

Farasha: A Provable Permutation-based Parallelizable PRF

Najwa Aaraj¹, Emanuele Bellini¹, Ravindra Jejurikar¹, Marc Manzano^{2,3},
Raghvendra Rohit¹, and Eugenio Salazar¹

¹ Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE
`firstname.lastname@tii.ae`

² SandboxAQ, Palo Alto, CA, United States `marc@sandboxaq.com`

³ Electronics and Computing Department, Faculty of Engineering, Mondragon
Unibertsitatea, Mondragon, Spain

Abstract. The pseudorandom function *Farfalle*, proposed by Bertoni *et al.* at ToSC 2017, is a permutation based arbitrary length input and output PRF. At its core are the public permutations and feedback shift register based rolling functions. Being an elegant and parallelizable design, it is surprising that the security of *Farfalle* has been only investigated against generic cryptanalysis techniques such as differential/linear and algebraic attacks and nothing concrete about its provable security is known. To fill this gap, in this work, we propose *Farasha*, a new permutation-based parallelizable PRF with provable security. *Farasha* can be seen as a simple and provable *Farfalle*-like construction where the rolling functions in the compression and expansion phases of *Farfalle* are replaced by a uniform almost xor universal (AXU) and a simple counter, respectively. We then prove that in the random permutation model, the compression phase of *Farasha* can be shown to be a uniform AXU function and the expansion phase can be mapped to an Even-Mansour block cipher. Consequently, combining these two properties, we show that *Farasha* achieves a security of $\min\{\text{keysize}, \text{permutation size}/2\}$. Finally, we provide concrete instantiations of *Farasha* with AXU functions providing different performance trade-offs. We believe our work will bring new insights in further understanding the provable security of *Farfalle*-like constructions.

Keywords: Pseudo random function · *Farfalle* · Almost xor universal function

1 Introduction

Designing a cryptographic primitive which is parallelizable and at the same time offers provable security bounds requires a holistic approach. The simplest example of such a design is Parallelizable Message Authentication Code (PMAC), proposed by Black and Rogaway, which is based on the Hash-then-PRF design paradigm [14]. The same design principle with variations is later adopted in authenticated encryption (AE) modes based on (tweakable) block ciphers. Some of

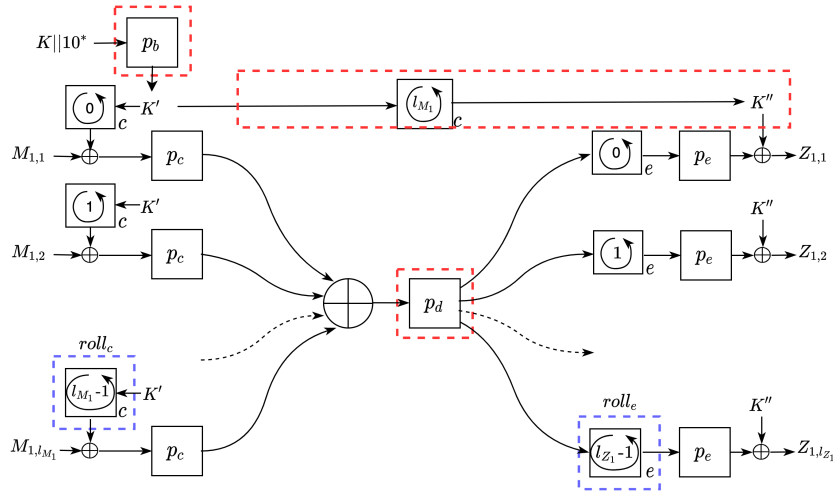
the examples include OCB [38], GCM [33,26], SIV [39], OCB-3 and Θ CB-3 [32], OTR [34], Deoxys-II [31] and SCT [37], to name a few.

Although parallelizable and provable, the aforesaid AE modes based on (tweakable) block ciphers employ a (tweakable) secret permutation as the core primitive. In order to have a mode which depends on an unkeyed cryptographic permutation rather than a (tweakable) secret permutation, Bertoni *et al.* introduced sponge functions [10].

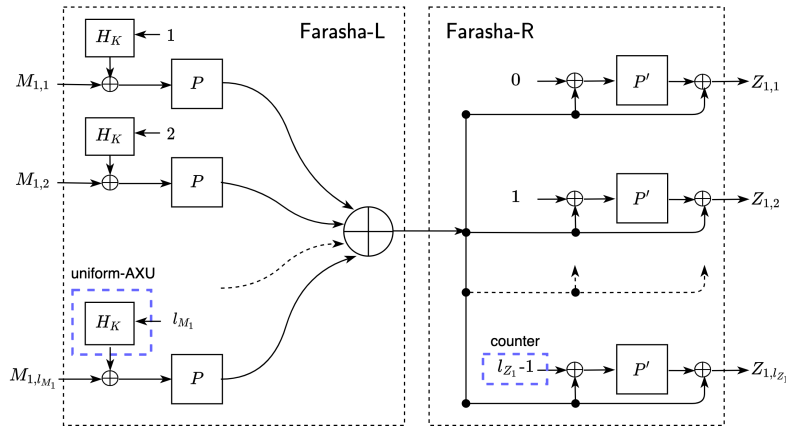
Initially, the sponge construction was proposed for the Keccak hash function [11], but later gained popularity because of its versatility in providing multiple cryptographic functionalities such as hash, authenticated encryption with associated data (AEAD), MAC or pseudo random number generator (PRNG) [8,9,12] using a single permutation. As such, researchers focused on designing unkeyed cryptographically secure permutations (the so-called permutation-based crypto) rather than designing an individual mode. Numerous cryptographic primitives based on sponge construction have thus been proposed, with their security thoroughly analyzed. Examples of hash functions include Photon [30], Spongint [15] and Quark [1], while Ascon [23], Norx [2], Ketje [6], Gimli [5], Subterranean-AE [22], Xoodyak [20] and Elephant [13] are few examples of AEAD schemes.

Albeit versatile, the sponge-based primitives are inherently sequential and therefore not able to exploit available parallelism on high-end CPUs. At ToSC 2017, Bertoni *et al.* [7] proposed *Farfalle*, a parallel counterpart of sponge functions. It is an arbitrary length input and output pseudo random function (also referred to as a deck function [19]) by design and modes have been built on top of it. [Figure 1a](#) shows the high level description of *Farfalle* that uses public permutations denoted as p_b, p_c, p_d and p_e for the different stages of the construction. It takes a secret key K and a sequence of data blocks $M_{1,1}, \dots, M_{1,\ell_{M_1}}$ as inputs (corresponding to message M_1) and outputs a sequence of keystream blocks $Z_{1,1}, \dots, Z_{1,\ell_{Z_1}}$. The feedback shift register based rolling functions $roll_c$ and $roll_e$ (denoted by a circular arc within a square in [Figure 1a](#)) are used for parallelism in the compression and expansion phases. An output block is computed by masking the permutation p_e output with a rolled key K^{roll} .

While the designers of *Farfalle* provided its security analysis based on unpredictability of state values and generating affine subspaces at the input of p_c , periodicity of rolling functions, differential/linear cryptanalysis and meet-in-the-middle attacks, a security proof for the construction is missing. Very recently, Dobraunig *et al.* [24] proposed a construction resembling *Farfalle*, but again it lacks a security proof. Thus, it is worth questioning whether *Farfalle* can be modified to a new construction which is provable and can simultaneously achieve all the benefits of *Farfalle*. In this work, we confirm the feasibility of the latter question through our proposed construction *Farasha*. In what follows, we first briefly describe our design approach in moving from *Farfalle* to *Farasha* and we then list our contributions.



(a) Farfalle (Figure adapted from [7])



(b) Farasha

Fig. 1: A high level overview of Farfalle and Farasha

1.1 Design Rationale of Farasha

In Figure 1a, a simple but careful observation depicts that Farfalle can alternatively be viewed as a composition of a parallel keyed hash construction (up to the input of p_d) and a parallel PRF construction (from p_d to the PRF output), which is similar to the well-known Hash-then-PRF composition for constructing a PRF. We focus on similar building blocks to construct a permutation-based parallelizable PRF with provable security. Our main idea is to extend the well-known parallelizable keyed hash algorithm [14] to a public permutation and then combine it with the permutation based CTR-mode PRF [3,4].

1.2 Our Contributions

We propose **Farasha**⁴, a permutation-based variable length input and output PRF which is parallelizable, provably secure and whose design is motivated by **Farfalle**'s missing security proof. We emphasize that the goal here is to analyze the provable security and not to compare the performance between **Farasha** and **Farfalle**. Our contributions are summarized as follows.

