

A Sponge-Based PRF with Good Multi-user Security^{*}

Arghya Bhattacharjee¹, Ritam Bhaumik^{1,2**}, and Mridul Nandi¹

¹ Indian Statistical Institute, Kolkata

² INRIA, Paris

{bhattacharjeearghya29 bhaumik.ritam mridul.nandi}@gmail.com

Abstract. Both multi-user PRFs and sponge-based constructions have generated a lot of research interest lately. Dedicated analyses for multi-user security have improved the bounds a long distance from the early generic bounds obtained through hybrid arguments, yet the bounds generally don't allow the number of users to be more than birthday-bound in key-size. Similarly, known sponge constructions suffer from being only birthday-bound secure in terms of their capacity. We present in this paper Muffler, a multi-user PRF built from a random permutation using a full-state sponge with feed-forward, which uses a combination of the user keys and unique user IDs to solve both the problems mentioned by improving the security bounds for multi-user constructions and sponge constructions. For D construction query blocks and T permutation queries, with key-size $\kappa = n/2$ and tag-size $\tau = n/2$ (where n is the state-size or the size of the underlying permutation), both D and T must touch birthday bound in n in order to distinguish Muffler from a random function.

Keywords: Sponge, Multi-User, PRF, public permutation

1 Introduction

Multi-User Security. The study of provably secure symmetric-key modes has traditionally revolved around the single-user setting, where a single user generates the keys of the various underlying primitives and uses them to respond to all subsequent adversary queries. However, it has long been recognised that often a more practically relevant scenario is the multi-user setting, where several users generate their own keys independently, and the adversary can query any or all of them. The notion of multi-user security was first introduced by Bellare, Boldyreva and Micali [BBM00].

One of the possible reasons why research in this direction did not garner sufficient interest is that it was established quite at the outset that when μ users are involved, the security bound does not degrade by more than a factor

^{*} A version of this paper with minor differences was presented at SAC 2022.

^{**} This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 714294 - acronym QUASYModo).

of μ . This generic bound looked like a satisfactory conclusion to the problem at that time; however, with the rapidly increasing expected number of users in practice, it has been evident for some time now that this degradation can be quite significant.

This realisation has led to a growing interest in recent times in dedicated security analyses of modes for the multi-user settings, which has been shown to yield bounds much better than the generic one.

Multi-user degradation. As noted by Biham [Bih02], there exists a faster generic key-recovery attack on any block cipher in the multi-key setting compared to the single-key setting. In this work Biham established the key-recovery trade-off $\mu T = 2^\kappa$ where μ denotes the number of users, κ is the key-size of a symmetric-key algorithm and T is the time (mainly the number of primitive calls used in the target algorithm).

As shown by Hong and Sarkar [HS05], and by Biryukov et al. [BMS05], the stream cipher time-memory-data tradeoffs can be applied to the block cipher setting as well, assuming that a plaintext is encrypted under multiple keys. Their work generalizes the findings of Biham. The observation that recovering one key out of a large group of keys can often be easier is applicable to any deterministic symmetric-key algorithm, as is done for MACs by Chatterjee, Menezes, and Sarkar [CMS11a].

By using standard hybrid argument the above trade-off can be shown to be tight [ML15]. Whenever the number of users increased rapidly, the multi-user security is no longer κ bits (which is usually a desirable level of security). In this paper, one of the main motivations is to recover κ -bit security even in the multi-user model. Previously there were some attempts based on randomization (e.g., randomized GCM in TLS 1.3 [BT16]). However, it still suffers from the high key-collision probability among users which adds a term like $\mu^2/2^\kappa$ to the bound. Thus, for a large number of users, this construction is still birthday bound on the key-size.

Sponge-based Constructions. A popular and useful design paradigm for modes in recent years has been the sponge-based constructions [BDPVA07], which was used in the SHA3 hash function Keccak. In the sponge mode the internal state of the public permutation is split into an r -bit part to be released to the adversary and a hidden c -bit part; r and c are called the *rate* and *capacity* of the construction respectively. With the rise in popularity of permutation-based designs, the efficiency and security of sponges have increasingly drawn the attention of researchers. Till now all sponge constructions have been limited to the birthday bound in terms of its capacity [ADMVA15, BDPVA11b, BDPVA11a]. This is mainly due to the length-extension of sponge construction. In other words, the security bound of the sponge-based PRF is mostly determined by the term $D^2/2^c$, where D is the data available to the adversary (measured by the number of construction queries).

Our second motivation of the paper is to improve this bound for sponge constructions by introducing a simple variant. Our variant enables to obtain

$\min\{(c+r)/2, c\}$ bit security. When we consider the multi-user model, our bound turns out to be roughly $\min\{(c+r)/2, c, \kappa\}$. With appropriate choice of parameters of c and r , we can achieve almost optimal security.

1.1 Our Contributions.

We present in this paper *Muffler*, a multi-user PRF built from a random permutation using a full-state sponge with feed-forward, which uses a combination of the user keys and unique user IDs to improve the security bounds both for multi-user constructions and sponge constructions. For D construction query blocks and T permutation queries, with key-size $\kappa = n/2$ and tag-size $\tau = n/2$, both D and T must touch birthday bound in n in order to distinguish *Muffler* from a random function.

Comparison to Full-State Keyed Duplex [DMA17]. Our proposed construction is similar to the full-state Duplex, with an added feed forward of the processed key into the final permutation call. We also replace the IV used in Duplex with a unique ID assigned to each user. The bound we obtain for the multi-user case is better than the bound in [DMA17]. Eqn. 2 in [DMA17] suggests that the dominating terms in their bound is $q_{iv}T/2^\kappa$ and $LT/2^c$, where κ is the key-length, c is the capacity, q_{iv} is the maximum number of keys which are called on the same IV, T is the offline complexity, and L is the number of *repeated paths*, i.e., repeated message prefix to the same key; unless IV is refreshed for every user (or for every t users for a small t), we can expect q_{iv} to be of the order of u , the number of users, and if IV is used as an ID, L is of the order of D , so their security term always has either a $uT/2^\kappa$ term (multi-user degradation of the single-user $T/2^\kappa$ term) or a $DT/2^c$ term (signifying birthday-bound in capacity). Our design solves both these problems.

1.2 Related Work.

Initial works can be traced back to Biham [Bih02] in symmetric cryptanalysis and Bellare et al. [BBM00] in public-key encryption. Biham [Bih02] considered the security of block ciphers in the multi-target setting and later Biryukov et al. [BMS05] refined it as a time-memory-data trade-off to demonstrate how one can take advantage of the fact that recovering a block cipher key out of a large group of keys is much easier than targeting a specific key. The same observation can be applied to any deterministic symmetric-key algorithm, as done for MACs by Chatterjee et al. [CMS11b]. Bellare et al. [BT16] first formalized a multi-user secure authenticated encryption scheme and also analyzed countermeasures against multi-key attacks in the context of TLS 1.3. Andreeva et al. [ADMVA15] considered the security of the outer and inner keyed sponge in the multi-user setting, a proof which internally featured a security analysis of the Even-Mansour block cipher in the multi-user setting. The direction of multi-user security got subsequently popularized by Mouha et al. [ML15], leading to various multi-user

security results [BBT16,HT16a] with security bounds almost independent of the number of users involved.

Since Chatterjee et al. [CMS11b], multi-user security of MACs has been studied by Morgan et al. [MPS20] and Bellare et al. [BKR98]. The security of Db-HtS (Double-block Hash-then-Sum) in the multi-user setting has been analysed by Shen et al. [SWGW21], Guo et al. [GW22] and Datta et al. [DDNT22]. Multi-user security of XORP[3] (bitwise-xor of 3 outputs of n -bit pseudorandom random permutations with domain separated inputs) has been analysed by Bhattacharya et al. [BN21]. Various other related works can also be found [Ber05,BHT18,HT17,HS05,LMP17].

The multi-user security of various other modes has been of significant research interest in recent years. One such class of functions is the cascade family [BCK96]; Bellare, Bernstein and Tessaro have studied the multi-user security of AMAC [BBT16], a cascade-based MAC function. Some other constructions of interest in the context of multi-user security have been the key-alternating ciphers [HT16a], Tweakable Even-Mansour [GWLZ17], and double encryption [HT17]. Bose, Hoang and Tessaro presented a multi-user security analysis of AES-GCM-SIV [BHT18]. Another direction of research for sponge constructions has been that of indistinguishability [BDPVA08].

