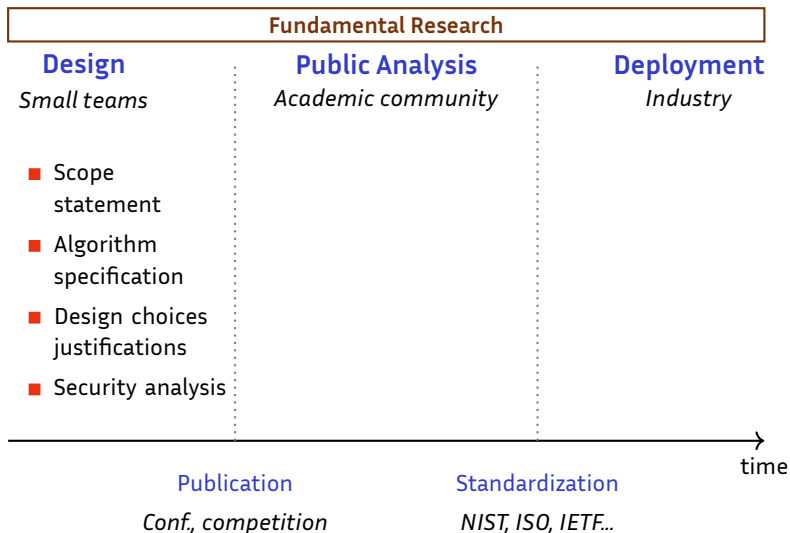
Life Cycle of a Cryptographic Primitive



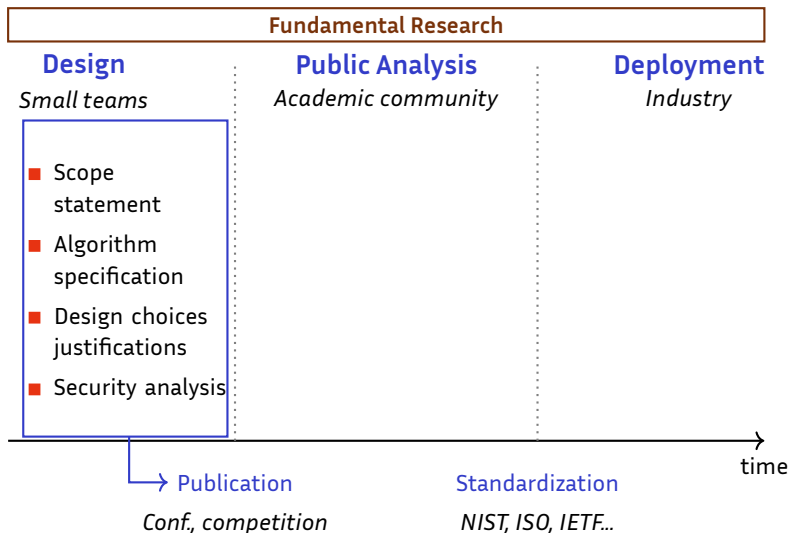
Life Cycle of a Cryptographic Primitive



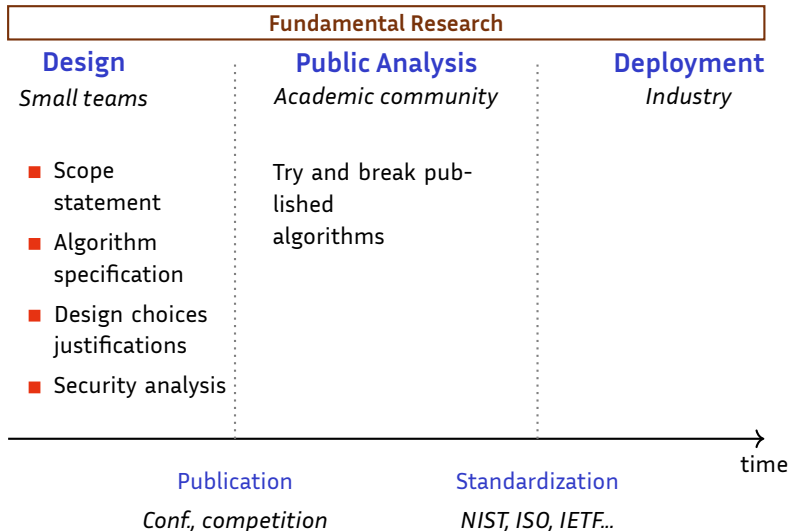
Life Cycle of a Cryptographic Primitive



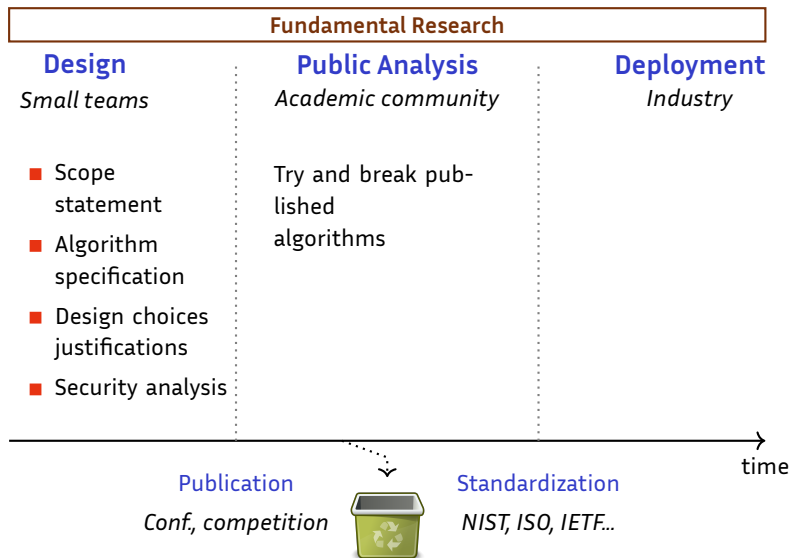
Life Cycle of a Cryptographic Primitive



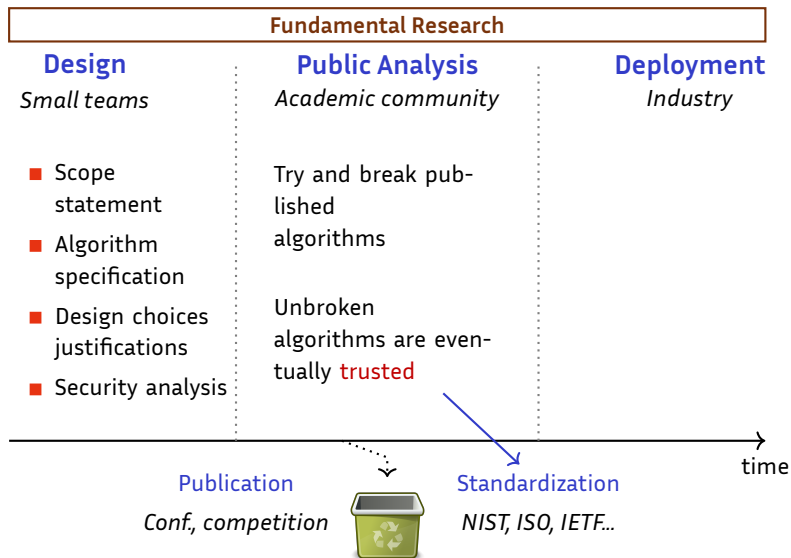
Life Cycle of a Cryptographic Primitive



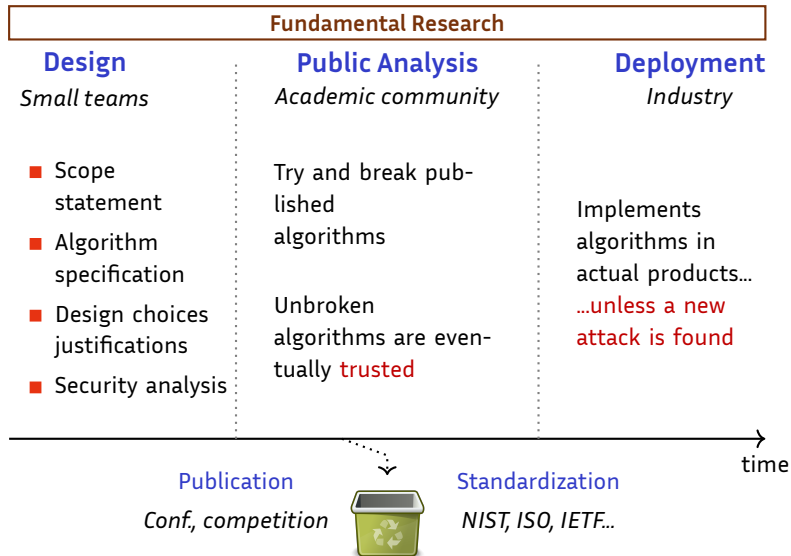
Life Cycle of a Cryptographic Primitive



Life Cycle of a Cryptographic Primitive



Life Cycle of a Cryptographic Primitive



Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

March 2014 First mentions of a LW standardization process

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions
- March 2019** Submission deadline/start of first round (62 candidates)

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions
- March 2019** Submission deadline/start of first round (62 candidates)



CRYPTANALYSIS TIME

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions
- March 2019** Submission deadline/start of first round (62 candidates)
- August 2019** start of second round (32 candidates left)

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions
- March 2019** Submission deadline/start of first round (62 candidates)
- August 2019** start of second round (32 candidates left)



CRYPTANALYSIS TIME

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions
- March 2019** Submission deadline/start of first round (62 candidates)
- August 2019** start of second round (32 candidates left)
- March 2021** start of third (final) round (10 candidates left)

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions
- March 2019** Submission deadline/start of first round (62 candidates)
- August 2019** start of second round (32 candidates left)
- March 2021** start of third (final) round (10 candidates left)



not much cryptanalysis...

