Streebog and Kuznyechik
Inconsistencies in the Claims of their Designers

Léo Perrin
leo.perrin@inria.fr
@lpp_crypto

IETF Workshop, Montréal
Partitions in the S-Box of Streebog and Kuznyechik
Léo Perrin
Inria, France
leo.perrin@inria.fr

Abstract. Streebog and Kuznyechik are the latest symmetric cryptographic primitives standardized by the Russian GOST. They share the same S-Box, \( \pi \), whose design process was not described by its authors. In previous works, Biryukov, Perrin and Udovenko recovered two completely different decompositions of this S-Box. We revisit their results and identify a third decomposition of \( \pi \). It is an instance of a fairly small family of permutations operating on \( 2^m \) bits which we call TKlog and which is closely related to finite field logarithms. Its simplicity and the small number of components it uses lead us to claim that it has to be the structure intentionally used by the designers of Streebog and Kuznyechik. The \( 2^m \)-bit permutations of this type have a very strong algebraic structure: they map multiplicative cosets of the subfield \( GF(2^m)^* \) to additive cosets of \( GF(2^n)^* \). Furthermore, the function relating each multiplicative coset to the corresponding additive coset is always essentially the same. To the best of our knowledge, we are the first to expose this very strong algebraic structure. We also investigate other properties of the TKlog and show in particular that it can always be decomposed in a fashion similar to the first decomposition of Biryukov et al., thus explaining the relation between the two previous decompositions. It also means that it is always possible to implement a TKlog efficiently in hardware and that it always exhibits a visual pattern in its LAT similar to the one present in \( \pi \). While we could not find attacks based on these new results, we discuss the impact of our work on the security of Streebog and Kuznyechik. To this end, we provide a new simpler representation of the linear layer of Streebog as a matrix multiplication in the exact same field as the one used to define \( \pi \). We deduce that this matrix interacts in a non-trivial way with the partitions preserved by \( \pi \).

Keywords: Boolean functions · Kuznyechik · Streebog · Reverse-Engineering · Partitions · Cosets · TKlog

1 Introduction

Many symmetric primitives rely on S-Boxes as their unique source of non-linearity, including the AES [AES1]. Such objects are small functions mapping \( \mathbb{F}_2^n \) to \( \mathbb{F}_2^n \) which are often specified via their look-up tables. Their choice is crucial as both the security and the efficiency of the primitive depend heavily on their properties. For example, a low differential uniformity [Ny94] implies a higher resilience against differential attacks [HS99, BS91b]. On the other hand, the existence of a simple decomposition greatly helps with an efficient bitized or hardware implementation [LW14, CDL16]. Thus, algorithm designers are expected to provide detailed explanation about their choice of S-Box. Each cipher that was published at a cryptography or security conference has provided such explanations. There are two prominent S-Boxes for which this information has not been provided. The first is the so-called “F-table” of Skipjack [L98], a lightweight block cipher designed...
# Outline

1. Standards and S-boxes
2. On the S-box of RFC 6986 and 7801
3. The Core Issue: the S-Box Generation Process
4. Conclusion
Outline

1 Standards and S-boxes

2 On the S-box of RFC 6986 and 7801

3 The Core Issue: the S-Box Generation Process

4 Conclusion
Life Cycle of a Cryptographic Primitive
Life Cycle of a Cryptographic Primitive

- **Design**
- **Public Analysis**
- **Deployment**

- **Publication**
- **Standardization**

**Time**
Standards and S-boxes
On the S-box of RFC 6986 and 7801
The Core Issue: the S-Box Generation Process
Conclusion

Life Cycle of a Cryptographic Primitive

**Design**
*Small teams*

**Public Analysis**
*Academic community*

**Deployment**
*Industry*

**Publication**
*Conf., competition*

**Standardization**
*NIST, ISO, IETF...*
Life Cycle of a Cryptographic Primitive

**Design**
*Small teams*
- Scope statement
- Algorithm specification
- Design choices justifications
- Security analysis

**Public Analysis**
*Academic community*

**Deployment**
*Industry*

**Publication**
*Conf., competition*

**Standardization**
*NIST, ISO, IETF...*
Life Cycle of a Cryptographic Primitive

Design
Small teams
- Scope statement
- Algorithm specification
- Design choices justifications
- Security analysis