2 Preliminaries

Mathematical and Notational Preliminaries. For integers i, j with $i \leq j$, $[i..j]$ will denote the set $\{k \mid i \leq k \leq j\}$. The notation for $[1..j]$ will be abbreviated to $[j]$. $\{0, 1\}^m$ will denote the set of all binary strings of length m , and $\{0, 1\}^{\geq m}$ will denote all binary strings of size at least m . For a finite set S , $|S|$ will denote its size. Thus,

$$|\{0, 1\}^m| = 2^m.$$

For a binary string x of length m , and i, j such that $i \leq j \leq m$, $x[i..j]$ will denote the contiguous substring of x starting at the i -th bit and ending at the j -th bit. For a finite set S and a random variable X , we say X is *uniformly sampled* from S , denoted $X \stackrel{\$}{\leftarrow} S$, if for each $x \in S$,

$$\Pr [X = x] = \frac{1}{|S|}.$$

Thus, when a binary string of length m is uniformly sampled, every string is picked with a probability $1/2^m$. A *random function* $f : S \rightarrow \{0, 1\}^m$ samples $f(x)$ uniformly from $\{0, 1\}^m$ for each $x \in S$. A function $f : S_1 \rightarrow S_2$ is called *injective* if for any distinct $x_1, x_2 \in S_1$, $f(x_1) \neq f(x_2)$. An injective function from S to S is called a *permutation* over S . For two binary strings x, y , $x||y$ will denote their concatenation. For $b \in \{0, 1\}$ and $m \geq 0$, b^m will denote the m -bit string with each bit identical and equal to b . We fix n to be the *block-size* for the rest of this paper, and each member of $\{0, 1\}^n$ is considered a block. The function $\text{fixl}(\cdot, \cdot)$ fixes the last bit of a block, i.e., for a block x and a bit b ,

$$\text{fixl}(x, b) := x[1..n-1]||b.$$

For $m \in [0..n-1]$ and $x \in \{0, 1\}^m$, $x||10^*$ denotes the block $x||1||0^{n-m-1}$. (We will sometimes use the term *incomplete block* to describe an m -bit string with $m \in [0..n-1]$.) $X \leftarrow x$ denotes the assignment of the value x to the variable X .

Sampling a Random Permutation. We say P is a *partially-determined permutation* if for two subsets $\text{dom}(P)$ and $\text{ran}(P)$ of $\{0, 1\}^n$ of equal size P is an injective function from $\text{dom}(P)$ to $\text{ran}(P)$. We take $|P| = |\text{dom}(P)| = |\text{ran}(P)|$. For a partially-determined permutation P and a pair (x, y) with

$$x \in \{0, 1\}^n \setminus \text{dom}(P), y \in \{0, 1\}^n \setminus \text{ran}(P),$$

we can *add* (x, y) to P , by extending the definition of P to include

$$P(x) := y.$$

Note that this adds x to $\text{dom}(P)$ and y to $\text{ran}(P)$. When sampling a *random permutation* P , queries to P or P^{-1} are answered while keeping track of the partially-determined P . For any *forward query* x (i.e., a query to P), if $x \in \text{dom}(P)$, $P(x)$ is returned; else a y is sampled uniformly from $\{0, 1\}^n \setminus \text{ran}(P)$ and returned, and (x, y) is added to P . Similarly, for any *backward query* y (i.e., a query to P^{-1}), if $y \in \text{ran}(P)$, $P^{-1}(y)$ is returned; else an x is sampled uniformly from $\{0, 1\}^n \setminus \text{dom}(P)$ and returned, and (x, y) is added to P . Thus, for any $x \notin \text{dom}(P)$ and $y \notin \text{ran}(P)$, the probability that a forward query x will return y or a backward query y will return x is $1/(2^n - |P|)$. As long as $|P| \leq 2^{n-1}$, we can use the simpler bound $1/2^{n-1}$.

Single-User PRF Game in the Public Permutation Model. Let $f_1[P] : S \rightarrow \{0, 1\}^m$ be a function which uses a permutation P as an underlying primitive (we assume all its components other than P and the secret key are publicly computable), and $f_0 : S \rightarrow \{0, 1\}^m$ be a random function. In the Single-User PRF Game in the Public Permutation Model, an adversary \mathcal{A} makes a series of forward and backward queries to P , called the *permutation queries* (or the *offline queries*), and a series of queries to an oracle \mathcal{O} , called the *construction queries* (or the *online queries*). \mathcal{O} is either the real oracle \mathcal{O}_1 , which returns $f_1[P](x)$ when x is queried; or it is the ideal oracle \mathcal{O}_0 , which returns $f_0(x)$ when x is queried. In the *post-query phase*, i.e., after all the permutation queries and construction queries have been answered, the oracle \mathcal{O} may decide to reveal certain additional information to \mathcal{A} . Finally, \mathcal{A} returns a bit b , and wins if $\mathcal{O} = \mathcal{O}_b$. Note that the permutation queries and responses are visible to \mathcal{O} , so we can assume that all queries are handled by \mathcal{O} itself. The advantage of \mathcal{A} against $f_1[P]$ when it makes D blocks of construction queries and T blocks of permutation queries is defined as

$$\mathbf{Adv}_{f_1[P]}^{\mathcal{A}}(D, T) := |\Pr_1[\mathcal{A} \text{ wins}] - \Pr_0[\mathcal{A} \text{ loses}]|,$$

where $\Pr_1[\cdot]$ denotes probability under the real oracle and $\Pr_0[\cdot]$ denotes probability under the ideal oracle. The (D, T) -PRF-advantage of $f_1[P]$ is defined as

$$\mathbf{Adv}_{f_1[P]}^{\text{PRF}}(D, T) := \max_{\mathcal{A}} \mathbf{Adv}_{f_1[P]}^{\mathcal{A}}(D, T).$$

The Multi-User Version of the Game. We consider the case where \mathcal{A} can access $f_1[P]$ as μ different users. Each user has an independently sampled secret key, and a unique public ID of variable length. The adversary's construction queries can specify an user index $u \in [\mu]$, specifying which user's key-ID pair to use for evaluating the call. The domain of the random function f_0 (that the ideal oracle uses) is $[\mu] \times S$ in this game. With D and T as before, the advantage of \mathcal{A} against $f_1[P]$ is defined identically as in the single-user version, except with the additional parameter μ in the notation. The (μ, D, T) -multi-user-PRF-advantage of $f_1[P]$ is defined as

$$\mathbf{Adv}_{f_1[P]}^{\text{MU-PRF}}(\mu, D, T) := \max_{\mathcal{A}} \mathbf{Adv}_{f_1[P]}^{\mathcal{A}}(\mu, D, T).$$

We will also call the Multi-User PRF game with the parameters μ, D, T as described above a (μ, D, T) -MU-PRF-game.

Coefficients H Technique. For bounding the MU-PRF-advantage of a function $f_1[P]$, we can use a result called the Coefficients H Technique. It is a proof method by Patarin [Pat09] that was modernized by Chen and Steinberger [CS14, CS13] and generalized by Hoang and Tessaro [HT16b] in their expectation method. Suppose the partially-determined P at the end of the query phase is revealed to the adversary by the real oracle. Note that this P contains the history of both the permutation queries and the calls to P or P^{-1} by the oracle itself while evaluating $f_1[P]$ at the construction queries. Since all the other parts of the construction calls are publicly computable, this partially-determined P includes complete information about the game. We will call it a *transcript* of the game and denote it as \tilde{P} . Suppose we can define a simulator for the ideal oracle, which can produce a valid transcript \tilde{P} which looks like it comes from the real oracle, unless it encounters certain *bad* events. If for some $\epsilon > 0$ for any (μ, D, T) -MU-PRF-game we can show that

$$\Pr_0 [\text{a bad event is encountered}] \leq \epsilon,$$

and that for a valid transcript \tilde{P} ,

$$\Pr_0 [\tilde{P}] \leq \Pr_1 [\tilde{P}],$$

then the Coefficients H Technique tells us that

$$\mathbf{Adv}_{f_1[P]}^{\text{MU-PRF}}(\mu, D, T) \leq \epsilon.$$

(Note that when we talk of the probability of a transcript \tilde{P} , what we really refer to is the probability that a game ends up with transcript \tilde{P} .) The original result of the Coefficients H Technique is slightly more general, but in this paper we will only be interested in the special case of it described above.

```

Module hashID
input :  $t - 1$  complete ID blocks  $\text{id}_1, \text{id}_2, \dots, \text{id}_{t-1}$  and one incomplete
        (possibly empty) ID block  $\text{id}_t^*$ 
output: hashed ID  $H$ 
begin
   $V \leftarrow 0$ 
  for  $j \leftarrow 1$  to  $t - 1$  do
     $U \leftarrow V \oplus \text{id}_j$ 
     $V \leftarrow P(U)$ 
  end for
   $U \leftarrow \text{fix1}(V \oplus \text{pad}(\text{id}_t^*), 1)$ 
   $H \leftarrow P(U)$ 
end

Module PRF
input : hashed ID  $H$ ,  $\kappa$ -bit user key  $K$ ,  $\ell - 1$  complete message blocks
         $M_1, M_2, \dots, M_{\ell-1}$  and one incomplete (possibly empty) message
        block  $M_\ell^*$ 
output:  $\tau$ -bit tag  $T$ 
begin
   $X \leftarrow K \parallel 0^{n-\kappa}$ 
   $Z \leftarrow P(X)$ 
   $X \leftarrow Z \oplus H$ 
   $Y \leftarrow P(X)$ 
  for  $j \leftarrow 1$  to  $\ell - 1$  do
     $X \leftarrow Y \oplus M_j$ 
     $Y \leftarrow P(X)$ 
  end for
   $X \leftarrow Y \oplus \text{pad}(M_\ell^*) \oplus Z$ 
   $Y \leftarrow P(X)$ 
   $T \leftarrow \text{chop}(Y)$ 
end

```

Algorithm 1: The algorithm for the Muffler $[P]$ construction. $\text{pad}(x)$ denotes $x \parallel 10^*$; $\text{chop}(x)$ denotes $x[1..\tau]$.

