Related-key attacks on the compression function of Streebog

Vitaly Kiryukhin

LLC «SFB Lab», JSC «InfoTeCS», Moscow, Russia
vitaly.kiryukhin@sfblaboratory.ru

Abstract

Related-key attacks against block ciphers are often considered unrealistic. In practice, as far as possible, the existence of a known «relation» between the secret encryption keys is avoided. Despite this, related keys arise directly in some widely used keyed hash functions. This is especially true for HMAC-Streebog, where known constants and manipulated parameters are added to the secret key. The relation is determined by addition modulo 2 and 2^n. The security of HMAC reduces to the properties of the underlying compression function. Therefore, as an initial analysis we propose key-recovery methods for 10 and 11 rounds (out of 12) of Streebog compression function in the related-key setting. The result shows that Streebog successfully resists attacks even in the model with such powerful adversaries.

Keywords: Streebog, related-key, truncated differentials, rebound

1 Introduction

A secure cryptographic keyless hash function H must meet many requirements, including the three most well-known: preimage resistance, second preimage resistance and collision resistance. Similar requirements are imposed on the compression function $\mathsf{g}(H, M)$ if the hash function is based on the Merkle-Damgård (MD) scheme [4, 3].

However, if the MD-like hash function is converted into the keyed one using HMAC [7] with the secret K

$$\mathsf{HMAC}(K, Msg) = \mathsf{H}((K \oplus opad)||\mathsf{H}((K \oplus ipad)||Msg)),$$

then other properties are expected from the compression function [8].

Firstly, $\mathsf{g}(H, \cdot)$ with the secret state H must be indistinguishable from a truly random function.

Secondly, the pair $\mathtt{g}(\cdot, K \oplus ipad)$ and $\mathtt{g}(\cdot, K \oplus opad)$ must be indistinguishable from a pair of random functions. In other words, $\mathtt{g}$ must be protected from attacks using two related keys [5].

Even more interesting is the situation when the Russian hash function Streebog [1] is used in HMAC. Streebog uses the Merkle-Damgård approach with some subtle differences, including the use of the checksum (modulo 2^n) of all message blocks in the finalization. In HMAC-Streebog there are four calls of the compression function where the secret key is used, one of them is

$$\mathtt{g}(\cdot, (K \oplus ipad) \boxplus \Sigma)$$

where Σ is the checksum that the attacker can freely manipulate by changing the message, «$\boxplus$» denotes the addition modulo 2^n.

Therefore, it would be reasonable for HMAC-Streebog to require $\mathtt{g}(\cdot, (K \oplus \Phi) \boxplus \Sigma)$ with the random secret K to be indistinguishable from a family of random functions. In general, the input and parameters Φ, Σ are adaptively chosen by the adversary. One can consider such significant capabilities of the attacker mostly exaggerated, but they are convenient for security proofs.

Streebog and its underlying transformations have received a lot of attention from cryptographers. Basically, the articles on the topic were devoted to analysis in the keyless settings [9–17, 19, 20].

We can cite only three works [17, 22, 23] devoted to the analysis of Streebog when using secret keys.

In [22] the key-recovery attacks on HMAC-Streebog were presented as the extension of the generic state-recovery attacks on HMAC. The time and data complexities of attacks are significantly more than «provable secure» bounds of HMAC [8]. The method also does not use the properties of the compression function.

Impossible differential properties of the compression function are utilized in [17] to mount secret-state recovery attacks on 6.75-rounds $\mathtt{g}(H, \cdot)$. The article [23] presents 7-round key-recovery attacks against $\mathtt{g}(H, \cdot)$ and $\mathtt{g}(\cdot, M)$, where H (resp. M) is secret.

As far as we know, the Streebog compression function has not been previously considered in the related-key settings. We extend the approach presented in [23] to attack $\mathtt{g}(\cdot, M)$ and propose the key-recovery method for $\mathtt{g}(\cdot, (K \oplus \Phi) \boxplus \Sigma)$. First, we construct the single-key method that works with a negligible success probability, but also with a relatively low time complexity. The rebound technique [24] and the truncated differentials [6] are the main parts of the method. Next, we present the effective way to convert

this method into a highly probable one by using the sets of related keys. As a result, we have a key-recovery method against 10 and 11 rounds (of 12). Comparative characteristics are presented in table 1.

We are convinced that our results provide an additional argument showing that Streebog compression function has a sufficient security margin even in the related-key setting.

Setting	Rounds	Time	Memory	Data	Keys	Description
secret H	6.75	$2^{399.5}$	2^{349}	2^{483}	1	[17]
	6.75	$2^{261.5}$	2^{205}	$2^{495.5}$	1	[17]
	7	2^{421}	2^{354}	2^{64}	1	[23]
	12	2^{256}	2^{256}	2^{256}	1	birthday-paradox distinguisher
	12	2^{512}	$\sim$	2	1	key guessing
secret M	7	2^{240}	2^{20}	2^{113}	1	[23]
	10	2^{224}	2^{94}	2^{225}	2^{198}	Section 6 (any relation)
	10	2^{232}	2^{91}	2^{168}	2^{145}	Section 6 (only $\oplus$)
	11	2^{224}	2^{68}	2^{225}	2^{224}	Section 6 (only $\oplus$)
	12	2^{367}	2^{145}	2^{145}	2^{145}	parallel key guessing
	12	2^{314}	2^{198}	2^{198}	2^{198}	
	12	2^{288}	2^{224}	2^{224}	2^{224}	
	12	2^{256}	2^{256}	2^{256}	2^{256}	
	12	2^{512}	$\sim$	2	1	key guessing

Table 1: Attacks on the Streebog compression functions in secret-key settings. «Time» (t) in g computations, «Memory» in n-bit blocks, «Data» (q) in chosen message-output pairs over all keys, «Keys» is the number of used related keys (single-key attack is denoted by «Keys = 1»).

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 a binary vector. Denote 8×8 matrix space over $\mathbb{F}_{2^8}$ by $\mathbb{F}_{2^8}^{8\times 8}$ (we also use symbol $\mathbb{F}_{2^8}^{8}$ as a vector space). Elements from $\mathbb{F}_{2^8}^{8\times 8}$ will be denoted by capital letters: A, B. Blocks of states and messages also belong to $\mathbb{F}_{2^8}^{8\times 8}$. Elements of a matrix are indexed by $0 \leq i, j \leq v-1$ (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. Elements from $\mathbb{F}_{2^8}/\mathbb{F}_{2^8}^{8}/\mathbb{F}_{2^8}^{8\times 8}$ can be represented as 8-, 64-, 512-bit strings, respectively.

Denote addition modulo 2 and addition modulo 2^n by symbols «$\oplus$» and «$\boxplus$» correspondingly, $n = 512$. These operations are defined naturally for all the objects under consideration.

We refer to $\boldsymbol{\Delta B} = B \oplus B' \in \mathbb{F}_{2^8}^{8\times 8}$ as a difference and indicate it in bold: $\boldsymbol{\Delta M}$, $\boldsymbol{\Delta K_4}$. If $\boldsymbol{\Delta B}[\boldsymbol{i}, \boldsymbol{j}] \neq 0$ then we say that the position (i, j) is active,

otherwise inactive. The differential trail is the sequence of the differences after each transformation in the cipher. The truncated differential trail is the set of the differential trails that have the same active positions.

The transformations over $\mathbb{F}_{2^8}^{8\times 8}$ (also over $\mathbb{F}_{2^8}^{8}$ and $\mathbb{F}_{2^8}$) are denoted in 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}.

3 Streebog

The state size of Streebog consists of $n = 512$ bits (8×8 bytes).

The message Msg is hashed as follows.

