

On the Hardness of Trivium and Grain with respect to Generic Time-Memory-Data Tradeoff Attacks

Matthias Krause

University of Mannheim
krause@uni-mannheim.de

Abstract. Time-Memory-Data tradeoff attacks (TMD-attacks) like those of Babbage [1], Biryukov and Shamir [2] and Dunkelman, Keller [5] reduce the security level of keystream generator based-stream ciphers to $L/2$, where L denotes the inner state length. This is one of the reasons why stream ciphers like Trivium [3] and Grain [8] use a session key length n of at most $L/2$. In this paper, we deal with the question if this *small key large state* design principle really provides the maximal possible security of $\min\{n, \frac{L}{2}\}$ w.r.t. to TMD-attacks, or if it opens the door for new TMD-attack approaches, which possibly allow to discover a secret inner state and/or the secret session key with cost essentially lower than 2^n . We give an answer by analyzing the security of stream ciphers like Trivium and Grain w.r.t. generic TMD-attacks in an appropriate random oracle model. We show that each TMD-attacker with TMD-costs bounded by M can only achieve an advantage of $\min\left\{\frac{2M}{2^n - M}, \frac{(8L-4)M^2}{2^L - (4L-2)M^2}\right\}$ for distinguishing a truly random bitstream from a pseudorandom bitstream generated by the stream cipher. This lower bound matches the security upper bounds defined by exhaustive key search and the classical TMD-attacks mentioned above.

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

1 Introduction

Security proofs in a so-called ideal primitive model (IPM) play an important role in the context of the security analysis of secret-key cryptographic constructions (see, e.g., [7] for a systematic overview). Typically, in an IPM, the components of the constructions are assumed to be idealized and the security proofs are w.r.t. computationally unbounded attackers who have black-box oracle access to the components and the construction and are only bounded in the number of oracle calls. While a large number of IPM-security results were achieved in the last decade for constructions like, e.g., block ciphers, operation modes of

block ciphers, block cipher-based hash functions¹, very little is known about IPM-approaches to stream cipher security.

In this paper, we present an IPM-approach for analyzing the security of stream ciphers against generic Time-Memory-Data tradeoff attacks (for short, TMD-attacks) and apply this method to the stream ciphers Trivium [3] and Grain v1 [8], which both belong to the final eSTREAM portfolio². In contrast to other practically used stream ciphers like the Bluetooth standard E0 [12] and the GSM standard [10], both ciphers use the *small key large state* design principle, i.e., they use a comparatively small length of $n = 80$ for the symmetric session key but a much larger inner state length L ($L = 160$ for Grain and $L = 288$ for Trivium). One reason for the large inner state length lies in the vulnerability of KSG-based stream ciphers against TMD-attacks (like those of Babbage [1], Biryukov and Shamir [2] and Dunkelman, Keller [5]), which allow to discover a secret inner state with time-, memory- and data costs of around $2^{\frac{L}{2}}$. Consequently, for having a chance to build a cipher with n -bit security, one has to choose an inner state length of $L \geq 2n$.

Trivium and Grain start a session by loading the concatenation (x, k) of the n -bit session key k and a public $(L - n)$ -bit string x as the starting inner state into the inner state registers, where x is the concatenation of a 64-bit initial value (IV) and some predefined constants. Then a mixing algorithm is applied which transforms, without producing output keystream bits, the starting state into an initial state. Finally, the keystream (corresponding to the initial state) is generated.

The essential question here is if this way of generating the keystream does really provide the maximal possible $\min\{n, \frac{L}{2}\}$ -security w.r.t. to TMD attacks, or if the *small key large state* principle opens the door for new TMD-attack approaches, which possibly allow to discover a secret inner state and/or the secret session key with cost essentially lower than 2^n .

With this paper, we give an answer to this question. We model a generic TMD-attack scenario for a class of stream ciphers including Trivium and Grain by a new random oracle model and show the following: Each TMD-attacker who poses M oracle queries achieves an advantage of at most

$$\min \left\{ \frac{2M}{2^n - M}, \frac{(8L - 4)M^2}{2^L - (4L - 2)M^2} \right\} \quad (1)$$

for distinguishing a truly random bitstream from a pseudorandom bitstream generated by our stream cipher w.r.t. a random session key. This lower bound matches the upper bounds defined by exhaustive key search and by the classical TMD-attacks in [1], [2], and [5]. It implies that for all $\epsilon > 0$, each TMD-attack with cost $O(2^{(1-\epsilon)n})$ can achieve only an exponentially small distinguishing advantage.

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.

² See e.g. <http://www.ecrypt.eu.org/stream/portfolio-revision1.pdf>.

In the remaining part of this introduction we describe stream ciphers, TMD-attacks and our results in more detail.

Stream ciphers are symmetric encryption algorithms intended for encrypting, in an online manner, plaintext bitstreams X which have to pass an insecure channel. Encryption is performed via bitwise addition of a keystream $S = S(k, x)$, which is generated in dependence of a secret symmetric session key k and, possibly, a public initial value x . 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 (for short: KSG).

KSGs are clockwise working devices which can be formally specified by finite automata, defined by an inner state length L and the corresponding set of inner states $\{0, 1\}^L$, a state update function $\pi : \{0, 1\}^L \rightarrow \{0, 1\}^L$ and an output function $out : \{0, 1\}^L \rightarrow \{0, 1\}^*$. Starting from an initial state q_1 , in each clock cycle $i \geq 1$, the KSG produces a piece of keystream $z_i = out(q_i)$ (usually, one bit long) and changes the inner state according to $q_{i+1} = \pi(q_i)$. The output bitstream $S(q_1)$ is defined by concatenating all the outputs $z_1 z_2 z_3 \dots$.

For many KSG-based stream ciphers, the keystream generation process of a session can be divided into the following four phases:

- (1) A session key generation phase, in which the secret session key $k \in \{0, 1\}^n$ is generated by running a key-exchange protocol between the legal communication partners. This phase will not be considered in this paper.
- (2) A loading phase, in which k together with an initial value and, possibly, some constants will be loaded into the inner state register cells of the KSG, and which results in a state $q_{load} = q_{load}(x, k) \in \{0, 1\}^L$.
- (3) A mixing phase, in which the KSG, without producing keystream bits, transforms the loading state q_{load} into the initial state

$$q_{init} = q_{init}(q_{load}).$$

The goal of this phase is to provide a sufficient level of diffusion and confusion w.r.t to the dependence of the initial state bits from the session key bits and the IV bits.

- (4) The output phase, in which the keystream $S(q_{init})$ for the session is generated in the way described above.

Our security results for KSG-based stream ciphers are based on modeling this process by functions $P, F : \{0, 1\}^L \rightarrow \{0, 1\}^L$.

For all inner states $y \in \{0, 1\}^L$, the function value $F(y) \in \{0, 1\}^L$ is defined as the sequence of the first L keystream bits generated on the inner state y . Standard security requirement of stream ciphers imply that F should be *preimage resistant* in the sense that it is infeasible to compute a value $y \in \{0, 1\}^L$ fulfilling $F(y) = z$ for a given value $z \in \{0, 1\}^L$. Otherwise, it would be feasible to predict the whole keystream on the basis of the first L keystream bits.

We use the function P for modeling the mixing phase, i.e., for all $u \in \{0, 1\}^L$, the value $P(u)$ denotes q_{init} if $u = q_{load}$. Standard efficiency and security assumptions on KSGs imply that P should be an efficiently computable function which behaves like a random function with respect to several combinatorial properties.

