# About "$k$-bit security" of MACs based on hash function Streebog

Vitaly Kiryukhin

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

**Abstract**

Various message authentication codes (MACs), including HMAC-Streebog and Streebog-K, are based on the keyless hash function Streebog. Under the assumption that the compression function of Streebog is resistant to the related key attacks, the security proofs of these algorithms were recently presented at CTCrypt 2022.

We carefully detail the resources of the adversary in the related key settings, revisit the proof, and obtain tight security bounds. Let $n$ be the bit length of the hash function state. If the amount of processed data is less than about $2^{n-k}$ blocks, then for HMAC-Streebog-512 and Streebog-K, the only effective method of forgery (or distinguishing) is guessing the $k$-bit secret key or the tag if it is shorter than the key. So, we can speak about "$k$-bit security" without specifying the amount of material, if the key length is no longer than half of a state. The bound for HMAC-Streebog-256 is worse and equal to $2^{\frac{n}{2}-k}$ blocks.

**Keywords:** Streebog, PRF, HMAC, provable security

## 1 Introduction

Russian hash function Streebog [1] is based on a modified Merkle-Damgård (MD) approach [6, 7]. The latter, as is well known, includes: padding the message $M$ and splitting it into $b$-bit blocks; iteratively applying the compression function $\mathsf{g}$ to the message block and the $n$-bit previous state; the initial state is the predefined constant; the last state is the result of hashing. Streebog uses $n = b = 512$, and its compression function is 12-rounds AES-like block cipher in Miyaguchi-Preneel mode. The output length can be either $\tau = 512$ or 256 bits.

Streebog has two features that differentiate it from the "plain" MD cascade:

– before processing the $i$-th block, the state is summed modulo 2 with the number of already hashed bits;

– the last call of the compression function is used to “mix” the checksum (modulo $2^n$) of all message blocks.

These features play an important role, especially when $\mathsf{Streebog}$ is used as the core for a keyed cryptoalgorithm, for example, a message authentication code (MAC) or a pseudorandom function (PRF).

Perhaps the most widespread and well-known way to construct a keyed transformation from a keyless hash function $\mathsf{H}$ is $\mathsf{HMAC}$ [8]

$$\mathsf{HMAC}(K, M) = \mathsf{H}\left((\overline{K} \oplus \mathit{opad})||\mathsf{H}(\overline{K} \oplus \mathit{ipad}||M)\right),$$

where $\overline{K}$ is obtained by padding the secret key $K$ with zero bits, *opad* and *ipad* are different nonzero constants. $\mathsf{Streebog}$ can also be used in $\mathsf{HMAC}$ [2, 3], but security proofs [8, 12, 11, 14, 17] were proposed for $\mathsf{HMAC}$ with the “plain” MD hash function and therefore cannot be directly applied for $\mathsf{HMAC\text{-}Streebog}$. The proof of the latter’s security was initially given in [16] and later detailed in [32] by the reduction to the properties of $\mathsf{g}$ and not to the hash function itself.

On the other hand, the features of $\mathsf{Streebog}$ give a rise to a more efficient keyed mode, namely $\mathsf{Streebog\text{-}K}$ (“Keyed Streebog”) [32]

$$\mathsf{Streebog\text{-}K}(K, M) = \mathsf{H}(\overline{K}||M),$$

where $\mathsf{H}$ is $\mathsf{Streebog}$ itself. Due to one hashing instead of two, the computation speed increases up to two times compared to $\mathsf{HMAC}$.

The checksum used in $\mathsf{Streebog}$ leads to many so-called related keys inside both $\mathsf{HMAC\text{-}Streebog}$ and $\mathsf{Streebog\text{-}K}$ even when these cryptoalgorithms themselves are used in the *single-key* setting. The input of the last call of $\mathsf{g}$ is the secret key $\overline{K}$ summed with the message’s blocks that are directly controlled by the adversary. As far as we know, by now there are no attacks for the compression function in the related-key setting that would be better than generic ones. Non-trivial results [23] were proposed only for the round-reduced version of $\mathsf{g}$.

Despite this, generic related-key attacks have the great impact on the security bounds. Guessing any one of the $q$ related keys (and therefore all of them due to known relations) can be $q$ times faster than guessing a single secret key. However, if different related keys are used to process different inputs, then the adversary should choose a specific key when guessing, not any one. This simple observation fortunately holds (sometimes partially) for MACs based on $\mathsf{Streebog}$.

We start by introducing the notation (section 2), and then provide high-level description of $\mathsf{Streebog}$ and the analyzed MACs (section 3).

Next, in section 4 we develop the $PRF\text{-}RKA$ threat model, which includes the above-mentioned observation by detailing the adversary's resources. We discuss the properties of $\mathsf{g}$ in the introduced model and give the corresponding heuristic estimates.

Only one observation is not enough to obtain a result, in section 5 we present a new security proof using the example of Streebog-K. Assuming "good" properties of $\mathsf{g}$ the obtained security bounds can be described quite simply: up to about $2^{n-k}$ processed blocks, the only effective method of forgery (or distinguishing) is guessing the $k$-bit secret key or the tag if it is shorter than the key. For a $k$-bit key, in any case, it should be assumed that the amount of data does not exceed $2^k$. Hence, if $k \le \frac{n}{2}$, then Streebog-K can be considered as "$k$-bit secure" without specifying the amount of material.

The attacks described in the sixth section demonstrate that the obtained estimates cannot be further improved (with the possible exception of a small multiplicative constant). In other words, each term in the upper bounds corresponds to a certain attack that has almost the same probability (the lower bound).

In the seventh section, similar results are given for HMAC-Streebog. Note that if the 256-bit version of Streebog is used in HMAC, then the bound is significantly worse ($2^{\frac{n}{2}-k}$).

## 2 Notations

We use the following notations throughout the paper:

$n = 512$ – block size in bits; $k \le 512$ – key size in bits; $\oplus$ – bitwise XOR operation; $\boxplus$, $\boxminus$ – addition and subtraction modulo $2^n = 2^{512}$; $||$ – concatenation of binary strings;

$V^n$ – the set of all $n$-bit strings with naturally defined operations "$\oplus$" and "$\boxplus$";

$\mathsf{sum}_{\boxplus}(M) = m_1 \boxplus m_2 \boxplus \ldots \boxplus m_l$ – the checksum (modulo $2^n$) of $l$ blocks from the padded message $M||\mathsf{10\ldots0} = m_1||m_2||...||m_l$;

$\mathrm{Func}(\mathbf{X}, \mathbf{Y})$ – the set of all mappings from the set $\mathbf{X}$ to the set $\mathbf{Y}$;

$X \overset{R}{\leftarrow} \mathbf{X}$ – uniform and random selection of element $X$ from the set $\mathbf{X}$.

The adversary is modeled by an interactive probabilistic algorithm that has access to other algorithms (oracles). We denote by $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A})$ a quantitative characterization (advantage) of the capabilities of the adversary $\mathcal{A}$ in realizing a certain threat, defined by the model $TM$, for the cryptographic scheme Alg. The resources of $\mathcal{A}$ are measured in terms of time ($t$) and query ($q$) complexities. The size of $\mathcal{A}$ description (its source code) is limited by some small value. The query complexity $q$ is measured in the number of adaptively chosen input/output pairs. Without loss of generality, we assume that $\mathcal{A}$ always uses exactly $q$ unique queries (with no redundant or repeating queries). The result of computations of $\mathcal{A}$ after interacting with oracle $\mathcal{O}$ is some binary value $x$, which is denoted as $\mathcal{A}^{\mathcal{O}} \Rightarrow x$.

The maximum of the advantage among all resource constrained adversaries is denoted by

$$\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t,q) = \max_{\mathcal{A}(t',q'):t' \le t,\ q' \le q,} \mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A}).$$

Some threat models, which would be addressed later, imply different types of resources, like the number of queries to different oracles, the length of these queries, etc. The advantage for such models is defined in similar way.

The cryptoalgorithm $\mathsf{Alg}$ is informally called secure in the threat model $TM$ ($TM$-secure) if $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t,q) < \varepsilon$, where $\varepsilon$ is some small value determined by the requirements for the strength of the cryptosystem and the resources $t$ and $q$ are comparable to those available to the adversary in practice.

To demonstrate the practical sense of the obtained results, we substitute heuristic estimates based on assumptions into derived security bounds. The resulting estimates are denoted by symbol "$\lessapprox$ " meaning "less or equal if the assumptions are true".

Next, we prove anew (see also [32]) that $\mathsf{HMAC\text{-}Streebog}$ and $\mathsf{Streebog\text{-}K}$ are secure presudorandom functions (PRF).

**Definition.** *The advantage of $\mathcal{A}$ in the model PRF (PRF-CMA – indistinguishability from a random function under chosen message attack) for the keyed cryptoalgorithm $\mathsf{F} : \mathbf{K} \times \mathbf{X} \to \mathbf{Y}$ is*

$$\mathrm{Adv}^{PRF}_{\mathsf{F}}(\mathcal{A}) = \Pr\left(K \overset{\mathrm{R}}{\leftarrow} \mathbf{K}; \mathcal{A}^{\mathsf{F}_K(\cdot)} \Rightarrow 1\right) - \Pr\left(\mathsf{R} \overset{\mathrm{R}}{\leftarrow} \mathrm{Func}(\mathbf{X},\mathbf{Y}); \mathcal{A}^{\mathsf{R}(\cdot)} \Rightarrow 1\right),$$

*where $\mathbf{K}$, $\mathbf{X}$, $\mathbf{Y}$ are spaces of the keys, messages, and outputs respectively. The resources of $\mathcal{A}$ are $t$ computations and $q$ queries to the oracle, $l$ the maximum length of the queries (in $n$-bit blocks) if elements from $\mathbf{X}$ have variable length.*

By "$k$-bit security" we informally mean that the probability of realizing a certain threat (or the distinguishing advantage) with the time complexity $t$ is about $t/2^k$. All our statements about "$k$-bit security" are, first of all, true for the distinguishers in the $PRF$ model, and, therefore, the same statements is true for more dangerous threats, including forgeries [9] and key recovery attacks.

