Streebog compression function as PRF in secret-key settings

Vitaly Kiryukhin

JSC «InfoTeCS», LLC «SFB Lab», Moscow, Russia
Vitaly.Kiryukhin@infotecs.ru

Abstract

Security of the many keyed hash-based cryptographic constructions (such as HMAC) depends on the fact that the underlying compression function $\mathsf{g}(H, M)$ is a pseudorandom function (PRF). This paper presents key-recovery algorithms for 7 rounds (of 12) of Streebog compression function. Two cases were considered, as a secret key can be used: the previous state H or the message block M. The proposed methods implicitly show that Streebog compression function has a large security margin as PRF in the above-mentioned secret-key settings.

Keywords: Streebog, PRF, truncated differentials, rebound, polytopic cryptanalysis

1 Introduction

Hash function is one of the most commonly used cryptographic primitives. Usually, the following three security properties are expected from a non-keyed hash function:

1) preimage resistance (for a given value $\mathsf{Hash}(Msg)$ it is hard to obtain Msg);

2) second preimage resistance (for a given message Msg it is difficult to find a different Msg' such that $\mathsf{Hash}(Msg) = \mathsf{Hash}(Msg')$);

3) collision resistance (it is hard to construct a nontrivial message pair (Msg, Msg') such that $\mathsf{Hash}(Msg) = \mathsf{Hash}(Msg')$).

For hash functions based on the Merkle-Damgård scheme [3, 2], similar requirements are imposed on the underlying compression function $\mathsf{g}(H_{prev}, M) = H_{next}$ (where M is a fixed-length block of the hashed message, H_{prev} and H_{next} are the previous and the next internal states respectively).

Russian hash function Streebog [1], like many others, uses slightly modified Merkle-Damgård approach. Its compression function is based on a 12-rounds AES-like [22] block cipher in Miyaguchi-Preneel mode. The previous

internal state is transformed to 13 round keys for the block cipher. The internal state consists of 8×8 bytes ($n = 512$ bits). The output length of hash function can be either 512 or 256-bit.

Over recent years, Streebog (as well as its compression function and block cipher) was subjected to a thorough analysis by many experts. We cite papers devoted to the preimage [11, 12, 16, 9], the second preimage [7], various types of collisions [10, 11, 12, 13, 14]. Many articles describe so-called «known-key» (and «chosen-key») distinguishers [8, 13, 12, 17, 18] demonstrating some non-random structural properties of the transformation (a compression function or a block cipher) by constructing the corresponding set of input-output pairs.

Keyless hash function is often used as part of the secret-key cryptoalgorithms. Some of the most well-known examples are HMAC and NMAC [6]. The security of such algorithms depends significantly on the fact that the compression function is a PRF. Let one of the arguments $\mathtt{g}(H, M)$ be a secret key and an adversary can adaptively choose blocks for the other input and observe outputs. It is clear that a simple key guessing with time-complexity about $t = 2^n$ can be used to distinguish between $\mathtt{g}(H, M)$ and a random function. In some cases, straightforward birthday-paradox distinguisher with data-complexity $q = 2^{n/2}$ can also be mounted. Is it possible to construct more efficient algorithms for a specific instance of $\mathtt{g}(H, M)$? In our paper we consider round-reduced Streebog compression function.

To the best of our knowledge, there is only one paper [15] on the subject[1]. The authors [15] utilize impossible differential properties to mount secret-state (secret-IV) recovery attacks on 6.75-rounds.

Next, we present key-recovery algorithms for 7-round Streebog compression function.

In section 3 we describe algorithm for the secret-state case. The proposed method is based on polytopic approach [5]. A naive algorithm for «generalized birthday problem» [23] is also an important part of the method.

In section 4 the second secret-message case is considered. The rebound technique [25] is used to obtain usable pairs of non-secret states. The truncated differential [20] method is then applied to recover the secret message.

Comparative characteristics of algorithms are presented in table 1. Note also that the initial data processing was not taken into account when calculating the complexity of attacks [15] (so $t < q$, «Time» is less than «Data»).

[1]For completeness, it is worth noting that key-recovery attack on HMAC-Streebog was presented in [24] as the extension of the generic state-recovery attack on HMAC with an arbitrary Merkle-Damgård hash-function. Data-complexity of attack [24] is significantly more than HMAC allowable «provable secure» bounds [6]. The attack also does not depend on the properties of the compression function.

Our results provide an additional argument showing that Streebog compression function (as a PRF) has a significant security margin.

Setting	Rounds	Time	Memory	Data	Description
secret H	6.75	$2^{399.5}$	2^{349}	2^{483}	[15]
	6.75	$2^{261.5}$	2^{205}	$2^{495.5}$	[15]
	7	2^{421}	2^{354}	2^{64}	Section 3
	12	2^{256}	2^{256}	2^{256}	birthday-paradox distinguisher
	12	2^{512}	$\sim$	2	key guessing
secret M	7	2^{240}	2^{20}	2^{113}	Section 4
	12	2^{512}	$\sim$	2	key guessing

Table 1: Attacks on Streebog compression functions in secret-key settings. «Time» (t) in g computations, «Memory» in n-bit blocks, «Data» (q) in chosen message-output pairs.

2 Definitions

Let $\mathbb{F}_{2^8}$ be a finite field. Each element of $\mathbb{F}_{2^8}$ can be interpreted as an integer or binary vector. Denote $v \times v$ matrix space over $\mathbb{F}_{2^8}$ by $\mathbb{F}_{2^8}^{v\times v}$ (we also use symbol $\mathbb{F}_{2^8}^{v}$ as a vector space). Elements from $\mathbb{F}_{2^8}^{v\times v}$ will be denoted by capital letters: A, B. Blocks of states and messages also belong to $\mathbb{F}_{2^8}^{v\times v}$. Elements of a matrix are indexed by $0 \le i, j \le v-1$ (for example, $a = A[0,0]$ is an element from the upper-left corner of the matrix). $A[i,\cdot]$ is i-th row of A, $A[\cdot, j]$ is j-th column of A.

Denote bitwise xor operation by symbol $\oplus$. This operation is defined naturally for all the objects under consideration.

Let us have a sequence of blocks

$$B_0, ..., B_d \in \mathbb{F}_{2^8}^{v\times v}, d \in \mathbb{N},$$

then we refer to $\boldsymbol{\Delta B} = B_0 \oplus B_1 \in \mathbb{F}_{2^8}^{v\times v}$ as a difference and to a sequence

$$\boldsymbol{\delta B} = (B_0 \oplus B_1, B_0 \oplus B_2, ..., B_0 \oplus B_d) \in \left(\mathbb{F}_{2^8}^{v\times v}\right)^d \tag{1}$$

as a d-difference. Differences are indicated in bold text: $\boldsymbol{\delta M}$, $\boldsymbol{\Delta K_4}$.

The d-difference $\boldsymbol{\delta B} \in \left(\mathbb{F}_{2^8}^{v\times v}\right)^d$ can also be interpreted as $v\times v$ «columns» of d bytes each: $\boldsymbol{\delta B} \in \left(\mathbb{F}_{2^8}^{d}\right)^{v\times v}$, $\boldsymbol{\delta B[i,j]} \in \mathbb{F}_{2^8}^{d}$. If $\boldsymbol{\Delta B[i,j]} \neq 0$ (resp. $\boldsymbol{\delta B[i,j]} \neq \mathbf{0}$) then we say that the position (i,j) is active, otherwise inactive, $0 \le i, j \le v-1$.

The differential (resp. polytopic) trail is the sequence of the differences (resp. d-differences) after each transformation in the cipher.