1. **DESIGN OF FARASHA**: Our construction as shown in [Figure 1b](#) is composed of two layers: a compression layer followed by an expansion layer similar to **Farfalle** ([Figure 1a](#)). To achieve a design with provable security bounds, we incorporate the following changes to **Farfalle**: 1) the linear rolling function in the compression layer is replaced by a uniform AXU function; 2) the nonlinear feedback shift register based rolling function in the expansion layer is changed to a counter mode Even-Mansour construction; and 3) the intermediate permutations p_b and p_d of **Farfalle** are removed. The changes are highlighted in [Figure 1](#) with a blue dotted box denoting a modification and a red one for removal. We also provide an instance of **Farasha**, named **Farasha-wLFSR** where the uniform AXU function is a word based LFSR. It is worth noting that in **Farasha**, all inputs to permutations P and P^0 can be computed in parallel unlike **Farfalle** where there is a dependency among input states (of p_e) because of the nonlinear rolling function.
2. **SECURITY ANALYSIS**: We provide a detailed formal security analysis of **Farasha** in the indistinguishability framework in a random permutation model. First, we show that the compression phase of **Farasha** is a ϵ -uniform-AXU function (for some $\epsilon > 0$) and the expansion phase can be mapped to multi-key Even-Mansour block cipher. We then show that **Farasha** achieves a security of minimum of keysize or half the permutation size. In the end, we give insights on the security of **Farfalle**.

1.3 Outline of the Paper

The rest of the paper is organized as follows. In [Section 2](#), we define our notation and give a brief overview of the security model and keyed hash functions. [Section 3](#) presents the design of **Farasha** along with its salient features. We provide the security analysis of **Farasha** in [Section 4](#). In [Section 5](#), we provide a discussion on improved security of **Farasha**, the choice of AXU functions and their performance trade-offs, and insights on the security of **Farfalle**. Finally, we conclude the paper in [Section 6](#).

2 Preliminaries

In this section, we describe the notation used throughout the paper, our security model, and some well-known constructions which are relevant to this work.

⁴ **Farasha** means butterfly in Arabic

2.1 Notation and Security Model

Fix $n, m \geq \mathbb{N}$. We use $\{0, 1\}^n$ and $\{0, 1\}^*$ to denote the set of all bit strings of length n and variable length bit strings, respectively. For any $X \in \{0, 1\}^*$, $|X|$ denotes the length of X in bits. The size of a set S is also denoted by $|S|$ if the meaning is clear from the context. We use X_1, \dots, X_u to denote the n -bit block partitioning of X where $|X_i| = n$ for $1 \leq i \leq u-1$ and $|X_u| \leq n$. Also, $\text{pad}_n(X)$ denotes the bit string obtained by appending X with 1, followed by 0's, so that its length becomes the nearest multiple of n . For $X, Y \in \{0, 1\}^*$, $X \oplus Y$ and $X \parallel Y$ denote the bitwise XOR and concatenation operators, respectively. By $X \leftarrow \{0, 1\}^n$, we mean X is picked uniformly at random from the set $\{0, 1\}^n$. Further, $\text{Perm}(n)$ denotes the set of all n -bit permutations while $\text{Func}(n, m)$ refers to the set of all n -bit to m -bit functions. For $n = m$, we write $\text{Func}(n, n)$ as $\text{Func}(n)$. We define $\text{msb}_i(X)$ as the leftmost i bits of a string X . Finally, $\Pr[X = x]$ denotes the probability that a random variable X takes value x .

For our security analysis, we consider an adversary A which is an algorithm that is given access to one or more oracles O . After interacting with O , it outputs a bit $w \in \{0, 1\}$. We denote this event by $A^O \leftarrow w$. We always consider computationally unbounded adversaries, i.e., their computational time is always measured in terms of number of oracle queries (say q). For any two oracles O and P , the advantage of distinguishing O from P is then defined as

$$\text{Adv}_O^P(A, q) := \left| \Pr[A^O \leftarrow 1] - \Pr[A^P \leftarrow 1] \right|. \quad (1)$$

2.2 Keyed Hash Functions

Let $k, t \geq \mathbb{N}$ and $\epsilon > 0$. A keyed hash function H is a deterministic algorithm from $\{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^t$ with $H(K, M) = T$. For a fixed key K , we denote it by H_K . We now state some relevant definitions related to the keyed hash functions (adapted from [16][Section 7.1 and 7.2]).

Definition 1. [Almost XOR Universal Hash Function (AXU)] Let $\Delta \in \{0, 1\}^t$. We say H_K is ϵ -AXU if for any two distinct messages M and M^θ , we have

$$\Pr[K \leftarrow \{0, 1\}^k \mid H_K(M) \oplus H_K(M^\theta) = \Delta] \leq \epsilon \quad (2)$$

Definition 2. [Uniform Hash Function] Let $\Delta \in \{0, 1\}^t$. We say H_K is ϵ -uniform if for any message M , the following holds:

$$\Pr[K \leftarrow \{0, 1\}^k \mid H_K(M) = \Delta] \leq \epsilon \quad (3)$$

Definition 3. [Uniform AXU Function] We say H_K is ϵ -uniform-AXU if H_K is both ϵ -uniform and ϵ -AXU.

Note that in the above definitions, the number of queries is not present. Accordingly, to formulate the security of H_K (which usually relies on the hash output being secret) against adversaries with q queries, we define the distinguishing game setup for each of these definition and give its adversarial advantage.

Adversarial Setup for H_K as an AXU. Here the goal of the adversary is to find the Xor-difference of the hash of two messages. For the security of the secret key, hash values are always kept secret (typically via encryption) and this process is often referred to as a blinding operation on the hash. The blinding operation is modelled with the help of a random oracle: a function that returns an independent random value for every new input.

For a query of the form (M, Δ) , the responses of the oracles are as follows.

- Real world \mathcal{O} : Here, the key $K \leftarrow \mathcal{K}$ is sampled along with a secret random oracle \mathcal{RO}_1 . The challenger returns $\mathcal{RO}_1(H_K(M) \oplus \Delta)$.
- Ideal world \mathcal{P} : $\mathcal{RO}_2(M, \Delta)$ with a secret random oracle \mathcal{RO}_2 .

For an adversary A making q queries, the advantage is given by

$$\text{Adv}_H^{\text{axu}}(A, q) := \left| \Pr[A^{\mathcal{O}} \neq 1] - \Pr[A^{\mathcal{P}} \neq 1] \right| \quad (4)$$

Note that Equation 4 is identical to the *blinded key hash (bkh)* model defined in [29].

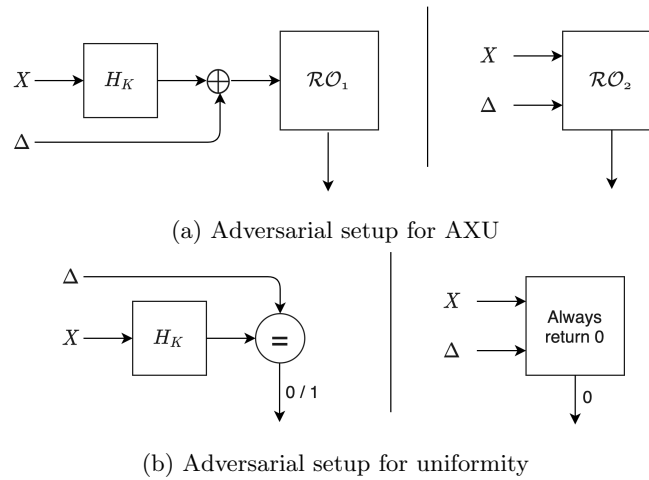


Fig. 2: Adversarial setups for keyed-hash functions

Adversarial Setup for H_K as Uniform Function. Here the adversary's goal is to find a message along with its hash-value and a query is of the form (X, Δ) . The real world oracle returns whether $H_K(X) = \Delta$ is true or false. The ideal world always returns false as shown in Figure 2b. The advantage is defined as,

$$\text{Adv}_H^{\text{uni}}(A, q) := \left| \Pr[K \leftarrow \mathcal{K} \wedge \exists M_i \text{ s.t. } H_K(M_i) = \Delta] \right| \quad (5)$$

Adversarial Setup for H_K as a Uniform-AXU. Here the adversary is given access to both oracles, that of an AXU function and a uniform function, and the goal is to distinguish the real world from the ideal world.

2.3 Even-Mansour Block Cipher

The Even-Mansour (EM) construction [27], builds a b -bit block cipher from a b -bit public permutation P . The original construction (dual key Even-Mansour), based on a pair of b -bit keys K_1 and K_2 , is defined as $E_{K_1, K_2}^P(M) = P(M \oplus K_1) \oplus K_2$ where M is a b -bit message. Dunkelman *et al.* [25] showed that the original EM scheme is not minimal in terms of key size and they presented the Single-key Even-Mansour (SEM) block cipher with the same security level. The single key Even-Mansour (SEM) is given by $E_K^P(M) = P(M \oplus K) \oplus K$.

SEM in Multi-Key Setting with Independent Keys. The security bounds of SEM in the multi-key setting is given by [Theorem 1](#).

Theorem 1 (Security of Multi-key EM [35]). *In a multi-key setting with μ EM block-cipher instances with μ independent keys, the distinguishing advantage of an adversary is bounded by*

$$\text{Adv}_{EM}^{\text{prp}}(A, \sigma, \mu) \leq \frac{\sigma^2}{2^b} + \frac{2\sigma q_p}{2^b} \quad (6)$$

where σ is the total number of construction queries (across all μ instances) and q_p is the number of primitive queries to P .

