# Key-Reduced Variants of 3kf9 with Beyond-Birthday-Bound Security

Yaobin Shen[1] and Ferdinand Sibleyras[2]

[1] Shanghai Jiao Tong University and UCLouvain
[2] NTT Social Informatics Laboratories

**Abstract.** 3kf9 is a three-key CBC-type MAC that enhances the standardized integrity algorithm f9 (3GPP-MAC). It has beyond-birthday-bound security and is expected to be a possible candidate in constrained environments when instantiated with lightweight blockciphers. Two variants 2kf9 and 1kf9 were proposed to reduce key size for efficiency, but recently, Leurent et al. (CRYPTO'18) and Shen et al. (CRYPTO'21) pointed out critical flaws on these two variants and invalidated their security proofs with birthday-bound attacks.

In this work, we revisit previous constructions of key-reduced variants of 3kf9 and analyze what went wrong in security analyses. Interestingly, we find that a single doubling near the end restores the intended beyond-birthday-bound security of both 2kf9 and 1kf9. We then propose two new key-reduced variants of 3kf9, called n2kf9 and n1kf9. By leveraging previous attempts, we prove that n2kf9 is secure up to $2^{2n/3}$ queries, and prove that n1kf9 is secure up to $2^{2n/3}$ queries when the message space is prefix-free. We also provide beyond-birthday analysis of n2kf9 in the multi-user setting. Note that compared to EMAC and CBC-MAC, the additional cost to provide a higher security guarantee is expected to be minimal for n2kf9 and n1kf9. It only requires one additional blockcipher call and one doubling.

**Keywords:** Message authentication code · CBC-MAC · 3kf9 · Beyond-birthday-bound security

## 1 Introduction

A Message Authentication Code (MAC) is a fundamental symmetric-key primitive used to ensure the authenticity of messages. A MAC is typically built from a blockcipher (e.g., CBC-MAC [6,1], OMAC [25], PMAC [10]), or from a hash function (e.g., HMAC [5], NMAC [5], NI-MAC [2]). At a high level, many of these constructions iterate the underlying primitive with an $n$-bit internal state size, and thus they are subject to a generic attack using $2^{n/2}$ queries by Preneel and Oorschot [34] exploiting internal state collisions. However, the birthday-bound security $2^{n/2}$ is not always enough in practice, particularly when a MAC is implemented with a lightweight blockcipher. To reduce implementation costs, these blockciphers often offer a block length $n$ of 64 bits or even shorter [11,12,3,38,39,17,4]. In the case of $n = 64$, the birthday-bound becomes $2^{32}$ and is vulnerable in certain practical applications [9].

Double-block Hash-then-Sum constructions. To overcome the birthday-bound barrier, a series of blockcipher-based MACs has been proposed, including SUM-ECBC [40], PMAC_Plus [41], 3kf9 [42], and LightMAC_Plus [29]. The first one is a rate-2 construction, whereas the last three are rate-1 constructions and thus more efficient in that aspect.[1] [2] These constructions follow a similar paradigm called Double-block Hash-then Sum (DbHtS), where the internal state of the hash function is $2n$-bit and two encrypted values each of $n$-bit half are xored to generate the tag. Datta et al. [14] formalized this paradigm and proved these DbHtS MACs including their two-key variants are secure up to $2^{2n/3}$ queries. Leurent et al. [28] proposed a generic attack on DbHtS MACs with query complexity $2^{3n/4}$. Later, a matching proof by Kim et al. [27] confirmed that the security of DbHtS MACs stands at $2^{3n/4}$ queries. Shen et al. [35] also proved that two-key variants of DbHtS MACs are secure against $2^{2n/3}$ queries in the multi-user setting.

Key-size reduction and field multiplications. All the above DbHtS MACs require at least three or two blockcipher keys. Although in some practical protocols, the multiple keys can be generated from a master key, it has two drawbacks: (i) the construction inherently requires multiple blockcipher key schedulings, and typically need more invocation time and more energy consumption; (ii) the previous provable results cannot be applied since they are done by assuming independent keys. Hence another popular direction is to study how to reduce the key size of these MACs for better efficiency, while at the same time keeping their high security. Datta et al. [16] showed that the single-key variant of PMAC_Plus dubbed 1k-PMAC_Plus is secure up to $2^{2n/3}$ queries. Naito [30] also showed that the single-key variant of LightMAC_Plus dubbed LightMAC_Plus1k remains secure up to $2^{2n/3}$ queries. Inheriting from their original versions, besides blockcipher invocations, both 1k-PMAC_Plus and LightMAC_Plus1k require at least one additional field multiplication per message block (and totally at least $\ell$ field multiplications if the message is $\ell$-block). On the contrary, as a CBC-type mode, 3kf9 does not need field multiplications, and its key-reduced version is likely to be particularly appealing to applications in serial processing. Yet, reducing its key size appears to be a challenging problem as discussed below.

A brief history of key-reduced variants of 3kf9. 3kf9 [42] is designed by combining f9 (3GPP-MAC) [37,24] and EMAC [33]. Datta et al. [15] initialized the study of key-reduced variants of 3kf9 and proposed a single-key variant called 1kf9. Later, Leurent et al. [28] showed a birthday-bound attack on 1kf9 and thus invalidated its security proof. In an other paper, Datta et al. [14]

[1] Rate is the average number of blockcipher invocations per message block [18,19].
[2] The rate of LightMAC_Plus will increase with the counter size.

proposed a two-key variant called 2kf9. Very recently, Shen et al. [35] found a flaw in 2kf9 that it can be forged by using a single-block message. They also attempted to fix 2kf9 with several variants, yet all subject to a birthday-bound attack.

Our contributions. We revisit previous constructions of key-reduced variants of 3kf9 and analyze what went wrong in previous proofs. *Interestingly*, we find that a single doubling near the end (which can be computed efficiently by one-bit shift and one conditional XOR with a constant string) restores the intended beyond-birthday-bound security of both 2kf9 and 1kf9. We then propose two key-reduced variants of 3kf9, namely a two-key variant called n2fk9 and a single-key variant called n1kf9 (illustrated in Fig. 6 and Fig. 7, respectively). Note that to provide a higher security guarantee that is beyond the birthday-bound, the additional cost compared to EMAC and CBC-MAC is expected to be minimal for n2kf9 and n1kf9: it only requires one additional blockcipher call and one finite field doubling.

We then give security analyses for n2kf9 and n1kf9. We prove that n2kf9 is secure up to $2^{2n/3}$ queries, and prove that n1kf9 is secure up to $2^{2n/3}$ queries when the message space is prefix-free. Prefix-free means that no query is a prefix of another as in the case of CBC-MAC, and can be realized by putting the $n$-bit length encoding of each message as its first block. Note that both our proofs and previous attempts [15,14] use a similar proof strategy: first show that any pair of the final $2n$-bit state $(\Sigma_i, \Lambda_i)$ is cover-free, that is at least one of them is fresh, and then apply the lemma of sum of two identical permutations to get to a beyond-birthday-bound security result. Yet, the difficulties lie in how to show that $(\Sigma_i, \Lambda_i)$ is cover-free, which is an essential part of the proof and where previous attempts failed. Learning from previous mistakes, we provide detailed analyses to show that $(\Sigma_i, \Lambda_i)$ of constructions n2kf9 and n1kf9 is indeed cover-free with the help of doubling, and thus prove that both of them are secure beyond the birthday-bound. These analyses require surmounting some obstacles and are based on the structure graph of CBC-MAC [8,26]. Moreover, the dominant term in our bound is $q^3\ell^2/2^{2n}$ for n2kf9 and $q^3\ell^3/2^{2n}$ for n1kf9 where $q$ is the number of MAC queries and $\ell$ is the maximal block length among these MAC queries. Both are better than the previous bound $q^3\ell^4/2^{2n}$ of 2kf9 [14] and 1kf9 [15] in terms of length $\ell$. The improvement of mitigating the influence of length $\ell$ on the bound is non-trivial since it requires a fine-grained analysis of cases with multiple 'accidents' (collisions) in CBC-MAC. We also provide a beyond-birthday analysis of n2kf9 in the multi-user setting.