The transformations over $\mathbb{F}_{2^8}^{v\times v}$ (also over $\mathbb{F}_{2^8}^{v}$ and $\mathbb{F}_{2^8}$) are denoted by sans serif font: f, S, L. The notation LS indicates a composition of transformations,

where S applies first (the reverse order «left-to-right» is used on the figures). The inverse transformations are specified by f^{-1}.

Streebog

Streebog compression function $\mathsf{g}_N(H, M)$ employs AES-like XSPL-cipher E in the Miyaguchi-Preenel mode

$$\mathsf{g}_N(H, M) = \mathsf{E}(H \oplus N, M) \oplus H \oplus M = R, \text{ where}$$

$H \in \mathbb{F}_{2^8}^{v \times v}$ is the previous state of the hash function;
$M \in \mathbb{F}_{2^8}^{v \times v}$ is the message block;
$N \in \mathbb{F}_{2^8}^{v \times v}$ is the number of previously hashed bits;
$R \in \mathbb{F}_{2^8}^{v \times v}$ is the output (the next state of hash function).

The block cipher E consists of 12 rounds and a post-whitening key addition. Each round consists of four operations:

X – modulo 2 addition of an input block with a round key;

S – parallel application of the fixed bijective substitution s to each byte of the state;

P – transposition of the state;

L – parallel application of the linear transformation l to each row of the state. In [22], it was shown that l-transformation can be represented as the MDS matrix $\mathbb{L}$ over $\mathbb{F}_{2^8}^{8 \times 8}$.

The block cipher formula is

$$\mathsf{E}(K, M) = \mathsf{X}[K_{13}]\mathsf{LPSX}[K_{12}] \ldots \mathsf{LPSX}[K_2]\mathsf{LPSX}[K_1](M).$$

The state size consists of $n = 512$ bits ($v \times v = 8 \times 8$ bytes).

The key schedule uses round constants $C_i \in \mathbb{F}_{2^8}^{v \times v}$, $i = 1, 2, \ldots, 12$, and round keys $K_i \in \mathbb{F}_{2^8}^{v \times v}$, $i = 1, 2, \ldots, 13$ are derived from a master key K_0 as follows:

$$K_0 = H \oplus N,\ K_1 = \mathsf{LPS}(H \oplus N),\ K_{i+1} = \mathsf{LPS}(K_i \oplus C_i),\ i = 1, 2, \ldots, 12.$$

We also denote the intermediate states before X, S, P, L transformations in i-th round as X_i, Y_i, Z_i, W_i correspondingly ($X_1 = M$, $Y_1 = M \oplus K_1$, $Z_1 = \mathsf{S}(Y_1)$, $W_1 = \mathsf{P}(Z_1)$, etc.). The states in the key schedule are denoted in a similar way $HX_i = K_i$, HY_i, HZ_i, HW_i, where $H = HX_0$, $HX_1 = \mathsf{LPS}(H \oplus N)$ etc.

We define an r-round compression function with $r + 1$ round keys as:

$$\mathsf{g}(H, M) = (\mathsf{X}[K_{r+1}]\mathsf{LPSX}[K_r] \ldots \mathsf{LPSX}[K_1](M)) \oplus H \oplus M.$$

Next, we also assume that N is an arbitrary constant C_0.

3 State as a secret key

Let the state H be a secret. An adversary knows a message M and an output R.

$$\mathsf{g}(H, M) = \mathsf{E}(H, M) \oplus H \oplus M = R.$$

Hence, the analysis is reduced to the block cipher

$$\mathsf{E}(H, M) \oplus H = R \oplus M = \widetilde{R},$$

$$\mathsf{E}(H, M) \oplus H = \mathsf{X}[K_{r+1} \oplus H]\mathsf{LPSX}[K_r] \ldots \mathsf{LPSX}[K_1](M),$$

where the last round key is $\widetilde{K}_{r+1} = K_{r+1} \oplus H$.

A secure block cipher can be used as a secure PRF up to about $q = 2^{n/2}$ queries [19]. Thus, any algorithm that requires more message-output pairs can't be considered as a direct threat to a PRF. The generic limit of the time complexity $t = 2^n$ is defined by straightforward key guessing.

We propose the polytopic (multidimensional differential) based key-recovery algorithm against 7-rounds. The method consists of the following steps:

1. Choose structure of 2^{64} messages M;

2. Guess 64 bits of the first key K_1. Partially encrypt all messages up to the second L-transformation;

3. Choose about 2^7 blocks (of 2^{64}) and obtain d-difference $\boldsymbol{\delta W_2}$ with only one active S-box;

4. Propagate $\boldsymbol{\delta W_2}$ forward to $\boldsymbol{\delta W_5[0,0]}$ by guessing 136 bits of the intermediate states;

5. Propagate $\boldsymbol{\delta \widetilde{R}}$ backward to $\boldsymbol{\delta X_6[0,0]}$ by guessing 72 bits of the intermediate states (similarly and independently for $\boldsymbol{\delta X_6[0,1]}$,...,$\boldsymbol{\delta X_6[0,7]}$);

6. Check by using a naive algorithm for «generalized birthday problem» that $\boldsymbol{\delta W_5[0,0]}$ can be obtained via inverse linear transformation $\mathsf{l}^{-1}(\boldsymbol{\delta X_6[0,0]}, \ldots, \boldsymbol{\delta X_6[0,7]})$;

7. If the check failed in the previous step then go back to step 2 and try another bits of K_1. If the check is passed then the key bits and the state bits are guessed correctly.

Let's look at the steps in more detail.

The first and second steps are designed to bypass the first round (figure 1). We use the structure of 2^{64} messages. One column in each message takes all possible values ($M[\cdot, 0]$ in the picture). All other seven columns are arbitrary constants ($M[\cdot, 1]$, ..., $M[\cdot, 7]$ in the picture). For any values of K_1 and K_2, this will also be true for the columns in $W_2 = \mathsf{PSX}[K_2]\mathsf{LPSX}[K_1](M)$.

Guess column $K_1[\cdot, 0]$ and compute row $K_2[0, \cdot]$. In this case, all the values in column $W_2[\cdot, 0]$ are exactly known.

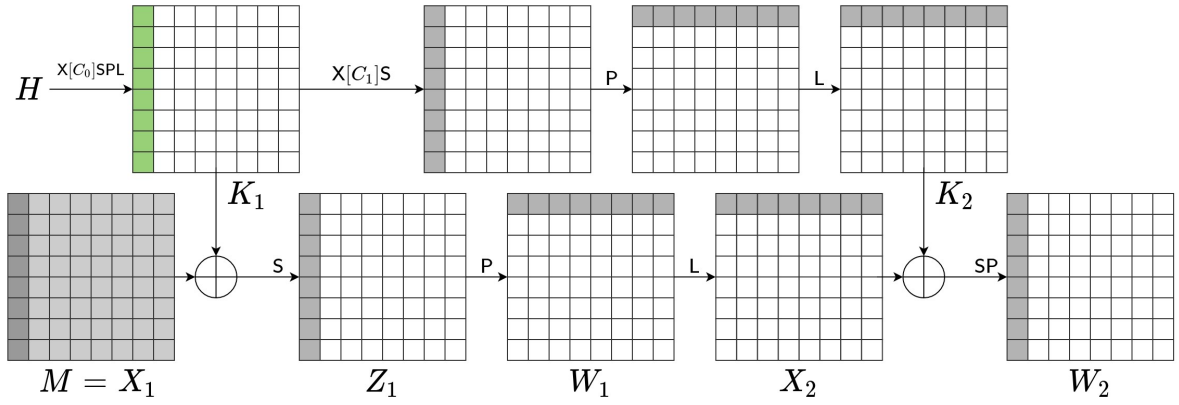

Figure 1: Steps 1-2. Guessed bits are highlighted with green. Computed or known values are denoted by gray cells. The formulas are given in reverse (left-to-right) notation.