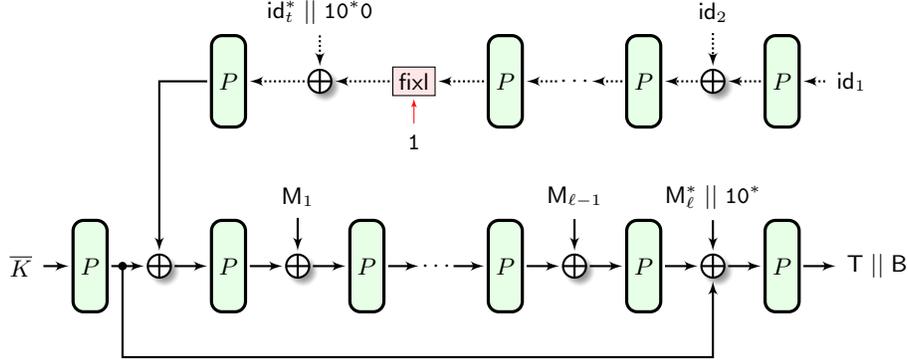


Fig. 1. The Muffler construction. P is a public random permutation; $\bar{K} := K || 0^{n-\kappa}$; id_1, \dots, id_ℓ^* is the ID corresponding to key K , where id_ℓ^* may have less than n bits (when id_ℓ^* has n or $n - 1$ bits, the final block being added after $fixl()$ in the figure is the padding block); M_1, \dots, M_ℓ^* is the message where M_ℓ^* may have less than n bits; tag T is obtained by chopping off the last $n - \tau$ bits of the final output. The dotted lines represent offline computations.

3 A PRF with Multi-User Security

The Muffler[P] Construction. Muffler[P] is a multi-user PRF based on a public random permutation P . Each user, in addition to having an independent random key of κ bits, for some $\kappa \in [n]$, also has a unique public ID. A user's ID is first hashed to a single block, which the adversary can compute by querying the public permutation, but cannot control. The key and hashed ID of an user are prefixed to each message queried to her, and the entire sequence of blocks is absorbed into a full-state sponge. The output of the first P call is squeezed out and fed-forward into the input of the last P call, and the top τ bits (for some $\tau \in [n]$) are squeezed out of the sponge at the end and released as the τ -bit output. The working of Muffler[P] is illustrated in Figure 1, and the algorithm is described in algorithm 1.

Notation for Security Game. In this game there are μ users $\mathcal{U}_1, \dots, \mathcal{U}_\mu$. We denote the key and ID for \mathcal{U}_u as $K^{(u)}$ and $id^{(u)}$ respectively. We let

$$H^{(u)} := \mathcal{H}[P](id^{(u)}),$$

where $\mathcal{H}[P]$ is the hash function to be defined shortly. The adversary \mathcal{A} makes T permutation queries. For $i \in [T]$, we say $i \in \mathcal{P}^f$ when the i -th permutation query is made to P , (i.e., it's a *forward* query); then the query is denoted U^i and the response is denoted V^i ; we say $i \in \mathcal{P}^b$ when the i -th permutation query is made to P^{-1} , (i.e., it's a *backward* query); then the query is denoted V^i and

the response is denoted U^i . Thus for each $i \in [T]$,

$$P(U^i) = V^i.$$

For $u \in [\mu]$, \mathcal{A} further makes q_u construction queries to \mathcal{U}_u , with

$$q := \sum_{u \in [\mu]} q_u;$$

for $i \in [q_u]$, the i -th query to \mathcal{U}_u is denoted $M^{(u,i)}$. There are $\ell^{(u,i)}$ blocks in $M^{(u,i)}$, with

$$D_u := \sum_{i \in [q_u]} \ell^{(u,i)};$$

for $j \in [\ell^{(u,i)}]$ the j -th block of $M^{(u,i)}$ is denote $M_j^{(u,i)}$, where $M_{\ell^{(u,i)}}^{(u,i)}$ just denotes the final incomplete (possibly empty) block of $M^{(u,i)}$ after 10^* padding. Finally, we let

$$D := \sum_{u \in [\mu]} D_u,$$

which denotes the total number of blocks queries to the construction. The total number of query blocks including construction queries and permutation queries is $D + T$. Let

$$\mathcal{Q} := \{(u, i) \mid u \in [\mu], i \in [q_u]\}$$

be the set of all construction query indices, and let

$$\mathcal{I} := \{(u, i, j) \mid (u, i) \in \mathcal{Q}, j \in [\ell^{(u,i)}]\}$$

be the set of all construction query block indices. In addition, we define a slightly different set of construction query block indices, to be useful in the subsequent analysis:

$$\mathcal{I}^\pm := \{(u, i, j) \mid (u, i) \in \mathcal{Q}, j \in [2 \cdot \ell^{(u,i)} + 1]\}.$$

Padding the IDs. We use a 10^*0 padding on the IDs to bring their length up to a multiple of n bits. This injective padding scheme ensures that the last bit is always 0, and the first bit following the un-padded ID is always 1, with a variable number of 0 bits inserted in between to adjust the length. (Note that when the final ID block is of length n bits or $n - 1$ bits, this padding scheme appends an entire padding block to the ID.)

Hashing the IDs. The hashing of the IDs consists of a series of chained calls to the permutation, with one bit tweaked before the final call. We assume that these calls are made as part of the forward permutation queries by the adversary. Let

$t^{(u)}$ be the number of blocks in $\text{id}^{(u)}$ after padding, and let the blocks be $\text{id}_1^{(u)}, \dots, \text{id}_{t^{(u)}}^{(u)}$. Then we assume for each $u \in [\mu]$ there are indices $i_1^{(u)}, \dots, i_{t^{(u)}}^{(u)} \in \mathcal{P}^f$ such that

$$\begin{aligned} \mathbf{U}^{i_1^{(u)}} &= \text{id}_1^{(u)}, \\ \mathbf{U}^{i_j^{(u)}} &= \mathbf{V}^{i_{j-1}^{(u)}} + \text{id}_j^{(u)} && \text{when } 2 \leq j \leq t^{(u)} - 1, \\ \mathbf{U}^{i_{t^{(u)}}^{(u)}} &= \text{fixl} \left(\mathbf{V}^{i_{t^{(u)}-1}^{(u)}} + \text{id}_{t^{(u)}}^{(u)}, \mathbf{1} \right). \end{aligned}$$

4 Main Security Result

Main Theorem. We now state the main result of this paper.

Theorem 1 (Security Bound). *Let \mathcal{A} be a PRF adversary, trying to differentiate a Muffler $[P](\kappa, \tau)$ construction for μ users from an ideal random function $f : [\mu] \times \{0, 1\}^{\geq n} \rightarrow \{0, 1\}^\tau$. Suppose \mathcal{A} makes q construction queries, consisting of D blocks in all, and T permutation queries, including forward and backward queries. Then, for integer parameters θ_1 and θ_2 ,*

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{PRF}} &\leq \frac{\theta_1 T}{2^{n-\tau}} + \frac{\theta_2 T}{2^\kappa} + \frac{2^\tau}{\theta_1!} \cdot \left(\frac{D}{2^\tau}\right)^{\theta_1} + \frac{2^n}{\theta_2!} \cdot \left(\frac{D^2}{2^n}\right)^{\theta_2} + \frac{\mu^2}{2^{2\kappa}} \\ &\quad + \frac{D^2 + 2DT + 2qD + 2\mu D + \mu^2 + 6\mu T}{2^n} + \frac{\mu^2 T + \mu^3}{2^{n+\kappa}}. \end{aligned}$$

Making Sense of the Bound. The bound above is rather complicated for taking in at a glance, so we now simplify it a bit by substituting certain typical parameter values we have in mind. First we assume that μ and q are of the same order as D , which gives the bound.

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{PRF}} &\leq \frac{\theta_1 T}{2^{n-\tau}} + \frac{\theta_2 T}{2^\kappa} + \frac{2^\tau}{\theta_1!} \cdot \left(\frac{D}{2^\tau}\right)^{\theta_1} + \frac{2^n}{\theta_2!} \cdot \left(\frac{D^2}{2^n}\right)^{\theta_2} \\ &\quad + \frac{D^2}{2^{2\kappa}} + \frac{6D^2 + 8DT}{2^n} + \frac{D^2 T + D^3}{2^{n+\kappa}}. \end{aligned}$$

Typical values for κ and τ would be about $n/2$ each, so substituting that gives us

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{PRF}} &\leq \frac{\theta_1 T + \theta_2 T}{2^{n/2}} + \frac{2^{n/2}}{\theta_1!} \cdot \left(\frac{D}{2^{n/2}}\right)^{\theta_1} + \frac{2^n}{\theta_2!} \cdot \left(\frac{D^2}{2^n}\right)^{\theta_2} \\ &\quad + \frac{7D^2 + 8DT}{2^n} + \frac{D^2 T + D^3}{2^{3n/2}}. \end{aligned}$$

We choose the parameters θ_1 and θ_2 large enough so that the coefficients $2^{n/2}/\theta_1!$ and $2^n/\theta_2!$ are small. For example, when $n = 128$, choosing $\theta_1 \geq 21$ and $\theta_2 \geq 34$

