Tight Security Bounds for Generic Stream Cipher Constructions

Matthias Hamann and Matthias Krause

University of Mannheim, Germany, {hamann,krause}@uni-mannheim.de

Abstract.
The design of modern stream ciphers is strongly influenced by the fact that Time-Memory-Data tradeoff attacks (TMD-TO attacks) reduce their effective key length to $SL/2$, where SL denotes the inner state length. The classical solution, employed, e.g., by eSTREAM portfolio members Trivium [CP05] and Grain v1 [HJM06], is to design the cipher in accordance with the Large-State-Small-Key construction, which implies that SL is at least twice as large as the session key length KL.
In the last years, a new line of research looking for alternative stream cipher constructions guaranteeing a higher TMD-TO resistance with smaller inner state lengths has emerged. So far, this has led to three generic constructions: the Lizard construction [HK18], having a provable TMD-TO resistance of $2 \cdot SL/3$; the Continuous-Key-Use construction, underlying the stream cipher proposals Sprout [AM15], Plantlet [MAM17], and Fruit [AH18]; and the Continuous-IV-Use construction, very recently proposed in [HKM17a]. Meanwhile, it could be shown that the Continuous-Key-Use construction is vulnerable against certain nontrivial distinguishing attacks [HKMZ17].
In this paper, we present a formal framework for proving security lower bounds on the resistance of generic stream cipher constructions against TMD-TO attacks and analyze two of the constructions mentioned above. First, we derive a tight security lower bound of approximately $\min\{KL, SL/2\}$ on the resistance of the Large-State-Small-Key construction. This shows that the feature $KL \leq SL/2$ does not open the door for new nontrivial TMD-TO attacks against Trivium and Grain v1 which are more dangerous than the known ones. Second, we prove a maximal security bound on the TMD-TO resistance of the Continuous-IV-Use construction, which shows that designing concrete instantiations of ultra-lightweight Continuous-IV-Use stream ciphers is a hopeful direction of future research.

Keywords: Stream Ciphers · Generic Time-Memory-Data Tradeoff Attacks · Security Lower Bound Proofs · Random Oracle Models

1 Introduction

Security proofs in so-called random oracle models (for short ROMs, sometimes also called ideal primitive models (IPMs), as in [GT15]) play an important role for the security analysis of symmetric cryptographic constructions (see, e.g., [GT15] for a systematic overview). A ROM is based on identifying the main components of the cryptographic construction and assuming them to behave as randomly chosen. The security proofs have an information-theoretic nature and refer to computationally unbounded attackers who have black-box oracle access to the components and to the construction. The security of the construction is measured by the minimal number of oracle queries needed for reaching certain attack goals like recovering the secret symmetric key or distinguishing the construction from an appropriate random counterpart.

In practice, this type of security expresses the resistance of concrete instantiations of the construction against generic attacks, which do not take the inner structure of the relevant components into account, but instead target the way these components interact.

While, in the last decade, a large number of ROM-based formal security bounds have been shown for constructions like, e.g., block ciphers, operation modes of block ciphers, block cipher-based hash functions, or message authentication codes,[1] comparatively little is known so far about ROM-based approaches to provable stream cipher security.

In this paper, we present a ROM-approach for analyzing the security of keystream generator-based (KSG-based) stream ciphers against generic Time-Memory-Data tradeoff (TMD-TO) attacks, a classical and powerful type of attack, which goes back to Babbage [Bab95] and Golić [Gol96]. We derive security lower bounds for several generic constructions, which underlie various stream cipher proposals made in the last years, including the eSTREAM portfolio members Trivium [CP05] and Grain v1 [HJM06].

TMD-TO attacks reduce the security level of classical types of stream ciphers to $SL/2$, where SL denotes the inner state length of the underlying keystream generator. One established way to handle this fact, employed, e.g., by Trivium and Grain v1, is to design the cipher according to the Large-State-Small-Key construction, which implies to choose SL at least twice as large as the session key length KL. Here, the intended security level corresponds to the session key length KL and it usually holds that $KL \geq 80$. However, the resulting comparatively large inner state length is a relevant issue in many practical situations, as stream ciphers are often designed for application scenarios with a desire for ultra-lightweight devices (see [HKMZ17] for a more detailed discussion).

This motivated a new line of research looking for stream ciphers constructions which provide a resistance higher than $SL/2$ against TMD-TO attacks and would hence allow to build secure stream ciphers with inner state lengths smaller than 160. So far, this research has led to three generic constructions: the Lizard construction [HK18], having a provable TMD-TO resistance of $2 \cdot SL/3$ and underlying the stream cipher proposal Lizard ($SL = 121$) [HKM17b]; the Continuous-Key-Use construction, underlying the stream cipher proposals Sprout [AM15], Plantlet [MAM17], and Fruit [AH18]; and the Continuous-IV-Use construction, very recently proposed in [HKM17a].

The last two constructions are based on dividing the set of the SL inner state cells into a set of VSL volatile ones (e.g., as flip-flops), and a set of $SL - VSL$ non-volatile ones (e.g., in EEPROM), which do not change their content during the keystream generation. Continuous-Key-Use stream ciphers employ the secret session key in the non-volatile part of the inner state during keystream generation, Continuous-IV-Use ciphers do the same with the initial value (IV). The aim here is to construct ciphers with a (preferably *provable*) TMD-TO resistance of 80 bits (i.e., 80-bit security) with VSL being significantly smaller than 160 bits. The justification of this approach is given by the fact that the hardware realization of a stream cipher usually has to provide separate memory cells for storing IVs and the secret session key anyway, so why not use them for the state transition. Note that while the situation of secret keys being kept persistently in, e.g., EEPROM may be apparent, also IVs require some separate memory location in most real-world applications, i.e., they cannot be simply overwritten by the stream cipher's state transition function during keystream generation. For example, in A5/1 of GSM [BGW99] the IV employed in the encryption of a data packet is the respective (sequentially incremented) 22-bit frame number. We refer to [HKMZ17] for further examples and a more detailed discussion on this.

Very recently, it came to light that the Continuous-Key-Use construction does not fulfill the expectations. By giving a corresponding attack, it was shown in [HKMZ17] that the resistance of the Continuous-Key-Use construction against generic TMD-TO

[1] See, e.g., the large body of recent work on the analysis of iterated Even-Mansour ciphers, or on IPM-analyzing block cipher-based constructions of cryptographic hash functions.

distinguishing attacks is only $VSL/2$. This once more emphasizes the importance of equipping serious proposals for new stream ciphers with lower bound proofs on their security against TMD-TO attacks, and we hope that our paper represents a valuable contribution in this context.

After introducing our random oracle model for KSG-based stream cipher constructions, we first derive a corresponding tight security lower bound of approximately $\min\{KL, SL/2\}$ on the TMD-TO resistance of the Large-State-Small-Key construction underlying, e.g., Trivium and Grain v1. This shows that the design feature $KL \leq SL/2$ of such ciphers does not open the door for TMD-TO attacks which are more efficient than the classical ones of Babbage [Bab95] and Golić [Gol96], and of Biryukov and Shamir [BS00].

Subsequently, as our main result, for keystreams of at most PL bits generated under the same key-IV pair, we prove a tight maximal security bound of approximately $\min\{KL, VSL - \log_2(PL)\}$ on the resistance of the Continuous-IV-Use construction against TMD-TO attacks. This emphasizes that designing concrete instantiations of ultra-lightweight Continuous-IV-Use stream ciphers is a hopeful direction of future research.

Note that the first formal security lower bound on the resistance of generic stream cipher constructions against TMD-TO attacks, a tight $2 \cdot SL/3$ lower bound for the Lizard construction, was presented in [HK18]. This bound refers to TMD-TO attacks with the goals *key recovery* and *packet prediction*. In this paper, for the first time, security lower bounds on the resistance of stream ciphers against TMD-TO *distinguishing* attacks are derived.

Before describing our results in more detail in Subsection 1.3, in the next two subsections of this introduction we provide further basics around stream ciphers and TMD-TO attacks.

1.1 Stream Cipher Basics

Stream ciphers are symmetric encryption algorithms intended for the online encryption of plaintext bitstreams X which have to pass an insecure channel. The encryption is performed via bitwise addition of a keystream $S = S(k, IV)$, which is generated in dependence of a secret symmetric session key k and, possibly, a public initial value IV. The legal recipient, who also knows k, decrypts the encrypted bitstream $Y = X \oplus S$ by generating S and computing $X = Y \oplus S$. In this paper, we consider KSG-based stream ciphers, i.e., stream ciphers which generate the keystream by a so-called keystream generator (KSG).

KSGs are stepwise working devices which can be formally specified by finite automata, defined by an inner state length SL and the corresponding set of inner states $\{0,1\}^{SL}$, a state update function $\pi : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$, and an output function $out : \{0,1\}^{SL} \longrightarrow \{0,1\}$. Starting from an initial state q_0, in each clock cycle $i \geq 0$, the KSG produces a keystream bit $z_i = out(q_i)$ and changes the inner state according to $q_{i+1} = \pi(q_i)$. The output bitstream $S(q_0)$ is defined by concatenating all the outputs $z_1 z_2 z_3 \cdots$.

In the context of TMD-TO security, it is convenient to express the output behavior of a stream cipher by the function $OUTBLOCK : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$, which is defined by π and out and which assigns to each inner state $q \in \{0,1\}^{SL}$ the block

$$OUTBLOCK(q) = (OUTBLOCK(q)_0, \cdots, OUTBLOCK(q)_{SL-1})$$

of the first SL keystream bits generated on q, where for all j, $0 \leq j \leq SL - 1$,

$$OUTBLOCK(q)_j = out(\pi^j(q)).$$

The keystream generation process of a KSG-based stream cipher usually depends on a further parameter $PL \geq SL$, the packet length, and can be divided into the following four phases:

(1) **The session key generation phase:** Here, the secret session key $k \in \{0,1\}^{KL}$ is generated by running a key-exchange protocol between the legal communication partners. This phase will not be considered in this paper.

(2) **The loading phase:** In this phase, the session key k together with an initial value $IV \in \{0,1\}^{IVL}$, and, possibly, some constants are loaded into the inner state register cells of the KSG. This phase results in a state $q_{\text{load}} = q_{\text{load}}(IV, k) \in \{0,1\}^{SL}$.

(3) **The mixing phase:** Here, the KSG transforms the loading state q_{load} into the initial state

$$q_{\text{init}}(IV, k) = MIX(q_{\text{load}}(IV, k))$$

by the help of a so-called mixing algorithm $MIX : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$. This is done without outputting keystream bits.

The goal of MIX is to provide a sufficiently large amount of diffusion, confusion, and high algebraic degree in the dependencies of the initial state bits from the session key and initial value bits.

(4) **The output phase**, in which the keystream packet corresponding to k and IV, consisting of the first PL bits of the keystream $S(q_{\text{init}}(IV, k))$, is generated in the way described above.

One distinguishes stream ciphers which work in one stream mode (like Trivium [CP05] or Grain v1 [HJM06]) and in packet mode (like the GSM standard $A5/1$ [BGW99] or Lizard [HKM17b]).

In the one-stream mode, the packet length is defined to be larger than the session length, i.e., the number of keystream bits needed to encrypt one session. So, the keystream for encrypting the communication of one session is the prefix of only one keystream packet and only one initial value per session is needed.