Recall that for each of the 2^{64} states W_2 in the structure, columns $W_2[\cdot, 1]$, ..., $W_2[\cdot, 7]$ are unknown constants. It is easy to find such 2^7 states $W_2^{(0)}$,$W_2^{(1)}$,...,$W_2^{(d)}$, $d = 2^7 - 1$ that d-difference

$$\boldsymbol{\delta W_2} = (W_2^{(0)} \oplus W_2^{(1)}, W_2^{(0)} \oplus W_2^{(2)}, ..., W_2^{(0)} \oplus W_2^{(d)}) \in \left(\mathbb{F}_{2^8}^{v \times v}\right)^d$$

will have only one active byte $\boldsymbol{\delta W_2[0, 0]}$. In other words, we select states W_2 so that the bytes $W_2[1, 0]$,...,$W_2[7, 0]$ are also constants. We choose the corresponding outputs $\widetilde{R}$ and compute d-difference $\boldsymbol{\delta \widetilde{R}}$.

The difference (and d-difference) is unambiguously propagated through X, L and P transformations, but we have to guess the state bytes to propagate the difference through S. Obviously, zero difference remains the same after any transformation.

If the bytes from $K_1[\cdot, 0]$ are guessed correctly, then the trail from $\boldsymbol{\delta W_2}$ to $\boldsymbol{\delta \widetilde{R}}$ must exist. Otherwise, it's possible to check that there are no appropriate trails (or almost none).

The d-difference $\boldsymbol{\delta W_2}$ propagates through L and $\mathsf{X}[K_3]$ to $\boldsymbol{\delta Y_3}$, which contains eight active bytes $\boldsymbol{\delta Y_3[0, \cdot]}$ (see figure 2). Recall that this is true due to the MDS property of L [22]. We guess $Y_3[0, \cdot]$ and obtain 2^{64} possible d-differences $\boldsymbol{\delta Z_3}$. Next, $\boldsymbol{\delta Y_4}$ is computed by linear propagation through P, L, $\mathsf{X}[K_4]$. All byte positions in $\boldsymbol{\delta Y_4}$ are active.

By guessing only one column $Y_4[\cdot, 0]$ we obtain 2^{128} possible d-differences $\boldsymbol{\delta Z_4[\cdot, 0]}$. The remaining seven columns in $\boldsymbol{\delta Z_4}$ are active but unknown to us.

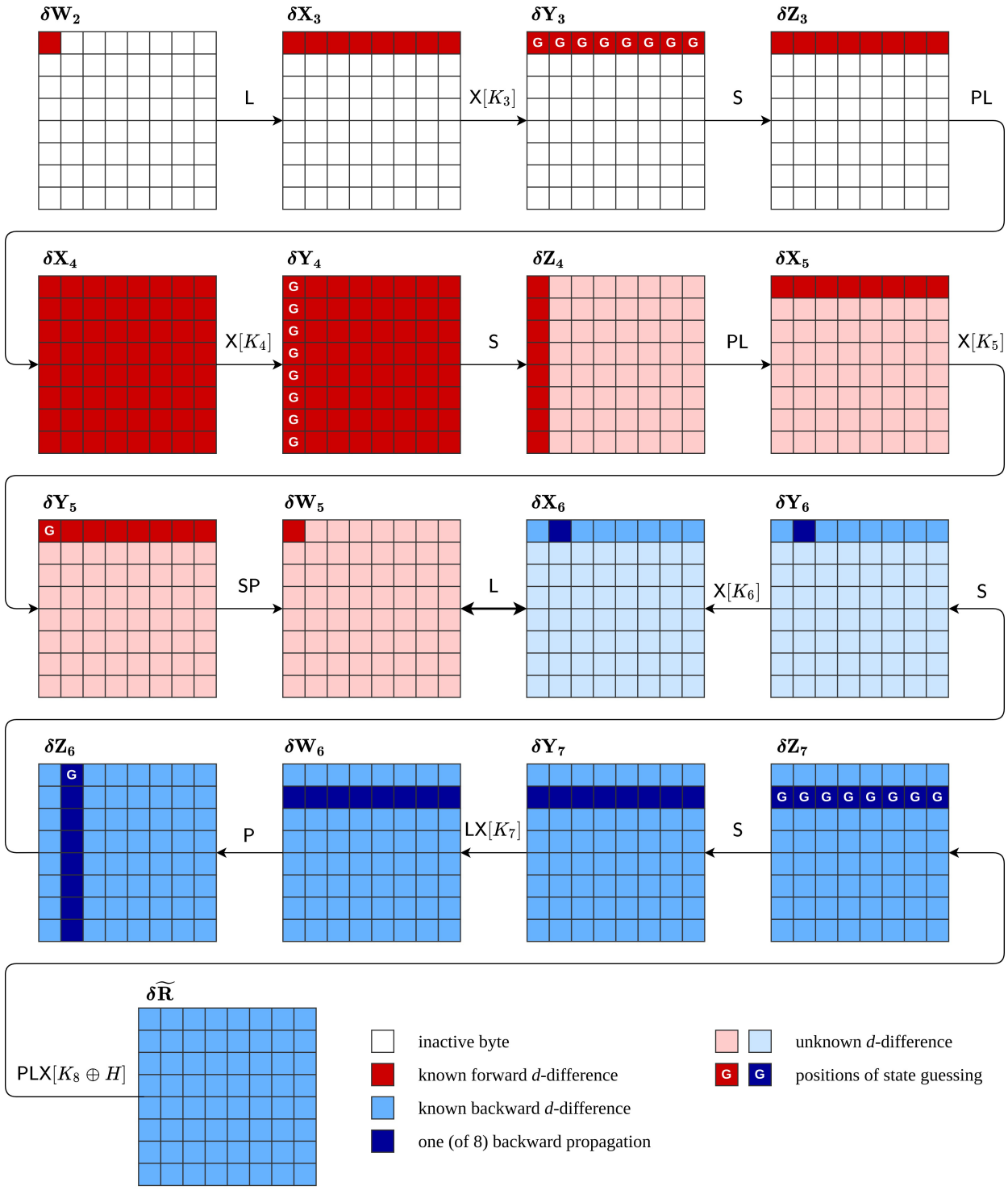

Figure 2: Steps 4-5. Forward and backward d-difference propagations.

The d-difference $\boldsymbol{\delta Y_5}[\mathbf{0}, \cdot]$ is calculated in the same way for each $\boldsymbol{\delta Z_4}[\cdot, \mathbf{0}]$. Another byte $Y_5[0, 0]$ allows us to compute $\boldsymbol{\delta Z_5}[\mathbf{0}, \mathbf{0}] \in \mathbb{F}_{2^8}^d$ and $\boldsymbol{\delta W_5}[\mathbf{0}, \mathbf{0}] = \boldsymbol{\delta Z_5}[\mathbf{0}, \mathbf{0}]$.

Thus, we have $2^{64} \cdot 2^{64} \cdot 2^8 = 2^{136}$ values of $\boldsymbol{\delta W_5}[\mathbf{0}, \mathbf{0}]$, stored in the array

$\mathcal{L}_{\text{frw}}$. Each d-difference corresponds to the sequence of bytes

$$Y_3[0,0], Y_3[0,1], \ldots, Y_3[0,7],\ Y_4[0,0], Y_4[1,0], \ldots, Y_4[7,0],\ Y_5[0,0].$$

Consider the backward direction. We know d-difference $\boldsymbol{\delta}\widetilde{\boldsymbol{R}}$ and can compute $\boldsymbol{\delta Z_7}$ by backward propagation through $\mathsf{X}[K_8 \oplus H]$, L^{-1}, P^{-1}.