Discussion of our bound. Our bound is interesting for beyond-birthday-bound security with practical interest, especially when communicated messages are of limited length. We show that for any adversary making $q$ MAC queries of maximal block length $\ell$, the advantages against the PRF security of n2kf9 and n1kf9 are of the order $q^3\ell^2/2^{2n}+q^2\ell^4/2^{2n}$ and $q^3\ell^3/2^{2n}+q^2\ell^4/2^{2n}$ respectively. [3] We compare the later term with the bound $q^2\ell/2^n$ of conventional rate-1 MACs such as CBC-MAC, OMAC and PMAC. With a 64-bit block size and a guarantee that adversaries do not forge with probability more than one in a million, one gets a restriction of the form

$$\frac{q^2\ell}{2^{64}} \leq \frac{1}{2^{20}} \text{ or } \frac{q^3\ell^3}{2^{128}} + \frac{q^2\ell^4}{2^{128}} \leq \frac{1}{2^{20}} \ .$$

If the messages are $2^6$ blocks long, then $2^{19}$ messages can be tagged and total $2^{31}$ bits = 256 MB of data for the bound $q^2\ell/2^n$, while $2^{29}$ messages and total $2^{41}$ bits = 256 GB for the bound

[3] To the best of our knowledge, all security bounds of CBC-like MACs (regardless of beyond the birthday-bound or not) include a similar term $(\ell^2/2^n)^a$ for $a \geq 1$ [8,31,14,27]. This seems to be inherent that arises from the collision analysis of CBC-like structure.

$q^3\ell^3/2^{2n}+q^2\ell^4/2^{2n}$. We stress that using 128-bit blockciphers with $\mathsf{n2kf9}$ and $\mathsf{n1kf9}$ can also provide higher security guarantees.

Organization. First, we set useful notations and security notions in section 2. In section 3 we revisit different variants of $\mathsf{3kf9}$ with their associated proofs, and motivate our constructions $\mathsf{n2kf9}$ and $\mathsf{n1kf9}$. Then, in section 4 and section 5 we give the security proofs for $\mathsf{n2kf9}$. In section 6, we demonstrate the proof for $\mathsf{n1kf9}$. We also provide multi-user security analysis for $\mathsf{n2kf9}$ in Appendix A.

## 2 Preliminaries

Notation. Let $\varepsilon$ denote the empty string. Let $\{0,1\}^*$ be the set of all finite bit strings including the empty string $\varepsilon$. For a finite set $S$, we let $x \leftarrow\!\!\$\, S$ denote the uniform sampling from $S$ and assigning the value to $x$. Let $|x|$ denote the length of string $x$. Let $|x|_n$ denote the $n$-bit encoding of the length of string $x$. Concatenation of strings $x$ and $y$ is written as $x \parallel y$ or simply $xy$. $x10^*$ denotes the padding that right padded with a single 1 and as few 0 bits so that the length of string to be a multiple of $n$ bits. We let $y \leftarrow A(x_1,\ldots;r)$ denote running algorithm $A$ with randomness $r$ on inputs $x_1,\ldots$ and assigning the output to $y$. We let $y \leftarrow\!\!\$\, A(x_1,\ldots)$ be the result of picking $r$ at random and letting $y \leftarrow A(x_1,\ldots;r)$. Let $\mathrm{Perm}(n)$ denote the set of all permutations over $\{0,1\}^n$, and let $\mathrm{Func}(*,n)$ denote the set of all functions from $\{0,1\}^*$ to $\{0,1\}^n$. For integer $1 \le a \le N$, let $(N)_a$ denote $N(N-1)\ldots(N-a+1)$.

Security definitions. An adversary $\mathcal{A}$ is an algorithm that always outputs a bit. We write $\mathcal{A}^O = 1$ to denote the event that $\mathcal{A}$ outputs 1 when given access to oracle $O$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $\pi \leftarrow\!\!\$\, \mathrm{Perm}(n)$ be a random permutation. The advantage of $\mathcal{A}$ against the PRP security of $E$ is defined as

$$\mathsf{Adv}^{\mathrm{prp}}_E(\mathcal{A}) = \Pr\left[\mathcal{A}^{E_K}=1\right] - \Pr\left[\mathcal{A}^{\pi}=1\right]$$

where $K$ is chosen uniformly at random from $\{0,1\}^k$.

Let $F : \mathcal{K}\times\{0,1\}^* \to \{0,1\}^n$ be a MAC algorithm. Let $\mathcal{R} \leftarrow\!\!\$\, \mathrm{Func}(*,n)$ be a random function. The advantage of $\mathcal{A}$ against the PRF security of $F$ is defined as

$$\mathsf{Adv}^{\mathrm{prf}}_F(\mathcal{A}) = \Pr\left[\mathcal{A}^{F_K}=1\right] - \Pr\left[\mathcal{A}^{\mathcal{R}}=1\right]$$

where $K$ is chosen uniformly at random from $\mathcal{K}$. We note that the above definition captures the security of a MAC as a pseudorandom function (PRF). It is well known that any PRF is a secure MAC [7].

The H-coefficient Technique. Following from Hoang and Tessaro [23], we consider interactions between an adversary $\mathcal{A}$ and an abstract system $\mathbf{S}$ which answers $\mathcal{A}$'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 $\tau$. It is known that $\mathsf{p}_{\mathbf{S}}(\tau)$ is the description of $\mathbf{S}$ and independent of the adversary $\mathcal{A}$. 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 [32,13]. 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 $\mathcal{A}$.

```
procedure f9-hash[E](L, M)
M[1] ‖ ... ‖ M[ℓ] ← M; Y_0 ← 0^n
for i ← 1 to ℓ do
   Y_i ← E_L(Y_{i−1} ⊕ M[i])
Σ = Y_ℓ; Λ = Y_1 ⊕ Y_2 ⊕ ··· ⊕ Y_ℓ
return (Σ, Λ)
```

Fig. 1: The f9-hash algorithm producing a $2n$-bit output.

**Lemma 1.** [32,13] *Suppose that the set of attainable transcripts for the ideal system can be partitioned into good and bad ones. If there exists $\epsilon \geq 0$ such that $\frac{\mathsf{ps}_1(\tau)}{\mathsf{ps}_0(\tau)} \geq 1-\epsilon$ for any good transcript $\tau$, then*

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

Sum of two identical permutations. The following result of sum of two identical permutations under conditional distribution is helpful in our analysis.

**Lemma 2.** [16] *For 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 $\{0,1\}^n \setminus \mathcal{Z}$ that can be regarded as the outputs of a random permutation where the subset $\mathcal{Z}$ is of size $z$, and satisfy $U_i \oplus V_i = T_i$ for $1 \leq i \leq q$. Denote by $\mathcal{S}$ the set of tuples of these $2q$ variables. Then*

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

*where $\mu = \frac{4qz^2+8q^2z+6q^3}{2^{2n}}$ by assuming $z + 2q \leq 2^{n-1}$.*

## 3 The n2kf9 and n1kf9 Constructions

In this section, we first go through previous constructions based on f9-hash (see Figure 1), including $\mathsf{3kf9}$ [42], $\mathsf{2kf9}$ [14], $\mathsf{1kf9}$ [15] and a plausible construction (see Figure 5) where $\mathsf{2kf9}$ and $\mathsf{1kf9}$ are actually broken. We then propose two new constructions called $\mathsf{n2kf9}$ and $\mathsf{n1kf9}$, and show that they are both secure beyond the birthday-bound.