Some Examples

- **Advanced Encryption Standard** competition → AES block cipher
- **Secure Hashing Algorithm-3** competition → SHA-3 hash function

Lightweight Cryptography (not) Competition

- March 2014** First mentions of a LW standardization process
- 2015-2018** Calls for comments (several iterations)
- August 2018** Call for submissions
- March 2019** Submission deadline/start of first round (62 candidates)
- August 2019** start of second round (32 candidates left)
- March 2021** start of third (final) round (10 candidates left)
- February 2023** NIST chose **ASCON**

The best approach is the one used in practice!

The best approach is the one used in practice!

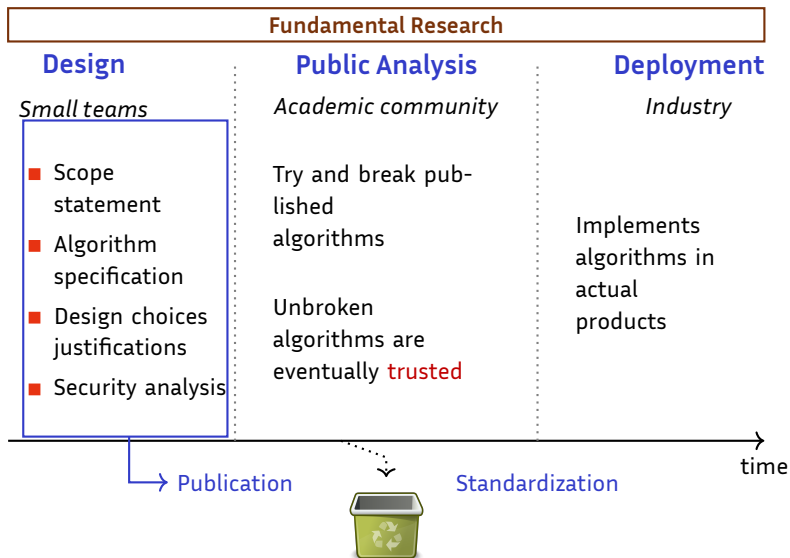
All is well?

The best approach is the one used in practice!

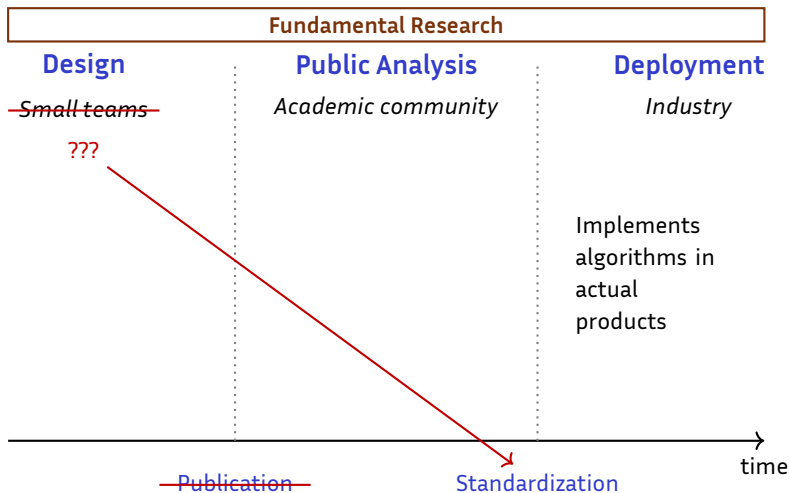
All is well?

... no.

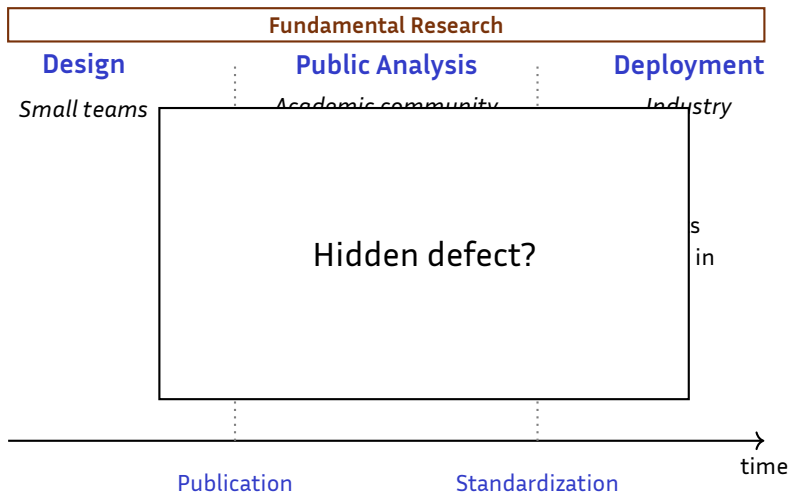
Breaking the Pipeline



Breaking the Pipeline



Breaking the Pipeline



Some (Counter-)Examples

`https://eprint.iacr.org/<year>/<number>`

A5/2

- stream cipher
- 1980's
- used in 2G
- designer: ??
- wikipedia

DUAL EC

- PRNG
- 2000's
- general standard
- designer: NSA
- 2015/767
(2016/376)