Guess one row of Z_7 (bytes $Z_7[1,0]$,...,$Z_7[1,7]$ on figure 2). We obtain 2^{64} values of corresponding column in $\boldsymbol{\delta Z}_6$. Guess one byte in Z_6 (byte $Z[0,1]$ on figure). Hence, we can compute 2^{72} possible values of $\boldsymbol{\delta Y_6[0,1]}$ and $\boldsymbol{\delta X_6[0,1] = \delta Y_6[0,1]}$.

Similar actions are performed in parallel for the other seven rows in $\boldsymbol{\delta Z_7}$. As a result, we computed values of $\boldsymbol{\delta X_6[0,0]}$, $\boldsymbol{\delta X_6[0,1]}$, ..., $\boldsymbol{\delta X_6[0,7]}$. Eight lists $\mathcal{L}_0$, $\mathcal{L}_1$, ...,$\mathcal{L}_7$ of 2^{72} values (d-difference) each were stored.

Hypothetically, all $(2^{72})^8 = 2^{576}$ values of $\boldsymbol{\delta X_6[0,\cdot]}$ can be computed, and therefore, $\boldsymbol{\delta W_5[0,\cdot]} = \mathsf{l}^{-1}(\boldsymbol{\delta X_6[0,\cdot]})$. Next, each variant of $\boldsymbol{\delta W_5[0,0]}$ can be checked by searching among previously computed ones in the forward direction. Obviously, this way is much expensive.

Let's rewrite the expression for the inverse linear transformation

$$W_5[0,\cdot] \times \mathbb{L} = X_6[0,\cdot],$$

$$W_5[0,\cdot] = X_6[0,\cdot] \times \mathbb{L}^{-1},$$

$$W_5[0,0] = c_0 \cdot X_6[0,0] \oplus c_1 \cdot X_6[0,1] \oplus \ldots \oplus c_7 \cdot X_6[0,7],$$

where: $\mathbb{L} \in \mathbb{F}_{2^8}^{8\times 8}$ (resp. $\mathbb{L}^{-1}$) is the MDS matrix of the linear transformation l (resp. the inverse transformation l^{-1}); $c_0, c_1, \ldots, c_7 \in \mathbb{F}_{2^8}$ are the coefficients from the column of $\mathbb{L}^{-1}$. The matrix representation from [22] is implicitly used here, but the expressions can be rewritten for the binary 64×64 matrix.

The same equality is also true for the correct pairs of the differences (d-differences)

$$\boldsymbol{\delta W_5[0,0]} = c_0 \cdot \boldsymbol{\delta X_6[0,0]} \oplus c_1 \cdot \boldsymbol{\delta X_6[0,1]} \oplus \ldots \oplus c_7 \cdot \boldsymbol{\delta X_6[0,7]},$$

where $c_i \cdot \boldsymbol{\delta X_6[0,i]} = (c_i \cdot \boldsymbol{\Delta x_1}, c_i \cdot \boldsymbol{\Delta x_2}, \ldots, c_i \cdot \boldsymbol{\Delta x_d})$, $\boldsymbol{\Delta x_j} \in \mathbb{F}_{2^8}$, $i = 0, ..., 7$, $j = 1, ..., d$, $\boldsymbol{\delta X_6[0,i]} \in \mathbb{F}_{2^8}^d$. Therefore, we can proceed to the simpler problem

$$\mathcal{L}_{\text{frw}}[p_{\text{frw}}] = c_0 \cdot \mathcal{L}_0[p_0] \oplus c_1 \cdot \mathcal{L}_1[p_1] \oplus \ldots \oplus c_7 \cdot \mathcal{L}_7[p_7], \tag{2}$$

we should find the indexes p_0, p_1, ..., p_7, p_{frw} so that the equation is correct, or prove that there are no such indexes. We obtain some example of a generalized birthday problem [23], but we have no task to find at least some «collision».

Our goal is to build only one unique correct solution. All others should be discarded. Because of this, we apply a naive approach.

Rearrange the components of the equation

$$\mathcal{L}_{\text{frw}}[p_{\text{frw}}] \oplus c_0 \cdot \mathcal{L}_0[p_0] \oplus c_1 \cdot \mathcal{L}_1[p_1] \oplus c_2 \cdot \mathcal{L}_2[p_2] = c_3 \cdot \mathcal{L}_3[p_3] \oplus \ldots \oplus c_7 \cdot \mathcal{L}_7[p_7]. \quad (3)$$

Combine all lists from the left side (3) into one. We obtain an array $\mathcal{L}_{\text{left}}$ containing $2^{136} \cdot \left(2^{72}\right)^3 = 2^{352}$ elements (d-differences) from $\mathbb{F}_{2^8}^d$. The hash table is used to store items. The d-difference is the «key», the guessed state bits are the «value». Hence, each item requires $(8 \cdot d + 352) < 3n$ bits of memory to be stored (in total, less than 2^{354} n-bit states).

It's not hard to see, that the right side (3) generates $(2^{72})^5 = 2^{360}$ items ($\mathcal{L}_{\text{right}}$) that can be constructed dynamically by iterating through 360 bits.

If the arbitrary element from $\mathcal{L}_{\text{right}}$ is found in $\mathcal{L}_{\text{left}}$, then we assume that the trail from $\boldsymbol{\delta W_2}$ to $\boldsymbol{\delta \widetilde{R}}$ exists and all the bits ($K_1[\cdot, 0]$, $Y_3[0, \cdot]$, $Y_4[\cdot, 0]$, $Y_5[0, 0]$, $Z_6[0, \cdot]$, Z_7) are guessed correctly. What is the average number of false assumptions? We have $2^{360+352} = 2^{712}$ pairs of d-differences ($8d$-bit values). Thus, under the hypothesis of a random and uniform distribution, we get $2^{64} \cdot 2^{352} \cdot 2^{360} \cdot 2^{-d \cdot 8} = 2^{-240} \approx 0$ false solutions (the factor 2^{64} emerges due to the key guessing at step 2, the probability of two random d-byte vectors matching is $2^{-d \cdot 8}$). The value of d can be reduced, but this does not significantly affect the estimation of the time complexity.

If no element from $\mathcal{L}_{\text{right}}$ is found in $\mathcal{L}_{\text{left}}$ then we guess the next value of $K_1[\cdot, 0]$. Steps 3-6 are repeated again.

The last round key $\widetilde{K}_8 = K_8 \oplus H$ is computed via the known state Z_7 and the corresponding output $\widetilde{R}$

$$\widetilde{K}_8 = \widetilde{R} \oplus \mathsf{LP}(Z_7).$$

In this way, the challenge is reduced to six rounds.

There is a different approach. The bytes of the other seven rows in Z_6 are determined by parallel guessing of $(Y_5[0, 1], Z_6[1, \cdot])$, $(Y_5[0, 2], Z_6[2, \cdot])$, ..., $(Y_5[0, 7], Z_6[7, \cdot])$. The correct values are obtained via similar check of the trail from $\boldsymbol{\delta X_6[i, \cdot]}$ to $\boldsymbol{\delta W_5[i, 0]}$ through inverse linear transformation l^{-1}, $i = 1, 2, \ldots 7$. Next, we use simple relation $Z_7 = \mathsf{S}(K_7 \oplus \mathsf{LP}(Z_6))$ and recover the round key

$$K_7 = \mathsf{S}^{-1}(Z_7) \oplus \mathsf{LP}(Z_6).$$

The secret H is computed due to the invertibility of the key schedule.