### 3.1 Previous Constructions

The $\mathsf{3kf9}$ construction uses 3 different keys (see Figure 2). It processes the message via f9-hash and then compute $T = E_{K_1}(\Sigma) \oplus E_{K_2}(\Lambda)$. It has a provable beyond-birthday-bound security. Intuitively, using two different keys to compute the tag makes it harder for an attacker to exploit some relations between $\Sigma$ and $\Lambda$. Events like $\Sigma_i = \Lambda_i$ for some message $M_i$ or again $\Sigma_i = \Lambda_j$, $\Sigma_j = \Lambda_i$ for some pair of messages $M_i, M_j$ are hardly detectable by looking at the output tags.

The $\mathsf{1kf9}$ construction uses a single-key for both the f9-hash and tag computation ($K = L$) (see Figure 3). It starts by processing an all-0 block before the message in f9-hash and then finishes by computing $T = E_L(\mathsf{fix0}(2\Sigma)) \oplus E_L(\mathsf{fix1}(2\Lambda))$ where the $\mathsf{fix0}$ and $\mathsf{fix1}$ functions set the least significant bit to 0 and 1 respectively, and multiplication by 2 is done in a Galois field. The $\mathsf{fix}$ function acts as a domain-separation ensuring that no $\mathsf{fix0}(2\Sigma)$ values can ever collide with a $\mathsf{fix1}(2\Lambda)$ value.

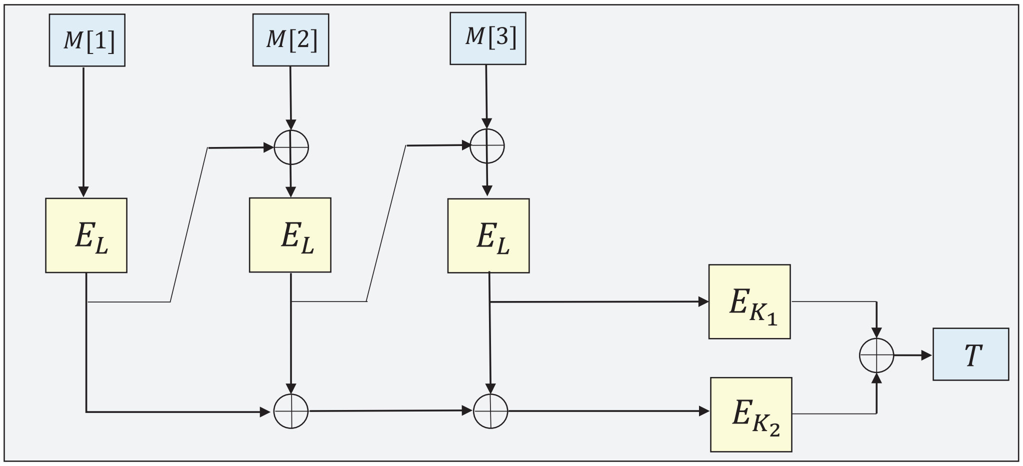


Fig. 2: The 3kf9 construction. It is built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with three keys $L$, $K_1$ and $K_2$.

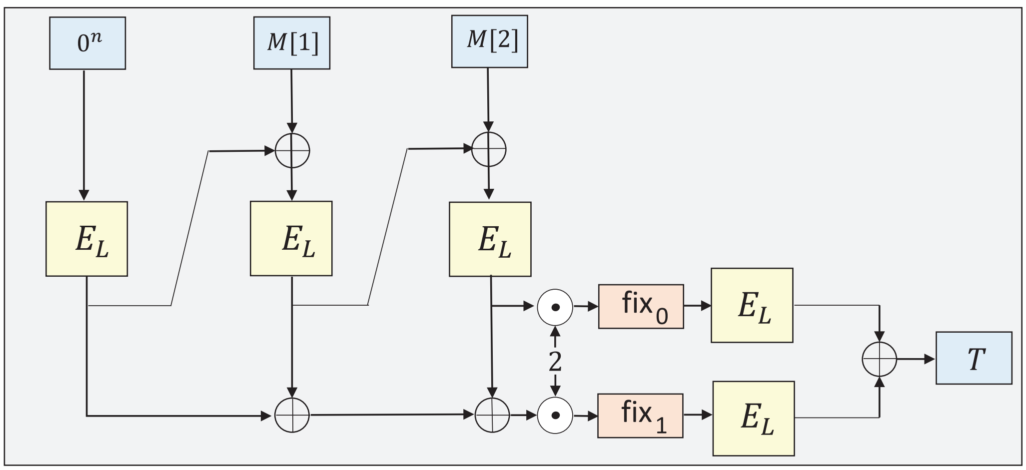


Fig. 3: The 1kf9 construction. It is built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with a single key $L$.

However, there is a birthday-bound attack by Leurent et al. [28] on 1kf9 that actually exploits the fix function. The attack looks for two values $x$ and $y$ such that $E_L(x \oplus E_L(0)) \oplus E_L(y \oplus E_L(0)) = d$, where $d$ is the inverse of 2, as it implies a collision between the tags of messages $x||0$ and $y||d$. Indeed, the $\Sigma$ parts will be equal as the injection of $d$ cancels the difference, and the $\Lambda$ parts will differ by $d$ which becomes 1 after multiplication and is absorbed by the fix function. This describes a full-state collision attack with birthday-bound complexity.

The 2kf9 construction uses two different keys (see Figure 4), one for f9-hash and the other for the tag computation as $T = E_K(\Sigma) \oplus E_K(\Lambda)$. It doesn't use any fix function or finite field multiplication. However, Shen et al. [35] realized that when f9-hash processes a single-block message then $\Sigma$ is always equal to $\Lambda$ and thus the tag is always 0. This is a single-query forgery attack which clearly demonstrates that one cannot simply use the raw f9-hash to get security beyond the birthday-bound. Shen et al. [35] further realized that adding a fix function and finite field multiplication leads to essentially the same birthday-bound attack as for 1kf9.

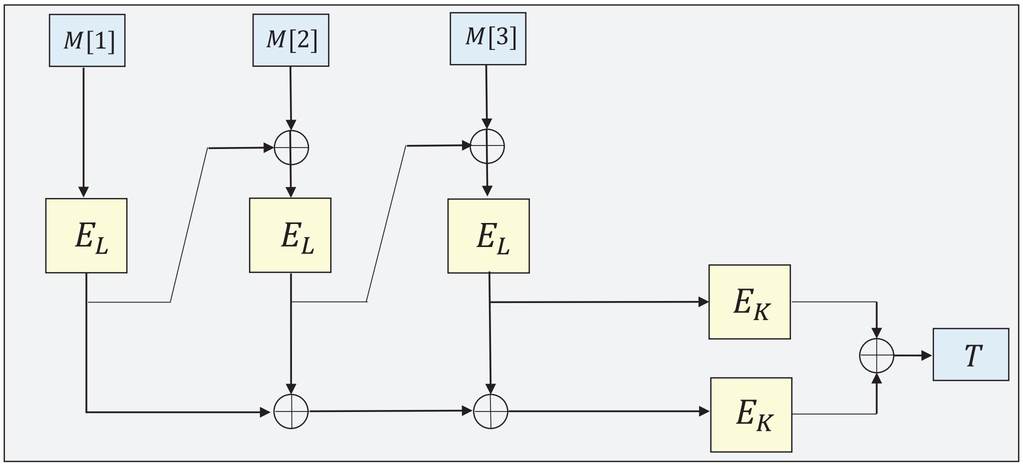


Fig. 4: The 2kf9 construction. It is built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with two keys $L$ and $K$.