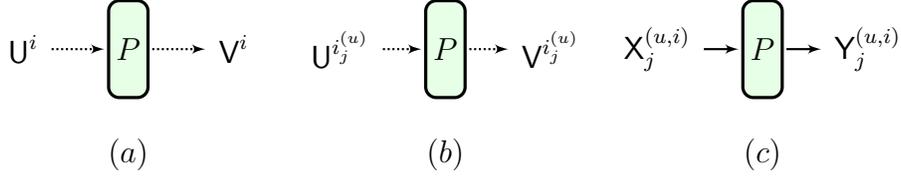


Fig. 2. Notation for real oracle computations. **(a)** i -th call to public permutation P ; **(b)** j -th (offline) call to P while hashing the ID of user u ; **(c)** j -th (online) call to P while processing i -th construction query to user u .

ensures that either coefficient is less than 1. For further simplification, we can choose $\theta_1 \approx 2n/3$ and $\theta_2 \approx n/3$ to get

$$\mathbf{Adv}_{\mathcal{A}}^{\text{PRF}} \leq \frac{nT}{2^{n/2}} + 2 \left(\frac{D}{2^{n/2}} \right)^{2n/3} + \frac{7D^2 + 8DT}{2^n} + \frac{D^2T + D^3}{2^{3n/2}}.$$

This shows that we can allow both D and T to go up to the birthday-bound in n without breaking the security of $\text{Muffler}[P]$. The only degradation is a factor of n in the first term of the simplified bound.

5 Proof of Security

Real Oracle. We begin by describing in detail the specific oracles we choose for this game. In the real oracle, the user key $K^{(u)}$ for each $u \in [\mu]$ is initially sampled uniformly from $\{0, 1\}^\kappa$. For each $i \in [q_u]$, we first set

$$\begin{aligned} X_0^{(u,i)} &= K^{(u)} \parallel 0^{n-\kappa}, \\ Y_0^{(u,i)} &= P\left(X_0^{(u,i)}\right). \end{aligned}$$

Next we incorporate the hashed user ID by setting

$$\begin{aligned} X_1^{(u,i)} &= Y_0^{(u,i)} + H^{(u)}, \\ Y_1^{(u,i)} &= P\left(X_1^{(u,i)}\right). \end{aligned}$$

Then comes the message blocks: for each $j \in [2..\ell^{(u,i)}]$ we set

$$\begin{aligned} X_j^{(u,i)} &= Y_{j-1}^{(u,i)} + M_{j-1}^{(u,i)}, \\ Y_j^{(u,i)} &= P\left(X_j^{(u,i)}\right). \end{aligned}$$

For the final call to P , we feed forward the output of the first call by setting

$$\begin{aligned} X_{\ell^{(u,i)}+1}^{(u,i)} &= Y_{\ell^{(u,i)}}^{(u,i)} + Y_0^{(u,i)} + M_{\ell^{(u,i)}}^{(u,i)}, \\ Y_{\ell^{(u,i)}+1}^{(u,i)} &= P\left(X_{\ell^{(u,i)}+1}^{(u,i)}\right). \end{aligned}$$

The output is

$$\mathsf{T}^{(u,i)} := \mathsf{Y}_{\ell^{(u,i)}+1}^{(u,i)}[1..\tau].$$

Additionally we denote

$$\mathsf{B}^{(u,i)} := \mathsf{Y}_{\ell^{(u,i)}+1}^{(u,i)}[\tau + 1..n].$$

$\mathsf{T}^{(u,i)}$ is returned immediately to the adversary at the end of the i -th query to \mathcal{U}_u . Note that all the P calls above are executed in the random permutation model. The partially determined P is revealed to the adversary at the end of the query phase.

Ideal Oracle. The sampling mechanism of the ideal oracle is described below. Certain bad events can be encountered during the sampling process. Once a bad event is encountered, the subsequent behaviour of the ideal oracle is left undefined. (One can for instance imagine that after encountering a bad event, the ideal oracle only outputs random bits for the rest of the game.)

- *Construction Queries [online]:* The queries are resolved in the random oracle model: for each $u \in [\mu]$, $i \in [q_u]$, we sample $\mathsf{Y}_{\ell^{(u,i)}+1}^{(u,i)}$ uniformly from $\{0, 1\}^n$, and define

$$\begin{aligned} \mathsf{T}^{(u,i)} &:= \mathsf{Y}_{\ell^{(u,i)}+1}^{(u,i)}[1..\tau], \\ \mathsf{B}^{(u,i)} &:= \mathsf{Y}_{\ell^{(u,i)}+1}^{(u,i)}[\tau + 1..n]. \end{aligned}$$

$\mathsf{T}^{(u,i)}$ is returned to the adversary in response to the i -th query to \mathcal{U}_u .

- **badCC** occurs if we can find θ_1 distinct pairs $(u_1, i_1), \dots, (u_{\theta_1}, i_{\theta_1}) \in \mathcal{Q}$ such that

$$\mathsf{T}^{(u_1, i_1)} = \dots = \mathsf{T}^{(u_{\theta_1}, i_{\theta_1})}.$$

- *Permutation Queries [offline]:* The offline queries to P or P^{-1} are resolved in the random permutation model, as described before. (These include the calls required for determining $\mathsf{H}^{(1)}, \dots, \mathsf{H}^{(\mu)}$.) This part is identical for the two oracles, and leaves us with a partially-determined P at the end of the query phase.

- **badCP** occurs if we can find $(u, i) \in \mathcal{Q}$, $i' \in [T]$ such that

$$\mathsf{Y}_{\ell^{(u,i)}+1}^{(u,i)} = \mathsf{V}^{i'}.$$

- **badPE** occurs if we can find θ_2 distinct pairs $(u_1, i_1), \dots, (u_{\theta_2}, i_{\theta_2}) \in \mathcal{Q} \in [\mu] \times \mathcal{P}^f$ such that

$$\mathsf{H}^{(u_1)} + \mathsf{V}^{i_1} = \dots = \mathsf{H}^{(u_{\theta_2})} + \mathsf{V}^{i_{\theta_2}}.$$

- *Internal Sampling:* A triple $(u, i, j) \in \mathcal{I}$ is called *fresh* if one of the following is true:

- $i = 1$;

- $i \geq 2, j = \ell^{(u,i)}$;
- $i \geq 2, j \in [2.. \ell^{(u,i)} - 1]$ and for each $i' \in [i - 1]$ with $\ell^{(u,i')} \geq j + 1$ we can find $j' \in [j - 1]$ such that $M_{j'}^{(u,i)} \neq M_{j'}^{(u,i')}$, i.e., the $(j - 1)$ -block prefix of the i -th query to user u was not a proper prefix of an earlier query to the same user.

Let \mathcal{F} denote the set of all fresh (u, i, j) triples. For $(u, i, j) \in \mathcal{I}$, the (u, j) -ancestor of i is the smallest i_0 such that for each $j' \in [j - 1]$, $M_{j'}^{(u,i)} = M_{j'}^{(u,i_0)}$. Note that this automatically implies that $(u, i_0, j) \in \mathcal{F}$. Now we describe the sampling order of the internal inputs and outputs of P , which is done at the end of the query phase:

Step 1: For each $(u, i, j) \in \mathcal{F}$, sample $Y_j^{(u,i)}$ uniformly from $\{0, 1\}^n$.

- badYC occurs if we can find $(u, i, j) \in \mathcal{F}, (u', i') \in \mathcal{Q}$ such that

$$Y_j^{(u,i)} = Y_{\ell^{(u',i')}}^{(u',i')};$$

- badYP occurs if we can find $(u, i, j) \in \mathcal{F}, i' \in [T]$ such that

$$Y_j^{(u,i)} = V^{i'};$$

- badYY occurs if we can find distinct $(u, i, j), (u', i', j') \in \mathcal{F}$ such that

$$Y_j^{(u,i)} = Y_{j'}^{(u',i')}.$$

Step 2: For each $(u, i, j) \in \mathcal{I} \setminus \mathcal{F}$, set

$$Y_j^{(u,i)} := Y_j^{(u,i_0)},$$

where i_0 is the (u, j) -ancestor of i .

Step 3: For each $(u, i, j) \in \mathcal{I}^\pm$, set

$$\begin{aligned} X_j^{(u,i)} &:= M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i)}, & j \leq \ell^{(u,i)}, \\ &:= M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i)} + Y_1^{(u,i)}, & j = \ell^{(u,i)} + 1. \end{aligned}$$

- badXP occurs if we can find $(u, i, j) \in \mathcal{I}^\pm, i' \in [T]$ such that

$$X_j^{(u,i)} = U^{i'};$$

- badXX occurs if we can find $(u, i, j), (u', i', j') \in \mathcal{I}^\pm$ such that

$$X_j^{(u,i)} = X_{j'}^{(u',i')}.$$

Step 4: For each $(u, i, j) \in \mathcal{I}^\pm$, add $(X_j^{(u,i)}, Y_j^{(u,i)})$ to P .

Step 5: For each $u \in [\mu]$ sample $K^{(u)}$ uniformly from $\{0, 1\}^\kappa$ and set $X_0^{(u,1)} := K^{(u)} \parallel 0^{n-\kappa}$.

