Revisiting the Security of DbHtS MACs: Beyond-Birthday-Bound in the Multi-User Setting

Yaobin Shen[†], Lei Wang[‡], and Jian Weng[§]

Abstract. Double-block Hash-then-Sum (DbHtS) MACs are a class of MACs that aim for achieving beyond-birthday-bound security, including SUM-ECBC, PMAC_Plus, 3kf9 and LightMAC_Plus. Recently Datta et al. (FSE'19), and then Kim et al. (Eurocrypt'20) proved that DbHtS constructions are secure beyond birthday bound in single-user setting. However, by a generic reduction, their results degrade to (or even worse than) the birthday bound in multi-user setting.

In this work, we revisit the security of DbHtS MACs in multi-user setting. We propose a generic framework to prove beyond-birthday-bound security for DbHtS constructions. We demonstrate the usability of this framework with applications to key-reduced variants of DbHtS MACs, including 2k-SUM-ECBC, 2k-PMAC_Plus and 2k-LightMAC_Plus. Our results show that the security of these constructions will not degrade as the number of users grows. On the other hand, our results also indicate that these constructions are beyond-birthday-bound secure in both single-user and multi-user setting without additional domain separation, which are used in prior works to simplify the analysis.

Moreover, we find a severe flaw in 2kf9, which is proved to be secure beyond birthday bound by Dattal et al. (FSE'19). We can successfully forge a tag with probability 1 without making any queries. We go further to show attacks with birthday-bound complexity on several variants of 2kf9.

1 Introduction

Message Authentication Code (MAC) is a fundamental symmetric-key primitive to ensure the authenticity of data. A MAC is typically built from a blockcipher (e.g., CBC-MAC [6], OMAC [22], PMAC [11], LightMAC [27]), or from a hash function (e.g., HMAC [5], NMAC [5], NI-MAC [1]). At a high level, many of these constructions generically follow the Hash-then-PRF paradigm. Firstly, a message is mapped by a universal hash function into an n-bit string. Then, the string is processed by a fixed-input-length Pseudo-Random Function (PRF) to produce the tag. This paradigm is simple and easy to analyze because (i) it

[†] Shanghai Jiao Tong University. Email: yb_shen@sjtu.edu.cn
[‡] Shanghai Jiao Tong University. Email: wanglei_hb@sjtu.edu.cn
[§] Jinan University. Email: cryptjweng@gmail.com

does not requires nonce or extra random coins, and hence is is deterministic and stateless; (ii)the produced tag is a random string as long as the input to PRF is fresh. The security of this method is usually capped at the so-called birthday bound $2^{n/2}$, since a collision at the output of the universal hash function will lead to a forgery for the construction. However, the birthday-bound security margin might not be enough in practice, especially when a MAC is instantiated with a lightweight blockcipher such as PRESENT [12], PRINCE [13], and GIFT [2] whose block size is small. In such case, the birthday bound becomes 2^{32} as $n = 64$ and is vulnerable in certain practical applications. For example, Bhargavan and Leurent [9] have demonstrated two practical attacks that exploit collision on short blockciphers.

Double-block Hash-then-Sum Construction. To go beyond birthday-bound security, a series of blockcipher-based MACs have been proposed, including SUM-ECBC [32], PMAC_Plus [33], 3kf9 [34] and LightMAC_Plus [27]. Interestingly, all of these MACs use a similar paradigm called Double-block Hash-then Sum (shorthand for DbHtS), where a message is first mapped into a $2n$-bit string by a double-block hash function and then the two encrypted values of each n-bit half is xor-summed to generate the tag. Datta et al. [17] abstract out this paradigm and divide it into two classes: (i) three-key DbHtS constructions, where apart from the hash key, two blockcipher keys are used in the finalization phase (including SUM-ECBC, PMAC_Plus, 3kf9 and LightMAC_Plus); (ii) two-key DbHtS,where apart from the hash key, only a single blockcipher key is used in the finalization phase (including all the two-key variants, i.e., 2k-SUM-ECBC, 2k-PMAC_Plus, 2k-LightMAC_Plus and 2kf9). Under a generic framework, they prove that both three-key and two-key DbHtS constructions can achieve beyond-birthday-bound security with a bound $q^3/2^{2n}$ where q is the number of queries. Leurent et al. [25] show attacks on all three-key DbHtS constructions with a query complexity $2^{3n/4}$. Very recently, Kim et al. [24] give a tight provable bound $q^{4/3}/2^n$ for three-key DbHtS constructions.

Multi-user security. All the above beyond-birthday-bound results only consider a single user. Yet, as one of the most commonly used cryptographic primitives in practice, MACs are typically deployed in contexts with a great number of users. For instance, they are a core element of real-world security protocols such as TLS, SSH, and IPSec, which are used by major websites with billions of daily active users. A natural question is to what extent the number of users will affect the security bound of DbHtS constructions, or more specifically, can DbHtS constructions still achieve beyond-birthday-bound security in multi-user setting?

The notion of multi-user (mu) security is introduced by Biham [10] in symmetric cryptanalysis and by Bellare, Boldyreva, and Micali [4] in the context of public-key encryption. Attackers can adaptively distribute its queries across multiple users with independent key. It considers attackers who succeed as long as they can compromise at least one user among many. As evident in a series of works [3,8,14,19–21,26,29,31] evaluating how security degrades as the number of

users grows is a challenging technical problem even when the security is known in the single-user setting. Unfortunately, until now research on provable mu security for MACs has been somewhat missing. The notable exceptions are the work of Chatterjee et al. [15] and very recently Andrew et al. [28], and Bellare et al. [3]. The first two consider a generic reduction for MACs and by using which the mu security of DbHtS constructions will be capped (or even worse than) the birthday bound, which will be discussed below, The last considers a hash-function based MAC which is quite different from our focus on blockcipher-based MACs.

Suppose the number of users is u. By using the generic reduction [15,28] from singsle-user (su) security to mu security, the above beyond-birthday bound for two-key DbHtS constructions becomes

$$\frac{uq^3}{2^{2n}}$$

in the mu setting. Suppose the adversary only issue one query per user and then the security bound becomes

$$\frac{uq^3}{2^{2n}} \leq \frac{q^4}{2^{2n}},$$

which is still capped at the worrisome birthday bound. Even for three-key DbHtS constructions with a better bound $q^{4/3}/2^n$ [1] in su setting, the mu security via generic reduction becomes

$$\frac{uq^{4/3}}{2^n} \leq \frac{q^{\frac{7}{3}}}{2^n},$$

which is worse than the birthday bound $2^{n/2}$. Thus it is worth to directly analyze the mu security of DbHtS constructions instead of relying on the generic reduction.

Our contributions. We revisit the security of DbHtS constructions in the mu setting, with a focus on two-key DbHtS constructions. Two-key DbHtS constructions such as 2k-PMAC_Plus, 2k-LightMAC_Plus and 2kf9, only uses two blocckpher keys in total. Assume the length of each key is $k = n$, then to resist a similar attack like's Biham's key-collision attack on DES, two keys is the minimal number of keys to potentially achieve beyond-birthday-bound security. On the other hand, the mu security results of two-key DbHtS constructions implicitly implies the same security of three-key ones with a slight modification.

We give a generic framework to prove beyond-birthday-bound security for two-key DbHtS constructions in mu setting. Our framework is easy to use, yet can achieve much better security bound comparing with prior generic reduction method. Under this framework, one only needs to show that the abstracted double-block hash function satisfy two properties, namely ϵ_1-regular and ϵ_2-almost universal. The first property implies that for a message, the probability

[1] This term is mainly due to the usage of Markov inequality and appears in all security bounds of three-key DbHtS constructions [24].

that the hashed value equals to any pre-fixed string is small for a key uniformally chosen from the key space. The second one implies that for any two distinct messages, the probability that the two hashed values collide is small for for a key uniformally chosen from the key space. These two properties are typically inherent in the hash part of DbHtS constructions.

We demonstrate the usability of this framework with applications to two-key DbHtS. More specifically, we prove that all of 2k-SUM-ECBC, 2k-PMAC_Plus and 2k-LightMAC_Plus are still beyond-birthday-bound secure in mu setting. Our bounds are independent of the number of users, and imply that the security of two-key DbHtS constructions will not degrade as the number of users grows. On the other hand, during the proof of these three constructions, we do not rely on domain separating functions, which are used by Datta et al. [17] in these constructions to simplify the su analysis while at the meantime complicate the construction. Thus our results also indicate these three constructions are beyond-birthday-bound secure in both su and mu setting without additional domain separating functions.

Moreover, we find a severe flaw in 2kf9 in su setting, not to mention in mu setting. Datta et al. [17] prove that 2kf9 without domain separating functions are beyond-birthday-bound secure, and then based on it they claim that the other three two-key DbHtS constructions can also achieve the same security level without domain separation. However, we can successfully forge a tag with probability 1 without making any queries. The flaw is that for any single-block message, the output of 2kf9 without domain separation is always zero. One may think that if we resume domain separation in 2kf9, then it can achieve beyond-birthday security. However, our attack shows that even with domain separation, 2kf9 cannot be secure beyond birthday bound. We go further to investigate whether the common tricks help 2kf9 to tweak a blockcipher-base MAC to go beyond-birthday-bound security. Unfortunately, a similar attack with birthday-bound complexity always exists for these variants of 2kf9.

Ideal cipher model. The proofs of this paper are done in the ideal cipher model, which is common in most analyzes for mu security. In mu setting, we are particularly concerned about how local computation (that is captured by the number of ideal cipher queries) affects security, and the classical assumption that regarding a blockciphers as a PRP is not helpful in this estimation.

2 Preliminaries

Notation. Let ε denote the empty string. For an integer i, we let $\langle i \rangle_m$ denote a m-bit representation of i. For a finite set S, we let $x \xleftarrow{\$} S$ denote the uniform sampling from S and assigning the value to x. Let $|x|$ denote the length of the string x. Let $|S|$ denote the size of the set S. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with randomness r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \xleftarrow{\$} A(x_1, \ldots)$ be the result of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. For a domain Dom and a range Rng,

procedure INITIALIZE	**procedure** Eval(i, M)
$K_1, K_2, \ldots, \leftarrow\!\!\$\ \mathcal{K}$; $b \leftarrow\!\!\$\ \{0,1\}$	$Y_1 \leftarrow F(K_i, M)$; $Y_0 \leftarrow f_i(M)$
$f_1, f_2, \ldots, \leftarrow\!\!\$\ \mathrm{Func}(\mathcal{M}, \{0,1\}^n)$	**return** Y_b
procedure FINALIZE(b')	
return $(b' = b)$	

Fig. 1: Game $\mathbf{G}_F^{\mathrm{prf}}$ defining multi-user PRF security of a function F.

let $\mathrm{Func}(\mathrm{Dom}, \mathrm{Rng})$ denote the set of functions $f : \mathrm{Dom} \to \mathrm{Rng}$. For integers $1 \le a \le N$, let $(N)_a$ denote $N(N-1)\ldots(N-a+1)$.