SEM in the Modified Key Setting. In contrast to μ independent keys as in [Theorem 1](#), we consider keys which are the output of an ϵ -uniform-AXU keyed hash function H_K , where $K \in \{0, 1\}^k$. [Figure 3](#) depicts this setting and we provide a security proof of SEM with these multiple keys in [Theorem 2](#).

Theorem 2 (Security of SEM in the modified key setting). *Let $\epsilon, k > 0$, $K \in \{0, 1\}^k$ and $H_K : \{0, 1\}^k \rightarrow \{0, 1\}^k$ be a ϵ -uniform-AXU. Consider μ SEM instances with keys $K_i = H_K(M_i)$ for $i = 1, \dots, \mu$. Then, in a multi-key setting, the distinguishing advantage of an adversary is bounded by*

$$\text{Adv}_{EM}^{\text{prp}}(A, \sigma, \mu) \leq \sigma^2 \epsilon + 2\sigma q_p \epsilon \quad (7)$$

where σ is the total number of construction queries (across all μ instances) and q_p is the number of primitive queries to P .

Proof. The proof is analogous to the proof of [Theorem 1](#) and details are provided in [Appendix A.3](#) for completeness.

Corollary 1. *[Theorem 2](#) can be trivially extended to the dual-key variant of EM cipher.*

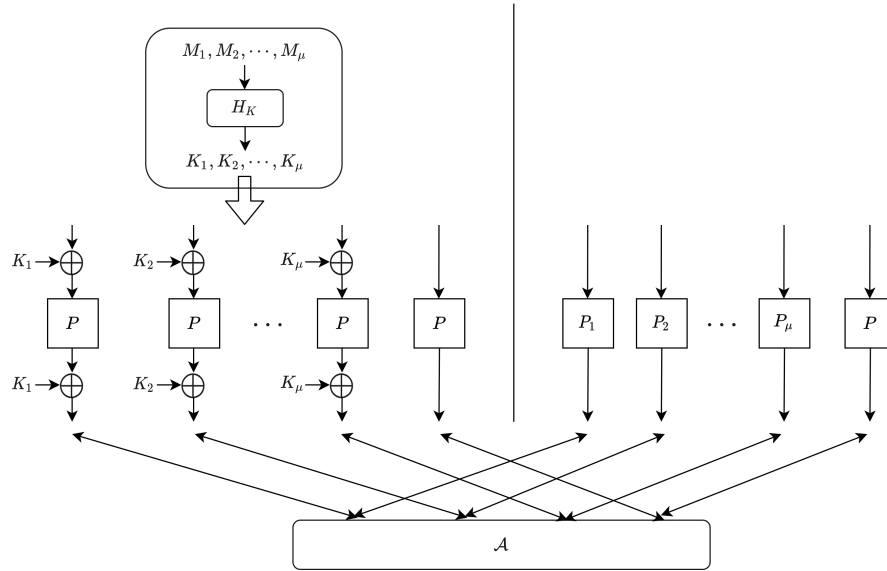


Fig. 3: Multi-key SEM game setup in the modified key setting

2.4 PRP-PRF Switching

To bound the advantage of an multiple EM block cipher as a PRF, we extend the PRP-PRF switching lemma [18] to the multi-key setting as follows.

Lemma 1. *The advantage of switching μ independent PRPs to PRFs is bounded by*

$$\text{Adv}_{prp}^{\text{prf}}(A, \sigma, \mu) \leq \frac{\sigma^2}{2} \frac{1}{2^b} \quad (8)$$

where σ is the total number of queries to μ PRP instances.

3 Farasha Specification and Features

In this section, we formally introduce **Farasha** - a PRF with provable security bounds. We describe its building blocks and one of the instance. We also highlight the salient features of **Farasha** by comparing it with **Farfalle**.

3.1 The Farasha PRF

Farasha is a variable length input and variable length output PRF. As shown in [Figure 1](#) (dashed boxes), **Farasha** consists of two layers⁵, namely **Farasha-L** and **Farasha-R**. The core components of these layers include (1) public permutations

⁵ **Farasha** means butterfly in Arabic. Here **Farasha-L** and **Farasha-R** correspond to the left and right wing of a butterfly, respectively.

P and P^θ which operate on a b -bit state; and (2) a fixed input-size keyed hash function H that takes as input a counter and a secret key K of length k satisfying $k \geq b$ and outputs a b -bit message digest. We now explain Farasha-L and Farasha-R in detail.

3.1.1 Description of Farasha-L In a nutshell, Farasha-L is a compression function similar to the compression layer of Farfalle. We simply abstract the linear rolling function of Farfalle (see Figure 1a) to a keyed hash function which is a uniform-AXU. Let $M_i \in \{0, 1\}^*$ be the i -th input message. Let $\ell_{M_i} \geq 1$ such that $(M_{i,1}, \dots, M_{i,\ell_{M_i}}) \stackrel{b}{\leftarrow} \text{pad}_b(M_i)$. For a fixed key K , denote the keyed hash function by $H_K(\cdot)$. Then Farasha-L computes a b -bit secret state X_i by first applying the function $P \circ H_K$ to ℓ_{M_i} blocks of message M_i in parallel and then computing the XORed value of outputs. Formally, we have $X_i = \bigoplus_{j=1}^{\ell_{M_i}} P(H_K(j) \oplus M_{i,j})$. An algorithmic description of Farasha-L is provided in Algorithm 1.

Algorithm 1: Farasha-L

Input: k -bit secret key K with $k \leq b$; message M_i with $(M_{i,1}, \dots, M_{i,\ell_{M_i}}) \stackrel{b}{\leftarrow} \text{pad}_b(M_i)$
Output: $X_i \in \{0, 1\}^b$
1 $X_i \leftarrow 0^b$
2 **for** $j = 1$ **to** ℓ_{M_i} **do**
3 $X_i \leftarrow X_i \oplus P(H_K(j) \oplus M_{i,j})$
4 **end**
5 **return** X_i

3.1.2 Description of Farasha-R Farasha-R is a fixed input and variable length output PRF analogous to Farfalle (see Figure 1a). The output of Farasha-L is used as the key for Farasha-R, such that each Farasha-R invocation has a different key (using a different input message). We modify the expansion layer of Farfalle to the counter-mode based Even-Mansour-like construction as follows. Let X_i be the output of Farasha-L corresponding to the message M_i . Further, let N be the required number of PRF output bits with $\ell_{Z_i} = \lceil \frac{N}{b} \rceil$. For $1 \leq j \leq \ell_{Z_i}$, we compute the j -th output block as $Z_{i,j} = P^\theta(X_i \oplus (j-1))$. Note that for the last output block, we do the truncation if N is not a multiple of b . The entire procedure is illustrated in Algorithm 2.

3.2 Farasha-wLFSR: An Instance of Farasha

We present an instances of Farasha, where H_K is an LFSR. We call this instance as Farasha-wLFSR. For the concrete LFSR, we consider word-based ones, using the method described in [28]. Consider an LFSR with state size of 256 bits

Algorithm 2: Farasha-R

Input: b -bit secret state X_i
Output: N bits of PRF with $\ell_{Z_i} = \lceil \frac{N}{b} \rceil$

- 1 $Z_i \leftarrow \varepsilon$
- 2 **for** $j = 1$ to ℓ_{Z_i} **do**
- 3 $Z_{i,j} \leftarrow X_i \oplus P^\theta(X_i \oplus (j - 1))$
- 4 $Z_i \leftarrow Z_i || Z_{i,j}$
- 5 **end**
- 6 **return** $\text{msb}_N(Z_i)$

arranged into 4 64-bit words, initialised with as $S_0 = K$. If $S_j = (x_0, x_1, x_2, x_3)$ is the state at time j , then the next state S_{j+1} is computed as

$$S_{j+1} = (x_1, x_2, x_3, (x_0 \lll 3) \oplus (x_3 \oplus 5)). \quad (9)$$

Note that this LFSR initialized with $S_0 = K$, is shown to be a uniform AXU [28,17].

3.3 Salient Features of Farasha

Our main goal while designing Farasha is to have a parallel permutation-based PRF construction with provable security. In addition to achieving this goal, Farasha, by design, incorporates the following salient features.

Simple and Versatile Design. Farasha is a simple construction with fewer building blocks compared to Farfalle. The utilized uniform AXUs in Farasha-L could be versatile and platform-specific rather than a specific LFSR based rolling function in Farfalle. The permutation p_d and function $roll_e$ are required in the Farfalle expansion phase to augment the non-linearity provided by a limited rounds permutation p_e (e.g. Xoodoo[6] in Xoofff). With p_e replaced by a random permutation, these two components can be replaced with a simple counter as in a CTR-mode PRF design.

