Tight Multi-User Security Bound of DbHtS (Long Paper)

Anonymous Submission

Abstract. In CRYPTO'21, Shen et al. have proved in the ideal cipher model that Two-Keyed-DbHtS construction is secure up to $2^{2n/3}$ queries in the multi-user setting independent of the number of users, where the underlying double-block hash function H of the Two-Keyed-DbHtS construction is realized as the concatenation of two independent n-bit keyed hash functions $(\mathsf{H}_{K_h,1}, \mathsf{H}_{K_h,2})$ such that each of the n-bit keyed hash function is $O(2^{-n})$ universal and regular. They have also demonstrated the applicability of their result to the key-reduced variants of DbHtS MACs, including 2K-SUM-ECBC, 2K-PMAC_Plus and 2K-LightMAC_Plus without requiring domain separation technique and proved $2n/3$-bit multi-user security of these constructions in the ideal cipher model. Recently, Guo and Wang have invalidated the security claim of Shen et al.'s result by exhibiting three constructions, which are the instantiations of the Two-Keyed-DbHtS framework, such that each of their n-bit keyed hash functions being $O(2^{-n})$ universal and regular, while the constructions themselves are secure only up to the birthday bound. In this work, we show a sufficient condition on the underlying Double-block Hash (DbH) function, under which we prove $3n/4$-bit multi-user security of the Two-Keyed-DbHtS construction in the ideal-cipher model. As an instantiation, we show that two-keyed Polyhash-based DbHtS construction is multi-user secure up to $2^{3n/4}$ queries in the ideal-cipher model. Furthermore, due to the generic attack on DbHtS constructions by Gaëtan et al. in CRYPTO'18, our derived bound for the construction is tight.

Keywords: DbHtS · PRF · Polyhash · H-Coefficient Technique · Mirror Theory.

1 Introduction

Hash-then-PRF [33] (or HtP) is a well-known paradigm for designing variable input-length PRFs, in which an input message of arbitrary length is hashed and the hash value is encrypted through a PRF to obtain a short tag. Most popular MACs including the CBC-MAC [3], PMAC [9], OMAC [19] and LightMAC [23] are designed using the HtP paradigm. Although the method is simple, in particular being deterministic and stateless, the security of MACs following the HtP paradigm is capped at the birthday bound due to the collision probability of the hash function. Birthday bound-secure constructions are acceptable in practice when any of these MACs are instantiated with a block cipher of moderately large block size. For example, instantiating PMAC with AES-128 permits roughly 2^{48} queries (using $5\ell q^2/2^n$ [30] bound) when the longest message size is 2^{16} blocks, and the success probability of breaking the scheme is restricted to 2^{-10}. However, the same construction becomes vulnerable if instantiated with some lightweight (smaller block size) block ciphers, whose number has grown tremendously in recent years, e.g. PRESENT [10], GIFT [1], LED [15], etc. For example, PMAC, when instantiated with the PRESENT block cipher (a 64-bit block cipher), permits only about 2^{16} queries when the longest message size is 2^{16} blocks, and the probability of breaking the scheme is 2^{-10}. Therefore, it becomes risky to use birthday bound-secure constructions instantiated with lightweight block ciphers. In fact, in a large number of financial sectors, web browsers still widely use 64-bit block ciphers 3-DES instead of AES in their legacy applications with backward compatibility feature, as using the latter in corporate mainframe computers is more expensive. However, it does not give adequate security if the mode in which 3-DES is used provides only birthday bound security, and hence a beyond birthday secure mode solves the issue. Although many secure practical applications use the standard AES-128, which provides 64-bit security in a birthday bound-secure mode, which is adequate for the current technology, it may not remain so in the near future. In such a situation, using a mode with beyond the birthday bound security instead of replacing the cipher with a larger block size is a better option. [1]

Double-Block Hash-then-Sum. Many studies tried to tweak the HtP design paradigm to obtain beyond the birthday bound secure MACs; while they possess a similar structural design, the internal state of the hash function is doubled and the two n-bit hash values are first encrypted and then xored together to produce the output. In [35], Yasuda proposed a beyond the birthday bound secure deterministic MAC called SUM-ECBC, a rate-1/2 sequential mode of construction with four block cipher keys. Followed by this work, Yasuda [36] came up with another deterministic MAC called PMAC_Plus, but unlike SUM-ECBC, PMAC_Plus is a rate-1 parallel mode of construction with three block cipher keys. Zhang et al. [37] proposed another rate-1 beyond the birthday bound secure deterministic MAC called 3kf9 with three block cipher keys. In [29], Naito proposed LightMAC_Plus, a rate $(1 - s/n)$ parallel mode of operation, where s is the size of the block counter. The structural design of all these constructions first applies a $2n$-bit hash function on the message, then the two n-bit output values are encrypted and xored together to produce the tag, where n is the block size of the block cipher. Moreover, all of them also give $2n/3$-bit security. In FSE 2019, Datta et al. [13] proposed a generic design paradigm dubbed as the *double-block hash-then-sum* or DbHtS, defined as follows:

$$\mathsf{DbHtS}(M) \triangleq \mathsf{E}_{K_1}(\Sigma) \oplus \mathsf{E}_{K_2}(\Theta), \quad (\Sigma, \Theta) \leftarrow \mathsf{H}_{K_h}(M),$$

where H_{K_h} is a double-block hash function that maps an arbitrary-length string to a $2n$-bit string. Within this unified framework, they revisited the security proof of existing DbHtS constructions, including PolyMAC [20], SUM-ECBC [35], PMAC_Plus [36], 3kf9 [37] and LightMAC_Plus [29] and also their two-keyed versions [13] and confirmed that all the constructions are secure up to $2^{2n/3}$ queries when they are instantiated with an n-bit block cipher.

In CRYPTO 2018, Gaëtan et al. [21] proposed a generic attack on all these constructions using $2^{3n/4}$ (short message) queries, leaving a gap between the upper and the lower bounds for the provable security of DbHtS constructions. Recently, Kim et al. [20] have improved the bound of DbHtS constructions from $2^{2n/3}$ to $2^{3n/4}$. They have shown that if the underlying $2n$-bit hash function is the concatenation of two independent n-bit-universal hash functions [2], then the resulting DbHtS paradigm is secure up to $2^{3n/4}$ queries. They have also improved the security bound of PMAC_Plus, 3kf9 and LightMAC_Plus from $2^{2n/3}$ to $2^{3n/4}$ and hence closed the gap between the upper and the lower bounds of the provable security of DbHtS constructions.

Multi-user security of DbHtS. We have so far discussed the security bounds of DbHtS constructions in which adversaries are given access to some keyed oracles for a single unknown randomly sampled key. Such a model is known as the *single-user security model*, i.e. when the adversary interacts with one specific machine in which the cryptographic algorithm is deployed and tries to compromise its security. However, in practice, cryptographic algorithms are usually deployed in more than one machine. For example, AES-GCM [24, 25] is now widely used in the TLS protocol to protect web traffic and is currently used by billions of users daily. Thus, the security of DbHtS constructions

[1] Note that there are no standard block ciphers of size higher than 128 bits.

[2] A family of keyed hash functions is said to be universal if for any distinct x and x', the probability of a collision in their hash values for a randomly sampled hash function from the family is negligible.

in the *multi-key setting* is worth investigating; in other words, we ask *to what extent the number of users will affect the security of* DbHtS *constructions*, where adversaries are successful if they compromise the security of one out of many users. That means the adversary's winning condition is a disjunction of single-key winning conditions.

The notion of multi-user (mu) security was introduced by Biham [7] in symmetric cryptanalysis and by Bellare, Boldyreva, and Micali [2] in the context of public-key encryption. In the multi-user setting, attackers have access to multiple machines such that a particular cryptographic algorithm F is deployed in each machine with independent secret keys. An attacker can adaptively distribute its queries across multiple machines with independent keys. Multi-user security considers attackers that succeed in compromising the security of at least one machine, among others.

Multi-user security for block ciphers is different from multi-user security for modes. In the single-key setting, the best attacks against block cipher such as AES do not improve with increased data complexity. However, in the multi-key environment, they do, as first observed by Biham [7] and later refined as a time-memory-data trade-off by Biryukov et al. [8]. These results 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.[12]. A more general result guarantees that the *multi-user advantage of an adversary for a cryptographic algorithm is at most u times its single user advantage.* Therefore, for any cryptographic algorithm, a multi-user security bound involving a factor u is easily established using a hybrid argument that shows the upper bound of the adversarial success probability to be roughly u times its single-user security advantage. Bellare and Tackmann [5] first formalized a multi-user secure authenticated encryption scheme and also analyzed countermeasures against multi-key attacks in the context of TLS 1.3. However, they derived a security bound that also contained the factor u. Such a bound implies a significant security drop of the construction when the number of users is large, and in fact, this is precisely the situation faced in large-scale deployments of AES-GCM such as TLS.

As evident from [4, 5, 11, 17, 18, 22, 28], it is a challenging problem to study the security degradation of cryptographic primitives with the number of users, even when its security is known in the single-user setting. Studies of multi-user security of MACs are somewhat scarce in the literature except for the work of Chatterjee et al. [12], and a very recent work of Andrew et al. [27], and Bellare et al. [4]. The first two consider a generic reduction for MACs, in which the security of the primitive in the multi-user setting is derived by multiplying the number of users u by the single-user security.

In CRYPTO'21, Shen et al. [32] have analyzed the security of DbHtS in the multi-user setting. It is worth noting here that by applying the generic reduction from the single-user to the multi-user setting, the security bound of DbHtS would have capped at worse than the birthday bound, i.e. $uq^{4/3}/2^n$, when each user made a single query and the number of users reached q. Thus, a direct analysis was needed for deriving the multi-user bound of the construction. Shen et al. [32] have shown that in the multi-user setting, the two-keyed [3] DbHtS paradigm,

$$\mathsf{Two\text{-}Keyed\text{-}DbHtS}(M) \triangleq \mathsf{E}_K(\mathsf{H}_{K_h,1}(M)) \oplus \mathsf{E}_K(\mathsf{H}_{K_h,2}(M)),$$

is secure up to $2^{2n/3}$ queries in the ideal-cipher model when the $2n$-bit double-block hash function is the concatenation of two independent n-bit keyed hash functions $\mathsf{H}_{K_h,1}$ and $\mathsf{H}_{K_h,2}$. In particular, they have shown that if both $\mathsf{H}_{K_h,1}$ and $\mathsf{H}_{K_h,2}$ are $O(2^{-n})$-regular

[3] two-keyed stands for one hash key and one block cipher key.

and $O(2^{-n})$-universal [4], then the multi-user security bound of the two-keyed DbHtS is of the order of

$$\frac{qp\ell}{2k+n} + \frac{q^3}{2^{2n}} + \frac{q^2p + qp^2}{2^{2k}},$$

where q is the total number of MAC queries across all u users, p is the total number of ideal-cipher queries, ℓ is the maximum number of message blocks among all queries and n, k are the block size and the key size of the block cipher respectively. Note that the above bound is independent of the number of users u, which can be adaptively chosen by the adversary and grows as large as q. Besides this result, Shen et al. have also shown that 2K-SUM-ECBC [13], 2K-PMAC_Plus [13] and 2K-LightMAC_Plus [13] are all secure roughly up to $2^{2n/3}$ queries (including all MAC and ideal-cipher queries) in the multi-user setting independent of the number of users, where these constructions do not employ domain separation techniques.

Remark 1. In their paper [13], Datta et al. named the two-keyed variants of SUM-ECBC, PMAC_Plus and LightMAC_Plus as 2K-SUM-ECBC, 2K-PMAC_Plus and 2K-LightMAC_Plus respectively, where for each of these constructions, the domain separation technique ensured disjointness of the set of values of Σ and Θ. However, in [32], Shen et al. considered the same constructions but without any domain separation, and refer to them using the same names. Henceforth, we shall implicitly mean the non domain-separated variants only (unless otherwise stated) when referring to the two-keyed constructions 2K-SUM-ECBC, 2K-PMAC_Plus and 2K-LightMAC_Plus.

1.1 Issue with the CRYPTO'21 Paper [32]

In this section, we discuss three issues with [32]. The first two issues examine flaws in the security analysis of the construction and the last issue points out a flawed security claim of the construction. We begin by identifying the first issue. The Two-Keyed-DbHtS framework was proven to be multi-user secure up to $2^{2n/3}$ queries in the ideal-cipher model [32] under the assumption that each of the underlying n-bit independent keyed hash functions is $O(2^{-n})$-universal and regular. As an instantiation of the framework, [32] showed $2n/3$-bit multi-user security of 2K-SUM-ECBC, 2K-LightMAC_Plus and 2K-PMAC_Plus in the ideal-cipher model. In the secuirty proof of these instantiated constructions, they only bounded the regular and the universal advantages of the corresponding hash functions (i.e., the DbH of 2K-SUM-ECBC, 2K-LightMAC_Plus and 2K-PMAC_Plus) up to $O(\ell/2^n)$, where ℓ is the maximum number of message blocks amongst all queries. However, the regular and universal advantages of the underlying double block hash functions of the above three constructions were not proven in the ideal-cipher model; instead, the authors bounded them in the standard model, where the adversary is not allowed to query the underlying block ciphers of the corresponding hash functions. In other words, considering the example of 2K-LightMAC_Plus, while bounding the probability of the event $\Sigma_i = \Sigma_j$ (where $\Sigma_i = \Sigma_j \Rightarrow Y_1^i \oplus Y_2^i \oplus \ldots \oplus Y_{\ell_i}^i = Y_1^j \oplus Y_2^j \oplus \ldots \oplus Y_{\ell_j}^j$ and $Y_a^i = \mathsf{E}_K(M_a^i \| \langle a \rangle_s)$), the authors have simply assumed that at least one of variables Y in the above equation will be fresh, thus providing sufficient entropy for bounding the event. However, the authors have miserably missed the fact that existence of such a variable Y may not always be guaranteed in the ideal-cipher model. For example, suppose an adversary makes the following three forward primitive queries:

1. forward query with $(x \| \langle 1 \rangle_s)$ and obtains y_1

[4] A family of keyed hash function is said to be ϵ_1-regular if for any x and y, the probability that a randomly sampled hash function from the family maps x to y is ϵ_1; it is said to be ϵ_2-universal if for any distinct x, x', the probability that a randomly sampled hash function from the family yields a collision on the pair (x, x') is ϵ_2.