Public Analysis
Academic community

Publication
Conf., competition

Deployment
Industry

Standardization
NIST, ISO, IETF...
Life Cycle of a Cryptographic Primitive

**Design**
- Small teams
  - Scope statement
  - Algorithm specification
  - Design choices justifications
  - Security analysis

**Public Analysis**
- Academic community
  - Try and break published algorithms

**Deployment**
- Industry

**Publication**
- Conf., competition

**Standardization**
- NIST, ISO, IETF...
Life Cycle of a Cryptographic Primitive

**Design**
- Small teams
  - Scope statement
  - Algorithm specification
  - Design choices justifications
  - Security analysis

**Public Analysis**
- Academic community
  - Try and break published algorithms

**Deployment**
- Industry

---

**Publication**
- Conf., competition

**Standardization**
- NIST, ISO, IETF...
Standards and S-boxes
On the S-box of RFC 6986 and 7801
The Core Issue: the S-Box Generation Process
Conclusion

Life Cycle of a Cryptographic Primitive

**Design**
*Small teams*
- Scope statement
- Algorithm specification
- Design choices justifications
- Security analysis

**Public Analysis**
*Academic community*
- Try and break published algorithms
- Well-studied algorithms are eventually trusted

**Deployment**
*Industry*

- Publication
  *Conf., competition*
- Standardization
  *NIST, ISO, IETF...

**Time**
Life Cycle of a Cryptographic Primitive

Design
Small teams
- Scope statement
- Algorithm specification
- Design choices justifications
- Security analysis

Public Analysis
Academic community
- Try and break published algorithms
- Well-studied algorithms are eventually trusted

Deployment
Industry
- Implements algorithms in actual products...
  ...unless a new attack is found

Publication
Conf., competition

Standardization
NIST, ISO, IETF...
Breaking the Pipeline

**Design**
- Small teams
  - Scope statement
  - Algorithm specification
  - Design choices justifications
  - Security analysis

**Public Analysis**
- Academic community
  - Try and break published algorithms
  - Well-studied algorithms are eventually trusted

**Deployment**
- Industry
  - Implements algorithms in actual products...
  - ...unless a new attack is found

- Publication
  - Conf., competition

- Standardization
  - NIST, ISO, IETF...
Breaking the Pipeline

Design

- Small teams

Public Analysis

- Academic community

Deployment

- Industry

- Implements algorithms in actual products...
  ...unless a new attack is found

Publication

- Conf., competition

Standardization

- NIST, ISO, IETF...
Breaking the Pipeline

Design
Small teams

Public Analysis
Academic community

Deployment
Industry

Publication
Conf., competition

Standardization
NIST, ISO, IETF...

Hidden defect?
S-Boxes

Definition (S(substitution)-box)

An S-box $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ is a small non-linear function operating on a small block size (typically $n \in \{4, 8\}$) which can be specified via its lookup table.
Specifying the AES S-box

7.2 The ByteSub S-box

The design criteria for the S-box are inspired by differential and linear cryptanalysis on the one hand and attacks using algebraic manipulations, such as interpolation attacks, on the other:

1. Invertibility;
2. Minimisation of the largest non-trivial correlation between linear combinations of input bits and linear combination of output bits;
3. Minimisation of the largest non-trivial value in the EXOR table;
4. Complexity of its algebraic expression in GF(2^8);
5. Simplicity of description.

In [Ny94] several methods are given to construct S-boxes that satisfy the first three criteria. For invertible S-boxes operating on bytes, the maximum input/output correlation can be made as low as 2^{-3} and the maximum value in the EXOR table can be as low as 4 (corresponding to a difference propagation probability of 2^{-6}).

We have decided to take from the candidate constructions in [Ny94] the S-box defined by the mapping $x \mapsto x^{-1}$ in GF(2^8).

By definition, the selected mapping has a very simple algebraic expression. This enables algebraic manipulations that can be used to mount attacks such as interpolation attacks [JaKn97]. Therefore, the mapping is modified by composing it with an additional invertible affine transformation. This affine transformation does not affect the properties with respect to the first three criteria, but if properly chosen, allows the S-box to satisfy the fourth criterion.