A PLAUSIBLE CONSTRUCTION. The 1kf9 construction does not need the fix functions to avoid the one-query attack, thanks to prepending an all-0 block at the beginning which forbids one-block calls to f9-hash. One can wonder if doing the same for 2kf9 would suffice to fix it (see Figure 5). Unfortunately, in this case, there is still a distinguisher attack with birthday-bound complexity that exploits another undesirable property of f9-hash. For any prefix $M$ (note that $\Sigma_M$ and $\Lambda_M$ as the internal state values of f9-hash after processing $M$), if we query $M||x$ for many $x$, then the tags should collide about twice often than expected. Indeed, by varying the last block only a new $\Sigma_x$ value is added to the bottom part to compute $\Lambda_x = \Lambda_M \oplus \Sigma_x$. Therefore, for any value $x$, the probability that $\Sigma_y = \Lambda_M \oplus \Sigma_x$ is about $1/2^n$ for another value $y$, which implies $\Sigma_y = \Lambda_x$ and $\Lambda_y = \Sigma_x$ and thus results in a non-random tag collision. Both non-random and random tag collisions happen at the birthday-bound which effectively doubles the chance of observing a tag collision compared with a PRF. Even though it is not clear whether we can use this property to forge a tag, we can easily construct a distinguisher with non-negligible advantage that looks at the number of tag collisions happening around the birthday-bound. Notice that this birthday-bound distinguisher also applies to the original 2kf9 construction.

### 3.2 Looking Back at Proofs

Those attacks often indicate flaws in the proof that we can learn from. In fact, there are flaws in the original proofs of 3kf9 (see the discussion in [14, Section 6.5]), 2kf9 (attacked by [35]) and 1kf9 (withdrawn by the authors [15] and attacked by [28]). Therefore, it is important to analyze what went wrong before moving forward to fix with new constructions.

The proof of 1kf9 was already known to have flaws and was withdrawn so the attack only confirmed that the proof couldn't be fixed.

The single-query attack on 2kf9 exploits the fact that the event $\Sigma_i = \Lambda_i$ automatically occurs for any single-block message $M_i$. In the proof of [14], they study the probability of the event $\Sigma_i = \Lambda_i$ as the event that the following equation occurs (namely the intermediate values as in Figure 1):

$$Y_1^i \oplus \cdots \oplus Y_{l_i-1}^i = 0$$

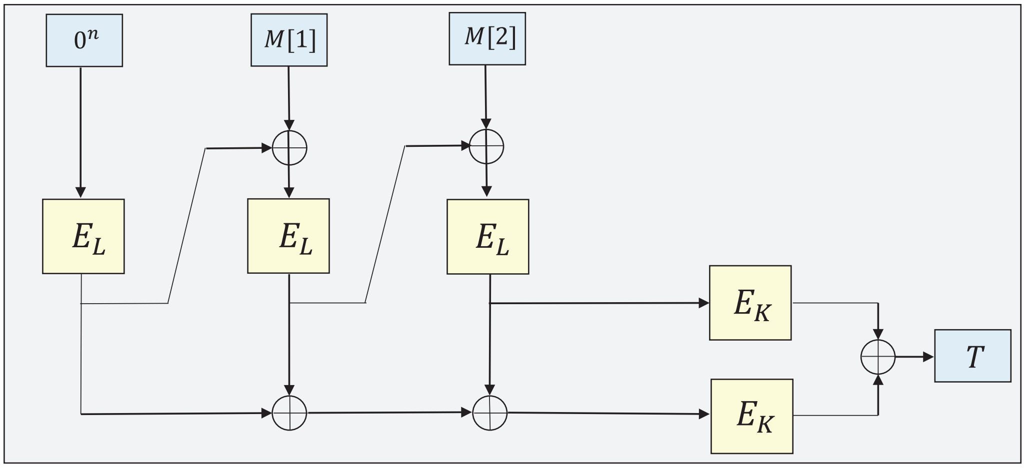


Fig. 5: A plausible construction. It is built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with two keys $L$ and $K$, and prepends an all-0 block at the beginning.

whose dotted notation may prevent to see that whenever $l_i - 1 = 0$, the case of a one-block message, the equation becomes trivial. Interestingly, even though they pointed out the attack, [35] missed this event from their multi-user setting analysis (which will be further discussed in Appendix A) . While the missing analysis is simple in most cases, it still shows that some terms are missing from the final bound.

The birthday-bound distinguisher of the plausible construction exploits the event that "$\Sigma_i = \Lambda_j$ **and** $\Sigma_j = \Lambda_i$" for two messages $M_i$ and $M_j$. The analysis of this event is simply missing from [14].

### 3.3 Our Constructions

In the rest of this paper, we will show that a simple doubling (multiply by 2) of the $\Lambda$ value can fix both 2kf9 and 1kf9 to go beyond the birthday-bound security. We now present the two new constructions n2kf9 and n1kf9.

Intuition behind the designs. Before the presentation of new constructions, we briefly discuss the intuition that the single doubling helps to avoid the problems in previous constructions. The reason is that multiplying the sum of $Y_1 \oplus Y_2 \cdots \oplus Y_\ell$ by 2 can break the relation between $\Sigma$ and $\Lambda$. More concretely, firstly, it avoids the single-query attack as finite field doubling has no fix point except for 0. Secondly, for any prefix $M$, playing with a single block suffix $x$ will introduce a unique $3 \cdot \Sigma_x$ difference between the top and bottom part and thus avoids the birthday-bound distinguishing attack. Thirdly, the removal of two fix functions fix0 and fix1 avoids the attack in 1kf9. Finally, as evidenced in the proof, for any three messages $M_i$, $M_j$ and $M_k$, the probability that $\Sigma_i = \Sigma_j$ or $\Sigma_i = \Lambda_j$, and $\Lambda_i = \Sigma_k$ or $\Lambda_i = \Lambda_k$ is small. Similar argument also holds for the case of two messages $M_i$ and $M_j$.

The n2kf9 construction. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. The n2kf9 is built from a blockcipher $E$ with two keys $L$ and $K$. Multiplication $\odot$ is done on a finite field. Note that the single doubling (multiply by 2) can be computed efficiently by one-bit shift and one conditional XOR with a constant string. The specification of n2kf9 is illustrated in Fig. 6.

**procedure** $\mathsf{n2kf9}[E](L, K, M)$
$M[1] \parallel \ldots \parallel M[\ell] \leftarrow M10^*$; $Y_0 \leftarrow 0^n$
**for** $i \leftarrow 1$ **to** $\ell$ **do**
  $Y_i \leftarrow E_L(Y_{i-1} \oplus M[i])$
$\Sigma = Y_\ell$; $\Lambda = 2 \cdot (Y_1 \oplus Y_2 \oplus \cdots \oplus Y_\ell)$
$(U, V) \leftarrow (E_K(\Sigma), E_K(\Lambda))$
$T \leftarrow U \oplus V$; **return** $T$

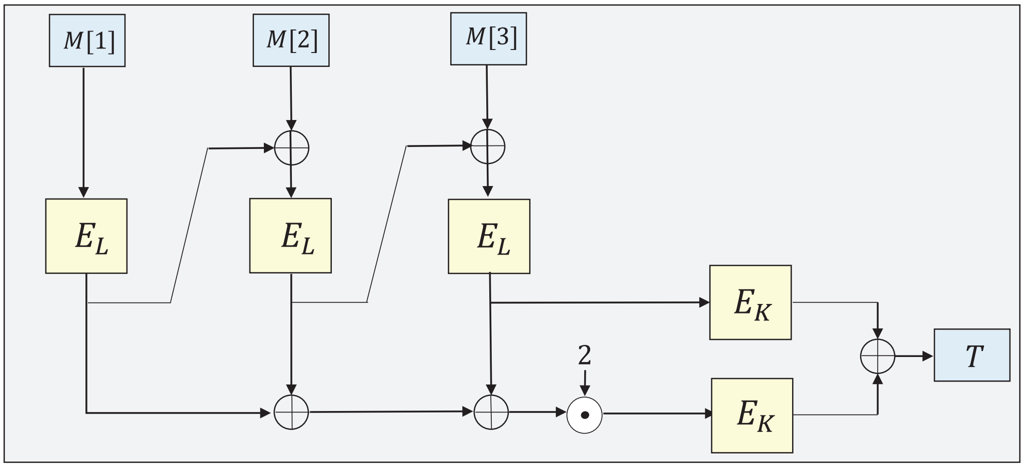


Fig. 6: **The** $\mathsf{n2kf9}[E]$ **construction.** It is built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with two keys $L$ and $K$.