By the end, the time complexity of the key-recovery algorithm is

$$t = \underbrace{2^{64}}_{K_1[\cdot,0]} \cdot \left(\underbrace{16 \cdot 2^{64}}_{\text{step 2}} + \underbrace{d' \cdot 2^{136}}_{\text{step 4}} + \underbrace{d' \cdot 8 \cdot 2^{72}}_{\text{step 5}} + \underbrace{d' \cdot 2^{352}}_{\mathcal{L}_{\text{left}}} + \underbrace{d' \cdot 2^{360}}_{\mathcal{L}_{\text{right}}} + \underbrace{7 \cdot d' \cdot 2^{72}}_{Z_6 \text{ recovery}} \right),$$

where $d' = d+1 = 2^7$. In total, $t \approx 2^{431}$ Sbox computations. We estimate the computation complexity of the 7-round compression function as $2 \cdot 7 \cdot 64 \approx 2^{10}$ Sbox computations (memory access operations). As a result, we get time complexity $t = 2^{431} \cdot 2^{-10} = 2^{421}$. The proposed method requires less than 2^{354} (n-bit states) of memory. The data complexity is 2^{64} chosen pairs (M, R).

The described algorithm is deterministic – the probability of success is equal to one. Meanwhile, the most effective method [21] against 7-round AES-128 uses a rare event (truncated differential).

Note also that the ideas of the proposed method can be applied to 6 rounds of AES-128 (similar to steps 3-7 above). We were able to build an attack with time complexity about 2^{120} memory access operations and a small amount of the chosen plaintexts $q = d + 1 < 2^5$. Due to the relatively high time complexity, we were unable to extend the attack to 7 rounds (as in steps 1-2).

4 Message as a secret key

Let the message M be a secret

$$\mathsf{g}(H, M) = \mathsf{E}(H, M) \oplus H \oplus M = R.$$

An adversary has a full control over the master-key H and the round keys of the underlying block cipher

$$\mathsf{E}(H, M) \oplus M = R \oplus H = \widetilde{R}.$$

The function $\mathsf{E}(H, M) \oplus M$ with secret M is a secure PRF in the ideal cipher model (i.e. if E is a family of random permutations). The proof can be found, for example, in [4, Theorem 8.5]. In this case, there is no simple birthday-paradox distinguisher. Only brute-force key search is applicable attack.

Consider the algorithm against seven rounds, which consists of two stages.

«Offline» stage. Following the rebound approach [25], about 2^{112} pairs (H, H') are formed ($q = 2^{113}$). Each pair generates a truncated differential trail $\boldsymbol{\Delta K_1} \to \boldsymbol{\Delta K_2} \to \ldots \to \boldsymbol{\Delta K_8}$ with the pattern

$$\text{«}8 - 1 - 8 - 64 - 16 - 16 - 64 - 64\text{»} \text{ of the active S-boxes.}$$

«Online» stage. For each H, we get the output $\widetilde{R}$ (resp. for H' and $\widetilde{R}'$). The truncated related-key differential trail $\boldsymbol{\Delta M} \to \ldots \to \boldsymbol{\Delta \widetilde{R}}$ is realized with a probability of at least 2^{-112}. The pattern of the active S-boxes is «$8-0-8-0-16-16-64-64$». For each pair $(\widetilde{R}, \widetilde{R}')$ we construct about 2^{128} possible values of the unknown internal state. Each solution is checked directly. If the rare event actually occurred, then among the constructed solutions there will be a true one.

In more detail.

We should construct the suitable round keys for the block cipher. Choose arbitrary nonzero bytes in one column of the difference $\boldsymbol{\Delta HW_3}$ (highlighted with green on figure 3). Propagate forward to $\boldsymbol{\Delta HY_4} = \mathsf{XL}(\boldsymbol{\Delta HW_3})$. Similarly in the backward direction $\boldsymbol{\Delta HZ_4} = \mathsf{P}^{-1}\mathsf{L}^{-1}(\boldsymbol{\Delta K_5})$. We choose two nonzero columns in $\boldsymbol{\Delta K_5}$. Almost all bytes in $\boldsymbol{\Delta HZ_4}$ are active. Thus, we have $255^8 \cdot (2^{16} - 10 \cdot 255 - 1)^8 \approx 2^{191.6}$ pairs $(\boldsymbol{\Delta HY_4}, \boldsymbol{\Delta HZ_4})$, where $(2^{16} - 10 \cdot 255 - 1)$ corresponds to the number of pairs $(\boldsymbol{\Delta HW_4[0][\cdot]}, \boldsymbol{\Delta K_5[0][\cdot]})$ with the required pattern «8 – 2» (this number is derived from the properties of the MDS code or by direct computations). Solve equation $\mathsf{S}(HY_4 \oplus \boldsymbol{\Delta HY_4}) \oplus \mathsf{S}(HY_4) = \boldsymbol{\Delta HZ_4}$. We get a total of about $2^{190.4}$ solutions (see also Appendix A).

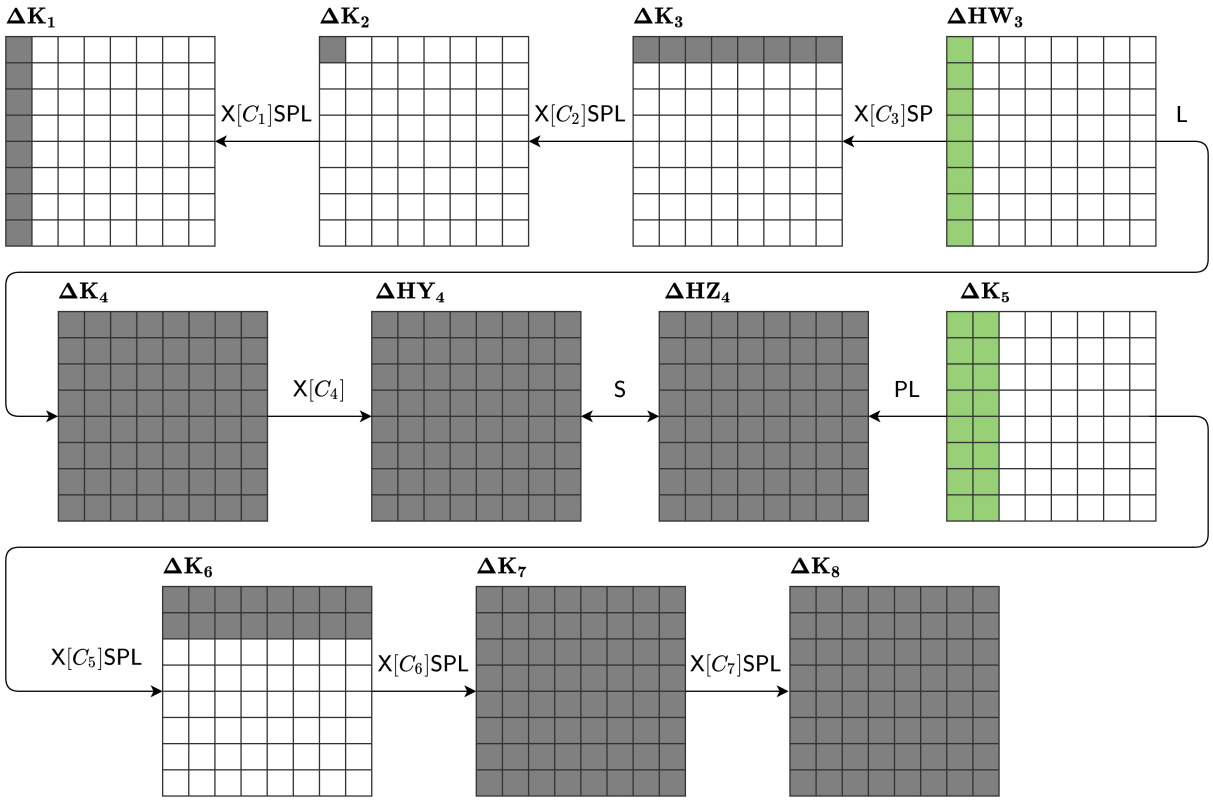

Figure 3: «Offline» stage. Truncated differential trail over round keys.