In the packet mode, the packet length is defined to be much shorter than the session length and the keystream is a concatenation of packets, where for each packet the initialization algorithm (phases (2) and (3)) has to be restarted with a new initial value. The motivation underlying this approach is that in many real-world communication scenarios (Ethernet, WLAN, Bluetooth, cellular networks etc.), data streams are transmitted packet-wise. It thus seems natural to consider stream ciphers running packet mode and, in particular, to look for corresponding design optimizations.

For illustrating the Large-State-Small-Key construction, we give a rough description of the stream ciphers Trivium and Grain v1, which belong to the final portfolio of the eSTREAM contest [BBV12].

Trivium: The stream cipher Trivium has an inner state length of $L = 288$ bits, distributed over three nonlinear feedback shift registers (NFSRs) of lengths $93, 84, 111$ bits. The state update function consists of the corresponding three feedback functions, which, in each case, are quadratic and take their inputs from two of the three NFSRs. The linear output function produces one keystream bit per clock cycle. It XORs six inner state bits, two from each NFSR. The loading state $q_{\text{load}}(IV, CONST, k)$ is defined to be the concatenation of the 80-bit session key k, the 80-bit initial value IV, and a predefined 128-bit constant $CONST$. The MIX operation consists in clocking the KSG $4 \cdot 288$ times without producing output (see [CP05] for more details).

Grain v1: The stream cipher Grain v1 has an inner state length of $L = 160$ bits, distributed over one NFSR and one linear feedback shift register (LFSR), both of length 80 bits. The state update function consists of the corresponding two feedback functions, where the NFSR feedback function depends also on one of the LFSR bits. Again, the output function produces one keystream bit per clock cycle and depends nonlinearly

on five LFSR bits and one NFSR bit and linearly on further seven NFSR bits. The loading state $q_{\text{load}}(IV, CONST, k)$ is defined to be the concatenation of the 80-bit session key k, the 64-bit initial value IV, and a predefined 16-bit constant $CONST$. In the mixing phase, the Grain-KSG is clocked 160 times, where, in each clock cycle, the corresponding output keystream bit is XORed to the result of each of the two feedback functions (see [HJM06] for more details).

We obtain that in both cases, the SL-block of bits r to $r + SL - 1$ of the keystream packet corresponding to k and IV can be expressed as

$$OUTBLOCK(\pi^r(MIX(IV, CONST, k))). \tag{1}$$

For the Continuous-IV-Use construction, which is also treated in this paper, a concrete instantiation has yet do be designed. However, as part of introducing the general idea of continuously using the IV during keystream generation, the authors of [HKM17a] conjecture that "cyclically XORing one IV bit per step to the volatile inner state" would already be sufficient. This actually corresponds to the way how the secret key is employed in the Continuous-Key-Use stream cipher Plantlet, where, in each step, cyclically one of the 80 key bits is XORed to the register feedback of one of its two feedback shift registers.

1.2 Time-Memory-Data Tradeoff Attacks and Small State Ciphers

During the last decades, many different techniques for cryptanalyzing KSG-based stream ciphers have been developed (correlation attacks, fast correlation attacks, guess-and-verify attacks, BDD attacks, cube attacks etc.). Attacks on stream ciphers typically refer to a known-IV scenario, in which the attacker knows a set $\mathcal{S}$ of keystream blocks having their origin in one session with secret session key k, and which were generated with respect to a set of known initial values. Typical goals of attacks on stream ciphers are to distinguish $\mathcal{S}$ from a set of blocks coming from a truly random source, to recover the inner state responsible for at least one keystream block contained in $\mathcal{S}$, or to predict a keystream packet corresponding to k and a new initial value IV.

In this paper, we concentrate on the proof of security lower bounds for TMD-TO attacks. TMD-TO attacks are generic in the sense that they have only black-box access to the component functions MIX and $OUTBLOCK$. TMD-TO attacks are often divided into a precomputation phase, in which some helping data structure is computed, and an online phase, in which on the basis of the keystream available for the attack and the helping data structure the goal of the attack is reached. The relevant costs of a TMD-TO attack are typically measured in the four cost dimensions D (the amount keystream (data) available in the online phase), T (the time consumption of the online phase), P (the time consumption of the precomputation phase), and M (the memory consumption including the size of the helping data structure). The costs are expressed in a so-called tradeoff curve, which is built by all 4-tuples (T, M, D, P) of cost values, which allow to reach the goal of the attack with high probability. For attacks without precomputation phase, the cost dimension P is not considered.

The first TMD-TO attacks against KSG-based stream ciphers go back to Babbage [Bab95] and Golić [Gol96] and yield the tradeoff curve $T \cdot D = 2^{SL}$, which contains the point $T = D = 2^{SL/2}$. We describe the idea of these attacks below. Biryukov and Shamir [BS00] combined the idea of the attacks of Babbage and Golić with the idea of Hellman's attack on block ciphers [Hel80], yielding an attack with tradeoff curve $T \cdot M^2 \cdot D^2 = 2^{2 \cdot SL}$ with $P = 2^{SL}/D$. In [HK18], a TMD-TO key recovery attack without precomputation phase against the Lizard construction is presented, which is successful with high probability for $T = D = M = 2^{2/3 \cdot SL}$ and matches the security lower bound shown in the same paper.

In our security proofs for the resistance of generic stream cipher constructions, we derive lower bounds on the overall time consumption $T + P$ holding for all TMD-TO

attacks against the respective construction. Note that always $M, D \leq T + P$ holds, as occupied memory blocks are the result of operations covered by P or T and, similarly, data blocks not treated by corresponding operations would be of no use. Furthermore, we refer to chosen-IV attackers who have access to blocks of keystream packets generated with respect to initial values of the attacker's choice.

For illustration, we describe the classical TMD-TO attack of Babbage [Bab95] and Golić [Gol96]: Suppose that the attacker knows a set $\mathcal{S}$ of D keystream blocks of length SL, having their origin in one session with secret session key k, and let $Q = \{q^1, \cdots, q^D\}$ denote the set of corresponding inner states. The attacker generates a set of T pairs $(y, OUTBLOCK(y))$ for randomly chosen inner states $y \in \{0,1\}^{SL}$. If $D \cdot T \approx 2^{SL}$, then, with high probability, there will occur a collision, i.e., some y falls into Q, which implies that $OUTBLOCK(y)$ falls into $\mathcal{S}$. As a result, the attacker knows the inner state q^j responsible for one keystream block of a packet generated with respect to a known initial value IV. This allows to compute the whole keystream packet corresponding to k and IV, and to recover the initial state $q_{\text{init}}(IV, k)$ for this packet. Moreover, for Trivium, Grain v1, and many other ciphers, it is even possible to efficiently compute k from $q_{\text{init}}(IV, k)$.

By setting $D = T = 2^{SL/2}$, we obtain an attack which lowers the security level of the respective cipher to $SL/2$, as it consumes data, time, and memory of at most $2^{SL/2}$. Consequently, for reaching the intended security level KL, the length of the secret session key, classical stream ciphers have to use an inner state length SL of at least $2 \cdot KL$.

In Section 3, we analyze a generic Large-State-Small-Key construction with $SL = VSL = IVL + KL$, $q_{\text{load}}(IV, k) = (IV|k)$, and $q_{\text{init}}(IV, k) = MIX(IV|k)$. For this construction, exhaustive key search and the Babbage-Golić attack yield a security upper bound of $\min\{KL, SL/2\}$. The first main result of this paper is to show a nearly tight (up to a factor of $2 \cdot SL$ w.r.t. attack complexity $2^{SL/2}$) lower bound for this upper bound w.r.t. TMD-TO attacks (see Relation (2)).

As already mentioned, the search for constructions yielding a TMD-TO resistance beyond $SL/2$ has led so far to three generic constructions: the Lizard-construction, the Continuous-Key-Use construction, and the Continuous-IV-Use construction.

LIZARD construction implies to run a stream cipher with $SL = KL = IVL$ in packet mode with packet length PL, where the packet initial states are computed according to $q_{\text{init}}(IV, k) = MIX(k \oplus IV) \oplus k$. This does not prevent recovering the initial state of *one* of at least $2^{SL/2}/PL$ known keystream packets (i.e., $2^{SL/2}$ bits of known data in total) by applying the Babbage-Golić attack with TMD-cost $2^{SL/2}$. However, one can prove beyond-the-birthday-bound resistance against key recovery and packet prediction attacks. More precisely, by considering a random oracle model approach similar to the one employed here, a corresponding $2/3 \cdot SL$ lower bound was shown in [HK18]. This justified the proposal of the stream cipher Lizard [HKM17b], which uses an inner state length of $SL = 121$.

CONTINUOUS-KEY-USE construction means that the secret session key is continuously employed during keystream generation and thus becomes a non-volatile part of the cipher's inner state. This principle underlies the stream cipher proposals Sprout [AM15] ($SL = 167$ and $VSL = 87$), Plantlet [MAM17] ($SL = 188$ and $VSL = 108$), and Fruit [AH18] ($SL = 167$ and $VSL = 87$). Remember here that for Continuous-Key-Use (and Continuous-IV-Use) constructions, we have to distinguish between SL, the size of the full inner state, and VSL, the size of its volatile part. In particular, depending on the concrete instantiation, also (possibly secret) counters used, e.g., for key/IV bit selection, can become part of the volatile inner state if they influence the state update during keystream generation (see, e.g., [HKMZ17] for further details).

Very recently, it could be shown in [HKMZ17] that the resistance of Continuous-Key-Use ciphers against generic TMD-TO distinguishing attacks does not exceed

$VSL/2$. In the following, we give a rough description of the corresponding approach. Note that at the beginning of the corresponding oracle game, in the pseudorandom case, the oracle randomly and independently chooses a secret session key on the basis of which it henceforth provides its replies.

First, the attacker obtains $2^{VSL/2}$ consecutive keystream blocks (each of length $\tilde{n}$ slightly larger than VSL) from about $2^{VSL/2}$ bits of keystream provided by the oracle (possibly generated under a single IV) and stores these blocks in an efficiently searchable data structure.[2] Then, for $2^{VSL/2}$ randomly and independently chosen IVs, he obtains the corresponding $\tilde{n}$-bit keystream prefix from the oracle. Due to the birthday paradox, in the pseudorandom scenario, he is likely to find a collision of one of the keystream prefixes generated in the second step with one of the keystream blocks stored in the first step. This holds as the session key is fixed and thus all inner states differ only in the volatile part. Hence, with high probability, some *initial state* underlying one of the keystream prefixes of the second step will be identical to some *inner state* underlying one of the keystream blocks from the first step. As $\tilde{n} > VSL$, this allows to distinguish the cipher from a truly random source in a generic way. For a detailed description of this attack, we refer the reader to [HKMZ17].

CONTINUOUS-IV-USE construction (as suggested in [HKM17a]) refers to a cipher working in packet mode with packet length PL where the IV is continuously employed during keystream generation and thus becomes a non-volatile, publicly known part of the cipher's inner state. It can be easily checked that the above attack against Continuous-Key-Use ciphers can not be applied to Continuous-IV-Use.