The text is always padded with bit string $\mathtt{10}\ldots\mathtt{0}$ and divided into l blocks of n bits $Msg||\mathtt{10}\ldots\mathtt{0} = M_1||\ldots||M_l$. The compression function is sequentially applied to the previous bit state and block

$$H_{i+1} = \mathsf{g}_{i\cdot n}(H_i, M_{i+1}),\ i = 0, ..., l-1,\ H_0 = IV \in \mathbb{F}_{2^8}^{8\times 8},$$

where IV is a predefined constant. The counter $N = i \cdot n \in \mathbb{F}_{2^8}^{8\times 8}$ is the number of already hashed bits.

The bit length L and the checksum $\Sigma = M_1 \boxplus ... \boxplus M_l$ are «mixed» with the state at the finalizing stage

$$H_{l+1} = \mathsf{g}_0(H_l, L),\ H_{l+2} = \mathsf{g}_0(H_{l+1}, \Sigma).$$

If 256-bit hash function is used, the output H_{l+2} is truncated to 256 bit.

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

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

$H \in \mathbb{F}_{2^8}^{8\times 8}$ is the previous state of the hash function;

$M \in \mathbb{F}_{2^8}^{8\times 8}$ is the message block;

$N \in \mathbb{F}_{2^8}^{8\times 8}$ is the number of previously hashed bits;

$R \in \mathbb{F}_{2^8}^{8\times 8}$ 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 [21], 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

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

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

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

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:

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

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

HMAC-Streebog (see figure 1) is defined in [2] as

$$\mathsf{HMAC\text{-}Streebog}(K, Msg) = \mathsf{H}\left((\overline{K} \oplus opad)||\mathsf{H}((\overline{K} \oplus ipad)||Msg)\right),$$

where $\overline{K} \in \mathbb{F}_{2^8}^{8\times 8}$ is obtained by padding the k-bit secret key K with zero bits, $256 \le k \le 512$, $opad$ and $ipad$ are different nonzero constants.

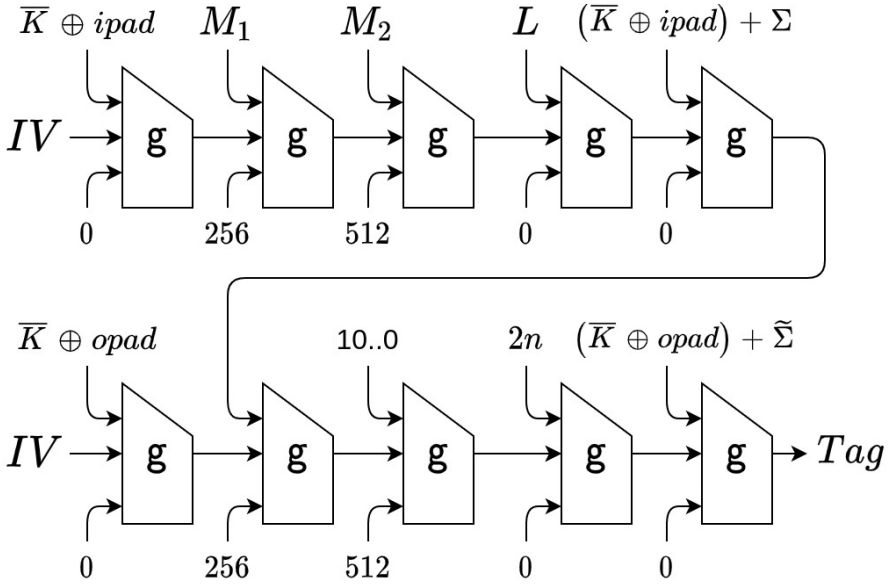

Figure 1: $\mathsf{HMAC\text{-}Streebog\text{-}512}$, $512 \le L < 1024$, $\Sigma = M_1 \boxplus M_2$.

The secret key $\overline{K}$ is used four times as part of the *message* input. The checksum Σ is directly controlled by the attacker and determines the relation between the keys. The sequence of the chosen messages Msg_1,...,Msg_q generates the sequence of the chosen relations Σ_1,...,Σ_q. Hence, the adversary has as many related keys as needed. The state H is usually not known to the attacker, but we assume the opposite. Firstly, H can be revealed as a result of some generic attack against HMAC, secondly, this may be convenient for a formal security proof. The output R is observed only after the last call of g, but if, for example, $\mathsf{H}(K||Msg)$ is used instead of HMAC, then R is known.

Thus, these considerations motivate us to examine the security of

$$\mathsf{g}(H, M) = R,\ M = (K \oplus \Phi) \boxplus \Sigma,$$

where K is the secret 512-bit key, the output R is observed, and H, Φ, Σ are chosen adaptively. If g is secure even in the described setting, then there is no reason to worry about cases when the opponent has fewer opportunities.

4 Generic attack

The key-recovery attacks on the cryptoalgorithm are usually compared to a simple guessing of the key. Obviously, a k-bit key can be found with 2^{k-1} trials on average.

However, in the related-key setting we have another generic attack. Let, for example, the adversary attacks an arbitrary block cipher E. The sequence of ciphertexts C_1,...,C_q is the result of encryption of the same text P, but with the different key

$$C_i = \mathsf{E}(K \oplus \Phi_i, P),\ i = 1, ..., q.$$

Pairs (C_i, Φ_i) are sorted by C_i and stored in memory. The attacker makes t guesses $\widetilde{K}$. If the value of $\widetilde{C} = \mathsf{E}(\widetilde{K}, P)$ exists in memory $\widetilde{C} = C_j$, then surely $K = \widetilde{K} \oplus \Phi_j$, $1 \le j \le q$. One revealed key allows to trivially find all the others. The probability of successful guessing in one attempt is $q \cdot 2^{-k}$. Hence, if $t \cdot q = 2^k$ then the probability of the successful attack exceeds $\frac{1}{2}$.

Therefore, 2^r related keys allow to mount generic attack with 2^{k-r} time and 2^r memory complexities. We emphasize that the time complexity of any related-key attack should be compared with 2^{k-r}, not 2^k.

The optimal time complexity is $2^{k/2}$ with $r = \frac{k}{2}$. Informally speaking, any cryptoalgorithm with a k-bit key provides only $\frac{k}{2}$-bit security if the number of the related keys available to the adversary is unlimited. Also note that the

type of relation can be rather arbitrary and include, for example, modular addition. The main thing is that the attacker has access to encryption with different keys and the relations between the keys are known.

5 Single-key attack

At the beginning, we consider the case when the message $M = (K \oplus \Phi) \boxplus \Sigma$ is secret and $\Phi = \Sigma = 0$. The attack against 7 rounds in such conditions was considered earlier in [23]. We use the similar approach and construct the low-probability attack against 10 rounds of

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

The master-key H of the underlying block cipher is directly chosen by the adversary

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

The key-recovery method consists of two stages.

«Offline» stage uses the rebound approach [24]. About 2^{28} pairs (H, H') are generated. Each pair determines a truncated differential trail $\boldsymbol{\Delta K_1} \to ... \to \boldsymbol{\Delta K_{11}}$. Some precomputations are also performed under the assumption that $\boldsymbol{\Delta Y_9} = \boldsymbol{\Delta K_9}$.

«Online» stage. For each input pair (H, H') we get the output $(\widetilde{R}, \widetilde{R}')$. The truncated related-key differential trail $\boldsymbol{\Delta M} \to \ldots \to \boldsymbol{\Delta \widetilde{R}}$ is realized with a probability of at least $p_{trail} = 2^{-224}$. For each pair $(\widetilde{R}, \widetilde{R}')$ we construct on average one possible value of the unknown internal state and check it directly. If the rare event actually occurred, then we definitely obtain the true key.