GEA-1

- 1990's
- stream cipher
- used in EDGE
- designer: ??
- 2021/819

Outline

- 1 What are S-Boxes?
- 2 On Standardization and Trust
- 3 Don't Trust, Verify (and How to do it)**
- 4 Conclusion

Plan of this Section

- 1 What are S-Boxes?
- 2 On Standardization and Trust
- 3 Don't Trust, Verify (and How to do it)
 - Streebog and Kuznyechik
 - Decomposing the Mysterious S-Box
 - Generation Process
 - Cryptographic Properties
- 4 Conclusion

Kuznyechik/Stribog

Stribog

Type Hash function

Publication 2012

Kuznyechik

Type Block cipher

Publication 2015



Kuznyechik/Stribog

Stribog

Type Hash function

Publication 2012

Kuznyechik

Type Block cipher

Publication 2015



Common ground

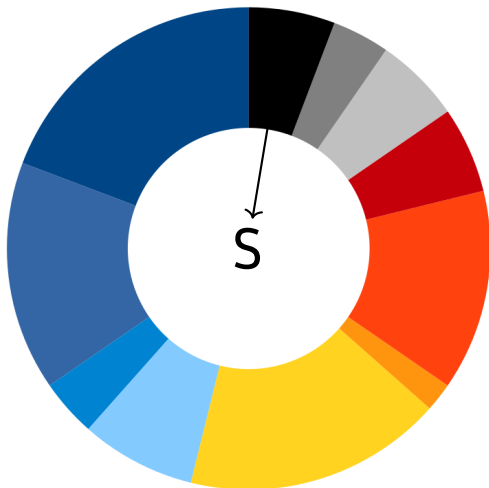
- Both are standard symmetric primitives in Russia.
- Both were designed by the FSB (TC26).
- Both use the same 8×8 S-Box, *π* .

π

$\pi' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241, 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).$

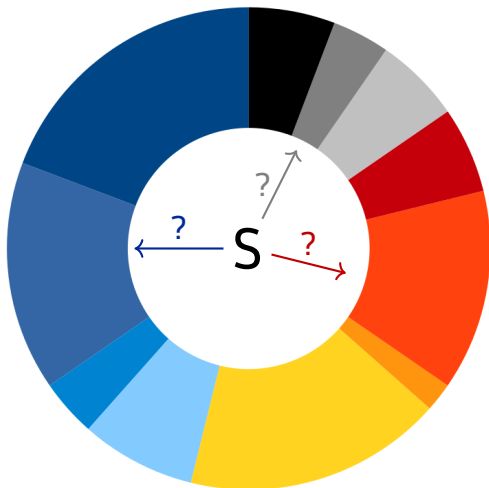
S-Box Reverse-Engineering

- AES S-Box
- Inverse (other)
- Exponential
- Math (other)
- SPN
- Misty
- Feistel
- Lai-Massey
- Pseudo-random
- Hill climbing
- Unknown



S-Box Reverse-Engineering

- AES S-Box
- Inverse (other)
- Exponential
- Math (other)
- SPN
- Misty
- Feistel
- Lai-Massey
- Pseudo-random
- Hill climbing
- Unknown



The Two Tables

Let $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be an S-Box.

The Two Tables

Let $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be an S-Box.

Definition (DDT)

The *Difference Distribution Table* of S is a matrix of size $2^n \times 2^n$ such that

$$\text{DDT}[a, b] = \#\{x \in \mathbb{F}_2^n \mid S(x \oplus a) \oplus S(x) = b\}.$$

The Two Tables

Let $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be an S-Box.

Definition (DDT)

The *Difference Distribution Table* of S is a matrix of size $2^n \times 2^n$ such that

$$\text{DDT}[a, b] = \#\{x \in \mathbb{F}_2^n \mid S(x \oplus a) \oplus S(x) = b\}.$$

Definition (LAT)

The *Linear Approximations Table* of S is a matrix of size $2^n \times 2^n$ such that

$$\begin{aligned} \text{LAT}[a, b] &= \#\{x \in \mathbb{F}_2^n \mid x \cdot a = S(x) \cdot b\} - 2^{n-1} \\ &= \frac{1}{2} \times \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x + b \cdot S(x)} \end{aligned}$$

Example

$$S = [4, 2, 1, 6, 0, 5, 7, 3]$$

The **DDT** of S .

$$\begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \end{bmatrix}$$

The **LAT** of S .

$$\begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & -2 \\ 0 & 2 & 2 & 0 & 0 & 2 & -2 & 0 \\ 0 & 2 & 0 & 2 & 0 & -2 & 0 & 2 \\ 0 & 2 & 0 & -2 & 0 & -2 & 0 & -2 \\ 0 & -2 & 2 & 0 & 0 & -2 & -2 & 0 \\ 0 & 0 & -2 & 2 & 0 & 0 & -2 & -2 \\ 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 \end{bmatrix}$$

Coding Time!