## 3 Streebog and MACs

Streebog hashes the message $M$ as follows. The text is padded with bit string $\mathtt{10}\ldots\mathtt{0}$. At least one bit is always added, even if the message bit length $L < 2^n$ is already divisible by $n$. The string $M' = M||\mathtt{10}\ldots\mathtt{0}$ is divided into $(l+1)$ blocks of $n = 512$ bits $M' = m_0||m_1||\ldots||m_l$. The compression function is sequentially applied to the previous state, the block and the counter

$$h_{i+1} = \mathsf{g}(h_i, m_i, \mathbf{i}),\ i = 0, ..., l,\ h_0 = IV_\tau \in V^n,$$

where $IV_\tau$ is a predefined constant which is different in both versions of the hash function, $\tau \in \{256, 512\}$, the $n$-bit counter $\mathbf{i} = (i \cdot n)$ represents the number of already hashed bits.

Two more transformations are performed at the finalizing stage: the bit length $L$ and the checksum $\varSigma = \mathsf{sum}_{\boxplus}(M) = m_0 \boxplus ... \boxplus m_l$ are "mixed" with the state

$$h_{l+2} = \mathsf{g}(h_{l+1}, L, \mathbf{0}),\ H = \mathsf{g}(h_{l+2}, \varSigma, \mathbf{0}).$$

If 256-bit hash function is used, the output $H$ is truncated to 256 bit. Here and further, the integers at the input of $\mathsf{g}$ are implicitly converted into $n$-bit vectors.

The compression function is based on a 12-rounds AES-like block cipher $\mathsf{E}$ in Miyaguchi-Preneel mode

$$\mathsf{g}(h_i, m_i, \mathbf{i}) = \mathsf{E}(h_i \oplus \mathbf{i}, m_i) \oplus h_i \oplus m_i = h_{i+1}.$$

In [19], the equivalent representation was proposed (see also details in [32]). The counter $\mathbf{i}$ ceases to be a parameter of the compression function. The latter is simplified to $\mathsf{g}(h, m) = \mathsf{E}(h, m) \oplus h \oplus m$ (and further in the text, this is what is meant by $\mathsf{g}$). After processing the $i$-th block, the state is summed modulo 2 with the constant $\Delta_i = \mathbf{i} \oplus (\mathbf{i} \boxplus \mathbf{1})$, $i = 0, \ldots, l-1$, but after the $l$-th block, another constant is used, namely $\widetilde{\Delta}_l = \mathbf{l}$, and $\Delta_i \neq \widetilde{\Delta}_i$, $\forall i = 0, ..., 2^n - 1$. Thus, both versions of Streebog can be expressed as

$$\mathsf{H}_\tau(M) = \mathsf{msb}_\tau\left(\mathsf{g}(\mathsf{g}(\ldots(\mathsf{g}(\mathsf{g}(IV_\tau, m_0) \oplus \Delta_0, m_1) \oplus \Delta_1) \ldots \oplus \widetilde{\Delta}_l, L), \varSigma)\right),$$

where $\mathsf{msb}_\tau : V^n \to V^\tau$ is the $\tau$ most significant bits. Next, we omit the index $\tau$ if any of its values are suitable.

Various keyed cryptoalgorithms use Streebog in a black-box way, without making any changes to the Streebog itself, but only preparing the input for it. These algorithms are usually used as message authentication codes (MAC) and key derivation functions (KDF).

Here we list the formulas of the analyzed algorithms based on Streebog [2, 32], and also mention their features,

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

The secret key $K \in V^k$ is padded with $(n-k)$ zero bits if necessary $\overline{K} = (K||0...0)$. Two different $n$-bit constants $ipad \neq opad$ are used in HMAC-Streebog.

The key length for HMAC-Streebog, according to [2], is $256 \leq k \leq 512$ bits, and the same restriction is proposed in [32] for Streebog-K. Further, for generality, we assume that $k \leq n$.

HMAC-Streebog and Streebog-K can use both versions of H (with 256-bit and 512-bit output). Due to the double hashing in the first one, this leads to a significant impact on the security bounds.

Next, we describe our results using the example of Streebog-K, the last section and Appendix B are devoted to HMAC-Streebog.

For the sake of consistency with the previously introduced notation, let $\overline{K} = m_0$ and $M = m_1||...||m_l$ (fig. 1). By the cascade transformation Csc we mean further

$$\mathsf{Csc}(K_{\mathsf{Csc}}, M) = \mathsf{g}(\ldots \mathsf{g}(\mathsf{g}(K_{\mathsf{Csc}} \oplus \Delta_0, m_1) \oplus \Delta_1, m_2) \ldots \oplus \widetilde{\Delta}_l, L), \ K_{\mathsf{Csc}} \in V^n,$$

assuming that the input $M$ of arbitrary bit length is padded by $\mathtt{10\ldots0}$, and the length $L$ increases by $n$ because of the key.

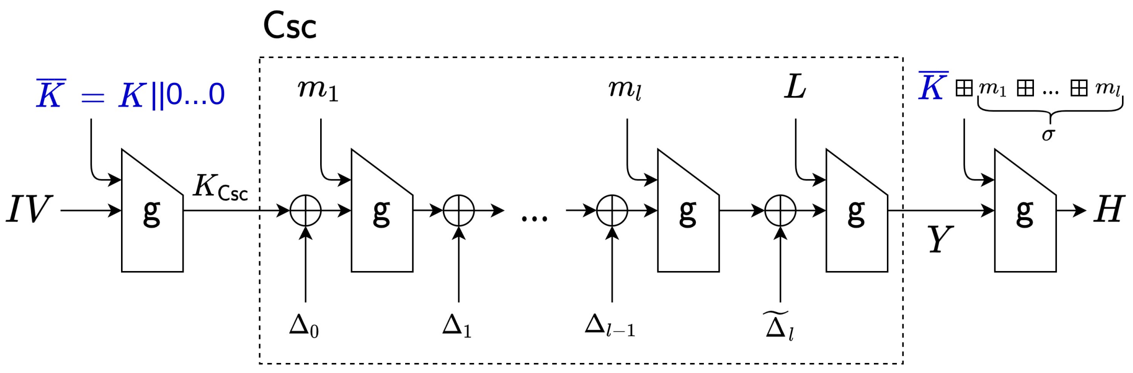

Figure 1: The equivalent representation of Streebog-K. The checksum is $\Sigma = \overline{K} \boxplus \sigma$, and $\sigma = \mathsf{sum}_{\boxplus}(M)$. The result of cascade is $Y = \mathsf{Csc}(K_{\mathsf{Csc}}, M)$.

## 4 Related key settings

For all the considered cryptoalgorithms, security is reduced to the properties of the compression function $\mathsf{g}$ under the various related key attacks (RKA). We capture all the required properties in the following definition.

**Definition**. *The advantage of $\mathcal{A}$ in the model $PRF\text{-}RKA_{\circledast}$ for the keyed cryptoalgorithm $\mathsf{F} : \mathbf{K} \times \mathbf{X} \to \mathbf{Y}$ is*

$$\mathrm{Adv}_{\mathsf{F}}^{PRF\text{-}RKA_{\circledast}}(\mathcal{A}) = \Pr\left(K \xleftarrow{\mathrm{R}} \overline{\mathbf{K}}; \mathcal{A}^{\mathsf{F}_{K\circledast\cdot}(\cdot)} \Rightarrow 1\right) -$$
$$- \Pr\left(K \xleftarrow{\mathrm{R}} \overline{\mathbf{K}}; \mathsf{R}_i \xleftarrow{\mathrm{R}} \mathrm{Func}(\mathbf{X}, \mathbf{Y}),\ \forall i \in \mathbf{K}; \mathcal{A}^{\mathsf{R}_{K\circledast\cdot}(\cdot)} \Rightarrow 1\right),$$

*where $\mathbf{K}$, $\mathbf{X}$, $\mathbf{Y}$ are spaces of the keys, messages, and outputs respectively. The subset $\overline{\mathbf{K}} \subseteq \mathbf{K}$ and the $w$-ary operation "$\circledast$" are the parameters of the model. The query from $\mathcal{A}$ consists of the input $x \in \mathbf{X}$ and the relation $\kappa \in \mathbf{K}^{w-1}$. The response is the value $y = \mathsf{F}_{K\circledast\kappa}(x)$ (resp. $y = \mathsf{R}_{K\circledast\kappa}(x)$). The resources of $\mathcal{A}$ are $t$ computations and $q$ queries to the oracle. The content of queries is limited by the number of relations ($r$) and by the number different relations ($d$) queried with the same $x$ ($d \le r \le q$).*

We omit $d$ in the notations if $d \le r$, and also omit $r$ if $r \le q$.

Note, if "$\circledast$" is the unary identity operation, then $PRF\text{-}RKA_{\circledast}$ is essentially the same as the usual $PRF$ model.

Through the paper we instantiate the $PRF\text{-}RKA_{\circledast}$ model using binary operations "$\oplus$" and "$\boxplus$" over $V^n$. The $\mathsf{HMAC\text{-}Streebog}$ analysis also required the introduction of a ternary operation, denoted by "$\boxplus \circ \oplus$" (so, the key $K$ under the relation $\kappa = (\phi, \sigma)$ is $(K \oplus \phi) \boxplus \sigma$). Further in the text, "$\circledast$" denotes any of these three operations.