In Section 3, we will analyze a variant of the Continuous-IV-Use construction in which a part of length $VIVL$ of the IV (having total length IVL) is actually not continuously employed during keystream generation but only enters the volatile part of the loading state q_{load} in the classical way known, e.g., from Trivium and Grain v1. The rest of the IV, which we will call its non-volatile part, forms the constant part of the inner state (during loading, mixing, and keystream generation). Correspondingly, we set $SL = IVL + KL$ and $VSL = VIVL + KL$, and require $VIVL \leq \log_2(PL)$. For all keys $k \in \{0,1\}^{KL}$ and initial values $IV \in \{0,1\}^{IVL}$, it holds $q_{\text{load}}(IV, k) = (IV|k)$ and $q_{\text{init}}(IV, k) = MIX(IV|k)$. As pointed out above, we assume that MIX leaves the non-volatile part of the state constant. The reason for considering this particular variant of the Continuous-IV-Use construction will be explained now.

Note that there are two ways of applying the Babbage-Golić TMD-TO attack to this cipher. The first approach is to mount the attack in its original form, which does not take the special structure of inner states into account. This attack has the tradeoff curve $T \cdot D = 2^{SL}$, yielding the point $T = D = 2^{SL/2} = 2^{(IVL+KL)/2}$. Note that the respective amount $D = 2^{SL/2}$ of data is only available if IVL is at least $SL/2 - \log_2(PL)$.

The second approach is to make use of the fact that the IVs for the keystream packets are publicly known. Let us hence assume that the keystream data consists of U keystream packets of length PL corresponding to the initial values $IV^{(1)}, \cdots, IV^{(U)}$. Note that U is variable, PL is constant, and the resulting data complexity is $D = U \cdot PL$. The attacker now generates at most S times a random state $z \in \{0,1\}^{VSL}$ and computes $OUTBLOCK(IV^{(u)}, z)$ for all u, $1 \leq u \leq U$, until a collision with the data occurs. This attack has the time complexity $T = U \cdot S$ and, based on the birthday paradox, needs to satisfy $U \cdot PL \cdot S \geq 2^{VSL}$, i.e., $T \geq 2^{VSL}/PL$. The best choice for an attacker is thus $S = 2^{VSL}/PL$ and $U = 1$ (i.e., attacking only a single packet of length PL) as it leads to the optimal values $T = 2^{VSL}/PL$ and

[2] That is, he slides an $\tilde{n}$-bit window over the given keystream.

$D = PL$ for time and data complexity, respectively. This now also immediately shows that considering fewer than PL keystream bits per packet in the attack would lead to worse results. Together with the trivial exhaustive key search attack, we hence obtain a security upper bound of $\min\{KL, VSL - \log_2(PL)\}$ on the resistance of this construction.

The second main result of this paper is to show a nearly matching (up to a factor of $2 \cdot SL$ w.r.t. attack complexity $2^{VSL-\log_2(PL)}$) security lower bound for this upper bound (see Relation (3)). The reason why, as described above, we additionally require $VIVL \leq \log_2(PL)$ in the analyzed variant of the Continuous-IV-Use construction will become clear in the course of the proof. However, note already that, in consequence, choosing $VIVL = \log_2(PL)$ is optimal, as $VSL = VIVL + KL$ then implies $VSL - \log_2(PL) = KL$.

1.3 Our Results

In this paper, we introduce a random oracle model (ROM) for KSG-based stream ciphers and prove tight security bounds on the resistance against TMD-TO attacks for two of the four generic stream cipher constructions discussed in the last subsection: the Large-State-Small-Key construction and the Continuous-IV-Use construction.

Our ROM refers to a packet length parameter PL and a predefined bijective state transition function π, which is assumed to have everywhere a large period. The ROM is based on identifying the functions MIX and out (respectively, $OUTBLOCK$, which is determined by out and π) as the main components of the cipher.

We derive our security bounds by analyzing the maximal success probability of an attacker Eve in a distinguishing game with players Eve and Alice, where Alice holds the secret session key k and randomly chosen instantiations of the components MIX and out. Attacker Eve is allowed to pose oracle queries to the components MIX and $OUTBLOCK$, and construction oracle queries with inputs (IV, r), $0 \leq r \leq PL - 1 - SL$. The answer to a construction query with input (IV, r) is the block of bits $r, \cdots, r + SL - 1$ of the keystream packet corresponding to initial value IV.

The goal of the attacker Eve is to distinguish the pseudorandom scenario, in which the answers to the construction queries refer to keystream packets generated on k and IV in accordance to the stream cipher construction under consideration, from a random scenario, in which randomly and independently for each initial value IV a keystream packet of length PL is generated.

As usual, our security proofs have an information-theoretic nature, i.e., we consider Eve to be a randomized algorithm of unbounded computational power. Eve is allowed to pose a predefined number M of oracle queries to Alice and has to output $b = 0$ (pseudorandom case) or $b = 1$ (random case) after posing these M queries. The success of Eve is expressed by the advantage $Adv(M)$, which is defined as

$$Adv(M) = \left|\Pr[b = 0|\text{pseudorandom scenario}] - \Pr[b = 0|\text{random scenario}]\right|,$$

where the probabilities are taken w.r.t. Alice's random choice of the components and the internal randomization of Eve.

Our main results are that in the game corresponding to the Large-State-Small-Key construction it holds

$$Adv(M) \leq \frac{M}{2^{KL} - M} + \frac{(2 \cdot SL + 1) \cdot M^2}{2^{SL} - (2 \cdot SL + 1) \cdot M^2} \tag{2}$$

and that in the game corresponding to the Continuous-IV-Use construction it holds

that $Adv(M)$ is bounded by

$$Adv(M) \leq \frac{M}{2^{KL} - M} + \frac{PL^2 \cdot (2 \cdot SL + 1) \cdot M}{2^{VSL} - M \cdot (2 \cdot SL + 1) \cdot PL}. \tag{3}$$

The first result says that no generic TMD-TO attack against Large-State-Small-Key stream ciphers can be significantly better than the Babbage-Golić attack (if $SL < 2 \cdot KL$) or exhaustive key search (if $SL \geq 2 \cdot KL$).

The second result says that under the conditions that the packet length PL is moderately bounded, no generic TMD-TO attack against Continuous-IV-Use stream ciphers is significantly better than exhaustive key search (if $VIVL = \log_2(PL)$, implying $VSL = KL + \log_2(PL)$) or the TMD-TO attacks described in the previous subsection (if $VIVL \leq \log_2(PL)$, implying $VSL \leq KL + \log_2(PL)$).

Structure of the paper: The remaining part of this paper is organized as follows. In *Section 2*, we formally define our generic constructions and the corresponding distinguishing games. *Section 3* contains the formulation of our main result Theorem 1, consisting in the relations (2) and (3), and the corresponding proof. Our proof does not explicitly use the H-coefficient technique of Patarin [Pat09], but it follows the typical structure of such proofs as it was described, e.g., in [CS14]. In particular, we operate with an appropriate definition of a *bad computation transcript*, show that the probability of a bad transcript is sufficiently small, and that Eve has no chance to distinguish the random case from the pseudorandom case during a computation associated with a good transcript. In *Section 4*, we discuss some resulting aspects for the design of practical instantiations of Continuous-IV-Use stream ciphers.

2 A Random Oracle Model for Stream Ciphers

In this section, we introduce the random oracle models for the Large-State-Small-Key construction (underlying, e.g., Grain v1 [HJM06] and Trivium [CP05]) and the Continuous-IV-Use construction introduced in [HKM17a], and start with the formal definitions for them.

Definition 1. A stream cipher designed according to the Large-State-Small-Key construction, resp. the Continuous-IV-Use construction, is defined in the following way:

- Both constructions depend on the parameters KL (the session key length), IVL (the initial value length), PL (the packet length), and SL (the inner state length).

  Inner states consist of a volatile part of length VSL and a non-volatile part of length $SL - VSL$. Initial values also consist of a volatile part (more exactly, a part which is not continuously employed during keystream generation but only enters the volatile part of the loading state q_{load}) of length $VIVL$ and a non-volatile part (forming the constant part of the inner state) of length $IVL - VIVL$.

  For all inner states $y \in \{0,1\}^{SL}$, we denote by $v(y) \in \{0,1\}^{VSL}$ and $nv(y) \in \{0,1\}^{SL-VSL}$ the volatile, resp. the non-volatile part of y. Clearly, $y = nv(y)||v(y)$.

  In the same way, for all initial values $x \in \{0,1\}^{IVL}$, we denote by $v(x) \in \{0,1\}^{VIVL}$ and $nv(x) \in \{0,1\}^{IVL-VIVL}$ the volatile, resp. the non-volatile part of x.

- In both constructions, it holds $SL = IVL + KL$. For the Large-State-Small-Key construction, it holds $SL = VSL$ (i.e., the whole inner state is volatile) and, correspondingly, $IVL = VIVL$. For the Continuous-IV-Use construction, it holds

$VSL = VIVL + KL$ and, correspondingly, $SL = VSL + IVL - VIVL$, i.e., the non-volatile part of the inner state equals the non-volatile part of the initial value.

For the CONTINUOUS-IV-USE construction we bound the length of the volatile part of the initial value, in particular we assume that $VIVL \leq \log_2(PL)$.

- **State transition:** Both constructions refer to a bijective state transition function $\pi : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ for which the period of the sequence $(\pi^i(z))_{i\geq 0}$ is greater PL for all inner states $z \in \{0,1\}^{SL}$. For the CONTINUOUS-IV-USE construction, the additional restriction holds that π leaves the non-volatile part of the state constant, i.e., for each $z \in \{0,1\}^{VSL}$ and $x \in \{0,1\}^{SL-VSL}$, there is some $z' \in \{0,1\}^{VSL}$ such that
$$\pi(x,z) = (x,z').$$

- **Loading:** In both constructions, the concatenation of the initial value IV and the session key k forms the loading state $q_{\text{load}} = (IV|k)$. In the case of the CONTINUOUS-IV-USE construction, the non-volatile part of the initial value, which has length $IV - VIVL$, represents the non-volatile part of this state.

- **Mixing and state initialization:** Both constructions use a bijective state mixing function $MIX : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$, which is allowed to be efficiently invertible. This implies that $q_{\text{init}}(IV,k) = MIX(IV|k)$. For the CONTINUOUS-IV-USE construction (where $VSL = VIVL + KL$), we assume that MIX leaves the non-volatile part of the state constant, i.e., for each $z \in \{0,1\}^{VSL}$ and $x \in \{0,1\}^{SL-VSL}$, there is some $z' \in \{0,1\}^{VSL}$ such that
$$MIX(x,z) = (x,z').$$

- **Keystream generation:** Both constructions employ an output bit function $out : \{0,1\}^{SL} \longrightarrow \{0,1\}$, which defines, together with π, the corresponding output block function $OUTBLOCK : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$, where for each inner state $y \in \{0,1\}^{SL}$, $OUTBLOCK(y) = (z_0, \cdots, z_{SL-1})$ and $z_i = out(\pi^i(y))$, $i = 0, \ldots, SL-1$.

  The keystream packet $(z_0, \cdots, z_{PL-1})$ corresponding to a key-IV pair (k, IV) is defined by
$$z_i = out(\pi^i(q_{\text{init}}(IV,k))),$$
  which implies that the output block starting at a position r, $0 \leq r \leq PL - SL$, is defined by
$$(z_r, \cdots, z_{r+SL-1}) = OUTBLOCK(\pi^r(q_{\text{init}}(IV,k))).$$

Each of the two constructions defines a distinguishing game between the two players Alice, the secret holder and legal user, and Eve, the attacker. Eve is assumed to have unbounded computational power and to have black-box access to the components of the cipher and to the output keystream, i.e., she is allowed to pose component oracle queries to the components MIX and $OUTBLOCK$ and construction oracle queries for blocks of size SL of keystream packets corresponding to initial values x of Eve's choice. After a predefined number of oracle queries, Eve has to decide whether this keystream stems from a random source, i.e., a source which generates for each initial value a truly random bitstream of length PL, or whether it stems from a pseudorandom source, i.e., a stream cipher designed according to the construction under consideration.