MULTI-USER PRF. Let $F : \mathcal{K} \times \mathcal{M} \to \{0,1\}^n$ be a function. For an adversary A, let

$$\mathsf{Adv}_F^{\mathrm{prf}}(A) = 2\Pr[\mathbf{G}_F^{\mathrm{prf}}(A)] - 1 \ ,$$

be the advantage of the adversary against the multi-user PRF security of F, where game $\mathbf{G}_F^{\mathrm{prf}}$ is defined in Fig. 1. Note that for any function F of key length k, the PRF advantage is at least $pq/2^{k+2}$ by adapting Biham's key-collision attack on DES [10], where q is the number of queries and p is the number of calls to F.

THE H-COEFFICIENT TECHNIQUE. Following the notation from Hoang and Tessaro [19], it is useful to consider interactions between an adversary A with an abstract system $\mathbf{S}$ which answers A's queries. The resulting interaction can then be recorded with a transcript $\tau = ((X_1, Y_1), \ldots, (X_q, Y_q))$. Let $\mathsf{p}_{\mathbf{S}}(\tau)$ denote the probability that $\mathbf{S}$ produces τ. It is known that $\mathsf{p}_{\mathbf{S}}(\tau)$ is the description of $\mathbf{S}$ and independent of the adversary A. We say that a transcript is attainable for the system $\mathbf{S}$ if $\mathsf{p}_{\mathbf{S}}(\tau) > 0$.

We now describe the H-coefficient technique of Patarin [16, 30]. Generically, it considers an adversary that aims at distinguishing a "real" system $\mathbf{S}_1$ from an "ideal" system $\mathbf{S}_0$. The interactions of the adversary with those systems induce two transcript distributions X_1 and X_0 respectively. It is well known that the statistical distance $\mathsf{SD}(X_1, X_0)$ is an upper bound on the distinguishing advantage of A.

Lemma 1. *[16, 30] Suppose that the set of attainable transcripts for the ideal system can be partitioned into good and bad ones. If there exists $\epsilon \ge 0$ such that $\frac{\mathsf{p}_{\mathbf{S}_1}(\tau)}{\mathsf{p}_{\mathbf{S}_0}(\tau)} \ge 1 - \epsilon$ for any good transcript τ. Then*

$$\mathsf{SD}(X_1, X_0) \le \epsilon + \Pr[X_0 \textit{ is bad}] \ .$$

REGULAR AND AU HASH FUNCTION. Let $H : \mathcal{K}_h \times \mathcal{X} \to \mathcal{Y}$ be a hash function where $\mathcal{K}_h$ is the key space, $\mathcal{X}$ is the domain space and $\mathcal{Y}$ is the range space. Hash

function H is said to be ϵ_1-*regular* if for any $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$,

$$\Pr\left[\, K_h \xleftarrow{\$} \mathcal{K}_h : H_{K_h}(X) = Y \,\right] \leq \epsilon_1$$

and it is said to be ϵ_2-*almost universal* if for any distinct $X, X' \in \mathcal{X}$,

$$\Pr\left[\, K_h \xleftarrow{\$} \mathcal{K}_h : H_{K_h}(X) = H_{K_h}(X') \,\right] \leq \epsilon_2 \ .$$

Conditional Sum Permutation. We will use the following result in some proofs.

Lemma 2. *[18, Theorem 2] For any set $\mathcal{X}$ of size r and any tuple $(T_1, \ldots, T_q)$ such that each $T_i \neq 0^n$, let $U_1, \ldots, U_q, V_1, \ldots, V_q$ be $2q$ random variables sampled without replacement from $\{0,1\}^n \setminus \mathcal{X}$ and satisfy $U_i \oplus V_i = T_i$ for $1 \leq i \leq q$. Denote by S the set of tuples of these $2q$ variables. Then*

$$|S| \geq \frac{(2^n - r)_{2q}}{2^{nq}}(1 - \mu) \ .$$

where $\mu = \frac{qr^2 + 2q^2r + 2q^3}{(2^n - r - 2q)^2}$.

3 Attack on 2kf9 Construction

In this section, we will show attacks on several variants of 2kf9 construction, which is proposed by Datta et al. [17] to achieve beyond-birthday-bound security.

The 2kf9 construction. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. The 2kf9 construction is based on a blockcipher E and two keys L and K. Let fix_0 and fix_1 be two separating functions that fix the least significant bit of an n-bit string to 0 and 1 respectively. The specification of 2kf9 with domain separating function is illustrated in Fig. 2.

3.1 Attack on 2kf9 without Domain Separation

Datta et al. [17] prove that 2kf9 without domain separating function can achieve beyond-birthday-bound security. In the proof, they claim that the collision probability between Σ and Λ (without fix_0 and fix_1) is small for any message M, namely $2/2^n$. However, this claim is essentially incorrect. For any singe-block message M, the probability of Σ colliding with Λ is exactly 1, since they are both the output of blockcipher E_L with input M. Hence, for any single-block message M, $(M, 0^n)$ is always a valid forgery for this construction.

procedure 2kf9$[E](L, K, M)$
$M[1] \| \ldots \| M[\ell] \leftarrow M$; $Y_0 \leftarrow 0^n$
for $i \leftarrow 1$ **to** ℓ **do**
  $Y_i \leftarrow E_L(Y_{i-1} \oplus M[i])$
$\Sigma = Y_\ell$; $\Lambda = Y_1 \oplus Y_2 \oplus \cdots \oplus Y_\ell$
$(\Sigma, \Lambda) \leftarrow (\mathrm{fix}_0(\Sigma), \mathrm{fix}_1(\Lambda))$; $(U, V) \leftarrow (E_K(\Sigma), E_K(\Lambda))$
$T \leftarrow U \oplus V$; **return** T

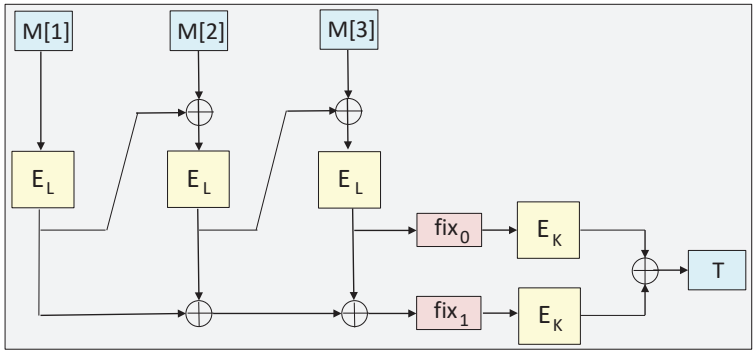

Fig. 2: The 2kf9$[E]$ construction, built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. Here fix_0 and fix_1 are two domain separating functions that fix the least significant bit of an n-bit string to 0 and 1 respectively.

3.2 Attack on 2kf9 with Domain Separation

One may think if we resume domain separation in 2kf9 (Fig. 2), then it can recover beyond-birthday-bound security. However, our attack shows that even with domain separation, 2kf9 cannot be secure beyond birthday bound.

For any two-block messages $M_1 = x \| z$ and $M_2 = y \| z \oplus 0^{n-1}1$ where $x, y \in \{0,1\}^n$, if $E_L(x) \oplus E_L(y) = 0^{n-1}1$, then $T_1 = T_2$ for any $z \in \{0,1\}^n$. The reason is as follows. For $M_1 = x \| z$, we have

$$\begin{aligned} \Sigma_1 &= \mathrm{fix}_0(E_L(z \oplus E_L(x))) \\ \Lambda_1 &= \mathrm{fix}_1(E_L(x) \oplus E_L(z \oplus E_L(x))) \ . \end{aligned}$$

Similarly, for $M_2 = y \| Z \oplus 0^{n-1}1$, we have

$$\begin{aligned} \Sigma_2 &= \mathrm{fix}_0(E_L(z \oplus 0^{n-1}1 \oplus E_L(y))) \\ \Lambda_2 &= \mathrm{fix}_1(E_L(y) \oplus E_L(z \oplus 0^{n-1}1 \oplus E_L(y))) \ . \end{aligned}$$

If $E_L(x) \oplus E_L(y) = 0^{n-1}1$, then

$$\begin{aligned} E_L(z \oplus E_L(x)) &= E_L(z \oplus 0^{n-1}1 \oplus E_L(y)) \\ E_L(x) \oplus E_L(z \oplus E_L(x)) &= E_L(y) \oplus E_L(z \oplus 0^{n-1}1 \oplus E_L(y)) \oplus 0^{n-1}1 \ . \end{aligned}$$

Obviously it holds that $\Sigma_1 = \Sigma_2$. On the other hand, due to one-bit fixing function fix_1, it also holds that $\Lambda_1 = \Lambda_2$. Hence $E_L(\Sigma_1) \oplus E_L(\Lambda_1) = E_L(\Sigma_2) \oplus E_L(\Lambda_2)$, namely $T_1 = T_2$.

The attack procedure is as follows. The adversary first chooses $2^{n/2+1}$ distinct n-bit strings $x_1, \ldots, x_{2^{n/2}}, y_1, \ldots, y_{2^{n/2}}$ from the set $\{0,1\}^n$. Fixing $z_1 \in \{0,1\}^n$, it then makes queries $x_i \parallel z_1$ and $y_i \parallel z_1 \oplus 0^{n-1}1$ to construction 2kf9, and receives the corresponding answers T_i^1 and T_i^2 for $1 \leq i \leq 2^{n/2}$. One can expect on average that there exists a pair of (x_i, y_j), such that $E_L(x_i) \oplus E_L(y_j) = 0^{n-1}1$ for $1 \leq i, j \leq 2^{n/2}$. The adversary can check it by seeing whether $T_i^1 = T_j^2$. To remove the case that $T_i^1 = T_j^2$ is not caused by $E_L(x_i) \oplus E_L(y_j) = 0^{n-1}1$, when $T_i^1 = T_j^2$ is found, the adversary can make two additional queries $x_i \parallel z_2$ and $y_j \parallel z_2 \oplus 0^{n-1}1$ to see whether the corresponding answers are identical. Finally, as soon as a desirable pair (x_i, y_j) is obtained, the adversary makes query $x_i \parallel z_3$ to receive T. Then (M, T) where $M = y_j \parallel z_3 \oplus 0^{n-1}1$ is a valid forgery. The complexity of this attack is $2^{n/2}$.