The main novelty introduced in the above definition is the parameter $d$. We show its importance for the generic attacks against arbitrary PRF $\mathsf{F} : \mathbf{K} \times \mathbf{X} \to \mathbf{Y}$. Let, for example, $\circledast = \boxplus$ and $\overline{\mathbf{K}} = \mathbf{K} = V^k$.

In the absence of restrictions ($d = r = q$), the adversary can query a single $x$ under the different relations $(x, \kappa_1)$,...,$(x, \kappa_q)$ and obtain $y_1$,...,$y_q$, $y_i = \mathsf{F}_{K\boxplus\kappa_i}(x)$. Next, the adversary repeats $t$ times: guess the key $\tilde{K}$; compute $\tilde{y} = \mathsf{F}_{\tilde{K}}(x)$; find $\tilde{y}$ among the stored $(y_1, ..., y_q)$. If $\tilde{K}$ is equal to ***any*** of the related keys used $(K \boxplus \kappa_1, ..., K \boxplus \kappa_q)$, let this be $K \boxplus \kappa_i$, then certainly $\tilde{y} = y_i$. Hence, the attacker obtains $\tilde{K} = K \boxplus \kappa_i$ and computes the key $K = \tilde{K} \boxminus \kappa_i$ using the known relation $\kappa_i$. The success probability is upper bounded by $t \cdot q \cdot 2^{-k}$. False positives $\tilde{y} \in (y_1, ..., y_q)$ do not affect the essence and are ignored here.

Now let $d = 1$ and $r \le q$. In this case, the queries are $(x_1, \kappa_1)$,...,$(x_q, \kappa_q)$. Regardless of the relations, all inputs are different, $x_i \ne x_j$, $1 \le i < j \le q$.

Therefore, before the guessing attempt, the adversary can choose only one key (i.e. under one relation) that he will try to find. In other words, he can compute $\tilde{y} = \mathsf{F}_{\tilde{K}}(x_i)$ and check $\tilde{y} = y_i$, hoping that $\tilde{K} = K \boxplus \kappa_i$. Matches with other keys $K \boxplus \kappa_j$, $j \neq i$, cannot be verified because of $x_i \neq x_j$. Therefore, in such conditions, the success probability is about $t \cdot 2^{-k}$ and does not depend on the total number of queries.

Thus, assuming the absence of specific vulnerabilities, the best distinguishing method is key guessing, and the advantage is bounded by

$$\mathrm{Adv}_{\mathsf{F}}^{PRF\text{-}RKA_{\boxplus}}(t,q,r,d) \lessapprox \frac{t \cdot d}{2^k} \leq \frac{t \cdot r}{2^k} \leq \frac{t \cdot q}{2^k}.$$

Note that the presented estimates are heuristic in nature. However, these inequalities are easy to prove if $\mathsf{F}$ is considered as a family of $2^k$ random functions (i.e. in the so-called random oracle model). The same considerations make it possible to ignore attacks based on "free" precomputations [15].

To the best of our knowledge, there are no attacks on the Streebog compression function that would be better than the generic ones. The round-reduced versions of $\mathsf{g}$ were considered in the secret-key [21, 22] and the related-key settings [23]. The situation is similar with the keyless settings (preimages and various collisions) [24, 25, 26, 27, 28, 29, 30, 31], that is also an indirect argument in favor of good cryptographic properties under the related key attacks.

Therefore, we use

$$\mathrm{Adv}_{\mathsf{g}^{\triangledown}}^{PRF\text{-}RKA_{\circledast}}(t,q,r,d) \lessapprox \frac{t \cdot d}{2^k}, \tag{1}$$

as an heuristic estimate for $\mathsf{g}^{\triangledown}_{\overline{K}}(\cdot) = \mathsf{g}(\cdot, \overline{K})$, $k \leq n$, $\overline{\mathbf{K}} = \{\overline{K} : \ K \in V^k\}$, $\mathbf{K} = V^n$, instead of $t \cdot q \cdot 2^{-k}$ used in [32].

If the secret key of $\mathsf{g}$ is the $n$-bit state $h$ of the hash function, $\mathsf{g}^{\triangleright}_h(\cdot) = \mathsf{g}(h, \cdot)$, then the bound [32] remains the same

$$\mathrm{Adv}_{\mathsf{g}^{\triangleright}}^{PRF\text{-}RKA_{\oplus}}(t,q,r \leq 2) \lessapprox \frac{2 \cdot t}{2^n} + \frac{q(q-1)}{2^{n+1}}. \tag{2}$$

Recall that, for all the cryptoalgorithms under consideration, $\mathsf{g}^{\triangleright}$ is used with no more than two related keys ($d \leq r \leq 2$). The relation is defined by $\phi_i = \widetilde{\Delta}_i \oplus \Delta_i$, $i = 1, ..., l$. The second term in (2) arises due to the birthday-paradox distinguisher (see also [22]).

We emphasize that some new effective attacks on full-round $\mathsf{g}$ can potentially affect the heuristic estimates and, consequently, the claims about "$k$-bit security". We are convinced that the construction of such methods is extremely difficult, but anyway the theorems presented below hold. Their statements are essentially about the high-level design of the Streebog-based cryptoalgorithms, and not the hash function itself.

## 5 Revision of PRF-security for Streebog-K

The PRF-security bound of Streebog-K (denoted here for compactness as KH) presented in [32] as

$$\mathrm{Adv}_{\mathsf{KH}}^{PRF}(t,q,l) \leq \mathrm{Adv}_{\mathsf{g}^{\triangledown}}^{PRF\text{-}RKA_{\boxplus}}(t',q',r=q',d=q')+ \\ +q\cdot l'\cdot \mathrm{Adv}_{\mathsf{g}^{\triangleright}}^{PRF\text{-}RKA_{\oplus}}(t',q,r=2)+\frac{q^2+q}{2^{n+1}}, \qquad (3)$$

where $t'=t+O(q\cdot l)$, $q'=q+1$, $l'=l+1$.

Recall that the $i$-th message $M_i$ is transformed as follows

$$\mathsf{KH}_K(M_i)=\mathsf{g}^{\triangledown}_{\overline{K\boxplus\sigma_i}}(\mathsf{Csc}(\mathsf{g}^{\triangledown}_{\overline{K}}(IV),M_i)),$$

and $K_{\mathsf{Csc}}=\mathsf{g}^{\triangledown}_{\overline{K}}(IV)$, the result of the cascade $Y_i=\mathsf{Csc}(K_{\mathsf{Csc}},M_i)$, the relation is determined by $\sigma_i=\mathsf{sum}_{\boxplus}(M_i)$, $1\leq i\leq q$, (see fig. 1).

By "collision" in this section we mean the coincidence of any pair of elements in the sequence $IV, Y_1, Y_2, ..., Y_q$, and denote it as C, the opposite event is denoted by $\overline{\mathrm{C}}$.

The term $\mathrm{Adv}_{\mathsf{g}^{\triangledown}}^{PRF\text{-}RKA_{\boxplus}}$ is the most significant when $k<n$, and is the only term that depends on the key length $k$. With $d=q'$, the $PRF\text{-}RKA_{\boxplus}$ model allows the inputs $(x,\sigma)$ for $\mathsf{g}^{\triangledown}$ to have *any* form, because of this, the estimate increases by $t\cdot q'\cdot 2^{-k}$. Whereas in fact, until there has been a "collision", the values of $x$ in all queries to $\mathsf{g}^{\triangledown}$ are *different* ($d=1$). Otherwise ("collision" has occurred), the attacker has already achieved his goal, the probability of this is taken into account in the third term of the bound.

The formalization of the above considerations is expressed by the following theorem.

**Theorem (PRF-security of Streebog-K).** *The advantage of the adversary in the PRF model attacking* Streebog-K *is bounded by* $\mathrm{Adv}_{\mathsf{KH}}^{PRF}(t,q,l)\leq$

$$\leq \mathrm{Adv}_{\mathsf{g}^{\triangledown}}^{PRF\text{-}RKA_{\boxplus}}(t',q',q',d=1)+\mathrm{Adv}_{\mathsf{Csc}}^{PRF}(t',q,l')+\frac{q^2+q}{2^{n+1}}, \qquad (4)$$

*where* $t'=t+O(q\cdot l)$, $q'=q+1$, $l'=l+1$.

Proof.

Let's consider $\widetilde{\mathsf{KH}}(M_i) = \mathsf{f}_{\overline{K}\boxplus\sigma_i}(\mathsf{Csc}(\mathsf{f}_{\overline{K}\boxplus 0}(IV), M_i))$, the first and the last calls of $\mathsf{g}^{\nabla}$ are replaced in $\mathsf{KH}$ by a family of $2^n$ random functions $\mathsf{f}$ indexed by $\sigma_i$. If the "collision" does not occur, then the cascade key $K_{\mathsf{Csc}} = \mathsf{f}_{\overline{K}\boxplus 0}(IV)$ is not observed by the attacker and is truly random. Moreover, under the same conditions, $\widetilde{\mathsf{KH}}$ is indistinguishable from a random function $\mathsf{R}$, due to the fact that regardless of $\sigma_i$, all values $IV, Y_1, ..., Y_q$ requested from $\mathsf{f}$ are not repeated.

The "collision" that occurred as a result of the adversary's interaction with $\mathsf{KH}$ is denoted as $\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (b, \mathrm{C})$. What we mean by this is that $b \in \{0, 1\}$ is the immediate result returned by the adversary, and the "collision" is an implicit side result. Note that anyone who knows $K_{\mathsf{Csc}}$ can easily determine whether there was the "collision" or not. We omit $b$ in the notation if its value can be any, $\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow \mathrm{C}) = \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \mathrm{C})) + \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (0, \mathrm{C}))$.