- 1 Using the `sage.crypto.sboxes` module.
- 2 The AES S-box: differential uniformity, etc
- 3 The Jackson Pollock representation
- 4 Comparison with a random permutation

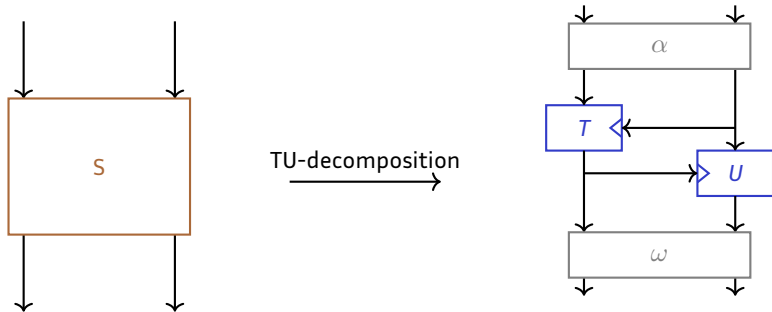
Coding Time! (Kuznyechik)

- 1 JP representation of the LAT of π
- 2 Reordering the columns
- 3 Reordering both rows and columns with linear permutations
- 4 Deduce an interesting permutation $L' \circ \pi \circ L$
- 5 Notice the **integral distinguisher**

The TU-Decomposition

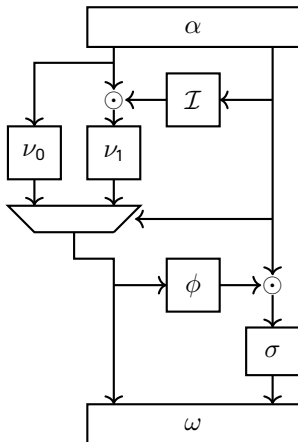
Definition

The **TU-decomposition** is a decomposition algorithm working against S-Boxes with **vector spaces** of zeroes in their LAT.



T and U are mini-block ciphers ; μ and η are linear permutations.

Final Decomposition Number 1



\odot Multiplication in \mathbb{F}_{2^4}

α Linear permutation

\mathcal{I} Inversion in \mathbb{F}_{2^4}

ν_0, ν_1, σ 4×4 permutations

ϕ 4×4 function

ω Linear permutation

Conclusion for Kuznyechik/Stribog?

The Russian S-Box was built like a strange Feistel...

Conclusion for Kuznyechik/Stribog?

**The Russian S-Box was built like a
strange Feistel...**

... or was it?

Conclusion for Kuznyechik/Stribog?

The Russian S-Box was built like a strange Feistel...

... or was it?

Belarussian inspiration

- The last standard of Belarus (BelT) uses an 8-bit S-box,
- somewhat similar to π ...

Conclusion for Kuznyechik/Stribog?

The Russian S-Box was built like a
strange Feistel...

... or was it?

Belarussian inspiration

- The last standard of Belarus (BelT) uses an 8-bit S-box,
- somewhat similar to π ...
- ... based on a **finite field exponential!**

Timeline

July 2012 GOST standardization of Stribog

Aug. 2013 RFC for Stribog (RFC6986)

June 2015 GOST standardization of Kuznyechik

Mar. 2016 RFC for Kuznyechik (RFC7801)

¹A. Biryukov, L. Perrin, A. Udovenko. *Reverse-engineering the S-box of Stribog, Kuznyechik and STRIBOBr1*. EUROCRYPT'16

Timeline

- July 2012 GOST standardization of Streebog
- Aug. 2013 RFC for Streebog (RFC6986)
- June 2015 GOST standardization of Kuznyechik
- Mar. 2016 RFC for Kuznyechik (RFC7801)
- May 2016 Publication of the first decomposition¹

¹A. Biryukov, L. Perrin, A. Udovenko. *Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1*. EUROCRYPT'16

Timeline

- July 2012 GOST standardization of Streebog
- Aug. 2013 RFC for Streebog (RFC6986)
- June 2015 GOST standardization of Kuznyechik
- Mar. 2016 RFC for Kuznyechik (RFC7801)
- May 2016 Publication of the first decomposition¹
- Oct. 2018 ISO standardization of Streebog (ISO 10118-3)

¹A. Biryukov, L. Perrin, A. Udovenko. *Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1*. EUROCRYPT'16

A Third and Final Decomposition: the TKlog

π is a TKlog!