2. forward query with $(x'\|\langle 1\rangle_s)$ and obtains y_2
3. forward query with $(x''\|\langle 2\rangle_s)$ and obtains y_3

Let us assume that the (albeit probabilistic) event $y_1 \oplus y_2 \oplus y_3 = 0$ occurs. Suppose the adversary makes two more queries: the first, a construction query with (x) and the second, a construction query with $(x'\|x'')$. Then, one cannot find any fresh variable Y in the following equations:

$$Y_1^1 = Y_1^2 \oplus Y_2^2.$$

Therefore, to prove the security of such block cipher-based DbHtS constructions in the ideal-cipher model, one needs to consider the fact that the regular or universal advantage of the underlying double block hash functions must be bounded under the assumption that the adversary makes primitive queries to the underlying block cipher. We therefore believe that to prove the security of the constructions in the ideal-cipher model for the block cipher-based DbH function, one needs to provide a generalized definition of the universal and regular advantages in the ideal-cipher model and prove their security under this model, which was missing in [32].

The second issue is regarding the good transcript analysis of the Two-Keyed-DbHtS construction. In Fig. 4 of [32], the authors have identified the set of $(i,a) \in [u] \times [q_i]$, which they denoted as $F(J)$, such that both Σ_a^i and Θ_a^i are freshThey have also defined a set $S(J)$,

$$S(J) := \{(W_a^i, X_a^i) \in \{0,1\}^n \setminus \mathsf{Ran}(\Phi_j)^{(2|F(J)|)} : W_a^i \oplus X_a^i = T_a^i\}.$$

Then for all $(i,a) \in F(J)$, (U_a^i, V_a^i) is sampled from $S(J)$ and is set as the permutation output of Σ_a^i and Θ_a^i, respectively. Finally, they have provided a lower bound on the cardinality of the set $S(J)$ from Lemma 2. Noting that Lemma 2 proves the cardinality of the set

$$S := \{(U_i, V_i) \in (\{0,1\}^n)^{(2q)} : U_i \oplus V_i = T_i\}$$

to be at least $2^n(2^n-1)\ldots(2^n-2q+1)/2^{nq} \cdot (1 - 6q^3/2^{2n})$, which is used to obtain a lower bound on $|S(J)|$, reveals a fallacy as the two sets S and $S(J)$ are not isomorphic to each other.

The third issue is regarding the flawed security claim of the Two-Keyed-DbHtS construction in [32]. In Theorem 1 of [32], Shen et al. show that when the underlying double block hash function of the Two-Keyed-DbHtS construction is the concatenation of two independent n-bit keyed hash functions such that each is $O(2^{-n})$-universal and $O(2^{-n})$-regular, Two-Keyed-DbHtS achieves $2n/3$-bit multi-user security in the ideal-cipher model. In a recent work by Guo and Wang [16], the authors came up with three concrete constructions that are instantiations of the Two-Keyed-DbHtS paradigm such that the underlying double block hash function of each of the three constructions is the concatenation of two independent n-bit keyed hash functions. Guo and Wang also show that each of the n-bit hash functions for these three constructions meets the $O(2^{-n})$-universal and $O(2^{-n})$-regular advantages. However, the constructions have a birthday bound distinguishing attack. As a consequence, the security bound of Two-Keyed-DbHtS as proven in Theorem 1 of [32] stands flawed. We would like to mention here that the attack holds only for those instances of Two-Keyed-DbHtS where the underlying DbH is the concatenation of two independent n-bit hash functions and it does not have any domain separation. In fact, authors of [16] were not able to show any birthday bound attack on 2K-PMAC_Plus and 2K-LightMAC_Plus as the underlying DbH function of these two constructions is not the concatenation of two independent n-bit keyed hash functions. However, it is to be noted that as the double block hash function for 2K-SUM-ECBC is the concatenation of two independent n-bit CBC functions, the attack of [16] holds for it.

1.2 Our Contribution

In this paper we prove that the Two-Keyed-DbHtS construction is multi-user secure up to $2^{3n/4}$ queries in the ideal-cipher model. To prove it, we first define the notion of a **good** double-block hash function, which informally means that the concatenation of two independent n-bit keyed hash functions is "good" if each has negligible universal and regular advantages, and the probability that the outputs of two hash function colliding for any pair of messages M, M' is zero. Then, we prove that if the underlying $2n$-bit DbH function of the Two-Keyed-DbHtS construction is *good*, such that each of the n-bit keyed hash functions is ϵ_{reg}-regular and ϵ_{univ}-universal, then the multi-user security of our construction in the ideal-cipher model is of the order

$$\frac{9q^{4/3}}{8\cdot 2^n}+\frac{3q^{8/3}}{2\cdot 2^{2n}}+\frac{q^2}{2^{2n}}+\frac{9q^{7/3}}{8\cdot 2^{2n}}+\frac{8q^4}{3\cdot 2^{3n}}+\frac{q}{2^n}+\frac{2u^2}{2^{k_h+k}}+\frac{2q^2}{2^{n+k}}$$
$$+\frac{2qp\epsilon_{\text{reg}}}{2^k}+\frac{q^2\epsilon_{\text{univ}}}{2^n}+\frac{2q^2\epsilon_{\text{reg}}}{2^{k_h}}+3q^{4/3}\epsilon_{\text{univ}}+\frac{q^2\epsilon_{\text{univ}}^2}{2}+\frac{2qp}{2^{n+k}},$$

where q is the total number of MAC queries across all u users, p is the total number of ideal-cipher queries, n is the block size of the block cipher, k_h is the size of the hash key and k is the key size of the block cipher of the construction. As an instantiation of the Two-Keyed-DbHtS framework, we have proved that $\mathsf{C}_2[\mathsf{PH\text{-}DbH}, E]$, the Polyhash-based Two-Keyed-DbHtS construction which was proposed in [13] and proven to be secure up to $2^{2n/3}$ queries in the single-user setting, is multi-user secure up to $2^{3n/4}$ queries in the ideal-cipher model. The security proof of the construction crucially depends on a refined result of mirror theory over an abelian group $(\{0,1\}^n, \oplus)$, where one systematically estimates the number of solutions to a system of equations to prove the security of the finalization function of the construction up to $2^{3n/4}$ queries. Due to the attack result of Leurent et al. [21] on the DbHtS paradigm with $2^{3n/4}$ queries, the multi-user security bound of our construction is tight.

Organization. We have developed the required notations and security definitions of cryptographic primitives in Sect. 2. We demonstrate the construction and present its security bound in Sect. 3 and in Sect. 4, we prove the security of the construction. We instantiate the framework along with its security result in Sect. 5.

2 Preliminaries

<u>General Notations:</u> For a positive integer q, $[q]$ denotes the set $\{1,\ldots q\}$, and for two natural numbers q_1, q_2 such that $q_2 > q_1$, $[q_1, q_2]$ denotes the set $\{q_1,\ldots,q_2\}$. For a fixed positive integer n, we write $\{0,1\}^n$ to denote the set of all binary strings of length n and $\{0,1\}^* = \cup_{i\geq 0}\{0,1\}^i$ to denote the set of all binary strings with arbitrary finite length. We refer to the elements of $\{0,1\}^n$ as *blocks*. For a pair of blocks $x = (x_\ell, x_{\mathtt{r}}) \in \{0,1\}^{2n}$, we write $\mathsf{left}(x)$ to denote x_ℓ and $\mathsf{right}(x)$ to denote $x_{\mathtt{r}}$. For any element $x \in \{0,1\}^*$, $|x|$ denotes the number of bits in x and for $x, y \in \{0,1\}^*$, $x\|y$ denotes the concatenation of x followed by y. We denote the bitwise xor operation of $x, y \in \{0,1\}^n$ by $x \oplus y$. We parse $x \in \{0,1\}^*$ as $x = x_1\|x_2\|\ldots\|x_l$, where for each $i = 1,\ldots,l-1$, x_i is a block and $1 \leq |x_l| \leq n$. For $x \in \{0,1\}^n$, where $x = x_{n-1}\|\ldots\|x_0$, $\mathsf{lsb}(x)$ denotes the least significant bit x_0 of x. For a given bit b, fix_b is a function from $\{0,1\}^n$ to $\{0,1\}^n$ that takes an n-bit binary string $x = x_{n-1}\|\ldots\|x_0$ and returns an another binary string $x' = (x_{n-1}\|\ldots\|b)$, where $\mathsf{lsb}(x)$ is fixed to bit b. Given a tuple $\widetilde{x} = (x_1, x_2, \ldots, x_q)$ of n-bit binary string, we say that an element x_i of the tuple $\widetilde{x}$ is *non-fresh* if there exists at least one $j \neq i$ such that $x_i = x_j$. Otherwise, we call that element x_i is *fresh*.

Given a finite set $\mathcal{S}$ and a random variable X, we write $X \leftarrow_{\$} \mathcal{S}$ to denote that X is sampled uniformly at random from $\mathcal{S}$. We say that $X_1, X_2, \ldots, X_q$ are sampled with replacement (wr) from $\mathcal{S}$, which we denote as $X_1, X_2, \ldots X_q \leftarrow\!\!\$\, \mathcal{S}$, if for each $i \in [q], X_i \leftarrow\!\!\$\, \mathcal{S}$. We also use this notation to denote that these random variables are sampled uniformly and independently from $\mathcal{S}$. For a finite subset $\mathcal{S}$ of $\mathbb{N}$, $\max \mathcal{S}$ denotes the maximum-valued element of $\mathcal{S}$. ϕ denotes the empty set. We write $\mathcal{S} \leftarrow \phi$ to denote that $\mathcal{S}$ is defined to be an empty set. We also use the same notation $\Phi \leftarrow \phi$ to denote that the function Φ is undefined at every point of its domain. Moreover, the notation $Y \leftarrow X$ is used to denote the assigment of variable X to Y.

The set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ is denoted by $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$. Similarly, the set of all permutations over $\mathcal{X}$ is represented by $\mathsf{Perm}(\mathcal{X})$. A function Φ is said to be a *block function* if it maps elements from an arbitrary domain to $\{0,1\}^n$. The set of all block functions with domain $\mathcal{X}$ is denoted as $\mathsf{Func}(\mathcal{X})$. [5] We call Φ to be a *double-block function* if it maps elements from an arbitrary set $\mathcal{X}$ to $(\{0,1\}^n)^2$. For a given double-block function $\Phi : \mathcal{X} \to \{0,1\}^{2n}$, we write $\Phi_\ell : \mathcal{D} \to \{0,1\}^n$ such that for every $x \in \mathcal{X}$, $\Phi_\ell(x) = \mathsf{left}(\Phi(x))$. Similarly, we write $\Phi_{\mathtt{r}} : \mathcal{X} \to \{0,1\}^n$ such that for every $x \in \mathcal{X}$, $\Phi_{\mathtt{r}}(x) = \mathsf{right}(\Phi(x))$. For two block functions $\Phi_\ell : \mathcal{X} \to \{0,1\}^n$ and $\Phi_{\mathtt{r}} : \mathcal{X} \to \{0,1\}^n$, one can naturally define a double-block function $\Phi : \mathcal{X} \to \{0,1\}^{2n}$ such that $\Phi(x) = (\Phi_\ell(x), \Phi_{\mathtt{r}}(x))$, which we write as $\Phi = (\Phi_\ell, \Phi_{\mathtt{r}})$. For a finite set $\mathcal{X}$ and an integer q, we write $\mathcal{X}^{(q)}$ to denote the set $\{(x_1, x_2, \ldots, x_q) : x_i \in \mathcal{X}, x_i \neq x_j\}$. For integers $1 \leq b \leq a$, we write $\mathbf{P}(a,b)$ to denote $a(a-1)\ldots(a-b+1)$, where $\mathbf{P}(a,0) = 1$ by convention. Therefore, $|\mathcal{X}^{(q)}| = \mathbf{P}(|\mathcal{X}|, q)$.

2.1 Distinguishing Advantage

An adversary A is modeled as a randomized algorithm with access to an external oracle $\mathcal{O}$. Such an adversary is called an *oracle adversary.* An oracle $\mathcal{O}$ is an algorithm that may be a cryptographic scheme being analyzed. The interaction between A and $\mathcal{O}$, denoted by $\mathsf{A}^{\mathcal{O}}$, generates a transcript $\tau = \{(x_1, y_1), (x_2, y_2), \ldots, (x_q, y_q)\}$, where $x_1, x_2, \ldots, x_q$ are q queries of A to oracle $\mathcal{O}$ and $y_1, y_2, \ldots, y_q$ be the corresponding responses, where $y_i = \mathcal{O}(x_i)$. We assume that A is **adaptive**, which means that x_i is dependent on the previous $i-1$ responses.

<u>Distinguishing Game.</u> Let F and G be two random systems and an adversary A is given oracle access to either of F or G. After interaction with an oracle $\mathcal{O} \in \{\mathsf{F}, \mathsf{G}\}$, A outputs 1, which is denoted as $\mathsf{A}^{\mathcal{O}} \Rightarrow 1$. Such an adversary is called a *distinguisher* and the game is called a *distinguishing game.* The task of the distinguisher in a distinguishing game is to tell with which of the two systems it has interacted. The advantage of the distinguisher A in distinguishing the random system F from G is defined as

$$\mathbf{Adv}^{\mathsf{F}}_{\mathsf{G}}(\mathsf{A}) \stackrel{\Delta}{=} |\ \Pr[\mathsf{A}^{\mathsf{F}} \Rightarrow 1] - \Pr[\mathsf{A}^{\mathsf{G}} \Rightarrow 1]\ |,$$

here the above probability is defined over the probability spaces of A and $\mathcal{O}$. The maximum advantage in distinguishing F from G is defined as

$$\max_{\mathsf{A} \in \mathcal{A}} \mathbf{Adv}^{\mathsf{F}}_{\mathsf{G}}(\mathsf{A}),$$

where $\mathcal{A}$ is the class of all possible distinguishers. One can easily generalize this setting when the distinguisher interacts with multiple oracles, which are separated by commas. For example, $\mathbf{Adv}^{\mathsf{F}_1, \ldots, \mathsf{F}_m}_{\mathsf{G}_1, \ldots, \mathsf{G}_m}(\mathsf{A})$ denotes the advantage of A in distinguishing $(\mathsf{F}_1, \ldots, \mathsf{F}_m)$ from $(\mathsf{G}_1, \ldots, \mathsf{G}_m)$.