SECURITY OF $\mathsf{n2kf9}$. Given that $E_L$ and $E_K$ are two good PRPs, we have the following result.

**Theorem 1.** *For any adversary $\mathcal{A}$ against the PRF security of* $\mathsf{n2kf9}$ *that runs in time at most $t$ and makes at most $q$ queries of block length at most $\ell$, we have*

$$\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n2kf9}[E]}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}_1) + \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}_2) + \frac{60q^3\ell^2}{2^{2n}} + \frac{8q^3}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} + \frac{30q^2\ell^4}{2^{2n}} + \frac{108q^3\ell^4}{2^{3n}} + \frac{2q^2}{2^{2n}} + \frac{q\ell^2}{2^n} + \frac{3q}{2^n}$$

*by assuming $\ell \le 2^{n-3}$, where $\mathcal{B}_1$ and $\mathcal{B}_2$ are two adversaries against the PRP security of the blockcipher $E_L$ and $E_K$ respectively, the former running in time at most $t_1 = t + O(q\ell)$ and making at most $q\ell$ queries while the latter running in time at most $t_2 = t + O(q)$ and making at most $q$ queries.*

The proof of Theorem 1 is in section 4 and section 5. We also provide beyond-birthday analysis of $\mathsf{n2kf9}$ in the multi-user setting in Appendix A.

THE $\mathsf{n1kf9}$ CONSTRUCTION. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. The $\mathsf{n1kf9}$ is built from a blockcipher $E$ with a single key $K$. Multiplication $\odot$ is done on a finite field. The specification of $\mathsf{n1kf9}$ is illustrated in Fig. 7. Note that the first block should always be the $n$-bit length encoding of the message to realize prefix-free as in the case for CBC-MAC.

SECURITY OF $\mathsf{n1kf9}$. Given that $E_K$ is a good PRP, the $\mathsf{n1kf9}$ is a good PRF with beyond-birthday-bound security as shown in the following theorem. The proof of this theorem is in section 6.

**procedure** $\mathsf{n1kf9}[E](K, M)$
$M[1] \parallel \ldots \parallel M[\ell] \leftarrow M10^*$; $Y_0 \leftarrow E_K(|M|_n)$
**for** $i \leftarrow 1$ **to** $\ell$ **do**
  $Y_i \leftarrow E_K(Y_{i-1} \oplus M[i])$
$\Sigma = Y_\ell$; $\Lambda = 2 \cdot (Y_0 \oplus Y_1 \oplus \cdots \oplus Y_\ell)$
$(U, V) \leftarrow (E_K(\Sigma), E_K(\Lambda))$
$T \leftarrow U \oplus V$; **return** $T$

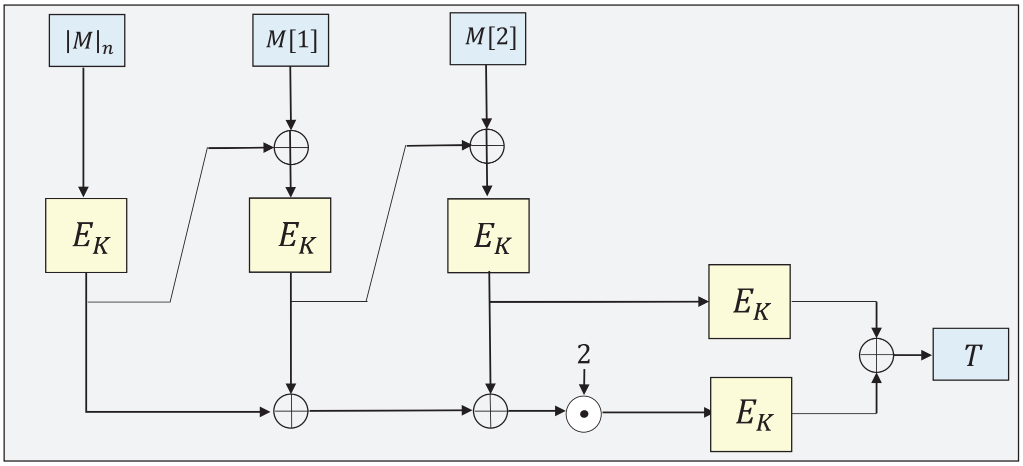


Fig. 7: **The $\mathsf{n1kf9}[E]$ construction.** It is built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with a single key $K$.

**Theorem 2.** *For any adversary $\mathcal{A}$ against the PRF security of* $\mathsf{n1kf9}$ *that runs in time at most $t$ and makes at most $q$ queries of block length at most $\ell$, we have*

$$\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n1kf9}[E]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{8q^3(\ell+3)^3}{2^{2n}} + \frac{129q^3(\ell+2)^6}{2^{3n}} + \frac{36q^2(\ell+2)^4}{2^{2n}} + \frac{6q^3}{2^{2n}} + \frac{q(\ell+2)^2}{2^n} + \frac{3q}{2^n}$$

*by assuming $\ell \leq 2^{n-3} - 2$, where $\mathcal{B}$ is an adversary against the PRP security of the blockcipher $E_K$ that runs in time at most $t = t + O(q(\ell+3))$ and makes at most $q(\ell+3)$ queries.*

Tightness of the bound. We remark that the provable $2n/3$-bit security for both $\mathsf{n2kf9}$ and $\mathsf{n1kf9}$ may not be tight. Currently we don't find a matching attack with $2^{2n/3}$ queries complexity. On the other hand, intuitively the difficulty of improving the bound lies in how to handle the case when $(\Sigma_i, \Lambda_i)$ is not cover-free instead of simply setting bad events since the final two blockciphers use the same key.

## 4 Security Analysis of n2kf9 Construction

In this section, we prove Theorem 1, which shows that $\mathsf{n2kf9}$ achieves beyond-birthday-bound security.

Overview of the proof. In the proof, we first replace blockciphers with random permutations in a standard way, and then adopt the H-coefficient technique as described in section 2 to bound the distance between real world and ideal world.

To upper bound the probability of bad transcripts in the ideal world, we define several bad conditions and grant the adversary simulated values which may be reminiscent of previous attempts [14,35]. Yet, we work on the case of the permutation instead of the key being revealed to the adversary, and some subtleties arise when calculating the ratio of good transcripts. Moreover, to analyze the bad conditions when $(\Sigma_i, \Lambda_i)$ is not cover-free and obtain a good bound (beyond birthday-bound), we need to show that the equations related to these two variables have a rank greater than or equal to 2. This analysis requires surmounting some obstacles and is based on the knowledge of structure graph of $\mathsf{CBC\text{-}MAC}$ [26,8]. In particular, to mitigate the influence of length $\ell$ on the bound, it requires to consider the event when there are two collisions among the computation of a triplet of messages, and show that these equations (including the ones related to variables $\Sigma_i$ and $\Lambda_i$ and the ones induced by these two collisions) have a rank greater than or equal to 3. Multiple subcases also occur when analyzing the event of one collision among the computation of a pair of messages. Finally, we conclude the proof by analyzing the ratio of good transcripts.