Independent Input States in Farasha-R. An LFSR based compression phases in both Farasha and Farfalle can be mapped to matrix multiplication operations, enabling independent computations for all input blocks to permutation P . The same holds true for inputs to P^θ in the expansion phase of Farasha, which uses a simple counter addition. On the other hand, Farfalle uses a non-linear function $roll_e$, which introduces a dependency amongst the inputs in the expansion phase. The independence of states in the expansion phase gives an additional advantage for Farasha in terms of implementation (for example, multi-CPU implementation).

Small State Size. Farfalle requires $3b$ bits of state registers in both the compression and expansion phases. For Farasha, the state size required for the two phases is $k + 2b$ and $2b$, respectively. Overall, the state size of Farasha is smaller than Farfalle.

4 Security Analysis of Farasha

In this section, we discuss the security of Farasha in a single-user setting and show that it is birthday-bound secure. We start this section by explaining our adversarial setup. Next, we state our main result in [Theorem 3](#) and provide its security proof. Before proceeding to the proofs, we recall some notations which will be used throughout this section.

Notation. Let $\epsilon > 0$, $t \geq 0$ and $k, b \geq \mathbb{N}$ with $k \leq b$. Fix $K = \{0, 1\}^k$ and $Y = \{0, 1\}^b$. Let $H : K \times \mathbb{N} \rightarrow Y$ be a ϵ -uniform-AXU. Let $K = \{K, P, P^\theta\} \subseteq \text{Perm}(b)$, and Rand be a function that for each input $M_i \in \{0, 1\}^*$ returns a random string $Z_i \in \{0, 1\}^*$. Let $M_i = M_{i,1} \parallel \dots \parallel M_{i,\ell_{M_i}}$ consists of $\ell_{M_i} = \lceil \frac{|M_i|}{b} \rceil$ b -bit blocks. We pad the last message block if it is not a multiple of b . Similarly, we have $Z_i = Z_{i,1} \parallel \dots \parallel Z_{i,\ell_{Z_i}}$ for some $\ell_{Z_i} \geq 1$. Note that $Z_{i,\ell_{Z_i}}$ may not be a full block, however, for the simplicity of analysis we take $|Z_{i,\ell_{Z_i}}| = b$. Furthermore, we write Farasha-L and Farasha-R as F_L and F_R , respectively.

For $M_i = M_{i,1} \parallel \dots \parallel M_{i,\ell_{M_i}}$, let $(j, M_{i,j})$ denote the tuple corresponding to the j -th message block $M_{i,j}$ of M_i . Now, for $M_i \neq M_i^\theta$, we say two tuples are identical if both $j = j^\theta$ and $M_{i,j} = M_{i^\theta,j^\theta}$. In our proof, we denote σ as the total number of distinct tuples $(j, M_{i,j})$ within the q messages M_1, \dots, M_q .⁶

4.1 Adversarial Setup

Consider a deterministic and computationally unbounded adversary A which tries to distinguish $O := (\text{Farasha}_K^{H,P,P^\theta}, P, P^\theta)$ (real world) from $P := (\text{Rand}, P, P^\theta)$ (ideal world). The notation P^\perp denotes that both P and P^\perp queries can be made. The advantage of adversary A for the PRF security of Farasha is defined as

$$\text{Adv}_{\text{Farasha}}^{\text{prf}}(A) := \left| \Pr \left[K \xleftarrow{\$} K : A^{\text{Farasha}_{K,P,P^\theta}} \not\approx 1 \right] - \Pr \left[A^{\text{Rand},P,P^\theta} \not\approx 1 \right] \right|. \quad (10)$$

To compute an upper bound on the adversarial advantage in [Equation 10](#) (later given in [Theorem 3](#)), we first define the data (inputs, outputs) available to an adversary after its interaction with oracle O or P . The input-output pairs are collected in a transcript τ . We say τ is attainable, if it can be obtained with a non-zero probability in the ideal world.⁷

Transcripts from construction queries. Let the adversary A makes q construction queries to O or P , in the forward direction only. We summarize the inputs and outputs of these queries in the following transcript.

$$\tau_c = f(M_i, Z_i) \mid M_i, Z_i \in \{0, 1\}^*, 1 \leq i \leq q. \quad (11)$$

We only consider valid queries here, i.e., $M_i \neq M_i^\theta$ for $i \neq i^\theta$ in τ_c .

⁶ For instance, if $M_1 = a \parallel a \parallel a$ and $M_2 = a \parallel a \parallel b$, then there are 6 distinct tuples.

⁷ This means there exists a random function Rand which on an input $M_i \in \{0, 1\}^*$ returns a random string $Z_i \in \{0, 1\}^*$.

Transcripts from primitive queries. Let the adversary A makes q_p and q_{p^θ} primitive queries to P and P^θ , respectively. The transcripts are summarized as follows.

$$\begin{aligned}\tau_p &= f(u_i, v_i) \mid u_i, v_i \in \{0, 1\}^b, P(u_i) = v_i, 1 \leq i \leq q_p \mathcal{G} \\ \tau_{p^\theta} &= f(u_i, v_i) \mid u_i, v_i \in \{0, 1\}^b, P^\theta(u_i) = v_i, 1 \leq i \leq q_{p^\theta} \mathcal{G}\end{aligned}\quad (12)$$

Note that each of the sets τ_p, τ_{p^θ} does not contain duplicates.

Transcripts after releasing keying material. After its interaction with oracles O and P , and before A outputs its final decision, we release the secret keying material used in the construction. This only improves the adversarial success probability and can be done without loss of generality. We first release the secret key K used in the construction. In the real world, K is the actual key used in the construction, and in the ideal world K is sampled uniformly from \mathcal{K} . The release of key-material provides additional inputs of permutation P arising from the key K , which are denoted as $\tau_{c,p}$ and are given by

$$\tau_{c,p} = fA_{i,j} \mid A_{i,j} = H_K(j) \oplus M_{i,j}, \text{ for } 1 \leq i \leq q, 1 \leq j \leq \ell_{M_i} \mathcal{G} \quad (13)$$

Thus, the entire transcripts can be denoted as $\tau = (\tau_c, \tau_p, \tau_{p^\theta}, K)$ or equivalently as $\tau = (\tau_c, \tau_p, \tau_{p^\theta}, \tau_{c,p})$.

4.2 Main Result on the PRF Security of Farasha

Theorem 3 (PRF security of Farasha). *Let $\epsilon > 0$ and $k, b \in \mathbb{N}$ with $k \leq b$ and $K \in \mathcal{K}$. Consider Farasha := $F_R \oplus F_L$ as defined in Section 3.1 where F_L is a uniform-AXU based on a ϵ -uniform-AXU H_K and a random permutation $P \in \text{Perm}(b)$, and F_R is a fixed input and variable output length PRF based on a random permutation $P^\theta \in \text{Perm}(b)$. For any adversary A making q construction queries (with σ being the number of distinct tuples $(j, M_{i,j})$ over all q queries), σ^θ output blocks (each query consisting of l_{z_i} output blocks and $\sigma^\theta = \sum_{i=1}^q l_{z_i}$), q_p primitive queries to P and q_{p^θ} primitive queries to P^θ ,*

$$\text{Adv}_{\text{Farasha}}^{\text{prf}}(A, q, q_p, q_{p^\theta}, \sigma, \sigma^\theta) \leq \left(\frac{\sigma^2}{2} \epsilon + \sigma q_p \epsilon + \frac{\sigma^2}{2} \frac{1}{2^b} \right) + \left(\frac{3\sigma^{\theta^2}}{2} \frac{1}{2^b} + \frac{2\sigma^\theta q_{p^\theta}}{2^b} \right).$$

Proof. The basic idea of proof is as follows. In Step-1, we observe that F_R can be mapped to a multi-key EM construction where the output of F_L is the key for every EM instance. In Step-2, we show that F_L is a uniform-AXU function and analyse F_R in the modified key-setting (given in Theorem 2). Next, we describe these two steps in details.

Step 1: In Farasha, the component F_R generates the PRF outputs. Thus, from the construction, it follows that

$$\text{Adv}_{\text{Farasha}}^{\text{prf}}(A, q, q_p, q_{p^\theta}, \sigma, \sigma^\theta) := \text{Adv}_{F_R}^{\text{prf}}(A^\theta, q, q_{p^\theta}, \sigma^\theta) \quad (14)$$

for an adversary A^ℓ . Note that in the following, we introduce the intermediate adversaries (similar to A^ℓ) if the meaning is clear from the context.