2.2 Block Cipher

A block cipher $\mathsf{E} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ is a function that takes a key $k \in \mathcal{K}$ and an n-bit input data $x \in \{0,1\}^n$ and produces an n-bit output y such that for each key $k \in \mathcal{K}$, $\mathsf{E}(k, \cdot)$

[5] When $\mathcal{X} = \{0,1\}^n$, we write Func to denote $\mathsf{Func}(\{0,1\}^n)$.

is a permutation over $\{0,1\}^n$. $\mathcal{K}$ is called the key space of the block cipher and $\{0,1\}^n$ is its input-output space. In shorthand notation, we write $\mathsf{E}_k(x)$ to represent $\mathsf{E}(k,x)$. Let $\mathsf{BC}(\mathcal{K},\{0,1\}^n)$ denotes the set of all n-bit block ciphers with key space $\mathcal{K}$. We say that a block cipher E is an (q,ϵ,t)-secure strong pseudorandom permutation, if for all distinguishers A that make a total of q forward and inverse queries with run time at most t, the following holds:

$$\mathbf{Adv}^{\mathsf{E}}_{\Pi}(\mathsf{A}) \triangleq |\ \Pr[K \leftarrow_{\$} \mathcal{K} : \mathsf{A}^{\mathsf{E}_K} \Rightarrow 1] - \Pr[\Pi \leftarrow_{\$} \mathsf{Perm} : \mathsf{A}^{\Pi} \Rightarrow 1]\ | \leq \epsilon.$$

2.3 PRF Security in the Ideal-Cipher Model

A *keyed function* with the key space $\mathcal{K}$, domain $\mathcal{X}$ and range $\mathcal{Y}$ is a function $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We denote $\mathsf{F}(k,x)$ by $\mathsf{F}_k(x)$. A random function RF from $\mathcal{X}$ to $\mathcal{Y}$ is a uniform random variable over the set $\mathsf{Func}(\mathcal{X},\mathcal{Y})$, i.e., $\mathsf{RF} \leftarrow_{\$} \mathsf{Func}(\mathcal{X},\mathcal{Y})$. We define the pseudorandom security of F under the ideal-cipher model. We assume that F makes internal calls to a publicly evaluated block cipher E with a randomly sampled block cipher key $K \leftarrow_{\$} \mathcal{K}$ (F can make calls to multiple block ciphers when all of them are independent and uniform over the set $\mathsf{BC}(\mathcal{K},\{0,1\}^n)$). For simplicity, we write $\mathsf{F}^{\mathsf{E}}_K$ to denote F with a uniformly sampled block cipher $\mathsf{E} \leftarrow_{\$} \mathsf{BC}(\mathcal{K},\{0,1\}^n)$, which is keyed by a randomly sampled $K \leftarrow_{\$} \mathcal{K}$. The distinguisher A is given access to either $(\mathsf{F}^{\mathsf{E}}_K, \mathsf{E}^{\pm})$ for $K \leftarrow_{\$} \mathcal{K}$ or $(\mathsf{RF}, \mathsf{E}^{\pm})$, where $\mathsf{E} \leftarrow_{\$} \mathsf{BC}(\mathcal{K},\{0,1\}^n)$ is a uniformly sampled n-bit block cipher such that A can make forward or inverse queries to E, which is denoted as $\mathsf{E}^{\pm}$. We define the prf-advantage of A against a keyed function F in the ideal cipher model as

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{F}}(\mathsf{A}) \triangleq \mathbf{Adv}^{(\mathsf{F}^{\mathsf{E}}_K,\mathsf{E}^{\pm})}_{(\mathsf{RF},\mathsf{E}^{\pm})}(\mathsf{A}).$$

We say F is a (q,p,ϵ,t)-PRF if $\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{F}}(\mathsf{A}) \leq \epsilon$ for all adversaries A that make q queries to F, p forward and inverse offline queries to E and run for time at most t.

2.4 Multi-User PRF Security in the Ideal-Cipher Model

We assume there are u users in the multi-user setting, such that the i-th user executes $\mathsf{F}^{\mathsf{E}}_{K_i}$. Furthermore, the i-th user key K_i is independent of the keys of all other users. An adversary A has access to all the u users as oracles. A make queries to the oracles in the form of (i,M) to the i-th user and obtains $T \leftarrow \mathsf{F}^{\mathsf{E}}_{K_i}(M)$. We call these **construction queries**. For $i \in [u]$, we assume A makes q_i queries to the i-th oracle. We also assume that A make queries to the underlying block cipher E and its inverse with some chosen keys k^j. We call these **primitive queries**. Suppose A chooses s distinct block cipher keys $(k^1,\ldots,k^s)$ and makes p_j primtive queries to the block cipher E with chosen keys k^j for $1 \leq j \leq s$. Let A be a (u,q,p,t)-adversary against the PRF security of F for all u users such that $q = q_1 + \ldots + q_u$ is the total number of construction queries and $p = p_1 + \ldots + p_s$ is the total number of primitive queries to the block cipher E with the total running time A being at most t. We assume that for any $i \in [u]$, A does not repeat any construction query to the i-th user. Similarly, A does not repeat any primitive query for any chosen block cipher key k^j to the block cipher E. The advantage of A in distinguishing $(\mathsf{F}^{\mathsf{E}}_{K_1}, \mathsf{F}^{\mathsf{E}}_{K_2}, \ldots, \mathsf{F}^{\mathsf{E}}_{K_u}, \mathsf{E}^{\pm})$ from $(\mathsf{RF}_1, \mathsf{RF}_2, \ldots, \mathsf{RF}_u, \mathsf{E}^{\pm})$ in the multi-user seting, where $\mathsf{RF}_1, \mathsf{RF}_2, \ldots, \mathsf{RF}_u \leftarrow_{\$} \mathsf{Func}(\mathcal{X},\mathcal{Y})$ are u independent random functions, is defined as

$$\mathbf{Adv}^{\mathrm{mu\text{-}PRF}}_{\mathsf{F}}(\mathsf{A}) \triangleq \left| \Pr\left[\mathsf{A}^{((\mathsf{F}^{\mathsf{E}}_{K_1},\ldots,\mathsf{F}^{\mathsf{E}}_{K_u}),\mathsf{E}^{\pm})} \Rightarrow 1\right] - \Pr\left[\mathsf{A}^{((\mathsf{RF},\ldots,\mathsf{RF}),\mathsf{E}^{\pm})} \Rightarrow 1\right]\right|,$$

where the randomness is defined over $K_1,\ldots,K_u \leftarrow_{\$} \mathcal{K}$, $\mathsf{E} \leftarrow_{\$} \mathsf{BC}(\mathcal{K},\{0,1\}^n)$ and the randomness of the adversary (if any). We write

$$\mathbf{Adv}^{\mathrm{mu\text{-}PRF}}_{\mathsf{F}}(u,q,p,\mathsf{t}) \triangleq \max_{\mathsf{A}} \mathbf{Adv}^{\mathrm{mu\text{-}PRF}}_{\mathsf{F}}(\mathsf{A}),$$

where the maximum is over all (u, q, p, t)-adversaries A. In this paper, we skip the time parameter of the adversary as we shall assume that the adversary is computationally unbounded. This also leads to the assumption that the adversary is deterministic. When $u = 1$, it makes $\mathbf{Adv}_{\mathsf{F}}^{\text{mu-PRF}}(u, q, p, \mathsf{t})$ the single-user distinguishing advantage.

2.5 Security of a Keyed Hash Function

Let $\mathcal{K}_h$ and $\mathcal{X}$ be two non-empty finite sets. A keyed function $\mathsf{H} : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$ is ϵ-almost-xor universal (axu) if for any distinct $x, x' \in \mathcal{X}$ and for any $\Delta \in \{0,1\}^n$,

$$\Pr[K_h \leftarrow_{\$} \mathcal{K}_h : \mathsf{H}_{K_h}(x) \oplus \mathsf{H}_{K_h}(x') = \Delta] \leq \epsilon_{\text{axu}}.$$

Moreover, H is an ϵ-universal hash function if for any distinct $x, x' \in \mathcal{X}$,

$$\Pr[K_h \leftarrow_{\$} \mathcal{K}_h : \mathsf{H}_{K_h}(x) = \mathsf{H}_{K_h}(x')] \leq \epsilon_{\text{univ}}.$$

A keyed hash function is said to be ϵ-regular if for any $x \in \mathcal{X}$ and for any $\Delta \in \{0,1\}^n$,

$$\Pr[K_h \leftarrow_{\$} \mathcal{K}_h : \mathsf{H}_{K_h}(x) = \Delta] \leq \epsilon_{\text{reg}}.$$

2.6 Mirror Theory

Mirror theory is a collection of combinatorial results that give a lower bound on the number of solutions to a system of bivariate affine equations $\mathbb{E}$ over an abelian group $(\{0,1\}^n, \oplus)$. We represent a system of equations by a simple graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ containing no loops or multiple edges, where each vertex denotes an n-bit unknown (for a fixed n), and we connect vertices P and Q with an edge labeled $\lambda \in \{0,1\}^n$ if $P \oplus Q = \lambda \in \mathcal{E}$. For a path $\mathcal{L} = P_1 \xrightarrow{\lambda_1} P_2 \xrightarrow{\lambda_2} \ldots \xrightarrow{\lambda_\ell} P_\ell$ in the graph $\mathcal{G}$, we define the label of the path

$$\lambda(\mathcal{L}) = \lambda_1 \oplus \lambda_2 \oplus \ldots \oplus \lambda_\ell.$$

In this work, we focus on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with certain properties as listed below:

1. $\mathcal{G}$ contains no isolated vertex, i.e., every vertex is incident with at least one edge.
2. The vertex set $\mathcal{V}$ is partitioned into two disjoint sets denoted by $\mathcal{P}$ and $\mathcal{Q}$, where there are no edges within the vertex set in partition $\mathcal{P}$ or in partition $\mathcal{Q}$. All edges connect a vertex in $\mathcal{P}$ to a vertex in $\mathcal{Q}$. We call such graphs *bipartition graphs*.
3. $\mathcal{G}$ contains no cycle.
4. $\lambda(\mathcal{L}) \neq 0^n$ for any path $\mathcal{L}$ in $\mathcal{G}$.

Any bipartition graph $\mathcal{G}$ satisfying the above properties shall be called a **good graph**. Note that a good bipartition graph $\mathcal{G}$ contains no cycle. Therefore, $\mathcal{G}$ can be decomposed into its connected components, all of which are trees; let

$$\mathcal{G} = \mathcal{C}_1 \sqcup \mathcal{C}_2 \sqcup \ldots \sqcup \mathcal{C}_\alpha \sqcup \mathcal{D}_1 \sqcup \mathcal{D}_2 \sqcup \ldots \sqcup \mathcal{D}_\beta$$

for some $\alpha, \beta \geq 0$, where $\mathcal{C}_i$ denotes a component of size greater than 2, and $\mathcal{D}_i$ denotes a componenent size of 2. We write $\mathcal{C} = \mathcal{C}_1 \sqcup \mathcal{C}_2 \sqcup \ldots \sqcup \mathcal{C}_\alpha$ and $\mathcal{D} = \mathcal{D}_1 \sqcup \mathcal{D}_2 \sqcup \ldots \sqcup \mathcal{D}_\beta$.

Definition 1. Let $\mathcal{E}_\mathcal{G}$ be a system of equations induced by a good biparite graph $\mathcal{G}$. An injective function $\Phi : \mathcal{P} \sqcup \mathcal{Q} \to \{0,1\}^n$ is said to be an *injective solution* to $\mathcal{E}_\mathcal{G}$ if $\Phi(P_i) \oplus \Phi(Q_j) = \lambda_{ij}$ for all $\{P_i, Q_j\} \in \mathcal{E}$.

We remark that assigning any value to a vertex in P allows the labeled edges to uniquely determine the values of all the other vertices in the component containing P, since $\mathcal{G}$ contains no cycle. The values in the same component are all distinct as $\lambda(\mathcal{L}) \neq 0^n$ for any path $\mathcal{L}$. The number of possible assignments of distinct values to the vertices in $\mathcal{G}$ is $\mathbf{P}(2^n, |\mathcal{P}| + |\mathcal{Q}|)$. One may expect that when such an assignment is chosen uniformly at random, it would satisfy all the equations in $\mathcal{G}$ with probability 2^{-nq}, where q denotes the number of edges (i.e., equations) in $\mathcal{G}$. Indeed, we can prove that the number of solutions is closed to $\mathbf{P}(2^n, |\mathcal{P}| + |\mathcal{Q}|)/2^{nq}$, up to a certain error. Formally, we have the following result:

Lemma 1. *Let $\mathcal{G}$ be a good bipartition graph, and let q and q^{c} denote the number of edges of $\mathcal{G}$ and $\mathcal{C}$, respectively. Let v be the number of vertices of $\mathcal{G}$. If $q < 2^n/8$, then the number of solutions to $\mathcal{G}$, denoted $h(\mathcal{G})$, satisfies*

$$\frac{h(\mathcal{G})2^{nq}}{\mathbf{P}(2^n, v)} \geq \left(1 - \frac{9(q^{\mathsf{c}})^2}{8 \cdot 2^n} - \frac{3q^{\mathsf{c}}q^2}{2 \cdot 2^{2n}} - \frac{q^2}{2^{2n}} - \frac{9(q^{\mathsf{c}})^2 q}{8 \cdot 2^{2n}} - \frac{8q^4}{3 \cdot 2^{3n}}\right).$$

We refer the reader to [20] for a proof of the lemma.

3 The Two-Keyed DbHtS Construction

In this section, we describe the Two-Keyed Double-block Hash-then-Sum or in short, Two-Keyed-DbHtS construction to build a beyond birthday bound secure variable input length PRF. Let $\mathsf{H}^1 : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ and $\mathsf{H}^2 : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ be two keyed hash functions. Based on H^1 and H^2, we define the Double-block Hash or in short DbH function $\mathsf{H} : \mathcal{K}_h \times \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^{2n}$ as follows:

$$\mathsf{H}_{(L_1,L_2)}(M) = (\mathsf{H}^1_{L_1}(M), \mathsf{H}^2_{L_2}(M)). \tag{1}$$

We compose this DbH function with a very simple and efficient single-keyed xor function $\mathsf{XOR}_K(x,y) = \mathsf{E}_K(x) \oplus \mathsf{E}_K(y)$, where E_K is an n-bit block cipher and the block cipher key K is independent over the hash key (L_1, L_2), to realize the two-Keyed-DbHtS construction as follows:

$$\mathsf{C}_2[\mathsf{H}, \mathsf{E}]_{(L_1,L_2,K)}(M) := \mathsf{XOR}_K(\mathsf{H}^1_{L_1}(M), \mathsf{H}^2_{L_2}(M)).$$

We use the name Two-Keyed-DbHtS construction, as we count the hash key as one key and the xor function requiring one key, which is independent of the hash key. Most of the beyond birthday bound secure variable input length PRFs like 2K-SUM-ECBC, 2K-PMAC_Plus, 2K-LightMAC_Plus are specific instantiations of the Two-Keyed-DbHtS paradigm. These constructions (with domain separation technique) have been proven secured up to $2^{2n/3}$ queries in the standard model [13] for a single-user setting. In [32], all these three constructions (without domain separation technique) have been proven secured up to $2^{2n/3}$ queries in the ideal-cipher model for a multi-user setting. We note here that as the xor function is not a PRF over two blocks, we can not apply the tradition *Hash-the-PRP* composition result directly to analyze the security of the two-keyed DbHtS. Thus, we need a different type of composition result for the security analysis of the Two-Keyed-DbHtS construction that utilizes higher security properties of its underlying DbH function instead of having only the universal or regular property.