Definition 2 (The Distinguishing Game)**.** The parameters KL, IVL, $VIVL$, PL, SL, and VSL and the function $\pi : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ have the same meaning as in Definition 1 and fulfill, for each of the two constructions, the respective conditions. We now describe the game:

(i) First, Alice chooses randomly and w.r.t. the uniform distribution a secret 5-tuple $\omega = (b_\omega, k_\omega, P_\omega, f_\omega, e_\omega)$, where

- $b_\omega \in \{0,1\}$ indicates pseudorandom case or random case,
- $k_\omega \in \{0,1\}^{KL}$ is the secret key,
- $P_\omega : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ is a valid permutation (definition see below) and corresponds to the mixing function,
- $f_\omega : \{0,1\}^{SL} \longrightarrow \{0,1\}$ corresponds to the output bit function,
- $e_\omega : \{0,1\}^{IVL} \times \{0, \cdots, PL-1\} \longrightarrow \{0,1\}$ defines the random bitstream generator.

Here, the definition of a valid permutation depends on the construction. In the Large-State-Small-Key case, each bijective mapping $P_\omega : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ is a valid permutation. For the Continuous-IV-Use construction, P_ω is required to leave the non-volatile part of the state constant, i.e., for each $z \in \{0,1\}^{VSL}$ and $x \in \{0,1\}^{SL-VSL}$, there is some $z' \in \{0,1\}^{VSL}$ such that

$$P_\omega(x,z) = (x,z').$$

This is equivalent to choosing a family $(P_\omega(x,\cdot))_{x\in\{0,1\}^{SL-VSL}}$ of mutually independent random bijective mappings $P_\omega(x,\cdot) : \{0,1\}^{VSL} \longrightarrow \{0,1\}^{VSL}$.

We denote by Ω the probability space consisting of all these 5-tuples together with the uniform distribution. Each elementary event $\omega = (b_\omega, k_\omega, P_\omega, f_\omega, e_\omega)$ defines one further component F_ω, corresponding to the output block function, and one further component E_ω, corresponding to the construction.

(ii) The function F_ω is for all inner states $y \in \{0,1\}^{SL}$ defined by

$$F_\omega(y) = (f_\omega(y), f_\omega(\pi(y)), \cdots, f_\omega(\pi^{SL-1}(y))). \tag{4}$$

(iii) The construction function $E_\omega : \{0,1\}^{IVL} \times \{0, \cdots, PL-SL\} \longrightarrow \{0,1\}^{SL}$ assigns to each initial value x and position value r, $0 \le r \le PL-SL$, the block

$$E_\omega(x,r) = (e_r, \cdots, e_{r+SL-1}) \tag{5}$$

of the keystream packet corresponding to x, starting at position r.

If $b_\omega = 1$, we are in the **random case** and it holds for all $i = 0, \cdots, SL-1$ that

$$e_{r+i} = e_\omega(x)_{r+i}. \tag{6}$$

If $b_\omega = 0$, we are in the **pseudorandom case** and it holds that

$$e_{r+i} = f_\omega(\pi^{r+i}(q_{\text{init}}(x,k_\omega))) = f_\omega(\pi^{r+i}(P_\omega(x,k_\omega))), \tag{7}$$

which is equivalent to

$$E_\omega(x,r) = F_\omega(\pi^r(q_{\text{init}}(x,k_\omega))) = F_\omega(\pi^r(P_\omega(x,k_\omega))). \tag{8}$$

(iv) The distinguisher Eve is supposed to be a randomized oracle algorithm of potentially unbounded computational power. She aims to find out if $b_\omega = 0$ or $b_\omega = 1$ on the basis of oracle queries of the following types, which she submits to Alice and which will be answered honestly by Alice:

- Eve accesses the mixing component via P/P^{-1}-queries $P(u) =?$ or $P^{-1}(v) =?$ for inputs $u, v \in \{0,1\}^{SL}$, which are answered by Alice with $P_\omega(u)$, resp. $(P_\omega)^{-1}(v)$.
- Eve accesses the output component via F-queries $F(y) =?$ for inner states $y \in \{0,1\}^{SL}$, which are answered by Alice with the keystream block $F_\omega(y)$ as defined in Relation (4).
- Eve accesses the construction via E-queries for input pairs (x, r), where $x \in \{0,1\}^{IVL}$ and $0 \leq r \leq PL - SL$, which are answered by Alice with the keystream packet block $E_\omega(x, r)$ as defined by the relations (5),(6),(7),(8).

(v) We suppose that in each computation, Eve poses the same number M of oracle queries and finishes the computation with some output $b \in \{0,1\}$. The advantage $Adv(M)$ reached by Eve with M oracle queries is defined to be

$$\begin{aligned} Adv(M) &= \left| \Pr_{\omega \in_U \Omega}[\text{Eve outputs } 1 | b_\omega = 1] - \Pr_{\omega \in_U \Omega}[\text{Eve outputs } 1 | b_\omega = 0] \right| \\ &= \left| \Pr_{\omega \in_U \Omega}[\text{Eve outputs } 0 | b_\omega = 1] - \Pr_{\omega \in_U \Omega}[\text{Eve outputs } 0 | b_\omega = 0] \right|. \end{aligned}$$

Obviously, TMD-TO attacks against a generic stream cipher construction can be formulated in a straightforward way as strategies for Eve in the corresponding distinguishing game, where the overall number of oracle queries lower bounds the overall time consumption $P + T$ of the attack, and the number of construction queries corresponds to the data consumption D. Note that not only component but also construction queries contribute to either P or T, as the corresponding data blocks would be useless without processing them in some way.

In our security lower bound proofs, we make use of the fact that the state transition function π defines an undirected graph structure $G_\pi = (V_\pi, E_\pi)$ on $V_\pi = \{0,1\}^{SL}$ with $E_\pi = \{(v, \pi(v)), v \in \{0,1\}^{SL}\}$. As π is bijective, the connected components of G_π, which we call π-components, are simple circuits of size at least PL. This graph structure implies the following distance metric on $\{0,1\}^{SL}$.

Definition 3.

- The π-distance $dist_\pi(v, v')$ of inner states $v, v' \in V_\pi = \{0,1\}^{SL}$ is defined to be ∞ if v and v' belong to different π-components. Otherwise, it is defined to be the number of edges of a shortest path connecting v and v' in G_π.
- For each $v \in V_\pi = \{0,1\}^{SL}$ and $s \geq 0$, we define the (π, s)-environment $Env^s_\pi(v) \subseteq \{0,1\}^{SL}$ of v as

$$Env^s_\pi(v) = \{v' \in \{0,1\}^{SL}; dist_\pi(v, v') \leq s\}.$$

  Note that $Env^s_\pi(v) = \{\pi^{-s}(v), \cdots, \pi^{-1}(v), v, \pi(v), \cdots, \pi^s(v)\}$, which implies that $|Env^s_\pi(v)| = 2s + 1$ if $s \leq PL/2$.
- For each set $Z \subseteq V_\pi = \{0,1\}^{SL}$ and $s \geq 0$, we define the (π, s)-environment $Env^s_\pi(Z) \subseteq \{0,1\}^{SL}$ of Z as

$$Env^s_\pi(Z) = \bigcup_{z \in Z} Env^s_\pi(z).$$

Note that inputs v, v' belong to the same π-component if and only if there is some integer r such that $v' = \pi^r(v)$. Note that in this case

$$dist_\pi(v, v') = \min\{|r|, v' = \pi^r(v)\}.$$

3 Security Lower Bounds

Theorem 1. *The advantage $Adv(M)$ reachable by Eve in the distinguishing game described in Definition 2 with $M \geq 0$ oracle queries is bounded*

(i) by

$$\frac{M}{2^{KL} - M} + \frac{(2 \cdot SL + 1) \cdot M^2}{2^{SL} - (2 \cdot SL + 1) \cdot M^2}$$

in the case of the Large-State-Small-Key *construction, and*

(ii) by

$$\frac{M}{2^{KL} - M} + \frac{PL^2 \cdot (2 \cdot SL + 1) \cdot M}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot M}$$

in the case of the Continuous-IV-Use *construction.*

In the remaining part of the section, we give the proof of Theorem 1. The proof is divided into subsections, where in the first two subsections, we introduce a number of notions and notations which are relevant in our context. This enables us to describe the idea of the proof in Subsection 3.3.

3.1 Near Collision and the Friendly Alice

For arbitrary subsets A, B of Ω, we denote by $\Pr[A]$ and by $\Pr_B[A] = \Pr[A|B]$ the probability for the event $\omega \in A$, resp. the probability for the event $\omega \in A$ conditioned to the event $\omega \in B$, where ω is chosen w.r.t. the uniform distribution over Ω.

Definition 4 (Near Collisions)**.** Let $\omega = (b_\omega, k_\omega, P_\omega, f_\omega, e_\omega) \in \Omega$ be an elementary event and F_ω and E_ω be the output block function and the construction function defined by ω.

- A pair $((x, r), y)$, where $(x, r) \in \{0,1\}^{IVL} \times \{0, \cdots, PL - 1\}$ and $y \in \{0,1\}^{SL}$, is called a near EF-collision w.r.t. ω if

$$dist_\pi(\pi^r(q_{\text{init}}(x, k_\omega)), y) \leq SL - 1.$$

- A pair $((x, r), (x', r'))$, where $(x, r), (x', r') \in \{0,1\}^{IVL} \times \{0, \cdots, PL - 1\}$, is called a near EE-collision w.r.t. ω if

$$dist_\pi(\pi^r(q_{\text{init}}(x, k_\omega)), \pi^{r'}(q_{\text{init}}(x', k_\omega))) \leq SL - 1.$$

In the following, we suppose that Alice behaves friendly in the sense that in certain situations she provides some additional information about her secret ω to Eve.

Definition 5 (The Friendly Alice)**.** Let $\omega = (b_\omega, k_\omega, P_\omega, f_\omega, e_\omega)$ denote the secret held by Alice.

- Whenever Eve poses a P-query with input (x, k_ω) or a P^{-1}-query with output (x, k_ω) for some $x \in \{0,1\}^{IVL}$, then, besides giving the correct answer to this query, the friendly Alice makes a **key-recovery announcement**.
- Whenever Eve poses a query with some input which causes a near collision w.r.t. ω with some other input asked before, then, besides giving the correct answer to this query, the friendly Alice makes a **near-collision announcement**.

From now on, we suppose that Alice behaves friendly. As the additional information provided by the friendly Alice does not lower Eve's chances to win the game, each security lower bound proved for the friendly Alice does also hold for the general Alice.

3.2 Formalizing Computations by Transcripts

As described in [CS14], we can assume that Eve is deterministic, i.e., Eve chooses new queries and the final decision deterministically in dependence of the answers of the queries asked before.

Let Ω_0 and Ω_1 denote the subsets of Ω formed by all $\omega = (b_\omega, k_\omega, P_\omega, f_\omega, e_\omega) \in \Omega$ fulfilling $b_\omega = 0$, resp. $b_\omega = 1$.