Now, a closer look at **Farasha** shows that F_R is a CTR-mode PRF where the counter values are encrypted using the EM block cipher with key X_i . Since the key changes per input message M_i , it is an instance of the multi-key Even Mansour block cipher. Note that F_R is a block-cipher based construction with its output indistinguishable from a random PRP. To compute the advantage of F_R as a PRF, we apply the PRP/PRF switching for the EM block-cipher instances (with a total of σ^ℓ queries). The advantage is captured by [Lemma 1](#) and we have the following:

$$\begin{aligned} \text{Adv}_{F_R}^{\text{prf}}(A^\ell, q, q_{p^\ell}, \sigma^\ell) &= \text{Adv}_{EM}^{\text{prf}}(A^{\ell\theta}, q, q_{p^\ell}, \sigma^\ell) \\ &= \text{Adv}_{EM}^{\text{prp}}(A^{\ell\theta}, q, q_{p^\ell}, \sigma^\ell) + \frac{\sigma^{\ell^2}}{2} \frac{1}{2^b} \end{aligned} \quad (15)$$

Step 2: The EM keys here resemble the SEM in the modified key setting (as discussed in [Theorem 2](#)) where they satisfy the ϵ^ℓ -uniform-AXU property, for a given $\epsilon^\ell > 0$. As shown in [Section 4.3](#), F_L is a ϵ^ℓ -AXU, however there is also an additional advantage arising from the number of AXU queries using the hash values ($X_i = F_L(K, M_i)$ for $i = 1, \dots, q$). We denote this advantage by $\text{Adv}_{F_L}^{\text{u-axu}}$. Consequently, the first term in [Equation 15](#) is given by

$$\text{Adv}_{EM}^{\text{prp}}(A^{\ell\theta}, q, q_{p^\ell}, \sigma^\ell) = \sigma^{\ell^2} \epsilon^\ell + 2\sigma^\ell q_{p^\ell} \epsilon^\ell + \text{Adv}_{F_L}^{\text{u-axu}}(B, q, q_p, \sigma) \quad (16)$$

Now, by [Lemma 2](#), we have

$$\text{Adv}_{F_L}^{\text{u-axu}}(B, q, q_p, \sigma) = \frac{\sigma^2}{2} \epsilon + \sigma q_p \epsilon + \frac{\sigma^2}{2} \frac{1}{2^b} \quad (17)$$

Note that $\epsilon^\ell = 2^{-b}$ when there is no collision among X_i 's. Thus, substituting $\epsilon^\ell = 2^{-b}$ in [Equation 16](#), and combining [Equations 15-17](#), we have

$$\text{Adv}_{\text{Farasha}}^{\text{prf}}(A, q, q_p, q_{p^\ell}, \sigma, \sigma^\ell) = \left(\frac{\sigma^2}{2} \epsilon + \sigma q_p \epsilon + \frac{\sigma^2}{2} \frac{1}{2^b} \right) + \left(\frac{3\sigma^{\ell^2}}{2} \frac{1}{2^b} + \frac{2\sigma^\ell q_{p^\ell}}{2^b} \right). \quad (18)$$

The first two terms in [Equation 16](#) are given by [Theorem 2](#) and the proof of [Equation 17](#) is given in [Section 4.3](#).

Remark 1. [Theorem 3](#) shows that for $\epsilon = 2^{-k}$, **Farasha** achieves a birthday-bound security in the key-size k and the permutation-size b , i.e., $\min\{k/2, b/2g\}$.

4.3 Uniform AXU Bound of Farasha-L

Recall that for $M_i \in \{0, 1\}^k$, we have $F_L(K, M_i) = \bigoplus_{j=1}^{\ell_{M_i}} P(H_K(j) \oplus M_{i,j})$ where $K \in \mathcal{K}$, $P \in \text{Perm}(b)$ and H_K is a ϵ -uniform-AXU. In [Lemma 2](#), we give the uniform-AXU bound of **Farasha-L**, i.e., the adversarial advantage $\text{Adv}_{F_L}^{\text{u-axu}}$.

Lemma 2 (Uniform AXU bound of F_L). *Let $\epsilon > 0$ and $k, b \geq 2 \mathbb{N}$ with $k \leq b$. Let $K \in \mathcal{K}$, $P \in \text{Perm}(b)$ and H_K be a ϵ -uniform-AXU. Consider F_L as defined above. For any adversary A making q construction queries (with σ being the number of distinct tuples $(j, M_{i,j})$ over all q queries) and q_p primitive queries to P ,*

$$\text{Adv}_{F_L}^{\text{u-axu}}(A, q, q_p, \sigma) \leq \frac{\sigma^2}{2} \epsilon + \sigma q_p \epsilon + \frac{\sigma^2}{2} \frac{1}{2^b}. \quad (19)$$

Proof. From the transcript $\tau = (\tau_c, \tau_p, \tau_{p^\circ}, \tau_{c,p})$, the subset of interest to A is $(\tau_c, \tau_p, \tau_{c,p})$. To prove the AXU (resp. uniformity) bound for F_L , we look at the XOR difference of the F_L output of two messages (resp. output of a message) when the number of queries made by the adversary are q . We divide the proof into three steps, with the first two steps capturing the adversary's advantage: (1) replace P by a random function $\rho \in \text{Func}(b)$, (2) define bad events and bound their probability and (3) prove the $\frac{1}{2^b}$ -uniform-AXU property when P is replaced by a random function ρ and no bad events occur.

Step 1: PRP-PRF switching. For the sake of brevity, let F_L° be F_L when P is replaced by a random function. Since there are σ distinct tuples $(j, M_{i,j})$ and $H_K(j) \oplus M_{i,j}$ is input to P , the number of calls to P is bounded by σ . By the PRP-PRF switching lemma [18], we have

$$\text{Adv}_{F_L}^{\text{u-axu}}(A, q, q_p, \sigma) \leq \frac{\sigma^2}{2} \frac{1}{2^b} + \text{Adv}_{F_L^\circ}^{\text{u-axu}}(A^\circ, q, q_p, \sigma). \quad (20)$$

Step 2: Accounting bad events. We define the following two events as bad events.

- **Bad₁:** Collision in the set $\tau_{c,p}$, i.e., $H_K(j) \oplus M_{i,j} = H_K(j^\circ) \oplus M_{i^\circ, j^\circ}$ for $(j, M_{i,j}) \neq (j^\circ, M_{i^\circ, j^\circ})$. If $j = j^\circ$, we have $M_{i,j} \neq M_{i^\circ, j^\circ}$ and hence there will be no collision. If $j \neq j^\circ$, a collision implies $H_K(j) \oplus H_K(j^\circ) = M_{i,j} \oplus M_{i^\circ, j^\circ}$, with the probability of this event bounded by ϵ (since H_K is ϵ -AXU). With the number of distinct input-tuples $(j, M_{i,j})$ being σ , the probability of a collision is at most $\binom{\sigma}{2} \epsilon \leq \sigma^2 \epsilon / 2$.
- **Bad₂:** Collision in the sets $\tau_{c,p}$ and τ_p , i.e., there exists $A_{i,j} \in \tau_{c,p}$ and $(u_r, v_r) \in \tau_p$ such that $A_{i,j} = u_r$. A collision between an element of $\tau_{c,p}$ (computed as $H_K(j) \oplus M_{i,j}$) and $(u_r, v_r) \in \tau_c$ implies $H_K(j) = u_r \oplus M_{i,j}$. Since H is ϵ -uniform, the probability of this event is bounded by ϵ . With $|\tau_{c,p}| = \sigma$ and $|\tau_p| = q_p$, the probability of a collision in the two sets is at most $\sigma q_p \epsilon$.

Now, we define $F_L^{\circ\circ}$ as F_L° when neither **Bad₁** nor **Bad₂** occurs. Then, accounting for the probability of the bad events, we have

$$\text{Adv}_{F_L}^{\text{u-axu}}(A, q, q_p, \sigma) \leq \left(\frac{\sigma^2}{2} \epsilon + \sigma q_p \epsilon + \frac{\sigma^2}{2} \frac{1}{2^b} \right) + \text{Adv}_{F_L^{\circ\circ}}^{\text{u-axu}}(A^{\circ\circ}, q, q_p, \sigma) \quad (21)$$

Step 3: Bounding $\text{Adv}_{\mathbb{F}_L^{\text{axu}}}^{\text{u-axu}}$. We show that $\mathbb{F}_L^{\text{axu}}$ is both $1/2^b$ -uniform and $1/2^b$ -AXU. We highlight that ρ is a random function in $\mathbb{F}_L^{\text{axu}}$ with no collisions in inputs to ρ .