Definition 2. Let $\mathsf{H}^1 : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ and $\mathsf{H}^2 : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ be two n-bit keyed hash functions. We say that the double-block hash function $\mathsf{H} : \mathcal{K}_h \times \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^{2n}$ defined in Eqn. (1) is **good** if it satisfies the following conditions:

- H^1 is a family of ϵ_{reg}-regular and ϵ_{univ}-universal functions.

- H^2 is a family of ϵ_{reg}-regular and ϵ_{univ}-universal functions.
- For every $M, M' \in \{0,1\}^*$, $\Pr[L_1 \leftarrow\!\!\$\, \mathcal{K}_h, L_2 \leftarrow\!\!\$\, \mathcal{K}_h : \mathsf{H}^1_{L_1}(M) = \mathsf{H}^2_{L_2}(M')] = 0$.

The first two condition imply that the regular and universal advantages of both the hash functions should be negligible, whereas the last condition indicates that the first hash output for any message cannot collide with the second hash output. Having defined the Two-Keyed-DbHtS construction, we now state and prove its security. For the sake of brevity, we refer to the Two-Keyed-DbHtS construction $\mathsf{C}_2[\mathsf{H}, \mathsf{E}]_{(L_1, L_2, K)}$ by simply C_2 without mentioning the underlying hash function, the block cipher and their associated keys.

Theorem 1. *Let $\mathcal{K}, \mathcal{K}_h$ and $\mathcal{M}$ be three non-empty finite sets. Let $\mathsf{E} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be an n-bit block cipher. Let $\mathsf{H}^1 : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ and $\mathsf{H}^2 : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ be two n-bit keyed hash functions such that each is ϵ_{reg}-regular and ϵ_{univ}-universal. Let $\mathsf{H} : \mathcal{K}_h \times \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^{2n}$ be a* **good** *double-block hash function as defined in Eqn.* (1). *Then any computationally unbounded distinguisher making a total of q construction queries across all u users and a total of p primitive queries to the block cipher E can distinguish C_2 from an n-bit uniform random function with prf advantage*

$$\begin{aligned}\mathbf{Adv}^{\text{mprf}}_{\mathsf{C}_2}(u,q,p,\ell) \quad \leq \quad & \frac{9q^{4/3}}{8 \cdot 2^n} + \frac{3q^{8/3}}{2 \cdot 2^{2n}} + \frac{q^2}{2^{2n}} + \frac{9q^{7/3}}{8 \cdot 2^{2n}} + \frac{8q^4}{3 \cdot 2^{3n}} + \frac{q}{2^n} + \frac{2u^2}{2^{k_h+k}} + \frac{2q^2}{2^{n+k}} \\ & + \frac{2qp\epsilon_{\text{reg}}}{2^k} + \frac{q^2\epsilon_{\text{univ}}}{2^n} + \frac{2q^2\epsilon_{\text{reg}}}{2^{k_h}} + 3q^{4/3}\epsilon_{\text{univ}} + \frac{q^2\epsilon^2_{\text{univ}}}{2} + \frac{2qp}{2^{n+k}}.\end{aligned}$$

4 Proof of Theorem 1

We consider a computationally unbounded non-trivial deterministic distinghisher A that interacts with a pair of oracles in either the real world or the ideal world, described as follows: in the real world, A is given access to u independent instances of the Two-Keyed-DbHtS construction, i.e., to a tuple of u oracles $(\mathsf{C}_2[\mathsf{H}, \mathsf{E}]_{(L_1^i, L_2^i, K^i)})_{i \in [u]}$, where each (L_1^i, L_2^i) is independent of (L_1^j, L_2^j), K^i is independent of K^j and $\mathsf{E} \leftarrow\!\!\$\, \mathsf{BC}(\mathcal{K}, \{0,1\}^n)$ is an ideal block cipher. Additionally, A has access to the oracle $\mathsf{E}^{\pm}$, underneath the construction C_2. In the ideal world, A is given access to (i) a tuple of u independent random functions $(\mathsf{RF}_1, \ldots, \mathsf{RF}_u)$, where each RF_i is the random function over $\{0,1\}^*$ to $\{0,1\}^n$ that can be equivalently described as a procedure that returns an n-bit uniform string on input of any arbitrary message, and (ii) the oracle $\mathsf{E}^{\pm}$, where $\mathsf{E} \leftarrow\!\!\$\, \mathsf{BC}(\mathcal{K}, \{0,1\}^n)$ is an ideal block cipher, sampled independent of the distribution of the sequence of u independent random functions. In both the worlds, the first oracle is called the *construction oracle* and the latter, the *ideal cipher oracle*. Using the ideal cipher oracle, a distinguisher A can evaluate any query x under its chosen key J. A query to the construction oracle is called a *construction query* and to that of the ideal cipher oracle is called an *ideal cipher query*. Note that A can make either *forward* (i.e., it evaluates E with a chosen key and input), or *inverse* ideal cipher queries (i.e., it evaluates E^{-1} with a chosen key and input). The ideal oracle is depicted in Fig.s 4.1 and 4.2.

4.1 Description of the Ideal World

The ideal world consists of two phases: (i) the online and (ii) the offline phase. Before the game begins, we sample u independent functions $f_1, f_2, \ldots, f_u$ uniformly at random from the set of all functions $\mathsf{Func}(\{0,1\}^*, \{0,1\}^n)$ that map an arbitrary-length string to an n-bit string. We also sample an n-bit block cipher E from the set of all block ciphers with a k-bit key and an n-bit input. In the online phase, when the distinguisher makes the a-th construction query for the i-th user M_a^i to the construction oracle, it returns $T_a^i \leftarrow f_i(M_a^i)$. Similarly, if the distinguisher makes a forward (resp. inverse) primitive query with a chosen block cipher key J and an input x to the ideal cipher oracle, it returns $\mathsf{E}(J, x)$ (resp. $\mathsf{E}^{-1}(J, x)$). However, if any response of the construction queries is an all-zero string 0^n, then the bad flag $\mathsf{Bad\text{-}Tag}$ is set to 1 and the game is aborted.

Online Phase of $\mathcal{O}_{\mathrm{ideal}}$

1 : $\mathsf{E} \leftarrow_{\$} \mathsf{BC}(\mathcal{K}, \{0,1\}^n)$;

Construction Query:

2 : On a-th query of i-th user M_a^i, **return** $T_a^i \leftarrow_{\$} \{0,1\}^n$;

3 : `if` $\exists (i, a) : T_a^i = \mathbf{0}$ `then` $\boxed{\mathsf{Bad\text{-}Tag} \leftarrow 1}$, $\perp$;

Primitive Query:

4 : On j-th forward query with chosen key J^j and input u_α^j, **return** $v_\alpha^j \leftarrow \mathsf{E}_{J^j}(u_\alpha^j)$;

5 : On j-th backward query with chosen key J^j and input v_α^j, **return** $u_\alpha^j \leftarrow \mathsf{E}_{J^j}^{-1}(v_\alpha^j)$;

6 : $\mathsf{Dom}(\mathsf{E}_{J^j}) \leftarrow \mathsf{Dom}(\mathsf{E}_{J^j}) \cup \{u_\alpha^j\}$, $\mathsf{Ran}(\mathsf{E}_{J^j}) \leftarrow \mathsf{Ran}(\mathsf{E}_{J^j}) \cup \{v_\alpha^j\}$;

Figure 4.1: Online Phase of the Ideal oracle \$: Boxed statements denote bad events. Whenever a bad event is set to 1, the ideal oracle immediately aborts (denoted as $\perp$) and returns the remaining values of the transcript in an arbitrary manner. So, if the game aborts for some bad event, then its previous bad events must not have occurred.

After this interaction is over, the offline phase begins. In this phase, we sample u pairs of dummy hash keys $(L_1^i, L_2^i)_{i \in [u]} \leftarrow_{\$} \mathcal{K}_h \times \mathcal{K}_h$ and u dummy block cipher keys $(K^i)_{i \in [u]} \leftarrow_{\$} \mathcal{K}$, where L_1^i (resp. L_2^i) is the *left (resp. right) hash key* for the i-th user and K^i is its *block cipher key.* If the block cipher key and a left (resp. right) hash key of the i_1-th user collides with the block cipher key and left (resp. right) hash key of the i_2-th user, then we set the flag BadK to 1 and abort the game. If the game is not aborted, then we can compute a pair of $2n$-bit hash values (Σ_a^i, Θ_a^i) for all queries across u users, where we often refer to $\Sigma_a^i \leftarrow \mathsf{H}^1_{L_1^i}(M_a^i)$ as the *left hash output* and to $\Theta_a^i \leftarrow \mathsf{H}^2_{L_2^i}(M_a^i)$ as the *right hash output* for the a-th query of the i-th user.

Now, if the block cipher key of the i-th user and the left hash or right hash output for its a-th query collides with some chosen ideal cipher key and one of the corresponding inputs of the forward ideal cipher query, then we set the bad flag $\mathsf{Bad1}$ to 1 and abort the game.

For the i-th user, if the left or right hash outputs for two of its queries collide and the corresponding responses also collide with each other (i.e., $\Sigma_a^i = \Sigma_b^i, T_a^i = T_b^i$), then we consider it to be a bad event. Similarly, for a pair of users i_1 and i_2, if their left or right hash outputs collide with each other and the corresponding responses also collide with each other, then we again consider it to be a bad event. If at least one of the above bad events occurs, we set $\mathsf{Bad2}$ to 1 and abort the game. We also set another flag $\mathsf{Bad3}$ to 1 and abort the game if for the i-th user, the number of the pairs of queries whose either left or right hash outputs collide with each other is at least $q_i^{2/3}$, where q_i is the number of queries made by the i-th user.

Finally, we set the flag $\mathsf{Bad4}$ to 1 if at least one of the following events holds: (a) for the i-th user, two left hash outputs collide and their corresponding right hash outputs also collide, or (b) for the i-th user, there exists a tuple of four query indices a, b, c, d such that either (i) $\Sigma_a^i = \Sigma_b^i, \Theta_b^i = \Theta_c^i, \Sigma_c^i = \Sigma_d^i$ holds or (ii) $\Theta_a^i = \Theta_b^i, \Sigma_b^i = \Sigma_c^i, \Theta_c^i = \Theta_d^i$ holds. As the DbH function H is **good**, Σ_a^i cannot collide with Θ_b^i. It is also to be noted here that as the hash function is good, i.e., the hash outputs of two hash functions never collide, it immediately rules out the attack of [16].

If the game is not aborted at this stage, then it follows that none of the bad events have occurred. All the query-response pairs belong to exactly one of the sets $\mathcal{Q}^{=}$ or $\mathcal{Q}^{\neq}$ as

Offline Phase of $\mathcal{O}_{\text{ideal}}$

1 : $(L_1^i, L_2^i)_{i\in[u]} \leftarrow\!\!\$\, \mathcal{K}_h \times \mathcal{K}_h;\quad (K^i)_{i\in[u]} \leftarrow\!\!\$\, \mathcal{K};$

2 : **if** $\exists \mathtt{b} \in \{1,2\}$ and $i_1, i_2 \in [u]$ such that $K^{i_1} = K^{i_2} \wedge L_b^{i_1} = L_b^{i_2}$;

3 : **then** $\boxed{\mathsf{BadK} \leftarrow 1}$, $\perp$;

4 : $\forall i \in [u], \forall a \in [q_i] :\quad (\Sigma_a^i, \Theta_a^i) \leftarrow (\mathsf{H}^1_{L_1^i}(M_a^i), \mathsf{H}^2_{L_2^i}(M_a^i))$;

5 : if one of the following holds:

(a) $\exists i \in [u], j \in [s], u[0]^j_\alpha \in \mathsf{Dom}(\mathsf{E}_{J^j})$, such that $K^i = J^j \ \wedge\ \Sigma_a^i = u[0]^j_\alpha$;

(b) $\exists i \in [u], j \in [s], u[1]^j_\alpha \in \mathsf{Dom}(\mathsf{E}_{J^j})$, such that $K^i = J^j \ \wedge\ \Theta_a^i = u[1]^j_\alpha$;

6 : **then** $\boxed{\mathsf{Bad1} \leftarrow 1}$, $\perp$;

7 : if one of the following holds:

(a) $\exists i \in [u], a, b \in [q_i]$, such that $\Sigma_a^i = \Sigma_b^i \ \wedge\ T_a^i = T_b^i$;

(b) $\exists i_1, i_2 \in [u], a \in [q_{i_1}], b \in [q_{i_2}]$, such that $K^{i_1} = K^{i_2} \ \wedge\ \Sigma_a^{i_1} = \Sigma_b^{i_2}$;

(c) $\exists i \in [u], a, b \in [q_i]$, such that $\Theta_a^{i_1} = \Theta_b^{i_1} \ \wedge\ T_a^{i_1} = T_b^{i_1}$;

(d) $\exists i_1, i_2 \in [u], a \in [q_{i_1}], b \in [q_{i_2}]$, such that $K^{i_1} = K^{i_2} \ \wedge\ \Theta_a^{i_1} = \Theta_b^{i_2}$;

8 : **then** $\boxed{\mathsf{Bad2} \leftarrow 1}$, $\perp$;

9 : if one of the following holds:

(a) $\exists i \in [u]$, such that $\left|\left\{(a,b) : \Sigma_a^i = \Sigma_b^i\right\}\right| \geq q_i^{2/3}$;

(b) $\exists i \in [u]$, such that $\left|\left\{(a,b) : \Theta_a^i = \Theta_b^i\right\}\right| \geq q_i^{2/3}$;

10 : **then** $\boxed{\mathsf{Bad3} \leftarrow 1}$, $\perp$;

11 : if one of the following holds:

(a) $\exists i \in [u], a, b \in [q_i]$ such that $\Sigma_a^i = \Sigma_b^i \ \wedge\ \Theta_a^i = \Theta_b^i$;

(b) $\exists i \in [u], a, b, c, d \in [q_i]$ such that $\Sigma_a^i = \Sigma_b^i \ \wedge\ \Theta_b^i = \Theta_c^i \ \wedge\ \Sigma_c^i = \Sigma_d^i$;