π operates on $\mathbb{F}_{2^{2m}}$ where $m = 4$ using:

- α : a generator of $\mathbb{F}_{2^{2m}}$,
- κ : an affine function $\mathbb{F}_2^m \rightarrow \mathbb{F}_{2^{2m}}$ with $\kappa(\mathbb{F}_2^m) \oplus \mathbb{F}_{2^m} = \mathbb{F}_{2^{2m}}$,
- s : a permutation of $\mathbb{Z}/(2^m - 1)\mathbb{Z}$;

it works as follows:

$$\begin{cases} \pi(0) & = \kappa(0), \\ \pi((\alpha^{2^m+1})^j) & = \kappa(2^m - j), \text{ for } 1 \leq j \leq 2^m - 1, \\ \pi(\alpha^{i+(2^m+1)j}) & = \kappa(2^m - i) \oplus (\alpha^{2^m+1})^{s(j)}, \text{ for } 0 < i, 0 \leq j < 2^m - 1. \end{cases}$$

Timeline

- July 2012** GOST standardization of Stribog
- Aug. 2013** RFC for Stribog (RFC6986)
- June 2015** GOST standardization of Kuznyechik
- Mar. 2016** RFC for Kuznyechik (RFC7801)
- May 2016** Publication of the first decomposition
- Oct. 2018** ISO standardization of Stribog (ISO 10118-3)
- Jan. 2019** Publication of the final decomposition²

²L. Perrin. *Partitions in the S-box of Stribog and Kuznyechik*. IACR ToSC. 2019.

Timeline

- July 2012** GOST standardization of Streebog
- Aug. 2013** RFC for Streebog (RFC6986)
- June 2015** GOST standardization of Kuznyechik
- Mar. 2016** RFC for Kuznyechik (RFC7801)
- May 2016** Publication of the first decomposition
- Oct. 2018** ISO standardization of Streebog (ISO 10118-3)
- Jan. 2019** Publication of the final decomposition²
- Feb. 2019** Kuznyechik at ISO: decision post-poned

²L. Perrin. *Partitions in the S-box of Streebog and Kuznyechik*. IACR ToSC. 2019.

Timeline

- July 2012** GOST standardization of Streebog
- Aug. 2013** RFC for Streebog (RFC6986)
- June 2015** GOST standardization of Kuznyechik
- Mar. 2016** RFC for Kuznyechik (RFC7801)
- May 2016** Publication of the first decomposition
- Oct. 2018** ISO standardization of Streebog (ISO 10118-3)
- Jan. 2019** Publication of the final decomposition²
- Feb. 2019** Kuznyechik at ISO: decision post-poned
- Sep. 2019** Kuznyechik at ISO: decision had to be taken!

²L. Perrin. *Partitions in the S-box of Streebog and Kuznyechik*. IACR ToSC. 2019.

From the Designers, at ISO

questioned is the S-box π . This S-box was chosen from Streebog hash-function and it was synthesized in 2007. Note that through many years of cryptanalysis no weakness of this S-box was found. The S-box π was obtained by pseudo-random search and the following properties were taken into account.

[...]

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

From the Designers, at ISO

questioned is the S-box π . This S-box was chosen from Streebog hash-function and it was synthesized in 2007. Note that through many years of cryptanalysis no weakness of this S-box was found. The S-box π was obtained by pseudo-random search and the following properties were taken into account.

[...]

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

Everything is wrong except for the green part.

The Russian S-box is too simple

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

- 165 ASCII characters that fit on 7 bits: this program is 1155-bit long
- It is **impossible** that all 2^{1684} 8-bit permutations have an implementation this short!

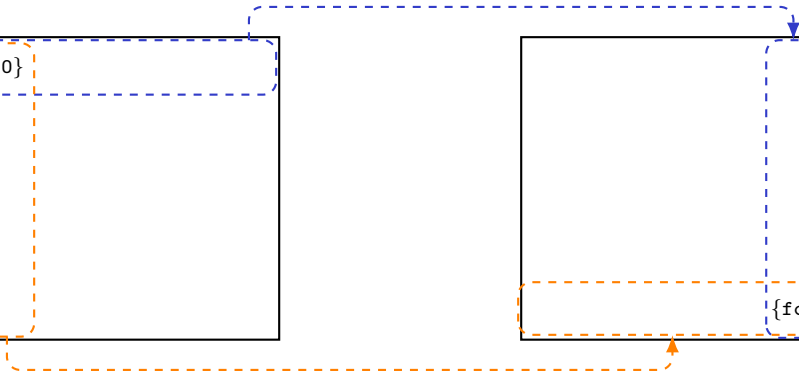
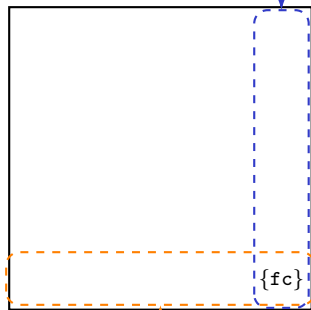
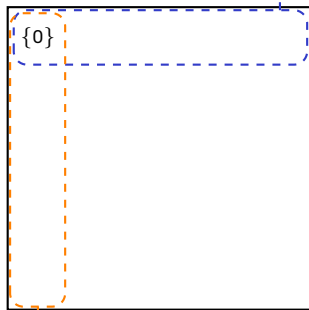
The Russian S-box is too simple