- $\mathbb{F}_L^{\text{axu}}$ is $1/2^b$ -uniform: $\mathbb{F}_L(K, M_i)$ computed as $\bigoplus_{j=1}^{\ell_{M_i}} \rho(H_K(j) \parallel M_{i,j})$ is uniformly distributed as each input to ρ is distinct with each output of ρ being uniformly distributed. Thus, $\Pr[\mathbb{F}_L(K, M_i) = Y]$ is $1/2^b$ and $\mathbb{F}_L^{\text{axu}}$ is $1/2^b$ -uniform.
- $\mathbb{F}_L^{\text{axu}}$ is $1/2^b$ -AXU: For any given message pair (M_1, M_2) , we show that the XOR difference of $\mathbb{F}_L(K, M_1) \oplus \mathbb{F}_L(K, M_2)$ is uniformly distributed. Let

$$\Delta = \bigoplus_{j=1}^{\ell_{M_1}} \rho(H_K(j) \parallel M_{1,j}) \oplus \bigoplus_{j=1}^{\ell_{M_2}} \rho(H_K(j) \parallel M_{2,j}). \quad (22)$$

Without loss of generality, assume that $\ell_{M_1} \geq \ell_{M_2}$. Since $M_1 \neq M_2$ there always exists an index j^* such that the message blocks M_{1,j^*} and M_{2,j^*} differ. If $\ell_{M_1} > \ell_{M_2}$, then take $j^* = \ell_{M_2} + 1$ and if $\ell_{M_1} = \ell_{M_2}$, choose j^* as a block-index where M_1 and M_2 differ. Accordingly, we rewrite Equation 22 as

$$\Delta = \rho(H_K(j^*) \parallel M_{1,j^*}) \oplus \left(\bigoplus_{j=1, j \neq j^*}^{\ell_{M_1}} \rho(H_K(j) \parallel M_{1,j}) \oplus \bigoplus_{j=1}^{\ell_{M_2}} \rho(H_K(j) \parallel M_{2,j}) \right). \quad (23)$$

Since the term at index j^* is independent of the rest of the right-hand side of Equation 23, and ρ is a random function, the term at index j^* is uniformly distributed. Hence, the probability of Δ being equal to $\mathbb{F}_L(K, M_1) \oplus \mathbb{F}_L(K, M_2)$ is $1/2^b$. Thus $\mathbb{F}_L^{\text{axu}}$ is 2^{-b} -AXU.

Substituting $\text{Adv}_{\mathbb{F}_L^{\text{axu}}}^{\text{u-axu}} = 1/2^b$ proves the lemma.

Remark 2. It is crucial to note that \mathbb{F}_L is not 2^{-b} -uniform-AXU if the adversary is allowed unbounded number of queries. It satisfies uniform-AXU property with a bounded number of queries with the advantage given in Lemma 2. When no bad event happens, \mathbb{F}_L is 2^{-b} -uniform-AXU (exactly as $\mathbb{F}_L^{\text{axu}}$ in Step 3 above). Furthermore, it is always possible to construct queries of the form $M_1 = (M_{1,1}, M_{1,2})$, $M_2 = (M_{2,1}, M_{2,2})$, $M_3 = (M_{1,1}, M_{2,2})$ and $M_4 = (M_{2,1}, M_{1,2})$ so that $\sum_{i=1}^4 X_i = 0$. However, since X_i 's are secret, nothing is revealed about $X_i \oplus X_j$.

5 Further Insights on Farasha and Farfalle

5.1 Farasha based on Regular-size Keys

In Theorem 3, we have shown that Farasha achieves only $k/2$ bits of security for k -bit key. A natural question is whether this can be improved to full k bits of security. To answer this, we present Farasha[#], a variant of Farasha. Farasha[#] is

exactly similar to **Farasha**, except for the fact that we expand the master key K to another $2k$ -bit key K° (satisfying $2k \geq b$) by a single call of the b -bit permutation $P^{\circ\circ}$. The key K° and the message M are then used as inputs to **Farasha**. This extension of **Farasha** to **Farasha[#]** is shown in [Figure 4](#) and the security proof is given in [Appendix A.2](#). Note that the permutation $P^{\circ\circ}$ is analogous to the permutation p_b in **Farfalle**.

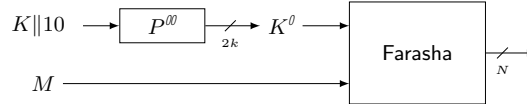


Fig. 4: A generic diagram of **Farasha[#]**

5.2 On Performance Trade-offs in **Farasha**

Although the goal of this work is to understand the provable security aspects, we now highlight that the design choices in **Farasha** could provide different performance trade-offs. For instance, given a 8-bit micro controller, we can choose H_K as an LFSR over finite field \mathbb{F}_{2^8} . We could further select a primitive feedback polynomial that allows either efficient implementations or require low latency. Similarly, one could choose an LFSR for 32 and 64-bit architectures. In the case where hardware resources are not limited, a designer may choose H_K as a finite field multiplier (similar to AES-GCM). Moreover, since **Farasha** use public permutations, then for concrete instantiations, we could choose any secure permutation or its round-reduced variant (given that it is also secure) based on the performance requirements. All-in-all, based on the use-case and performance requirements, we could select an appropriate H_K and permutation.

We also believe that the **Farasha-L** can be abstracted to any generic uniform AXU function. Then it gives many new choices for AXUs. One interesting example is that 4-round AES with uniform and independent keys is an 2^{113} -AXU [21].

5.3 Discussion on the Security of **Farfalle**

While the goal of **Farfalle** is to have an efficient design with security claims, as opposed to provable security, we can analyse **Farfalle** in the random permutation model under the following assumptions:

- $roll_c$ is a uniform-AXU: The authors of **Farfalle** mention that $roll_c$ should possess the properties of a uniform-AXU, i.e., “*Informally, an adversary not knowing K shall not be able to predict the mask value $roll_c^i(K)$ for any i in a reasonable range, nor the difference between any pair of mask values $roll_c^i(K)$ and $roll_c^j(K)$ for any $i \neq j$ in that range*” [7, Section 2.3]. In fact

for $roll_c$ being an LFSR with primitive feedback polynomial, this property holds.

- p_b, p_c, p_d and p_e are random permutations: While the authors do not make this assumption (and use a 6-round Xoodoo instead), we analyze the overall Farfalle construction where permutations are modelled as a random permutation.

Under these assumptions, we can show that Farfalle is birthday bound secure in the random permutation model. We omit the details of proof due to space limitation and will provide them in the full version of the paper.

6 Conclusion

In this work, we have proposed Farasha, a permutation-based variable length input and output pseudo random function, which is parallelizable and provably secure. The Farasha PRF relies on a uniform almost xor universal hash function and a counter for its provable security. We presented Farasha-wLFSR, where the uniform AXU is an LFSR whose output state is always secret. We then proved that Farasha is birthday-bound secure and also have shown that a slight modification ensures full security in the key-size. Moreover, we discussed different AXUs and the security of Farfalle in the random permutation model. Finally, since our work presents the first formal treatment of Farasha and Farfalle, finding more tight bounds for these constructions is another interesting research direction. We also believe this work will bring new insights to the readers in further understanding the provable security of Farfalle-like constructions.

7 Acknowledgments

The authors would like to thank the reviewers of SAC 2022 for their insightful comments which improved the quality of the paper.

References

1. Aumasson, J., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A lightweight hash. *J. Cryptol.* **26**(2), 313–339 (2013), <https://doi.org/10.1007/s00145-012-9125-6>
2. Aumasson, J., Jovanovic, P., Neves, S.: NORX: parallel and scalable AEAD. In: Kutyłowski, M., Vaidya, J. (eds.) *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security*, Wroclaw, Poland, September 7–11, 2014. *Proceedings, Part II. Lecture Notes in Computer Science*, vol. 8713, pp. 19–36. Springer (2014), https://doi.org/10.1007/978-3-319-11212-1_2
3. Bernstein, D.J.: Chacha, a variant of Salsa20. In: *Workshop Record of SASC (2008)*, cr.yp.to/papers.html#chacha
4. Bernstein, D.J.: The salsa20 family of stream ciphers. In: Robshaw, M.J.B., Billet, O. (eds.) *New Stream Cipher Designs - The eSTREAM Finalists*, *Lecture Notes in Computer Science*, vol. 4986, pp. 84–97. Springer (2008), https://doi.org/10.1007/978-3-540-68351-3_8

5. Bernstein, D.J., Kölbl, S., Lucks, S., Massolino, P.M.C., Mendel, F., Nawaz, K., Schneider, T., Schwabe, P., Standaert, F., Todo, Y., Viguier, B.: Gimli : A cross-platform permutation. In: Fischer, W., Homma, N. (eds.) *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference*, Taipei, Taiwan, September 25-28, 2017, Proceedings. *Lecture Notes in Computer Science*, vol. 10529, pp. 299–320. Springer (2017), https://doi.org/10.1007/978-3-319-66787-4_15
6. Bertoni, G., Daemen, J., Peeters, M., Assche, G.: CAESAR submission: Ketje v2 (2014), <http://ketje.noekeon.org/Ketjev2-doc2.0.pdf>
7. Bertoni, G., Daemen, J., Hoffert, S., Peeters, M., Assche, G.V., Keer, R.V.: Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.* **2017**(4), 1–38 (2017), <https://tosc.iacr.org/index.php/ToSC/article/view/801>
8. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Sponge-based pseudo-random number generators. In: Mangard, S., Standaert, F. (eds.) *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop*, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings. *Lecture Notes in Computer Science*, vol. 6225, pp. 33–47. Springer (2010), https://doi.org/10.1007/978-3-642-15031-9_3
9. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the sponge: Single-pass authenticated encryption and other applications. In: Miri, A., Vaudenay, S. (eds.) *Selected Areas in Cryptography - 18th International Workshop, SAC 2011*, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers. *Lecture Notes in Computer Science*, vol. 7118, pp. 320–337. Springer (2011), https://doi.org/10.1007/978-3-642-28496-0_19
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. In: *Hash Functions Workshop* (2007), <https://keccak.team/files/SpongeFunctions.pdf>
11. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Keccak specifications. Submission to NIST (Round 2) (2009)
12. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Permutation-based encryption, authentication and authenticated encryption. *DIAC* (2012)
13. Beyne, T., Chen, Y.L., Dobraunig, C., Mennink, B.: Dumbo, jumbo, and delirium: Parallel authenticated encryption for the lightweight circus. *IACR Trans. Symmetric Cryptol.* **2020**(S1), 5–30 (2020), <https://doi.org/10.13154/tosc.v2020.iS1.5-30>
14. Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In: Knudsen, L.R. (ed.) *Advances in Cryptology - EUROCRYPT 2002*, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. *Lecture Notes in Computer Science*, vol. 2332, pp. 384–397. Springer (2002), https://doi.org/10.1007/3-540-46035-7_25
15. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: spongent: A lightweight hash function. In: Preneel, B., Takagi, T. (eds.) *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop*, Nara, Japan, September 28 - October 1, 2011. Proceedings. *Lecture Notes in Computer Science*, vol. 6917, pp. 312–325. Springer (2011), https://doi.org/10.1007/978-3-642-23951-9_21
16. Boneh, D., Shoup, V.: *A Graduate Course in Applied Cryptography*. cryptobook.us (2020)