The patterns of the active S-boxes are

$$\boldsymbol{\Delta K_1} \to \ldots \to \boldsymbol{\Delta K_{11}} : \text{«} \quad 8{-}1{-}8{-}64{-}8{-}1{-}8{-}64{-}64{-}64{-}64\text{»},$$
$$\boldsymbol{\Delta M} \to \boldsymbol{\Delta Y_1} \to \ldots \to \boldsymbol{\Delta \widetilde{R}} : \text{«}0{-}8{-}0{-}8{-}0\ {-}8{-}0{-}8{-}0\ {-}64{-}64{-}64\text{»}.$$

The offline stage constructs the suitable round keys for the block cipher. Choose arbitrary nonzero bytes in one arbitrary column of the difference $\boldsymbol{\Delta HW_3}$ (highlighted with green on figure 2).

Propagate forward to $\boldsymbol{\Delta HY_4} = \mathsf{X}[C_4]\mathsf{L}(\boldsymbol{\Delta HW_3})$. Similarly in the backward direction $\boldsymbol{\Delta HZ_4} = \mathsf{P}^{-1}\mathsf{L}^{-1}(\boldsymbol{\Delta K_5})$. Thus, we have $255^8 \cdot 8 \cdot 255^8 \cdot 8 \approx 2^{134}$ pairs $(\boldsymbol{\Delta HY_4}, \boldsymbol{\Delta HZ_4})$.

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 more than 2^{132} solutions (see Appendix A).

In «outbound phase» we compute

$$K_1 = \mathsf{X}[C_1]\mathsf{S}^{-1}\ldots\mathsf{P}^{-1}\mathsf{L}^{-1}\mathsf{X}[C_4](HY_4) \text{ and } K_{11} = \mathsf{LPSX}[C_{10}]\ldots\mathsf{LPS}(HY_4).$$

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 $8 \cdot 255/255^8 \approx 2^{-53}$ due to the transition «1 ← 8». Note that any of eight possible patterns «1 ← 8» is suitable. Similar reasoning is true for $\boldsymbol{\Delta K_6} \rightarrow \boldsymbol{\Delta K_7} \rightarrow \boldsymbol{\Delta K_8}$ (and any values of $\boldsymbol{\Delta K_9} \rightarrow \boldsymbol{\Delta K_{10}} \rightarrow \boldsymbol{\Delta K_{11}}$ is appropriate.). Actually 64 truncated trails are used, eight appropriate propagation possibilities in the backward and the same for forward.

As a result we obtain about $q_{pair} = 2^{26} = 2^{132-53-53}$ pairs (H, H') and approximately $2^{23} = 2^{26}/8$ of them have the active first column. The time complexity of the offline stage is about $t_{offline} = 2^{134}$ operations.

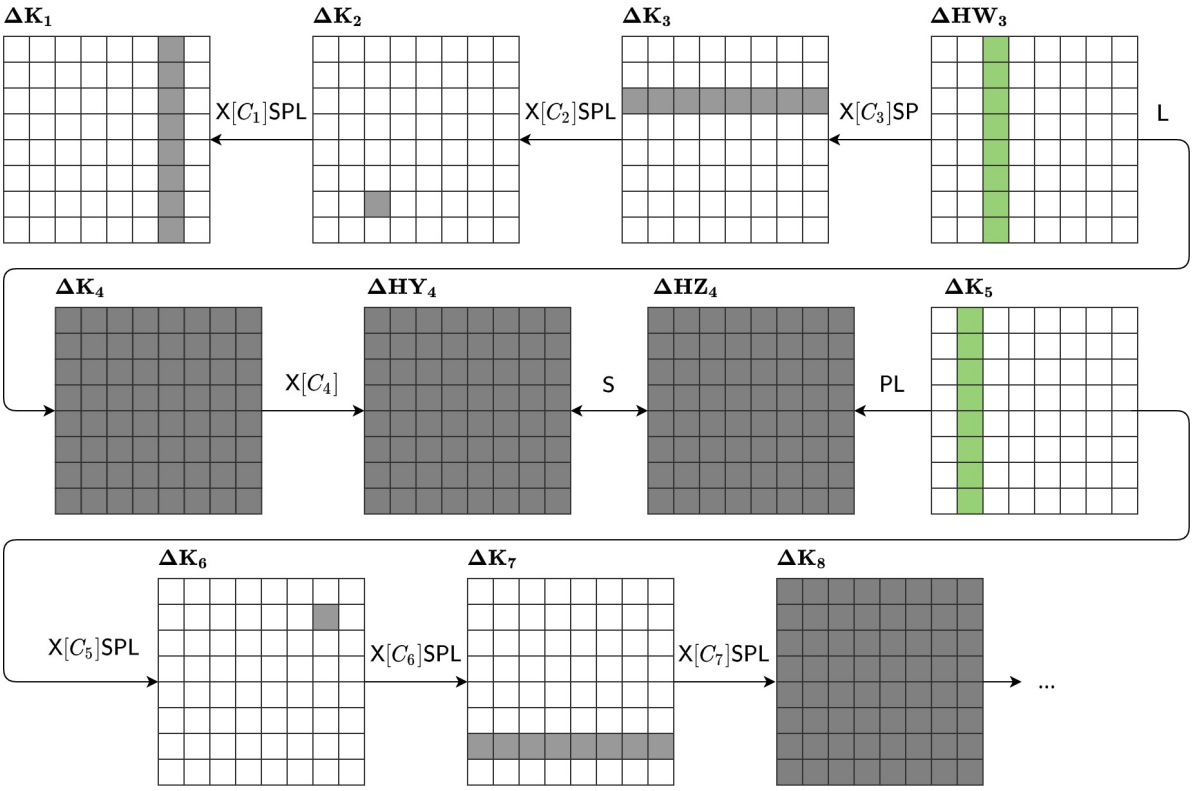

Figure 2: Offline stage. One of the possible truncated differential trail over first eight round keys.

At the online stage, pairs $(\widetilde{R}, \widetilde{R}')$ are requested for each (H, H') from the «oracle».

We expect four internal collisions at the same time (figure 3). In the considered single-key setting, $M = M'$ and $\boldsymbol{\Delta M} = 0$. The differences $\boldsymbol{\Delta K_1}$, $\boldsymbol{\Delta K_3}$, $\boldsymbol{\Delta K_5}$, $\boldsymbol{\Delta K_7}$ induce eight active bytes (one row or one column of the

state) in the «encryption». If the transitions through S are the same in both «encryption» and «key schedule» then $\boldsymbol{\Delta K_2}$, $\boldsymbol{\Delta K_4}$, $\boldsymbol{\Delta K_6}$, $\boldsymbol{\Delta K_8}$ make a zero difference in «encryption».