We have chosen an affine mapping that has a very simple description per se, but a complicated algebraic expression if combined with the 'inverse' mapping. It can be seen as modular polynomial multiplication followed by an addition:

$$b(x) = (x^7 + x^6 + x^3 + x^1 + 1) + a(x)(x^7 + x^6 + x^3 + x^1 + 1) \mod x^8 + 1$$

The modulus has been chosen as the simplest modulus possible. The multiplication polynomial has been chosen from the set of polynomials coprime to the modulus as the one with the simplest description. The constant has been chosen in such a way that that the S-box has no fixed points (S-box(a) = a) and no ‘opposite fixed points’ (S-box(a) = 7).

Note: other S-boxes can be found that satisfy the criteria above. In the case of suspicion of a trapdoor being built into the cipher, the current S-box might be replaced by another one. The cipher structure and number of rounds as defined even allow the use of an S-box that does not optimise the differential and linear cryptanalysis properties (criteria 2 and 3). Even an S-box that is “average” in this respect is likely to provide enough resistance against differential and linear cryptanalysis.
7.2 The ByteSub S-box

The design criteria for the S-box are inspired by differential and linear cryptanalysis on the one hand and algebraic attacks on the other:

1. Invertibility;
2. Minimisation of the largest non-trivial correlation between linear combinations of input bits and linear combination of output bits;
3. Minimisation of the largest non-trivial value in the EXOR table;
4. Complexity of its algebraic expression in $GF(2^8)$;
5. Simplicity of description.

In [Ny94] several methods are given to construct S-boxes that satisfy the first three criteria. For invertible S-boxes operating on bytes, the maximum input/output correlation can be made as low as $2^{-3}$ and the maximum value in the EXOR table can be as low as 4 (corresponding to a difference propagation probability of $2^{-6}$).

We have decided to take from the candidate constructions in [Ny94] the S-box defined by the mapping $x \rightarrow x^{-1}$ in $GF(2^8)$.

By definition, the selected mapping has a very simple algebraic expression. This enables algebraic manipulations that can be used to mount attacks such as interpolation attacks [JaKn97]. Therefore, the mapping is modified by composing it with an additional invertible affine transformation. This affine transformation does not affect the properties with respect to the first three criteria, but if properly chosen, allows the S-box to satisfy the fourth criterion.

We have chosen an affine mapping that has a very simple description per se, but a complicated algebraic expression if combined with the 'inverse' mapping. It can be seen as modular polynomial multiplication followed by an addition:

$$b(x) = (x^2 + x + 1 + 1) \cdot a(x) \cdot (x^2 + x + 1) \bmod x^8 + 1$$

The modulus has been chosen as the simplest modulus possible. The multiplication polynomial has been chosen from the set of polynomials coprime to the modulus as the one with the simplest description. The constant has been chosen in such a way that the S-box has no fixed points (S-box(a) = a) and no 'opposite fixed points' (S-box(a) = $\overline{a}$).

Note: other S-boxes can be found that satisfy the criteria above. In the case of suspicion of a trapdoor being built into the cipher, the current S-box might be replaced by another one. The cipher structure and number of rounds as defined even allow the use of an S-box that does not optimise the differential and linear cryptanalysis properties (criteria 2 and 3). Even an S-box that is "average" in this respect is likely to provide enough resistance against differential and linear cryptanalysis.

7.2 The ByteSub S-box

The design criteria for the S-box are inspired by differential and linear cryptanalysis on the one hand and attacks using algebraic manipulations, such as interpolation attacks, on the other:

1. Invertibility;
2. Minimisation of the largest non-trivial correlation between linear combinations of input bits and linear combination of output bits;
3. Minimisation of the largest non-trivial value in the EXOR table;
4. Complexity of its algebraic expression in GF(2^8);
5. Simplicity of description.

In [Ny94] several methods are given to construct S-boxes that satisfy the first three criteria. For invertible S-boxes operating on bytes, the maximum input/output correlation can be made as low as 2^−3 and the maximum value in the EXOR table can be as low as 4 (corresponding to a difference propagation probability of 2^−6).

We have decided to take from the candidate constructions in [Ny94] the S-box defined by the mapping \( x \mapsto x^3 \) in GF(2^8).

By definition, the selected mapping has a very simple algebraic expression. This enables algebraic manipulations that can be used to mount attacks such as interpolation attacks [JaKn97]. Therefore, the mapping is modified by composing it with an additional invertible affine transformation. This affine transformation does not affect the properties with respect to the first three criteria, but if properly chosen, allows the S-box to satisfy the fourth criterion.