17. Chakraborty, D., Sarkar, P.: A general construction of tweakable block ciphers and different modes of operations. *IEEE Trans. Inf. Theory* **54**(5), 1991–2006 (2008), <https://doi.org/10.1109/TIT.2008.920247>
18. Chang, D., Nandi, M.: A short proof of the PRP/PRF switching lemma (2008), <http://eprint.iacr.org/2008/078>
19. Daemen, J., Hoffert, S., Assche, G.V., Keer, R.V.: The design of xoodoo and xooff. *IACR Trans. Symmetric Cryptol.* **2018**(4), 1–38 (2018), <https://doi.org/10.13154/tosc.v2018.i4.1-38>
20. Daemen, J., Hoffert, S., Peeters, M., Assche, G.V., Keer, R.V.: Xoodyak, a lightweight cryptographic scheme. *IACR Trans. Symmetric Cryptol.* **2020**(S1), 60–87 (2020), <https://doi.org/10.13154/tosc.v2020.iS1.60-87>
21. Daemen, J., Lamberger, M., Pramstaller, N., Rijmen, V., Vercauteren, F.: Computational aspects of the expected differential probability of 4-round AES and aes-like ciphers. *Computing* **85**(1-2), 85–104 (2009), <https://doi.org/10.1007/s00607-009-0034-y>
22. Daemen, J., Massolino, P.M.C., Mehrdad, A., Rotella, Y.: The subterranean 2.0 cipher suite. *IACR Trans. Symmetric Cryptol.* **2020**(S1), 262–294 (2020), <https://doi.org/10.13154/tosc.v2020.iS1.262-294>
23. Dobraunig, C., Eichlseder, M., Mendel, F., Schl affer, M.: Ascon v1.2: Lightweight authenticated encryption and hashing (2021), <https://doi.org/10.1007/s00145-021-09398-9>
24. Dobraunig, C., Grassi, L., Guinet, A., Kuijsters, D.: Ciminion: Symmetric encryption based on toffoli-gates over large finite fields. In: Canteaut, A., Standaert, F. (eds.) *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II. *Lecture Notes in Computer Science*, vol. 12697, pp. 3–34. Springer (2021), https://doi.org/10.1007/978-3-030-77886-6_1
25. Dunkelmann, O., Keller, N., Shamir, A.: Minimalism in cryptography: The even-mansour scheme revisited. In: Pointcheval, D., Johansson, T. (eds.) *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Cambridge, UK, April 15-19, 2012. Proceedings. *Lecture Notes in Computer Science*, vol. 7237, pp. 336–354. Springer (2012), https://doi.org/10.1007/978-3-642-29011-4_21
26. Dworkin, M.J.: Sp 800-38d: Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. National Institute of Standards & Technology (2007)
27. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. *J. Cryptol.* **10**, 151–162 (1997), <https://doi.org/10.1007/s001459900025>
28. Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved masking for tweakable blockciphers with applications to authenticated encryption. In: Fischlin, M., Coron, J. (eds.) *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Vienna, Austria, May 8-12, 2016, Proceedings, Part I. *Lecture Notes in Computer Science*, vol. 9665, pp. 263–293. Springer (2016), https://doi.org/10.1007/978-3-662-49890-3_11
29. Guning, A., Daemen, J., Mennink, B.: Deck-based wide block cipher modes and an exposition of the blinded keyed hashing model. *IACR Trans. Symmetric Cryptol.* **2019**(4), 1–22 (2019), <https://doi.org/10.13154/tosc.v2019.i4.1-22>

30. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON family of lightweight hash functions. In: Rogaway, P. (ed.) *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference*, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 222–239. Springer (2011), https://doi.org/10.1007/978-3-642-22792-9_13
31. Jean, J., Nikolic, I., Peyrin, T., Seurin, Y.: Deoxys v1. 41. Submitted to CAESAR **124** (2016), <https://competitions.cr.yj.to/round3/deoxysv141.pdf>
32. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*. Lecture Notes in Computer Science, vol. 6733, pp. 306–327. Springer (2011), https://doi.org/10.1007/978-3-642-21702-9_18
33. McGrew, D., Viega, J.: The Galois/counter mode of operation (GCM). Submission to NIST Modes of Operation Process **20**, 0278–0070 (2004)
34. Minematsu, K.: Parallelizable rate-1 authenticated encryption from pseudorandom functions. In: Nguyen, P.Q., Oswald, E. (eds.) *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 275–292. Springer (2014), https://doi.org/10.1007/978-3-642-55220-5_16
35. Mouha, N., Luykx, A.: Multi-key security: The even-mansour construction revisited. In: Gennaro, R., Robshaw, M. (eds.) *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference*, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 209–223. Springer (2015), https://doi.org/10.1007/978-3-662-47989-6_10
36. Patarin, J.: The "Coefficients H" Technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*. Lecture Notes in Computer Science, vol. 5381, pp. 328–345. Springer (2008), https://doi.org/10.1007/978-3-642-04159-4_21
37. Peyrin, T., Seurin, Y.: Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In: Robshaw, M., Katz, J. (eds.) *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference*, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 33–63. Springer (2016), https://doi.org/10.1007/978-3-662-53018-4_2
38. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Trans. Inf. Syst. Secur.* **6**(3), 365–403 (2003), <https://doi.org/10.1145/937527.937529>
39. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4004, pp. 373–390. Springer (2006), https://doi.org/10.1007/11761679_23

A Security Proofs

A.1 Proof of Lemma 1

Proof. Let σ_i be the number of queries to P_i . By the PRP/PRF switching lemma, the advantage after switching P_i by PRF ρ_i is bounded by $\frac{\sigma_i^2}{2 \cdot 2^b}$. Since the permutations as well as the functions are independent of each other, the total advantage of the μ switchings is given by

$$\frac{1}{2} \frac{1}{2^b} \sum_{i=1}^{\mu} \sigma_i^2 \leq \frac{1}{2} \frac{1}{2^b} \left(\sum_{i=1}^{\mu} \sigma_i \right)^2 = \frac{\sigma^2}{2 \cdot 2^b}. \quad (24)$$

A.2 PRF Security of Farasha[#]

In Farasha[#] (see Section 5.1), we first expand a k -bit key K to a $2k$ -bit key $K^0 = P^{00}(K)$. The key K^0 is then used as a key for Farasha. Denote the modified left-side as $\text{Farasha-L}^{\#}(K, M) = \text{Farasha-L}(P^{00}(K), M)$. Then, the security of Farasha[#] follows from Lemma 3.

Lemma 3 (PRF security of Farasha[#]). *Consider Farasha as defined in Theorem 3 and additionally let $P^{00} \in \text{Perm}(b)$ and Farasha-L[#] as defined above. Then for adversaries A, A^0 , we have*

$$\begin{aligned} & \text{Adv}_{\text{Farasha-L}^{\#}}^{\text{u-axu}}(A) \leq \frac{q_{p^{00}}}{2^k} + \text{Adv}_{\text{Farasha-L}}^{\text{u-axu}} \\ \Rightarrow & \text{Adv}_{\text{Farasha}^{\#}}^{\text{prf}}(A^0) \leq \frac{q_{p^{00}}}{2^k} + \text{Adv}_{\text{Farasha}}^{\text{prf}}, \end{aligned} \quad (25)$$

where $q_{p^{00}}$ is the number of primitive queries to P^{00} .