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 $p_{coll} = (2/256)^8 = 2^{-56}$. If actually $\boldsymbol{\Delta HZ_1} = \boldsymbol{\Delta Z_1}$ then we obtain the first internal collision

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

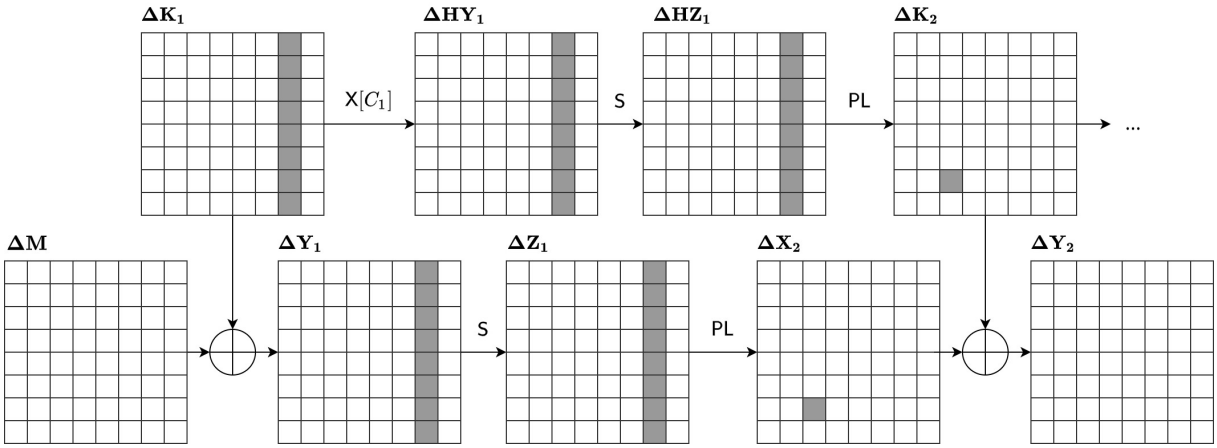

Figure 3: Online stage. Truncated related-key differential trail. The first round.

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 4). We also assume that $\Pr(\boldsymbol{\Delta Z_3} = \boldsymbol{\Delta HZ_3}) = p_{coll}$.

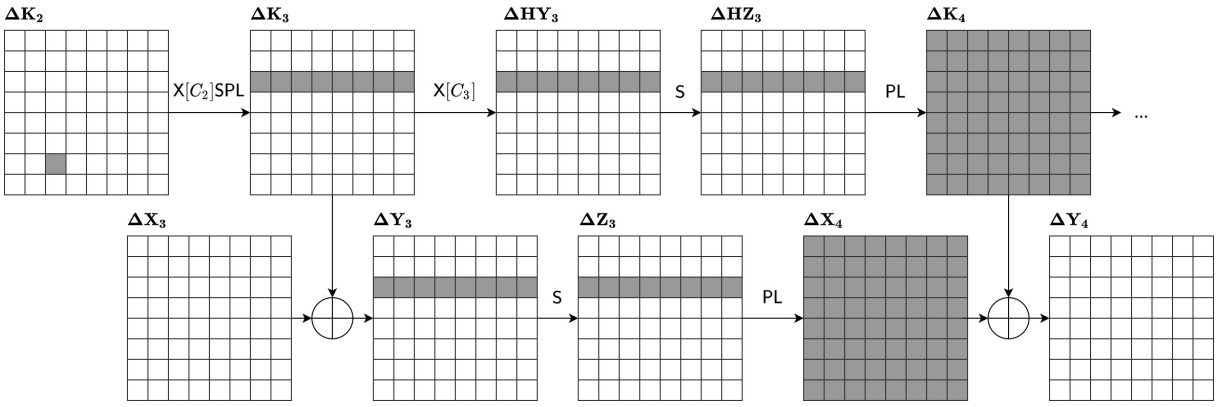

Figure 4: Online stage. The third round.

Similarly for the third and the fourth internal collision (figure 5), $\Pr(\boldsymbol{\Delta Z_5} = \boldsymbol{\Delta HZ_5}) = p_{coll}$, $\Pr(\boldsymbol{\Delta Z_7} = \boldsymbol{\Delta HZ_7}) = p_{coll}$.

Therefore, we have

$$p_{\text{trail}} = \Pr\left(\boldsymbol{\Delta X_9} = 0\right) = \Pr\left(\boldsymbol{\Delta Y_9} = \boldsymbol{\Delta K_9}\right) \geq (p_{coll})^4 = 2^{-56 \cdot 4} = 2^{-224}.$$

We use this distinguishing feature to construct the attack.

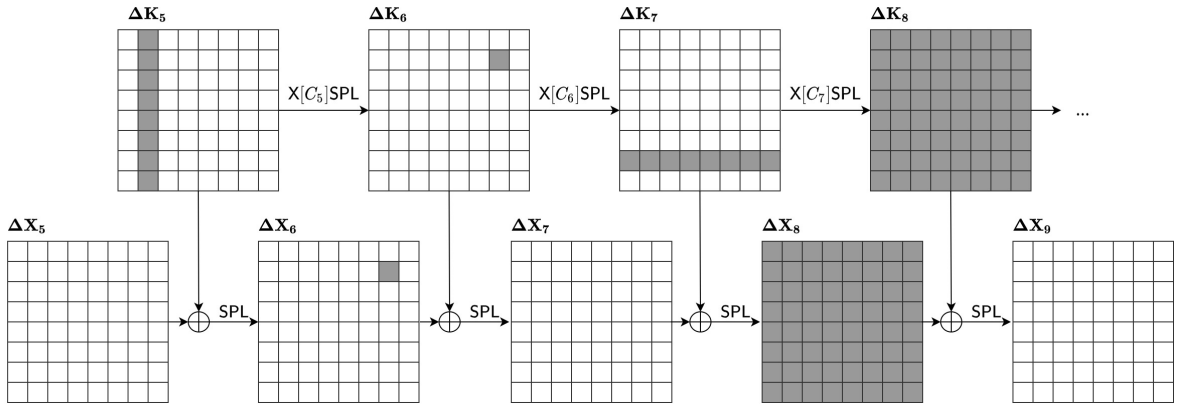

Figure 5: Online stage. Rounds 5, 6, 7 and 8.

After the rebound, the precomputations are performed at the offline stage. We have 2^{26} pairs (H, H') and derived round keys $(K_1, ..., K_{11})$, $(K'_1, ..., K'_{11})$. Assuming that the trail is realized, (H, H') determines the only one $\boldsymbol{\Delta Y_9} = \boldsymbol{\Delta K_9}$. Try all possible values in the column $Y_9[\cdot, i]$ and propagate $\boldsymbol{\Delta Y_9[\cdot, i]}$ to $\boldsymbol{\Delta W_{10}[\cdot, i]}$, $i = 0, .., 7$.

For fixed (H, H') eight tables are stored in memory, i-th table contains the sequence of sorted values $\boldsymbol{\Delta W_{10}[\cdot, i]} = W_{10}[\cdot, i] \oplus W'_{10}[\cdot, i]$,

$$W_{10}[\cdot, i] = \left(\mathsf{PS}(K_{10}[i, \cdot] \oplus \mathsf{LPS}(Y_9[\cdot, i]))\right),$$
$$W'_{10}[\cdot, i] = \left(\mathsf{PS}(K'_{10}[i, \cdot] \oplus \mathsf{LPS}(\boldsymbol{\Delta Y_9[\cdot, i]} \oplus Y_9[\cdot, i]))\right).$$

and corresponding set of $W_{10}[\cdot, i]$. Assuming that after two nonlinear layers, $\boldsymbol{\Delta W_{10}[\cdot, i]}$ is distributed uniformly, one value of $\boldsymbol{\Delta W_{10}[\cdot, i]}$ corresponds to one value of $W_{10}[\cdot, i]$ on average.