We have chosen an affine mapping that has a very simple description per se, but a complicated algebraic expression if combined with the ‘inverse’ mapping. It can be seen as modular polynomial multiplication followed by an addition:

\[
    b(x) = (x^2 + x + 1) + a \cdot (x^2 + x + 1) \mod x^8 + 1
\]

The modulus has been chosen as the simplest modulus possible. The multiplication polynomial has been chosen from the set of polynomials coprime to the modulus as the one with the simplest description. The constant has been chosen in such a way that the S-box has no fixed points (S-box(a) = a) and no ‘opposite fixed points’ (S-box(a) = \( \overline{a} \)).

Note: other S-boxes can be found that satisfy the criteria above. In the case of suspicion of a trapdoor being built into the cipher, the current S-box might be replaced by another one. The cipher structure and number of rounds as defined even allow the use of an S-box that does not optimise the differential and linear cryptanalysis properties (criteria 2 and 3). Even an S-box that is “average” in this respect is likely to provide enough resistance against differential and linear cryptanalysis.
Specifying the AES S-box

7.2 The ByteSub S-box

The design criteria for the S-box are inspired by differential and linear cryptanalysis on the one hand and attacks using algebraic manipulations, such as interpolation attacks, on the other:

1. Invertibility;
2. Minimisation of the largest non-trivial correlation between linear combinations of input bits and linear combination of output bits;
3. Minimisation of the largest non-trivial value in the EXOR table;
4. Complexity of its algebraic expression in GF(2^8);
5. Simplicity of description.

In [Ny94] several methods are given to construct S-boxes that satisfy the first three criteria. For invertible S-boxes operating on bytes, the maximum input/output correlation can be made as low as 2^{-3} and the maximum value in the EXOR table can be as low as 4 (corresponding to a difference propagation probability of 2^{-6}).

We have decided to take from the candidate constructions in [Ny94] the S-box defined by the mapping $x \mapsto x^{-1}$ in GF(2^8).

By definition, the selected mapping has a very simple algebraic expression. This enables algebraic manipulations that can be used to mount attacks such as interpolation attacks [JaKn97]. Therefore, the mapping is modified by composing it with an additional invertible affine transformation. This affine transformation does not affect the properties with respect to the first three criteria, but if properly chosen, allows the S-box to satisfy the fourth criterion.

We have chosen an affine mapping that has a very simple description per se, but a complicated algebraic expression if combined with the 'inverse' mapping. It can be seen as modular polynomial multiplication followed by an addition:

$$b(x) = (x^7 + x^6 + x^5 + 1) \cdot a(x)(x^7 + x^6 + x^5 + 1) \mod x^8 + 1$$

The modulus has been chosen as the simplest modulus possible. The multiplication polynomial has been chosen from the set of polynomials coprime to the modulus as the one with the simplest description. The constant has been chosen in such a way that the S-box has no fixed points (S-box(a) = a) and no 'opposite fixed points' (S-box(a') = a).

Note: other S-boxes can be found that satisfy the criteria above. In the case of suspicion of a trapdoor being built into the cipher, the current S-box might be replaced by another one. The cipher structure and number of rounds as defined even allow the use of an S-box that does not optimise the differential and linear cryptanalysis properties (criteria 2 and 3). Even an S-box that is "average" in this respect is likely to provide enough resistance against differential and linear cryptanalysis.


1. Clear design goals
2. Motivation for the specific solution chosen
3. A possible pitfall and how it is avoided
7.2 The ByteSub S-box

The design criteria for the S-box are inspired by differential and linear cryptanalysis on the one hand and attacks using algebraic manipulations, such as interpolation attacks, on the other:

1. Invertibility;
2. Minimisation of the largest non-trivial correlation between linear combinations of input bits and linear combination of output bits;
3. Minimisation of the largest non-trivial value in the EXOR table;
4. Complexity of its algebraic expression in GF(2^8);
5. Simplicity of description.

In [Ny94] several methods are given to construct S-boxes that satisfy the first three criteria. For invertible S-boxes operating on bytes, the maximum input/output correlation can be made as low as 2^{-3} and the maximum value in the EXOR table can be as low as 4 (corresponding to a difference propagation probability of 2^{-6}).