### 4.1 Game Description

*Proof.* Without loss of generality, we assume that the adversary $\mathcal{A}$ never repeats a previous query since otherwise it will receive the same answer. It is helpful to decompose the $2n$-bit hash function $H$ of $\mathsf{n2kf9}$ into two $n$-bit hash function $H^1$ and $H^2$ where $H^1_L(M) = Y_\ell$ and $H^2_L(M) = 2 \cdot (Y_1 \oplus Y_2 \oplus \cdots \oplus Y_\ell)$, and thus $\mathsf{n2kf9}[E](L, K, M) = E_K(H^1_L(M)) \oplus E_K(H^2_L(M))$. We first replace the blockciphers $E_L$ and $E_K$ of $\mathsf{n2kf9}$ with two independent random permutations $\pi_1$ and $\pi_2$, and by using the standard argument, we have

$$\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n2kf9}[E]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}_1) + \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}_2) + \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n2kf9}[\pi_1,\pi_2]}(\mathcal{A}) \ ,$$

where $\mathcal{B}_1$ is an adversary against the PRP security of $E_L$ that runs in time at most $t_1 = t + O(q\ell)$ and makes at most $q\ell$ queries, $\mathcal{B}_1$ is an adversary against the PRP security of $E_K$ that runs in time at most $t_2 = t + O(q)$ and makes at most $q$ queries. To bound the last term on the right side of the inequality (the main part of the proof), we will use the H-coefficient technique. At this stage, we can further assume that the adversary $\mathcal{A}$ is computationally unbounded and thus is deterministic. Here the real system corresponds to the world when $\mathcal{A}$ is interacting with the scheme $\mathsf{n2kf9}[\pi_1, \pi_2]$, and the ideal system corresponds to the world when $\mathcal{A}$ is interacting with a random function $\mathcal{R} \leftarrow \mathrm{Func}(*, n)$.

Setup. After the adversary $\mathcal{A}$ finishes querying, it obtains a sequence of query-answer entries $(M_1, T_1), \ldots, (M_q, T_q)$ that records the interaction between the adversary and its oracle, where $T_i = \mathsf{n2kf9}[\pi_1, \pi_2](M_i)$ in the real world and $T_i = \mathcal{R}(M_i)$ in the ideal world. In the real world, we denote by $\Sigma_i$ and $\Lambda_i$ the internal outputs of $H$ during the computation of entry $(M_i, T_i)$, namely $\Sigma_i = H^1(M_i)$ and $\Lambda_i = H^2(M_i)$. We denote by $U_i$ and $V_i$ the corresponding outputs of permutation $\pi_2$, namely $U_i = \pi_2(\Sigma_i)$ and $V_i = \pi_2(\Lambda_i)$. After the interaction, we will reveal the encoding of permutation $\pi_1$ to the adversary, and grant it all the internal values $U_i$ and $V_i$. While in the ideal world, we will instead give the adversary a permutation $\pi_1 \leftarrow\!\!\$\ \mathrm{Perm}(n)$ that is independent of its queries, and grant it $q$ pairs of dummy values $U_i$ and $V_i$ sampled as follows: the simulation oracle $\textsc{Off}(q)$ is invoked which is illustrated in Fig. 8 and returns $q$ pairs of $(U_i, V_i)$ to the adversary. Note that this additional information can only help the adversary as it can simply ignore them. In addition, the internal values $\Sigma_i$ and $\Lambda_i$ appeared during the computation of $\textsc{Off}(q)$ are uniquely determined by message $M_i$ and permutation $\pi_1$. Hence a transcript consists of the query-answer pairs $(M_i, T_i)$, the permutation $\pi_1$, and the internal values $(U_i, V_i)$.

### 4.2 Bad Transcripts

DEFINING BAD TRANSCRIPTS. We now give the definition of bad transcripts. The goal of this definition is to ensure that for each query, the corresponding pair of $(\Sigma_i, \Lambda_i)$ is always cover-free. That is, at least one of $\Sigma_i$ and $\Lambda_i$ is fresh. Formally, we say a transcript is *bad* if at least one of the following conditions is triggered:

(1) There exists an entry $(M_i, T_i)$ such that $T_i = 0^n$. This will force $U_i = V_i$ in the real world even when both $\Sigma_i$ and $\Lambda_i$ are fresh, while there is no such constraint in the ideal world.
(2) There exists an entry $(M_i, T_i)$ such that $\Sigma_i = \Lambda_i$. This will force $T_i = 0^n$, while there is no such constraint in the ideal world.
(3) There exists a pair of entries $(M_i, T_i)$ and $(M_j, T_j)$ such that $\Sigma_i = \Sigma_j$ and $\Lambda_i = \Lambda_j$, or $\Sigma_i = \Lambda_j$ and $\Lambda_i = \Sigma_j$. This will force $T_i = T_j$ in the real world, while there is no such constraint in the ideal world.
(4) There exists a pair of entries $(M_i, T_i)$ and $(M_j, T_j)$ such that $\Sigma_i \in \{\Sigma_j, \Lambda_j\}$ and $V_i \in \{V_j, U_j\}$. This guarantees that the outputs of $\Phi$ in the simulation oracle $\mathrm{OFF}(q)$ are compatible with a permutation in all good transcripts; namely, when the inputs are distinct the corresponding outputs should also be distinct.
(5) There exists a pair of entries $(M_i, T_i)$ and $(M_j, T_j)$ such that $\Lambda_i \in \{\Sigma_j, \Lambda_j\}$ and $U_i \in \{V_j, U_j\}$. Again, this guarantees that the outputs of $\Phi$ in the simulation oracle $\mathrm{OFF}(q)$ are compatible with a permutation in all good transcripts.
(6) There exists a triplet of entries $(M_i, T_i)$, $(M_j, T_j)$ and $(M_k, T_k)$ such that $\Sigma_i \in \{\Sigma_j, \Lambda_j\}$ and $\Lambda_i \in \{\Sigma_k, \Lambda_k\}$. This guarantees that for each query of good transcripts, at least one of $\Sigma_i$ and $\Lambda_i$ is fresh, and thus at least one of corresponding outputs $U_i$ and $V_i$ has fresh randomness in the real world.
(7) There exists a triplet of entries $(M_i, T_i)$, $(M_j, T_j)$ and $(M_k, T_k)$ such that $\Sigma_i \in \{\Sigma_j, \Lambda_j\}$ and $V_i \in \{U_k, V_k\}$. This guarantees that the outputs of $\Phi$ in the simulation oracle $\mathrm{OFF}(q)$ are compatible with a permutation in all good transcripts; namely, distinct inputs lead to distinct outputs.
(8) There exists a triplet of entries $(M_i, T_i)$, $(M_j, T_j)$ and $(M_k, T_k)$ such that $\Lambda_i \in \{\Sigma_j, \Lambda_j\}$, and $U_i \in \{U_k, V_k\}$. Again, this guarantees that the outputs of $\Phi$ in the simulation oracle $\mathrm{OFF}(q)$ are compatible with a permutation in all good transcripts.

If none of above conditions is met, then we say it is a *good* transcript. Denote by $X_1$ and $X_0$ the random variables for the transcript distribution in the real and ideal worlds respectively.