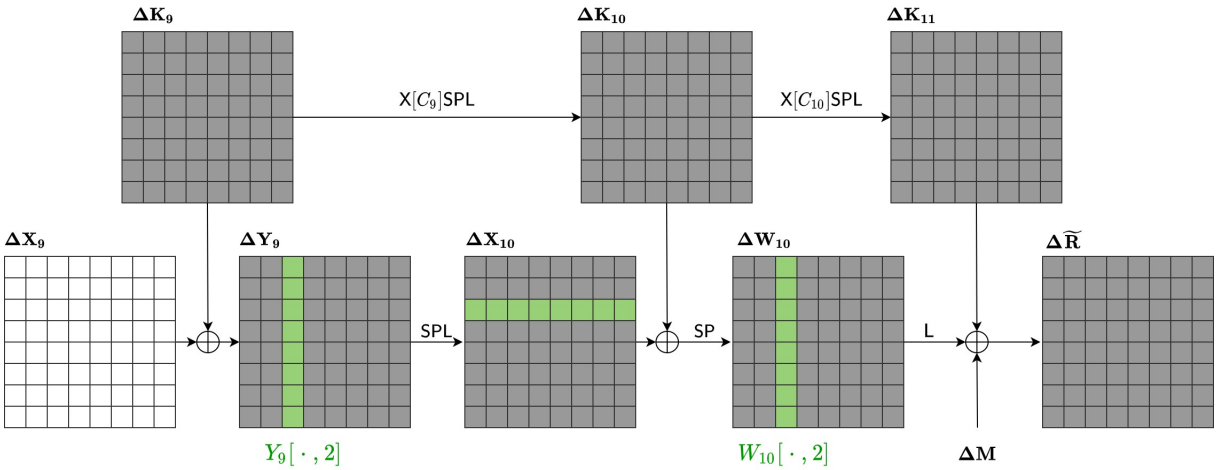

Figure 6: Additional precomputation at the offline stage.

For all (H, H') about $2^{26} \cdot 8 \cdot 2^{64}$ tables are constructed. In total, this step requires about $2 \cdot 2^{26} \cdot 8 \cdot 2^{64} = 2^{94}$ computations and the same number of n-bit blocks in memory. Hence, the complexity of the offline stage almost does not increase, $t_{offline} = 2^{134} + 2^{93} \approx 2^{134}$.

Consider again the online stage. The pair (H, H') defines the last round keys (K'_{11}, K'_{11}) and the output pair $(\widetilde{R}, \widetilde{R}')$. The difference $\boldsymbol{\Delta M}$ is also known to the adversary (in the single-key setting $\boldsymbol{\Delta M} = 0$). If the differential trail really happened ($\boldsymbol{\Delta X_9} = 0$), then i-th column of

$$\boldsymbol{\Delta W_{10}} = \mathsf{L}^{-1}(\boldsymbol{\Delta \widetilde{R}} \oplus \boldsymbol{\Delta K_{11}} \oplus \boldsymbol{\Delta M})$$

must be in i-th table. Otherwise, the pair will surely be discarded. Usually one solution $W_{10}[\cdot, i]$ for i-th column is found. We construct internal state W_{10}, compute $M = \mathsf{L}(W_{10}) \oplus K_{11} \oplus \widetilde{R}$ and check them with the other input-output pair (H, R). The average time complexity of the online stage is estimated as $t_{\text{online}} \approx q_{pair}$. The probability of success is negligible

$$p_{1k\text{-}attack} \approx q_{pair} \cdot p_{trail} = 2^{26} \cdot 2^{-224} = 2^{-198}.$$

6 Related-key attacks

The single-key low-probability attack presented above can be easily transformed into attack in the related-key setting.

Let's perform the offline stage once and store 2^{26} convenient pairs (H, H') and precomputed tables in memory. We use about $2^r = 2^{198} = (p_{1k\text{-}attack})^{-1}$ related keys and try the online stage against each of them independently. If one key is recovered, then all the others are can, too, be easily found from

known relations. The probability of success is now significant and is estimated as $1-(1-p_{1k\text{-}attack})^{2^r} \approx 1-e^{-1} \approx 0.63$.

The time complexity is $t = 2^{134} + 2^{26} \cdot 2^{198} \approx 2^{224}$ (for comparison, the generic method $t = 2^{k-r} = 2^{314}$). It is not difficult to see that almost any possible relation can be used ($M \oplus \Phi$, $M \boxplus \Sigma$, $M \oplus \Phi \boxplus \Sigma$ etc.).

However, a more effective attacks exists if the relation is bitwise xor (i.e. $M \oplus \Phi$). We describe them in the following two subsections.

6.1 Reducing the number of related keys

At the offline stage, we select only those pairs (H, H') that activate only one chosen column $\boldsymbol{\Delta K_1}[\cdot, \mathbf{0}] \neq 0$, $\boldsymbol{\Delta K_1}[\cdot, \mathbf{1}] = ... = \boldsymbol{\Delta K_1}[\cdot, \mathbf{7}] = 0$. The number of convenient pairs has been reduced to $q_{pair} = 2^{23} = 2^{26}/8$.

At the online stage, we use many sets

$$\mathbb{M}_i = \{M \oplus \Phi'_i \oplus \Phi_j\},\ \Phi_j[\cdot, 0] \neq 0,\ \Phi_j[\cdot, 1] = ... = \Phi_j[\cdot, 7] = 0,\ j = 0, ..., 2^{64}-1,$$

of the related keys. The values of Φ'_i are chosen so that $\mathbb{M}_{i_1} \cap \mathbb{M}_{i_2} = \emptyset$, $\forall i_1 \neq i_2$. The set induces $(2^{128} - 2^{64}) \approx 2^{128}$ different pairs (M, M'), where also only the first column of the difference may be active $\boldsymbol{\Delta M}[\cdot, 0] \neq 0$, other columns are obviously inactive. Note that the pairs (M, M) are also used. The pairs (M, M') and (M', M) are distinct if $M \neq M'$. Indeed, (M, M') and (H, H'), $H \neq H'$ generates two related-key differential trails,

$$(M \oplus H) \oplus (M' \oplus H') = (M' \oplus H) \oplus (M \oplus H'), \text{ but in general}$$
$$\mathsf{S}(M \oplus H) \oplus \mathsf{S}(M' \oplus H') \neq \mathsf{S}(M' \oplus H) \oplus \mathsf{S}(M \oplus H').$$

Hence, about $2^{128} \cdot q_{pair} = 2^{151}$ starting points are obtained with one $\mathbb{M}$ (at the same time, the number of the required queries is $2 \cdot q_{pair} \cdot 2^{64} = 2^{88}$).

The probability of the resulting trail is slightly worse. If $\boldsymbol{\Delta M} = 0$, then $\boldsymbol{\Delta Y_1} = \boldsymbol{\Delta K_1}$ and there is always a possibility to the transition $\boldsymbol{\Delta Y_1} \to \boldsymbol{\Delta Z_1}$, where $\boldsymbol{\Delta Z_1} = \boldsymbol{\Delta H Z_1}$. Otherwise, $\boldsymbol{\Delta Y_1} \neq \boldsymbol{\Delta K_1}$ and the target transition $\boldsymbol{\Delta Y_1} \to \boldsymbol{\Delta Z_1}$ may be impossible. However, assuming that $\boldsymbol{\Delta Y_1}[\cdot, \mathbf{0}]$ is random, we can treat $\boldsymbol{\Delta Z_1}$ also as random value and estimate

$$p_{rand\text{-}coll} = \Pr(\boldsymbol{\Delta Z_1} = \boldsymbol{\Delta H Z_1}) = 2^{-64} < p_{coll}.$$

The probability of the modified truncated trail is

$$p'_{trail} = p_{rand\text{-}coll} \cdot (p_{coll})^3 = 2^{-232}.$$

The rest of the attack is the same.

Thus, we need about $q_{set} = 2^{81} = (p'_{trail} \cdot 2^{151})^{-1}$ sets $\mathbb{M}$. The total number of the related keys is $q_{key} = q_{set} \cdot 2^{64} = 2^{145} = 2^r$. The number of the queries $q = 2 \cdot q_{pair} \cdot 2^{64} \cdot q_{set} = 2^{169}$. The memory for tables at the precomputation is $2 \cdot q_{pair} \cdot 8 \cdot 2^{64} = 2^{91}$. The time complexity slightly increases $t = q_{set} \cdot 2^{128} \cdot q_{pair} = (p'_{trail})^{-1} = 2^{232}$, but for the generic attack $t = 2^{k-r} = 2^{369}$.