We identify computations by transcripts τ, which are defined to be the sequence of the M oracle queries posed during the computation, together with the corresponding answers, and followed by a single output bit $b(\tau)$ corresponding to Eve's final decision. Note that possible announcements about key recovery and near collisions are part of the corresponding oracle answers and, thus, part of the transcript.

As Eve is deterministic, it holds that for each $\omega \in \Omega$ there is a unique transcript $\tau(\omega)$ corresponding to the computation of Eve under the condition that Alice has chosen ω. We denote by $\mathcal{T}^M$ the set of all transcripts τ of length M for which there is some $\omega \in \Omega$ with $\tau(\omega) = \tau$.

Let $\tau \in \mathcal{T}^M$ be a transcript and fix some index j, $1 \leq j \leq M$. Then $\tau^{\leq j}$ denotes the sub-transcript defined by the first j queries of τ. We denote by $\mathcal{T}^j$ the set of all sub-transcripts of length j of transcripts from $\mathcal{T}^M$. Moreover, we define $\mathcal{T} = \bigcup_{j=1}^{M} \mathcal{T}^j$.

Each transcript $\tau \in \mathcal{T}$ will be associated with the following sets $\tau_E \subseteq \{0,1\}^{IVL} \times \{r; 0 \leq r \leq PL-1\}$, $\tau_F \subseteq \{0,1\}^{SL}$, $\tau_P \subseteq \{0,1\}^{SL}$, and $\tau_{P^{-1}} \subseteq \{0,1\}^{SL}$ of the inputs of the oracle queries occuring during τ, and a set of keys $K(\tau) \subseteq \{0,1\}^{KL}$:

- $\tau_E = \{(x,r) \in \{0,1\}^{IVL} \times \{r; 0 \leq r \leq PL-1\}; \tau \text{ contains } E\text{-query with input } (x,r)\}$,
- $\tau_F = \{y \in \{0,1\}^{SL}; \tau \text{ contains } F\text{-query with input } y\}$,
- $\tau_P = \{u \in \{0,1\}^{SL}; \tau \text{ contains } P\text{-query with input } u \text{ or } P^{-1}\text{-query with output } u\}$,
- $\tau_{P^{-1}} = \{v \in \{0,1\}^{SL}; \tau \text{ contains } P^{-1}\text{-query with input } v \text{ or } P\text{-query with output } v\}$.
- $K(\tau) \subseteq \{0,1\}^{KL}$ denotes the set of all keys k which occur during τ in the sense that τ_P contains an element of the form (x,k) for some $x \in \{0,1\}^{IVL}$.[3]
- In the case of the Continuous-IV-Use construction, we denote for all non-volatile IV parts $\tilde{x} \in \{0,1\}^{SL-VSL}$

$$\tau_F(\tilde{x}) = \{y \in \tau_F; nv(y) = \tilde{x}\},$$

and

$$\tau_E(\tilde{x}) = \{(x,r) \in \tau_E; nv(x) = \tilde{x}\}.$$

Let us denote by $\mathcal{T}_0^M$ and $\mathcal{T}_1^M$ the set of all transcripts of length M with output bit 0, resp. 1.

For all computations $\tau \in \mathcal{T}^M$, we denote by $\Omega_0(\tau)$ and $\Omega_1(\tau)$ the sets of all elementary events $\omega \in \Omega_0$, resp. $\omega \in \Omega_1$, for which $\tau(\omega) = \tau$, and by $\Omega(\tau)$ the set $\Omega_0(\tau) \cup \Omega_1(\tau)$.

Moreover, for $b \in \{0,1\}$ let Ω_b^0 and Ω_b^1 denote the set of elementary events from Ω_b for which $\tau(\omega)$ outputs 0, resp. 1, and let $\Omega^1 = \Omega_0^1 \cup \Omega_1^1$ and $\Omega^0 = \Omega_0^0 \cup \Omega_1^0$.

For all $b \in \{0,1\}$ and transcripts $\tau \in \mathcal{T}^M$, we denote by

$$\Pr_b[\tau] = \Pr_{\Omega_b}[\Omega_b(\tau)]$$

[3] Remember from Definition 1 that in both constructions, the loading state $q_{\text{load}} \in \{0,1\}^{SL}$ is formed by concatenating IV and key.

the probabilities of the transcript τ in the pseudorandom case ($b = 0$) and the random case ($b = 1$).

Note that the advantage $Adv(M)$ can be written as

$$\begin{aligned} Adv(M) &= \left| \Pr_{\Omega_0}\left[\Omega_0^0\right] - \Pr_{\Omega_1}\left[\Omega_1^0\right] \right| = \left| \Pr_{\Omega_0}\left[\Omega_0^1\right] - \Pr_{\Omega_1}\left[\Omega_1^1\right] \right| \\ &= \left| \sum_{\tau \in \mathcal{T}_1^M} \Pr_0[\tau] - \Pr_1[\tau] \right| = \left| \sum_{\tau \in \mathcal{T}_0^M} \Pr_0[\tau] - \Pr_1[\tau] \right|. \end{aligned} \tag{9}$$

3.3 Bad Elementary Events and Bad Transcripts and the Idea of the Proof of Theorem 1

Definition 6 (Badness)**.** An elementary event ω is called bad if during the computation $\tau(\omega)$ the correct key k_ω or a near collision (i.e., a near EF-collision or a near EE-collision) is discovered. Here, the correct key k_ω is considered to be discovered during $\tau(\omega)$ if $\tau(\omega)$ contains a P-query with input (x, k_ω) or a P^{-1}-query with output (x, k_ω) for some $x \in \{0,1\}^{IVL}$. An elementary event ω is called good if it is not bad.

For all $b \in \{0,1\}$, we denote by Ω_b^{bad} and Ω_b^{good} the set of all elementary events in Ω_b which are bad, resp. good. Moreover, let $\Omega^{\text{bad}} = \Omega_0^{\text{bad}} \cup \Omega_1^{\text{bad}}$ and $\Omega^{\text{good}} = \Omega_0^{\text{good}} \cup \Omega_1^{\text{good}}$.

A transcript is called bad if it contains some key-recovery announcement or some near-collision announcement.

For all $b \in \{0,1\}$, we denote by $\mathcal{T}_b^{M,\text{bad}}$ and $\mathcal{T}_b^{M,\text{good}}$ the set of all transcripts in $\mathcal{T}_b^M$ which are bad, resp. good. Moreover, let $\mathcal{T}^{M,\text{bad}} = \mathcal{T}_0^{M,\text{bad}} \cup \mathcal{T}_1^{M,\text{bad}}$ and $\mathcal{T}^{M,\text{good}} = \mathcal{T}_0^{M,\text{good}} \cup \mathcal{T}_1^{M,\text{good}}$.

Note here that in the CONTINUOUS-IV-USE mode, near EE-collisions $(x, r), (x', r')$ have the property that the non-volatile parts of the initial values x and x' coincide.

The next lemma shows that with good transcripts, the random and the pseudorandom case cannot be distinguished.

Lemma 1. *For all transcripts $\tau \in \mathcal{T}^{M,\text{good}}$, it holds $\Pr_0[\tau] - \Pr_1[\tau] = 0$.*

Proof: Let us fix an arbitrary good transcript $\tau \in \mathcal{T}^{M,\text{good}}$. As τ does not contain near collisions, it holds that in both cases, from Eve's point of view, the answers to the E-queries with inputs (x, r) and (x', r'), $x \neq x'$, and the answers to the E-queries and to the F-queries are mutually independent random variables which are all distributed according to the uniform distribution over $\{0,1\}^{SL}$.

This allows to construct the following bijective mapping from $\Omega_0(\tau)$ to $\Omega_1(\tau)$, assigning to each elementary event $\omega = (0, k_\omega, P_\omega, f_\omega, e_\omega)$ an elementary event $\bar{\omega} = (1, k_{\bar{\omega}}, P_{\bar{\omega}}, f_{\bar{\omega}}, e_{\bar{\omega}})$ which is defined as follows:

- $k_\omega = k_{\bar{\omega}}$ and $P_\omega = P_{\bar{\omega}}$.
- For all $(x, r) \in \tau_E$, exchange the function values of $E_\omega(x, r)$ with the function values of $F_\omega(\pi^r(q_{\text{init}}(x, k_\omega)))$, i.e.,
  - $F_{\bar{\omega}}(\pi^r(q_{\text{init}}(x, k_\omega))) := E_\omega(x, r)$.
  - $E_{\bar{\omega}}(x, r) := F_\omega(\pi^r(q_{\text{init}}(x, k_\omega)))$.

From the fact that $k_\omega \notin K(\tau)$ for all $\omega \in \Omega_0(\tau) \cup \Omega_1(\tau)$ and as τ does not contain near-collision announcements, it follows that the mapping described above is correctly defined and bijective. The existence of a bijective mapping from $\Omega_0(\tau)$ to $\Omega_1(\tau)$ proves Lemma 1. □

Lemma 1 implies

Lemma 2. $Adv(M) \leq \Pr_{\Omega}[\Omega^{\text{bad}}]$.

Proof: By Lemma 1 and Relation 9, it holds

$$\begin{aligned}
2 \cdot Adv(M) &= \left| \sum_{\tau \in \mathcal{T}_1^M} \Pr_0[\tau] - \Pr_1[\tau] \right| + \left| \sum_{\tau \in \mathcal{T}_0^M} \Pr_0[\tau] - \Pr_1[\tau] \right| \\
&= \left| \sum_{\tau \in \mathcal{T}_1^{M,\text{bad}}} \Pr_0[\tau] - \Pr_1[\tau] \right| + \left| \sum_{\tau \in \mathcal{T}_0^{M,\text{bad}}} \Pr_0[\tau] - \Pr_1[\tau] \right| \\
&\leq \sum_{\tau \in \mathcal{T}^{M,\text{bad}}} \left| \Pr_0[\tau] - \Pr_1[\tau] \right| \\
&\leq \sum_{\tau \in \mathcal{T}^{M,\text{bad}}} \Pr_0[\tau] + \Pr_1[\tau] \\
&\leq \sum_{\tau \in \mathcal{T}^{M,\text{bad}}} 2 \cdot \Pr_{\Omega}[\tau] = 2 \cdot \Pr_{\Omega}[\Omega^{\text{bad}}].
\end{aligned}$$

In the last line we used the fact that $\Pr_{\Omega}[\Omega_0] = \Pr_{\Omega}[\Omega_1] = \frac{1}{2}$. □

For estimating the probability $\Pr_{\Omega}[\Omega^{\text{bad}}]$ of bad elementary events, we slightly change the perspective of the computational behaviour of Eve. So far, each computation of Eve has $M+1$ rounds, i.e., M rounds in each of which Eve poses an oracle query, followed by round $M+1$ in which Eve propagates her final decision.

We assume now that the computation stops immediately if Eve manages to pose a query in such a way that the corresponding answer makes the computation bad. This implies that $\Pr_{\Omega}[\Omega^{\text{bad}}]$ equals the probability that Eve stops in some round j, $1 \leq j \leq M$.