(c) $\exists i \in [u], a, b, c, d \in [q_i]$ such that $\Theta_a^i = \Theta_b^i \ \wedge\ \Sigma_b^i = \Sigma_c^i \ \wedge\ \Theta_c^i = \Theta_d^i$;

12 : **then** $\boxed{\mathsf{Bad4} \leftarrow 1}$, $\perp$;

13 : go to subroutine 4.3;

Figure 4.2: Offline Phase of the Ideal oracle \$: Boxed statements denote bad events. Whenever a bad event is set to 1, the ideal oracle immediately aborts (denoted as $\perp$) and returns the remaining values of the transcript in an arbitrary manner. So, if the game aborts for some bad event, then we may assume that the previous bad events have not occurred.

defined in lines 13 and 14 of Fig. 4.2, where $\mathcal{Q}^=$ is the set of all queries across all users such that the block cipher key of the i-th user collides with an ideal cipher key, but none of its hash outputs collide with any ideal cipher query, and $\mathcal{Q}^{\neq}$ is the set of all queries across all users such that the block cipher key of the i-th user does not collide with any ideal cipher key. We also define two additional sets: $\mathcal{I}^=$ and $\mathcal{I}^{\neq}$ for $\mathcal{Q}^=$ and $\mathcal{Q}^{\neq}$, where $\mathcal{I}^=$ (resp. $\mathcal{I}^{\neq}$) is the set of all i such that $(i, \star) \in \mathcal{Q}^=$ (resp. $(i, \star) \in \mathcal{Q}^{\neq}$). We partition $\mathcal{I}^=$ into r non-empty equivalence classes $\mathcal{I}_1^=, \mathcal{I}_2^=, \ldots, \mathcal{I}_r^=$ based on the relation that the i-th user key K^i collides with J^j if and only if $i \in \mathcal{I}_j^=$. Similarly, we partition $\mathcal{I}^{\neq}$ into s equivalence classes based on the equivalence relation $i \sim j$ if and only if $K^i = K^j$. Now, for the j-th equivalence class of $\mathcal{I}^=$, we consider the tuple

$$\widetilde{\Sigma}_j := \bigcup_{i\in\mathcal{I}_j^=} \{(\Sigma_1^i, \Sigma_2^i, \ldots, \Sigma_{q_i}^i)\},\quad \widetilde{\Theta}_j := \bigcup_{i\in\mathcal{I}_j^=} \{(\Theta_1^i, \Theta_2^i, \ldots, \Theta_{q_i}^i)\}.$$

Offline Phase of $\mathcal{O}_{\text{ideal}}$, Sampling Phase

1 : $\mathcal{Q}^{=} := \{(i,a) \in [u] \times [q_i] : \exists j \in [s], K^i = J^j, \Sigma_a^i \notin \mathsf{Dom}(\mathsf{E}_{J^j}), \Theta_a^i \notin \mathsf{Dom}(\mathsf{E}_{J^j})\};$

2 : $\mathcal{I}^{=} := \{i \in [u] : (i,\star) \in \mathcal{Q}^{=}\} = \mathcal{I}_1^{=} \sqcup \mathcal{I}_2^{=} \sqcup \ldots \sqcup \mathcal{I}_r^{=}; \qquad // \ i \in \mathcal{I}_j^{=} \Leftrightarrow K^i = J^j$

3 : $\forall j \in [r] : \ \widetilde{\Sigma^j} = \bigcup_{i \in \mathcal{I}_j^{=}} \{(\Sigma_1^i, \Sigma_2^i, \ldots, \Sigma_{q_i}^i)\}, \ \widetilde{\Theta^j} = \bigcup_{i \in \mathcal{I}_j^{=}} \{(\Theta_1^i, \Theta_2^i, \ldots, \Theta_{q_i}^i)\};$

4 : $\forall j \in [r]$ do the following steps:

5 : $\quad \forall i \in \mathcal{I}_j^{=}$ let Σ_a^i be not fresh in $(\Sigma_1^i, \Sigma_2^i, \ldots, \Sigma_{q_i}^i)$;

6 : $\quad$ `if` $\Sigma_a^i \notin \mathsf{Dom}(\mathsf{E}_{J^j})$, `then` $\Psi_j(\Sigma_a^i) \leftarrow Z_{1,a}^i \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathsf{Ran}(\mathsf{E}_{J^j}), \quad Z_{2,a}^i \leftarrow Z_{1,a}^i \oplus T_a^i;$

7 : $\quad$ `else` $Z_{1,a}^i \leftarrow \Psi_j(\Sigma_a^i), \quad Z_{2,a}^i \leftarrow Z_{1,a}^i \oplus T_a^i;$

8 : $\quad$ `if` $Z_{2,a}^i \in \mathsf{Ran}(\mathsf{E}_{J^j})$ `then` $\boxed{\mathsf{Bad\text{-}Samp} \leftarrow 1}, \ \perp;$

9 : $\quad$ `else` $\mathsf{Dom}(\mathsf{E}_{J^j}) \leftarrow \mathsf{Dom}(\mathsf{E}_{J^j}) \cup \{(\Sigma_a^i, \Theta_a^i)\}, \ \mathsf{Ran}(\mathsf{E}_{J^j}) \leftarrow \mathsf{Ran}(\mathsf{E}_{J^j}) \cup \{(Z_a^i, Z_a^i \oplus T_a^i)\};$

10 : $\quad$ `Set` $\Psi_j(\Sigma_a^i) \leftarrow Z_{1,a}^i, \ \Psi_j(\Theta_a^i) \leftarrow Z_{2,a}^i, \ \forall i \in \mathcal{I}_j^{=}, a \in [q_i];$

11 : $\mathcal{Q}^{\neq} := \{(i,a) \in [u] \times [q_i] : \forall j \in [s], K^i \neq J^j\};$

12 : $\mathcal{I}^{\neq} := \{i \in [u] : (i,\star) \in \mathcal{Q}^{\neq}\} = \mathcal{I}_1^{\neq} \sqcup \mathcal{I}_2^{\neq} \sqcup \ldots \sqcup \mathcal{I}_{r'}^{\neq}; \qquad // \ i \in \mathcal{I}_j^{\neq} \Leftrightarrow K^i = K^j$

13 : $\forall j \in [r'] : \ f_j :=$ distinct number of elements in the tuple $\widetilde{\Sigma_j} \cup \widetilde{\Theta_j}$;

14 : $\forall j \in [r'] : \ (Z_{1,a}^i, Z_{2,a}^i)_{i \in \mathcal{I}_j^{\neq}, a \in [q_i]} \leftarrow\!\!\$\ \mathcal{S}_j := \{(Q_a^i, R_a^i)_{i \in \mathcal{I}_j^{\neq}, a \in [q_i]} \in (\{0,1\}^n)^{(f_j)} : \ Q_a^i \oplus R_a^i = T_a^i\};$

15 : $\forall j \in [r'] :$ do the following steps:

16 : $\quad \mathsf{Dom}(\mathsf{E}_J) \leftarrow \mathsf{Dom}(\mathsf{E}_J) \cup \{(\Sigma_a^i, \Theta_a^i) : i \in \mathcal{I}_j^{\neq}, a \in [q_i]\};$

17 : $\quad \mathsf{Ran}(\mathsf{E}_J) \leftarrow \mathsf{Ran}(\mathsf{E}_J) \cup \{(Z_{1,a}^i, Z_{2,a}^i) : i \in \mathcal{I}_j^{\neq}, a \in [q_i]\};$

18 : $\quad$ `Set` $\Psi_j(\Sigma_a^i) \leftarrow Z_{1,a}^i, \Psi_j(\Theta_a^i) \leftarrow Z_{2,a}^i, \ \forall i \in \mathcal{I}_j^{\neq}, a \in [q_i];$

19 : **return** $(\Sigma_a^i, \Theta_a^i, Z_{1,a}^i, Z_{2,a}^i)_{(i,a) \in [u] \times [q_i]};$

Figure 4.3: Offline Phase of the Ideal oracle \$, where we sample the output of the hash values.

Note that due to the event in line number 7.(b) (resp. 7.(d)) of Fig. 4.2, we have $\Sigma_a^{i_1} \neq \Sigma_b^{i_2}$ (resp. $\Theta_a^{i_1} \neq \Theta_b^{i_2}$) for $i_1, i_2 \in \mathcal{I}_j^{=}$ and $a \in [q_{i_1}], b \in [q_{i_2}]$. If Σ_a^i is not fresh in the tuple $(\Sigma_1^i, \Sigma_2^i, \ldots, \Sigma_{q_i}^i)$ for some $(i,a) \in \mathcal{I}_j^{=} \times [q_i]$ and the output of Σ_a^i has not been sampled yet, then we sample the its output $Z_{1,a}^i$ from outside the range of E_{J^j} and set the output of Θ_a^i as the xor of Z_a^i and T_a^i (see line 6 of Fig. 4.3). Otherwise, we set the output of Σ_a^i to the already defined element and adjust the output of the other hash vaue accordingly (see line 7 of Fig. 4.3). Note that in the latter case, the we do not sample the output. In the above adjustment, if the output of Θ_a^i happens to collide with any previously sampled output, then we set flag Bad-Samp to 1 and abort the game (see line 8 of Fig. 4.3) and abort the game. Note that this event cannot hold for the real oracle, as Θ_a^i is fresh in $(\Theta_1^i, \Theta_2^i, \ldots, \Theta_{q_i}^i)$ for $i \in \mathcal{I}_j^{=}$ and $a \in [q_i]$. If the above flag is not set to 1, then the sampling for the output of Σ_a^i, where $(i,a) \in \mathcal{Q}^{=}$ preserves permutation compatibility. Finally, for all other $(i,a) \in \mathcal{Q}^{\neq}$, we sample $Z_{1,a}^i$ and $Z_{2,a}^i$ such that $Z_{1,a}^i \oplus Z_{2,a}^i = T_a^i$.

4.2 Attack Transcript

We summarize here, the interaction between the distinguisher and the challenger in a transcript. The set of all construction queries for u instances are summarized in a transcript $\tau_c = \tau_c^1 \cup \tau_c^2 \cup \ldots \cup \tau_c^u$, where $\tau_c^i = \{(M_1^i, T_1^i), \ldots, (M_{q_i}^i, T_{q_i}^i)\}$ denotes the query-response transcript generated from the i-th instance of the construction. Moreover, we assume that A has chosen s distinct ideal cipher keys $J^1, \ldots, J^s$ such that it makes p_j ideal cipher queries to the block cipher with the chosen key J^j. We summarize the ideal cipher queries in a transcript $\tau_p = \tau_p^1 \cup \tau_p^2 \cup \ldots \cup \tau_p^s$, where $\tau_p^j = \{(u_1^j, v_1^j), \ldots, (u_{p_j}^j, v_{p_j}^j), J^j\}$ denotes the transcript of the ideal cipher queries when the chosen ideal cipher key is J^j. We assume that A makes q_i construction queries for the i-th instance and p_j ideal cipher queries (including forward and inverse queries) with chosen ideal cipher key J^j. We also assume that the total number of construction queries across u instances is q, i.e., $q = (q_1 + \ldots + q_u)$ and the total number of ideal cipher queries is $p = (p_1 + \ldots + p_s)$. Since A is non-trivial, none of the transcripts contain any duplicate elements.

We modify the experiment by releasing internal information to A after it has finished its interaction but has not yet output the decision bit. In the real world, we reveal all the keys (L_1^i, L_2^i, K^i) for all u instances used in the construction. In the ideal world, we sample them uniformly at random from their respective key spaces and reveal them to the distinguisher. Once the keys are revealed to the distinguisher, A can compute $(\Sigma_a^i, \Theta_a^i, \Psi_j(\Sigma_a^i), \Psi_j(\Theta_a^i))$, where $i \in \mathcal{I}_j^{=}$ or $i \in \mathcal{I}_j^{\neq}$ and the function Ψ_j defined for the ideal world is given in Fig. 4.3. On the other hand, for the real world, we define Ψ_j as follows:

$$\Psi_j(\Sigma_a^i) = \mathsf{E}_{K^i}(\Sigma_a^i), \quad \Psi_j(\Theta_a^i) = \mathsf{E}_{K^i}(\Theta_a^i),$$

for $i \in \mathcal{I}_j^{=}$ or $i \in \mathcal{I}_j^{\neq}$ Therefore, each transcript τ_i^c, where $i \in \mathcal{I}_j^{=}$ or $i \in \mathcal{I}_j^{\neq}$, is now modified to include the corresponding intermediate input-output values for the i-th instance of the construction. Thus,

$$\tau_c^i = \{(M_1^i, T_1^i, \Sigma_1^i, \Theta_1^i, \Psi_j(\Sigma_1^i), \Psi_j(\Theta_1^i)), \ldots, (M_{q_i}^i, T_{q_i}^i, \Sigma_{q_i}^i, \Theta_{q_i}^i, \Psi_j(\Sigma_{q_i}^i), \Psi_j(\Theta_{q_i}^i))\}.$$

In all the following, the complete construction query transcript is

$$\tau_c = \bigcup_{i=1}^{u} \tau_c^i$$

and the overall transcript is $\tau = \tau_c \cup \tau_p$. The modified experiment only makes the distinguisher more powerful and hence the distinguishing advantage of A in this experiment is no less than its distinguishing advantage in the former. Let X_{re} denote the random variable that takes a transcript τ realized in the real world. Similarly, X_{id} denotes the random variable that takes a transcript τ realized in the ideal world. The probability of realizing a transcript τ in the ideal (resp. real) world is called the *ideal* (*resp. real*) *interpolation probability.* A transcript τ is said to be attainable with respect to A if its ideal interpolation probability is non-zero, and Θ denotes the set of all such attainable transcripts. Following these notations, we now state the main theorem of the H-coefficient technique [31]:

Theorem 2 (**H-Coefficient Technique**)**.** *Let* $\Theta = \mathsf{GoodT} \sqcup \mathsf{BadT}$ *be a partition of the set of attainable transcripts. Suppose there exists* $\epsilon_{\mathrm{ratio}} \geq 0$ *such that for any* $\tau = (\tau_c, \tau_p) \in \mathsf{GoodT}$*,*

$$\frac{\mathsf{p}_{\mathrm{re}}(\tau)}{\mathsf{p}_{\mathrm{id}}(\tau)} \triangleq \frac{\Pr[\mathsf{X}_{\mathrm{re}} = \tau]}{\Pr[\mathsf{X}_{\mathrm{id}} = \tau]} \geq 1 - \epsilon_{\mathrm{ratio}},$$

and there exists $\epsilon_{\mathrm{bad}} \geq 0$ *such that* $\Pr[\mathsf{X}_{\mathrm{id}} \in \mathsf{BadT}] \leq \epsilon_{\mathrm{bad}}$*. Then*

$$\mathbf{Adv}_{\Pi}^{\mathrm{mprf}}(\mathsf{A}) \leq \epsilon_{\mathrm{ratio}} + \epsilon_{\mathrm{bad}}. \tag{2}$$

Therefore, to prove the security of the construction using the H-coefficient technique, we need to identify the set of bad transcripts and compute an upper bound for their probability in the ideal world. Then we find a lower bound for the ratio of the real to ideal interpolation probability for a good transcript. We have already identified the bad transcripts in Fig. 4.1 and Fig. 4.2. Therefore, it only remains to bound the probability of bad transcripts in the ideal world and provide a lower bound for the ratio of the real to ideal interpolation probability for a good transcript. Having explained the H-coefficient technique in the view of our construction, it follows that for each $i \in [u]$, $\mathsf{C}_2[\mathsf{H}, \mathsf{E}]_{(L_1^i, L_2^i, K^i)} \mapsto \tau_c^i$ denotes the following:

1. $\Sigma_a^i = (\mathsf{H}^1_{L_1^i}(M_a^i)), \Theta_a^i = (\mathsf{H}^2_{L_2^i}(M_a^i))$,
2. $\mathsf{E}_{K^i}(\Sigma_a^i) = \Psi(\Sigma_a^i), \mathsf{E}_{K^i}(\Theta_a^i) = \Psi(\Theta_a^i)$, and
3. $\mathsf{E}_{K^i}(\Sigma_a^i) \oplus \mathsf{E}_{K^i}(\Theta_a^i) = T_a^i$.