REMARK 1. If Λ is further multiplied by 2 before applying fix_1 function as is done in 2k-LightMAC_Plus and 2k-PMAC_Plus, then a similar birthday-bound attack as above also works. Instead of searching a pair of (x, y) such that $E_L(x) \oplus E_L(y) = 0^{n-1}1$ for two-block messages $M_1 = x \parallel z$ and $M_2 = x \parallel z \oplus 0^{n-1}1$, here we need to find a pair of (x, y) such that $E_L(x) \oplus E_L(y) = d$ for two-block messages $M_1 = x \parallel z$ and $M_2 = x \parallel z \oplus d$, where d is the inverse of 2 in the finite field.

REMARK 2. Even if using more complicated multiplication in Λ, i.e. $\Lambda = 2^\ell \cdot Y_1 \oplus \cdots 2 \cdot Y_\ell$ as used in 2k-LightMAC_Plus (or $\Lambda = 2 \cdot Y_1 \oplus \cdots 2^\ell \cdot Y_\ell$ as used in 2k-PMAC_Plus), then we can also propose a similar attack as above. The core idea of the attack is to find a pair of (x, y) such that $E_L(x) \oplus E_L(y) = u$ for two-block messages $M_1 = x \parallel z$ and $M_2 = y \parallel z \oplus u$, where u is the inverse of 4 in the finite field. The difference between 2kf9 and other three two-key DbHtS constructions (2k-SUM-ECBC, 2k-LightMAC_Plus, 2k-PMAC_Plus) is that for 2kf9, each Y_i depends on previous values $Y_1, \ldots, Y_{i-1}$, and we can always find some relation between variables Σ and Λ despite the usage of field multiplication. While for 2k-LightMAC_Plus and 2k-PMAC_Plus, each Y_i is generated from the corresponding message block $M[i]$, and we can prove that Σ and Λ are somewhat independent due to the usage of field multiplication. And for SUM-ECBC, the two variables Σ and Λ are generated by using two independent keys, and thus being independent of each other.

4 Multi-User Security Proof Framework for DbHtS MACs

In this section, we consider a generic proof framework for DbHtS MACs. We begin with the description of DbHtS constructions. Here we focus on two-key DbHtS constructions, including 2k-SUM-ECBC, 2k-LightMAC_Plus and 2k-PMAC_Plus.

THE DbHtS CONSTRUCTION. Let $H : \mathcal{K}_h \times \mathcal{M} \to \{0,1\}^n \times \{0,1\}^n$ be a $2n$-bit hash function with key space $\mathcal{K}_h$ and message space $\mathcal{M}$. We will always

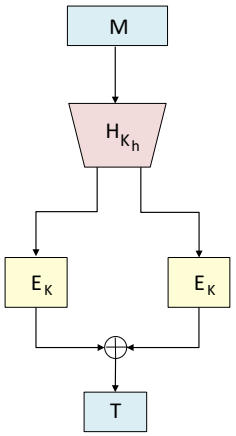

Fig. 3: **The DbHtS construction.** Here H is a $2n$-bit hash function from $\mathcal{K}_h \times \mathcal{M}$ to $\{0,1\}^n \times \{0,1\}^n$, and E is a n-bit blockcipher from $\mathcal{K} \times \{0,1\}^n$ to $\{0,1\}^n$.

decompose H into two n-bit hash function H^1 and H^2 for convenience, and thus have $H_{K_h}(M) = (H^1_{K_{h,1}}(M), H^2_{K_{h,2}}(M))$ where $K_h = (K_{h,1}, K_{h,2})$. Given a blockcipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and hash function H as defined above, one can define the DbHtS construction as follows

$$\mathsf{DbHtS}[H,E](K_h, K, M) = E_K(H^1_{K_{h,1}}(M)) \oplus E_K(H^2_{K_{h,2}}(M)) \ .$$

In blockcipher-based MACs, the hash function H is also built from an n-bit blockcipher E. The message M (after padding) is always split into n-bit blocks without being more specific, namely $M = M[1] \| M[2] \| \ldots \| M[\ell]$ where $|M[i]| = n$. For message M, we denote by $X[i]$ the i-th input to underlying blockcipher E of H.

Security analysis of DbHtS construction. Given that H is a good $2n$-bit hash function and the underlying blockcipher E is ideal, we have the following result.

Theorem 1. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we model as an ideal blockcipher. Suppose that the hash function H is ϵ_1-regular and ϵ_2-almost universal. Then for any adversary A that makes at most p ideal-cipher queries and q evaluation queries,*

$$\begin{aligned}\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{DbHtS}}(A) \leq \frac{2q}{2^k} + \frac{q(3q+p)(6q+p)}{2^{2k}} + \frac{2qp\ell}{2^{k+n}} + \frac{2qp\epsilon_1}{2^k} + \frac{4qp}{2^{n+k}} \\ + \frac{2q^2(\epsilon_1+\epsilon_2)}{2^k} + \frac{2q^2\ell\epsilon_1}{2^k} + 2q^3(\epsilon_1+\epsilon_2)^2 + \frac{8q^3(\epsilon_1+\epsilon_2)}{2^n} \\ + \frac{2qp^2 + 4qp\ell + 2q\ell^2 + 4q^2p + 4q^2\ell + 28q^3}{2^{2n}} \ .\end{aligned}$$

procedure INITIALIZE
$(K_h^1, K_1), (K_h^2, K_2), \cdots, \leftarrow\!\!\$\ \mathcal{K}_h \times \mathcal{K}$
$f_1, f_2, \cdots, \leftarrow\!\!\$\ \mathrm{Func}(\mathcal{M}, \{0,1\}^n)$
$b \leftarrow\!\!\$\ \{0,1\}$

procedure $\mathsf{Prim}(J, X)$
if $X = (+, x)$ **then return** $E_J(x)$
if $X = (-, y)$ **then return** $E_J^{-1}(y)$

procedure $\mathrm{Eval}(i, M)$
$T_1 \leftarrow \mathsf{DbHtS}[H, E](K_h^i, K_i, M)$
$T_0 \leftarrow f_i(M)$
return T_b

procedure FINALIZE(b')
return $(b' = b)$

Fig. 4: Game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{DbHtS}}$ defining multi-user prf security of the construction DbHtS.

Proof. Our proof is based on the H-coefficient technique. We will consider a computationally unbounded adversary, and without loss of generality assume that the adversary is deterministic and never repeats a prior query. Assume further that the adversary never makes a redundant query: if it queries $y \leftarrow E(J, x)$ then it won't query $E^{-1}(J, y)$ and vice versa. The security game is detailed in Fig. 4. The real system corresponds to game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{DbHtS}}$ with challenge bit $b = 1$, and the ideal system corresponds to game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{DbHtS}}$ with challenge bit $b = 0$.

SETUP. In both of the two worlds, after the adversary finishes querying, it obtains the following information:

- **Ideal-cipher queries:** For each query $\mathsf{Prim}(J, (x, +))$ with answer y, we associate it with an entry $(\mathtt{prim}, J, x, y, +)$. Similarly, for each query $\mathsf{Prim}(J, (y, -))$ with answer x, we associate it with an entry $(\mathtt{prim}, J, x, y, -)$.
- **Evaluation queries:** For each query $T \leftarrow \mathrm{Eval}(i, M)$, we associate it with an entry (eval, i, M, T).

We denote by $(\mathsf{eval}, i, M_a^i, T_a^i)$ the entry obtained when the adversary makes the a-th query to user i. Denote by ℓ_a^i the block length of M_a^i and denote by ℓ the maximal block length among these q evaluation queries. During the computation of entry $(\mathsf{eval}, i, M_a^i, T_a^i)$, we denote by Σ_a^i and Λ_a^i the internal output of hash function H, namely $\Sigma_a^i = H^1_{K_{h,1}}(M_a^i)$ and $\Lambda_a^i = H^2_{K_{h,2}}(M_a^i)$ respectively, and denote by U_a^i and V_a^i the outputs of blockcipher E with inputs Σ_a^i and Λ_a^i respectively, namely $U_a^i = E(K_i, \Sigma_a^i)$ and $V_a^i = E(K_i, \Lambda_a^i)$ respectively. For a key $J \in \{0,1\}^k$, let $P(J)$ be the set of entries $(\mathtt{prim}, J, x, y, *)$, and let $Q(J)$ be the set of entries $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_i = J$. In the real world, after the adversary finishes all its queries, we will further give it: (i) the keys (K_h^i, K_i) where $K_h^i = (K_{h,1}^i, K_{h,2}^i)$ and (ii) the internal values U_a^i and V_a^i. In the ideal world, we instead give the adversary truly random strings $(K_h^i, K_i) \leftarrow\!\!\$\ \mathcal{K}_h \times \mathcal{K}$, independent of the queries. Moreover, we give the adversary dummy values U_a^i and V_a^i computed as follows: for each set $Q(J)$, the simulation oracle $\mathrm{SIM}(Q(J))$ (depicted in Fig. 5) will be invoked and returns corresponding values U_a^i and V_a^i to the adversary. On the other hand, the internal values Σ_a^i and Λ_a^i during the computation of SIM are uniquely determined by the message M and key

(K_h^i, K_i). These additional information can only help the adversary. Thus a transcript consists of the revealed keys (K_h^i, K_i), the internal values U_a^i and V_a^i, and the ideal-cipher queries and evaluation queries.

Defining bad transcripts. We say a transcript is *bad* if one of the following happens:

1. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_i = K_{h,d}^i$ for $d \in \{1, 2\}$.
2. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that both K_i and $K_{h,d}^i$ for $d \in \{1, 2\}$ have been used in other entries, namely either in entries $(\mathsf{eval}, j, M_b^j, T_b^j)$ or entries $(\mathtt{prim}, J, x, y, *)$.
3. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_{h,d}^i = J$ for $d \in \{1, 2\}$ and $x = X_a^i[j]$ for some entry $(\mathtt{prim}, J, x, y, -)$ and some $1 \le j \le \ell_a^i$.
4. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_i = J$, and either $\Sigma_a^i = x$ or $\Lambda_a^i = x$ for some entry $(\mathtt{prim}, J, x, y, *)$.
5. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_i = J$, and either $U_a^i = y$ or $V_a^i = y$ for some entry $(\mathtt{prim}, J, x, y, *)$.
6. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_i = K_j$, and either $\Sigma_a^i = \Sigma_b^j$ or $\Sigma_a^i = \Lambda_b^j$ for some entry $(\mathsf{eval}, j, M_b^j, T_b^j)$.
7. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_i = K_j$, and either $\Lambda_a^i = \Lambda_b^j$ or $\Lambda_a^i = \Sigma_b^j$ for some entry $(\mathsf{eval}, j, M_b^j, T_b^j)$.
8. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $K_i = K_{h,1}^j$ and $\Sigma_a^i = X_b^j[k]$, or $K_i = K_{h,2}^j$ and $\Lambda_a^i = X_b^j[k]$ for some entry $(\mathsf{eval}, j, M_b^j, T_b^j)$ and $1 \le k \le \ell_b^j$.
9. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $\Sigma_a^i = \Sigma_b^i$ or $\Sigma_a^i = \Lambda_b^i$, and either $\Lambda_a^i = \Lambda_b^i$ or $\Lambda_a^i = \Sigma_b^i$ for some entry $(\mathsf{eval}, i, M_b^i, T_b^i)$.
10. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $\Sigma_a^i = \Sigma_b^i$ or $\Sigma_a^i = \Lambda_b^i$, and either $V_a^i = V_b^i$ or $V_a^i = U_b^i$ for some entry $(\mathsf{eval}, i, M_b^i, T_b^i)$.
11. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $\Lambda_a^i = \Lambda_b^i$ or $\Lambda_a^i = \Sigma_b^i$, and either $U_a^i = U_b^i$ or $U_a^i = V_b^i$ for some entry $(\mathsf{eval}, i, M_b^i, T_b^i)$.
12. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $\Sigma_a^i = \Sigma_b^i$ or $\Sigma_a^i = \Lambda_b^i$, and either $\Lambda_a^i = \Lambda_c^i$ or $\Lambda_a^i = \Sigma_c^i$ for some entries $(\mathsf{eval}, i, M_b^i, T_b^i)$ and $(\mathsf{eval}, i, M_c^i, T_c^i)$.
13. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $\Sigma_a^i = \Sigma_b^i$ or $\Sigma_a^i = \Lambda_b^i$, and either $V_a^i = V_c^i$ or $V_a^i = U_c^i$ for some entries $(\mathsf{eval}, i, M_b^i, T_b^i)$ and $(\mathsf{eval}, i, M_c^i, T_c^i)$.
14. There is an entry $(\mathsf{eval}, i, M_a^i, T_a^i)$ such that $\Lambda_a^i = \Lambda_b^i$ or $\Lambda_a^i = \Sigma_b^i$, and either $U_a^i = U_c^i$ or $U_a^i = V_c^i$ for some entries $(\mathsf{eval}, i, M_b^i, T_b^i)$ and $(\mathsf{eval}, i, M_c^i, T_c^i)$.

If a transcript is not bad then we say it's *good*. Note that the goal of these conditions is to ensure that (i) for queries to the same user, at least one of two inputs to blockcipher E is fresh; (ii) for queries to different users, if the key of blockcipher E collides, then the input to E should be different. We briefly comments on these conditions. Condition (1) and (2) are to guarantee that at least one of two keys of any user i is fresh. Note in blockcipher-based MACs, hash function is usually based on blockcipher E. Condition (3) is to prevent that the adversary can somehow control the (partial) output of $H_{K_h}(M)$ by using its backward ideal-cipher queries for some $1 \le j \le \ell_a^i$ where $M_a^i = M_a^i[1] \parallel \ldots \parallel M_a^i[\ell_a^i]$ and

$X_a^i[k]$ is the k-th corresponding input to underlying blockcipher of H. Condition (4) and (5) are to remove the case that either the inputs or outputs of E_{K_i} collide with those in the ideal cipher queries. Condition (6) and (7) are to guarantee that when the key K_i and K_j of user i and j collides, then all the inputs of E_{K_i} are distinct, and all the outputs of E_{K_i} are also distinct. Condition (8) is to guarantee that when there is a collision between K_i and $K_{h,d}^i$ for $d \in \{1,2\}$, then the input to E_{K_i} do not collide with the input in the hash part with key $K_{h,d}^i$, and thus keep the freshness of the final output. Condition (9) is to guarantee that for any pair of entries $(\mathsf{eval}, i, M_a^i, T_a^i)$ and $(\mathsf{eval}, i, M_b^i, T_b^i)$, at least one of Σ_a^i and Λ_a^i is fresh. Condition (10) and (11) are to guarantee that the outputs of Φ_{K_i} are compatible with the permutation in the ideal world, namely when the inputs are distinct, then the corresponding outputs should also be distinct. Condition (12) is to guarantee that for any triple of entries $(\mathsf{eval}, i, M_a^i, T_a^i)$, $(\mathsf{eval}, i, M_b^i, T_b^i)$ and $(\mathsf{eval}, i, M_c^i, T_c^i)$, at least one of Σ_a^i and Λ_a^i is fresh. Condition (11) and (12) are to guarantee that the outputs of Φ_{K_i} are compatible with the permutation in the ideal world, namely when the inputs are distinct, then the corresponding outputs should also be distinct. Let X_1 and X_0 be the random variables for the transcripts in the real and ideal system respectively.

Probability of bad transcripts. We now bound the chance that X_0 is bad. Let Bad_i be the event that X_0 violates the i-th condition. By the union bound,

$$\begin{aligned}\Pr[X_0 \text{ is bad}] &= \Pr[\mathrm{Bad}_1 \vee \cdots \vee \mathrm{Bad}_{11}] \\ &\leq \sum_{i=1}^{3} \Pr[\mathrm{Bad}_i] + \sum_{i=4}^{8} \Pr[\mathrm{Bad}_i \mid \overline{\mathrm{Bad}_3}] + \sum_{i=9}^{14} \Pr[\mathrm{Bad}_i] \ .\end{aligned}$$

We first bound the probability $\Pr[\mathrm{Bad}_1]$. Recall that in the ideal world, K_i and $K_{h,d}^i$ are uniformly random, independent of each other and those entries. Thus the chance that $K_i = K_{h,d}^i$ is at most $1/2^k$. Summing over at most q evaluation queries and $d \in \{1,2\}$,

$$\Pr[\mathrm{Bad}_1] \leq \frac{2q}{2^k} \ .$$

Next, we bound the probability $\Pr[\mathrm{Bad}_2]$. Recall that in the ideal world, K_i and $K_{h,d}^i$ are uniformly random, independent of each other and those entries. Thus the probability that $K_i = K_j$ or $K_i = K_{h,d'}^j$ for at most $q-1$ other users or $K_i = J$ for at most p ideal-cipher queries is at most $(3q+p)/2^k$. The probability that $K_{h,d}^i = K_j$ or $K_{h,d}^i = K_{h,d'}^j$ for at most $q-1$ other users or $K_{h,d}^i = J$ for at most p ideal-cipher queries is also at most $(6q+p)/2^k$. Since K_i and $K_{h,d}^i$ are independent of each other, and summing over at most q evaluation queries,

$$\Pr[\mathrm{Bad}_2] \leq \frac{q(3q+p)(6q+p)}{2^{2k}} \ .$$

Next, we bound the probability $\Pr[\mathrm{Bad}_3]$. Recall that in the ideal world, K_i is uniformly random, independent of those entries. Thus the chance that $K_i = J$

for at most p ideal-cipher queries is at most $p/2^k$. On the other hand, for each ideal-cipher entry $(\mathtt{prim}, J, x, y, -)$, the probability that $x = X_a^i[j]$ is at most $1/(2^n - p - q\ell) \leq 2/2^n$. Summing over at most q evaluation queries and $1 \leq j \leq \ell_a^i \leq \ell$,

$$\Pr[\mathrm{Bad}_3] \leq \frac{2qp\ell}{2^{k+n}} \ .$$

Next, we bound the probability $\Pr[\mathrm{Bad}_4 \mid \overline{\mathrm{Bad}_2}]$. Recall that in the ideal world, K_i is uniformly random, independent of those entries. Thus for each entry $(\mathtt{prim}, J, x, y, *)$, the chance that $K_i = J$ is $1/2^k$. On the other hand, conditioned on $\overline{\mathrm{Bad}_2}$, the key $K_{h,d}^i$ is fresh for $d \in \{1, 2\}$. The event that $\Sigma_a^i = x$ or $\Lambda_a^i = x$ is the same as

$$H^1_{K^i_{h,1}}(M_a^i) = x \ \vee \ H^2_{K^i_{h,2}}(M_a^i) = x \ ,$$

which holds with probability at most $2\epsilon_1$. Summing over at most q evaluation queries and p ideal-cipher queries,

$$\Pr[\mathrm{Bad}_4 \mid \overline{\mathrm{Bad}_2}] \leq \frac{2qp\epsilon_1}{2^k} \ .$$

Bounding the probability $\Pr[\mathrm{Bad}_5 \mid \overline{\mathrm{Bad}_2}]$ is similar to handling $\Pr[\mathrm{Bad}_4 \mid \overline{\mathrm{Bad}_2}]$, but now the event $U_a^i = y$ or $V_a^i = y$ is the same as $\Phi_{K_i}(\Sigma_a^i) = y$ or $\Phi_{K_i}(\Lambda_a^i) = y$. If $\Sigma_a^i \in \mathrm{Dom}(\Phi_{K_i})$, then the probability that $\Phi_{K_i}(\Sigma_a^i) = y$ is at most $1/(2^n - \ell - p) \leq 2/2^n$; If $\Sigma_a^i \notin \mathrm{Dom}(\Phi_{K_i})$, then the probability that $T_a^i \oplus V_a^i = y$ is at most $1/2^n$ since T_a^i is a random string. Hence the probability that $\Phi_{K_i}(\Sigma_a^i) = y$ is at most $2/2^n$. Similarly, the probability that $\Phi_{K_i}(\Lambda_a^i) = y$ is at most $2/2^n$. Thus,

$$\Pr[\mathrm{Bad}_5 \mid \overline{\mathrm{Bad}_2}] \leq \frac{4qp}{2^{n+k}} \ .$$

We now bound the probability $\Pr[\mathrm{Bad}_6 \mid \overline{\mathrm{Bad}_2}]$. Recall that in the ideal world, K_i is uniformly random, independent of those entries. Thus the chance that $K_i = K_j$ is $1/2^k$ for $i \neq j$. On the other hand, conditioned on $\overline{\mathrm{Bad}_2}$, the key $K_{h,1}^i$ is fresh. The event that $\Sigma_a^i = \Sigma_b^j$ is the same as

$$H^1_{K^i_{h,1}}(M_a^i) = H^1_{K^j_{h,1}}(M_b^j)$$

which holds with probability at most ϵ_2. Similarly, the event that $\Sigma_a^i = \Lambda_b^j$ holds with probability at most ϵ_1. Summing over at most q^2 pairs of i and j,

$$\Pr[\mathrm{Bad}_6 \mid \overline{\mathrm{Bad}_2}] \leq \frac{q^2(\epsilon_1 + \epsilon_2)}{2^k} \ .$$

Bounding $\Pr[\mathrm{Bad}_7 \mid \overline{\mathrm{Bad}_2}]$ is similar to handling $\Pr[\mathrm{Bad}_6 \mid \overline{\mathrm{Bad}_2}]$, and thus

$$\Pr[\mathrm{Bad}_7 \mid \overline{\mathrm{Bad}_2}] \leq \frac{q^2(\epsilon_1 + \epsilon_2)}{2^k} \ .$$