PROBABILITY OF BAD TRANSCRIPTS. We now proceed to bound the probability that $X_0$ is bad in the ideal world. For $1 \le i \le 8$, denote by $\mathsf{bad}_i$ the event when the $i$th condition is triggered. We analyze each event in turn. We begin with the first event. Recall that in the ideal world, each $T_i$ is a random $n$-bit string. Hence the probability that $T_i = 0^n$ is exactly $1/2^n$. Summing over at most $q$ queries,

$$\Pr[\,\mathsf{bad}_1\,] = \frac{q}{2^n} \quad . \tag{1}$$

The probability of events from 2 to 8 is bounded by the following lemma. The proof of this lemma is postponed to section 5, as its analysis is based on the structure graph of CBC-MAC [8,26] and is involved.

**Lemma 3.** *For any adversary that makes at most $q$ queries of block length at most $\ell$,*

$$\sum_{j=2}^{8} \Pr[\,\mathsf{bad}_j\,] \le \frac{60q^3\ell^2}{2^{2n}} + \frac{2q^3}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} + \frac{22q^2\ell^2}{2^{2n}} + \frac{108q^3\ell^4}{2^{3n}} + \frac{8q^2\ell^4}{2^{2n}} + \frac{2q^2}{2^{2n}} + \frac{q\ell^2}{2^n} + \frac{2q}{2^n} \ .$$

### 4.3 Good Transcripts

Transcript ratio. Let $\tau$ be a good transcript. Note that for any good transcript and for any pair of $(\Sigma_i, \Lambda_i)$, at least one of $\Sigma_i$ and $\Lambda_i$ is fresh. Hence the set $\mathcal{N}$ in $\textsc{Off}(q)$ (see Fig. 8) is empty, and the game will not abort. In the set $\mathcal{H}$, there are exactly $q + |\mathcal{F}|$ fresh values ($2|\mathcal{F}|$ fresh values for all indices in $\mathcal{F}$ and additional $(2q - 2|\mathcal{F}|)/2$ fresh values for some indices in $\mathcal{G}$), and $q - |\mathcal{F}|$ non-fresh values. For the entries that are recorded by the set $\mathcal{G}$, suppose that there are $g$ classes among the values $\Sigma_i$ and $\Lambda_i$: the elements in the same class are either connected by the equation of $\Phi(\Sigma_i) \oplus \Phi(\Lambda_i) = T_i$, or connected by the equation of $\Sigma_i = \Sigma_j$ or $\Sigma_i = \Lambda_j$, or $\Lambda_i = \Sigma_j$ or $\Lambda_i = \Lambda_j$. That is, the pair $(\Sigma_i, \Lambda_i)$ is obviously in the same class. And if $\Sigma_i = \Sigma_j$, then $(\Sigma_i, \Lambda_i)$ and $(\Sigma_j, \Lambda_j)$ are also in the same class. Note that each class contains at least three elements, and has only one corresponding sampled value since other values will be determined by the equations. On the other hand, since $\tau$ is good, the corresponding values $U_i$ and $V_i$ of these $g$ distinct classes are compatible with a permutation. That is, these $g$ sampled values are sampled such that they are distinct from each other and do not collide with other values during the computation of the set $\mathcal{F}$.

We now proceed to compute the transcript ratio. In the ideal world, since $\tau$ is good, the event $X_0 = \tau$ is the composition of the following independent events:

- We sample a random permutation $\pi_1 \leftarrow_{\$} \mathrm{Perm}(n)$ to compute the internal $Y$ state values in $\tau$. Let $\sigma$ the number of unique inputs, this happens with probability $1/(2^n)_\sigma$.
- The answers of these $q$ queries are the same as the values defined in $\tau$. This happens with probability $2^{-qn}$. On the other hand, the internal values $(U_i, V_i)_{1\le i\le q}$ from $\textsc{Off}(q)$ (Figure 8) are the same as the values defined in $\tau$. This happens with probability $1/|\mathcal{S}| \cdot 1/(2^n - 2|\mathcal{F}|)_g$: the variables $(U_i, V_i)_{i\in\mathcal{F}}$ are uniformly at random sampled from the set $\mathcal{S}$, and there are $g$ variables sampled without replacement from the remaining $2^n - 2|\mathcal{F}|$ elements for the rest $(U_i, V_i)_{i\in\mathcal{G}}$.

Therefore,

$$\Pr[\,X_0 = \tau\,] = \frac{1}{(2^n)_\sigma} \cdot \frac{1}{2^{qn}} \cdot \frac{1}{|\mathcal{S}|} \cdot \frac{1}{(2^n - 2|\mathcal{F}|)_g} \ .$$

On the other hand, in the real world, the probability of the event $X_1 = \tau$ entirely comes from the two random permutations:

- For the first permutation $\pi_1 \leftarrow_{\$} \mathrm{Perm}(n)$, the number of unique inputs appearing in $\tau$ is $\sigma$ as defined in the ideal world analysis. This happens with probability $1/(2^n)_\sigma$.

– The number of unique inputs to the second permutation is the number of unique $(U_i, V_i)_{1\le i\le q}$ as appearing in $\tau$. That is exactly $q+|\mathcal{F}|+g$, because we have a total of $q+|\mathcal{F}|$ fresh input-output tuples, and for each class in $\mathcal{G}$, we have one additional input-output tuple.

Hence,

$$\Pr[X_1 = \tau] = \frac{1}{(2^n)_\sigma} \cdot \frac{1}{(2^n)_{q+|\mathcal{F}|+g}} .$$

Therefore,

$$\begin{aligned}\frac{\Pr[X_1=\tau]}{\Pr[X_0=\tau]} &= \frac{2^{qn}\cdot|\mathcal{S}|\cdot(2^n-2|\mathcal{F}|)_g}{(2^n)_{q+|\mathcal{F}|+g}}\\ &\ge \frac{2^{(q-|\mathcal{F}|)n}\cdot(2^n)_{2|\mathcal{F}|}\cdot(2^n-2|\mathcal{F}|)_g}{(2^n)_{q+|\mathcal{F}|+g}}\cdot\left(1-\frac{6|\mathcal{F}|^3}{2^{2n}}\right)\\ &\ge \frac{2^{(q-|\mathcal{F}|)n}}{(2^n-2|\mathcal{F}|-g)_{q-|\mathcal{F}|}}\cdot\left(1-\frac{6|\mathcal{F}|^3}{2^{2n}}\right)\\ &\ge 1-\frac{6q^3}{2^{2n}} ,\end{aligned} \tag{2}$$

where the first inequality comes from Lemma 2 by fixing the conditional set to be empty.

### 4.4 Conclusion

Wrapping up. From Lemma 1, and combining Equation (1), Lemma 3 and Equation (2), we obtain

$$\begin{aligned}\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n2kf9}[\pi_1,\pi_2]}(\mathcal{A}) \le\ & \frac{60q^3\ell^2}{2^{2n}} + \frac{8q^3}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} + \frac{30q^2\ell^4}{2^{2n}} + \frac{108q^3\ell^4}{2^{3n}} + \frac{2q^2}{2^{2n}}\\ &+ \frac{q\ell^2}{2^n} + \frac{3q}{2^n}\end{aligned}$$

and conclude the proof of Theorem 1.

## 5 Proof of Lemma 3

In this section, we analyze the probability of events from 2 to 8 and prove Lemma 3. In $\mathsf{n2kf9}[\pi_1, \pi_2]$, the first $n$-bit hash function $H^1(M)$ is exactly the CBC-MAC on message $M$, while the second $n$-bit hash function $H^2(M)$ simply xor-sums all the internal outputs of CBC-MAC and then doubles it. In Appendix B, we recall the definition and properties of a combinatorial tool called the structure graph of CBC-MAC [8,26] that is useful in our analysis.