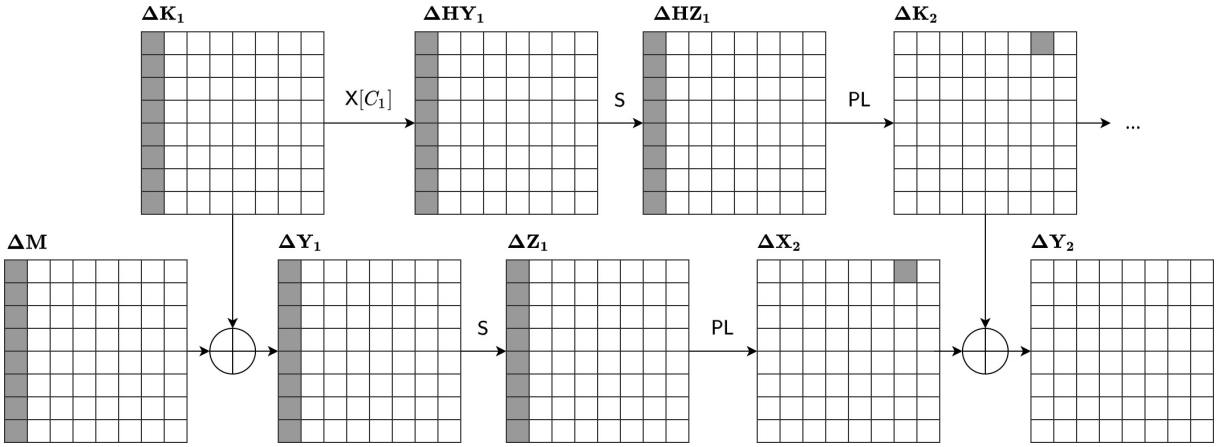

Figure 7: Online stage. Truncated related-key differential trail. The first round. $\boldsymbol{\Delta M}[\cdot, \mathbf{0}] \neq 0$.

6.2 Extension to 11 round

We change the truncated differential trail used by adding one round at the beginning (figure 8). The patterns of the active S-boxes are

$$\boldsymbol{\Delta K_1} \to \ldots \to \boldsymbol{\Delta K_{12}} : \text{«} \quad 64{-}8{-}1{-}8{-}64{-}8{-}1{-}8{-}64{-}64{-}64{-}64\text{»},$$
$$\boldsymbol{\Delta M} \to \boldsymbol{\Delta Y_1} \to \ldots \to \boldsymbol{\Delta Y_{12}} : \text{«}64{-}0 \;\; {-}8{-}0{-}8{-}0 \;\; {-}8{-}0{-}8{-}0 \;\; {-}64{-}64{-}64\text{»}.$$

The rebound starts with $\boldsymbol{\Delta HW_4}$ and $\boldsymbol{\Delta K_6}$ instead of $\boldsymbol{\Delta HW_3}$ and $\boldsymbol{\Delta K_5}$. The remaining steps are similarly «shifted to the right» (see Appendix B).

To provide $\boldsymbol{\Delta Y_1} = \boldsymbol{\Delta X_2} = 0$, we must use $\boldsymbol{\Delta M} = \boldsymbol{\Delta K_1}$. In this case, the probability of the rare event does not change $p_{trail} = (p_{coll})^4 = 2^{-224}$.

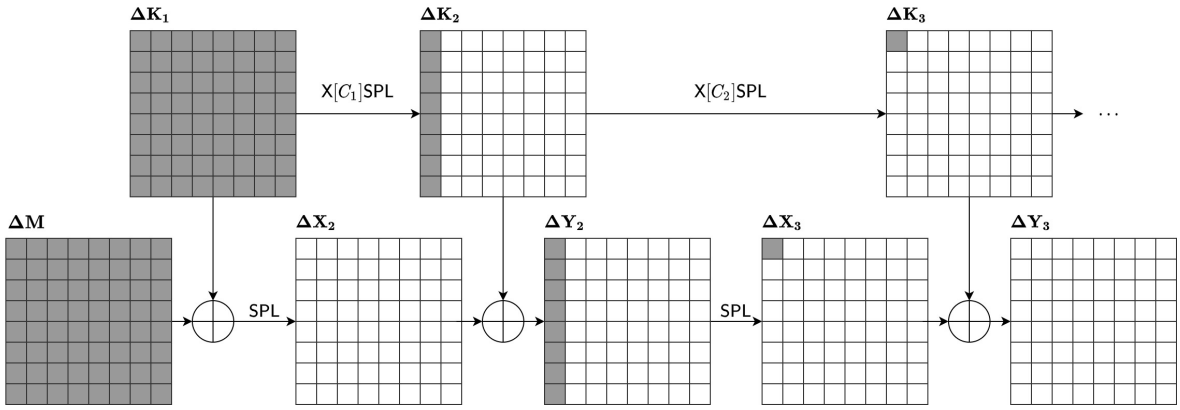

Figure 8: Truncated related-key differential trail, $\boldsymbol{\Delta M} = \boldsymbol{\Delta K_1}$.

We use only one pair (H, H') after the offline stage. Hence, only one sequence $(K_1, K_1') \to \ldots \to (K_{12}, K_{12}')$ is used, and $2 \cdot 8 \cdot 2^{64} = 2^{68}$ n-bit blocks of memory are required to store the rows of $\boldsymbol{\Delta W_{11}}$ and W_{11} after the precomputation.

The i-th pair of the related keys is

$$(M_i, M_i') = (M \oplus \Phi_i, M \oplus \Phi_i \oplus \boldsymbol{\Delta K_1}),\ i = 1, ..., \frac{q_{key}}{2},$$

different values of Φ_i should give q_{key} different keys, and $\boldsymbol{\Delta M} = \boldsymbol{\Delta K_1}$ is always true. Again, each pair of keys gives two points to start for the online stage

$$((H, M_i), (H', M_i'))\ \text{ and }\ ((H, M_i'), (H', M_i))\,.$$

The success probability is also $(1 - e^{-1}) \approx 0.63$ with $q_{key} = (p_{trail})^{-1}$. The query complexity is $q = 2 \cdot q_{key} = 2^{225}$. As in previous attacks, the time complexity is equal to the number of starting points

$$t \approx t_{offline} + 2 \cdot \frac{q_{key}}{2} = q_{key} = 2^{224} = 2^r.$$

For comparison, the complexity of the generic method $t = 2^{k-r} = 2^{288}$.

7 Conclusion

In many practical cases, Streebog hashes the secret key joined to the message. Due to the checksum modulo 2^n in the finalization, the related keys always arise. For example, in HMAC-Streebog the one processed block is $M = (K \oplus \Phi) \boxplus \Sigma$, where K is the secret key, Φ is known, Σ is chosen adaptively by the adversary. Therefore, this motivates us to investigate round-reduced Streebog compression function $\mathbf{g}(H, M)$ with the secret M under above mentioned relations.

Among all the threat models for symmetric keyed cryptoalgorithms, the related-key setting is one of the most powerful. We present key-recovery algorithms up to 10 rounds (out of 12) when almost any relations exist (e.g. addition modulo 2 or modulo 2^n). If only bitwise xor is used then the attack can be extended for 11 rounds. The rebound approach and the related-key truncated differential trails are extensively used. The time complexity of the methods is close to that of the generic approaches.