Next, in so-called «outbound phase» we compute

$$K_8 = \mathsf{LPSX}[C_7] \dots \mathsf{LPS}(HY_4) \text{ and } K_1 = \mathsf{X}[C_1]\mathsf{S}^{-1} \dots \mathsf{P}^{-1}\mathsf{L}^{-1}\mathsf{X}[C_4](HY_4).$$

We expect almost all trails $\boldsymbol{\Delta K_6} \to \boldsymbol{\Delta K_7} \to \boldsymbol{\Delta K_8}$ match the pattern «16 – 64 – 64». The trails with smaller number of active S-boxes are also appropriate. We assume that the part $\boldsymbol{\Delta K_1} \leftarrow \boldsymbol{\Delta K_2} \leftarrow \boldsymbol{\Delta K_3}$ of the constructed trail match the pattern «8 – 1 – 8» with probability $255/255^8 \approx 2^{-56}$ due to the transition «1 ← 8».

As a result we obtain about $2^{134.4} = 2^{190.4-56}$ pairs (H, H').

We request $(\widetilde{R}, \widetilde{R}')$ for each (H, H') from the «oracle». Consider the propagation of the differences with secret M (figure 4). Obviously, $M = M'$ and $\boldsymbol{\Delta M} = 0$. Before the first non-linear layer $\boldsymbol{\Delta Y_1} = \boldsymbol{\Delta K_1} \oplus \boldsymbol{\Delta M} = \boldsymbol{\Delta K_1}$. We hope that $\boldsymbol{\Delta HZ_1} = \boldsymbol{\Delta Z_1}$. The transition $\boldsymbol{\Delta HY_1} \to \boldsymbol{\Delta HZ_1}$ is possible, hence, the probability $\boldsymbol{\Delta Y_1} \to \boldsymbol{\Delta Z_1}$ is not less than $(2/256)^8 = 2^{-56}$. If actually $\boldsymbol{\Delta HZ_1} = \boldsymbol{\Delta Z_1}$ then

$$\boldsymbol{\Delta Y_2} = \boldsymbol{\Delta K_2} \oplus \boldsymbol{\Delta X_2} = \mathsf{LP}(\boldsymbol{\Delta HZ_1}) \oplus \mathsf{LP}(\boldsymbol{\Delta Z_1}) = 0.$$

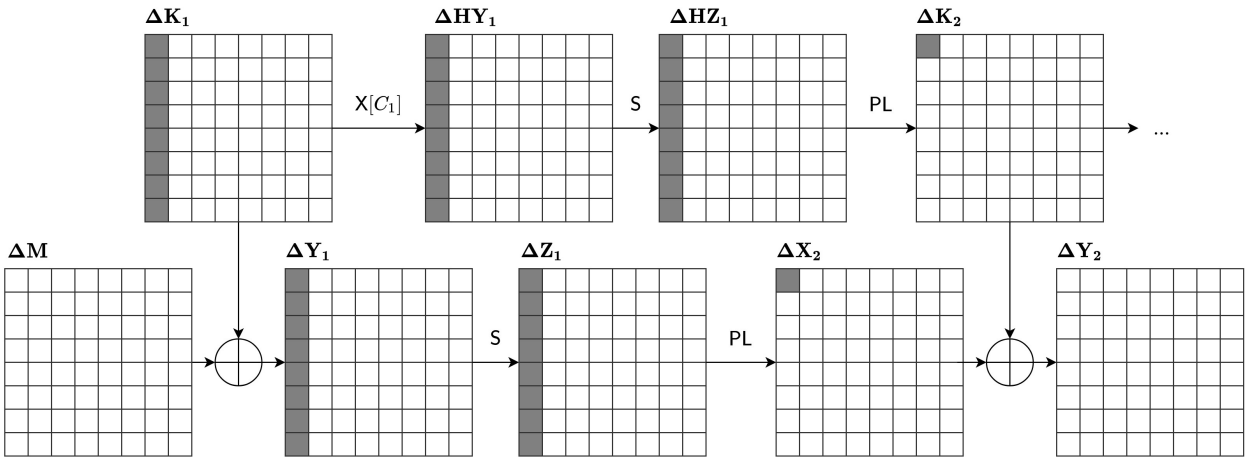

Figure 4: «Online» stage. Truncated related-key differential trail. The first round.

The same is true for $\boldsymbol{\Delta Y_3} = \boldsymbol{\Delta K_3}$ and «parallel» transitions $\boldsymbol{\Delta HY_3} \to \boldsymbol{\Delta HZ_3}$, $\boldsymbol{\Delta Y_3} \to \boldsymbol{\Delta Z_3}$ (figure 5). We also assume that $\Pr(\boldsymbol{\Delta Z_3} = \boldsymbol{\Delta HZ_3}) \geq 2^{-56}$.

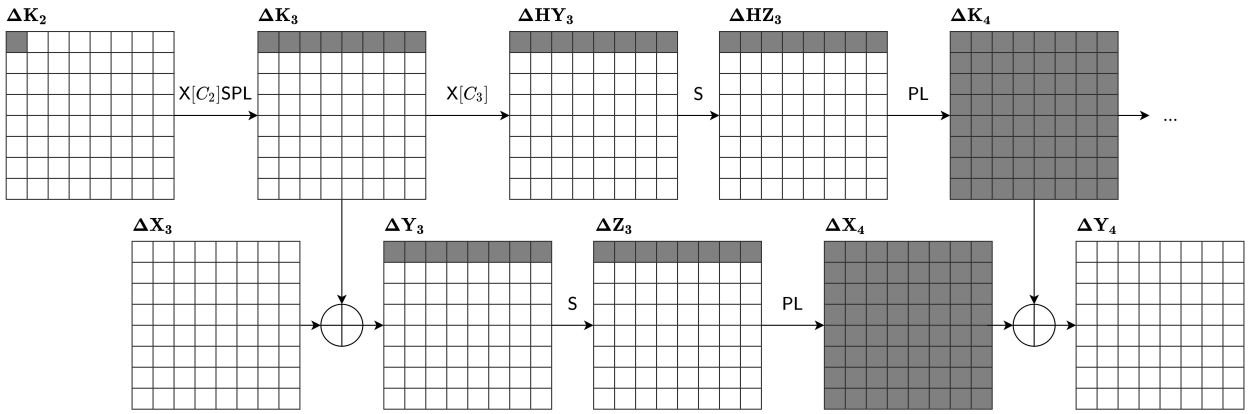

Figure 5: «Online» stage. The third round.