Next, we bound the probability $\Pr[\mathrm{Bad}_8]$. Recall that in the ideal world, K_i is uniformly random, independent of those entries. Thus the chance that $K_i = K^j_{h,1}$ for some other j is at most $1/2^k$. On the other hand, for each other entry $(\mathsf{eval}, j, M^j_b, T^j_b)$, the probability that $\Sigma^i_a = X^j_b[k]$ is at most ϵ_1. Hence the chance that $K_i = K^j_{h,1}$ and $\Sigma^i_a = X^j_b[k]$ is at most $\epsilon_1/2^k$. Similarly, the probability that $K_i = K^j_{h,2}$ and $\Lambda^i_a = X^j_b[k]$ is also at most $\epsilon_1/2^k$. Summing over at most q^2 pairs of evaluation queries and $1 \le k \le \ell$,

$$\Pr[\mathrm{Bad}_8] \le \frac{2q^2\ell\epsilon_1}{2^k} \ .$$

Next, we bound the probability $\Pr[\mathrm{Bad}_9]$. The event $\Sigma^i_a = \Sigma^i_b$ or $\Sigma^i_a = \Lambda^i_b$ is the same as

$$H^1_{K^i_{h,1}}(M^i_a) = H^1_{K^i_{h,1}}(M^i_b) \vee H^1_{K^i_{h,1}}(M^i_a) = H^2_{K^i_{h,2}}(M^i_b) \ ,$$

which holds with probability at most $\epsilon_1 + \epsilon_2$. Similarly, the probability of the event $\Lambda^i_a = \Sigma^i_b$ or $\Lambda^i_a = \lambda^i_b$ is at most $\epsilon_1 + \epsilon_2$. Note that for each user i, there are at most q_i^2 pairs of (a, b). By the assumption that $K^i_{h,1}$ and $K^2_{h,2}$ are two independent keys, and summing among u users,

$$\Pr\left[\,\mathrm{Bad}_9\,\right] \le \sum_{i=1}^{u} q_i^2(\epsilon_1 + \epsilon_2)^2 \le q^2(\epsilon_1 + \epsilon_2)^2 \ .$$

Next, we bound the probability $\Pr[\mathrm{Bad}_{10}]$. The event $\Sigma^i_a = \Sigma^i_b$ or $\Sigma^i_a = \Lambda^i_b$ is the same as

$$H^1_{K^i_{h,1}}(M^i_a) = H^1_{K^i_{h,1}}(M^i_b) \vee H^1_{K^i_{h,1}}(M^i_a) = H^2_{K^i_{h,2}}(M^i_b) \ ,$$

which holds with probability at most $\epsilon_1 + \epsilon_2$. On the other hand, the event $V^i_a = V^i_b$ or $V^i_a = U^i_b$ is the same as

$$T^i_a \oplus U^i_a = V^i_b \vee T^i_a \oplus U^i_a = U^i_b \ ,$$

which holds with probability at most $2/2^n$ since T^i_a is a random string and independent of these entries. Summing among u users,

$$\Pr\left[\,\mathrm{Bad}_{10}\,\right] \le \sum_{i=1}^{u} \frac{2q_i^2(\epsilon_1 + \epsilon_2)}{2^n} \le \frac{2q^2(\epsilon_1 + \epsilon_2)}{2^n} \ .$$

Bounding the probability $\Pr[\mathrm{Bad}_{11}]$ is similar to handling $\Pr[\mathrm{Bad}_{10}]$, and thus

$$\Pr\left[\,\mathrm{Bad}_{11}\,\right] \le \frac{2q^2(\epsilon_1 + \epsilon_2)}{2^n} \ .$$

Bounding the probability $\Pr[\mathrm{Bad}_{12}]$ is similar to handling $\Pr[\mathrm{Bad}_9]$, except that now for each user i, there are at most q_i^3 tuples of (a, b, c). Hence summing among these u users,

$$\Pr\left[\,\mathrm{Bad}_{12}\,\right] \le \sum_{i=1}^{u} q_i^3(\epsilon_1 + \epsilon_2)^2 \le q^3(\epsilon_1 + \epsilon_2)^2 \ .$$

Bounding the probability $\Pr[\mathrm{Bad}_{13}]$ is similar to handling $\Pr[\mathrm{Bad}_{10}]$, except that now for each user i, there are at most q_i^3 tuples of (a, b, c). Hence summing among these u users,

$$\Pr[\,\mathrm{Bad}_{13}\,] \le \sum_{i=1}^{u} \frac{2q_i^3(\epsilon_1+\epsilon_2)}{2^n} \le \frac{2q^3(\epsilon_1+\epsilon_2)}{2^n} \quad .$$

Bounding the probability $\Pr[\mathrm{Bad}_{14}]$ is similar to handling $\Pr[\mathrm{Bad}_{13}]$, and thus

$$\Pr[\mathrm{Bad}_{14}] \le \frac{2q^3(\epsilon_1+\epsilon_2)}{2^n} \quad .$$

Summing up,

$$\begin{aligned}\Pr[X_0 \text{ is bad}] \le{}& \frac{2q}{2^k} + \frac{q(3q+p)(6q+p)}{2^{2k}} + \frac{2qp\ell}{2^{k+n}} + \frac{2qp\epsilon_1}{2^k} + \frac{4qp}{2^{n+k}} \\ &+ \frac{2q^2(\epsilon_1+\epsilon_2)}{2^k} + \frac{2q^2\ell\epsilon_1}{2^k} + 2q^3(\epsilon_1+\epsilon_2)^2 + \frac{8q^3(\epsilon_1+\epsilon_2)}{2^n} \quad . (1)\end{aligned}$$

Transcript ratio. Let τ be a good transcript. Denote by p_J and q_J the size of set $P(J)$ and $Q(J)$ respectively. Denote by f_J the size of $F(J)$and denote by s_J the size of set $S(J)$. Denote by ℓ_J the number of underlying blockcipher queries appear in the hash part when the value of secret key is J. Note that among the set $H(J)$, there are exactly $q_J + f_J$ fresh values, and $q_J - f_J$ non-fresh values. For the entries in $G(J)$, suppose that there are g_J classes among the values Σ_a^i and Λ_a^i: the elements in the same class either connected by a value T_a^i such that $\Sigma_a^i \oplus \Lambda_a^i = T_a^i$, or connected by the equation such that $\Sigma_a^i = \Sigma_b^j$ or $\Lambda_a^i = \Lambda_b^j$. Note that each class contains at least three elements, and only has one sample value in sim Fig. 5. Since τ is good, the corresponding samples U_a^i and V_a^i of these g_J distinct classes are compatible with the permutation, namely these g_J outputs are sampled in a manner such that they are distinct and do not collide with the corresponding values during the computation of the set $F(J)$.

Suppose that this transcript contains exactly u users. Then in the ideal world, since τ is good,

$$\begin{aligned}&\Pr[X_0 = \tau] \\ &= 2^{-2uk} \cdot 2^{-qn} \prod_{J\in\{0,1\}^k} \prod_{i=0}^{p_J-1} \frac{1}{2^n - i} \cdot \frac{1}{s_J} \cdot \frac{1}{(2^n - 2f_J - p_J - \ell_J)_{g_J}} \quad .\end{aligned}$$

On the other hand, in the real world, the number of permutation outputs that we need to consider for each $J \in \{0,1\}^k$ is exactly $q_J + f_J + g_J$. The reason is that, we have $q_J + f_J$ fresh input-output tuples in total, and for each class in $G(J)$, we have one additional input-output tuple. Thus,

$$\Pr[X_1 = \tau] = 2^{-2uk} \prod_{J\in\{0,1\}^k} \prod_{i=0}^{p_J-1} \frac{1}{2^n - i} \cdot \frac{1}{(2^n - p_J - \ell_J)_{q_J+f_J+g_J}} \quad .$$

Hence,

$$\begin{aligned}
\frac{\Pr[X_1=\tau]}{\Pr[X_0=\tau]} &= 2^{qn}\prod_{J\in\{0,1\}^k}\frac{s_J\cdot(2^n-2f_J-p_J-\ell_J)_{g_J}}{(2^n-p_J-\ell_J)_{q_J+f_J+g_J}}\\
&\geq \prod_{J\in\{0,1\}^k}\frac{2^{q_Jn}(2^n-2f_J-p_J-\ell_J)_{g_J}(2^n-p_J-\ell_J)_{2f_J}}{(2^n-p_J-\ell_J)_{q_J+f_J+g_J}\cdot 2^{f_Jn}}\\
&\quad\cdot(1-\frac{q_J(p_J+\ell_J)^2+2q_J^2(p_J+\ell_J)+2q_J^3}{(2^n-p_J-2q_J-\ell_J)^2})\\
&\geq \prod_{J\in\{0,1\}^k}\frac{2^{n(q_J-f_J)}}{(2^n-p_J-2f_J-g_J-\ell_J)_{q_J-f_J}}\\
&\quad\cdot(1-\frac{2q_Jp^2+4q_Jp\ell+2q_J\ell^2+4q_Jqp+4q_Jq\ell+4q_Jq_2}{2^{2n}})\\
&\geq 1-\frac{2qp^2+4qp\ell+2q\ell^2+4q^2p+4q^2\ell+4q^3}{2^{2n}}\ , \qquad (2)
\end{aligned}$$

where the first inequality comes from Lemma 2.

Wrapping up. From Lemma 1 and Equations (1) and (2), we conclude that

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{DbHtS}}(A) \leq\ & \frac{2q}{2^k}+\frac{q(3q+p)(6q+p)}{2^{2k}}+\frac{2qp\ell}{2^{k+n}}+\frac{2qp\epsilon_1}{2^k}+\frac{4qp}{2^{n+k}}\\
&+\frac{2q^2(\epsilon_1+\epsilon_2)}{2^k}+\frac{2q^2\ell\epsilon_1}{2^k}+2q^3(\epsilon_1+\epsilon_2)^2+\frac{8q^3(\epsilon_1+\epsilon_2)}{2^n}\\
&+\frac{2qp^2+4qp\ell+2q\ell^2+4q^2p+4q^2\ell+28q^3}{2^{2n}}\ .
\end{aligned}$$

5 Multi-user Security of Three Constructions

In this section, we demonstrate the usability of mu proof framework with applications to key-reduced DbHtS MACs, and prove that 2k-SUM-ECBC, 2k-LightMAC_Plus and 2k-PMAC_Plus are secure beyond-birthday-bound in mu setting.