By the definition of the $PRF$ model,

$$\mathrm{Adv}^{PRF}_{\mathsf{KH}}(\mathcal{A}) = \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow 1) - \Pr(\mathcal{A}^{\mathsf{R}(\cdot)} \Rightarrow 1).$$

Using the formula of total probability, we get

$$\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow 1) = \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \mathrm{C})) + \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})).$$

As we explained above,

$$\Pr(\mathcal{A}^{\mathsf{R}(\cdot)} \Rightarrow 1) = \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})).$$

By grouping the terms and using the triangle inequality, we obtain $\mathrm{Adv}^{PRF}_{\mathsf{KH}}(\mathcal{A}) =$

$$\begin{aligned}
&= \Big(\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \mathrm{C})) + \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}}))\Big) - \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) \le \\
&\le \Big(\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) - \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}}))\Big) + \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \mathrm{C})) \le \\
&\le \Big(\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) - \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}}))\Big) + \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow \mathrm{C}) \le \\
&\le \Big(\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) - \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}}))\Big) + \\
&\quad + \Big(\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow \mathrm{C}) - \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow \mathrm{C})\Big) + \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow \mathrm{C}) = \\
&= \epsilon + \epsilon_{coll} + p_{coll}.
\end{aligned}$$

Let's use both $\epsilon$ and $\epsilon_{coll}$ at the same time. In other words, we utilize in the single algorithm $\mathcal{B}_1$: the ability of $\mathcal{A}$ to distinguish between $\mathsf{KH}$ and $\widetilde{\mathsf{KH}}$ when there are no collisions (term $\epsilon$); the advantage $\epsilon_{coll}$ arising from the difference in the probability of collisions. We assume that $\epsilon \geq 0$ and $\epsilon_{coll} \geq 0$, otherwise, we can invert the corresponding result.

$\mathcal{B}_1$ attacks $\mathsf{g}^{\triangledown}$ in the $PRF\text{-}RKA_{\boxplus}$ model. Initially, $\mathcal{B}_1$ queries the cascade key $K_{\mathsf{Csc}} = \mathcal{O}(IV, 0)$ from the oracle $\mathcal{O} \in \{\mathsf{g}^{\triangledown}, \mathsf{f}\}$. When processing each query $M_i$ from $\mathcal{A}$, the algorithm $\mathcal{B}_1$ computes $Y_i = \mathsf{Csc}(K_{\mathsf{Csc}}, M_i)$ and $\sigma_i = \mathsf{sum}_{\boxplus}(M_i)$. The value of $Y_i$ is written in memory. Next, $\mathcal{B}_1$ checks the "collision" condition. If $Y_i \in \{IV, Y_1, ..., Y_{i-1}\}$ then $\mathcal{B}_1$ returns 1 (due to $\epsilon_{coll} \geq 0$, the "collision" is interpreted as an interaction with $\mathsf{KH}$) and turns off $\mathcal{A}$ (further interaction does not make sense). Otherwise ($Y_i \notin \{IV, Y_1, ..., Y_{i-1}\}$), $\mathcal{B}_1$ makes query $(Y_i, \sigma_i)$ to the oracle and transmits the response to $\mathcal{A}$. If the "collision" conditions have never been met after $q$ queries, then the result of $\mathcal{B}_1$ is the result of $\mathcal{A}$.

The computation resources of $\mathcal{B}_1$ is $t' = t + O(q \cdot l)$, no more than $(q+1)$ queries are made to the oracle, the number of the related keys $r \leq q+1$, no value is requested from the oracle twice, $d = 1$.

Until the "collision" occurs, $\mathcal{B}_1$, interacting with $\mathsf{g}^{\triangledown}$ or $\mathsf{f}$, perfectly simulates for $\mathcal{A}$ oracles $\mathsf{KH}$ or $\widetilde{\mathsf{KH}}$ respectively. The distinguishing advantage of $\mathcal{B}_1$ is equal to

$$\begin{aligned}\mathrm{Adv}_{\mathsf{g}^{\triangledown}}^{PRF\text{-}RKA_{\boxplus}}(\mathcal{B}_1) &= \Pr(\mathcal{B}_1^{\mathsf{g}^{\triangledown}_{\overline{K}\boxplus\cdot}(\cdot)} \Rightarrow 1) - \Pr(\mathcal{B}_1^{\mathsf{f}_{\overline{K}\boxplus\cdot}(\cdot)} \Rightarrow 1) = \\ &= \Big(\Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) + \Pr(\mathcal{A}^{\mathsf{KH}(\cdot)} \Rightarrow \mathrm{C})\Big) - \\ &- \Big(\Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) + \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow \mathrm{C})\Big) = \epsilon + \epsilon_{coll}.\end{aligned}$$

All that remains is to limit the value of $p_{coll} = \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow \mathrm{C})$. We construct the algorithm $\mathcal{B}_2$ that can effectively distinguish $\mathsf{Csc}$ from a random function $\mathsf{R}$. $\mathcal{B}_2$ passes the request $M_i$ from $\mathcal{A}$ to its own oracle $\mathcal{O} \in \{\mathsf{Csc}, \mathsf{R}\}$, receives the response $Y_i$ and stores this value in memory. When processing each query, $\mathcal{B}_2$ checks if the "collision" occurred. If it did, $\mathcal{B}_2$ returns 1 and turns off $\mathcal{A}$. Otherwise, $\mathcal{B}_2$ generates a random value (simulates $\mathsf{f}$) and returns it to $\mathcal{A}$. If there is no "collision" after $q$ queries, then the result of $\mathcal{B}_2$ is 0. The advantage of $\mathcal{B}_2$ is lower bounded by

$$\begin{aligned}\mathrm{Adv}_{\mathsf{Csc}}^{PRF}(\mathcal{B}_2) &= \Pr(\mathcal{B}_2^{\mathsf{Csc}(K_{\mathsf{Csc}},\cdot)} \Rightarrow 1) - \Pr(\mathcal{B}_2^{\mathsf{R}(\cdot)} \Rightarrow 1) \geq \\ &\geq \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow \mathrm{C}) - \left(\frac{q \cdot (q-1)}{2^{n+1}} + \frac{q}{2^n}\right),\end{aligned}$$

where $\frac{q}{2^n}$ is the probability of $IV \in \{Y_1, ..., Y_q\}$, and $\frac{q \cdot (q-1)}{2^{n+1}}$ corresponds to the collision among $q$ values $Y_1, ..., Y_q$ returned from the random oracle

$\mathsf{R}(\cdot)$. Therefore,

$$\Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}(\cdot)} \Rightarrow \mathrm{C}) \leq \mathrm{Adv}_{\mathsf{Csc}}^{PRF}(\mathcal{B}_2) + \frac{q^2+q}{2^{n+1}}.$$

$\mathcal{B}_2$ appends a block containing $L$ to the messages from $\mathcal{A}$, so that because of the extra block we have $l' = l + 1$. $\square$

The PRF-security of the cascade $\mathsf{Csc}$ is proved in [32] as a separate lemma

$$\mathrm{Adv}_{\mathsf{Csc}}^{PRF}(t', q, l') \leq q \cdot l' \cdot \mathrm{Adv}_{\mathsf{g}^{\triangleright}}^{PRF\text{-}RKA_{\oplus}}(t', q, 2). \tag{5}$$

Direct substitution of the heuristic estimate (2) into (5) leads to an inaccurate result, due to the fact that (2) depends quadratically on $q$.

Hence, a more general bound, also stated in [32], is convenient for us here

$$\mathrm{Adv}_{\mathsf{Csc}}^{PRF}(\mathcal{A}) \leq \sum_{i=1}^{q}\sum_{j=1}^{l} \mathrm{Adv}_{\mathsf{g}^{\triangleright}}^{PRF\text{-}RKA_{\oplus}}(\mathcal{B}_{i,j}),$$

where $\mathcal{B}_{i,j}$ corresponds to some inner node of the special tree formed by queries. The root of the tree is $K_{\mathsf{Csc}}$, the nodes are the intermediate secret states, the results $(Y_1,...,Y_q)$ are stored in leaves. Each edge of the tree is labeled with the the block from the messages.

So, this tree has at least $l$ and at most $(1 + q \cdot (l-1)) \leq ql$ nodes (the multiplier in (5)). The lower bound is exact when all messages differ only in the last $l$-th block. The opposite is achieved when all messages are different in the first block. The adversary $\mathcal{B}_{i,j}$ makes $1 \leq q_{i,j} \leq q$ queries to the oracle, and $q_{i,j}$ also corresponds to the number of the edges from the node. So if $q_{i,j}$ increases by one, then the number of leaves also becomes one more, but there are no more leaves in total than $q$. Thus, we have inequality

$$\sum_{i}\sum_{j}(q_{i,j} - 1) \leq q,$$

and the PRF-security of the cascade is estimated as $\mathrm{Adv}_{\mathsf{Csc}}^{PRF}(t', q, l') \lessapprox$

$$\lessapprox \sum_{i=1}^{q}\sum_{j=1}^{l'} \left(\frac{2 \cdot t'}{2^n} + \frac{q_{i,j} \cdot (q_{i,j} - 1)}{2^{n+1}}\right) \leq \frac{2 \cdot t' \cdot q \cdot l'}{2^n} + \frac{q^2}{2^{n+1}}. \tag{6}$$

By using the result (4) of the theorem and the estimates (1), (6), we finally obtain

$$\mathrm{Adv}_{\mathsf{KH}}^{PRF}(t, q, l) \lessapprox \frac{t'}{2^k} + \frac{2 \cdot t' \cdot q \cdot l'}{2^n} + \frac{q^2+q}{2^n},\ t' \approx t,\ l' = l + 1, \tag{7}$$

and make a claim about "$k$-bit security". If the key length $\frac{n}{2} \le k \le n$ and the amount of the processed blocks $q \cdot l < 2^{n-k-1}$, then the most significant term is $\frac{t}{2^k}$ the probability of successfully guessing the key. Obviously, the data constraint $q \cdot l < 2^k$ is always assumed. Hence, for a key of shorter length $k \le \frac{n}{2}$, again, the most significant term is the first one.