Thus, we have significant evidence that Streebog compression function is hard to break even in the threat model under consideration. Therefore, another argument was obtained in favor of the security of Streebog-based keyed algorithms.

8 Acknowledgements

The author is grateful to Andrey Scherbachenko for the careful consideration of the article and many suggestions that significantly improved the quality of the text.

References

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

[2] *R 50.1.113-2016 – Information technology – Cryptographic data security – Cryptographic algorithms accompanying the use of electronic digital signature algorithms and hash functions*, 2016.

[3] Damgård I., "A design principle for hash functions", CRYPTO 1989, Lect. Notes Comput. Sci., **435**, 1990, 416–427.

[4] Merkle R., "One way wash functions and DES", CRYPTO 1989, Lect. Notes Comput. Sci., **435**, 1990, 428–446.

[5] Biham E., "New types of cryptoanalytic attacks using related keys (extended abstract)", EUROCRYPT 93, Lect. Notes Comput. Sci., **765**, 1993, 398–409.

[6] Knudsen L., "Truncated and higher order differentials", FSE 1994, Lect. Notes Comput. Sci., **1008**, 1994, 196–211.

[7] Bellare M., Canetti R., Krawczyk H., "Keying Hash Functions for Message Authentication", Crypto'96, Lect. Notes Comput. Sci., **1109**, 1996, 1–15.

[8] Bellare M., "New proofs for NMAC and HMAC: security without collision-resistance", CRYPTO 2006, Lect. Notes Comput. Sci., **4117**, April 2014, 602–619.

[9] Guo J., Jean J., Leurent G., Peyrin T., Wang L., "The usage of counter revisited: second-preimage attack on new Russian standardized hash function", SAC 2014, Lect. Notes Comput. Sci., **8781**, 2014, 195–211.

[10] AlTawy R., Youssef A. M., "Integral distinguishers for reduced-round Stribog", *Information Processing Letters*, **114** (2014), 426–431.

[11] AlTawy R., Youssef A. M., "Preimage attacks on reduced-round Stribog", AFRICACRYPT 2014, Lect. Notes Comput. Sci., **8469**, 2014, 109–125.

[12] AlTawy R., Kircanski A., Youssef A. M., "Rebound attacks on Stribog", ICISC 2013, Lect. Notes Comput. Sci., **8565**, 2014, 175–188.

[13] Lin D., Xu S., Yung M., "Cryptanalysis of the round-reduced GOST hash function", Inscrypt 2013, Lect. Notes Comput. Sci., **8567**, 2014, 309–322.

[14] Ma B., Li B., Hao R., Li X., "Improved cryptanalysis on reduced-round GOST and Whirlpool hash function", ACNS 2014, Lect. Notes Comput. Sci., **8479**, 2014, 289–307.

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

[16] Kölbl S., Rechberger C., "Practical attacks on AES-like cryptographic hash functions", LATINCRYPT 2014, Lect. Notes Comput. Sci., **8895**, 2014, 259–273.

[17] Abdelkhalek A., AlTawy R., Youssef A. M., "Impossible differential properties of reduced round Streebog", C2SI 2015, Lect. Notes Comput. Sci., **9084**, 2015, 274–286.

[18] 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", IWSEC 2015, Lect. Notes Comput. Sci., **9241**, 2015, 79–96.

[19] Rongjia Li, Chenhui Jin, Ruya Fan, "Improved integral distinguishers on compression function of GOST R hash function", *Computer Journal*, **62** (2019), 535–544.

[20] Tingting Cui, Wei Wang, Meiqin Wang, "Distinguisher on full-round compression function of GOST R", *Information Processing Letters*, **156** (2020), 105902.

[21] Kazymyrov O., Kazymyrova V., "Algebraic aspects of the Russian hash standard GOST R 34.11-2012", *Cryptology ePrint Archive, Report 2013/556*, 2013.

[22] Dinur I., Leurent G., "Improved generic attacks against hash-based MACs and HAIFA", CRYPTO 2014, Lect. Notes Comput. Sci., **8616**, 2014, 149–168.

[23] Kiryukhin V., "Streebog compression function as PRF in secret-key settings", CTCrypt 2021, *Mat. Vopr. Kriptogr.*, **13**:2 (2022), 99–116, https://eprint.iacr.org/2022/118.pdf.

[24] Mendel F., Rechberger C., Schläffer M., Søren S. Thomsen, "The rebound attack: cryptanalysis of relduced Whirlpool and Grøstl", FSE 2009, Lect. Notes Comput. Sci., **5665**, 2009, 260–276.

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. Fix the position of columns in $\boldsymbol{\Delta HW_3}$ and $\boldsymbol{\Delta K_5}$.

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 is $1 - (1 - p^8)^{255} \approx 2^{-2.2}$.

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 on average of $z^{64} \approx 2^{80.4}$ solutions.

We have $255^8 \approx 2^{64}$ possible values $\boldsymbol{\Delta HZ_4}$.

Repeat for all pairs of columns in $\boldsymbol{\Delta HW_3}$ and $\boldsymbol{\Delta K_5}$.

As a result we obtain about

$$8 \cdot 8 \cdot 2^{64+80.4-17.6} \approx 2^{132}$$

valid states HY_4.