5.1 Security of 2k-SUM-ECBC

The $2n$-bit hash function used in 2k-SUM-ECBC is simply the concatenation of two CBC MACs with two independent keys $K_{h,1}$ and $K_{h,2}$. Let $E:\{0,1\}^k\times\{0,1\}^n\to\{0,1\}^n$ be a blockcipher. For a message $M=M[1]\,\|\,M[2]\,\|\,\ldots\,\|\,M[\ell]$ where $|M[i]|=n$, the CBC MAC algorithm $\mathsf{CBC}[E](K,M)$ is defined as Y_ℓ, where

$$Y_i=E_K(M[i]\oplus Y_{i-1})$$

for $i=1,\ldots,\ell$ and $Y_0=0^n$. Then 2k-SUM-ECBC is defined as $\mathsf{DbHtS}[H,E]$, where

$$H_{K_h}(M)=(H^1_{K_{h,1}}(M),H^2_{K_{h,2}}(M))=(\mathsf{CBC}[E](K_{h,1},M),\mathsf{CBC}[E](K_{h,2},M))\ ,$$

procedure SIM($Q(J)$)
$\forall\,(\mathsf{eval}, i, M_a^i, T_a^i) \in Q(J) : (\Sigma_a^i, \Lambda_a^i) \leftarrow H_{K_h}(M_a^i)$
$I(J) = \{(i,a) : 1 \le i \le u, 1 \le a \le q_i, (\mathsf{eval}, i, M_a^i, T_a^i) \in Q(J)\}$
$H(J) = \{(\Sigma_a^i, \Lambda_a^i) : (i,a) \in I(J)\}$
$F(J) = \{(i,a) :$ both Σ_a^i and Λ_a^i are fresh in $H(J)\}$; $f = |F(J)|$
$G(J) = \{(i,a) :$ only one of Σ_a^i and Λ_a^i is fresh in $H(J)\}$
$R(J) = \{(i,a) :$ neither Σ_a^i nor Λ_a^i is fresh in $H(J)\}$
$O(J)$: the set of tuples of $2f$ distinct values from $\{0,1\}^n \setminus \mathrm{Rng}(\Phi_J)$
$S(J) = \{(W_a^i, X_a^i)_{(i,a)\in F(J)} \in O(J) : W_a^i \oplus X_a^i = T_a^i\}$
$(U_a^i, V_a^i)_{(i,a)\in F(J)} \xleftarrow{\$} S(J)$
$\forall\,(i,a) \in F(J) : (\Phi_J(\Sigma_a^i), \Phi_J(\Lambda_a^i)) \leftarrow (U_a^i, V_a^i)$
$\forall\,(i,a) \in G(J) :$
  if Σ_a^i is not fresh in H **then**
    if $\Sigma_a^i \notin \mathrm{Dom}(\Phi_J)$
      then $U_a^i \xleftarrow{\$} \{0,1\}^n \setminus \mathrm{Rng}(\Phi_J)$; $\Phi_J(\Sigma_a^i) \leftarrow U_a^i$
    else $U_a^i \leftarrow \Phi_J(\Sigma_a^i)$
    $V_a^i \leftarrow T_a^i \oplus U_a^i$
  else
    if $\Lambda_a^i \notin \mathrm{Dom}(\Phi_J)$
      then $V_a^i \xleftarrow{\$} \{0,1\}^n \setminus \mathrm{Rng}(\Phi_J)$; $\Phi_J(\Lambda_a^i) \leftarrow V_a^i$
    else $V_a^i \leftarrow \Phi_J(\Lambda_a^i)$
    $U_a^i \leftarrow T_a^i \oplus V_a^i$
$\forall\,(a,i) \in R(J) :$
  if $\Sigma_a^i \notin \mathrm{Dom}(\Phi_J)$
    then $U_a^i \xleftarrow{\$} \{0,1\}^n \setminus \mathrm{Rng}(\Phi_J)$; $\Phi_J(\Sigma_a^i) \leftarrow U_a^i$
  else $U_a^i \leftarrow \Phi_J(\Sigma_a^i)$; $V_a^i \leftarrow T_a^i \oplus U_a^i$
return $(U_a^i, V_a^i)_{(a,i)\in I}$

Fig. 5: **Offline oracle in the ideal world. For each J, Φ_J is a partial function that used to simulate a random permutation. The domain and range of Φ_J are initialized to be the domain and range of E_J respectively.**

and $K_{h,1}$ and $K_{h,2}$ are two independent keys. The specification of 2k-SUM-ECBC is illustrated in Fig. 6. For any two distinct messages M_1 and M_2 of at most $\ell \le 2^{n/4}$ blocks, Bellare et al. [7, 23] show that

$$\Pr\left[\,\mathsf{CBC}[E](K, M_1) = \mathsf{CBC}[E](K, M_2)\,\right] \le \frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}} \quad .$$

This directly implies that CBC MAC is ϵ_2-almost universal where $\epsilon_2 = \frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}}$.

Below we prove that CBC MAC is ϵ_1-regular, where $\epsilon_1 = \epsilon_2 = \frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}}$.

Lemma 3. *For any $X \in \{0,1\}^{\ell n}$ and $Y \in \{0,1\}^n$, we have*

$$\Pr\left[\,\mathsf{CBC}[E](K, X) = Y\,\right] \le \frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}} \quad .$$

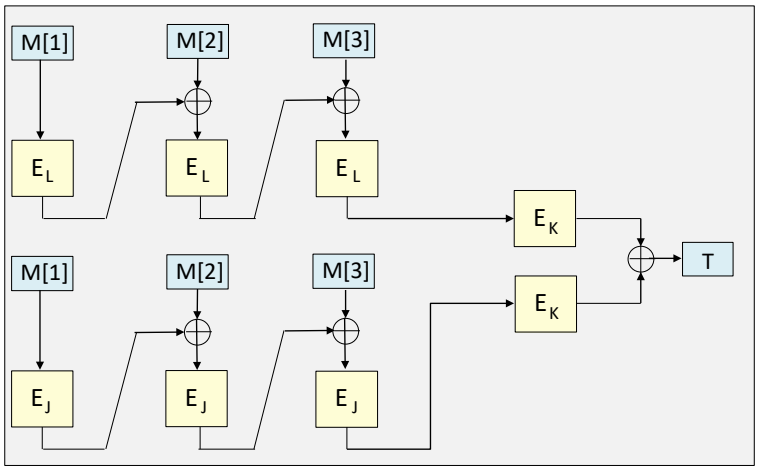

Fig. 6: 2k-SUM-ECBC MAC built from a blockcipher E.

Proof. Let $M_1 = X \,\|\, Y$ and $M_2 = 0^n$. Then the event $\mathsf{CBC}[E](K, X) = Y$ is the same as $\mathsf{CBC}[E](K, M_1) = \mathsf{CBC}[E](K, M_2)$. Hence

$$\Pr\left[\,\mathsf{CBC}[E](K,X) = Y\,\right] = \Pr\left[\,\mathsf{CBC}[E](K,M_1) = \mathsf{CBC}[E](K,M_2)\,\right] \leq \frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}} \enspace ,$$

where the last inequality comes from the fact that CBC MAC is ϵ_2-almost universal.

By using Theorem 1, we obtain the following result.

Theorem 2. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we model as an ideal blockcipher. Assume that $\ell \leq 2^{n/4}$. Then for any adversary A that makes at most p ideal-cipher queries and q evaluation queries,*

$$\begin{aligned}\mathsf{Adv}^{\mathrm{prf}}_{\text{2k-SUM-ECBC}}(A) \leq{}& \frac{2q}{2^k} + \frac{q(3q+p)(6q+p)}{2^{2k}} + \frac{38qp\ell}{2^{k+n}} + \frac{108q^2\ell^2}{2^{k+n}} + \frac{4qp}{2^{n+k}} \\ &+ \frac{2880q^3\ell}{2^{2n}} + \frac{2qp^2 + 4qp\ell + 2q\ell^2 + 4q^2p + 4q^2\ell + 28q^3}{2^{2n}} \enspace .\end{aligned}$$

5.2 Security of 2k-LightMAC_Plus

The $2n$-bit hash function H used in 2k-LightMAC_Plus is the concatenation of two functions H^1 and H^2 where H^1 and H^2 are both based on a blockcipher E with the same key, namely $K_{h,1} = K_{h,2} = L$. For a message $M = M[1] \| \ldots \| M[\ell]$ where $M[i]$ is a $(n-m)$-bit block, $H^1_L(M)$ and $H^2_L(M)$ are defined as follows

$$\begin{aligned}H^1_L(M) &= E_L(Y_1) \oplus \cdots \oplus E_L(Y_\ell) \enspace , \\ H^2_L(M) &= 2^\ell \cdot E_L(Y_1) \oplus E_L(Y_2) \oplus \cdots \oplus 2 \cdot E_L(Y_\ell)\end{aligned}$$

where $Y_i = \langle i \rangle_m \| M[i]$ and $\langle i \rangle_m$ represents the m-bit encoding of integer i. The description of hash function H is illustrated in the top of Fig. 7. Then 2k-LightMAC_Plus is defined as $\mathsf{DbHtS}[H, E]$ and is illustrated in the bottom of Fig. 7. To prove that H^1 and H^2 are both ϵ_1-regular and ϵ_2-almost universal, we will use the following algebraic result, the proof of which can be found in [17].

Lemma 4. *[17] Let $Z = (Z_1, \ldots, Z_\ell)$ be q random variables that sampled from $\{0,1\}^n$ without replacement. Let A be a matrix of dimension $s \times \ell$ defined over $\mathsf{GF}(2^n)$. Then for any given column vector c of dimension $s \times 1$ over $\mathsf{GF}(2^n)$,*

$$\Pr[A \cdot Z^T = c] \leq \frac{1}{(2^n - \ell + r)_r} \ ,$$

where r is the rank of the matrix A.

We first show that H^1 is ϵ_1-regular. Note that for any message M and any n-bit string $Y \in \{0,1\}^n$, the rank of equation

$$E_L(Y_1) \oplus \cdots \oplus E_L(Y_\ell) = Y$$

is 1 since $Y_1, \ldots, Y_\ell$ are all distinct from each other. Hence by Lemma 4, the equation $H^1_L(M) = Y$ holds with probability at most $1/(2^n - \ell + 1) \leq 2/2^n$, namely H^1 is $2/2^n$-regular. Similarly, we can prove that H^2 is $2/2^n$-regular.

Next, we will show that H^1 is ϵ_2-almost universal. Note that for any two distinct messages M_1 and M_2, the equation $H^1_L(M_1) = H^2_L(M_2)$ can be written as

$$E_L(Y^1_1) \oplus \cdots \oplus E_L(Y^1_{\ell_1}) = E_L(Y^2_1) \oplus \cdots \oplus E_L(Y^2_{\ell_2}) \ ,$$

where $Y^1_i = \langle i \rangle_m \| M_1[i]$ and $Y^2_i = \langle i \rangle_m \| M_2[i]$. Without loss of generality, we assume $\ell_1 \leq \ell_2$. If $\ell_1 = \ell_2$, then there must exist some i such that $M_1[i] \neq M_2[i]$. If $\ell_1 < \ell_2$, then $Y^2_{\ell_2}$ must be different from the values $Y^1_1, \ldots, Y^1_{\ell_1}$. So in either of these two cases, the rank of above equation is exactly 1. By Lemma 4, the equation $H^1_L(M_1) = H^1_L(M_2)$ holds with probability at most $1/(2^n - \ell_1 - \ell_2 + 1) \leq 2/2^n$. Hence H^1 is $2/2^n$-almost universal. Similarly, we can prove that H^2 is $2/2^n$-almost universal.