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

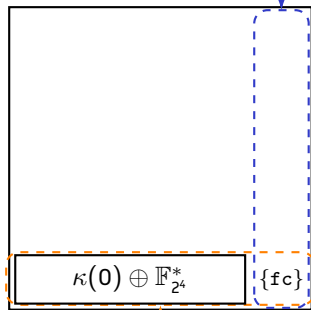
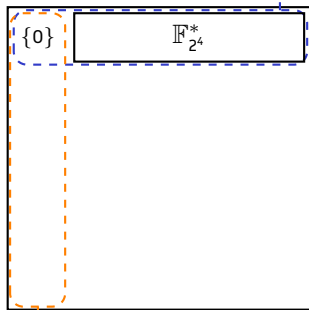
- 165 ASCII characters that fit on 7 bits: this program is 1155-bit long
- It is **impossible** that all 2^{1684} 8-bit permutations have an implementation this short!

<https://codegolf.stackexchange.com/questions/186498/proving-that-a-russian-cryptographic-standard-is-too-structured>

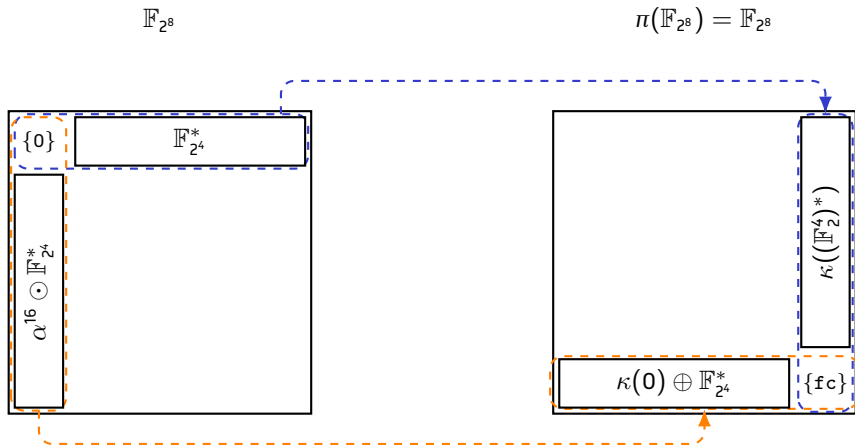
Cosets to Cosets

 \mathbb{F}_{2^8}
 $\pi(\mathbb{F}_{2^8}) = \mathbb{F}_{2^8}$


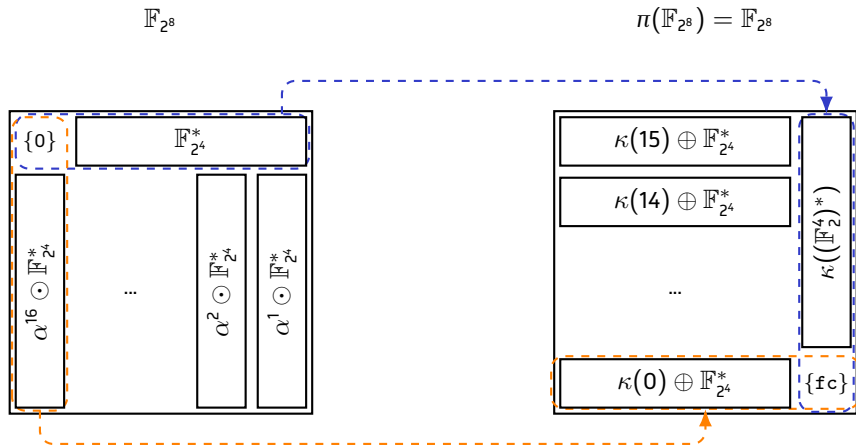
Cosets to Cosets

 \mathbb{F}_{2^8}
 $\pi(\mathbb{F}_{2^8}) = \mathbb{F}_{2^8}$


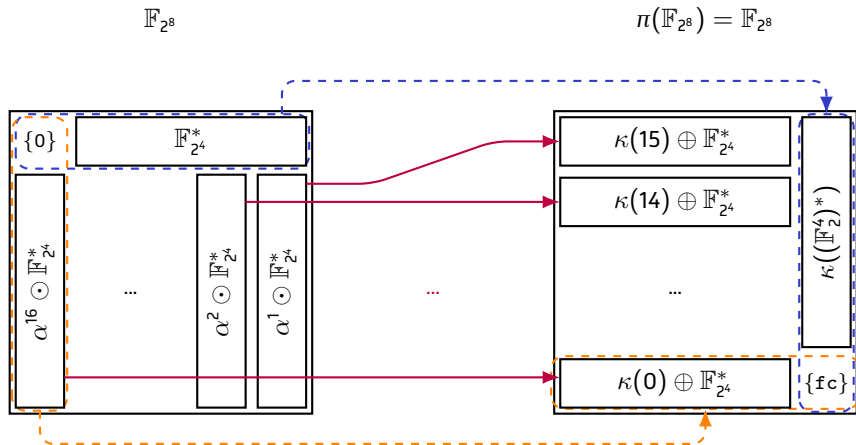
Cosets to Cosets



Cosets to Cosets

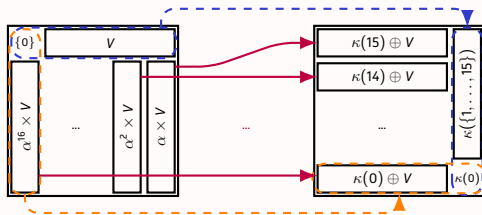


Cosets to Cosets

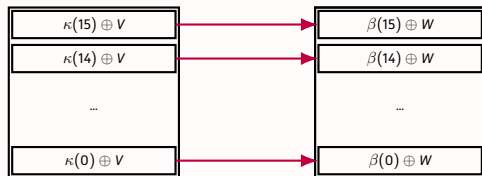


Why it is Worrying

Russia's π



Backdoored S-box



Timeline

- July 2012** GOST standardization of Stribog
- Aug. 2013** RFC for Stribog (RFC6986)
- June 2015** GOST standardization of Kuznyechik
- Mar. 2016** RFC for Kuznyechik (RFC7801)
- May 2016** Publication of the first decomposition
- Oct. 2018** ISO standardization of Stribog (ISO 10118-3)
- Jan. 2019** Publication of the final decomposition³
- Feb. 2019** Kuznyechik at ISO: decision post-poned
- Sep. 2019** Kuznyechik at ISO: **thanks, but no thanks**

The Russian delegation failed to convince the experts that they were honest: the standardization of Kuznyechik was **refused**

Outline

- 1 What are S-Boxes?
- 2 On Standardization and Trust
- 3 Don't Trust, Verify (and How to do it)
- 4 Conclusion**

Plan of this Section

- 1 What are S-Boxes?