The statement above concerns distinguishing attacks, but the same holds if the adversary's goal is to forge. The *probability* of at least one successful forgery in $\nu$ attempts is bounded by [9, Proposition 7.3] (SUF – Strong UnForgeablility)

$$\mathrm{Adv}_{\mathsf{KH}}^{SUF}(t, q, l, \nu) \le \mathrm{Adv}_{\mathsf{KH}}^{PRF}(t', q + \nu, l) + \frac{\nu}{2^\tau},\ t' \approx t.$$

So, if the output length is sufficiently large ($\tau \ge k$), then the statement about "$k$-bit security" is also true in this case. For small $\tau < k$, we make a reservation that another attack strategy is tag guessing.

We emphasize that exceeding the border of $2^{n-k}$ blocks may be quite *acceptable*. The probability of a forgery in one attempt is greater than "ideal" $2^{-\tau}$, but in most practical cases it is *negligible*, even if the number of the processed blocks far exceeds $2^{n-k}$.

Note for completeness that the bound similar to (7) can be obtained by using results of [14, 17] for the HMAC with the "plain" MD hash function, say HMAC-SHA-512 [4]. However, SHA-512($K||\cdot$), unlike Streebog, is completely insecure as PRF.

The "sponge"-based hash functions (for example, SHA-3 [5]) can be used with the key in the prefix as a secure PRF. The security bound for the keyed sponge [10, Theorem 1] is also close to (7) if we consider the sponge "capacity" $c$ as the state size $n$.

## 6 Attacks and tightness of the upper bounds

The attacks and the proofs are "the two sides of the same coin" [18]. The proofs give us the upper bounds of the insecurity (it can't be worse than that), the attacks provide the lower bounds (the attacker can definitely act with such an advantage or probability of success).

In this section we show that for Streebog-K "as a high-level design" both bounds are close. Therefore, (7) cannot be improved by more than a small multiplicative factor, so we can consider it tight enough.

On the contrary, we find it somewhat paradoxical that Streebog-K "as a real MAC with a real compression function g" may be even *more secure*.

Further, it can be refuted by some kind of sophisticated attack showing that (7) is tight for this case as well. Another way to refine the estimates seems to be random oracle model, which in simple words means an unconditional belief that $\mathsf{g}$ is a family of random functions.

The first term in (7) obviously corresponds to the straightforward key guessing. We should consequently also note that 4 computations of $\mathsf{g}$ are required to verify one key. Hence, the probability of success (correct guessing using $t$ computations) is 4 times less than the upper bound $t \cdot 2^{-k}$.

The third term in (7) is almost achievable with the birthday-paradox attack. The adversary: queries tags $H_i$ for the messages $M_i = m_i || m'_i$, $m_i \boxplus m'_i = const$, $1 \le i \le q$; looks for a collision ($H_i = H_j$); makes one additional query $M_i || P$, $P \in V^n$ and obtains the tag $H_{q+1}$; finally makes a forgery $M_j || P$ with tag $H_{q+1}$. The probability of a collision is about $q^2 \cdot 2^{-(n+1)}$, which is approximately half the upper estimate.

The reason for the above-mentioned "paradox" is the second term $t \cdot q \cdot l \cdot 2^{-n}$ arising from the imperfection of the cascade. If we consider time- and data-balanced attacks ($t \approx q \cdot l$), then the bound depends quadratically on the amount of data $q^2 \cdot l^2 \cdot 2^{-n}$. From this point of view, the best known attack is $l$ times worse. The probability of the birthday-paradox attack equal to $\approx q^2 \cdot l \cdot 2^{-n}$ if long $l$-block messages are used [20, 32]. We also don't know the matching attack for "real" Streebog-K if the computational power of the adversary is greater ($t \gg q \cdot l$).

The attack for the "plain" MD cascade (i.e. without counters and $\Delta_i$) with tight distinguishing advantage $t \cdot q \cdot l \cdot 2^{-n}$ can be easily constructed by using the properties of the random mapping graph, but in HMAC and Streebog-K the output of the cascade is not directly observed due to the key-dependent finalization. Counters also make attacks more difficult.

To demonstrate the accuracy of the upper estimates, we use a rather artificial trick proposed in [13, 14] for HMAC with the "plain" cascade. We construct a weak compression function $\mathsf{w}$ so that the cascade degrades as well as with $\mathsf{g}$, but with $\mathsf{w}$ it would be easily exploited in an attack (see details in Appendix A).

**Proposition**. *For any arbitrary compression function* $\mathsf{g}$ *and any resources* $(t, q)$ *of the adversary there exists "weak" function* $\mathsf{w}$*,*

$$\mathrm{Adv}_{\mathsf{w}^{\triangledown}}^{PRF\text{-}RKA_{\boxplus}} \approx 3 \cdot \mathrm{Adv}_{\mathsf{g}^{\triangledown}}^{PRF\text{-}RKA_{\boxplus}}, \ \mathrm{Adv}_{\mathsf{w}^{\triangleright}}^{PRF\text{-}RKA_{\oplus}} \approx 3 \cdot \mathrm{Adv}_{\mathsf{g}^{\triangleright}}^{PRF\text{-}RKA_{\oplus}} = 3 \cdot \epsilon^{\triangleright}.$$

*If* $\mathsf{w}$ *is used in* Streebog-K *instead of* $\mathsf{g}$*, then there is an attack with distinguishing advantage of about* $\frac{1}{2}\epsilon^{\triangleright} \cdot q \cdot l$*.*

So we can imply $\epsilon^{\triangleright} \approx t \cdot 2^{-n}$ and obtain the matching attack for the second term in (7).

It is interesting to note that $\mathsf{w}$ by construction depends on $\epsilon^{\triangleright}$, and hence on the resources of the adversary. All this only emphasizes the artificiality of the approach.

Nevertheless, one way or another, each of the three terms in (7) corresponds to an attack with approximately the same advantage. Hence, the proved security bound is tight.

## 7 HMAC-Streebog

The results obtained were presented using the example of Streebog-K, but the main ideas are applicable to other cryptoalgorithms. Here we briefly present our results for HMAC-Streebog, the proofs are given in Appendix B.

The relation between the keys is defined in HMAC-Streebog by "$\oplus$" and "$\boxplus$" simultaneously. When processing a message, up to 4 related keys are used, two of them are new for each message ($2 \cdot q$ in total), the two remaining ones ($\overline{K} \oplus ipad$ and $\overline{K} \oplus opad$) don't change. The value of $IV$ is queried at least twice under the different related keys ($\mathsf{g}^{\triangledown}_{\overline{K}\oplus ipad}(IV)$ and $\mathsf{g}^{\triangledown}_{\overline{K}\oplus opad}(IV)$), hence, in the $PRF\text{-}RKA_{\boxplus\circ\oplus}$ model, we are bounded by $d = 2$.

HMAC uses two hash function calls and consequently two cascades. When analyzing the "collision" event, we look at the outputs of two transformations at once, so two terms $\mathrm{Adv}^{PRF}_{\mathsf{Csc}}$ arise.

The "$k$-bit security" statement holds for HMAC-Streebog-512, in fact, as well as for Streebog-K. Whereas for the 256-bit version ($\tau = \frac{n}{2}$), the "inner" collision occurs after the first call of the hash function with the probability $\approx q^2 \cdot 2^{-\frac{n}{2}+1}$, that strongly affects the estimate. Thus, we can speak about "$k$-bit security" of HMAC-Streebog-256 only if $q \cdot l < 2^{\frac{n}{2}-k}$.

**Theorem (PRF-security of HMAC-Streebog).** *The advantage of the adversary in the PRF model attacking* HMAC-Streebog *is bounded by*

$$\mathrm{Adv}^{PRF}_{\mathsf{HMAC\text{-}Streebog}}(t, q, l) \leq \mathrm{Adv}^{PRF\text{-}RKA_{\boxplus\circ\oplus}}_{\mathsf{g}^{\triangledown}}(t', q', q', d = 2) + \\ + \mathrm{Adv}^{PRF}_{\mathsf{Csc}}(t', q, l') + \mathrm{Adv}^{PRF}_{\mathsf{Csc}}(t', q, l'_\tau) + \frac{2q^2 + q}{2^n} + \frac{q^2}{2^{\tau+1}},$$

*where* $t' = t + O(q \cdot l)$, $\tau \in \{256, 512\}$, $q' = 2 \cdot q + 2$, $l' = l + 1$, $l'_\tau \in \{2, 3\}$.

## 8 Conclusion

The security of Streebog-based MACs (including HMAC-Streebog and Streebog-K) as PRF and MAC in the single-key setting is reduced to the security of the compression function in the related key settings ($PRF\text{-}RKA$). We observed that, if the adversary does not query the same input under the different related keys, then the advantage is many times lower than in the general case. An appropriate refinement for the formal model was proposed, and then we re-proved the $PRF$-security of the mentioned MACs based on Streebog. The resulting security bounds are tight and cannot be significantly improved.

In fact, up to $2^{n-k}$ processed blocks, the only effective way of forgery (or distinguishing) is guessing the $k$-bit key or tag, $n = 512$ is the bit length of the hash function state. For HMAC-Streebog-256, this bound is worse and is equal to $2^{\frac{n}{2}-k}$. If the amount of data is much larger than $2^{n-k}$, then the probability of forgery remains insignificant for most practical cases, we just cannot talk about the "ideality".

The new estimates are especially important in practice for the Streebog-based MACs using relatively short keys (for example, 128 bit), and for some lightweight Streebog-like solutions.

The security proofs themselves use only the "standard model" without any heuristics. All statements about "$k$-bit security" are consequences that are obtained under the assumption of "good" properties of the compression function. The latter are confirmed by numerous negative results of cryptanalysis.