We have decided to take from the candidate constructions in [Ny94] the S-box defined by the mapping x ↦ x^2 in GF(2^8).

By definition, the selected mapping has a very simple algebraic expression. This enables algebraic manipulations that can be used to mount attacks such as interpolation attacks [JaKn97]. Therefore, the mapping is modified by composing it with an additional invertible affine transformation. This affine transformation does not affect the properties with respect to the first three criteria, but if properly chosen, allows the S-box to satisfy the fourth criterion.

We have chosen an affine mapping that has a very simple description per se, but a complicated algebraic expression if combined with the ‘inverse’ mapping. It can be seen as modular polynomial multiplication followed by an addition:

\[ b(x) = (x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + 1) \mod x^8 + 1 \]

The modulus has been chosen as the simplest modulus possible. The multiplication polynomial has been chosen from the set of polynomials coprime to the modulus as the one with the simplest description. The constant has been chosen in such a way that that the S-box has no fixed points (S-box(a) = a) and no ‘opposite fixed points’ (S-box(a) = \overline{a}).

Note: other S-boxes can be found that satisfy the criteria above. In the case of suspicion of a trapdoor being built into the cipher, the current S-box might be replaced by another one. The cipher structure and number of rounds as defined even allow the use of an S-box that does not optimise the differential and linear cryptanalysis properties (criteria 2 and 3). Even an S-box that is “average” in this respect is likely to provide enough resistance against differential and linear cryptanalysis.

Outline

1. Standards and S-boxes
2. On the S-box of RFC 6986 and 7801
3. The Core Issue: the S-Box Generation Process
4. Conclusion
## Kuznyechik/Streebog