- 2 On Standardization and Trust
- 3 Don't Trust, Verify (and How to do it)
- 4 Conclusion**

Conclusion

- 1 Cryptographers use mathematics but mathematicians could also use crypto!

Conclusion

- 1 Cryptographers use mathematics but mathematicians could also use crypto!
- 2 If you **design** a cipher, **justify** every step of your design.

Conclusion

- 1 Cryptographers use mathematics but mathematicians could also use crypto!
- 2 If you **design** a cipher, **justify** every step of your design.
- 3 If you **choose** a cipher, **demand** a full design explanation.

Conclusion

- 1 Cryptographers use mathematics but mathematicians could also use crypto!
- 2 If you **design** a cipher, **justify** every step of your design.
- 3 If you **choose** a cipher, **demand** a full design explanation.
- 4 **When in doubt, don't trust!** (at least as far as crypto is concerned)

The Last S-Box

14	11	60	6d	e9	10	e3	2	b	90	d	17	c5	b0	9f	c5
d8	da	be	22	8	f3	4	a9	fe	f3	f5	fc	bc	30	be	26
bb	88	85	46	f4	2e	e	fd	76	fe	b0	11	4e	de	35	bb
30	4b	30	d6	dd	df	df	d4	90	7a	d8	8c	6a	89	30	39
e9	1	da	d2	85	87	d3	d4	ba	2b	d4	9f	9c	38	8c	55
d3	86	bb	db	ec	e0	46	48	bf	46	1b	1c	d7	d9	1b	e0
23	d4	d7	7f	16	3f	3	3	44	c3	59	10	2a	da	ed	e9
8e	d8	d1	db	cb	cb	c3	c7	38	22	34	3d	db	85	23	7c
24	d1	d8	2e	fc	44	8	38	c8	c7	39	4c	5f	56	2a	cf
d0	e9	d2	68	e4	e3	e9	13	e2	c	97	e4	60	29	d7	9b
d9	16	24	94	b3	e3	4c	4c	4f	39	e0	4b	bc	2c	d3	94
81	96	93	84	91	d0	2e	d6	d2	2b	78	ef	d6	9e	7b	72
ad	c4	68	92	7a	d2	5	2b	1e	d0	dc	b1	22	3f	c3	c3
88	b1	8d	b5	e3	4e	d7	81	3	15	17	25	4e	65	88	4e
e4	3b	81	81	fa	1	1d	4	22	0	6	1	27	68	27	2e
3b	83	c7	cc	25	9b	d8	d5	1c	1f	e5	59	7f	3f	3f	ef