4.3 Bounding the Probability of Bad Transcripts

We call a transcript $\tau = (\tau_c, \tau_p)$ **bad** if at least one of the flags is set to 1 during the generation of the transcript in Fig. 4.1 and Fig. 4.2. Recall that $\mathsf{BadT} \subseteq \Theta$ is the set of all attainable bad transcripts and $\mathsf{GoodT} = \Theta \setminus \mathsf{BadT}$ is the set of all attainable good transcripts. We bound the probability of bad transcripts in the ideal world as follows.

Lemma 2. *Let $\tau = (\tau_c, \tau_p)$ be any attainable transcript. Let X_{id} and BadT be defined as above. Then*

$$\begin{aligned}\Pr[\mathsf{X}_{\mathrm{id}} \in \mathsf{BadT}] \quad &\leq \quad \frac{q}{2^n} + \frac{2u^2}{2^{k_h+k}} + \frac{2qp\epsilon_{\mathrm{reg}}}{2^k} + \frac{q^2\epsilon_{\mathrm{univ}}}{2^n} + \frac{2q^2\epsilon_{\mathrm{reg}}}{2^{k_h}} + 3q^{4/3}\epsilon_{\mathrm{univ}} \\ &\quad + \frac{q^2\epsilon_{\mathrm{univ}}^2}{2} + \frac{2qp}{2^{n+k}} + \frac{2q^2}{2^{n+k}}.\end{aligned}$$

Proof. By abusing the notation, we refer the bad events by their corresponding flag variables as defined in Fig. 4.1, Fig. 4.2 and Fig. 4.3. That is we use Bad-Tag to refer to that event for which Bad-Tag flag has been set to 1. In other words, we say that the event Bad-Tag holds if and only if Bad-Tag flag has been set to 1. Using the union bound, we write

$$\Pr[\mathsf{X}_{\mathrm{id}} \in \mathsf{BadT}] \quad \leq \quad \Pr[\mathsf{Bad\text{-}Tag}] + \Pr[\mathsf{BadK}] + \sum_{i=1}^{4} \Pr[\mathsf{Badi} \mid \overline{\mathsf{BadK}}] + \Pr[\mathsf{Bad\text{-}Samp} \mid \overline{\mathsf{BadK}}] \quad (3)$$

We individually bound each bad event and then use Eqn. (3) to derive the result. In the subsequent analysis, we assume that $|\mathcal{K}_h| = k_h$ and $|\mathcal{K}| = k$.

4.3.1 Bounding Event Bad-Tag

For a fixed choice of indices, the probability of the event can be bound by $1/2^n$ as the outputs of the construction queries are sampled uniformly and independently of other random variables. Therefore, by summing over all possible choices of indices, we have

$$\Pr[\mathsf{Bad\text{-}Tag}] \leq \frac{q}{2^n}. \quad (4)$$

4.3.2 Bounding Event BadK

For a fixed choice of indices, the probability of the event can be bound by $1/2^{k_h+k}$ as the event $K^{i_1} = K^{i_2}$ is independent of $L_b^{i_1} = L_b^{i_2}$ for each $b \in \{1, 2\}$. Therefore, summing over all possible choices of indices, we have

$$\Pr[\mathsf{BadK}] \leq \frac{2u^2}{2^{k_h+k}}. \quad (5)$$

4.3.3 Bounding Event Bad1 | $\overline{\mathsf{BadK}}$

We say that the event $\mathsf{Bad1} \mid \overline{\mathsf{BadK}}$ holds if either of the events defined in line 5.(a) or in line 5.(b) of Fig. 4.2 holds. We refer to the event defined in line 5.(a) as $\mathsf{B.11}$ and refer to the event defined in line 5.(b) as $\mathsf{B.12}$

▷ BOUNDING $\mathsf{B.11} \mid \overline{\mathsf{BadK}}$: For a fixed choice of indices, $\Sigma^i_a = u[0]^j_\alpha$ is bound by the regular advantage of the hash function $\mathsf{H}^1_{L^i_1}$. As the hash key L^i_1 is independent of the block cipher key K^i, we have

$$\begin{aligned}
\Pr[\mathsf{B.11} \mid \overline{\mathsf{BadK}}] &\leq \sum_{\substack{i\in[u]\\ a\in[q_i]}} \sum_{\substack{j\in[s]\\ \alpha\in[p_j]}} \Pr[K^i = J^j] \cdot \Pr[\Sigma^i_a = u[0]^j_\alpha] \\
&= \sum_{\substack{i\in[u]\\ a\in[q_i]}} \sum_{\substack{j\in[s]\\ \alpha,\beta\in[p_j]}} \epsilon_{\mathrm{reg}} \cdot \frac{1}{2^k} \overset{(1)}{\leq} \frac{qp\epsilon_{\mathrm{reg}}}{2^k}, \qquad (6)
\end{aligned}$$

where (1) holds due to the fact that $(q_1 + \ldots + q_u) = q$ and $(p_1^2 + \ldots + p_s^2) \leq p^2$.

▷ BOUNDING $\mathsf{B.12} \mid \overline{\mathsf{BadK}}$: With an identical argument, one can show that the probability of the event $\mathsf{B.12}$ can be bounded by $\frac{qp\epsilon_{\mathrm{reg}}}{2^k}$, i.e.,

$$\Pr[\mathsf{B.12} \mid \overline{\mathsf{BadK}}] \leq \frac{qp\epsilon_{\mathrm{reg}}}{2^k}. \qquad (7)$$

Therefore, by combining Eqn. (6) and Eqn. (7), we have

$$\Pr[\mathsf{Bad1} \mid \overline{\mathsf{BadK}}] = \Pr[\mathsf{B.11} \mid \overline{\mathsf{BadK}} \vee \mathsf{B.12} \mid \overline{\mathsf{BadK}}] \leq \frac{2qp\epsilon_{\mathrm{reg}}}{2^k}. \qquad (8)$$

4.3.4 Bounding Event Bad2 | $\overline{\mathsf{BadK}}$

We say that the event $\mathsf{Bad2} \mid \overline{\mathsf{BadK}}$ holds if either of the events defined in line 7.(a) or in line 7.(b) or line 7.(c) or in line 7.(d) of Fig. 4.2 holds. We refer to the event defined in line 7.(a) as $\mathsf{B.21}$, in line 7.(b) as $\mathsf{B.22}$, in line 7.(c) as $\mathsf{B.23}$ and finally in line 7.(d) as $\mathsf{B.24}$

▷ BOUNDING $\mathsf{B.21} \mid \overline{\mathsf{BadK}}$: For a fixed choice of indices, we analyze the probability of the event

$$\Sigma^i_a = \Sigma^i_b \wedge T^i_a = T^i_b.$$

Due to independence of the hash key L^i_1 and T^i_a, the probability of this joint event can be bound by the universal property of the H^1 hash function and the randomness of T^i_a. Therefore,

$$\Pr[\mathsf{B.21} \mid \overline{\mathsf{BadK}}] \leq \sum_{i\in[u],\ a,b\in[q_i]} \Pr[\Sigma^i_a = \Sigma^i_b \wedge T^i_a = T^i_b] \leq \frac{q^2\epsilon_{\mathrm{univ}}}{2^{n+1}}. \qquad (9)$$

▷ BOUNDING $\mathsf{B.22} \mid \overline{\mathsf{BadK}}$: We bound the event given $\overline{\mathsf{BadK}}$, i.e. even if the block cipher keys for users i_1 and i_2 collide, their corresponding hash keys, i.e., $L^{i_1}_1$ and $L^{i_2}_2$ do not collide. Given this event, for a fixed choice of indices, we bound $\Sigma^{i_1}_a = \Sigma^{i_2}_b$ using the regular property of the hash function H^1 with the randomness of the hash key $L^{i_1}_1$. Moreover, the first event is independent of the second event and can thus be bound exactly by 2^{-k_h}. Therefore,

$$\Pr[\mathsf{B.22} \mid \overline{\mathsf{BadK}}] \leq \sum_{\substack{i_1,i_2\in[u]\\ a\in[q_{i_1}],b\in[q_{i_2}]}} \epsilon_{\mathrm{reg}} \cdot \frac{1}{2^{k_h}} \leq \frac{q^2\epsilon_{\mathrm{reg}}}{2^{k_h}}. \qquad (10)$$

▷ Bounding $\mathsf{B.23} \mid \overline{\mathsf{BadK}}$ and $\mathsf{B.24} \mid \overline{\mathsf{BadK}}$: Bounding $\mathsf{B.23} \mid \overline{\mathsf{BadK}}$ and $\mathsf{B.24} \mid \overline{\mathsf{BadK}}$ is identical to bounding $\mathsf{B.21} \mid \overline{\mathsf{BadK}}$ and $\mathsf{B.22} \mid \overline{\mathsf{BadK}}$ respectively. Hence,

$$\Pr[\mathsf{B.23} \mid \overline{\mathsf{BadK}}] \leq \frac{q^2\epsilon_{\text{univ}}}{2^{n+1}}, \quad \Pr[\mathsf{B.24} \mid \overline{\mathsf{BadK}}] \leq \frac{q^2\epsilon_{\text{reg}}}{2^{k_h}}. \tag{11}$$

Therefore, by combining Eqn. (9)-Eqn. (11),

$$\begin{aligned}\Pr[\mathsf{Bad2} \mid \overline{\mathsf{BadK}}] &\leq \Pr[\mathsf{B.21} \mid \overline{\mathsf{BadK}}] + \Pr[\mathsf{B.22} \mid \overline{\mathsf{BadK}}] + \Pr[\mathsf{B.23} \mid \overline{\mathsf{BadK}}] + \Pr[\mathsf{B.24} \mid \overline{\mathsf{BadK}}] \\ &\leq \frac{q^2\epsilon_{\text{univ}}}{2^n} + \frac{2q^2\epsilon_{\text{reg}}}{2^{k_h}}. \end{aligned} \tag{12}$$

4.3.5 Bounding Event $\mathsf{Bad3} \mid \overline{\mathsf{BadK}}$

We say that the event $\mathsf{Bad3} \mid \overline{\mathsf{BadK}}$ holds if either of the events defined in line 9.(a) or in line 9.(b) of Fig. 4.2 holds. We refer to the event defined in line 9.(a) as $\mathsf{B.31}$ and in line 9.(b) as $\mathsf{B.32}$

▷ Bounding $\mathsf{B.31} \mid \overline{\mathsf{BadK}}$ and $\mathsf{B.32} \mid \overline{\mathsf{BadK}}$: We first bound the event $\mathsf{B.31} \mid \overline{\mathsf{BadK}}$. For a fixed choice of indices, we define an indicator random variable $\mathbb{I}^i_{a,b}$ which takes the value 1 if $\Sigma^i_a = \Sigma^i_b$, and 0 otherwise. Let $\mathbb{I}^i = \sum_{a,b} \mathbb{I}^i_{a,b}$. By linearity of expectation,

$$\mathbf{E}[\mathbb{I}^i] = \sum_{a,b} \mathbf{E}[\mathbb{I}^i_{a,b}] = \sum_{a,b} \Pr[\Sigma^i_a = \Sigma^i_b] \leq \frac{q_i^2\epsilon_{\text{univ}}}{2}.$$

Now,

$$\begin{aligned}\Pr[\mathsf{B.31} \mid \overline{\mathsf{BadK}}] &\leq \sum_{i\in[u]} \Pr[|\{(a,b) \in [q_i]^2 : \Sigma^i_a = \Sigma^i_b\}| \geq q_i^{2/3}] \\ &= \sum_{i=1}^{u} \Pr[\mathbb{I}^i \geq q_i^{2/3}] \overset{(1)}{\leq} \sum_{i=1}^{u} \frac{q_i^2\epsilon_{\text{univ}}}{2q_i^{2/3}} \leq \frac{q^{4/3}\epsilon_{\text{univ}}}{2}, \end{aligned} \tag{13}$$

where (1) holds due to the Markov inequality.

Similar to $\mathsf{B.31} \mid \overline{\mathsf{BadK}}$, we bound $\mathsf{B.32} \mid \overline{\mathsf{BadK}}$ as follows:

$$\Pr[\mathsf{B.32} \mid \overline{\mathsf{BadK}}] \leq \frac{q^{4/3}\epsilon_{\text{univ}}}{2}. \tag{14}$$

Therefore, by combining Eqn. (13) and Eqn. (14), we have

$$\Pr[\mathsf{Bad3} \mid \overline{\mathsf{BadK}}] = \Pr[\mathsf{B.31} \mid \overline{\mathsf{BadK}} \vee \mathsf{B.32} \mid \overline{\mathsf{BadK}}] \leq q^{4/3}\epsilon_{\text{univ}}. \tag{15}$$

4.3.6 Bounding Event $\mathsf{Bad4} \mid \overline{\mathsf{BadK}}$

We say that the event $\mathsf{Bad4} \mid \overline{\mathsf{BadK}}$ holds if either of the events defined in line 11.(a) or in line 11.(b) or in line 11.(c) of Fig. 4.2 holds. We refer to the event defined in line 11.(a) as $\mathsf{B.41}$, line 11.(b) as $\mathsf{B.42}$ and in line 11.(c) as $\mathsf{B.43}$.