Note that in this terminology, the L -block of the first L bits of the keystream $S = S(k, x)$ generated on (k, x) can be expressed as

$$(S_0, \dots, S_{L-1}) = F(P(q_{load}(x, k))) \quad (2)$$

and that the L -block of bits r to $r + L - 1$ of S can be expressed as

$$(S_r, \dots, S_{r+L-1}) = F(\pi^r(P(q_{load}(x, k)))) \quad (3)$$

Trivium: The stream cipher Trivium has an inner state length $L = 288$ distributed over three nonlinear feedback shift registers (NFSRs) of lengths 93, 84, 111. 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 $q_{load}(IV, CONST, k)$ is defined to be the concatenation of the 80-bit session key k , the 64-bit IV and a predefined 144-bit constant $CONST$. In the mixing phase, the KSG is clocked $4 \cdot 288$ times without producing output (see [3] for more details).

Grain v1: The stream cipher Grain v1 has an inner state length $L = 160$ distributed over one NFSR and one linear feedback shift register (LFSR), both of length 80. 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 $q_{load}(IV, CONST, k)$ is defined to be the concatenation of the 80-bit session key k , a 64-bit 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 [8] for more details).

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

$$(S_r, \dots, S_{r+L-1}) = F(\pi^r(P(IV, CONST, k))) \quad (4)$$

During the last decades, many KSGs for practical use have been suggested and many different techniques for cryptanalyzing stream ciphers have been developed (correlation attacks, fast correlation attacks, guess-and-verify attacks, BDD attacks, cube attacks etc.).

Attacks on stream ciphers typically suppose that the attacker knows a piece S of keystream. Typical goals of attacks are to distinguish S from a truly random bitstream, to recover the inner state responsible for S for predicting the whole keystream, or to recover the secret session key.

In this paper, we concentrate on generic TMD tradeoff attacks, which try to reach their goals on the basis of black-box access to the functions F and P and on the basis of a set of keystream pieces.

Let us first recall the idea of the well-known *known-keystream* TMD tradeoff attack of Babbage [1]. Suppose that the attacker knows a part or a set of parts of a keystream, which contain s different L -blocks $z^j \in \{0, 1\}^L$, $j = 1, \dots, s$ of L consecutive keystream bits, and let $Q = \{q^1, \dots, q^s\}$ denote the set of corresponding inner states satisfying $z^j = F(q^j)$.

The attacker generates a set T of pairs $(y, F(y))$ for randomly chosen inner states $y \in \{0, 1\}^L$. If $s \cdot |T| \approx 2^L$, then, with high probability, there will occur a collision, i.e., there is some pair $(y, F(y)) \in T$ such that $F(y) = z^j$ for some j , $j = 1, \dots, s$. Then, with high probability, $y = q^j$, i.e., the attacker can reconstruct the initial state responsible for the keystream and consequently the whole keystream. If $s = |T| = 2^{L/2}$, we obtain an attack which lowers the security level of the cipher to $L/2$.

In this paper, we introduce and analyze the following formal security model for generic *chosen-IV* TMD tradeoff attacks against KSG-based stream ciphers defined by their loading mechanism and functions P and F , which correspond to the mixing and keystream generation algorithm as described above.

We assume that adversaries (named Eve) are randomized oracle algorithms which try to distinguish a pseudorandom scenario, defined by the stream cipher, from a corresponding random scenario. In particular, we suppose that Eve has black-box oracle access to the functions P , P^{-1} and F , which represent the components of the cipher. Moreover, Eve is allowed to pose *construction queries* of the form $E(x, r)$, where $x \in \{0, 1\}^{L-n}$ and r is a natural number. In the pseudorandom scenario, Eve gets as answer the L -block (S_r, \dots, S_{r+L-1}) of the keystream $S = S(x, k)$ generated w.r.t. to the initial value x and a secret session key $k \in \{0, 1\}^n$, i.e.,

$$E(x, r) = F(\pi^r(P(x, k))). \quad (5)$$

In the random scenario, Eve gets as answer the L -block $(b_r^x, \dots, b_{r+L-1}^x)$, where $\{b^x = (b_0^x, b_1^x, b_2^x, \dots); x \in \{0, 1\}^{L-n}\}$ denotes a collection of mutually independent truly random bitstreams.

Note that relation (5) models exactly the state initialization and keystream generation algorithm of stream ciphers like Trivium and Grain (see relation (4))³.

As in similar frameworks for modeling cryptographic algorithms and protocols by random oracle models (like, e.g., the modeling of modern blockciphers by iterated Even-Mansour ciphers, see, e.g., [6], or [4] and the citations therein) we assume our mixing function P to be a random, efficiently invertible permutation.

Concerning the modeling of the keystream generation function F , observe the following. If the stream cipher produces one keystream bit per clock cycle, the function F is π -iterative in the sense that for all inputs $y \in \{0, 1\}^L$ it holds that

³ Note that relation (5) refers to IVs of length $L-n$ and ignores the fact that for Trivium and Grain, a part of the loading state is an IV-independent constant. However, this rather enlarges the power of the attacker.

the suffix of length $L - 1$ of $F(y)$ equals the prefix of length $L - 1$ of $F(\pi(y))$. We reflect this in our random oracle model by assuming F to be a random π -iterative function.

In the preliminary section 2 we first describe how our state transition function $\pi : \{0, 1\}^L \rightarrow \{0, 1\}^L$ defines a metric on $\{0, 1\}^L$ (subsection 2.1). Then, we introduce some basic notions and notations about random and pseudorandom bitstreams (subsection 2.2) and give the exact definition of our security model (subsection 2.3).

In section 3, we formulate and analyze two *known keystream* attacks against stream ciphers of type (4), one corresponding to exhaustive key search, the other to the classical TMD-tradeoff attacks from [1], in our security model.

Section 4 contains our main result, the proof of the nearly tight security lower bound for chosen-IV attacks (see (1)). Our proof does not explicitly use the H -coefficient technique of Patarin [11], but it follows the typical structure of such proofs as it was described, e.g., in [4]. 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.

2 Preliminaries

In the remaining part of this paper we refer to a stream cipher which is defined by parameters n , m and $L = m + n$ corresponding to the secret session key length, the initial value (IV) length and the inner state length, and by functions $\pi, P, F : \{0, 1\}^L \rightarrow \{0, 1\}^L$ corresponding to the state transition function, the mixing algorithm and the keystream generation function of the cipher. As discussed already in section 1, we suppose that for each secret key $k \in \{0, 1\}^n$ and each IV $x \in \{0, 1\}^m$, the stream cipher produces a keystream $S(k, x) = S_0, S_1, S_2, \dots$, where for each $r \geq 0$, the block of the L keystream bits (S_r, \dots, S_{r+L-1}) is defined to be

$$(S_r, \dots, S_{r+L-1}) = F(\pi^r(P(x, k))). \quad (6)$$

We showed in section 1 that Trivium and Grain are stream ciphers of this form.

2.1 The structure on $\{0, 1\}^L$ defined by the state transition function π

We assume in the following that the state transition function π has everywhere a large period in the sense that there is a number $R > 2^n$ which has the property that for all inner states $v \in \{0, 1\}^{m+n}$ the period of the sequence $(\pi^r(v))_{r \geq 0}$ is not smaller than R . This implies that we assume that the stream cipher is not vulnerable against small period attacks in the sense that such attacks cannot do better than exhaustive key search⁴.

⁴ Note here that this large period property could not be proved so far for Trivium and Grain.

Note that π defines an undirected graph structure $G_\pi = (V_\pi, E_\pi)$ on $V_\pi = \{0, 1\}^L$ with $E_\pi = \{(v, \pi(v)), v \in \{0, 1\}^L\}$. As π is bijective, the connected components of G_π , which we call π -components, are simple circuits of size at least R .

Definition 1. – The π -distance $dist_\pi(v, v')$ of $v, v' \in V_\pi = \{0, 1\}^{m+n}$ is defined to be ∞ if v and v' belong to different π -components, and is defined to be the number of edges of a shortest path connecting v and v' in G_π if they belong to the same π -component.