Proof. With another public permutation P^{00} for the key expansion, P^{00} is the additional (primitive) oracle available to the adversary A . Let A make $q_{p^{00}}$ primitive queries to P^{00} and denote its input-output pairs as $\tau_{p^{00}} = \{(u_i, v_i) \mid P^{00}(u_i) = v_i, 1 \leq i \leq q_{p^{00}}\}$. Now, in addition to all cases in proof of Theorem 3, we need to consider an additional bad event, i.e., when one of the queries in $\tau_{p^{00}}$ matches the key K . The probability of this event is at most $q_{p^{00}}/2^k$. Given that the construction after the key expansion is identical to Farasha-L, accounting for this term in Lemma 2 (resp. Theorem 3) gives the adversarial advantage of Farasha-L[#] (resp. Farasha[#]) as given in Equation 25. This proves the lemma.

A.3 Proof of Theorem 2

Proof Idea. Let (M_{ij}, C_{ij}) denote the j -th plaintext and ciphertext pair corresponding to the EM instance with key K_i , i.e., E_{K_i} . Furthermore, (x_j, y_j)

denote the input/output of the j -th public permutation query. The bad events are identical to the independent keys analysis done in [35], and are as follows.

$$\mathcal{G}_{i, i^0, j, j^0} : i \notin i^0 : M_{ij} \quad M_{i^0 j^0} = K_i \quad K_{i^0} \quad C_{ij} \quad C_{i^0 j^0} = K_i \quad K_{i^0} \quad (26a)$$

$$\mathcal{G}_{i, j, j^0} : M_{ij} \quad x_{j^0} = K_i \quad C_{ij} \quad y_{j^0} = K_i \quad (26b)$$

The probabilities that Equation 26a and Equation 26b hold is bounded by ϵ due to the ϵ -uniform-AXU property. The rest of the proof is identical to the one in [35], and the $1/2^b$ term can be generalised to ϵ with $\epsilon = 1/2^b$ for independent keys.

Proof. The adversarial model for the modified key setting is depicted in Figure 3, with an adversary A having bidirectional access to $\mu+1$ oracles (O_1, \dots, O_μ, O) . In the ideal world, these are $(P_1, \dots, P_\mu, P) \in \text{Perm}(n)^{\mu+1}$. In the real world, these are $(E_{K_1}, \dots, E_{K_\mu}, P)$, where $E_{K_i}(M_{ij}) = P(M_{ij} \oplus K_i) \oplus K_i$ for $i = 1, \dots, \mu$. Note that, in this setting, the keys K_i are the output of a keyed hash function H_K and satisfy the ϵ -uniform-AXU property. The adversary makes σ_i queries to oracle O_i (resp. q_p queries to oracle O), which are captured in transcripts τ_i for $1 \leq i \leq \mu$ (resp. τ_p). Thus, the transcripts in the real-world are:

$$\begin{aligned} \tau_i &= f(M_{ij}, C_{ij}) \parallel M_{ij}, C_{ij} \parallel f0, 1g^b, E_{K_i}(M_{ij}) = C_{ij}, 1 \leq i \leq \mu \parallel \sigma_i g \\ \tau_p &= f(x_i, y_i) \parallel x_i, y_i \parallel f0, 1g^b, P(x_i) = y_i, 1 \leq i \leq q_p g \end{aligned} \quad (27)$$

We assume the adversary never makes duplicate queries, so that $M_{ij} \neq M_{i^0 j^0}, C_{ij} \neq C_{i^0 j^0}, x_j \neq x_{j^0}, y_i \neq y_{j^0}$ for all i, j, j^0 where $j \neq j^0$. We denote the total number of keyed (or construction) queries by $\sigma = \sum_{i=1}^{\mu} \sigma_i$.

After all the queries by A are done, but before it outputs its decision, the key K (of the hash function H_K) in the real world and a dummy key in the ideal world is released to the adversary. This enables the adversary to compute the keys (K_1, \dots, K_μ) . The interaction of A with the oracles can be summarized by a transcript $\tau = fK, \tau_1, \dots, \tau_\mu, \tau_p g$ or equivalently, $\tau = fK_1, \dots, K_\mu, \tau_1, \dots, \tau_\mu, \tau_p g$.

Without loss of generality we assume that A is deterministic. Given the fixed deterministic adversary A , we denote the probability distribution of transcripts in the real world by X , and in the ideal world by Y . We say that a transcript τ is attainable if it can be obtained from interacting with (P_1, \dots, P_μ, P) , i.e., $\Pr[Y = \tau] > 0$. In our proof, we use the H-coefficient technique, as given by Lemma 4.

Lemma 4. (*H-coefficient Technique [36]*). *Let us consider a fixed deterministic adversary A , and let $T = T_{\text{good}} \parallel T_{\text{bad}}$ be a partition of the set of attainable transcripts. Let δ be such that for all $\tau \in T_{\text{good}}$*

$$\frac{\Pr[X = \tau]}{\Pr[Y = \tau]} \leq 1 + \delta \quad (28)$$

Then, $\text{Adv}_{EM}^{\text{PPP}}(A, \sigma) \leq \delta + \Pr[Y \in T_{\text{bad}}]$.

We say that a transcript $\tau \geq T$ is bad if two different queries result in the same input (or output) to P , were A interacting with the real world. Stated formally, τ is bad if one of the following conditions is met:

$$\mathcal{Q}i, i^0, j, j^0 : i \neq i^0 : M_{ij} \quad M_{i^0j^0} = K_i \quad K_{i^0} \quad - C_{ij} \quad C_{i^0j^0} = K_i \quad K_i^0 \quad (29a)$$

$$\mathcal{Q}i, j, j^0 : M_{ij} \quad x_{j^0} = K_i \quad - C_{ij} \quad y_{j^0} = K_i \quad (29b)$$

A transcript that is not a bad transcript, is referred to as a good transcript.

Upper Bounding $\Pr[Y \geq T_{bad}]$. We want to upper bound the event that a transcript τ in the ideal world satisfies Equation 29a or 29b. Recall that for $i = 1, \dots, \mu$, keys K_i satisfy the ϵ -uniform-AXU property. For any fixed $i \neq i^0$, there are at most $2\sigma_i\sigma_{i^0}$ possible plaintext pairs and ciphertext pairs. With keys satisfying ϵ -AXU property, the probability of satisfying the condition in Equation 29a is bounded by $2\sigma_i\sigma_{i^0}\epsilon$. Analogously, for any fixed i , there are at most $2\sigma_iq_p$ distinct values in Equation 29b. Since the keys are also ϵ -uniform, the probability of satisfying the condition in Equation 29b is bounded by $2\sigma_iq_p\epsilon$. Therefore,

$$\Pr[Y \geq T_{bad}] \leq (\sum_i \sum_{i^0 < i} 2\sigma_i\sigma_{i^0}\epsilon) + (\sum_i 2\sigma_iq_p\epsilon) \\ \leq \sigma^2\epsilon + 2\sigma q_p\epsilon.$$

Lower Bounding Ratio $\Pr[X = \tau] / \Pr[Y = \tau]$. Let us consider a good and attainable transcript $\tau \geq T_{good}$. Then, denote by $\Omega_X = 2^b \cdot 2^b!$ the set of all possible oracles in the real world and by $comp_X(\tau) \subseteq \Omega_X$ the set of oracles in Ω_X compatible with transcript τ . Define $\Omega_Y = 2^b \cdot (2^b!)^{\mu+1}$ and $comp_Y(\tau)$ similarly. According to the H-coefficient technique:

$$\Pr[X = \tau] = \frac{|comp_X(\tau)|}{|\Omega_X|} \quad \text{and} \quad \Pr[Y = \tau] = \frac{|comp_Y(\tau)|}{|\Omega_Y|} \quad (31)$$

First, we calculate $|comp_X(\tau)|$. As $\tau \geq T_{good}$, there are no two queries in τ with the same input or output of the underlying permutation. Any query tuple in τ , therefore, fixes exactly one input-output pair of the underlying oracle. Because τ consists of $\sigma + q_p$ query tuples, the number of possible oracles in the real world equals $(2^b - \sigma - q_p)!$. By a similar reasoning, the number of possible oracles in the ideal world equals $\prod_{i=1}^{\mu} (2^b - \sigma_i)! \cdot (2^b - q_p)!$. Therefore,

$$\Pr[X = \tau] = \frac{(2^b - \sigma - q_p)!}{2^b \cdot 2^b!} \quad (32)$$

$$\Pr[Y = \tau] = \frac{\prod_{i=1}^{\mu} (2^b - \sigma_i)! \cdot (2^b - q_p)!}{2^b \cdot (2^b!)^{\mu+1}} \\ = \frac{(2^b - \sigma - q_p)! \cdot (2^b!)^{\mu}}{2^b \cdot (2^b!)^{\mu+1}} \quad (33) \\ = \frac{(2^b - \sigma - q_p)!}{2^b \cdot 2^b!} = \Pr[X = \tau]$$

It then follows that $\Pr[X = \tau] / \Pr[Y = \tau] \leq 1$. Thus,

$$\text{Adv}_{EM}^{\text{PP}}(A, \sigma) \leq \Pr[Y \in T_{bad}] \leq \sigma^2 \epsilon + 2\sigma q_p \epsilon.$$

This proves the theorem.