Step 6: For each $u \in [\mu]$, if $X_0^{(u,1)} \in \text{dom}(P)$ set $Y_0^{(u,1)} := P(X_0^{(u,1)})$, else sample $Y_0^{(u,1)}$ uniformly from $\{0, 1\}^n \setminus \text{ran}(P)$ and add $(X_0^{(u,1)}, Y_0^{(u,1)})$ to P .

Step 7: For each $u \in [\mu]$ set

$$X_1^{(u,1)} := Y_0^{(u,1)} + H^{(u)}.$$

- **badHP** occurs if we can find $u \in [\mu], i \in [T]$ such that

$$X_1^{(u,1)} = U^i;$$

- **badHX** occurs if we can find $(u', i', j') \in \mathcal{I}^\pm$ such that

$$X_1^{(u,1)} = X_j^{(u',i)};$$

- **badHH** occurs if we can find distinct $u, u' \in [\mu]$ such that

$$X_1^{(u,1)} = X_1^{(u',1)}.$$

- **badHK** occurs if we can find $u, u' \in [\mu]$ such that

$$X_1^{(u,1)} = X_0^{(u',1)}.$$

Step 8: For each $u \in [\mu]$ add $(X_1^{(u,1)}, Y_1^{(u,1)})$ to P .

Step 9: For each $(u, i) \in \mathcal{Q}$ with $i > 1$ set

$$\begin{aligned} X_0^{(u,i)} &:= X_0^{(u,1)}, \\ Y_0^{(u,i)} &:= Y_0^{(u,1)}, \\ X_1^{(u,i)} &:= X_1^{(u,1)}. \end{aligned}$$

At the end of the internal sampling phase, the partially-determined P is revealed to the adversary. (Note that the last step of the ideal oracle does not affect the game, and is only included for the convenience of our analysis.)

Proof of Theorem. Let \tilde{P} denote the partially-revealed P at the end of the interaction of \mathcal{A} with the chosen oracle. When obtained from the real oracle, \tilde{P} contains all the probabilistic information of the game; when obtained from the ideal oracle, in the absence of bad events (which could result in unpredictable, inconsistent or incomplete transcripts), it also contains all the probabilistic information of the game. Let σ be the size of $\text{dom}(\tilde{P})$. Since all P -responses are sampled in the random permutation model in the real oracle,

$$\Pr_1 \left[\tilde{P} \right] = \frac{1}{2^n \cdot (2^n - 1) \cdot \dots \cdot (2^n - \sigma + 1)}.$$

In the ideal oracle, some of the P -responses are sampled in the random permu-

Fig. 3. Behaviour of Ideal Oracle

Query-Response Phase	Internal Sampling Phase
<p>Online</p> <p>for $(u, i) \in \mathcal{Q}$:</p> $Y_{\ell(u,i)+1}^{(u,i)} \xleftarrow{\mathbb{S}} \{0, 1\}^n$ $T^{(u,i)} \leftarrow Y_{\ell(u,i)+1}^{(u,i)}[1..\tau]$ $B^{(u,i)} \leftarrow Y_{\ell(u,i)+1}^{(u,i)}[\tau + 1..n]$ $T^{(u,i)} \rightarrow \mathcal{A}$ <p>check for badCC</p> <p>Offline</p> <p>for $i \in E_p$:</p> $V^i \xleftarrow{\mathbb{S}} \{0, 1\}^n \setminus \text{ran}(P)$ $P \leftarrow P \cup \{(U^i, V^i)\}$ $V^i \rightarrow \mathcal{A}$ <p>check for badCP, badPE</p> <p>for $i \in D_p$:</p> $U^i \xleftarrow{\mathbb{S}} \{0, 1\}^n \setminus \text{ran}(P)$ $P \leftarrow P \cup \{(U^i, V^i)\}$ $U^i \rightarrow \mathcal{A}$	<p>for $(u, i, j) \in \mathcal{F}$:</p> $Y_j^{(u,i)} \xleftarrow{\mathbb{S}} \{0, 1\}^n$ <p>check for badYC, badYP, badYY</p> <p>for $(u, i, j) \in \mathcal{I} \setminus \mathcal{F}$:</p> $i_0 \leftarrow (u, j)\text{-ancestor of } i$ $Y_j^{(u,i)} \leftarrow Y_j^{(u,i_0)}$ <p>for $(u, i, j) \in \mathcal{I}^-$:</p> $X_j^{(u,i)} \leftarrow M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i)}$ <p>for $(u, i) \in \mathcal{Q}$:</p> $X_{\ell(u,i)+1}^{(u,i)} \leftarrow M_{\ell(u,i)}^{(u,i)} + Y_{\ell(u,i)}^{(u,i)} + Y_1^{(u,i)}$ <p>check for badXP, badXX</p> <p>for $(u, i, j) \in \mathcal{I}^\pm$:</p> $P \leftarrow P \cup \{(X_j^{(u,i)}, Y_j^{(u,i)})\}$ <p>for $u \in [\mu]$:</p> $K^{(u)} \xleftarrow{\mathbb{S}} \{0, 1\}^\kappa$ $X_0^{(u,1)} \leftarrow K^{(u)} \parallel 0^{n-\kappa}$ <p>if $X_0^{(u,1)} \in \text{dom}(P)$:</p> $Y_0^{(u,1)} \leftarrow P(X_0^{(u,1)})$ <p>else :</p> $Y_0^{(u,1)} \xleftarrow{\mathbb{S}} \{0, 1\}^n \setminus \text{ran}(P)$ $P \leftarrow P \cup \{(X_0^{(u,1)}, Y_0^{(u,1)})\}$ <p>$X_1^{(u,1)} \leftarrow Y_0^{(u,1)} + H^{(u)}$</p> <p>check for badHP, badHX, badHH, badHK</p> $P \leftarrow P \cup \{(X_1^{(u,1)}, Y_1^{(u,1)})\}$ <p>$P \rightarrow \mathcal{A}$</p>

Table 1. Classification table of bad events.

Event	Definition	Range of Indices	Bound
badCC	θ_1 -collision in $\mathbf{T}^{(u,i)}$	$(u, i) \in \mathcal{Q}$	$(2^\tau / \theta_1!) \cdot (q/2^\tau)^{\theta_1}$
badCP	$Y_{\ell(u,i)+1}^{(u,i)} = V^{i'}$	$(u, i) \in \mathcal{Q}, i' \in [T]$	$\theta_1 T / 2^{n-\tau}$
badPE	θ_2 -collision in $\mathbf{H}^{(u)} + \mathbf{V}^i$	$u \in [\mu], i \in \mathcal{P}^f$	$(2^n / \theta_2!) \cdot (\mu D / 2^n)^{\theta_2}$
badYC	$Y_j^{(u,i)} = Y_{\ell(u',i')}^{(u',i')}$	$(u, i, j) \in \mathcal{F}, (u', i') \in \mathcal{Q}$	$qD / 2^n$
badYP	$Y_j^{(u,i)} = V^{i'}$	$(u, i, j) \in \mathcal{F}, i' \in [T]$	$DT / 2^n$
badYY	$Y_j^{(u,i)} = Y_{j'}^{(u',i')}$	$(u, i, j), (u', i', j') \in \mathcal{F}$	$D^2 / 2^{n+1}$
badXP	$X_j^{(u,i)} = U^{i'}$	$(u, i, j) \in \mathcal{I}^\pm, i' \in [T]$	$DT / 2^n$
badXX	$X_j^{(u,i)} = X_{j'}^{(u',i')}$	$(u, i, j), (u', i', j') \in \mathcal{I}^\pm$	$D^2 / 2^{n+1}$
badHP	$X_1^{(u,1)} = U^i$	$u \in [\mu], i \in [T]$	$\mu T / 2^{n-1} + \mu D / 2^n +$ $\mu^2 T / 2^{n+\kappa} + \theta_2 T / 2^\kappa$ $+ \mu T / 2^{n-1}$
badHX	$X_1^{(u,1)} = X_j^{(u',i')}$	$(u', i', j') \in \mathcal{I}^\pm$	$qD / 2^n$
badHH	$X_1^{(u,1)} = X_1^{(u',1)}$	$u, u' \in [\mu]$	$\mu^2 / 2^{n+1}$
badHK	$X_1^{(u,1)} = U^i$	$u \in [\mu], i \in [T]$	$\mu T / 2^n + \mu D / 2^n +$ $\mu^3 / 2^{n+\kappa} + \mu^2 / 2^{2\kappa}$

tation model, and some are sampled uniformly. Suppose σ_1 of the P -responses are sampled uniformly. Then

$$\Pr_0 \left[\tilde{P} \right] \leq \frac{1}{(2^n - \sigma_1) \cdot (2^n - \sigma_1 - 1) \cdot \dots \cdot (2^n - \sigma_1 + 1)} \cdot \left(\frac{1}{2^n} \right)^{\sigma_1}.$$

Thus, $\Pr_0 \left[\tilde{P} \right] \leq \Pr_1 \left[\tilde{P} \right]$, given that no bad event occurs. Coefficient H Technique tells us then that

$$\mathbf{Adv}_{\mathcal{A}}^{\text{PRF}} \leq \Pr_0 [\text{bad}],$$

where **bad** is the event that one of the twelve events **badCC**, **badCP**, **badPE**, **badYC**, **badYP**, **badYY**, **badXP**, **badXX**, **badHP**, **badHX**, **badHH** and **badHK** is encountered by the ideal oracle. If we rename them $\text{bad}_1, \dots, \text{bad}_{12}$, we have by the union-bound

$$\mathbf{Adv}_{\mathcal{A}}^{\text{PRF}} \leq \sum_{i=1}^{12} \Pr_0 [\text{bad}_i].$$

We shall show that the twelve probability terms on the right-hand side can be bounded as shown in Table 1. The bound in the theorem is obtained simply by adding them up.