– For each $v \in V_\pi = \{0, 1\}^{m+n}$ and $s \geq 0$ we define the (π, s) -environment $Env_\pi^s(v) \subseteq \{0, 1\}^{m+n}$ of v as

$$Env_\pi^s(v) = \{v' \in \{0, 1\}^{m+n}; dist_\pi(v, v') \leq s\}.$$

Note that $|Env_\pi^s(v)| = 2s + 1$ if $s \leq R/2$.

– For each set $Z \subseteq V_\pi = \{0, 1\}^{m+n}$ and $s \geq 0$ we define the (π, s) -environment $Env_\pi^s(Z) \subseteq \{0, 1\}^{m+n}$ of Z as

$$Env_\pi^s(Z) = \bigcup_{z \in Z} Env_\pi^s(z).$$

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

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

2.2 Modeling pseudorandom and random bistream generation

We refer to stream ciphers (like Trivium and Grain) which generate one keystream bit per state transition. This implies that the keystream generation function $F : \{0, 1\}^L \rightarrow \{0, 1\}^L$ is π -iterative in the following sense:

Definition 2. We call a function $F : \{0, 1\}^{m+n} \rightarrow \{0, 1\}^{m+n}$ to be π -iterative, if for all $y \in \{0, 1\}^{m+n}$ it holds that the suffix of length $m + n - 1$ of $F(y)$ equals the prefix of length $m + n - 1$ of $F(\pi(y))$.

In our **pseudorandom scenario** the ideal F -component will be a random π -iterative function. Note that a truly random π -iterative function can be generated as follows:

- (1) For each π -component $C \subseteq \{0, 1\}^{m+n}$, we fix an arbitrary starting point $y_C^0 \in C$.
- (2) For each π -component C , we generate a truly random bitstream

$$b^C = (b_0^C, \dots, b_{|C|-1}^C)$$

of length $|C|$.

- (3) For each $y \in \{0, 1\}^{m+n}$, the function value $F(y)$ of the corresponding random π -iterative function F is defined as follows: Fix the π -component C for which $y \in C$ and let r be the smallest nonnegative value such that $y = \pi^r(y_C^0)$. Then

$$F(y) = (b_r^C \bmod |C|, \dots, b_{(r+m+n-1) \bmod |C|}^C).$$

We model the **random scenario** by an *ideal random bit generator* which generates for each possible initial value $x \in \{0, 1\}^m$ randomly and independently a truly random bitstream

$$b^x = (b_0^x, \dots, b_{R+m+n-2}^x)$$

of length $R + m + n - 1$.

We identify such families $\{b^x, x \in \{0, 1\}^m\}$ of bitstreams by so-called *iterative functions*

$$E : \{0, 1\}^m \times \{0, \dots, R-1\} \longrightarrow \{0, 1\}^{m+n}.$$

Definition 3. We call a function $E : \{0, 1\}^m \times \{0, \dots, R-1\} \longrightarrow \{0, 1\}^{m+n}$ to be *iterative*, if for all $x \in \{0, 1\}^m$ and r , $0 \leq r \leq R-2$, the suffix of length $m+n-1$ of $E(x, r)$ equals the prefix of length $m+n-1$ of $E(x, r+1)$.

Note that the one-to-one correspondence between families $\{b^x, x \in \{0, 1\}^m\}$ of bitstreams of length $R + m + n - 1$ and iterative functions $E : \{0, 1\}^m \times \{0, \dots, R-1\} \longrightarrow \{0, 1\}^{m+n}$ is defined by the relation

$$E(x, r) = (b_r^x, \dots, b_{r+m+n-1}^x),$$

where $x \in \{0, 1\}^m$ and $0 \leq r \leq R-1$.

2.3 The Distinguishing Game

We analyze the security of stream ciphers of type (6) w.r.t. generic TMD-attacks by analyzing the success probability of a distinguisher Eve of winning the following distinguishing game against Alice. The game can informally be described as follows. According to the ideal primitive model, Alice chooses a random secret session key k , a random function P , and a random π -iterative function F as ideal components for the pseudorandom case, and an iterative function E as component for the random case. Then, Alice generates a secret random bit b . Eve has black box oracle access to the component functions P , P^{-1} and F . Moreover, Eve is allowed to pose queries of type $E(x, r)$ for IVs $x \in \{0, 1\}^m$ and natural numbers r , $0 \leq r \leq R-1$. If $b = 0$, then Alice answers such question according to the pseudorandom real-world scenario (see relation (6)), i.e., with the block of the bits at positions $r, \dots, r+m+n-1$ of the keystream generated with respect to initial value $x \in \{0, 1\}^m$. If $b = 1$, then Alice answers such question according to the random ideal-world scenario, i.e., according to the ideal random bit generator defined by E . Eve's goal is to find out if $b = 0$ or if $b = 1$.

Definition 4. (i) Alice chooses randomly and w.r.t. the uniform distribution a secret 5-tuple $\omega = (b_\omega, P_\omega, F_\omega, k_\omega, E_\omega)$, where

- $b_\omega \in \{0, 1\}$,
- $P_\omega : \{0, 1\}^{m+n} \rightarrow \{0, 1\}^{m+n}$ is a permutation,
- $F_\omega : \{0, 1\}^{m+n} \rightarrow \{0, 1\}^{m+n}$ is a π -iterative function,
- $k_\omega \in \{0, 1\}^n$ is a secret key.
- $E_\omega : \{0, 1\}^m \times \{0, \dots, R-1\} \rightarrow \{0, 1\}^{m+n}$ is an iterative function.

Let us denote by Ω the set of all such 5-tuples, resp. the probability space consisting of all these 5-tuples together with the uniform distribution.

- (ii) The distinguisher Eve is supposed to be a randomized oracle algorithm of potentially unbounded computational power who aims to find out if $b_\omega = 0$ or $b_\omega = 1$. She is allowed to pose component oracle queries of type $P(u) = ?$, or $P^{-1}(v) = ?$, or $F(y) = ?$ for inputs $u, v, y \in \{0, 1\}^{m+n}$, which are answered by Alice by $P_\omega(u)$, $(P_\omega)^{-1}(v)$, or $F_\omega(y)$ respectively. Moreover, Eve can pose construction oracle queries of the form $E(x, r) = ?$, where $0 \leq r \leq R-1$ and $x \in \{0, 1\}^m$, which will be answered by Alice with $F_\omega(\pi^r(P_\omega(x, k_\omega)))$ if $b_\omega = 0$ (the pseudorandom real-world scenario), or with $E_\omega(x, r)$ if $b_\omega = 1$ (the random ideal-world scenario).

We suppose that in each computation, Eve poses the same number M of oracle queries and finishes the computation with some output $b \in \{0, 1\}$.

- (iii) As usual, the advantage $\text{Adv}(M)$ reached by Eve with M oracle queries is defined to be

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

3 Upper Bounds

In this section we prove the following

Theorem 1. In the distinguishing game specified in definition 4, Eve can reach a distinguishing advantage greater $\frac{1}{2}$ with

$$\min \left\{ 3 \cdot 2^n + 1, \left(1 + \frac{1}{m+n} \right) \cdot 2^{\frac{m+n}{2}} + 2 \right\}$$

oracle queries.

Proof: We prove theorem 1 by presenting two simple distinguishing algorithms.

The first performs exhaustive key search:

- 1 Eve fixes some $x \in \{0, 1\}^m$ and asks for $E(x, 0)$.
- 2 For all $k \in \{0, 1\}^n$, Eve asks for $P(x, k)$ and $F(P(x, k))$.
- 3 If there is some $k \in \{0, 1\}^n$ such that $F(P(x, k)) = E(x, 0)$, then Eve outputs 0 (real-world case), if not, then Eve outputs 1 (ideal-world case).

Note that in the real-world case, Eve outputs 0 with probability 1, while in the ideal-world case, the probability that Eve outputs 0 is $2^{-(m+n)} \cdot 2^n = 2^{-m}$. This yields an advantage of $1 - 2^{-m}$ with at most $3 \cdot 2^n + 1$ oracle queries.