We fix some arbitrary round number j, $1 \leq j \leq M$. If Eve has completed the first $j-1$ rounds without stopping with some good transcript $\tau \in \mathcal{T}^{M,\text{good}}$, then Eve chooses deterministically the j-th query $q(\tau)$ in dependence of τ. The computation stops with the answer to $q(\tau)$ with probability $\Pr_{\Omega(\tau)}[Bad(\tau)]$, where $Bad(\tau)$ denotes the set of all elementary events $\omega \in \Omega(\tau)$ for which the next query along $\tau(\omega)$ makes $\tau(\omega)$ bad (i.e., $\tau(\omega)^{\leq j-1} = \tau$ is still good and $\tau(\omega)^{\leq j}$ is bad). Consequently, the computation does not stop before round $M+1$, i.e., produces a transcript $\tau \in \mathcal{T}^{M,\text{good}}$, if for all j, $1 \leq j \leq M$, the event $Bad(\tau^{\leq j-1})$ does not happen. This implies

Lemma 3. $\Pr_{\Omega}[\Omega^{\text{bad}}] \leq \max\left\{ \sum_{j=1}^{M} \Pr_{\Omega(\tau^{\leq j-1})}\left[Bad(\tau^{\leq j-1})\right] ; \tau \in \mathcal{T}^{M,\text{good}} \right\}$. □

We prove Theorem 1 by carefully bounding the probabilities $\Pr_{\Omega(\tau)}[Bad(\tau)]$ for transcripts $\tau \in \mathcal{T}^{j-1,\text{good}}$, $1 \leq j \leq M$. In particular, we show

Lemma 4. *For all j, $1 \leq j \leq M$, and all $\tau \in \mathcal{T}^{j-1,\text{good}}$, the following holds:*

- *If query $q(\tau)$ is a P-query, then*

$$Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{1}{2^{KL} - (j-1)}. \tag{10}$$

- *If query $q(\tau)$ is a P^{-1}-query, then*

$$Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{1}{2^{KL} - (j-1)} + \frac{1}{2^{SL} - (2 \cdot SL + 1)(j-1)^2} \tag{11}$$

if the construction is Large-State-Small-Key *and*

$$Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{1}{2^{KL} - (j-1)} + \frac{1}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot (j-1)} \tag{12}$$

if the construction is Continuous-IV-Use.

- *If query $q(\tau)$ is an E-query with input (x, r), then*

$$Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{(2 \cdot SL + 1)(j-1)}{2^{SL} - (2 \cdot SL + 1)(j-1)^2} \tag{13}$$

if the construction is Large-State-Small-Key *and*

$$Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{(2 \cdot SL + 1) \cdot |\tau_F(nv(x))| + \tau_E(nv(x))|}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot (j-1)}. \tag{14}$$

if the construction is Continuous-IV-Use.

- *If query $q(\tau)$ is an F-query, then*

$$Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{(2 \cdot SL - 1)(j-1)}{2^{SL} - (2 \cdot SL + 1)(j-1)^2} \tag{15}$$

if the construction is Large-State-Small-Key *and*

$$Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{(2 \cdot SL - 1) \cdot PL^2}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot (j-1)} \tag{16}$$

if the construction is Continuous-IV-Use.

Due to space restrictions, the proof of Lemma 4 has been shifted into appendices A and B.

Together with Lemma 3, we obtain

$$Pr_{\Omega}[\Omega^{\text{bad}}] \leq \frac{M}{2^{KL} - M} + \frac{M^2 \cdot (2 \cdot SL + 1)}{2^{SL} - (2 \cdot SL + 1) \cdot M^2} \tag{17}$$

if the construction is Large-State-Small-Key and

$$Pr_{\Omega}[\Omega^{\text{bad}}] \leq \frac{M}{2^{KL} - M} + \frac{M \cdot PL^2 \cdot (2 \cdot SL + 1)}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot M} \tag{18}$$

if the construction is Continuous-IV-Use.

Relation (18) holds as for each value $\tilde{x} \in \{0,1\}^{VIVL}$ during each computation of Eve there are at most $2^{VIVL} \cdot PL \leq PL^2$ E-queries with input (x, r) fulfilling $nv(x) = \tilde{x}$.

Together with Lemma 2, relations (17) and (18) prove Theorem 1.

4 Discussion

We derived a tight security bound of $\min\{KL, SL/2\}$ on the resistance of the Large-State-Small-Key construction (underlying Trivium and Grain v1) against TMD-TO attacks and a tight bound of $\min\{KL, VSL - \log_2(PL)\}$ on the resistance of the Continuous-IV-Use construction against TMD-TO attacks.

In particular, the latter bound provides design guidance for future instances of Continuous-IV-Use stream ciphers which realize the common security level of 80 bits w.r.t. TMD-TO attacks. A corresponding choice of parameters would be a volatile state length of $VSL = 100$ bits, a key length of $KL = 80$ bits, an IV length of $IVL = 80$ bits, and a packet length of $PL = 2^{20}$ bits. The loading state to a key-IV pair (IV, k) would here be $(IV|k)$, where $VIVL = 20$ bits of the IV would not be continuously employed during keystream generation but only enter the volatile part of the loading state in the classical way known, e.g., from Trivium and Grain v1.

We consider the design of a corresponding practical instantiation a promising next step in the search for ultra-lightweight stream ciphers. In fact, it would be the first such cipher with a volatile state length below 160 bits that still offers (even *provable*) 80-bit security against generic TMD-TO-based inner state recovery *and* distinguishing.

References

[AH18] Vahid Amin Ghafari and Honggang Hu. Fruit-80: A Secure Ultra-Lightweight Stream Cipher for Constrained Environments. *Entropy*, 20(3):180, 2018.

[AM15] Frederik Armknecht and Vasily Mikhalev. On Lightweight Stream Ciphers with Shorter Internal States. In Gregor Leander, editor, *Fast Software Encryption: 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, pages 451–470. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[Bab95] Steve H. Babbage. Improved "exhaustive search" attacks on stream ciphers. In *Security and Detection, 1995., European Convention on*, pages 161–166, May 1995.

[BBV12] Steve Babbage, Julia Borghoff, and Vesselin Velichkov. D.SYM.10 - The eSTREAM Portfolio in 2012. eSTREAM: the ECRYPT Stream Cipher Project, 2012. http://www.ecrypt.eu.org/ecrypt2/documents/D.SYM.10-v1.pdf.

[BGW99] Marc Briceno, Ian Goldberg, and David Wagner. A pedagogical implementation of A5/1, 1999. Available at http://www.scard.org/gsm/a51.html.

[BS00] Alex Biryukov and Adi Shamir. Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[CP05] Christophe De Cannière and Bart Preneel. Trivium – Specifications. eSTREAM: the ECRYPT Stream Cipher Project, 2005. http://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf.

[CS14] Shan Chen and John Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer Berlin Heidelberg, 2014.

[Gol96] Jovan Dj. Golić. On the security of nonlinear filter generators. In Dieter Gollmann, editor, *Fast Software Encryption: Third International Workshop Cambridge, UK, February 21–23 1996 Proceedings*, pages 173–188. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.

[GT15] Peter Gazi and Stefano Tessaro. Secret-key cryptography from ideal primitives: A systematic overview. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5, April 2015.

[Hel80] Martin Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, Jul 1980.

[HJM06] Martin Hell, Thomas Johansson, and Willi Meier. Grain - A Stream Cipher for Constrained Environments. eSTREAM: the ECRYPT Stream Cipher Project, 2006. http://www.ecrypt.eu.org/stream/p3ciphers/grain/Grain_p3.pdf.

[HK18] Matthias Hamann and Matthias Krause. On stream ciphers with provable beyond-the-birthday-bound security against time-memory-data tradeoff attacks. *Cryptography and Communications*, 10(5):959–1012, Sep 2018.

[HKM17a] Matthias Hamann, Matthias Krause, and Willi Meier. A Note on Stream Ciphers that Continuously Use the IV. Cryptology ePrint Archive, Report 2017/1172, 2017. https://eprint.iacr.org/2017/1172.

[HKM17b] Matthias Hamann, Matthias Krause, and Willi Meier. LIZARD – A Lightweight Stream Cipher for Power-constrained Devices. *IACR Transactions on Symmetric Cryptology*, 2017(1):45–79, 2017.

[HKMZ17] Matthias Hamann, Matthias Krause, Willi Meier, and Bin Zhang. Design and analysis of small-state grain-like stream ciphers. *Cryptography and Communications*, Nov 2017.

[MAM17] Vasily Mikhalev, Frederik Armknecht, and Christian Müller. On Ciphers that Continuously Access the Non-Volatile Key. *IACR Transactions on Symmetric Cryptology*, 2016(2):52–79, 2017.

[Pat09] Jacques Patarin. The "coefficients H" technique. In RobertoMaria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer Berlin Heidelberg, 2009.

A The Structure of the Probability Space $\Omega(\tau)$

We fix some index j, $1 \leq j \leq M$, and a transcript $\tau \in \mathcal{T}^{j-1,\text{good}}$. We have to derive an upper bound for the probability $\Pr_{\Omega(\tau)}[Bad(\tau)]$ that an elementary event $\omega \in \Omega_b(\tau)$ becomes bad with the j-th query along $\tau(\omega)$.

In this section, we first analyze the structure of the probability space $\Omega(\tau)$.

Note that an elementary event $\omega = (b_\omega, k_\omega, P_\omega, f_\omega, e_\omega)$ belongs to $\Omega(\tau)$ if and only if all of the following conditions are satisfied:

(a) $P_\omega|_{\tau_P}$ is consistent with the answers to all P, P^{-1}-queries contained in τ.

(b) $F_\omega|_{\tau_F}$ is consistent with the answers to all F-queries contained in τ.

(c) If $b = 1$ (random case), then $E_\omega|_{\tau_E}$ is consistent with the answers to all E-queries contained in τ.

(d) If $b = 0$ (pseudorandom case), then for all inputs $(x, r) \in \tau_E$ the answer is

$$F_\omega(\pi^r(q_{\text{init}}(x, k_\omega))),$$

where the definition of the packet initial state $q_{\text{init}}(x, k_\omega)$ depends on in which mode the game is played (see Definition 2).

Moreover, as τ is good it holds

(e) $k_\omega \notin K(\tau)$.

(f) For all $(x, r) \in \tau_E$ and $y \in \tau_F$, it holds

$$dist_\pi\left(\pi^r(q_{\text{init}}(x, k_\omega)), y\right) \geq SL.$$

(g) For all $(x, r), (x', r') \in \tau_E$ with $x \neq x'$ it holds

$$dist_\pi\left(\pi^r(q_{\text{init}}(x, k_\omega)), \pi^{r'}(q_{\text{init}}(x', k_\omega))\right) \geq SL.$$

For bounding the probability $\Pr_{\Omega(\tau)}[Bad(\tau)]$ we will first show that the probability space $\Omega(\tau)$ has a very regular structure.

Note that the uniform distribution on $\Omega(\tau)$ induces a probability distribution on the set of keys $\{0,1\}^{KL}$, given through

$$\Pr_{\Omega(\tau)}[k] := \Pr_{\Omega(\tau)}[\{\omega \in \Omega(\tau); k_\omega = k\}],$$

and a probability distribution on the set of all pairs (k, P) of keys $k \in \{0,1\}^{KL}$ and permutations P over $\{0,1\}^{IVL+KL}$, given through

$$\Pr_{\Omega(\tau)}[k, P] := \Pr_{\Omega(\tau)}[\{\omega \in \Omega(\tau); k_\omega = k, P_\omega = P\}].$$

The proof of Lemma 4 is based on the nontrivial observation that these two probability distributions have the following property:

Lemma 5. *For all keys $k, k' \in \{0,1\}^{KL}$ and permutations $P, P' : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ it holds the following:*

(I) From $\Pr_{\Omega(\tau)}[k] > 0$ and $\Pr_{\Omega(\tau)}[k'] > 0$ it follows $\Pr_{\Omega(\tau)}[k] = \Pr_{\Omega(\tau)}[k']$.