However, we cannot directly apply Theorem 1 at this stage since the two function keys $K_{h,1}$ and $K_{h,2}$ are identical in 2k-LightMAC_Plus while it is assumed that $K_{h,1}$ and $K_{h,2}$ are two independent keys in Theorem 1. The only problematic term in Theorem 1 is $(\epsilon_1 + \epsilon_2)^2$ since only this term relies on the independence of these two keys (i.e., condition 9 and condition 12 in the proof of Theorem 1). To handle this issue, for condition 9, we should consider for any two distinct messages M_1 and M_2, the probability of equations

$$\begin{cases} E_L(Y^1_1) \oplus \cdots \oplus E_L(Y^1_{\ell_1}) = E_L(Y^2_1) \oplus \cdots \oplus E_L(Y^2_{\ell_2}) \\ 2^{\ell_1} \cdot E_L(Y^1_1) \oplus \cdots \oplus 2 \cdot E_L(Y^1_{\ell_1}) = 2^{\ell_2} \cdot E_L(Y^2_1) \oplus \cdots \oplus 2 \cdot E_L(Y^2_{\ell_2}) \end{cases} .$$

Note that since M_1 and M_2 are two distinct messages, we can always find two random variables $E_L(Y^a_i)$ and $E_L(Y^b_j)$ where $a, b \in \{1,2\}$, $1 \leq i \leq \ell_a$, $1 \leq j \leq \ell_b$ such that the rank of above two equations is 2. For other three cases in condition

9, we can analyze them similarly. By Lemma 4, the above two equations hold with probability at most $1/(2^n - \ell_1 - \ell_2 + 2)_2 \leq 4/2^n$. Hence condition 9 holds with probability at most $16q^2/2^{2n}$. For condition 12, we should consider for three distinct messages M_1, M_2 and M_3 such that

$$\begin{cases} E_L(Y_1^1) \oplus \cdots \oplus E_L(Y_{\ell_1}^1) = E_L(Y_1^2) \oplus \cdots \oplus E_L(Y_{\ell_2}^2) \\ 2^{\ell_1} \cdot E_L(Y_1^1) \oplus \cdots \oplus 2 \cdot E_L(Y_{\ell_1}^1) = 2^{\ell_3} \cdot E_L(Y_1^3) \oplus \cdots \oplus 2 \cdot E_L(Y_{\ell_3}^3) \end{cases} .$$

Similarly, it holds with probability at most $16q^3/2^{2n}$.

Therefore, by using Theorem 1 and combined with above analysis, we can obtain the multi-user security of 2k-LightMAC_Plus.

Theorem 3. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we model as an ideal blockcipher. Then for any adversary A that makes at most p ideal-cipher queries and q evaluation queries,*

$$\begin{aligned} \mathsf{Adv}^{\mathrm{prf}}_{\text{2k-LightMAC\_Plus}}(A) \leq & \frac{2q}{2^k} + \frac{q(3q+p)(6q+p)}{2^{2k}} + \frac{2qp\ell}{2^{k+n}} + \frac{4qp}{2^{k+n}} + \frac{4qp}{2^{n+k}} \\ & + \frac{8q^2}{2^{k+n}} + \frac{4q^2\ell}{2^{k+n}} + \frac{64q^3}{2^{2n}} \\ & + \frac{2qp^2 + 4qp\ell + 2q\ell^2 + 4q^2p + 4q^2\ell + 28q^3}{2^{2n}} . \end{aligned}$$

5.3 Security of 2k-PMAC_Plus

The $2n$-bit hash function H used in 2k-PMAC_Plus is the concatenation of two functions H^1 and H^2 where H^1 and H^2 are both based a blockcipher E with the same key, namely $K_{h,1} = K_{h,2} = L$. For a message $M = M[1] \| \ldots \| M[\ell]$ where $M[i]$ is a n-bit block, $H^1_L(M)$ and $H^2_L(M)$ are defined as follows

$$\begin{aligned} H^1_L(M) &= E_L(Y_1) \oplus \cdots \oplus E_L(Y_\ell) , \\ H^2_L(M) &= 2 \cdot E_L(Y_1) \oplus \cdots \oplus 2^\ell \cdot E_L(Y_\ell) \end{aligned}$$

where $Y_i = M[i] \oplus 2^i \cdot \Delta_0 \oplus 2^{2i} \cdot \Delta_1$, and $\Delta_0 = E_L(0)$ and $\Delta_1 = E_L(1)$. The detailed code description of hash function H is illustrated in the top of Fig. 8. Then 2k-PMAC_Plus is defined as $\mathsf{DbHtS}[H, E]$ and is illustrated in the bottom of Fig. 8.

We now show that both H^1 and H^2 are ϵ_1-regular and ϵ_2-almost universal. For any message $M = M[1] \| \ldots \| M[\ell]$, we denote by E_1 the event that $Y_i = Y_j$ for $1 \leq i, j \leq \ell$ and $i \neq j$. Note that the rank of equation

$$M[i] \oplus M[j] \oplus (2^i \oplus 2^j) \cdot \Delta_0 \oplus (2^{2i} \oplus 2^{2j}) \cdot \Delta_1 = 0$$

is 1. Hence by Lemma 4,

$$\Pr[\mathsf{E}_1] \leq \frac{\binom{\ell}{2}}{2^n - 2 + 1} \leq \frac{2\ell^2}{2^n} .$$

procedure $H(L, M)$
$M[1] \| \ldots \| M[\ell] \leftarrow M$
for $i \leftarrow 1$ **to** ℓ **do**
  $Y_i \leftarrow \langle i \rangle_m \| M[i]$; $Z_i \leftarrow E_L(Y_i)$
$\Sigma = Z_1 \oplus Z_2 \oplus \ldots \oplus Z_\ell$; $\Lambda = 2^\ell \cdot Z_1 \oplus 2^{\ell-1} \cdot Z_2 \oplus \cdots \oplus 2 \cdot Z_\ell$
return (Σ, Λ)

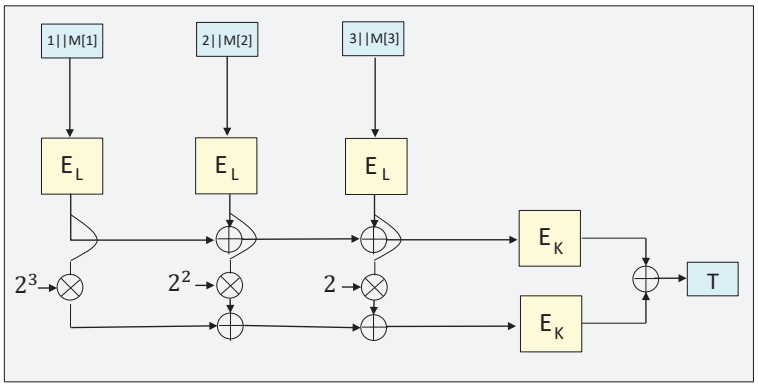

Fig. 7: **Top.** The $2n$-bit hash function used in 2k-LightMAC_Plus. Here the hash key is $K_h = (K_{h,1}, K_{h,2})$ where $K_{h,1} = K_{h,2} = L$. **Bottom.** The 2k-LightMAC_Plus construction built from a blockcipher E.

For any n-bit string $Y \in \{0,1\}^n$, the rank of equation

$$E_L(Y_1) \oplus \cdots \oplus E_L(Y_\ell) = Y$$

is 1 when event E_1 does not happen. Hence by Lemma 4, the equation $H_L^1(M) = Y$ holds with probability at most

$$\begin{aligned}\Pr\left[H_L^1(M) = Y\right] &= \Pr\left[H_L^1(M) = Y \wedge \overline{\mathsf{E}}_1\right] + \Pr\left[H_L^1(M) = Y \wedge \mathsf{E}_1\right] \\ &\le \Pr\left[H_L^1(M) = Y \mid \overline{\mathsf{E}}_1\right] + \Pr\left[\mathsf{E}_1\right] \\ &\le \frac{1}{2^n - \ell + 1} + \frac{2\ell^2}{2^n} \le \frac{4\ell^2}{2^n} \ .\end{aligned}$$

Thus H^1 is $\ell^2/2^n$-regular. Similarly, we can prove that H^2 is $\ell^2/2^n$-regular.

Next, we will show that H^1 is ϵ_2-almost universal. For any two distinct messages $M_1 = M_1[1] \| \ldots \| M_1[\ell_1]$ and $M_2 = M_2[1] \| \ldots \| M_2[\ell_2]$, we denote by E_2 the event that $Y_i^a = Y_j^b$ for $a, b \in \{1,2\}$ and $1 \le i \le \ell_a$, $1 \le j \le \ell_b$, $i \ne j$. Then similarly to the analysis of event E_1, we have $\Pr[\mathsf{E}_2] \le 8\ell^2/2^n$. Hence the rank of equation

$$E_L(Y_1^1) \oplus \cdots \oplus E_L(Y_{\ell_1}^1) = E_L(Y_1^2) \oplus \cdots \oplus E_L(Y_{\ell_2}^2)$$

is 1 when event E_2 does not happen. Hence by Lemma 4, the equation $H_L^1(M_1) = H_L^1(M_2)$ holds with probability at most $1/(2^n - 2\ell + 1) + 8\ell^2/2^n \le 10\ell^2/2^n$. This

implies that H^1 is $10\ell^2/2^n$-almost universal. By using similar argument, we can prove that H^2 is also $10\ell^2/2^n$-almost universal.