The second algorithm corresponds to the classical TMD-attack of Babbage [1] and has the aim to find so-called *structural EF-collisions* $\{(x, r), y\}$, where $x \in \{0, 1\}^m$, $0 \leq r \leq R - 1$, and $y = \pi^r(P_\omega(x, k_\omega))$. Note that for all structural EF-collisions $\{(x, r), y\}$ it holds that $E(x, r) = F(y)$ (i.e., the $E(x, r) = ?$ -query and the $F(y) = ?$ -query yield the same answer).

Let $s = 2^{\frac{m+n}{2}}$. Let us suppose here that $m \leq n$.

- 1 Eve chooses some IV $x \in \{0, 1\}^m$ and generates the corresponding keystream

$$b^x = (b_0^x, \dots, b_{s+m+n-2}^x)$$

of length s corresponding to x . This can be done by concatenating the answers of the sequence of $\lceil \frac{s}{m+n} \rceil$ oracle queries $E(x, 0), E(x, m+n), E(x, 2(m+n)), \dots$.

- 2 Then, Eve does at most s times the following: She chooses randomly and independently an element $y \in \{0, 1\}^{m+n}$. If

$$F(y) = (b_r^x, \dots, b_{r+m+n-1}^x),$$

for some r , $0 \leq r \leq s - 1$, then Eve checks a neighbour block, i.e. asks for $F(\pi^a(y))$, where $a = m + n$ if $r \leq s - n - m - 1$ and $a = -(m + n)$ if not. If

$$F(\pi^a(y)) = (b_{r+a}^x, \dots, b_{r+a+m+n-1}^x)$$

then Eve outputs 0 and stops.

- 3 If Eve did not stop in step 2, then she stops with output 1.

We only sketch the analysis of this algorithm. Let us suppose that Alice has chosen the elementary event $\omega = (b_\omega, P_\omega, F_\omega, k_\omega, E_\omega)$.

In the real-world case where $b_\omega = 0$, we know that for all r , $0 \leq r \leq s - 1$, it holds that

$$(b_r^x, \dots, b_{r+m+n-1}^x) = F_\omega(\pi^r(P_\omega(x, k_\omega))).$$

Let us denote by $T \subseteq \{0, 1\}^{m+n}$ the corresponding set of hidden values, i.e.,

$$T = \{\pi^r(P_\omega(x, k_\omega)); 0 \leq r \leq s - 1\}.$$

We know that with probability around $(1 - (1 - s^{-1})^s) \approx (1 - e^{-1})$, in at least one of the s rounds of step 2 it holds $y \in T$, which implies that Eve stops with output 0 in this round. Consequently, Eve outputs 0 with probability not smaller than $(1 - e^{-1})$, which is greater than $1/2$.

In the ideal-world case where $b_\omega = 1$, in each round, the probability that Eve stops with output 0 in this round is not greater than $s^{-1}2^{-(m+n)}$, i.e., Eve outputs 0 with probability not greater than $2^{-(m+n)}$. \square

4 A Matching Lower Bound

Theorem 2. *The advantage $Adv(M)$ reachable by Eve in the distinguishing game described in definition 4 with $M \geq 0$ oracle queries is bounded by*

$$Adv(M) \leq \min \left\{ \frac{2M}{2^n - M}, \frac{(8(m+n) - 4)M^2}{2^{m+n} - (4(m+n) - 2)M^2} \right\}.$$

Proof: 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 .

Let Ω_0 and Ω_1 denote the subsets of Ω formed by all 5-tuples

$$\omega = (b_\omega, P_\omega, F_\omega, k_\omega, E_\omega) \in \Omega$$

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 corresponding to Eve's final decision.

Let us denote by \mathcal{T}_0^M and \mathcal{T}_1^M the set of all transcripts of length M (i.e., with M query-and-answer-pairs) with output bit 0, resp. 1, and $\mathcal{T}^M = \mathcal{T}_0^M \cup \mathcal{T}_1^M$.

As described in [4], we can assume that Eve is deterministic, i.e., Eve chooses new queries and the final decision deterministically in dependence of the transcript of the queries asked before.

This implies that for each $\omega = (b_\omega, P_\omega, F_\omega, k_\omega, E_\omega) \in \Omega$ there is a unique transcript $\tau(\omega) \in \mathcal{T}^M$ corresponding to the computation of Eve under the condition that Alice has chosen ω .

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 $\tau(\omega) = \tau$.

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

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

the probabilities of the transcript τ in the real world ($b = 0$) and the ideal world ($b = 1$).

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

$$Adv(M) = \left| \sum_{\tau \in \mathcal{T}_1^M} Pr_0[\tau] - Pr_1[\tau] \right|.$$

The idea of proving our theorem is as follows. We will define an elementary event $\omega \in \Omega$ to be *bad*, if the computation $\tau(\omega)$ fulfills some combinatorial properties w.r.t. ω , which will be specified later and which will help Eve to distinguish between the ideal-world and the real-world case. An elementary event

$\omega \in \Omega$ which is not bad is called to be *good*. We will denote by Ω_b^{bad} and Ω_b^{good} the set of all elementary events in Ω_b being bad, resp. good, and by $\Omega_b^{bad}(\tau)$ and $\Omega_b^{good}(\tau)$ the events $\Omega_b^{bad} \cap \Omega_b(\tau)$ and $\Omega_b^{good} \cap \Omega_b(\tau)$.

Furthermore, for all $b \in \{0, 1\}$ and all computations τ , let us denote by

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

and

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

the probabilities that τ occurs as the result of a bad elementary event or as the result of a good elementary event.

Obviously, $Pr_b(\tau) = Pr_b^{bad}(\tau) + Pr_b^{good}(\tau)$.

The proof of theorem 2 will be based on proving two technical results:

- (1) There is an appropriate exponentially small bound $\epsilon > 0$ such that

$$Pr_b^{bad}(\tau) \leq \epsilon \cdot Pr_b(\tau)$$

for all $\tau \in \mathcal{T}^M$ and $b \in \{0, 1\}$.

- (2) For all $\tau \in \mathcal{T}^M$ it holds that

$$Pr_0^{good}(\tau) = Pr_1^{good}(\tau).$$

Results (1) and (2) imply that $Adv(M)$ can be upper bounded as follows.

$$\begin{aligned} Adv(M) &= \left| \sum_{\tau \in \mathcal{T}_1^M} Pr_0(\tau) - Pr_1(\tau) \right| \leq \sum_{\tau \in \mathcal{T}_1^M} |Pr_0(\tau) - Pr_1(\tau)| \\ &= \sum_{\tau \in \mathcal{T}_1^M} \left| Pr_0^{bad}(\tau) + Pr_0^{good}(\tau) - Pr_1^{bad}(\tau) - Pr_1^{good}(\tau) \right| \\ &\leq \sum_{\tau \in \mathcal{T}_1^M} \left| Pr_0^{good}(\tau) - Pr_1^{good}(\tau) \right| + Pr_0^{bad}(\tau) + Pr_1^{bad}(\tau) \\ &\leq \sum_{\tau \in \mathcal{T}_1^M} \left| Pr_0^{good}(\tau) - Pr_1^{good}(\tau) \right| + \epsilon \cdot Pr_0(\tau) + \epsilon \cdot Pr_1(\tau) \\ &= \sum_{\tau \in \mathcal{T}_1^M} \left| Pr_0^{good}(\tau) - Pr_1^{good}(\tau) \right| + \epsilon \cdot \sum_{\tau \in \mathcal{T}_1^M} (Pr_0(\tau) + Pr_1(\tau)) \\ &\leq 2 \cdot \epsilon. \end{aligned}$$

For showing results (1) and (2), we have to introduce some basic notions and notations first.