(II) From $\Pr_{\Omega(\tau)}[k, P] > 0$ and $\Pr_{\Omega(\tau)}[k', P'] > 0$ it follows

$$\Pr_{\Omega(\tau)}[k, P] = \Pr_{\Omega(\tau)}[k', P'].$$

Moreover, for all keys $k \in \{0,1\}^{KL}$ it holds

(III) $\Pr_{\Omega(\tau)}[k] > 0$ if and only if $k \notin K(\tau)$.

The Proof of Lemma 5:

We start the proof with some technical definitions.

Definition 7 (τ-consistency)**.** A valid permutation $P : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ is called τ-consistent if for all inputs $u \in \tau_P$ it holds that $P(u)$ equals the answer of the P-query with input u, resp. the input of the P^{-1}-query with output u.

Definition 8 (Environment)**.** For all inner states $y \in \{0,1\}^{SL}$ we denote by $Env(y) \in \{0,1\}^{SL}$ the set of all inner states $y' \in \{0,1\}^{SL}$ with $dist_\pi(y, y') \leq SL - 1$, i.e.,

$$Env(y) = \{\pi^{-(SL-1)}(x), \pi^{-(SL-2)}(x), \cdots, \pi^{-1}(x), x, \pi(x), \cdots, \pi^{(SL-1)}(x)\}.$$

For all subsets $Y \subseteq \{0,1\}^{SL}$ we denote by $Env(Y) \subseteq \{0,1\}^{SL}$ the set $Env(Y) = \bigcup_{y \in Y} Env(y)$.

Note that $|Env(y)| = 2 \cdot SL - 1$ and that $|Env(Y)| \leq (2 \cdot SL - 1)|Y|$ for all $Y \subseteq \{0,1\}^{SL}$.

Let us denote by $X \subseteq \{0,1\}^{IVL}$ the set of all $x \in \{0,1\}^{IVL}$ for which there is some r, $0 \leq r \leq PL - 1$, such that $(x, r) \in \tau_E$. For all $x \in X$ we denote

$$\rho(x) = \{r, (x, r) \in \tau_E\}.$$

The proof of Lemma 5 is based on a more detailed characterization of elementary events ω which fulfill conditions (e),(f),(g) formulated at the beginning of Subsection A.

Remember that all $\omega \in \Omega(\tau)$ have the property that $k_\omega \notin K(\tau)$, otherwise τ would contain a key-recovery announcement.

This implies that $(x, k_\omega) \notin \tau_P$ for all $x \in X$.

We will assign to each pair (k, P), where $k \in \{0,1\}^{KL} \setminus K(\tau)$ and $P : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ a valid permutation, an injective mapping $V_{P,k} : X \longrightarrow \{0,1\}^{SL}$, defined for all $x \in X$ by

$$V_{P,k}(x) := P(x,k).$$

The proof of Lemma 5 is based on

Lemma 6. *For all τ-consistent valid permutations $P : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ and keys $k \in \{0,1\}^{KL}$, it holds that $\Pr_{\Omega(\tau)}[k,P] > 0$ if and only if $k \notin K(\tau)$ and $V_{P,k}$ is (τ,k)-collision free.*

Here, an injective mapping $V : X \longrightarrow \{0,1\}^{SL}$ is called (τ,k)-collision free if $V(x) \notin$ Forbidden$_{\tau,k}(x,V)$ for all $x \in X$.

The underlying definition of the sets *Forbidden*$_{\tau,k}(x,V)$ for injective mappings $V : X \longrightarrow \{0,1\}^{SL}$ are driven by the definition of near collisions in the sense that $V(x) \in$ *Forbidden*$_{\tau,k}(x,V)$ would imply a near collision.

In particular, we define

$$\mathit{Forbidden}_\tau(x,V) = \tau_{P^{-1}} \cup \mathit{Coll}_{EE}(x) \cup \mathit{Coll}_{EF}(x),$$

where the definitions of $\mathit{Coll}_{EE}(x)$ and $\mathit{Coll}_{EF}(x)$ depend on the construction.

Definition 9 (Forbidden Sets)**.**

(i) In the case of the Large-State-Small-Key construction we have

$$\mathit{Coll}_{EF}(x) = \bigcup_{r \in \rho(x)} \pi^{-r}\left(\mathit{Env}(\tau_F)\right),$$

and

$$\mathit{Coll}_{EE}(x) = \bigcup_{r \in \rho(x)} \pi^{-r}\left(\mathit{Env}(\{\pi^{r'}(V(x')); x' \in X \setminus \{x\}, r' \in \rho(x')\}\right).$$

(ii) In the case of the Continuous-IV-Use construction let

$$\mathit{Coll}_{EF}(x) = \bigcup_{r \in \rho(x)} \pi^{-r}\left(\mathit{Env}(\tau_F(x))\right),$$

and

$$\mathit{Coll}_{EE}(x) = \bigcup_{r \in \rho(x)} \pi^{-r}\left(\bigcup_{(x',r') \in \tau_E(x), x' \neq x} \mathit{Env}(\pi^{r'}(V(x'))\right),$$

where $\tau_F(x) \subseteq \tau_F$ contains the set of all inner state F-query inputs $y \in \tau_F$ for which $nv(y) = nv(x)$, and $\tau_E(x)$ contains all initial state E-query inputs $(x',r') \in \tau_E$ with $nv(x') = nv(x)$.

Note that in the case of the Large-State-Small-Key construction it holds

$$\begin{aligned}|\mathit{Forbidden}_{\tau,k}(x,V)| &\leq (j-1) + (j-2) + |\rho(x)| \cdot (2 \cdot SL - 1) \cdot (|\tau_E| + |\tau_F|) \\ &\leq |\rho(x)|(2 \cdot SL + 1) \cdot (j-1) \leq (2 \cdot SL + 1) \cdot (j-1)^2. \end{aligned} \tag{19}$$

In the case of the Continuous-IV-Use construction it holds

$$|\mathit{Forbidden}_{\tau,k}(x,V)| \leq (j-1) + (j-2) + |\rho(x)| \cdot (2 \cdot SL - 1) \cdot (|\tau_F(nv(x))| + |\tau_E(nv(x))|)$$

$$\leq (2 \cdot SL + 1) \cdot PL \cdot (|\tau_F(x)| + |\tau_E(x)|) \leq (2 \cdot SL + 1) \cdot PL \cdot (j - 1). \quad (20)$$

The second inequality is due to the fact that $|\rho(x)| \leq PL$.

The Proof of Lemma 6: We start with the if-direction and fix some $\omega \in \Omega(\tau)$. It holds $k_\omega \notin K(\tau)$, otherwise τ would contain a key-recovery announcement which would contradict the assumption that τ is good.

Further we know that there do not occur near collisions during τ which implies that for all $x \in X$ it holds $P_\omega(x, k_\omega) \notin \mathit{Forbidden}_{\tau,k_\omega}(x, V_{P_\omega,k_\omega})$, i.e., that V_{P_ω,k_ω} is (τ, k)-collision free.

For proving the only-if part let us fix some $b \in \{0,1\}$, some key $k \in \{0,1\}^{KL} \setminus K(\tau)$ and some valid permutation $P : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$, so that P is τ-consistent, and $V_{P,k}$ is τ-collision free.

Note first that, as $k \notin K(\tau)$, it holds that (x, k) does not belong to τ_P for all $x \in X$.

The fact that $V_{P,k}$ is τ-collision free implies that the E-queries and the F queries during τ do not produce any near collision.

The only thing which remains to do is to construct functions $f : \{0,1\}^{SL} \longrightarrow \{0,1\}$ and $e : \{0,1\}^{IVL} \times \{0, \cdots, PL-1\} \longrightarrow \{0,1\}$ in such a way the (b, k, P, e, f) belongs to $\Omega(\tau)$.

We do this by constructing the corresponding block output functions $F : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ and $E : \{0,1\}^{IVL} \times \{0, \cdots, PL-1\} \longrightarrow \{0,1\}^{SL}$.

(1) First we define $F(y)$ for all $y \in \tau_F$ to be equal to the answer of the corresponding F-query during τ.

(2) If $b = 0$ (the pseudorandom case) then we have to define for all $(x, r) \in \tau_E$ the value $F(\pi^r(P(x, k)))$ as to be equal to the answer of the E-query with input (x, r) during τ. This can be done without contradiction to F-queries during τ or to other E-queries during τ. This is because for all $y \in \tau_F$ the π-distance between y and $\pi^r(P(x, k))$ is at least SL, and for all $(x', r') \in \tau_E$, $x' \neq x$, the π-distance between $\pi^{r'}(P(x', k))$ and $\pi^r(P(x, k))$ is at least SL.

(3) If $b = 1$ (random case) we define $E(x, r)$ to be equal to the answer of the E-query with input (x, r) during τ for all $(x, r) \in \tau_E$.

At all positions which were not affected by (1),(2),(3) the functions e and f can be defined in an arbitrary way. This proves Lemma 6. □

For completing the proof of Lemma 5 it is sufficient to show the following two claims:

- **Claim 1:** For all $b \in \{0,1\}$, keys $k \in \{0,1\}^{KL} \setminus K(\tau)$ and valid permutations $P : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ which are τ-consistent and for which $V_{P,k}$ is τ-collision free, the number of output bits of the functions e and f, which have to be fixed for ensuring $(b, k, P, f, e) \in \Omega(\tau)$ is the same.

- **Claim 2:** For all keys $k \in \{0,1\}^{KL} \setminus K(\tau)$ the number of valid permutations $P : \{0,1\}^{SL} \longrightarrow \{0,1\}^{SL}$ which are τ-consistent, and for which $V_{P,k}$ is (τ, k)-collision free, is the same.

Note that Claim 1 follows straightforwardly from the construction rule described in items (2) and (3) above.

Claim 2 is equivalent to

- **Claim 3** saying that for all keys $k \in \{0,1\}^{KL} \setminus K(\tau)$ the number of (τ, k)-collision free injective mappings $V : X \longrightarrow \{0,1\}^{SL}$ is the same.

The proof of Claim 3 is obvious as the definition of (τ, k)-collision freeness does not depend on k. □

B The Proof of Lemma 4

In the following we prove Lemma 4 with Lemma 5.

Let us denote by q the query posed by Eve after τ. Note that the type of this query and the input of this query is determined by τ, while the answer depends on which $\omega \in \Omega(\tau)$ is held by Alice. This answer determines if the j-th query along τ_ω makes $\tau(\omega)$ and ω bad (i.e., $\omega \in Bad(\tau)$) or not. Corresponding to the possible types queries we distinguish four cases:

Case 1: The query q is a P-query with input $u = (x,k) \in \{0,1\}^{SL} \setminus \tau_P$. It holds by definition that a P-query cannot cause a new near collision. Thus, the only possibility to generate badness with q is to choose $u = (x,k)$ in such a way that $k = k_\omega$.

As τ is good, Eve knows that $k_\omega \notin K(\tau)$ and that all keys outside $K(\tau)$ are equally likely to be k_ω (see Lemma 5). As $|K(\tau)| \leq j-1$ we obtain

$$\Pr_{\Omega(\tau)}[Bad(\tau) | q \text{ is } P\text{-query}] \leq \frac{1}{2^{KL} - (j-1)} \tag{21}$$

Case 2: The query q is a P^{-1}-query with input $v \in \{0,1\}^{SL} \setminus \tau_{P^{-1}}$. Then the only possibility to generate badness with q is to choose v in such a way that the answer to q belongs to $\{(x,k_\omega); x \in \{0,1\}^{IVL}\}$. This event is the union of the following two events $BadEv_1$ and $BadEv_2$.