B Detailed pictures for 11-round attack

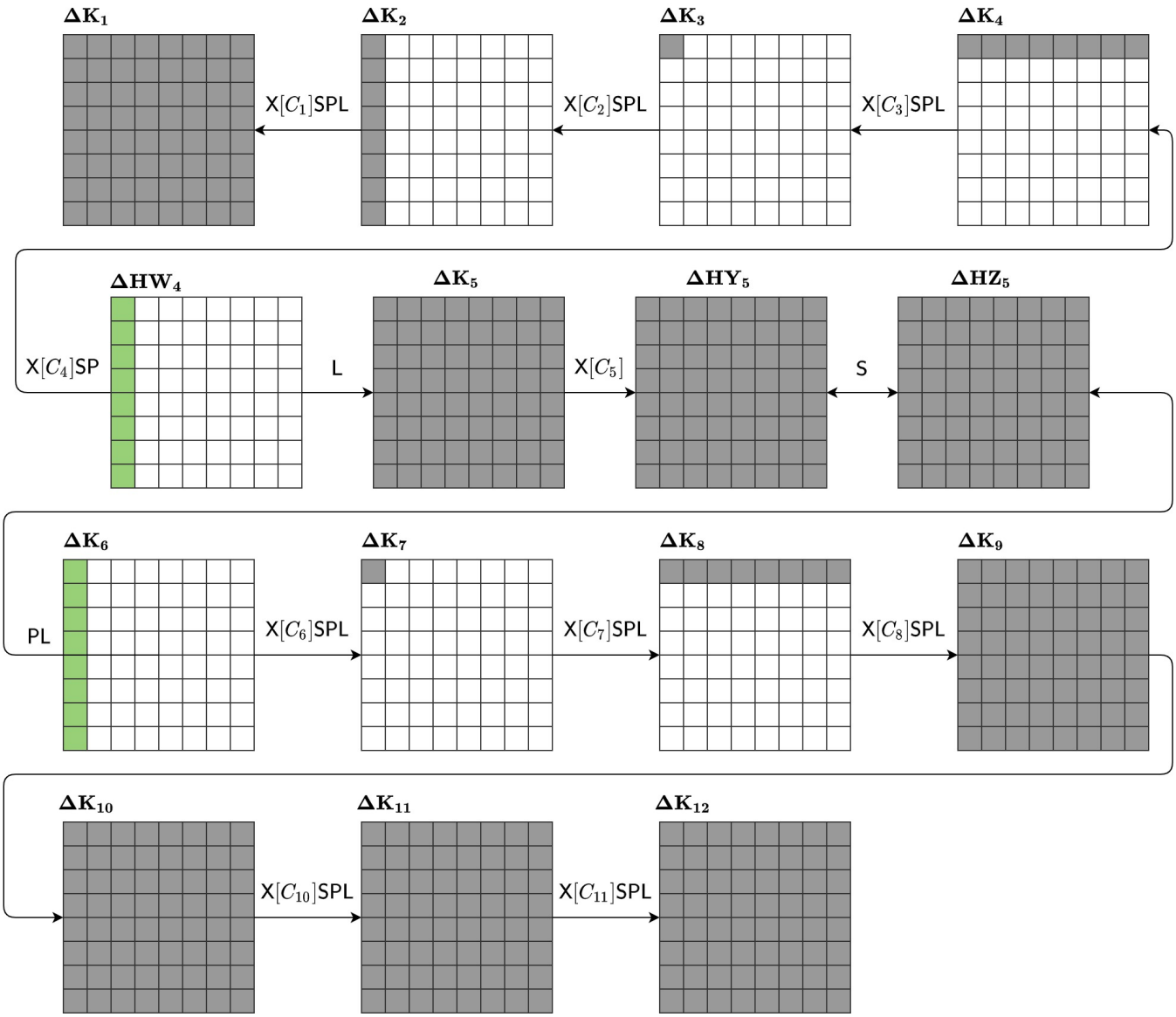

Figure 9: Offline stage. Rebound.

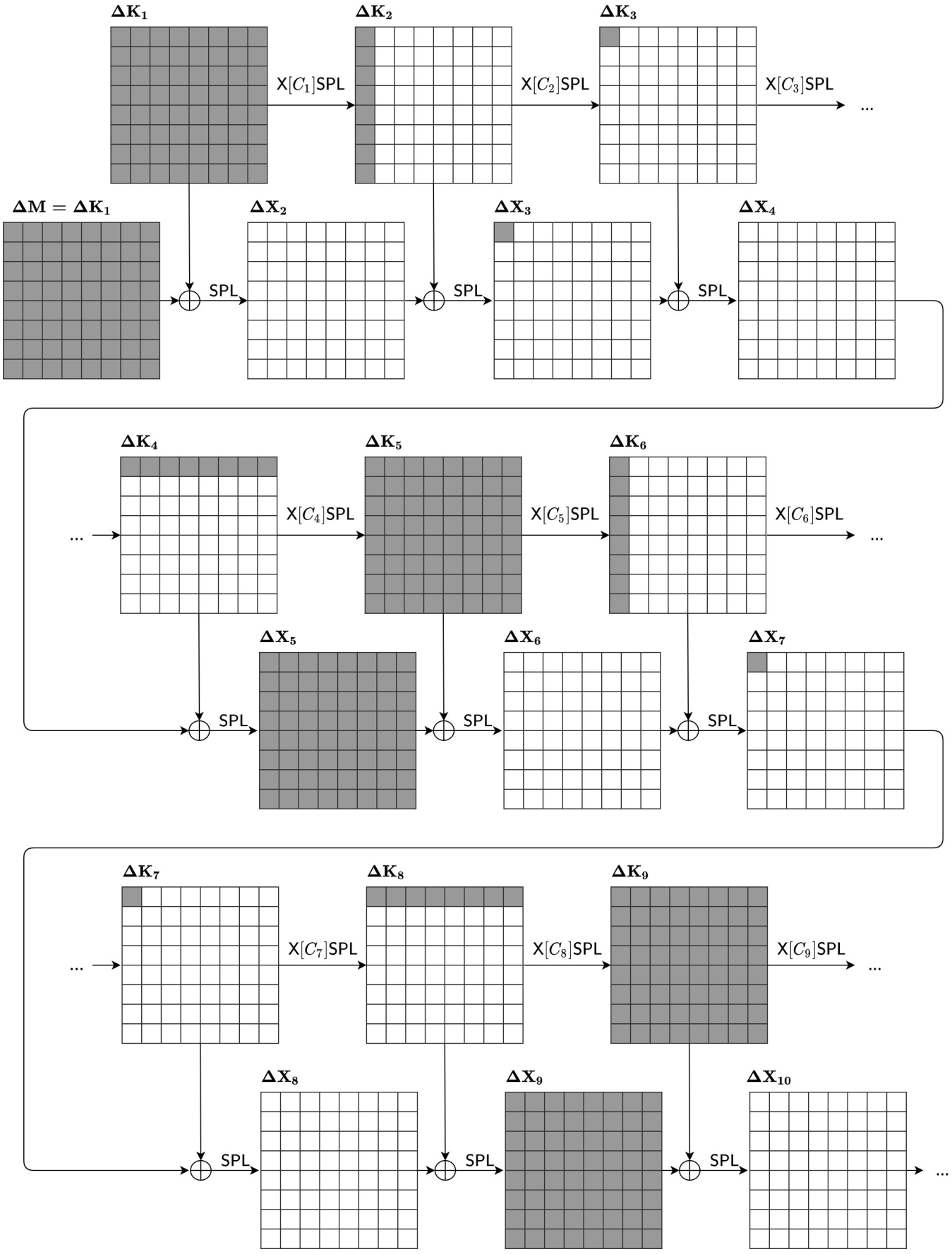

Figure 10: The truncated related-key differential trail. 9 rounds.

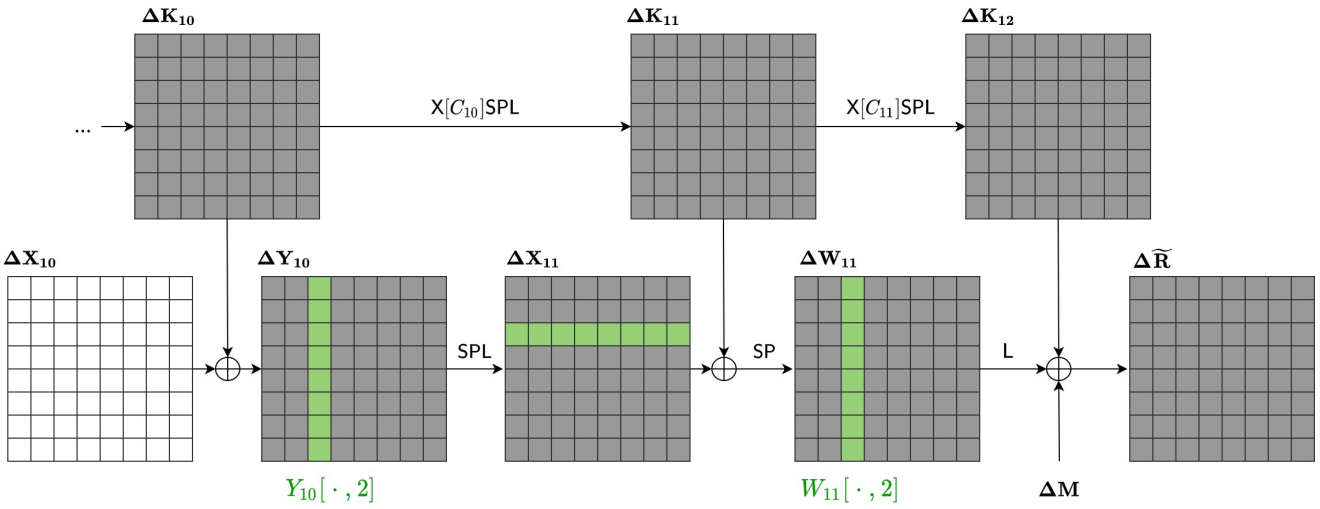

Figure 11: Additional precomputation at the offline stage.