Definition 5. Each transcript τ will be associated with the following sets $\tau_E \subseteq \{0, 1\}^m \times \{r; 0 \leq r \leq R-1\}$ and $\tau_F, \tau_P, \tau_{P-1} \subseteq \{0, 1\}^{m+n}$ of the inputs of the oracle queries occuring during τ :

- $\tau_E = \{(x, r) \in \{0, 1\}^m \times \{r; 0 \leq r \leq R - 1\}; \tau \text{ contains an } E\text{-query with input } (x, r)\}$,
- $\tau_F = \{y \in \{0, 1\}^{m+n}; \tau \text{ contains an } F\text{-query with input } y\}$,
- $\tau_P = \{u \in \{0, 1\}^{m+n}; \tau \text{ contains a } P\text{-query with input } u \text{ or a } P^{-1}\text{-query with output } u\}$,
- $\tau_{P^{-1}} = \{v \in \{0, 1\}^{m+n}; \tau \text{ contains a } P^{-1}\text{-query with input } v \text{ or a } P\text{-query with output } v\}$.

Definition 6. For each transcript τ and all inputs z of oracle queries in τ we denote by $\tau(z)$ the corresponding answer.

Definition 7. Let $\tau \in \mathcal{T}_M$ be a transcript and fix some index j , $1 \leq j \leq M$. Then $\tau^{\leq j}$ defines the sub-transcript defined by the first j queries of τ .

Definition 8. Given a key $k \in \{0, 1\}^n$, a transcript $\tau \in \mathcal{T}^M$ and some j , $1 \leq j \leq M$, we say that $k \in \{0, 1\}^n$ is contained in $\tau^{\leq j}$ (for short, $k \in \tau^{\leq j}$) if the set $(\tau^{\leq j})_P$ contains an input $u = (x, k) \in \{0, 1\}^{m+n}$ with suffix k .

Now we are ready to define badness of a transcript τ with respect to an elementary event ω , resp. badness of elementary events. Informally, an elementary event ω is bad if the corresponding computation $\tau(\omega)$ yields some critical information which helps to distinguish the pseudorandom from the random case. One critical situation is given if Eve manages to pose a P -query with input (x, k_ω) , which contains the secret session key as suffix. Then, the queries for $E(x, 0)$ and $F(y)$, where $y = P_\omega(x, k_\omega)$ denotes the answer of the critical P -query, yield a collision, and two further queries (e.g., for $E(x, m+n)$ and $F(\pi^{m+n}(y))$) distinguish the pseudorandom from the random case with high probability. Critical situations are also caused by structural EF-collisions $\{(x, r), y\}$, as described in the proof of theorem 1. Here, too, a few further queries allow to distinguish the pseudorandom from the random case with high probability.

There is another type of critical situation, so-called *structural EE-collisions*, which are pairs of inputs $\{(x, r), (x', r')\}$ to E -queries. Such a pair is a structural EE-collision w.r.t. ω if $x \neq x'$, but

$$\pi^r(P_\omega(x, k_\omega)) = \pi^{r'}(P_\omega(x', k_\omega)). \quad (7)$$

Also in this case, the queries for $E(x, r)$ and $E(x', r')$ yield the same answer, and two further queries (e.g., for $E(x, r+m+n)$ and $E(x', r'+m+n)$) allow to distinguish the pseudorandom from the random case with high probability.

According to this, we will call a transcript τ bad with respect to ω , if the secret key k_ω occurs as suffix of the input of a P -query or the output of a P^{-1} -query, or if τ contains a *near* EF-collision $\{(x, r), y\}$ or a *near* EE-collision $\{(x, r), (x', r')\}$ w.r.t. to ω . Near means here that the π -distance of the corresponding inputs y and $\pi^r(P_\omega(x, k_\omega))$, resp. $\pi^r(P_\omega(x, k_\omega))$ and $\pi^{r'}(P_\omega(x', k_\omega))$ is smaller than $2(m+n) - 1$.

The choice of the distance bound $2(m+n) - 1$ is motivated by the observation that if $\text{dist}_\pi(y, y') < 2(m+n) - 1$, then there is at least one point z between y and y' for which $F_\omega(z)$ statistically depends on $F_\omega(y)$ and on $F_\omega(y')$ (see definition 2 of π -iterative functions).

Definition 9. For $\omega = (b_\omega, k_\omega, P_\omega, F_\omega, E_\omega) \in \Omega$, $\tau \in \mathcal{T}^M$ and j , $1 \leq j \leq M$, we say that ω is bad for $\tau^{\leq j}$ if $\tau(\omega) = \tau$ and at least one of the following conditions is fulfilled:

- (1) $k_\omega \in (\tau^{\leq j})_P$.
- (2) $\tau^{\leq j}$ contains a near structural EF-collision, i.e., there is some $(x, r) \in (\tau^{\leq j})_E$ and some $y \in (\tau^{\leq j})_F$ such that

$$\text{dist}_\pi(\pi^r(P_\omega(x, k_\omega)), y) \leq 2(m+n) - 1.$$

- (3) $\tau^{\leq j}$ contains a near structural EE-collision, i.e., there are $(x, r), (x', r') \in (\tau^{\leq j})_E$, where $x \neq x'$, such that

$$\text{dist}_\pi(\pi^r(P_\omega(x, k_\omega)), \pi^{r'}(P_\omega(x', k_\omega))) \leq 2(m+n) - 1.$$

The elementary event ω is called good w.r.t. to $\tau^{\leq j}$ if $\tau(\omega) = \tau$ and ω is not bad w.r.t. $\tau^{\leq j}$.

The elementary event ω is called good resp. bad if it is good resp. bad w.r.t. $\tau(\omega)^{\leq M}$.

For proving result (1), we have to upper bound the fraction $\frac{Pr_b^{bad}(\tau)}{Pr_b(\tau)}$ for all $\tau \in \mathcal{T}_M$ and $b \in \{0, 1\}$.

Note that $\omega \in \Omega_b^{bad}(\tau)$ if and only if there is some j , $1 \leq j \leq M$, such that $\omega \in \Omega^{good}(\tau^{\leq j-1})$ but $\omega \in \Omega^{bad}(\tau^{\leq j})$, i.e., the j -th query along $\tau(\omega)$ makes the secret ω bad. This implies that

$$\Omega_b^{bad}(\tau) = \bigcup_{j=1}^M \Omega_b^{good}(\tau^{\leq j-1}) \cap \Omega_b^{bad}(\tau^{\leq j}),$$

where $\Omega_b^{good}(\tau^{\leq 0}) = \Omega_b(\tau)$.

Consequently

$$\begin{aligned} Pr_b^{bad}(\tau) &= \sum_{j=1}^M Pr_{\Omega_b}[\Omega_b^{good}(\tau^{\leq j-1}) \cap \Omega_b^{bad}(\tau^{\leq j})] \\ &= \sum_{j=1}^M Pr_{\Omega_b}[\Omega_b^{bad}(\tau^{\leq j}) | \Omega_b^{good}(\tau^{\leq j-1})] \cdot Pr_{\Omega_b}[\Omega_b^{good}(\tau^{\leq j-1})] \\ &\leq \sum_{j=1}^M Pr_{\Omega_b}[\Omega_b^{bad}(\tau^{\leq j}) | \Omega_b^{good}(\tau^{\leq j-1})] \cdot Pr_b[\tau], \end{aligned}$$

which implies that

$$\frac{Pr_b^{bad}(\tau)}{Pr_b[\tau]} \leq \sum_{j=1}^M Pr_{\Omega_b}[\Omega_b^{bad}(\tau^{\leq j}) | \Omega_b^{good}(\tau^{\leq j-1})],$$

i.e.,

$$\frac{Pr_b^{bad}[\tau]}{Pr_b[\tau]} \leq p^* \cdot M, \quad (8)$$

where $p^* = \max\{Pr_{\Omega_b}[\Omega_b^{bad}(\tau \leq j) | \Omega_b^{good}(\tau \leq j-1)], 1 \leq j \leq M\}$.