$BadEv_1$ corresponds to the case that q is a P^{-1}-query and that Eve manages to choose v in such a way that $v = P_\omega(x,k_\omega)$ for some $x \in X$. As in the proof of Lemma 5, we denote by X the set of all inputs $x' \in \{0,1\}^{IVL}$ for which there is some r', $0 \leq r' \leq PL-1$, such that $(x',r') \in \tau_E$.

We again denote by $V_{k_\omega,P_\omega} : X \longrightarrow \{0,1\}^{SL}$ the mapping assigning to each $x' \in X$ the value $P_\omega(x',k_\omega)$. Lemma 5 implies that Eve knows that $P_\omega(x,k_\omega) \notin Forbidden_{\tau,k_\omega}(x, V_{k_\omega,P_\omega})$.

In the case of the Large-State-Small-Key construction this set has at most $(2 \cdot SL + 1)(j-1)^2$ elements (see Relations (A)), while in case of the Continuous-IV-Use construction it has at most $(2 \cdot SL + 1) \cdot PL \cdot (j-1)$ elements.

All values outside $Forbidden_{\tau,k_\omega}(x, V_{k_\omega,P_\omega})$ are equally likely to be equal to $P_\omega(x,k_\omega)$. This implies that

$$\Pr_{\Omega(\tau)}[BadEv_1] \leq \frac{1}{2^{SL} - (2 \cdot SL + 1)(j-1)^2} \tag{22}$$

if the construction is Large-State-Small-Key, and

$$\Pr_{\Omega(\tau)}[BadEv_1] \leq \frac{1}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot (j-1)} \tag{23}$$

if the construction is Continuous-IV-Use.

$BadEv_2$ corresponds to the case that q is a P^{-1}-query and that $v \neq P_\omega(x,k_\omega)$ for all $x \in \tau_E$ but $P_\omega^{-1}(v)$ falls into $\{(x,k_\omega); x \in \{0,1\}^{IVL} \setminus X\}$.

Lemma 5 ensures that from Eve's point of view all values in $\{0,1\}^{IVL+KL} \setminus \tau_P$ are equally likely to be $P_\omega(v)$. Consequently,

$$\Pr_{\Omega(\tau)}[BadEv_2] \leq \frac{2^{IVL}}{2^{SL} - (j-1)} = \frac{2^{IVL}}{2^{IVL+KL} - (j-1)} = \frac{1}{2^{KL} - (j-1)}. \tag{24}$$

Case 3: The query q is an E-query for some input (x, r), where $0 \leq r \leq PL - 1$ and $x \in \{0,1\}^{IVL}$. We have to distinguish two subcases:

Subcase 3a: $x \notin X$. For all keys $k \in \{0,1\}^{KL}$, mappings $\tilde{V} : X \longrightarrow \{0,1\}^{SL}$, and permutation $P : \{0,1\}^{IVL+KL} \longrightarrow \{0,1\}^{SL}$, we denote by $P(X,k) = \tilde{V}$ the event that $P(x,k) = \tilde{V}(x)$ for all $x \in X$.

We fix some $k \in \{0,1\}^{KL} \setminus K(\tau)$ (which implies that $(x,k) \notin \tau_P$) and some (τ,k)-collision free mapping $\tilde{V} : X \longrightarrow \{0,1\}^{SL}$ and estimate the probability that q makes $\omega \in \Omega(\tau)$ bad under the condition that $k_\omega = k$ and $P_\omega(X,k) = \tilde{V}$.

For all elementary events ω fulfilling this condition we denote by $\tilde{V}_\omega : X \cup \{x\} \longrightarrow \{0,1\}^{SL}$ the mapping $\tilde{V} \cup (x \to P_\omega(x,k))$.

Query q makes ω bad if $P_\omega(x,k) \in \mathit{Forbidden}_{\tau(\omega)^{\leq j},k}(x, \tilde{V}_\omega)$ (see Lemma 6 and Definition 9).

As $\rho(x)$ contains only one element w.r.t. $\tau(\omega)^{\leq j}$, Relations (A) and (A) imply that constructions

$$|\mathit{Forbidden}_{\tau(\omega)^{\leq j}}(x, \tilde{V}_\omega)| \leq (2 \cdot SL + 1) \cdot j$$

for the Large-State-Small-Key construction, and

$$|\mathit{Forbidden}_{\tau(\omega)^{\leq j}}(x, \tilde{V}_\omega)| \leq (2 \cdot SL + 1) \cdot |\tau_F(nv(x))|$$

for the Continuous-IV-Use construction

As all values outside $\tau_{P^{-1}}$ are equally likely to be equal to $P_\omega(x,k)$ it holds

$$\Pr_{\Omega(\tau)}[P_\omega(x,k) \in \mathit{Forbidden}_\tau(x, \tilde{V}_\omega) | k_\omega = k, P_\omega(X, k_\omega) = \tilde{V}] \leq$$

$$\frac{(2 \cdot SL + 1) \cdot j}{2^{SL} - (j-1)}. \tag{25}$$

if the construction is Large-Key-Small-State, and

$$\Pr_{\Omega(\tau)}[P_\omega(x,k) \in \mathit{Forbidden}_\tau(x, \tilde{V}_\omega) | k_\omega = k, P_\omega(X, k_\omega) = \tilde{V}] \leq$$

$$\frac{(2 \cdot SL + 1) \cdot |\tau_F(nv(x))|}{2^{VSL} - (j-1)}. \tag{26}$$

if the construction is Continuous-IV-Use.

Subcase 3b $x \in X$. Note that $r \notin \rho(x)$, otherwise the same query q would have been posed already during τ. We denote $X' = X \setminus \{x\}$.

Now we fix some $k \in \{0,1\}^{KL} \setminus K(\tau)$ (which implies that $(x,k) \notin \tau_P$) and some (τ,k)-collision free mapping $\tilde{V}' : X' \longrightarrow \{0,1\}^{SL}$ and denote by $\Omega(\tau, k, \tilde{V}')$ the set of all elementary events $\omega \in \Omega(\tau)$ for which $k_\omega = k$ and $P_\omega(X', k) = \tilde{V}'$.

For all $\omega \in \Omega(\tau, k, \tilde{V}')$ we denote by $\tilde{V}'_\omega : X \longrightarrow \{0,1\}^{SL}$ the mapping $\tilde{V}' \cup \{(x, P_\omega(x,k))\}$.

We estimate the probability over $\Omega(\tau, k, \tilde{V}')$ of the event that q makes $\omega \in \Omega(\tau)(k, \tilde{V}')$ bad.

As $\tilde{V}'_\omega$ is (τ, k_ω)-collision we know that $P_\omega(x,k) \notin B := \mathit{Forbidden}_{\tau,k_\omega}(x, \tilde{V}'_\omega)$ and that, from Eve's point of view, all values outside of B are equally likely to be equal to $P_\omega(x,k)$.

Moreover, we know that q makes $\omega \in \Omega(\tau, k, \tilde{V}')$ bad if $\tilde{V}'_\omega$ is not $(\tau(\omega)^{\leq j}, k_\omega)$-collision free, where $\tau(\omega)^{\leq j}$ denotes the transcript of length j obtained from τ

by adding the E-query with input (x, r) as j-th query to τ, which corresponds to adding r to $\rho(x)$. This is equivalent to $P_\omega(x, k) \in A \setminus B$, where $A := \mathit{Forbidden}_{\tau(\omega)\leq j, k_\omega}(x, \tilde{V}'_\omega)$.
Note that by Relation (A)

$$|A \setminus B| \leq (2 \cdot S + 1)(j-1), \text{ and}$$
$$|B| \leq (2 \cdot SL + 1)(j-1)^2,$$

in the case of the Large-State-Small-Key construction, and that by Relation (A)

$$|A \setminus B| \leq (2 \cdot SL + 1) \cdot (|\tau_F(nv(x))| + |\tau_E(nv(x))|), \text{ and}$$
$$|B| \leq (2 \cdot SL + 1) \cdot PL \cdot (j-1).$$

in the case of the Continuous-IV-Use construction.
Consequently, in the case of the Large-State-Small-Key construction,

$$\Pr_{\Omega(\tau)(k,\tilde{V}')}[Bad(\tau)|x \in X] \leq \frac{|A \setminus B|}{2^{SL} - |B|}$$
$$\leq \frac{(2 \cdot SL + 1)(j-1)}{2^{SL} - (2 \cdot SL + 1)(j-1)^2}. \tag{27}$$

In the case of the Continuous-IV-Use construction it holds

$$\Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{(2 \cdot SL + 1) \cdot (|\tau_F(nv(x))| + |\tau_E(nv(x))|)}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot (j-1)}. \tag{28}$$

Case 4 q is an F-query for some input $y \notin \tau_F$.

We consider first the case of the Large-State-Small-Key construction. We fix an arbitrary key $k \in \{0,1\}^{KL} \setminus K(\tau)$ and a (τ, k)-collision free mapping $V : X \longrightarrow \{0,1\}^{SL}$ and denote by $\Omega(\tau, k, V)$ the set of all $\omega \in \Omega(\tau)$ with $k_\omega = k$ and $P_\omega(X, k) = V$.

For all $\omega \in \Omega(\tau, k, V)$ it holds that q makes ω bad if and only if there is some $(x, r) \in \tau_E$ such that $P_\omega(x, k)$ belongs to $Env(\pi^{-r}(y))$, a set of size at most $(2 \cdot SL - 1)$.

Moreover, from Eve's point of view, each point outside $\mathit{Forbidden}_{\tau,k}(x, V)$, a set of size at most $(2 \cdot SL + 1)(j-1)^2$, is equally likely to be equal to be $P_\omega(x, k)$. Consequently, the probability that q makes $\omega \in \Omega(\tau, k, V)$ bad is at most

$$\frac{(2 \cdot SL - 1)(j-1)}{2^{SL} - (2 \cdot SL + 1)(j-1)^2}.$$

Let us now consider the case of the Continuous-IV-Use construction.

We write y as $y = (\tilde{x}, z)$ for $z \in \{0,1\}^{VSL}$ and $\tilde{x} \in \{0,1\}^{IVL - VIVL}$. If $\tilde{x} \neq nv(x)$ for all $x \in X$ than $\Pr_{\Omega(\tau)}[Bad(\tau)] = 0$.

Otherwise, for all $x \in X$ with $\tilde{x} \neq nv(x)$ it holds that q makes an elementary event $\omega \in \Omega(\tau)$ bad if and only if

$$dist_\pi\left(\pi^r\left(x, P_\omega(x, k_\omega)\right), (x, z)\right) \leq SL - 1 \tag{29}$$

for some $r \in \rho(x)$.

As $\tau_E(x)$ contains at most $2^{VIVL} \cdot PL \leq PL^2$ queries (x, r) with $\tilde{x} = nv(x)$ we obtain by the same arguments used in Subcase 3b that

$$\Pr_{\Omega(\tau)}[Bad(\tau)] \leq \frac{PL^2 \cdot (2 \cdot SL - 1)}{2^{VSL} - (2 \cdot SL + 1) \cdot PL \cdot (j-1)}. \tag{30}$$

□