6 Bounding the Bad Probabilities

Probability of badHP. Recall from Section 4 that badHP occurs if we can find $(u, i) \in \mathcal{Q}$ such that

$$X_1^{(u,1)} = U^i.$$

We can rewrite this as

$$Y_0^{(u,1)} + V_{i^{(u)}}^{i^{(u)}} = U^i.$$

Now we consider several cases:

Case 1: $X_0^{(u,1)} \notin \text{dom}(P)$. For fixed u , by the randomness of $Y_0^{(u,1)}$ the collision has a probability $\leq 1/2^{n-1}$. There are at most μ choices for u and at most T choices for i . Thus,

$$\Pr_0 [\text{badHP} : \text{Case 1}] \leq \frac{\mu T}{2^{n-1}}.$$

For each of the other cases, we need two simultaneous collisions, so we bound the joint probability by looking at one or the other or both, as needed.

Case 2: $X_0^{(u,1)} = X_{j'}^{(u',i')}$ for some $(u', i', j') \in \mathcal{I}^\pm$. We can rewrite this second collision equation as

$$X_0^{(u,1)} = M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i')}$$

when $j' \leq \ell^{(u',i')}$, and as

$$X_0^{(u,1)} = M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i')} + Y_0^{(u',i')}$$

when $j' = \ell^{(u',i')} + 1$. Fix u, u', i', j' , and let i'_0 be the $(u', j' - 1)$ -ancestor of i' . Then we can further rewrite this collision equation as

$$\begin{aligned} X_0^{(u,1)} &= M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i'_0)} \text{ or} \\ X_0^{(u,1)} &= M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i'_0)} + Y_0^{(u',i')}, \end{aligned}$$

depending on whether $j \leq \ell^{(u,i)}$ or $j = \ell^{(u,i)} + 1$. By the randomness of $Y_{j'-1}^{(u',i'_0)}$ this collision has a probability of $1/2^n$. There are μ choices for u and D choices for (u', i', j') . Thus,

$$\Pr_0 [\text{badHP} : \text{Case 2}] \leq \frac{\mu D}{2^n}.$$

Case 3: $X_0^{(u,1)} = X_0^{(u',1)}$ for some $u' < u$ such that $X_0^{(u',1)}$ was not in $\text{dom}(P)$ when it was sampled. Then we can rewrite the first collision equation as

$$Y_0^{(u',1)} + V_{t^{(u)}}^{i^{(u)}} = U^i.$$

For fixed u, u', i , by the randomness of $Y_0^{(u',1)}$ this collision has a probability $\leq 1/2^{n-1}$. We can rewrite the second collision equation as

$$K^{(u)} = K^{(u')}.$$

For fixed u, u' , this collision has a probability of $1/2^\kappa$. Thus the joint collision has a probability $\leq 1/2^{n+\kappa-1}$. There are at most $\binom{\mu}{2}$ choices for u, u' and T choices for i . Thus,

$$\Pr_0 [\text{badHP} : \text{Case 3}] \leq \frac{\binom{\mu}{2} \cdot T}{2^{n+\kappa-1}} \leq \frac{\mu^2 T}{2^{n+\kappa}}.$$

Case 4: $X_0^{(u,1)} = U^{i'}$ for some $i' \in \mathcal{P}^f$. Note that i' cannot be $i_{t^{(u)}}^{(u)}$, since $X_0^{(u,1)}$ and $U^{i_{t^{(u)}}^{(u)}}$ differ in the last bit. For fixed u, i' , by the randomness of $K^{(u)}$, this second collision has a probability of $1/2^\kappa$ if $U^{i'}[\kappa+1..n] = 0^{n-\kappa}$, and 0 otherwise. We can rewrite the first collision equation as

$$V^{i'} + V_{t^{(u)}}^{i^{(u)}} = U^i.$$

Since badPE has not occurred, for each choice of i , there are at most θ_2 choices of (u, i') , which makes the total number of choices for (u, i, i') at most $\theta_2 T$. Thus,

$$\Pr_0 [\text{badHP} : \text{Case 4}] \leq \frac{\theta_2 T}{2^\kappa}.$$

Case 5: $X_0^{(u,1)} = U^{i'}$ for some $i' \in \mathcal{P}^b$. For fixed u, i' , by the randomness of $U^{i'}$, this second collision has a probability $\leq 1/2^{n-1}$. As in the previous case, we can rewrite the collision equation as

$$V^{i'} + V_{t^{(u)}}^{i^{(u)}} = U^i.$$

Since choosing i and u fixes i' , there are at most μT choices for u, i, i' . Thus,

$$\Pr_0 [\text{badHP} : \text{Case 5}] \leq \frac{\mu T}{2^{n-1}}.$$

Taking union-bound over the four cases, we have

$$\Pr_0 [\text{badHP}] \leq \frac{\mu T}{2^{n-1}} + \frac{\mu D}{2^n} + \frac{\mu^2 T}{2^{n+\kappa}} + \frac{\theta_2 T}{2^\kappa} + \frac{\mu T}{2^{n-1}}.$$

Probability of badCC. Recall from Section 4 that **badCC** occurs if we can find θ_1 distinct pairs $(u_1, i_1), \dots, (u_{\theta_1}, i_{\theta_1}) \in \mathcal{Q}$ such that

$$\mathsf{T}^{(u_1, i_1)} = \dots = \mathsf{T}^{(u_{\theta_1}, i_{\theta_1})}.$$

For fixed $(u_1, i_1), \dots, (u_{\theta_1}, i_{\theta_1})$,

$$\Pr_0 \left[\mathsf{T}^{(u_1, i_1)} = \dots = \mathsf{T}^{(u_{\theta_1}, i_{\theta_1})} \right] = \frac{1}{2^{\tau(\theta_1-1)}}.$$

There are $\binom{q}{\theta_1}$ choices for $(u_1, i_1), \dots, (u_{\theta_1}, i_{\theta_1})$. Thus,

$$\Pr_0 [\mathbf{badCC}] = \frac{\binom{q}{\theta_1}}{2^{\tau(\theta_1-1)}} \leq \frac{q}{\theta_1!} \cdot \left(\frac{q}{2^\tau}\right)^{\theta_1-1}.$$

Probability of badCP. Recall from Section 4 that **badCP** occurs if we can find $u \in [\mu], i \in [q_u], i' \in [T]$ such that

$$\mathsf{Y}_{\ell^{(u, i)}+1}^{(u, i)} = \mathsf{V}^{i'}.$$

For this to happen, we need the collision

$$\mathsf{B}^{(u, i)} = \mathsf{V}^{i'}[\tau + 1..n]$$

with u, i, i' satisfying

$$\mathsf{T}^{(u, i)} = \mathsf{V}^{i'}[1..\tau].$$

By randomness of $\mathsf{B}^{(u, i)}$, the probability of the collision for fixed u, i, i' is $1/2^{n-\tau}$. Since **badCC** has not occurred, for each choice of i' , there are at most θ_1 choices for (u, i) . It follows that there are at most $\theta_1 T$ choices in all for u, i, i' . Thus,

$$\Pr_0 [\mathbf{badCC}] \leq \frac{\theta_1 T}{2^{n-\tau}}.$$

Probability of badPE. Recall from Section 4 that **badPE** occurs if we can find θ_2 distinct pairs $(u_1, i_1), \dots, (u_{\theta_2}, i_{\theta_2})$ in $[\mu] \times \mathcal{P}^f$ such that

$$\mathsf{H}^{(u_1)} + \mathsf{V}^{i_1} = \dots = \mathsf{H}^{(u_{\theta_2})} + \mathsf{V}^{i_{\theta_2}}.$$

For fixed $(u_1, i_1), \dots, (u_{\theta_2}, i_{\theta_2})$,

$$\Pr_0 \left[\mathsf{H}^{(u_1)} + \mathsf{V}^{i_1} = \dots = \mathsf{H}^{(u_{\theta_2})} + \mathsf{V}^{i_{\theta_2}} \right] = \left(\frac{1}{2^n}\right)^{\theta_2-1}.$$

There are $\binom{\mu D}{\theta_2}$ choices for $(u_1, i_1), \dots, (u_{\theta_2}, i_{\theta_2})$. Thus,

$$\Pr_0 [\mathbf{badPE}] = \frac{\binom{\mu D}{\theta_2}}{(2^n)^{\theta_2-1}} = \frac{\mu D}{\theta_2!} \cdot \frac{(\mu D - 1)^{\theta_2-1}}{(2^n)^{\theta_2-1}} \leq \frac{\mu D}{\theta_2!} \cdot \left(\frac{\mu D}{2^n}\right)^{\theta_2-1}.$$