Let us fix a transcript $\tau \in \mathcal{T}_M$ and some index j , $1 \leq j \leq M$. We now derive an upper bound for p^* , i.e., for the probabilities $Pr[\Omega_b^{bad}(\tau \leq j) | \Omega_b^{good}(\tau \leq j-1)]$, $1 \leq j \leq M$, that an elementary event $\omega \in \Omega_b(\tau)$ becomes bad with the j -th query. Note

Lemma 1. *An elementary event ω belongs to $\Omega_b^{good}(\tau \leq j-1)$ if and only if all of the following conditions are satisfied:*

- (a) $b_\omega = b$.
- (b) $P_\omega|_{\tau_P} = \tau$, i.e., P_ω coincides with the τ -answers on τ_P (in the sense of definition 6).
- (c) $F_\omega|_{\tau_F} = \tau$.
- (d) If $b = 1$, then $E_\omega|_{\tau_E} = \tau$.
- (e) If $b = 0$, then for all $(x, r) \in \tau_E^{\leq j-1}$ it holds that $F_\omega(\pi^r(P_\omega(x, k_\omega))) = \tau(x, r)$.
- (f) $k_\omega \notin \tau^{\leq j-1}$ in the sense of definition 8.
- (g) For all $(x, r) \in \tau_E^{\leq j-1}$ and $y \in \tau_F^{\leq j-1}$, it holds

$$dist_\pi(\pi^r(P_\omega(x, k_\omega)), y) \geq 2(m+n) - 1.$$

- (h) For all $(x, r), (x', r') \in \tau_E^{\leq j-1}$ which fulfill $x \neq x'$, it holds

$$dist_\pi(\pi^r(P_\omega(x, k_\omega)), \pi^{r'}(P_\omega(x', k_\omega))) \geq 2(m+n) - 1.$$

Note that conditions (a)-(e) characterize $\omega \in \Omega_b(\tau)$ and conditions (a)-(h) characterize $\omega \in \Omega_b^{good}(\tau \leq j-1)$. For simplicity, let us denote $\Omega'_b = \Omega_b^{good}(\tau \leq j-1)$ for $b \in \{0, 1\}$.

We have to bound the probability $Pr_{\Omega'_b}(\Omega_b^{bad}(\tau \leq j))$.

Therefore, we will first show that, as a result of our definition of badness, the probability space Ω'_b has a very regular structure. In particular, we will show that all keys k which occur in Ω'_b (more exactly, which occur as component of an elementary event in Ω'_b), occur with the same probability in Ω'_b . Moreover, all pairs (k, P) which occur in Ω'_b (i.e., which occur as components of an elementary event in Ω'_b), occur with the same probability in Ω'_b .

For keys $k \in \{0, 1\}^n$ and permutations P over $\{0, 1\}^{m+n}$ we denote by

$$Pr_{\Omega'_b}[k] = Pr_{\Omega'_b}[k_\omega = k]$$

and

$$Pr_{\Omega'_b}[k, P] = Pr_{\Omega'_b}[k_\omega = k, P_\omega = P]$$

the corresponding probabilities.

Lemma 2. *For all keys $k, k' \in \{0, 1\}^n$ and permutations $P, P' : \{0, 1\}^{m+n} \rightarrow \{0, 1\}^{m+n}$ and all $b \in \{0, 1\}$, holds the following:*

- From $Pr_{\Omega'_b}[k] > 0$ and $Pr_{\Omega'_b}[k'] > 0$ it follows $Pr_{\Omega'_b}[k] = Pr_{\Omega'_b}[k']$.
- From $Pr_{\Omega'_b}[k, P] > 0$ and $Pr_{\Omega'_b}[k', P'] > 0$ it follows

$$Pr_{\Omega'_b}[k, P] = Pr_{\Omega'_b}[k', P'].$$

Proof:

Note first that from $Pr_{\Omega'_b}[k] > 0$ it follows that $k \notin \tau^{\leq j-1}$ (lemma 1, (f)). Let us fix such a key $k \in \{0, 1\}^n$ and let us first consider the case $b = 0$.

We present a randomized algorithm which computes an arbitrary pair P, F such that $(0, k, P, F, E) \in \Omega'_b$. It will become obvious that the number of pairs P, F which can be an output of this algorithm will not depend on k , which proves the first statement of lemma 2 for $b = 0$.

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

$$\rho(x) = \{r, (x, r) \in (\tau^{\leq j-1})_E\}.$$

STEP 1: For all $u \in (\tau^{\leq j-1})_P$ define $P(u) = \tau(u)$ and for all $y \in (\tau^{\leq j-1})_F$ define $F(y) = \tau(y)$.

STEP 2: Note first that, as k satisfies condition (f) of lemma 1, $(x, k) \notin (\tau^{\leq j-1})_P$ for all $x \in X$. We fix an arbitrary order on X and define $P(x, k)$ for all $x \in X$ in this order in such a way that conditions (g) and (h) of lemma 1 are fulfilled. Observe that for all $x \in X$, there can be identified a set $Forbidden(x)$ such that $P(x, k)$ respects conditions (g) and (h) if and only if

$$P(x, k) \notin \bigcup_{r \in \rho(x)} \pi^{-r} \left(Env_{\pi}^{2(m+n)-1}(Forbidden(x)) \right) \cup Im_{before}(P, x), \quad (9)$$

where $Im_{before}(P, x)$ denotes the set of all images of P which were defined before $P(x, k)$ in STEP-1 and STEP-2.

The definition of $Forbidden(x)$ depends on the order in which we go through X when defining $P(x, k)$. For the first x it holds

$$Forbidden(x) = (\tau^{\leq j-1})_F.$$

If $P(x', k)$ is defined directly after $P(x, k)$ then

$$Forbidden(x') = Forbidden(x) \cup \{\pi^r(P(x, k)); r \in \rho(x)\}.$$

We go through X in the fixed order and define for all $x \in X$ the value $P(x, k)$ in such a way that condition (9) is fulfilled and that P remains injective.

Note that $|Im_{before}(P, x)| \leq j - 1$ and that

$$\left| \bigcup_{r \in \rho(x)} \pi^{-r} \left(Env_{\pi}^{2(m+n)-1}(Forbidden(x)) \right) \right| \leq (4(m+n) - 2)(j - 1)^2. \quad (10)$$

STEP 3: For all $u \in \{0, 1\}^{m+n}$ for which $P(u)$ has not been defined in STEP-1 or STEP-2 we define $P(u) \in \{0, 1\}^{m+n}$ in such a way that P remains injective.

STEP 4: We define $F(\pi^r(P(x, k))) = \tau(x, r)$ for all $x \in X$ and $r \in \rho(x)$ (i.e., $(x, r) \in (\tau^{\leq j-1})_E$).

STEP 5: For all $y \in \{0, 1\}^{m+n}$ for which $F(y)$ has not been defined in STEP-1 or STEP-4, we define $F(y) \in \{0, 1\}^{m+n}$ in an arbitrary way that respects the required π -iterativeness of F .

It can be checked easily that $Pr_{\Omega'_0}[k, P, F] > 0$ if and only if P, F can be constructed through STEPs 1-5.

Note that the number of permutations P which can be constructed in STEP-1, STEP-2 and STEP-3 depends on the sizes of the sets $Forbidden(x)$, where $x \in X$. It can be easily seen that these sizes do not depend on the choice of k .