Thus, in the fifth round $\Pr(\boldsymbol{\Delta X_5 = 0}) = \Pr(\boldsymbol{\Delta K_5 = \Delta Y_5}) \geq 2^{-56 \cdot 2}$. So both differences $\boldsymbol{\Delta HW_6}$ and $\boldsymbol{\Delta W_6}$ have only two active columns each (figure 6). Each row in $\boldsymbol{\Delta Y_7}$ belongs to a set of 2^{16} differences (not 2^{64})

$$\boldsymbol{\Delta Y_7}[\boldsymbol{i}, \cdot] = \mathsf{l}(\boldsymbol{\Delta W_6}[\boldsymbol{i}, \boldsymbol{0}]) \oplus \mathsf{l}(\boldsymbol{\Delta HW_6}[\boldsymbol{i}, \boldsymbol{0}]) = \mathsf{l}(\boldsymbol{\Delta W_6}[\boldsymbol{i}, \boldsymbol{0}] \oplus \boldsymbol{\Delta HW_6}[\boldsymbol{i}, \boldsymbol{0}]),$$

where the difference $(\boldsymbol{\Delta W_6}[\boldsymbol{i}, \boldsymbol{0}] \oplus \boldsymbol{\Delta HW_6}[\boldsymbol{i}, \boldsymbol{0}])$ contains no more than two active bytes, $i = 0, 1, ..., 7$. For simplicity, it is assumed that all the rows in $\boldsymbol{\Delta Y_7}$ are active (this is not the case with a probability of only about $1 - (1 - 2^{-16})^8 \approx 2^{-13}$).

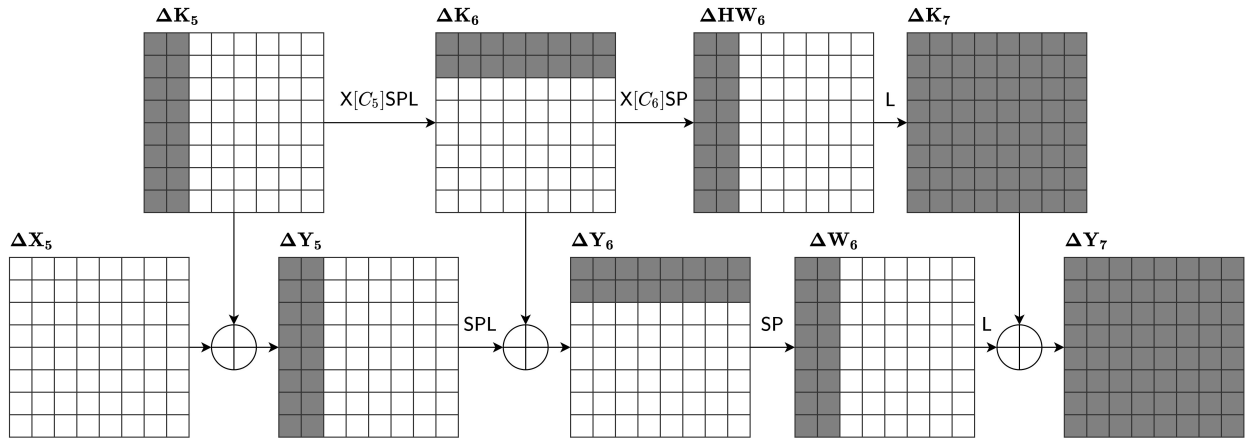

Figure 6: «Online» stage. Propagation to $\boldsymbol{\Delta Y_7}$.

Recall that $\boldsymbol{\Delta K_8}$ and the output difference $\boldsymbol{\Delta \widetilde{R}}$ are known, $\boldsymbol{\Delta M} = 0$. Therefore, the equation

$$\mathsf{S}(\boldsymbol{\Delta Y_7} \oplus Y_7) \oplus \mathsf{S}(Y_7) = \mathsf{P}^{-1}\mathsf{L}^{-1}(\boldsymbol{\Delta \widetilde{R}} \oplus \boldsymbol{\Delta K_8} \oplus \boldsymbol{\Delta M})$$

can be solved row-by-row. We expect *an average* (see also Appendix A) $2^{128} = 2^{16 \cdot 8}$ solutions Y_7. The possible secret value M is calculated by knowing

Y_7 and the round keys. The truth of each value M is checked on an arbitrary input-output pair (H, R).

The time complexity of the proposed method is

$$t = \underbrace{2^{128} \cdot 2^{64}}_{\text{"offline"}} + \underbrace{2^{112} \cdot 2^{128}}_{\text{"online"}} \approx 2^{240} \text{ operations.}$$

«Offline» and «Online» stages can be performed simultaneously. Hence, the memory is only used to store the possible values of $\boldsymbol{\Delta Y_7}$ and similar tables (no more than 2^{20} states). The described algorithm is probabilistic. We estimate the lower bound of the success probability as $1 - (1 - 2^{-112})^{q/2} \approx 1 - e^{-1} \approx 0.63$ with $q = 2^{113}$ chosen pairs (H, R).

5 Conclusion

In this paper we examine Streebog compression function as preudo-random function (PRF). Each of the two inputs (the previous state and the message block) can be used as a secret parameter and these two cases were considered.

We present two key-recovery algorithms for 7 rounds (of 12).

Setting	Rounds	Time	Memory	Data	Method
secret state	7	2^{421}	2^{354}	2^{64}	impossible polytopic
secret message	7	2^{240}	2^{20}	2^{113}	truncated differentials

The security proofs of many keyed hash-based cryptoalgorithms rely on PRF-properties of the underlying compression function. Our results demonstrate a great security margin of the Streebog 12-round compression function as a PRF in the above-mentioned secret-key settings. Thus, we have another yet informal argument that Streebog-based keyed algorithms are secure.

References

[1] *GOST R 34.11-2012 – National standard of the Russian Federation – Information technology – Cryptographic data security – Hash function*, 2012.

[2] Damgård I., "A Design Principle for Hash Functions", *LNCS*, CRYPTO 1989, **435**, ed. Brassar G., Springer, Heidelberg, 1990, 416–427.

[3] Merkle R., "One way Hash Functions and DES", *LNCS*, CRYPTO 1989, **435**, ed. Brassard G., Springer, Heidelberg, 1990, 428–446.

[4] Boneh D., Shoup V., "A Graduate Course in Applied Cryptography", 2020.

[5] Tiessen T., "Polytopic Cryptanalysis", *LNCS*, Advances in Cryptology – EUROCRYPT 2016, **9665**, ed. Fischlin M., Coron JS., Springer, Berlin, Heidelberg, 2016.

[6] Bellare M., "New Proofs for NMAC and HMAC: Security without Collision-Resistance", *LNCS*, Advances in Cryptology – CRYPTO 2006, **4117**, ed. Dwork C., Springer, Berlin, Heidelberg, April 2014.

[7] Guo J., Jean J., Leurent G., Peyrin T., Wang L., "The Usage of Counter Revisited: Second-Preimage Attack on New Russian Standardized Hash Function", *LNCS*, Selected Areas in Cryptography – SAC 2014, **8781**, ed. Joux A., Youssef A., Springer, Cham, 2014.

[8] AlTawy R., Youssef A. M., "Integral distinguishers for reduced-round Stribog", *Information Processing Letters*, **114** (2014).

[9] AlTawy R., Youssef A. M., "Preimage Attacks on reduced-round Stribog", *LNCS*, Progress in Cryptology – AFRICACRYPT 2014, **8469**, ed. Pointcheval D., Vergnaud D., Springer, Cham, 2014.

[10] AlTawy R., Kircanski A., Youssef A. M., "Rebound attacks on Stribog", *LNCS*, Information Security and Cryptology – ICISC 2013, **8565**, ed. Lee HS., Han DG., Springer, Cham, 2014.

[11] Lin D., Xu S., Yung M., "Cryptanalysis of the Round-Reduced GOST Hash Function", *LNCS*, Information Security and Cryptology. Inscrypt 2013., **8567**, Springer, Cham, 2014.