▷ Bounding $\mathsf{B.41} \mid \overline{\mathsf{BadK}}$: Due to independence of the hash key L^i_1 and L^i_2, for a fixed choice of indices, the probability of this joint event can be bound by the universal property of the individual hash functions H^1 and H^2. Therefore, varying over all possible choices of indices, we have

$$\begin{aligned}\Pr[\mathsf{B.41} \mid \overline{\mathsf{BadK}}] &\leq \sum_{\substack{i\in[u]\\ a,b\in[q_i]}} \Pr[\Sigma^i_a = \Sigma^i_b \wedge \Theta^i_a = \Theta^i_b] = \sum_{\substack{i\in[u]\\ a,b\in[q_i]}} \Pr[\Sigma^i_a = \Sigma^i_b] \cdot \Pr[\Theta^i_a = \Theta^i_b] \\ &\leq \frac{q^2\epsilon^2_{\text{univ}}}{2}. \end{aligned} \tag{16}$$

$\triangleright$ BOUNDING $\mathsf{B.42} \mid \overline{\mathsf{BadK}}$ and $\mathsf{B.43} \mid \overline{\mathsf{BadK}}$: We first bound the event $\mathsf{B.42} \mid \overline{\mathsf{BadK}}$. We bound this event given $\overline{\mathsf{B.31}}$. This results in the fact that for a fixed $i \in [u]$, the number of quadruples (a, b, c, d) such that $\Sigma_a^i = \Sigma_b^i$, $\Sigma_c^i = \Sigma_d^i$ holds is at most $q_i^{4/3}$. For a fixed choice of such quadruples, the event $\Theta_b^i = \Theta_c^i$ holds with probability at most ϵ_{univ} due to the universal property of the hash function H^2. Therefore,

$$\Pr[\mathsf{B.42} \mid \overline{\mathsf{B.31}} \wedge \overline{\mathsf{BadK}}] \leq \sum_{i \in [u]} q_i^{4/3} \epsilon_{\text{univ}} \leq q^{4/3} \epsilon_{\text{univ}}. \tag{17}$$

Similar to $\mathsf{B.42}$, we bound $\mathsf{B.43}$ as follows:

$$\Pr[\mathsf{B.43} \mid \overline{\mathsf{B.31}} \wedge \overline{\mathsf{BadK}}] \leq q^{4/3} \epsilon_{\text{univ}}. \tag{18}$$

By combining Eqn. (16), Eqn. (17) and Eqn. (18), we have

$$\Pr[\mathsf{Bad4} \mid \overline{\mathsf{BadK}}] \leq \frac{q^2 \epsilon_{\text{univ}}^2}{2} + 2q^{4/3} \epsilon_{\text{univ}}. \tag{19}$$

4.3.7 Bounding Event Bad-Samp | $\overline{\mathsf{BadK}}$

We consider bounding this event as a union of several events, namely for a fixed $i \in [u], j \in [s]$ and $a \in [q_i]$, we define

$$\mathsf{BS}_{i,j,a} \overset{\Delta}{=} K^i = J^j \wedge Z_a^i \oplus T_a^i \in \mathsf{Ran}(\mathsf{E}_{J^j}).$$

Then we say that the event $\mathsf{Bad\text{-}Samp} \mid \overline{\mathsf{BadK}}$ holds if there exists an $i \in [u]$ and $j \in [s]$ such that $\mathsf{BS}_{i,j,a}$ holds, where $Z_a^i \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{Ran}(\mathsf{E}_{J^j})$. We first fix an index $j \in [s]$, which determines $\mathcal{I}_j^{=}$, an index $i \in \mathcal{I}_j^{=}$ and $a \in [q_i]$. For this choice of indices, the probability that $K^i = J^j \wedge Z_{1,a}^i \oplus T_a^i \in \mathsf{Ran}(\mathsf{E}_{J^j})$ holds is at most $2^{-(k+n)} \cdot (p_j + q_j)$. This is due to the fact that the cardinality of $\mathsf{Ran}(\mathsf{E}_{J^j})$ is bounded above by $(p_j + q_j)$, where q_j is the number of tuples $(\Sigma_a^i, \Theta_a^i)_{i \in \mathcal{I}_j^{=}, a \in [q_i]}$ which have been added into the set $\mathsf{Dom}(\mathsf{E}_{J^j})$ such that $K^i = J^j$. Moreover, as the event $K^i = J^j$ is independent of $Z_{1,a}^i \oplus T_a^i \in \mathsf{Ran}(\mathsf{E}_{J^j})$, by taking the union bound, we have

$$\Pr[\mathsf{Bad\text{-}Samp}] \leq \sum_{j=1}^{s} \sum_{i \in \mathcal{I}_j^{=}} \sum_{a \in [q_i]} \frac{1}{2^k} \cdot \frac{p_j + q_j}{2^n - (p_j + q_j)} \leq \frac{2qp + 2q^2}{2^{n+k}}. \tag{20}$$

Note that the number of choices for (i, a) is at most q and the number of choices for j is s. Thus, summing over all possible choices of (i, j, a) and by assuming $(p_j + q_j) \leq 2^{n-1}$ and $\sum_{j=1}^{s} (p_j + q_j) \leq (p + q)$, we have the result.

Finally, the result follows by combining Eqn. (4)-Eqn. (20). $\square$

4.4 Analysis of Good Transcripts

In this section, we compute a lower bound for the ratio of the real to ideal interpolation probability for a good transcript. We first consider the set of transcripts $\mathcal{Q}^{=}$. For each $j \in [s]$ and for each $i \in \mathcal{I}_j^{=}$, we consider the sequence

$$\widetilde{\Sigma}^i := (\Sigma_1^i, \Sigma_2^i, \ldots, \Sigma_{q_i}^i), \widetilde{\Theta}^i := (\Theta_1^i, \Theta_2^i, \ldots, \Theta_{q_i}^i).$$

From this sequence, we construct a bipartite graph G_i, where the nodes in one partition represent values Σ_a^i and the nodes in other, Θ_a^i; an edge connects the nodes Σ_a^i and Θ_a^i. If $\Sigma_a^i = \Sigma_b^i$, then we merge the correponding nodes into a single node, and similarly for

$\Theta_a^i = \Theta_b^i$. This leads us to break the graph into w_i components. As the transcript is good, it is easy to see that each component is acyclic (otherwise, B.41 would have been satisfied) and contains a path of length at most 3 (otherwise either B.42 or B.43 would have been satisfied). Let v_i be the total number of nodes of the graph G_i. Similar to $\mathcal{Q}^{=}$, we consider $\mathcal{Q}^{\neq}$. For each $j \in [r']$ and for each $i \in \mathcal{I}_j^{\neq}$, consider the sequence

$$\widetilde{\Sigma}^i := (\Sigma_1^i, \Sigma_2^i, \ldots, \Sigma_{q_i}^i), \widetilde{\Theta}^i := (\Theta_1^i, \Theta_2^i, \ldots, \Theta_{q_i}^i).$$

Similar to G_i, we construct a bipartite graph H_i, one of whose partitions represents the nodes corresponding to Σ_a^i and the other, the nodes corresponding to Θ_a^i; an edge connects the nodes corresponding to Σ_a^i and Θ_a^i. If two nodes represent the same values, we merge them into a single node. Let w_i' be the number of components of H_i and v_i' be the total number of vertices. Then for a good transcript $\tau = (\tau_c, \tau_p)$, realizing τ is almost as likely in the real world as in the ideal world:

Lemma 3 (Good Lemma). *Let $\tau = (\tau_c, \tau_p) \in \mathsf{GoodT}$ be a good transcript. Let X_{re} and X_{id} be defined as above. Then*

$$\frac{\Pr[\mathsf{X}_{\mathrm{re}} = \tau]}{\Pr[\mathsf{X}_{\mathrm{id}} = \tau]} \geq 1 - \frac{9q^{4/3}}{8 \cdot 2^n} - \frac{3q^{8/3}}{2 \cdot 2^{2n}} - \frac{q^2}{2^{2n}} - \frac{9q^{7/3}}{8 \cdot 2^{2n}} - \frac{8q^4}{3 \cdot 2^{3n}}.$$

Proof. We are now ready to calculate the real interpolation probability. For this, we must bound the total number of input-output pairs on which the block cipher E with different keys is executed. As the transcript releases the $2k_h$-bit hash keys and the k-bit block cipher key for each user, it contributes to a term $2^{-(2k_h+k)}$ in the real interpolation probability calculation. Now, for each $j \in [r]$, the block cipher E with key J^j is evaluated on a total of

$$p_j + \sum_{i \in \mathcal{I}_j^{=}} v_i$$

input-output pairs. For the remaining ideal cipher keys, with which none of the users' block cipher keys have collided, we have p_j input-output pairs, which are fixed due to the evaluation of the block cipher with those ideal cipher keys. Moreover, for each $j \in [r']$, the block cipher E is evalued on a total of $\sum_{i \in \mathcal{I}_j^{\neq}} v_i'$ input-output pairs with key K^j. Summarizing the above,

$$\Pr[\mathsf{X}_{\mathrm{re}} = \tau] = \prod_{i=1}^{u} \frac{1}{2^{2k_h+k}} \cdot \left(\prod_{j=1}^{r} \frac{1}{\mathbf{P}(2^n, p_j + \sum_{i \in \mathcal{I}_j^{=}} v_i)} \right) \cdot \prod_{j \in [s] \setminus [r]} \frac{1}{\mathbf{P}(2^n, p_j)} \cdot \left(\prod_{j=1}^{r'} \frac{1}{\mathbf{P}(2^n, \sum_{i \in \mathcal{I}_j^{\neq}} v_i')} \right). \tag{21}$$

<u>Ideal Interpolation Probability:</u> The term $\prod_{i=1}^{u} 2^{-nq_i}$, which is contributed to the ideal interpolation probability due to the sampling of responses of the adversarial query, samples $2k_h$-bit hash keys and k-bit block cipher keys for all u users. For each $j \in [r]$, and for each $i \in \mathcal{I}_j^{=}$, we construct the graph G_i as defined above. It is easy to see that for each $j \in [r]$ and for each $i \in \mathcal{I}_j^{=}$, the graph G_i *good*. Next, for each $j \in [r]$ and for each $i \in \mathcal{I}_j^{=}$, we sample the value of a node for each component of the graph G_i. Hence, for $j \in [r]$, the total number of sampled points is

$$p_j + \sum_{i \in \mathcal{I}_j^{=}} w_i.$$

Moreover, for each $j \in [s] \setminus [r]$, the total number of sample points is p_j. Subsequently, we consider the set of transcripts $\mathcal{Q}^{\neq}$. For each $j \in [r']$, and for each $i \in \mathcal{I}_j^{\neq}$, we construct the graph H_i as defined above, and compute the set $\mathcal{S}_j$ for each $j \in [r']$ as defined in line 14 of Fig. 4.3 (which is defined as the number of tuples (Q^i_a, R^i_a) such that $Q^i_a \oplus R^i_a = T^i_a$ for all $i \in \mathcal{I}^{\neq}_j$ and for all $a \in [q_i]$). In summary,

$$\Pr[\mathsf{X}_{\text{id}} = \tau] = \prod_{i=1}^{u} \frac{1}{2^{nq_i}} \cdot \prod_{i=1}^{u} \frac{1}{2^{2k_h+k}} \cdot \left(\prod_{j=1}^{r} \frac{1}{\mathbf{P}(2^n, p_j + \sum\limits_{i \in \mathcal{I}^{=}_j} w_i)} \right) \cdot \prod_{j \in [s] \setminus [r]} \frac{1}{\mathbf{P}(2^n, p_j)} \cdot \left(\prod_{j=1}^{r'} \frac{1}{|\mathcal{S}_j|} \right). \tag{22}$$

<u>Calculation of the ratio:</u> By plugging in the value of $|\mathcal{S}_j|$ from Lemma 1 into Eqn. (22) and then taking the ratio of Eqn. (21) to Eqn. (22), we have

$$\begin{aligned}
\mathsf{p}(\tau) &= \prod_{i=1}^{u} 2^{nq_i} \cdot \prod_{j=1}^{r} \frac{\mathbf{P}(2^n, p_j + \sum\limits_{i \in \mathcal{I}^{=}_j} w_i)}{\mathbf{P}(2^n, p_j + \sum\limits_{i \in \mathcal{I}^{=}_j} v_i)} \cdot \prod_{j=1}^{r'} \frac{|\mathcal{S}_j|}{\mathbf{P}(2^n, \sum\limits_{i \in \mathcal{I}^{\neq}_j} v'_i)} \\
&= \prod_{i=1}^{u} 2^{nq_i} \cdot \prod_{j=1}^{r} \frac{1}{\mathbf{P}(2^n - p_j - \sum\limits_{i \in \mathcal{I}^{=}_j} w_i, \sum\limits_{i \in \mathcal{I}^{=}_j} (v_i - w_i))} \cdot \prod_{j=1}^{r'} \frac{\mathbf{P}(2^n, \sum\limits_{i \in \mathcal{I}^{\neq}_j} v'_i) \cdot \left(1 - \epsilon_j\right)}{\mathbf{P}(2^n, \sum\limits_{i \in \mathcal{I}^{\neq}_j} v'_i) \cdot 2^{n \sum\limits_{i \in \mathcal{I}^{\neq}_j} (v'_i - w'_i)}} \\
&= \prod_{i=1}^{u} 2^{nq_i} \cdot \prod_{j=1}^{r} \frac{1}{\mathbf{P}(2^n - p_j - \sum\limits_{i \in \mathcal{I}^{=}_j} w_i, \sum\limits_{i \in \mathcal{I}^{=}_j} (v_i - w_i))} \cdot \prod_{j=1}^{r'} \frac{1}{2^{n \sum\limits_{i \in \mathcal{I}^{\neq}_j} (v'_i - w'_i)}} \cdot \prod_{j=1}^{r'} \left(1 - \epsilon_j\right) \\
&= \prod_{j=1}^{r} \underbrace{\frac{2^{n \sum\limits_{i \in \mathcal{I}^{=}_j} q_i}}{\mathbf{P}(2^n - p_j - \sum\limits_{i \in \mathcal{I}^{=}_j} w_i, \sum\limits_{i \in \mathcal{I}^{=}_j} (v_i - w_i))}}_{\geq 1} \cdot \prod_{j=1}^{r'} \underbrace{\frac{2^{n \sum\limits_{i \in \mathcal{I}^{\neq}_j} q_i}}{2^{n \sum\limits_{i \in \mathcal{I}^{\neq}_j} (v'_i - w'_i)}}}_{\geq 1} \cdot \prod_{j=1}^{r'} \left(1 - \epsilon_j\right) \\
&\geq \left(1 - \sum_{j=1}^{r'} \epsilon_j\right) \geq 1 - \sum_{j=1}^{r'} \sum_{i \in \mathcal{I}^{\neq}_j} \left(\frac{9(q^{\mathsf{c}})^2_i}{8 \cdot 2^n} + \frac{3q^{\mathsf{c}}_i q_i^2}{2 \cdot 2^{2n}} + \frac{q_i^2}{2^{2n}} + \frac{9(q^{\mathsf{c}})^2_i q_i}{8 \cdot 2^{2n}} + \frac{8q_i^4}{3 \cdot 2^{3n}} \right) \\
&\overset{(1)}{\geq} 1 - \sum_{j=1}^{r'} \sum_{i \in \mathcal{I}^{\neq}_j} \left(\frac{9q_i^{4/3}}{8 \cdot 2^n} + \frac{3q_i^{8/3}}{2 \cdot 2^{2n}} + \frac{q_i^2}{2^{2n}} + \frac{9q_i^{7/3}}{8 \cdot 2^{2n}} + \frac{8q_i^4}{3 \cdot 2^{3n}} \right) \\
&\geq 1 - \left(\frac{9q^{4/3}}{8 \cdot 2^n} + \frac{3q^{8/3}}{2 \cdot 2^{2n}} + \frac{q^2}{2^{2n}} + \frac{9q^{7/3}}{8 \cdot 2^{2n}} + \frac{8q^4}{3 \cdot 2^{3n}} \right),
\end{aligned}$$