Let us now fix a permutation P obtained after STEP-3. The number of π -iterative functions F for which $Pr_{\Omega'_0}[k, P, F] > 0$ depends on the distances between the inputs for which the F -values were defined in STEP-4. For seeing this, note that, due to the fact that F is π -iterative, the definition of $F(y)$ for some $y \in \{0, 1\}^{m+n}$ determines at least one bit of $F(y')$ for all y' with $dist_\pi(y, y') \leq m + n - 1$. Now observe that P has been chosen in such a way that conditions (g) and (h) of lemma 1 are fulfilled. This implies that for all inputs $y \neq y'$ for which the F -output is defined in STEP-4 it holds that

$$Env_{m+n-1}^\pi(y) \cap Env_{m+n-1}^\pi(y') = \emptyset$$

and

$$Env_{m+n-1}^\pi(y) \cap Env_{m+n-1}^\pi((\tau^{\leq j-1})_F) = \emptyset.$$

This implies that the number of functions F for which $Pr_{\Omega'_0}[k, P, F] > 0$ is the same for all permutations which were constructed according to STEP-1, STEP-2 and STEP-3. This implies the proof of our lemma for $b = 0$.

In the ideal world case $b = 1$, a pair (P, E) for which $Pr_{\Omega'_1}[k, P, E]$ can be computed by a similar algorithm as in the case $b = 0$. We have to replace STEP-4 and STEP-5 by

STEP-4a: For all $x \in X$ and $r \in \rho(x)$ define $E(x, r) = \tau(x, r)$.

STEP-5a: For all $(z, r) \in \{0, 1\}^{m+n} \times \{0, \dots, R-1\}$ for which $E(z, r)$ has not been defined in STEP-4a, we define $E(z, r) \in \{0, 1\}^{m+n}$ in an arbitrary way that respects that E has to be iterative.

The fact that the number of permutations P for which $Pr_{\Omega'_1}[k, P] > 0$ holds does not depend on k can be proved in the same way as in the case $b = 0$. Moreover, for any such pair (k, P) the number of iterative functions E for which $Pr_{\Omega'_1}[k, P, E] > 0$ holds is the same. This follows from the fact that this number only depends on τ . \square

Now we are able to show

Lemma 3. *It holds that for each $b \in \{0, 1\}$*

$$Pr_{\Omega'_b}[\Omega_b^{bad}(\tau^{\leq j})] \leq \min \left\{ \frac{1}{2^n - (j-1)}, \frac{(4(m+n)-2)(j-1)}{2^{m+n} - (4(m+n)-2)(j-1)^2} \right\}.$$

Proof: Let us denote by q the j -th query of transcript τ .

We have to distinguish three cases according to whether q is a P/P^{-1} -query, an E -query or an F -query.

Case 1: The query q is a P -query with input $u = (x, k) \in \{0, 1\}^{m+n}$ or a P^{-1} -query with answer $u = (x, k) \in \{0, 1\}^{m+n}$. Then either $k \in \tau^{\leq j-1}$, i.e., k already occurs as suffix of an element in $(\tau^{\leq j-1})_P$, or not. In the first case, due to lemma 1, it holds that $Pr_{\Omega'_b}[k_\omega = k] = 0$, which implies

$$Pr_{\Omega'_b}[\Omega_b^{bad}(\tau^{\leq j})] = 0.$$

In the second case, from lemma 2 it follows that

$$Pr_{\Omega'_b}[\Omega_b^{bad}(\tau^{\leq j})] \leq \frac{1}{2^n - (j-1)}.$$

This is because all keys which do not occur as suffix of an element in $(\tau^{\leq j-1})_P$ are equally likely w.r.t. Ω'_b (see lemma 2) and because there are at least $2^n - |(\tau^{\leq j-1})_P| \geq 2^n - (j-1)$ such keys.

Case 2: The query q is an E -query for some input (x, r) , where $0 \leq r \leq R-1$ and $x \in \{0, 1\}^m$. As in the proof of lemma 2, we denote by $X = \{x^1, \dots, x^{|X|}\} \subseteq \{0, 1\}^m$ the set of all $x \in \{0, 1\}^m$ which occur as prefix of an input in $(\tau^{\leq j-1})_E$. We have to distinguish two subcases.

Subcase 2a: $x \notin X$. We fix some key k with $Pr_{\Omega'_b}[k] > 0$. For a sequence $\tilde{V} = (v^1, \dots, v^{|X|})$ of elements from $\{0, 1\}^{m+n}$ and a permutation P over $\{0, 1\}^{m+n}$, we denote by $P_\omega(X, k) = \tilde{V}$ the event that $P(x^j, k) = v^j$ for $j = 1, \dots, |X|$.

We denote by \mathcal{V} the set of all sequences \tilde{V} which can be constructed via STEP-2 of the algorithm in the proof of lemma 2, i.e., for which there is some $\omega \in \Omega'_b$ such that $P_\omega(X, k_\omega) = \tilde{V}$. Note that \mathcal{V} does not depend on the choice of k_ω .

We estimate the probability

$$Pr_{\Omega'_b}[\omega \in \Omega_b^{bad}(\tau^{\leq j}) | k_\omega = k, P_\omega(X, k) = \tilde{V}]$$

for an arbitrarily fixed $\tilde{V} \in \mathcal{V}$.

We know that $(x, k) \notin (\tau^{\leq j-1})_P$, otherwise $Pr_{\Omega'_b}[k] = 0$. Consequently, $P_\omega(x, k)$ can take any value in $\{0, 1\}^{m+n}$ outside of $(\tau^{\leq j-1})_{P^{-1}}$ and \tilde{V} with the same probability (see lemma 2).

It holds that $\omega \in \Omega_b^{bad}(\tau^{\leq j})$ if and only if

$$P_\omega(x, k) \in \pi^{-r} \left(Env_\pi^{2(m+n)-1}(\text{Forbidden}(x)) \right),$$

where the set $\text{Forbidden}(x) \subseteq \{0, 1\}^{m+n}$ depends on $(\tau^{\leq j-1})_F$ and \tilde{V} and is defined as in STEP-2 of the algorithm in the proof of lemma 2. Note that $|\text{Forbidden}(x)| \leq j-1$ and consequently

$$|Env_\pi^{2(m+n)-1}(\pi^{-r}(\text{Forbidden}(x)))| \leq (4(m+n)-2)(j-1).$$

We obtain

$$Pr_{\Omega'_b}[\omega \in \Omega_b^{bad}(\tau^{\leq j}) | k_\omega = k, P_\omega(X, k) = \tilde{V}] \leq \frac{(4(m+n)-2)(j-1)}{2^{m+n} - (j-1)}$$

for all keys k with $Pr_{\Omega'_b}[k] > 0$ and $\tilde{V} \in \mathcal{V}$, which implies that

$$Pr_{\Omega'_b}[\Omega_b^{bad}(\tau \leq j)] \leq \frac{(4(m+n)-2)(j-1)}{2^{m+n} - (j-1)}.$$

Subcase 2b $x \in X$. Note that $r \notin \rho(x)$, otherwise the same query q would have been posed already during $\tau \leq j-1$. We suppose w.l.o.g. that $x = x^{|X|}$ and denote $X' = X \setminus \{x^{|X|}\}$. Moreover, we denote by \mathcal{V}' the set of all sequences $\tilde{V}' = (v^1, \dots, v^{|X|-1})$ of elements from $\{0, 1\}^{m+n}$ for which there is some $\omega \in \Omega'_b$ such that $P_\omega(X', k_\omega) = \tilde{V}'$. Note again that \mathcal{V}' does not depend on the choice of k_ω (see subcase 2a).

We estimate the probability

$$Pr_{\Omega'_b}[\omega \in \Omega_b^{bad}(\tau \leq j) | k_\omega = k, P_\omega(X', k) = \tilde{V}']$$

for an arbitrarily fixed key k with $Pr_{\Omega'_b}[k] > 0$ and an arbitrarily fixed $\tilde{V}' \in \mathcal{V}'$.