[12] Ma B., Li B., Hao R., Li X., "Improved cryptanalysis on reduced-round GOST and Whirlpool hash function", *LNCS*, Applied Cryptography and Network Security. ACNS 2014., **8479**, ed. Boureanu I., Owesarski P., Vaudenay S., Springer, Cham, 2014.

[13] Wang Z., Yu H., Wang X., "Cryptanalysis of GOST R Hash Function", *Information Processing Letters*, **114** (2014), 655–662.

[14] Kölbl S., Rechberger C., "Practical Attacks on AES-like Cryptographic Hash Functions", *LNCS*, Progress in Cryptology – LATINCRYPT 2014, **8895**, ed. Aranha D., Menezes A., Springer, Cham, 2014.

[15] Abdelkhalek A., AlTawy R., Youssef A. M., "Impossible Differential Properties of Reduced Round Streebog", *LNCS*, Codes, Cryptology, and Information Security. C2SI 2015, **9084**, ed. El Hajji S., Nitaj A., Carlet C., Souidi E., Springer, Cham, 2015, 274–286.

[16] Ma B., Li B., Hao R., Li X., "Improved (Pseudo) Preimage Attacks on Reduced-Round GOST and Grøstl-256 and Studies on Several Truncation Patterns for AES-like Compression Functions", *LNCS*, Advances in Information and Computer Security. IWSEC 2015, **9241**, ed. Tanaka K., Suga Y., Springer, Cham, 2015, 79–96.

[17] Rongjia Li, Chenhui Jin, Ruya Fan, "Improved Integral Distinguishers on Compression Function of GOST R Hash Function", *Computer Journal*, **62** (2019), 535–544.

[18] Tingting Cui, Wei Wang, Meiqin Wang, "Distinguisher on full-round compression function of GOST R", *Information Processing Letters*, **156** (2019).

[19] Chang D., Nandi M., "A Short Proof of the PRP/PRF Switching Lemma", *Cryptology ePrint Archive, Report 2008/078*, 2008.

[20] Knudsen L., "Truncated and Higher Order Differentials", 2nd International Workshop on Fast Software Encryption (FSE 1994), 1994, 196–211.

[21] Derbez P., Fouque P.-A., Jean J., "Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting", *LNCS*, Advances in Cryptology – EUROCRYPT 2013, **7881**, Springer, Berlin, Heidelberg, 2013, 371–387.

[22] Kazymyrov O., Kazymyrova V., "Algebraic Aspects of the Russian Hash Standard GOST R 34.11-2012", *Cryptology ePrint Archive, Report 2013/556*, 2013.

[23] Wagner D., "A Generalized Birthday Problem", *LNCS*, Advances in Cryptology – CRYPTO 2002, **2442**, ed. Yung M., Springer, Berlin, Heidelberg, 2002.

[24] Dinur I., Leurent G., "Improved Generic Attacks Against Hash-based MACs and HAIFA", *LNCS*, Advances in Cryptology – CRYPTO 2014, **8616**, ed. Garay J.A., Gennaro R., Springer, Berlin, Heidelberg, 2014.

[25] Mendel F., Rechberger C., Schläffer M., Søren S. Thomsen, "The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl", *LNCS*, Fast Software Encryption. FSE 2009, **5665**, ed. Dunkelman O., Springer, Berlin, Heidelberg, 2009.

A Differential properties of Streebog's S-box

The differential distribution table (DDT) is defined as follows

$$\mathrm{DDT}[\boldsymbol{\Delta x}][\boldsymbol{\Delta y}] = \left|\{x : \mathsf{s}(x) \oplus \mathsf{s}(x \oplus \boldsymbol{\Delta x}) = \boldsymbol{\Delta y}\}\right|,$$

where $\mathsf{s} : \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$, $x, \boldsymbol{\Delta x}, \boldsymbol{\Delta y} \in \mathbb{F}_{2^8}$.

The distribution of the number of solutions for Streebog's S-box is shown in the table below.

Solutions	0	2	4	6	8	256
Number	38235	22454	4377	444	25	1

For random non-zero $\boldsymbol{\Delta x}, \boldsymbol{\Delta y} \in \mathbb{F}_{2^8} \backslash 0$ the probability that at least some solution exists is

$$p = \Pr\left(\left|\{x : \boldsymbol{\Delta y} = \mathsf{s}(x) \oplus \mathsf{s}(x \oplus \boldsymbol{\Delta x})\}\right| > 0\right) = \frac{22454 + 4377 + 444 + 25}{255^2}.$$

Let $\boldsymbol{\Delta x} \neq 0$, $\boldsymbol{\Delta y} \neq 0$, and it is also known that the equation

$$\mathsf{s}(x) \oplus \mathsf{s}(x \oplus \boldsymbol{\Delta x}) = \boldsymbol{\Delta y}$$

has a solution x. Then we get a conditional distribution of the number of solutions

$$\begin{pmatrix} 2 & 4 & 6 & 8 \\ \frac{22454}{27300} & \frac{4377}{27300} & \frac{444}{27300} & \frac{25}{27300} \end{pmatrix}.$$

The expected value of such a distribution (i.e., the average number of solutions provided that at least one solution exists) is

$$\frac{1}{27300}\left(2 \cdot 22454 + 4 \cdot 4377 + 6 \cdot 444 + 8 \cdot 25\right) = \frac{2^{16} - 2^8}{27300} = 2.39\ldots = z.$$

The case «$\mathsf{S}(\boldsymbol{\Delta HY_4} \oplus HY_4) \oplus \mathsf{S}(HY_4) = \boldsymbol{\Delta HZ_4}$»

We assume, that $\boldsymbol{\Delta HZ_4}$ is a random difference. We also know that $\boldsymbol{\Delta HZ_4}$ consisting only of non-zero bytes.

Each row in $\boldsymbol{\Delta HY_4}$ is also completely non-zero and belongs to a set of 255 elements.

The probability that a single byte matches is $p \approx 0.419$. Hence a row matches with a probability of $p^8 \approx 2^{-10}$.

The probability that among the allowed $\boldsymbol{\Delta HY_4}[\mathbf{0}, \cdot]$ there is a suitable one $1 - (1 - p^8)^{255} \approx 2^{-2.2}$.

Therefore the probability for a match of all 8 rows equals to $2^{-2.2 \cdot 8} = 2^{-17.6}$.

Each pair $(\boldsymbol{\Delta HY_4}, \boldsymbol{\Delta HZ_4})$ for which the equation is solvable gives an average of $z^{64} \approx 2^{80.4}$ solutions.

We have $(2^{16} - 10 \cdot 255 - 1)^8 \approx 2^{127.6}$ possible values $\boldsymbol{\Delta HZ_4}$. As a result we obtain about

$$2^{127.6+80.4-17.6} = 2^{190.4}$$

valid states HY_4.

The case «$\mathsf{S}(\boldsymbol{\Delta Y_7} \oplus Y_7) \oplus \mathsf{S}(Y_7) = \boldsymbol{\Delta Z_7}$»

The case is similar to the previous one. We also assume, that $\boldsymbol{\Delta Z_7}$ is a random fully active difference.

Each row in $\boldsymbol{\Delta Y_7}$ belongs to a set of $u = (2^{16} - 1)$ elements.

We expect that about $u \cdot p^8 \approx 2^6$ suitable $\boldsymbol{\Delta Y_7}[\boldsymbol{i}, \cdot]$ for each $i = 0, \ldots, 7$.

In total, we have about $(2^6)^8 = 2^{48}$ possible variants of $\boldsymbol{\Delta Y_7}$.

Thus, the average number of solutions Y_7 is equal to $z^{64} \cdot 2^{48} \approx 2^{128}$.

The assumptions and estimates presented in the Appendix were also experimentally verified using software from [14].