Probability of badYC. Recall from Section 4 that **badYC** occurs if we can find $(u, i, j) \in \mathcal{F}, (u', i') \in \mathcal{Q}$ such that

$$Y_j^{(u,i)} = Y_{\ell(u',i')}^{(u',i')}.$$

For fixed u, i, j, i', j' , by the randomness of $Y_{\ell(u,i)+1}^{(u,i)}$, this collision has a probability of $1/2^n$. There are at most D choices for (u, i, j) and at most q choices for (u', i') . Thus,

$$\Pr_0 [\text{badYC}] \leq \frac{qD}{2^n}.$$

Probability of badYP. Recall from Section 4 that **badYP** occurs if we can find $(u, i, j) \in \mathcal{F}, i' \in [T]$ such that

$$Y_j^{(u,i)} = V^{i'}.$$

Fix u, i, j, i' , and let i_0 be the (u, j) -ancestor of i . Then we can rewrite the collision as

$$Y_j^{(u,i_0)} = V^{i'}.$$

By the randomness of $Y_j^{(u,i_0)}$, this collision has a probability of $1/2^n$. There are at most D choices for (u, i, j) and at most T choices for i' . Thus,

$$\Pr_0 [\text{badYP}] \leq \frac{DT}{2^n}.$$

Probability of badYY. Recall from Section 4 that **badYY** occurs if we can find distinct $(u, i, j), (u', i', j') \in \mathcal{F}$ such that

$$Y_j^{(u,i)} = Y_{j'}^{(u',i')}.$$

Fix u, i, j, u', i', j' , and let i_0 be the (u, j) -ancestor of i . Then we can rewrite the collision as

$$Y_j^{(u,i_0)} = Y_{j'}^{(u',i')}.$$

By the randomness of $Y_j^{(u,i_0)}$, this collision has a probability of $1/2^n$. There are at most $\binom{D}{2}$ choices for $(u, i, j), (u', i', j')$. Thus,

$$\Pr_0 [\text{badYY}] \leq \frac{\binom{D}{2}}{2^n} \leq \frac{D^2}{2^{n+1}}.$$

Probability of badXP. Recall from Section 4 that **badXP** occurs if we can find $(u, i, j) \in \mathcal{I}^\pm, i' \in [T]$ such that

$$X_j^{(u,i)} = U^{i'}.$$

We can rewrite this as

$$M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i)} = U^{i'}$$

when $j \leq \ell^{(u,i)}$, and as

$$M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i)} + Y_1^{(u,i)} = U^{i'}$$

when $j = \ell^{(u,i)} + 1$. Fix u, i, j, i' and let i_0 be the $(u, j - 1)$ -ancestor of i . Then we can rewrite the collision as

$$\begin{aligned} M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i_0)} &= U^{i'} \text{ or} \\ M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i_0)} + Y_1^{(u,i)} &= U^{i'} \end{aligned}$$

depending on whether $j \leq \ell^{(u,i)}$ or $j = \ell^{(u,i)} + 1$. In either case, by the randomness of $Y_{j-1}^{(u,i_0)}$, the probability of the collision is $1/2^n$. There are at most D choices for (u, i, j) and at most T choices for i' . Thus,

$$\Pr_0 [\text{badXP}] \leq \frac{DT}{2^n}.$$

Probability of badXX. Recall from Section 4 that **badXX** occurs if we can find distinct $(u, i, j), (u', i', j') \in \mathcal{I}^\pm$ such that

$$X_j^{(u,i)} = X_{j'}^{(u',i')}.$$

Again, we can rewrite this as

$$M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i)} = X_{j'}^{(u',i')}$$

when $j \leq \ell^{(u,i)}$, and as

$$M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i)} + Y_1^{(u,i)} = X_{j'}^{(u',i')}$$

when $j = \ell^{(u,i)} + 1$. Fix u, i, j, u', i', j' , and let i_0 be the $(u, j - 1)$ -ancestor of i . Then we can rewrite the collision as

$$\begin{aligned} M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i_0)} &= X_{j'}^{(u',i')} \text{ or} \\ M_{j-1}^{(u,i)} + Y_{j-1}^{(u,i_0)} + Y_1^{(u,i)} &= X_{j'}^{(u',i')} \end{aligned}$$

depending on whether $j \leq \ell^{(u,i)}$ or $j = \ell^{(u,i)} + 1$. In either case, by the randomness of $Y_{j-1}^{(u,i_0)}$, the probability of the collision is $1/2^n$. There are at most $\binom{D}{2}$ choices for $(u, i, j), (u', i', j')$. As in the case of **badYY**,

$$\Pr_0 [\text{badXX}] \leq \frac{D^2}{2^{n+1}}.$$

Probability of badHX. Recall from Section 4 that **badHX** occurs if we can find $u \in [\mu], (u', i, j) \in \mathcal{I}^\pm$ such that

$$X_1^{(u,1)} = X_j^{(u',i)}.$$

We can rewrite this as

$$\mathbf{X}_1^{(u,1)} = \mathbf{M}_{j-1}^{(u',i)} + \mathbf{Y}_{j-1}^{(u',i)}$$

when $j \leq \ell^{(u',i)}$, and as

$$\mathbf{X}_1^{(u,1)} = \mathbf{M}_{j-1}^{(u',i)} + \mathbf{Y}_{j-1}^{(u',i)} + \mathbf{Y}_1^{(u',i)}$$

when $j = \ell^{(u',i)} + 1$. Fix u, u', i, j , and let i_0 be the $(u', j-1)$ -ancestor of i . Then we can rewrite the collision as

$$\begin{aligned} \mathbf{X}_1^{(u,1)} &= \mathbf{M}_{j-1}^{(u',i)} + \mathbf{Y}_{j-1}^{(u',i_0)} \text{ or} \\ \mathbf{X}_1^{(u,1)} &= \mathbf{M}_{j-1}^{(u',i)} + \mathbf{Y}_{j-1}^{(u',i_0)} + \mathbf{Y}_1^{(u,i)} \end{aligned}$$

depending on whether $j \leq \ell^{(u',i)}$ or $j = \ell^{(u',i)}$. In either case, by the randomness of $\mathbf{Y}_{j-1}^{(u',i_0)}$, the probability of the collision is $1/2^n$. There are at most μ choices for u and at most D choices for (u', i', j') . Thus,

$$\Pr_0 [\text{badHX}] \leq \frac{qD}{2^n}.$$

Probability of badHH. Recall from Section 4 that **badHH** occurs if we can find distinct $u, u' \in [\mu]$ such that

$$\mathbf{X}_1^{(u,1)} = \mathbf{X}_1^{(u',1)}.$$

We can rewrite this as

$$\mathbf{Y}_0^{(u,1)} + \mathbf{V}_{i_t^{(u)}}^{i_t^{(u)}} = \mathbf{Y}_0^{(u',1)} + \mathbf{V}_{i_t^{(u')}}^{i_t^{(u')}}.$$

Fix u and u' . Without loss of generality, assume $i_t^{(u)} > i_t^{(u')}$. Since $i_t^{(u)} \in \mathcal{P}^f$, by the randomness of $\mathbf{V}_{i_t^{(u)}}^{i_t^{(u)}}$, the equation has a probability $\leq 1/2^{n-1}$. There are at most $\binom{\mu}{2}$ choices for u, u' . Thus,

$$\Pr_0 [\text{badHH}] \leq \frac{\mu^2}{2^n}.$$

Probability of badHK. Recall from Section 4 that **badHK** occurs if we can find $u \in [\mu], u' \in [\mu]$ such that

$$\mathbf{X}_1^{(u,1)} = \mathbf{X}_0^{(u',1)}.$$

We can rewrite this as

$$\mathbf{Y}_0^{(u,1)} + \mathbf{V}_{i_t^{(u)}}^{i_t^{(u)}} = \mathbf{X}_0^{(u',1)}.$$

Now we consider several cases:

Case 1: $X_0^{(u,1)} \notin \text{dom}(P)$. Then by the randomness of $Y_0^{(u,1)}$ the collision has a probability $\leq 1/2^{n-1}$. There are at most μ choices for u and at most T choices for i . Thus,

$$\Pr_0 [\text{badHK} : \text{Case 1}] \leq \frac{\mu T}{2^{n-1}}.$$

For each of the other cases, we need two simultaneous collisions, so we bound the joint probability by looking at one or the other or both, as needed.