We know that for all $\omega \in \Omega'_b$ which fulfill $k_\omega = k$ and $P_\omega(X', k) = \tilde{V}'$ it holds that

$$P_\omega(x, k) \notin \bigcup_{r' \in \rho(x)} \pi^{-r'} \left(Env_\pi^{2(m+n)-1} (Forbidden(x)) \right)$$

and that the size of the set at the right hand side is not greater than $(4(m+n)-2)(j-1)^2$ (see (10)).

An element $\omega \in \Omega'_b$ fulfilling $k_\omega = k$ and $P_\omega(X', k) = \tilde{V}'$ belongs to $\Omega_b^{bad}(\tau \leq j)$ if and only if

$$P_\omega(x, k) \in \pi^{-r} \left(Env_\pi^{2(m+n)-1} (Forbidden(x)) \right),$$

where the size of the set at the right hand side is not greater than $(4(m+n)-2)(j-1)$. Consequently,

$$Pr_{\Omega'_b}[\omega \in \Omega_b^{bad}(\tau \leq j)] \leq \frac{(4(m+n)-2)(j-1)}{2^{m+n} - (4(m+n)-2)(j-1)^2}.$$

Case 3 q is an F -query for some input $y \notin (\tau \leq j-1)_F$. For all $\omega \in \Omega'_b$ it holds that $\omega \in \Omega_b^{bad}(\tau \leq j)$ if and only if there is some $(x, r) \in (\tau \leq j-1)_E$ (i.e., $x \in X$ and some $r \in \rho(x)$) such that

$$dist_\pi(y, \pi^r(P_\omega(x, k_\omega))) \leq 2(m+n) - 1,$$

which is equivalent to the event

$$P_\omega(x, k_\omega) \in \pi^{-r} \left(Env_\pi^{2(m+n)-1}(y) \right).$$

By exactly the same arguments as used in subcase 2b, the probability of this event can be shown to be not greater than

$$\frac{4(m+n) - 2}{2^{m+n} - (4(m+n) - 2)(j-1)^2}.$$

Consequently, also in this case, we have

$$Pr_{\Omega'_b}[\omega \in \Omega_b^{bad}(\tau^{\leq j})] \leq \frac{(4(m+n)-2)(j-1)}{2^{m+n} - (4(m+n)-2)(j-1)^2}. \quad \square$$

Looking at relation (8) we obtain

Corollary 1. *For both $b \in \{0, 1\}$ and arbitrary transcripts $\tau \in \mathcal{T}_M$, it holds that*

$$\frac{Pr_b^{bad}[\tau]}{Pr_b[\tau]} \leq \min \left\{ \frac{M}{2^n - M}, \frac{(4(m+n)-2)M^2}{2^{m+n} - (4(m+n)-2)M^2} \right\}.$$

This proves result (1) with $\epsilon = \min \left\{ \frac{M}{2^n - M}, \frac{((4(m+n)-2)M^2)}{2^{m+n} - (4(m+n)-2)M^2} \right\}$.

For proving theorem 2, it is now sufficient to prove result (2), i.e. that

$$Pr_{\Omega_0}[\Omega^{good}(\tau)] = Pr_{\Omega_1}[\Omega^{good}(\tau)] \quad (11)$$

for all $\tau \in \mathcal{T}^M$.

Let us fix some transcript $\tau \in \mathcal{T}^M$. Remember that an elementary event $\omega = (b_\omega, k_\omega, P_\omega, F_\omega, E_\omega)$ belongs to $\Omega^{good}(\tau)$, if all of the following conditions hold:

- (1) P_ω and F_ω are consistent with the answers of all P -, P^{-1} - and F -queries belonging to τ .
- (2) The key k_ω does not occur in τ .
- (3) k_ω and P_ω are chosen in such a way that no near structural EE - or EF -collisions occur in τ .
- (4) If $b_\omega = 0$, then F_ω fulfills

$$F_\omega(\pi^r(P_\omega(x, k_\omega))) = E(x, r)$$

for all inputs (x, r) of an E -query belonging to τ .

- (5) If $b_\omega = 1$, then E_ω fulfills

$$E_\omega(x, r) = E(x, r)$$

for all inputs (x, r) of an E -query belonging to τ .

Consequently, we have to show that for all 4-tuples $(k_\omega, P_\omega, F_\omega, E_\omega)$ which fulfill conditions (1),(2),(3), the probability p_0 that F_ω satisfies (4) equals the probability p_1 that E_ω satisfies condition (5).

This can be derived quite straightforwardly by looking at the relation between random π -iterative functions and random bitstreams defined over the π -components of $\{0, 1\}^{m+n}$, as it was described in subsection 2.2. In the real-world case, for all $x \in \{0, 1\}^m$ the answers of the E -queries with inputs (x, r) , $r \in \rho(x)$, are subsets of bits of a random bitstream defined w.r.t. to the π -component of $P_\omega(x, k_\omega)$. As we do not have near EE -collisions, for all pairs of different $x, x' \in \{0, 1\}^m$, these subsets do not overlap. Moreover, as we do not have near

EF-collisions, they do not overlap with subsets of this random bitstream which correspond to F -queries belonging to τ . Consequently, for all $x \in \{0, 1\}^m$, the answers of the E -queries with inputs (x, r) , $r \in \rho(x)$, can be thought of as being determined by mutually independent truly random bitstreams b^x . But this corresponds exactly to the ideal-world scenario determining the probability p_1 , which implies that $p_0 = p_1$. \square

5 Conclusion

With this paper, we introduced for the first time an ideal primitive model for KSG-based stream ciphers and showed that the small-key-large-state principle (used, e.g., by the eSTREAM finalists Trivium and Grain) yields asymptotically the maximal possible security of $\min\{n, L/2\}$ against generic chosen-IV TMD-tradeoff attacks. This implies that, from now on, designers of stream ciphers can use this construction principle unscrupulously. We hope that our security model will be used in the future for analyzing other types of state initialization- and keystream generation algorithms of specific stream ciphers. Note that very recently in [9] Hoang and Tessaro presented the ϵ -proximity method for proving information-theoretic security lower bounds w.r.t. to distinguishing attacks. It would be interesting to check if this method allows to simplify our lower bound proof.

References

1. S.H. Babbage. Improved "exhaustive search" attacks on stream ciphers. In *Security and Detection, 1995.*, *European Convention on*, pages 161–166, May 1995.
2. Alex Biryukov and Adi Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2000.
3. Christophe De Cannière and Bart Preneel. Trivium - specifications (eSTREAM). Technical report, ECRYPT (European Network of Excellence for Cryptology), 2005. http://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf.
4. Shan Chen and John Steinberger. Tight security bounds for key-alternating ciphers. In PhongQ. 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.
5. Orr Dunkelman and Nathan Keller. Treatment of the initial value in time-memory-data tradeoff attacks on stream ciphers. *Cryptology ePrint Archive*, Report 2008/311, 2008. <http://eprint.iacr.org/2008/311>.
6. Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudo-random permutation. In Hideki Imai, RonaldL. Rivest, and Tsutomu Matsumoto, editors, *Advances in Cryptology — ASIACRYPT '91*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer Berlin Heidelberg, 1993.
7. 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.

8. Martin Hell, Thomas Johansson, and Willi Meier. Grain - a stream cipher for constrained environments (eSTREAM). Technical report, ECRYPT (European Network of Excellence for Cryptology), 2005. http://www.ecrypt.eu.org/stream/p3ciphers/grain/Grain_p3.pdf.
9. Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. Cryptology ePrint Archive, Report 2016/578, 2016. <http://eprint.iacr.org/2016/578>.
10. 3GPP Organizational Partners. 3GPP TS 55.216 V6.2.0 (2003-09), 2003. <http://www.gsma.com/technicalprojects/wp-content/uploads/2012/04/a53andgea3specifications.pdf>.
11. 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.
12. Bluetooth SIG. Bluetooth core specification 4.2, 2014. https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=286439.