Since H^1 and H^2 use the same key, similarly to the case of 2k-LightMAC_Plus, we should handle the problematic term $(\epsilon_1+\epsilon_2)^2$ in Theorem 1 before applying it. This term arises from condition 9 and condition 12. Denote by E_3 the event that among q evaluation queries, there exits some message M such that $E_L(Y_i) = 0$ for $1 \le i \le \ell$. It is easy to see that $\Pr[\mathsf{E}_3] \le q\ell/2^n$. We proceed to analyze condition 9 and condition 12 when E_3 does not occur. For condition 9, we should consider for any two distinct messages M_1 and M_2, the probability of equations

$$\begin{cases} E_L(Y_1^1) \oplus \cdots \oplus E_L(Y_{\ell_1}^1) = E_L(Y_1^2) \oplus \cdots \oplus E_L(Y_{\ell_2}^2) \\ 2 \cdot E_L(Y_1^1) \oplus \cdots \oplus 2^{\ell_1} \cdot E_L(Y_{\ell_1}^1) = 2 \cdot E_L(Y_1^2) \oplus \cdots \oplus 2^{\ell_2} \cdot E_L(Y_{\ell_2}^2) \end{cases} .$$

Then since M_1 and M_2 are two distinct messages, by using the result in [18, Appendix C], we can always find two random variables $E_L(Y_i^a)$ and $E_L(Y_j^b)$ where $a, b \in \{1, 2\}$ and $1 \le i \le \ell_a$, $1 \le j \le \ell_b$ such that the rank of above two equations is 2 when E_2 does not happen. On the other hand, if E_2 happens, then it is easy to see that the rank of above two equations is at least 1. By Lemma 4, the above two equations hold with probability at most

$$\frac{1}{(2^n - 2\ell + 2)_2} + \frac{8\ell^2}{2^n} \cdot \frac{1}{2^n - 2\ell + 1} \le \frac{20\ell^2}{2^{2n}} .$$

For other three cases in condition 9, we can analyze them similarly. Hence condition 9 holds with probability at most $80q^2\ell^2/2^{2n} + q\ell/2^n$. For condition 12, we should consider for any there distinct messages M_1, M_2 and M_3

$$\begin{cases} E_L(Y_1^1) \oplus \cdots \oplus E_L(Y_{\ell_1}^1) = E_L(Y_1^2) \oplus \cdots \oplus E_L(Y_{\ell_2}^2) \\ 2 \cdot E_L(Y_1^1) \oplus \cdots \oplus 2^{\ell_1} \cdot E_L(Y_{\ell_1}^1) = 2 \cdot E_L(Y_1^3) \oplus \cdots \oplus 2^{\ell_3} \cdot E_L(Y_{\ell_3}^3) \end{cases} .$$

Denote by E_4 the event that $Y_i^a = Y_j^b$ for $a, b \in \{1, 2, 3\}$ and $1 \le i \le \ell_a$, $1 \le j \le \ell_b$, $i \ne j$. Then similarly to the analysis of E_2, we have $\Pr[\mathsf{E}_4] \le 18\ell^2/2^n$. Hence, by using the result in [18, Appendix C], we can always find two random variables $E_L(Y_i^a)$ and $E_L(Y_j^b)$ where $a, b \in \{1, 2, 3\}$ and $1 \le i \le \ell_a$, $1 \le j \le \ell_b$ such that the rank of above two equations is 2 when E_4 dose not occur. On the other hand, if E_4 happens, then it is easy to see that the rank of above two equations is at least 1. By Lemma 4, we can obtain that the above two equations hold with probability at most $38\ell^2/2^{2n}$. Thus, condition 12 holds with probability at most $152q^3\ell^2/2^{2n} + q\ell/2^n$.

Therefore, by using Theorem 1 and combined with above analysis, we can obtain the multi-user security of 2k-PMAC_Plus.

Theorem 4. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we model as an ideal blockcipher. Then for any adversary A that makes at most p ideal-cipher queries and q evaluation queries,*

$$\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{2k\text{-}PMAC\_Plus}}(A) \le \frac{2q}{2^k} + \frac{q(3q+p)(6q+p)}{2^{2k}} + \frac{2qp\ell}{2^{k+n}} + \frac{2qp\ell^2}{2^k} + \frac{4qp}{2^{n+k}} + \frac{24q^2\ell^3}{2^{k+n}} + \frac{320q^3\ell^2}{2^{2n}} + \frac{2qp^2 + 4qp\ell + 2q\ell^2 + 4q^2p + 4q^2\ell + 28q^3}{2^{2n}} .$$

procedure $H(L, M)$
$M[1] \| \ldots \| M[\ell] \leftarrow M$; $\Delta_0 \leftarrow E_L(0)$; $\Delta_1 \leftarrow E_L(1)$
for $i \leftarrow 1$ **to** ℓ **do**
  $Y_i \leftarrow M[i] \oplus 2^i \cdot \Delta_0 \oplus 2^{2i} \cdot \Delta_1$; $Z_i \leftarrow E_L(Y_i)$
$\Sigma = Z_1 \oplus Z_2 \oplus \ldots \oplus Z_\ell$; $\Lambda = 2 \cdot Z_1 \oplus 2^2 \cdot Z_2 \oplus \cdots \oplus 2^\ell \cdot Z_\ell$
return (Σ, Λ)

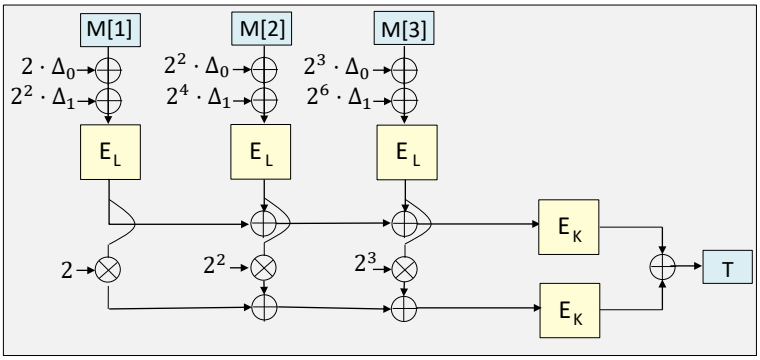

Fig. 8: **Top.** The $2n$-bit hash function used in 2k-PMAC_Plus. Here the hash key is $K_h = (K_{h,1}, K_{h,2})$ where $K_{h,1} = K_{h,2} = L$. **Bottom.** The 2k-PMAC_Plus construction built from a blockcipher E.

Acknowledgments

Yaobin Shen is more than grateful to Viet Tung Hoang for motivating this work and many helpful discussions.

References

1. J. H. An and M. Bellare. Constructing VIL-MACs from FIL-MACs: Message authentication under weakened assumptions. In M. J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 252–269. Springer, Heidelberg, Aug. 1999.
2. S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In W. Fischer and N. Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 321–345. Springer, Heidelberg, Sept. 2017.
3. M. Bellare, D. J. Bernstein, and S. Tessaro. Hash-function based PRFs: AMAC and its multi-user security. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 566–595. Springer, Heidelberg, May 2016.
4. M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

5. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 1–15. Springer, Heidelberg, Aug. 1996.
6. M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.
7. M. Bellare, K. Pietrzak, and P. Rogaway. Improved security analyses for CBC MACs. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 527–545. Springer, Heidelberg, Aug. 2005.
8. M. Bellare and B. Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 247–276. Springer, Heidelberg, Aug. 2016.
9. K. Bhargavan and G. Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and OpenVPN. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 2016*, pages 456–467. ACM Press, Oct. 2016.
10. E. Biham. How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. *Information Processing Letters*, 84(3):117–124, 2002.
11. J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 384–397. Springer, Heidelberg, Apr. / May 2002.
12. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In P. Paillier and I. Verbauwhede, editors, *CHES 2007*, volume 4727 of *LNCS*, pages 450–466. Springer, Heidelberg, Sept. 2007.
13. J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 208–225. Springer, Heidelberg, Dec. 2012.
14. P. Bose, V. T. Hoang, and S. Tessaro. Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 468–499. Springer, Heidelberg, Apr. / May 2018.
15. S. Chatterjee, A. Menezes, and P. Sarkar. Another look at tightness. In A. Miri and S. Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 293–319. Springer, Heidelberg, Aug. 2012.
16. S. Chen and J. P. Steinberger. Tight security bounds for key-alternating ciphers. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Heidelberg, May 2014.
17. N. Datta, A. Dutta, M. Nandi, and G. Paul. Double-block hash-then-sum: A paradigm for constructing BBB secure PRF. *IACR Trans. Symm. Cryptol.*, 2018(3):36–92, 2018.
18. N. Datta, A. Dutta, M. Nandi, G. Paul, and L. Zhang. Single key variant of PMAC_Plus. *IACR Trans. Symm. Cryptol.*, 2017(4):268–305, 2017.
19. V. T. Hoang and S. Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 3–32. Springer, Heidelberg, Aug. 2016.
20. V. T. Hoang and S. Tessaro. The multi-user security of double encryption. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 381–411. Springer, Heidelberg, Apr. / May 2017.

21. V. T. Hoang, S. Tessaro, and A. Thiruvengadam. The multi-user security of GCM, revisited: Tight bounds for nonce randomization. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *ACM CCS 2018*, pages 1429–1440. ACM Press, Oct. 2018.
22. T. Iwata and K. Kurosawa. OMAC: One-key CBC MAC. In T. Johansson, editor, *FSE 2003*, volume 2887 of *LNCS*, pages 129–153. Springer, Heidelberg, Feb. 2003.
23. A. Jha and M. Nandi. Revisiting structure graph and its applications to CBC-MAC and EMAC. Cryptology ePrint Archive, Report 2016/161, 2016. http://eprint.iacr.org/2016/161.
24. S. Kim, B. Lee, and J. Lee. Tight security bounds for double-block hash-then-sum macs. In *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*, pages 435–465, 2020.
25. G. Leurent, M. Nandi, and F. Sibleyras. Generic attacks against beyond-birthday-bound MACs. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 306–336. Springer, Heidelberg, Aug. 2018.
26. A. Luykx, B. Mennink, and K. G. Paterson. Analyzing multi-key security degradation. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 575–605. Springer, Heidelberg, Dec. 2017.
27. A. Luykx, B. Preneel, E. Tischhauser, and K. Yasuda. A MAC mode for lightweight block ciphers. In T. Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 43–59. Springer, Heidelberg, Mar. 2016.
28. A. Morgan, R. Pass, and E. Shi. On the adaptive security of macs and prfs. 2016. To appear.
29. N. Mouha and A. Luykx. Multi-key security: The Even-Mansour construction revisited. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 209–223. Springer, Heidelberg, Aug. 2015.
30. J. Patarin. The "coefficients H" technique (invited talk). In R. M. Avanzi, L. Keliher, and F. Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Heidelberg, Aug. 2009.
31. S. Tessaro. Optimally secure block ciphers from ideal primitives. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 437–462. Springer, Heidelberg, Nov. / Dec. 2015.
32. K. Yasuda. The sum of CBC MACs is a secure PRF. In J. Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *LNCS*, pages 366–381. Springer, Heidelberg, Mar. 2010.
33. K. Yasuda. A new variant of PMAC: Beyond the birthday bound. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 596–609. Springer, Heidelberg, Aug. 2011.
34. L. Zhang, W. Wu, H. Sui, and P. Wang. 3kf9: Enhancing 3GPP-MAC beyond the birthday bound. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 296–312. Springer, Heidelberg, Dec. 2012.