Case 2: $X_0^{(u,1)} = X_{j'}^{(u',i')}$ for some $(u', i', j') \in \mathcal{I}^\pm$. We can rewrite this second collision equation as

$$X_0^{(u,1)} = M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i')}$$

when $j' \leq \ell(u', i')$, and as

$$X_0^{(u,1)} = M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i')} + Y_0^{(u',i')}$$

when $j' = \ell(u', i') + 1$. Fix u, u', i', j' , and let i'_0 be the $(u', j' - 1)$ -ancestor of i' . Then we can further rewrite this collision equation as

$$\begin{aligned} X_0^{(u,1)} &= M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i'_0)} \text{ or} \\ X_0^{(u,1)} &= M_{j'-1}^{(u',i')} + Y_{j'-1}^{(u',i'_0)} + Y_0^{(u',i')}, \end{aligned}$$

depending on whether $j \leq \ell(u, i)$ or $j = \ell(u, i) + 1$. By the randomness of $Y_{j'-1}^{(u',i'_0)}$ this collision has a probability of $1/2^n$. There are μ choices for u and D choices for (u', i', j') . Thus,

$$\Pr_0 [\text{badHK} : \text{Case 2}] \leq \frac{\mu D}{2^n}.$$

Case 3: $X_0^{(u,1)} = X_0^{(u'',1)}$ for some $u'' < u$ such that $X_0^{(u'',1)}$ was not in $\text{dom}(P)$ when it was sampled. Then we can rewrite the first collision equation as

$$Y_0^{(u'',1)} + V_{i_t^{(u)}}^{i_t^{(u)}} = X_0^{(u',1)}.$$

For fixed u, u', u'' , by the randomness of $Y_0^{(u'',1)}$ this collision has a probability $\leq 1/2^{n-1}$. We can rewrite the second collision equation as

$$K^{(u)} = K^{(u'')}.$$

For fixed u, u'' , this collision has a probability of $1/2^\kappa$. Thus the joint collision has a probability $\leq 1/2^{n+\kappa-1}$. There are at most $\binom{\mu}{2}$ choices for u, u'' and μ choices for u . Thus,

$$\Pr_0 [\text{badHK} : \text{Case 3}] \leq \frac{\binom{\mu}{2} \cdot \mu}{2^{n+\kappa-1}} \leq \frac{\mu^3}{2^{n+\kappa}}.$$

Case 4: $X_0^{(u,1)} = U^i$ for some $i \in [T]$. As in **badHP**, i' cannot be $i_{t(u)}^{(u)}$, since $X_0^{(u,1)}$ and $U^{i_{t(u)}^{(u)}}$ differ in the last bit. For fixed u, i , by the randomness of $K^{(u)}$, this second collision has a probability of $1/2^\kappa$ if $U^i[\kappa + 1..n] = 0^{n-\kappa}$, and 0 otherwise. We can rewrite the first collision equation as

$$V^i + V_{i_{t(u)}^{(u)}}^{i_{t(u)}^{(u)}} = X_0^{(u',1)}.$$

Again, for fixed u, i, u' , by the randomness of $K^{(u')}$, this collision has a probability of $1/2^\kappa$ if $(V^i + V_{i_{t(u)}^{(u)}}^{i_{t(u)}^{(u)}})[\kappa + 1..n] = 0^{n-\kappa}$, and 0 otherwise. Since choosing u and u' fixes i , there are at most μ^2 choices for u, u', i . Thus,

$$\Pr_0 [\text{badHK} : \text{Case 4}] \leq \frac{\mu^2}{2^{2\kappa}}.$$

Taking union-bound over the four cases, we have

$$\Pr_0 [\text{badHK}] \leq \frac{\mu T}{2^{n-1}} + \frac{\mu D}{2^n} + \frac{\mu^3}{2^{n+\kappa}} + \frac{\mu^2}{2^{2\kappa}}.$$

7 Conclusion and Future Works

This paper proposed a simple variant of the full-state absorption sponge PRF. This variant provides higher security than existing constructions which were birthday-bound in the capacity. In addition we consider the multi-user security which is stifled by the $T\mu/2^k$ bound (for all basic constructions). To get something close to k -bit security, nonce-based or randomized constructions have been considered in the past. In this paper we consider a completely different and more realistic approach, based on a user-id (which is assumed to be unique for each user). We use this approach to bypass the need to maintain nonce or generate random salts. Our construction also allows arbitrary lengths of ID. To the best of our knowledge, it is the first deterministic stateless construction to achieve k -bit security in the multi-user security game. There are some natural follow up research questions which could be studied in future:

1. We prove the security in the known ID model in which user-IDs are given to an adversary. Can we consider a stronger adversary in which adversary can choose an ID adaptively (chosen-ID model)?
2. For the sake of simplicity of proof, we keep tag output to be at most $n/2$. Can we consider multiple squeezing phase to generate larger tag output (without degrading the security)?
3. Is it possible to analyse the security of this construction eliminating the multicollision factor? This would help us to obtain some matching attack. In other words, can we prove that our bound is tight?
4. Can we find another efficient design which can solve all above problems and may give better security results?

References

- ADMVA15. Elena Andreeva, Joan Daemen, Bart Mennink, and Gilles Van Assche. Security of keyed sponge constructions using a modular proof approach. In *International Workshop on Fast Software Encryption*, pages 364–384. Springer, 2015.
- BBM00. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 259–274. Springer, 2000.
- BBT16. Mihir Bellare, Daniel J. Bernstein, and Stefano Tessaro. Hash-function based prfs: AMAC and its multi-user security. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016, Proceedings, Part I*, volume 9665 of *LNCS*, pages 566–595. Springer, 2016.
- BCK96. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 514–523. IEEE, 1996.
- BDPVA07. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT hash workshop*, volume 2007. Citeseer, 2007.
- BDPVA08. Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. On the indistinguishability of the sponge construction. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 181–197. Springer, 2008.
- BDPVA11a. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: single-pass authenticated encryption and other applications. In *International Workshop on Selected Areas in Cryptography*, pages 320–337. Springer, 2011.
- BDPVA11b. Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. On the security of the keyed sponge construction. In *Symmetric Key Encryption Workshop*, volume 2011, 2011.
- Ber05. Daniel J. Bernstein. The poly1305-aes message-authentication code. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21–23, 2005, Revised Selected Papers*, volume 3557 of *LNCS*, pages 32–49. Springer, 2005.
- BHT18. Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting aes-gcm-siv: Multi-user security, faster key derivation, and better bounds. 2018.
- Bih02. Eli Biham. How to decrypt or even substitute des-encrypted messages in 228 steps. *Information Processing Letters*, 84(3):117–124, 2002.
- BKR98. Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-rackoff backwards: Increasing security by making block ciphers non-invertible. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, Proceedings*, volume 1403 of *LNCS*, pages 266–280. Springer, 1998.
- BMS05. Alex Biryukov, Sourav Mukhopadhyay, and Palash Sarkar. Improved time-memory trade-offs with multiple data. In *International Workshop on Selected Areas in Cryptography*, pages 110–127. Springer, 2005.

- BN21. Srimanta Bhattacharya and Mridul Nandi. Luby-rackoff backwards with more users and more security. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021, Proceedings, Part III*, volume 13092 of *LNCS*, pages 345–375. Springer, 2021.
- BT16. Mihir Bellare and Björn Tackmann. The multi-user security of authenticated encryption: Aes-gcm in tls 1.3. In *Annual Cryptology Conference*, pages 247–276. Springer, 2016.
- CMS11a. Sanjit Chatterjee, Alfred Menezes, and Palash Sarkar. Another look at tightness. In *International Workshop on Selected Areas in Cryptography*, pages 293–319. Springer, 2011.
- CMS11b. Sanjit Chatterjee, Alfred Menezes, and Palash Sarkar. Another look at tightness. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *LNCS*, pages 293–319. Springer, 2011.
- CS13. Shan Chen and John Steinberger. Tight security bounds for key-alternating ciphers. Cryptology ePrint Archive, Report 2013/222, 2013. <https://ia.cr/2013/222>.
- CS14. Shan Chen and John Steinberger. Tight security bounds for key-alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014*, pages 327–350, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- DDNT22. Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Tight multi-user security bound of DbHtS. Cryptology ePrint Archive, Paper 2022/689, 2022. <https://eprint.iacr.org/2022/689>.
- DMA17. Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-state keyed duplex with built-in multi-user support. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017, Proceedings, Part II*, volume 10625 of *LNCS*, pages 606–637. Springer, 2017.
- GW22. Tingting Guo and Peng Wang. A note on the security framework of two-key dbhts macs. Cryptology ePrint Archive, Paper 2022/375, 2022. <https://eprint.iacr.org/2022/375>.
- GWLZ17. Zhiyuan Guo, Wenling Wu, Renzhang Liu, and Liting Zhang. Multi-key analysis of tweakable even-mansour with applications to minalpher and opp. *IACR Transactions on Symmetric Cryptology*, 2016(2):288–306, 2017.
- HS05. Jin Hong and Palash Sarkar. New applications of time memory data tradeoffs. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 353–372. Springer, 2005.
- HT16a. Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: exact bounds and multi-user security. In *Annual Cryptology Conference*, pages 3–32. Springer, 2016.
- HT16b. Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016*, pages 3–32, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
- HT17. Viet Tung Hoang and Stefano Tessaro. The multi-user security of double encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 381–411. Springer, 2017.

- LMP17. Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017, Proceedings, Part II*, volume 10625 of *LNCS*, pages 575–605. Springer, 2017.
- ML15. Nicky Mouha and Atul Luykx. Multi-key security: The even-mansour construction revisited. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015, Proceedings, Part I*, volume 9215 of *LNCS*, pages 209–223. Springer, 2015.
- MPS20. Andrew Morgan, Rafael Pass, and Elaine Shi. On the adaptive security of macs and prfs. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020, Proceedings, Part I*, volume 12491 of *LNCS*, pages 724–753. Springer, 2020.
- Pat09. Jacques Patarin. The “coefficients h” technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography*, pages 328–345, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- SWGW21. Yaobin Shen, Lei Wang, Dawu Gu, and Jian Weng. Revisiting the security of dbhts macs: Beyond-birthday-bound in the multi-user setting. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021, Proceedings, Part III*, volume 12827 of *LNCS*, pages 309–336. Springer, 2021.