### Streebog (**RFC 6986**)
- **Type**: Hash function
- **Publication**: 2012 (RFC in Aug. 2013)

### Kuznyechik (**RFC 7801**)
- **Type**: Block cipher
- **Publication**: 2015 (RFC in Mar. 2016)

### Common ground
- Both are standards in Russia.
- They were designed by the TC26 (supervised by the FSB).
- Their RFCs come from the independent stream (≠ CFRG)
- Both use the same 8-bit S-Box, \( \pi \).
## Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2012</td>
<td>GOST standardization of Streebog</td>
<td>GOST</td>
</tr>
<tr>
<td>Aug. 2013</td>
<td>RFC for Streebog (RFC 6986)</td>
<td>IETF</td>
</tr>
<tr>
<td>June 2015</td>
<td>GOST standardization of Kuznyechik</td>
<td>GOST</td>
</tr>
<tr>
<td>Mar. 2016</td>
<td>RFC for Kuznyechik (RFC 7801)</td>
<td>IETF</td>
</tr>
</tbody>
</table>
## Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2012</td>
<td>GOST standardization of Streebog</td>
<td>GOST</td>
</tr>
<tr>
<td>Aug. 2013</td>
<td>RFC for Streebog (RFC 6986)</td>
<td>IETF</td>
</tr>
<tr>
<td>June 2015</td>
<td>GOST standardization of Kuznyechik</td>
<td>GOST</td>
</tr>
<tr>
<td>Mar. 2016</td>
<td>RFC for Kuznyechik (RFC 7801)</td>
<td>IETF</td>
</tr>
<tr>
<td>May 2016</td>
<td>Publication of the first decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Biryukov, Perrin, Udovenko. Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1. <strong>EUROCRYPT'16</strong></td>
<td></td>
</tr>
<tr>
<td>Mar. 2017</td>
<td>Publication of the second decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Perrin, Udovenko. <em>Exponential S-Boxes: a Link Between the S-Boxes of BeIT and Kuznyechik/Streebog</em>. <strong>IACR ToSC 2016</strong></td>
<td></td>
</tr>
</tbody>
</table>
Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2012</td>
<td>GOST standardization of Streebog</td>
<td>GOST</td>
</tr>
<tr>
<td>Aug. 2013</td>
<td>RFC for Streebog (RFC 6986)</td>
<td>IETF</td>
</tr>
<tr>
<td>June 2015</td>
<td>GOST standardization of Kuznyechik</td>
<td>GOST</td>
</tr>
<tr>
<td>Mar. 2016</td>
<td>RFC for Kuznyechik (RFC 7801)</td>
<td>IETF</td>
</tr>
<tr>
<td>May 2016</td>
<td>Publication of the first decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Biryukov, Perrin, Udovenko. Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1. EUROCRYPT'16</td>
<td></td>
</tr>
<tr>
<td>Mar. 2017</td>
<td>Publication of the second decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td>Oct. 2018</td>
<td>ISO standardization of Streebog (ISO 10118-3)</td>
<td>ISO</td>
</tr>
<tr>
<td>Date</td>
<td>Event</td>
<td>Organization</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>July 2012</td>
<td>GOST standardization of Streebog</td>
<td>GOST</td>
</tr>
<tr>
<td>Aug. 2013</td>
<td>RFC for Streebog (RFC 6986)</td>
<td>IETF</td>
</tr>
<tr>
<td>June 2015</td>
<td>GOST standardization of Kuznyechik</td>
<td>GOST</td>
</tr>
<tr>
<td>Mar. 2016</td>
<td>RFC for Kuznyechik (RFC 7801)</td>
<td>IETF</td>
</tr>
<tr>
<td>May 2016</td>
<td>Publication of the first decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Biryukov, Perrin, Udovenko. <em>Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1</em>. EUROCRYPT'16</td>
<td></td>
</tr>
<tr>
<td>Mar. 2017</td>
<td>Publication of the second decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td>Oct. 2018</td>
<td>ISO standardization of Streebog (ISO 10118-3)</td>
<td>ISO</td>
</tr>
<tr>
<td>Jan. 2019</td>
<td>Publication of the final decomposition</td>
<td>IACR</td>
</tr>
</tbody>
</table>
# Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2012</td>
<td>GOST standardization of Streebog</td>
<td>GOST</td>
</tr>
<tr>
<td>Aug. 2013</td>
<td>RFC for Streebog (RFC 6986)</td>
<td>IETF</td>
</tr>
<tr>
<td>June 2015</td>
<td>GOST standardization of Kuznyechik</td>
<td>GOST</td>
</tr>
<tr>
<td>Mar. 2016</td>
<td>RFC for Kuznyechik (RFC 7801)</td>
<td>IETF</td>
</tr>
<tr>
<td>May 2016</td>
<td>Publication of the first decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Biryukov, Perrin, Udovenko. Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1. EUROCRYPT'16</td>
<td></td>
</tr>
<tr>
<td>Mar. 2017</td>
<td>Publication of the second decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Perrin, Udovenko. Exponential S-Boxes: a Link Between the S-Boxes of BeIT and Kuznyechik/Streebog. IACR ToSC 2016</td>
<td></td>
</tr>
<tr>
<td>Oct. 2018</td>
<td>ISO standardization of Streebog (ISO 10118-3)</td>
<td>ISO</td>
</tr>
<tr>
<td>Jan. 2019</td>
<td>Publication of the final decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Perrin. Partitions in the S-box of Streebog and Kuznyechik. IACR ToSC 2019</td>
<td></td>
</tr>
<tr>
<td>Feb. 2019</td>
<td>Kuznyechik at ISO: decision post-poned</td>
<td>ISO</td>
</tr>
</tbody>
</table>
## Timeline

<table>
<thead>
<tr>
<th>Month</th>
<th>Event</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2012</td>
<td>GOST standardization of Streebog</td>
<td>GOST</td>
</tr>
<tr>
<td>Aug. 2013</td>
<td>RFC for Streebog (RFC 6986)</td>
<td>IETF</td>
</tr>
<tr>
<td>June 2015</td>
<td>GOST standardization of Kuznyechik</td>
<td>GOST</td>
</tr>
<tr>
<td>Mar. 2016</td>
<td>RFC for Kuznyechik (RFC 7801)</td>
<td>IETF</td>
</tr>
<tr>
<td>May 2016</td>
<td>Publication of the first decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Biryukov, Perrin, Udovenko. <em>Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1.</em> EUROCRYPT'16</td>
<td></td>
</tr>
<tr>
<td>Mar. 2017</td>
<td>Publication of the second decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td>Oct. 2018</td>
<td>ISO standardization of Streebog (ISO 10118-3)</td>
<td>ISO</td>
</tr>
<tr>
<td>Jan. 2019</td>
<td>Publication of the final decomposition</td>
<td>IACR</td>
</tr>
<tr>
<td></td>
<td>Perrin. <em>Partitions in the S-box of Streebog and Kuznyechik.</em> IACR ToSC 2019</td>
<td></td>
</tr>
<tr>
<td>Feb. 2019</td>
<td>Kuznyechik at ISO: decision post-poned</td>
<td>ISO</td>
</tr>
<tr>
<td>Sep. 2019</td>
<td>Kuznyechik at ISO: decision must be taken!</td>
<td>ISO</td>
</tr>
</tbody>
</table>
Outline