As always, we warn that the estimates do not take into account, say, side-channel attacks and other threats not included in the formal model. All the presented results are about adaptively chosen messages attacks in the single-key setting.

## 9 Acknowledgements

The author is grateful to Andrey Shcherbachenko and the anonymous reviewer(s) of CTCrypt 2023 for the careful consideration of the article and a lot of useful detailed comments and suggestions.

## 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] Smyshlyaev S., Alekseev E., Oshkin I., Popov V., Leontiev S., Podobaev V., Belyavsky D., "RFC 7836 - Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012", March 2016.

[4] *Secure Hash Standard (SHS) – NIST FIPS – 180-4*, 2015.

[5] *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions – NIST FIPS – 202*, 2015.

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

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

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

[9] Bellare M., Goldreich O., Mityagin A., "The power of verification queries in message authentication and authenticated encryption", *Cryptology ePrint Archive: Report 2004/304*, 2004.

[10] Bertoni G., Daemen J., Peeters M., Van Assche G., "On the security of the keyed sponge construction", Symmetric Key Encryption Workshop, **2011** (2011).

[11] Koblitz N., Menezes A., "Another look at HMAC", *J. Math. Cryptol.*, **7:3** (2013), 225–251.

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

[13] Krzysztof Pietrzak, "A Closer Look at HMAC", 2013.

[14] Gaži P., Pietrzak K., Rybár M., "The Exact PRF-Security of NMAC and HMAC", CRYPTO 2014, Lect. Notes Comput. Sci., **8616**, August 2014, 113–130.

[15] Bernstein D.J., Lange T., "Non-uniform cracks in the concrete: the power of free precomputation", ASIACRYPT 2013, Lect. Notes Comput. Sci., **8270**, 2013, 321–340.

[16] Alekseev E.K., Oshkin I.B., Popov V.O., Smyshlyaev S.V., "On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012", *Mat. Vopr. Kriptogr.*, **7**:1 (2016), 5–38.

[17] Nandi M., "A New and Improved Reduction Proof of Cascade PRF", *Cryptology ePrint Archive: Report 2021/097*, 2021.

[18] Bellare M., "Practice-Oriented Provable-Security", ISW 97, Lect. Notes Comput. Sci., **1396**, 1998, 221–231.

[19] 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.

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

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

[22] Kiryukhin V. A., "Streebog compression function as PRF in secret-key settings", *Mat. Vopr. Kriptogr.*, **13**:2 (2022), 99–116.

[23] Kiryukhin V. A., "Related-key attacks on the compression function of Streebog", *Mat. Vopr. Kriptogr.*, **14**:2 (2023), 59–76.

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

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

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

[27] 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.

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

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

[30] 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.

[31] Hua J., Dong X., Sun S., Zhang Z., Hu L., Wang X., "Improved MITM Cryptanalysis on Streebog", *Cryptology ePrint Archive, Paper 2022/568*, 2022.

[32] Kiryukhin V. A., "Keyed Streebog is a secure PRF and MAC", *Mat. Vopr. Kriptogr.*, **14**:2 (2023), 77–96.

## A Attack on the "weakened" cascade

Let for some arbitrary compression function $\mathsf{g}$

$$\mathrm{Adv}_{\mathsf{g}^{\triangledown}}^{PRF\text{-}RKA_{\boxplus}}(t,q,q,1) = \epsilon^{\triangledown},$$
$$\mathrm{Adv}_{\mathsf{g}^{\triangleright}}^{PRF\text{-}RKA_{\oplus}}(t,q,2) = \epsilon^{\triangleright},$$

As a special illustrative case, we can consider $\epsilon^{\triangledown} = t \cdot 2^{-k}$, $\epsilon^{\triangleright} = 2 \cdot t \cdot 2^{-n}$, and we also recall that $t \geq q \cdot l$.

We "spoil" two keys $P, P' \in V^n$ in $\mathsf{g}^{\triangledown}$ and $\epsilon^{\triangleright} \cdot 2^n$ keys in $\mathsf{g}^{\triangleright}$. Let $\mathbf{W} \subset V^n$ be the set of "weak" keys, $|\mathbf{W}| \approx \epsilon^{\triangleright} \cdot 2^n$. For any weak key $W \in \mathbf{W}$ and any constant $\Delta_i = \mathbf{i} \oplus (\mathbf{i} \boxplus \mathbf{1})$, $\mathbf{i} = i \cdot n$, we require that $(W \oplus \Delta_i) \in \mathbf{W}$. Each $\Delta_i$ belongs to the set

$$\boldsymbol{\Delta} = \{(2^u - 1) \cdot n,\ 1 \leq u \leq (n - \log_2(n))\} = \{\mathbf{1}, \mathbf{3}, \mathbf{7}, \mathbf{15}, ...\}.$$

We begin with an empty $\mathbf{W}$, choose an arbitrary $W \notin \mathbf{W}$, add elements from the following set to $\mathbf{W}$:

$$\{W\} \cup \{W \oplus \Delta;\ \Delta \in \boldsymbol{\Delta}\} \cup \{W \oplus \Delta \oplus \Delta';\ \Delta, \Delta' \in \boldsymbol{\Delta}\};$$

continue until there are less than $\epsilon^{\triangleright} \cdot 2^n$ elements in $\mathbf{W}$. At each iteration, no more than $n^2$ elements are added to $\mathbf{W}$. Therefore, the cardinality of $\mathbf{W}$ can differ from $\epsilon^{\triangleright} \cdot 2^n$ only by this insignificant value.

The "weakened" version of the compression function is defined as

$$\mathsf{w}(x,y) = \begin{cases} W_0, & x \in \mathbf{W} \text{ and } y \in \{P, P'\}, \\ \mathsf{g}(x,y), & \text{otherwise,} \end{cases}$$

where $W_0 \in \mathbf{W}$ is some fixed element. Thus, in total we redefine $2 \cdot |\mathbf{W}|$ values.

If the adversary does not interact with "weak" keys, then $\mathsf{g}^{\triangledown}$ and $\mathsf{w}^{\triangledown}$ (also as $\mathsf{g}^{\triangleright}$ and $\mathsf{w}^{\triangleright}$) are indistinguishable. In the first case ($\mathsf{w}^{\triangledown}_{\overline{K}}(\cdot) = \mathsf{w}(\cdot, \overline{K})$, $K \in V^k$), the interaction is carried out with $q$ related-keys. The probability that there are $P$ or $P'$ among them does not exceed a negligible $\frac{2 \cdot q}{2^k}$. Only two related keys are used in $\mathsf{w}^{\triangleright}_{h}(\cdot) = \mathsf{w}(h, \cdot)$, $h \in V^n$, the probability of their belonging to the set of weak ones is estimated as $2 \cdot \epsilon^{\triangleright}$. Thus, due to the appearance of weak keys, the distinguishing advantage in both cases increases slightly

$$\mathrm{Adv}^{PRF\text{-}RKA_{\boxplus}}_{\mathsf{w}^{\triangledown}}(t, q, q, 1) \leq \epsilon^{\triangledown} + \frac{2 \cdot q}{2^k} \leq 3 \cdot \epsilon^{\triangledown},$$
$$\mathrm{Adv}^{PRF\text{-}RKA_{\oplus}}_{\mathsf{w}^{\triangleright}}(t, q, 2) \leq \epsilon^{\triangleright} + 2 \cdot \epsilon^{\triangleright} = 3 \cdot \epsilon^{\triangleright}.$$

In distinguishing attack on "weakened" Streebog-K (instantiated with $\mathsf{w}$ instead of $\mathsf{g}$), $\frac{q}{2}$ pairs of queries $(M_i, M_i')$ to the oracle $\mathcal{O} \in \{\mathsf{Streebog\text{-}K}, \mathsf{R}\}$ are made

$$M_i = m_i^{(1)}||m_i^{(2)}||...||m_i^{(l-2)}||P||P',$$
$$M_i' = m_i^{(1)}||m_i^{(2)}||...||m_i^{(l-2)}||P'||P,$$

and the tags obtained are $H_i = \mathcal{O}(M_i)$, $H_i' = \mathcal{O}(M_i')$, $1 \leq i \leq \frac{q}{2}$. The blocks $m_i^{(j)}$ are randomly chosen from $\{P, P'\}$, $1 \leq j \leq l-2$. Note, that $M_i \neq M_i'$, but the first $(l-2)$ blocks, the lengths, and the checksums are equal.

In the case $\mathcal{O} = \mathsf{Streebog\text{-}K}$, we assume, as usual, that after processing the $j$-th block, the secret state looks random, but if the state falls into the set $\mathbf{W}$, then it remain as such until the end of the cascade. The longer the message being processed, the more likely it is that a weak key will occur during processing. We recall, that for all $W \in \mathbf{W}$ the following holds

$$\mathsf{w}(W, m_i^{(j)}) = W_0 \in \mathbf{W}, \quad W_0 \oplus \Delta_j \in \mathbf{W},$$
$$\mathsf{w}(W, P) = \mathsf{w}(W, P') = W_0 \in \mathbf{W}.$$

Hence, the probability of the collision $H_i = H_i'$ for one pair $(M_i, M_i')$ is about $\approx l \cdot \epsilon^{\triangleright}$, and for $\frac{q}{2}$ attempts we have $\approx \frac{q}{2} \cdot l \cdot \epsilon^{\triangleright}$. The collision event is used as a distinguishing feature.

If the attacker interacts with a random function $\mathsf{R}$, then the probability of at least one collision among $\frac{q}{2}$ independent attempts is upper bounded by $\frac{q}{2^n}$. We emphasize that the collision is considered only between messages in the same pair, and not among all possible pairs, and therefore the probability increases linearly, not quadratically. Hence, the distinguishing advantage is about $\left(\frac{q}{2} \cdot l \cdot \epsilon^{\triangleright} - \frac{q}{2^n}\right) \approx \frac{q}{2} \cdot l \cdot \epsilon^{\triangleright}$.