where (1) holds due to the fact that $q^{\mathsf{c}}_i \leq q_i^{2/3}$ for all $i \in \mathcal{I}^{\neq}_j$ such that $j \in [r']$. Note that for each $j \in [r]$, $\sum_{i \in \mathcal{I}^{=}_j} (v_i - w_i)$ denotes the total number of edges in the graph $\bigcup_{i \in \mathcal{I}^{=}_j} G_i$, which is $\sum_{i \in \mathcal{I}^{=}_j} q_i$. Similarly, for each $j \in [r']$, $\sum_{i \in \mathcal{I}^{\neq}_j} (v'_i - w'_i)$ denotes the total number of edges in the graph $\bigcup_{i \in \mathcal{I}^{\neq}_j} H_i$, which is $\sum_{i \in \mathcal{I}^{\neq}_j} q_i$.

5 Tight Security Bound of Two-Keyed Polyhash based DbHtS Construction

Two-keyed Polyhash-based DbHtS construction $\mathsf{C}_2[\mathsf{PH\text{-}DbH}, \mathsf{E}]$, as proposed in [13], is the instantiation of the Two-Keyed-DbHtS framework which is build on the Polyhash based double block hash function PH-DbH. In [13], the PRF security of $\mathsf{C}_2[\mathsf{PH\text{-}DbH}, \mathsf{E}]$ has been proven to be roughly in the order of $q^3\ell^2/2^{2n}$ in the single-user setting. In this section we improve its bound up to $2^{3n/4}$ queries in the multi-user setting. Moreover, the proof is based on the ideal cipher model. Before going to the security proof of the construction, we first revisit to the two-keyed Polyhash-based DbHtS construction.

PolyHash [14, 6, 34] is a very efficient algebraic hash function. For a fixed natural number n, it first samples an n-bit key L uniformly at random from $\{0,1\}^n$. To apply this function on a message $M \in \{0,1\}^*$, we first apply an injective padding function 10^* (i.e. append a bit 1 followed by a minimum number of zeroes to the message M so that the total number of bits in the padded message becomes a multiple of n). Let the padded message be $M^* = M_1 \| M_2 \| \ldots \| M_l$, where l is the number of n-bit blocks in it. Then, we define the PolyHash function as follows:

$$\mathsf{PH}_L(M^*) \triangleq M_1 \cdot L^l \oplus M_2 \cdot L^{l-1} \oplus \ldots \oplus M_l \cdot L,$$

where l is the number of blocks of M and the multiplications are defined in the field $\mathrm{GF}(2^n)$. Then Polyhash [26] is $\ell/2^n$-regular, $\ell/2^n$-axu and $\ell/2^n$-universal, as shown in the following lemma, where ℓ is the maximum number of message blocks (the proof of the lemma is related to a result on the number of distinct roots of a polynomial):

Lemma 4. *Let* PH *be the PolyHash function as defined above. Then* PH *is $\ell/2^n$-regular, $\ell/2^n$-almost-xor universal and $\ell/2^n$-universal.*

From Lemma 4, a simple corollary immediately follows:

Corollary 1. *Let $\mathsf{fix}_b(\mathsf{PH})$ be the variant of the Polyhash function in which the least significant bit of the n-bit output of the function is fixed to bit b. Then, $\mathsf{fix}_b(\mathsf{PH})$ is a $2\ell/2^n$-regular, $2\ell/2^n$-almost-xor universal and $2\ell/2^n$-universal hash function.*

We now define the Polyhash-based double-block hash function, (PH-DbH function):

$$\mathsf{PH\text{-}DbH}_{(L_1,L_2)}(M) \triangleq \Big(\underbrace{\mathsf{fix}_0(\mathsf{PH}_{L_1}(M))}_{\mathsf{H}^1_{L_1}}, \underbrace{\mathsf{fix}_1(\mathsf{PH}_{L_2}(M))}_{\mathsf{H}^2_{L_2}}\Big). \tag{23}$$

Thus, two independent instances of the Polyhash function keyed with two independent keys L_1 and L_2 are applied separately to a message M, and the least significant bit of their output is chopped and prepended with bits 0 and 1 respectively. The two-keyed PolyHash-based DbHtS construction can now be defined directly from the Two-Keyed-DbHtS construction as follows: encrypt $\mathsf{fix}_0(\mathsf{PH}_{L_1}(M))$ and $\mathsf{fix}_1(\mathsf{PH}_{L_2}(M))$ through a block cipher E_K and xor the result together to produce the output. An algorithmic description of the construction is shown in Fig. 5.1.
Clearly, the PH-DbH function is a good double-block hash function as the individual hash functions H^1 and H^2 are both $2\ell/2^n$-regular and universal. Furthermore, for a randomly chosen pair of keys L_1, L_2, and for any pair of messages $M, M' \in \{0,1\}^*$,

$$\Pr[\mathsf{fix}_0(\mathsf{PH}_{L_1}(M)) = \mathsf{fix}_1(\mathsf{PH}_{L_2}(M'))] = 0.$$

Therefore, combining the Corollary 1 with Theorem 1, we derive the following security of the two-keyed PolyHash-based DbHtS construction $\mathsf{C}_2[\mathsf{PH\text{-}DbH}, \mathsf{E}]$.

$\mathsf{C}_2[\mathsf{PH\text{-}DbH}, \mathsf{E}]_{(K_1,K_2,K)}(M)$	$\mathsf{PH}_L(M)$
$\Sigma = \mathsf{fix}_0(\mathsf{PH}_{K_1}(M));$	$M_1\\|\ldots\\|M_\ell \xleftarrow{n} M\\|10^*;$
$\Theta = \mathsf{fix}_1(\mathsf{PH}_{K_2}(M));$	$Y = M_1 \cdot L^\ell \oplus M_2 \cdot L^{\ell-1} \oplus \cdots \oplus M_\ell \cdot L;$
$T = \mathsf{E}_K(\Sigma) \oplus \mathsf{E}_K(\Theta);$	**return** Y;
return T;	

Figure 5.1: The two-keyed Polyhash-based DbHtS construction $\mathsf{C}_2[\mathsf{PH\text{-}DbH}, \mathsf{E}]$ with PH-DbH as the underlying double-block hash function. $M_1\|M_2\|\ldots\|M_\ell \xleftarrow{n} M\|10^*$ denotes the parsing of message $M\|10^*$ into n bit strings.

Theorem 3. *Let $\mathcal{K}$ be a non-empty finite set. Let $\mathsf{E} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be an n-bit block cipher and $\mathsf{PH\text{-}DbH} : (\{0,1\}^n \times \{0,1\}^n) \times \{0,1\}^* \to (\{0,1\}^n)^2$ be the PolyHash-based double-block hash function as defined above. Then any computationally unbounded distinguisher making a total of q construction queries across all u users such that each queried message is at most ℓ blocks long with $\ell \le 2^{n-2}$ and a total of p primitive queries to the block cipher E can distinguish $\mathsf{C}_2[\mathsf{PH\text{-}DbH}, \mathsf{E}]$ from an n-bit uniform random function with advantage*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{mprf}}_{\mathsf{C}_2[\mathsf{PH\text{-}DbH},\mathsf{E}]}(u,q,p,\ell) \;\le\; & \frac{9q^{4/3}}{8\cdot 2^n} + \frac{3q^{8/3}}{2\cdot 2^{2n}} + \frac{q^2}{2^{2n}} + \frac{9q^{7/3}}{8\cdot 2^{2n}} + \frac{8q^4}{3\cdot 2^{3n}} + \frac{q}{2^n} + \frac{2u^2}{2^{n+k}} \\ & + \frac{4qp\ell}{2^{n+k}} + \frac{4q^2\ell}{2^{2n}} + \frac{4q^2\ell}{2^{n+k}} + \frac{8q^{4/3}\ell}{2^n} + \frac{4q^2\ell^2}{2^{2n}} + \frac{2qp}{2^{n+k}} + \frac{2q^2}{2^{n+k}}.\end{aligned}$$

Remark 2. We would like to mention that the definition of the Polyhash function used in this paper is different from that used in [16]. Nevertheless, one can also establish the $3n/4$-bit multi-user security of the two-keyed PolyHash-based DbHtS construction with the Polyhash function used in [16].

6 Conclusion and Future Problems

In this paper, we have shown that the Two-Keyed-DbHtS construction is multi-user secured up to $2^{3n/4}$ queries in the ideal-cipher model. As an instantiation of the result, we have shown that Polyhash-based DbHtS provides $3n/4$-bit multi-user security in the ideal-cipher model. Combining it with the generic result on the attack complexity of the DbHtS construction makes the bound tight. However, we cannot apply this result to analyze the security of 2K-SUM-ECBC, 2K-PMAC_Plus and 2K-LightMAC_Plus, as their underlying DbH functions are based on block ciphers, and our proof technique does not support their security analysis in the ideal-cipher model. This is because the underlying DbH function of these constructions is build on the top of block ciphers. We believe that proving $3n/4$-bit security of the DbHtS construction based on block cipher-based double-block hash functions needs a careful study.

References

[1] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017.

[2] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 259–274. Springer, 2000.

[3] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.

[4] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-rackoff backwards: Increasing security by making block ciphers non-invertible. In *EUROCRYPT '98, Proceeding.*, pages 266–280, 1998.

[5] Mihir Bellare and Björn Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 247–276. Springer, 2016.

[6] Jürgen Bierbrauer, Thomas Johansson, Gregory Kabatianskii, and Ben J. M. Smeets. On families of hash functions via geometric codes and concatenation. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 331–342, 1993.

[7] Eli Biham. How to decrypt or even substitute des-encrypted messages in 2^{28} steps. *Inf. Process. Lett.*, 84(3):117–124, 2002.

[8] Alex Biryukov, Sourav Mukhopadhyay, and Palash Sarkar. Improved time-memory trade-offs with multiple data. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*, pages 110–127. Springer, 2005.

[9] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *EUROCRYPT 2002*, pages 384–397, 2002.

[10] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In *CHES 2007, Proceedings*, pages 450–466, 2007.

[11] Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting AES-GCM-SIV: multi-user security, faster key derivation, and better bounds. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, pages 468–499, 2018.

[12] 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 *Lecture Notes in Computer Science*, pages 293–319. Springer, 2011.

[13] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block hash-then-sum: A paradigm for constructing bbb secure prf. *IACR Transactions on Symmetric Cryptology*, 2018(3):36–92, 2018.

[14] Bert den Boer. A simple and key-economical unconditional authentication scheme. *Journal of Computer Security*, 2:65–72, 1993.

[15] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. *IACR Cryptology ePrint Archive*, 2012:600, 2012.

[16] Tingting Guo and Peng Wang. A note on the security framework of two-key dbhts macs. Cryptology ePrint Archive, Report 2022/375, 2022.

[17] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 3–32, 2016.

[18] Viet Tung Hoang and Stefano Tessaro. The multi-user security of double encryption. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, pages 381–411, 2017.

[19] Tetsu Iwata and Kaoru Kurosawa. OMAC: one-key CBC MAC. In *Fast Software Encryption, 2003*, pages 129–153, 2003.

[20] Seongkwang Kim, ByeongHak Lee, and Jooyoung Lee. Tight security bounds for double-block hash-then-sum macs. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 435–465. Springer, 2020.

[21] Gaëtan Leurent, Mridul Nandi, and Ferdinand Sibleyras. Generic attacks against beyond-birthday-bound macs. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 306–336. Springer, 2018.

[22] 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 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 575–605. Springer, 2017.

[23] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC mode for lightweight block ciphers. In *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 43–59, 2016.

[24] David A. McGrew and John Viega. The security and performance of the galois/counter mode (GCM) of operation. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

[25] M.Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac, 2007.

[26] Kazuhiko Minematsu and Tetsu Iwata. Building blockcipher from tweakable blockcipher: Extending FSE 2009 proposal. In *Cryptography and Coding - 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*, pages 391–412, 2011.

[27] 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 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 724–753. Springer, 2020.

[28] 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 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2015.

[29] Yusuke Naito. Blockcipher-based macs: Beyond the birthday bound without message length. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, volume 10626 of *Lecture Notes in Computer Science*, pages 446–470. Springer, 2017.

[30] Mridul Nandi. Birthday attack on dual ewcdm. Cryptology ePrint Archive, Report 2017/579, 2017. https://eprint.iacr.org/2017/579.

[31] Jacques Patarin. The "coefficients h" technique. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 328–345, 2008.

[32] 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 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 309–336. Springer, 2021.

[33] Victor Shoup. A composition theorem for universal one-way hash functions. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 445–452. Springer, 2000.

[34] Richard Taylor. An integrity check value algorithm for stream ciphers. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 40–48, 1993.

[35] Kan Yasuda. The sum of CBC macs is a secure PRF. In *CT-RSA 2010*, pages 366–381, 2010.

[36] Kan Yasuda. A new variant of PMAC: beyond the birthday bound. In *CRYPTO 2011*, pages 596–609, 2011.

[37] Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. 3kf9: Enhancing 3gpp-mac beyond the birthday bound. In *ASIACRYPT 2012*, pages 296–312, 2012.