1. Standards and S-boxes
2. On the S-box of RFC 6986 and 7801
3. The Core Issue: the S-Box Generation Process
4. Conclusion
The Russian S-box


Screen capture of the specification of Kuznyechik (2015).
How Was it Generated?

According to the designers (April 2018)

questioned is the S-box \( \pi \). 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 \( \pi \) was obtained by pseudo-random search and the following properties were taken into account.

\[
\pi_{i,j} = (2^m j + 1) \cdot s(2^m i), \quad \text{for } 0 < i < 0, j < 2^m - 1
\]

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).

- Source: https://cdn.virgilsecurity.com/assets/docs/memo-on-kuznyechik-s-box.pdf
- See also the discussion summary: https://cdn.virgilsecurity.com/assets/docs/meeting-report-for-the-discussion-on-kuznyechik-and-streebog.pdf
How Was it Generated?

According to the designers (April 2018)

questioned is the S-box $\pi$. 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 $\pi$ 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).

Source: https://cdn.virgilsecurity.com/assets/docs/memo-on-kuznyechik-s-box.pdf
See also the discussion summary: https://cdn.virgilsecurity.com/assets/docs/meeting-report-for-the-discussion-on-kuznyechik-and-streebog.pdf

What I proved (IACR ToSC 2019)

$$\pi \left\{ \begin{array}{c}
\mathbb{F}_2^8 \\
0 \\
(\alpha^{2m+1})^j \\
(\alpha^i + (2^m+1)^j)
\end{array} \right\} \rightarrow \mathbb{F}_2^8$$

\[ \leftrightarrow \kappa(0), \]
\[ \leftrightarrow \kappa(2^m - j), \text{ for } 1 \leq j \leq 2^m - 1, \]
\[ \leftrightarrow \kappa(2^m - i) \oplus (\alpha^{2m+1})^{s(j)}, \text{ for } 0 < i, 0 \leq j < 2^m - 1. \]
Such a Structure is Beyond Unlikely

Lemma (more details available online\(^1\))

\[
\text{There are } 256! \approx 2^{1684} \text{ different 8-bit permutations, meaning you need at least } 1684 \text{ bits to represent all of them in any language.}
\]

---

2. Credit to @odzhan on stackexchange.
3. [https://codegolf.stackexchange.com/questions/186498/proving-that-a-russian-cryptographic-standard-is-too-structured](https://codegolf.stackexchange.com/questions/186498/proving-that-a-russian-cryptographic-standard-is-too-structured)
Such a Structure is Beyond Unlikely

Lemma (more details available online)

There are $256! \approx 2^{1684}$ different 8-bit permutations, meaning you need at least 1684 bits to represent all of them in any language.

- 165 ASCII characters that fit on 7 bits: this program is 1155-bit long

---

2 Credit to @odzhan on stackexchange.
3 https://codegolf.stackexchange.com/questions/186498/proving-that-a-russian-cryptographic-standard-is-too-structured
Such a Structure is Beyond Unlikely

**Lemma (more details available online¹)**

There are $256! \approx 2^{1684}$ different 8-bit permutations, meaning you need at least 1684 bits to represent all of them in any language.

- 165 ASCII characters that fit on 7 bits: this program is 1155-bit long
- An AMD64 binary implementation fits² on 78 bytes, i.e. 624 bits.
- Many more short implementations have been found by code golfers!³

---

² Credit to @odzhan on stackexchange.
³ https://codegolf.stackexchange.com/questions/186498/proving-that-a-russian-cryptographic-standard-is-too-structured
Such a Structure is Beyond Unlikely

Lemma (more details available online\(^1\))

There are 256! \(\approx 2^{1684}\) different 8-bit permutations, meaning you need at least 1684 bits to represent all of them in any language.