The resulting lower bound and the upper bound $(q \cdot l \cdot 3\epsilon^{\triangleright})$ differ by about 6 times, this is negligible. Thus, the described extremely synthetic example shows, that the second term in (7) is also tight.

The same is true for "weakened" HMAC-Streebog.

## B HMAC-Streebog

**Theorem (PRF-security of HMAC-Streebog).** *The advantage of the adversary in the PRF model attacking* HMAC-Streebog *is bounded by*

$$\begin{aligned}\mathrm{Adv}^{PRF}_{\mathsf{HMAC\text{-}Streebog}}(t,q,l) \leq & \mathrm{Adv}^{PRF\text{-}RKA_{\boxplus\circ\oplus}}_{\mathsf{g}^{\triangledown}}(t',q',q',d=2)+\\ & +\mathrm{Adv}^{PRF}_{\mathsf{Csc}}(t',q,l') + \mathrm{Adv}^{PRF}_{\mathsf{Csc}}(t',q,l'_\tau) + \frac{2q^2+q}{2^n} + \frac{q^2}{2^{\tau+1}},\end{aligned}$$

*where* $t' = t + O(q \cdot l)$, $\tau \in \{256, 512\}$, $q' = 2 \cdot q + 2$, $l' = l + 1$, $l'_\tau \in \{2, 3\}$.

Proof.

Recall that HMAC-Streebog (for compactness, we denote it here as HMAC) is represented as

$$\mathsf{HMAC}(K, M) = \mathsf{H}\left((\overline{K} \oplus \mathit{opad}) || \mathsf{H}(\overline{K} \oplus \mathit{ipad} || M)\right),$$

where $\mathit{ipad}$, $\mathit{opad} \in V^n$, $\mathit{ipad} \neq \mathit{opad}$, $\overline{K} = (K||0...0) \in V^n$, $K \in V^k$. Streebog-512 or Streebog-256 can be used as H (see also figures 2 and 3).

Let the values in the first (resp. the second) call of the hash function be indicated by the superscript "$I$" (resp. "$O$")

$$\begin{aligned} K^I &= \overline{K} \oplus \mathit{ipad}, & K^O &= \overline{K} \oplus \mathit{opad},\\ K^I_{\mathsf{Csc}} &= \mathsf{g}^{\triangledown}_{K^I}(IV), & K^O_{\mathsf{Csc}} &= \mathsf{g}^{\triangledown}_{K^O}(IV),\\ H^I &= \mathsf{H}(K^I \,||\, M), & H^O &= \mathsf{H}(K^O \,||\, H^I),\\ Y^I &= \mathsf{Csc}(K^I_{\mathsf{Csc}}, M), & Y^O &= \mathsf{Csc}(K^O_{\mathsf{Csc}}, H^I),\\ \sigma^I &= \mathsf{sum}_{\boxplus}(M), & \sigma^O &= \mathsf{sum}_{\boxplus}(H^I). \end{aligned}$$

Just as in the case of Streebog-K, we define "idealized" function

$$\widetilde{\mathsf{HMAC}}(M) = \mathsf{f}_{K^O \boxplus \sigma^O_i}\left(...\mathsf{f}_{K^I \boxplus \sigma^I_i}(\mathsf{Csc}(\mathsf{f}_{K^I \boxplus 0}(IV), M_i))\right),$$

where the first and the last calls of $\mathsf{g}^{\triangledown}$ in both hash functions are replaced by a family of $2^n$ random functions $\mathsf{f}$ indexed by $(\phi, \sigma) \in V^n \times V^n$. Related keys can be represented as $(\overline{K} \oplus \phi) \boxplus \sigma$, and $\phi \in \{\mathit{ipad}, \mathit{opad}\}$.

The "collision" (C) here is treated as a coincidence among $(2q+1)$ values

$$IV, Y_1^I, ..., Y_q^I, Y_1^O, ..., Y_q^O.$$

As before, if there is no "collision" ($\overline{\mathrm{C}}$), then $\widetilde{\mathsf{HMAC}}$ is indistinguishable from a random function $\mathsf{R}(\cdot)$,

$$\Pr(\mathcal{A}^{\mathsf{R}(\cdot)} \Rightarrow 1) = \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})),$$

and we obtain $\mathrm{Adv}_{\mathsf{HMAC}}^{PRF}(\mathcal{A}) =$

$$= \Big(\Pr(\mathcal{A}^{\mathsf{HMAC}(\cdot)} \Rightarrow (1, \mathrm{C})) + \Pr(\mathcal{A}^{\mathsf{HMAC}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}}))\Big) - \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) \leq$$

$$\leq \Big(\Pr(\mathcal{A}^{\mathsf{HMAC}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}})) - \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow (1, \overline{\mathrm{C}}))\Big) +$$

$$+ \Big(\Pr(\mathcal{A}^{\mathsf{HMAC}(\cdot)} \Rightarrow \mathrm{C}) - \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow \mathrm{C})\Big) + \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow \mathrm{C}) =$$

$$= \epsilon + \epsilon_{coll} + p_{coll}.$$

Algorithm $\mathcal{B}_1$ attacks $\mathsf{g}^\triangledown$ in the $PRF\text{-}RKA_{\boxplus\circ\oplus}$ model, its actions are also similar to the previous case. Initially, $\mathcal{B}_1$ queries the cascade keys

$$K_{\mathsf{Csc}}^I = \mathcal{O}(IV, (ipad, 0)) \quad \text{and} \quad K_{\mathsf{Csc}}^O = \mathcal{O}(IV, (opad, 0)),$$

from the oracle $\mathcal{O} \in \{\mathsf{g}^\triangledown, \mathsf{f}\}$.

When processing each query $M_i$ from $\mathcal{A}$, the algorithm $\mathcal{B}_1$ computes

$$Y_i^I = \mathsf{Csc}(K_{\mathsf{Csc}}^I, M_i) \quad \text{and} \quad \sigma_i^I = \mathsf{sum}_\boxplus(M_i).$$

The value of $Y_i^I$ is written in memory. Next, $\mathcal{B}_1$ checks the "collision" condition. If $Y_i^I \in \{IV, Y_1^I, Y_1^O ..., Y_{i-1}^I, Y_{i-1}^O\}$ then $\mathcal{B}_1$ returns 1 and turns off $\mathcal{A}$. Otherwise, $\mathcal{B}_1$ makes query $(Y_i^I, (ipad, \sigma_i^I))$ to the oracle, receives $H_i^I$, computes $Y_i^O = \mathsf{Csc}(K_{\mathsf{Csc}}^O, H_i^I)$ and $\sigma_i^O = \mathsf{sum}_\boxplus(H_i^I)$, saves $Y_i^O$ in memory. The "collision" is checked again, if $Y_i^O \in \{IV, Y_1^I, Y_1^O ..., Y_i^I\}$, then $\mathcal{B}_1$ returns 1 and turns off $\mathcal{A}$. Otherwise, $\mathcal{B}_1$ makes query $(Y_i^O, (opad, \sigma_i^O))$ and transmits the response to $\mathcal{A}$.

If the "collision" conditions have never been met after $q$ queries, then the result of $\mathcal{B}_1$ is the result of $\mathcal{A}$. No more than $(2q+2)$ queries are made to the oracle, only the $IV$ value is requested from the oracle twice under different keys, $d = 2$.

The distinguishing advantage of $\mathcal{B}_1$ is equal to

$$\mathrm{Adv}_{\mathsf{g}^\triangledown}^{PRF\text{-}RKA_{\boxplus\circ\oplus}}(\mathcal{B}_1) = \Pr(\mathcal{B}_1^{\mathsf{g}^\triangledown_{(K\oplus\cdot)\boxplus\cdot}(\cdot)} \Rightarrow 1) - \Pr(\mathcal{B}_1^{\mathsf{f}_{(K\oplus\cdot)\boxplus\cdot}(\cdot)} \Rightarrow 1) = \epsilon + \epsilon_{coll}.$$

We utilize $p_{coll}$ in the algorithm $\mathcal{B}_2$ to distinguish between a pair of cascades $(\mathsf{Csc}(K^I_{\mathsf{Csc}},\cdot),\mathsf{Csc}(K^O_{\mathsf{Csc}},\cdot))$ and a pair of random functions $(\mathsf{R}^I(\cdot),\mathsf{R}^O(\cdot))$. Recall that in $\widetilde{\mathsf{HMAC}}$, keys $K^I_{\mathsf{Csc}}$ and $K^O_{\mathsf{Csc}}$ are random and independent.

Algorithm $\mathcal{B}_2$ on the $i$-th query $M_i$ from $\mathcal{A}$: makes query $M_i$ to the first oracle ($\mathsf{Csc}(K^I_{\mathsf{Csc}},\cdot)$ or $\mathsf{R}^I(\cdot)$); obtains $Y^I_i$; checks "collision". If so, then, $\mathcal{B}_2$ turns off $\mathcal{A}$ and returns 1. Otherwise, $\mathcal{B}_2$ generates random $H^I_i$ (simulation of $\mathsf{f}$), makes query $H^I_i$ to the second oracle ($\mathsf{Csc}(K^O_{\mathsf{Csc}},\cdot)$ or $\mathsf{R}^O(\cdot)$), and obtains $Y^O_i$ (or finds this value in memory if $H^I_i$ was previously requested). If the "collision" occurs, then $\mathcal{B}_2$ turns off $\mathcal{A}$ and returns 1. Otherwise, $\mathcal{B}_2$ passes the randomly generated $H^O_i$ to $\mathcal{A}$.

If there is no "collision" after $q$ queries, then the result of $\mathcal{B}_2$ is 0.