\[
p(x)\{\text{unsigned char \textit{k}="@\~rFTDVbpPB}
vdtfR@\textit{x}acp?\textit{x}e2>4\textit{x}a6\textit{x}e9\{\textit{z}\textit{x}e3q}
5\textit{\text{'x}a7\textit{x}e8"}, \textit{a}=2, \textit{l}=0, \textit{b}=17; \textbf{while}(x&&
(l++, a^x)) a=2*a^a/128*29; \textbf{return} l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;\}
\]

- 165 ASCII characters that fit on 7 bits: this program is 1155-bit long
- An AMD64 binary implementation fits\(^2\) on 78 bytes, i.e. 624 bits.
- Many more short implementations have been found by code golfers!\(^3\)

The probability that a random S-box is that simple is completely negligible \((\leq 2^{-1059})\).

---


\(^2\) Credit to @odzhan on stackexchange.

\(^3\) https://codegolf.stackexchange.com/questions/186498/proving-that-a-russian-cryptographic-standard-is-too-structured
Outline

1. Standards and S-boxes
2. On the S-box of RFC 6986 and 7801
3. The Core Issue: the S-Box Generation Process
4. Conclusion
Conclusion

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).

- This claim and this fact **cannot** be reconciled.
Conclusion

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).

■ This claim and this fact cannot be reconciled.

■ In my opinion, the designers of these algorithms have provided misleading information for the external analysis of their design.
Conclusion

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).

- This claim and this fact **cannot** be reconciled.

- In my opinion, the designers of these algorithms **have provided misleading information** for the external analysis of their design.

- Security analysis is hard enough with proper information: **there is no good reason** to complicate it further with wrong data!
Conclusion

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).

- This claim and this fact cannot be reconciled.
- In my opinion, the designers of these algorithms have provided misleading information for the external analysis of their design.
- Security analysis is hard enough with proper information: there is no good reason to complicate it further with wrong data!

\[ p(x)\{\text{unsigned char }*k"=\text{'0'}\text{FTDVbpPBvdfR0\text{\xacp?\xe2\xe4\xa6\xe9\z\xe3q5\xa7\xe8", a=2, l=0, b=17; \text{while}(x\&\&}\}\text{(l+=a\^x))a=2*a/a/128*29; \text{return l}\{\text{z\xb}\}^{k[b+l/b]}\b:k[l/b]\}^{188};} \]

\[ \Rightarrow \text{These algorithms cannot be trusted and I believe they should be deprecated.} \]
Conclusion

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).

- This claim and this fact cannot be reconciled.
- In my opinion, the designers of these algorithms have provided misleading information for the external analysis of their design.
- Security analysis is hard enough with proper information: there is no good reason to complicate it further with wrong data!

⇒ These algorithms cannot be trusted and I believe they should be deprecated.

Thank you!
$F_{2^8}$

$\pi(F_{2^8}) = F_{2^8}$
Cosets to Cosets

\[ \mathbb{F}_{2^8} \]

\[ n(\mathbb{F}_{2^8}) = \mathbb{F}_{2^8} \]

\[ \{0\} \quad \mathbb{F}_{2^4}^* \]

\[ \kappa(0) \oplus \mathbb{F}_{2^4}^* \quad \{fc\} \]
Cosets to Cosets

\[ \mathbb{F}^{2^8} \]

\[ \pi(\mathbb{F}^{2^8}) = \mathbb{F}^{2^8} \]

\[ \{0\} \oplus \mathbb{F}^{*}_{2^4} \]

\[ \kappa(0) \oplus \mathbb{F}^{*}_{2^4} \]

\[ \mathbb{F}^{*}_{2^4} \]

\[ \mathbb{F}^{*}_{2^4} \]

\[ \alpha^{16} \odot \mathbb{F}^{*}_{2^4} \]
Cosets to Cosets

$$\mathbb{F}_{2^8}$$

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

\[ \kappa(15) \oplus \mathbb{F}_{2^4}^* \]
\[ \kappa(14) \oplus \mathbb{F}_{2^4}^* \]
\[ \kappa(0) \oplus \mathbb{F}_{2^4}^* \]
Cosets to Cosets

\[
\mathbb{F}^*_2^{28} \quad \quad n(\mathbb{F}^*_2^{28}) = \mathbb{F}^*_2^{28}
\]
Why it is Worrying

Russian S-box

Backdoored S-box

(https://ia.cr/2016/493)