Interaction with cascades makes it possible to perfectly simulate $\widetilde{\mathsf{HMAC}}$ for $\mathcal{A}$, as long as there is no "collision", hence

$$\Pr(\mathcal{B}_2^{\mathsf{Csc}(K^I_{\mathsf{Csc}},\cdot),\mathsf{Csc}(K^O_{\mathsf{Csc}},\cdot)} \Rightarrow 1) = \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow \mathrm{C}).$$

The probability of "collision" in the case when $\mathcal{B}_2$ interacts with $(\mathsf{R}^I(\cdot),\mathsf{R}^O(\cdot))$ is estimated as

$$\Pr(\mathcal{B}_2^{\mathsf{R}^I(\cdot),\mathsf{R}^O(\cdot)} \Rightarrow 1) \leq \frac{q^2}{2^{\tau+1}} + \frac{(2q+1)\cdot(2q)}{2^{n+1}},$$

where the first term takes into account the collision among $H^I_1, ...H^I_q$. Thus,

$$\begin{aligned}\epsilon_{\mathsf{Csc}} &= \Pr(\mathcal{B}_2^{\mathsf{Csc}(K^I_{\mathsf{Csc}},\cdot),\mathsf{Csc}(K^O_{\mathsf{Csc}},\cdot)} \Rightarrow 1) - \Pr(\mathcal{B}_2^{\mathsf{R}^I(\cdot),\mathsf{R}^O(\cdot)} \Rightarrow 1) \geq \\ &\geq \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow \mathrm{C}) - \left(\frac{q^2}{2^{\tau+1}} + \frac{2q^2+q}{2^n}\right), \\ p_{coll} &= \Pr(\mathcal{A}^{\widetilde{\mathsf{HMAC}}(\cdot)} \Rightarrow \mathrm{C}) \leq \epsilon_{\mathsf{Csc}} + \left(\frac{q^2}{2^{\tau+1}} + \frac{2q^2+q}{2^n}\right).\end{aligned}$$

In turn, the advantage $\epsilon_{\mathsf{Csc}}$ may simply be bounded by the "hybrid argument"

$$\epsilon_{\mathsf{Csc}} \leq \mathrm{Adv}^{PRF}_{\mathsf{Csc}}(\mathcal{B}^I_2) + \mathrm{Adv}^{PRF}_{\mathsf{Csc}}(\mathcal{B}^O_2).$$

The algorithms $\mathcal{B}^I_2$ and $\mathcal{B}^O_2$ make $q$ queries each. Queries from the first algorithm are no longer than $l' = (l+1)$ block. If $\tau = 256$ (resp. $\tau = 512$) then $\mathcal{B}^O_2$ makes 2-block (resp. 3-block) queries. □

By using the heuristic estimates (1), (6), we obtain $\mathrm{Adv}^{PRF}_{\mathsf{HMAC\text{-}Streebog}}(t,q,l) \lessapprox$

$$\begin{aligned}&\lessapprox \frac{2\cdot t'}{2^k} + \left(\frac{2\cdot t'\cdot q\cdot l'}{2^n} + \frac{q^2}{2^{n+1}}\right) + \left(\frac{2\cdot t'\cdot q\cdot l'_\tau}{2^n} + \frac{q^2}{2^{n+1}}\right) + \frac{2q^2+q}{2^n} + \frac{q^2}{2^{\tau+1}} \leq \\ &\leq \frac{t'}{2^{k-1}} + \frac{t'\cdot q\cdot l''}{2^{n-1}} + \frac{3q^2+q}{2^n} + \frac{q^2}{2^{\tau+1}}, \qquad (8)\end{aligned}$$

where $l' = l + 1$, $l'_\tau \in \{2, 3\}$, $l'' = l + 4$.

Hence, for the case $\tau = n$, estimate (8) is close to the same (7) for Streebog-K.

However, for the 256-bit version ($\tau = \frac{n}{2}$), the most significant may be the last $\tau$-dependent term. In this case, we can speak about "$k$-bit security" only if $ql < 2^{n-k-1}$ and $q < 2^{\frac{n}{2}-k}$. We don't know the matching forgery attack for this case, but the distinguishing is trivial. For $q = 2^{\frac{n}{2}-1}$, the probability of a collision among the outputs of a random function is about half as low as the corresponding probability for HMAC-Streebog-256.

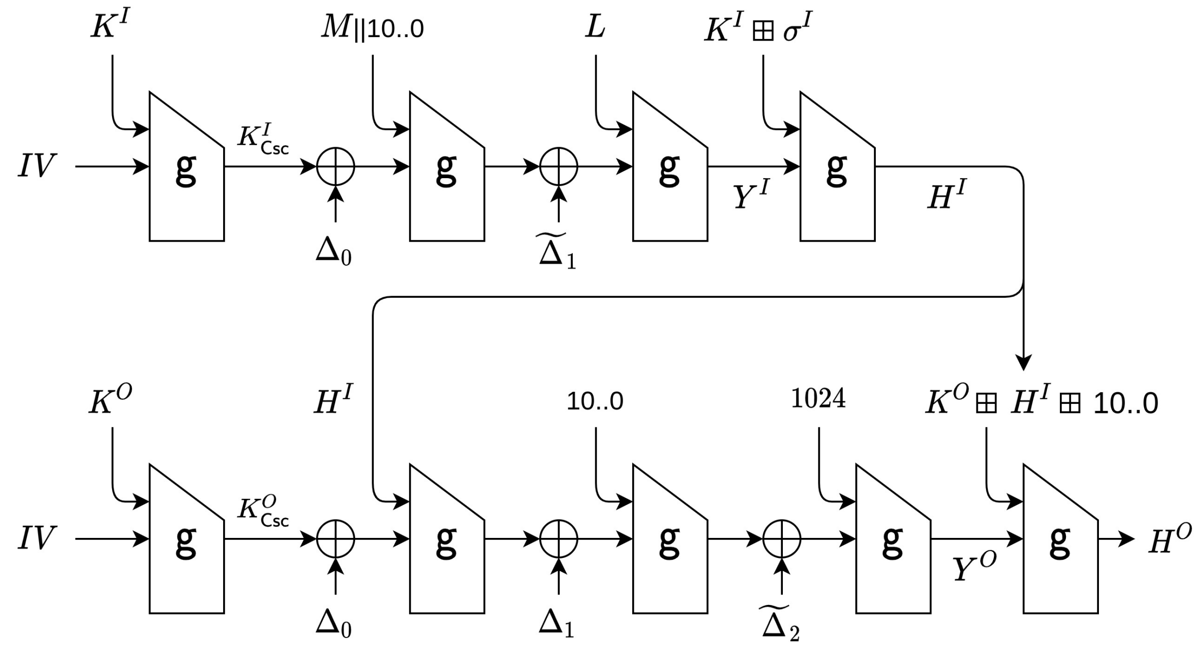

Figure 2: HMAC-Streebog-512 with equivalent representation. The message $M$ consists of $L < 512$ bits.

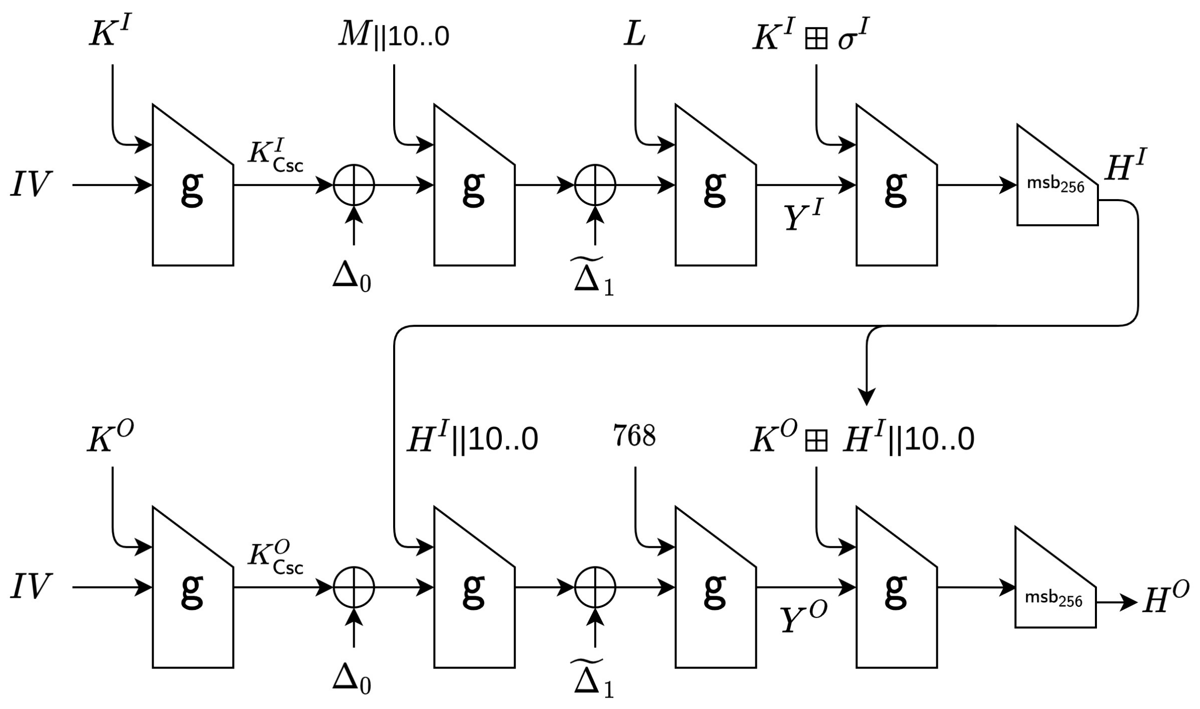

Figure 3: HMAC-Streebog-256 with equivalent representation. The message $M$ consists of $L < 512$ bits.