Intuitively, a structure graph $G_\pi^{\boldsymbol{M}}$ is a directed graph that is generated from the computation of CBC-MAC on various inputs $\boldsymbol{M} = \{M_1, M_2, \ldots\}$. The starting node of a structure graph is always the value $0^n$, and each output of the permutation $\pi$ is regarded as a node in the graph. In the structure graph $G_\pi^{\boldsymbol{M}}$, there may be some accidental collisions (called accidents) on the nodes that is captured by the set $\mathsf{Acc}(G_\pi^{\boldsymbol{M}})$. We will first limit the number of accidents, and then analyze the probability of bad events conditioned on it.

```
procedure OFF(q)
∀1 ≤ i ≤ q : (Σ_i, Λ_i) ← (H^1(M_i), H^2(M_i))
H = {(Σ_i, Λ_i) : 1 ≤ i ≤ q}
F = {i : both Σ_i and Λ_i are fresh in H}
G = {i : only one of Σ_i and Λ_i is fresh in H}
N = {i : neither Σ_i nor Λ_i is fresh in H}
I: set of tuples of 2|F| distinct values from {0,1}^n
S = {(W_i, X_i)_{i∈F} ∈ I : W_i ⊕ X_i = T_i}
(U_i, V_i)_{i∈F} ←$ S
∀i ∈ F : (Φ(Σ_i), Φ(Λ_i)) ← (U_i, V_i)
∀ i ∈ G :
   if Σ_i is not fresh in H then
      if Σ_i ∉ Dom(Φ)
         then U_i ←$ {0,1}^n \ Rng(Φ); Φ(Σ_i) ← U_i
      else U_i ← Φ(Σ_i)
      V_i ← T_i ⊕ U_i; Φ(Λ_i) ← V_i
   else
      if Λ_i ∉ Dom(Φ)
         then V_i ←$ {0,1}^n \ Rng(Φ); Φ(Λ_i) ← V_i
      else V_i ← Φ(Λ_i)
      U_i ← T_i ⊕ V_i; Φ(Σ_i) ← U_i
∃ i ∈ N : return ⊥
return (U_i, V_i)_{1≤i≤q}
```

Fig. 8: **Offline oracle used in the proof of n2kf9.** Here $\Phi$ is a partial function that aims to simulate a random permutation. Variables $\Sigma_i$ and $\Lambda_i$ are inputs of a random permutation, and $U_i$ and $V_i$ are corresponding outputs of this random permutation. The domain and range of $\Phi$ are both initialized to be empty.

RESTRICTING THE ACCIDENTS. We limit the number of accidents that can arise within any single, pair or triplet of messages. Consider the following event for any distinct messages $M_i, M_j, M_k$:

$$\mathsf{crash} = |\mathsf{Acc}(G_\pi^{M_i})| \geq 1 \textbf{ or } |\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| \geq 2 \textbf{ or } |\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| \geq 3 \ .$$

From Lemma 8 and the union bound, and summing over $q$ messages, $\binom{q}{2}$ pairs of messages, $\binom{q}{3}$ triplets of messages:

$$\Pr[\,\mathsf{crash}\,] \leq \frac{q\ell^2}{2^n} + \binom{q}{2} \cdot \frac{16\ell^4}{2^{2n}} + \binom{q}{3} \cdot \frac{729\ell^6}{2^{3n}} \leq \frac{q\ell^2}{2^n} + \frac{8q^2\ell^4}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} \ . \tag{3}$$

We now analyze the probability of events from 2 to 8 in conjunction with $\neg\mathsf{crash}$. That is when there is no accident within any single message, at most one accident within any pair of messages, and at most two accidents within any triplet of messages.

PROOF IDEAS OF EACH EVENT. We provide some intuition before the formal analysis of each event. For event 2, it involves only one message and is easy to show that the rank of one equation produced by this event is 1. For event 3, it consists of two sub-cases from two messages. The crucial part is to show that the rank of two equations produced by each sub-case is 2 when $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 1$. The analyses of event 4 and 5 are a bit easier than the one of event 3 since one of two equations comes from the string $T_i$ which is random and independent of queries in the ideal world. For event 6, it includes totally four sub-cases that are involved three messages. Each sub-case should be analyzed separately but the main idea is similar. The point is to show that the rank of two equations produced by each sub-case is 2 when $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 1$. Moreover, when $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 2$, it requires to show that the rank of two equations produced by each sub-case and the additional equation introduced by accidents is 3. Some details are required in this analysis. Finally, the analyses of event 7 and 8 are analogous to those of event 4 and 5, since one of two equations comes from the random string $T_i$.

<u>*Event 2.*</u> For the event 2, it is the same as the equation

$$Y_\ell^i = 2 \cdot (Y_1^i \oplus \cdots \oplus Y_\ell^i) \ ,$$

which is equivalent to

$$3 \cdot Y_\ell^i \oplus 2 \cdot (Y_1^i \oplus \cdots \oplus Y_{\ell-1}^i) = 0 \ .$$

Since the number of accidents of the structure graph $G_\pi^{M_i}$ is 0, $Y_1^i, \ldots, Y_\ell^i$ are all distinct from each other, and thus the rank of this equation is exactly 1. According to Lemma 9, the probability that this equation holds is at most $1/(2^n - \ell + 1) \le 2/2^n$ by assuming $\ell \le 2^{n-1}$. Summing over at most $q$ queries,

$$\Pr\left[\, \mathsf{bad}_2 \wedge \neg \mathsf{crash} \,\right] \le \frac{2q}{2^n} \ . \tag{4}$$

<u>*Event 3.*</u> Next, we bound the probability of event 3. This event consists of two subcases: (i) $\varSigma_i = \varSigma_j \wedge \varLambda_i = \varLambda_j$; (ii)$\varSigma_i = \varLambda_j \wedge \varLambda_i = \varSigma_j$. The first subcase is the same as

$$\begin{cases} Y_{\ell_i}^i = Y_{\ell_j}^j \\ 2 \cdot (Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i) = 2 \cdot (Y_1^j \oplus \cdots \oplus Y_{\ell_j}^j) \ . \end{cases}$$

If the number of accidents of the structure graph $G_\pi^{\{M_i,M_j\}}$ is 0, then this subcase cannot happen since the first equation requires at least one accident. If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 1$, then the rank of the above two equations is 2, which will be justified below. Hence from Lemma 9,

$$\Pr\left[\, \varSigma_i = \varSigma_j \wedge \varLambda_i = \varLambda_j \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 1 \,\right] \le \frac{1}{(2^n - 2\ell + 2)_2} \cdot \binom{2\ell}{2} \le \frac{8\ell^2}{2^{2n}}$$

where we assume $\ell \le 2^{n-2}$ and the number of structure graphs $G_\pi^{\{M_i,M_j\}}$ with one accident is at most $\binom{2\ell}{2}$ from Lemma 7. We now justify that when $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 1$, the rank of above two equations is 2. Without loss of generality, assume that $\ell_i \ge \ell_j$. Let $\alpha$ be the length of common suffix of $M_i$ and $M_j$. Then the above two equations are the same as

$$\begin{cases} Y_{\ell_i-\alpha}^i \oplus Y_{\ell_j-\alpha}^j = 0 \\ Y_1^i \oplus \cdots \oplus Y_{\ell_i-\alpha-1}^i \oplus Y_1^j \oplus \cdots \oplus Y_{\ell_j-\alpha-1}^j = 0 \ . \end{cases}$$

If $\alpha = \ell_j$, namely $M_j$ is a suffix of $M_i$, then these two equations degenerate to

$$\begin{cases} Y_{\ell_i-\ell_j}^i = 0 \\ Y_1^i \oplus \cdots \oplus Y_{\ell_i-\ell_j-1}^i = 0 \ . \end{cases}$$

In this case, the first equation cannot hold otherwise it contradicts the assumption that $|\mathsf{Acc}(G_\pi^{M_i})| = 0$. If $\alpha + 1 \le \ell_j$, then these two equations are the same as

$$\begin{cases} Y^i_{\ell_i-\alpha-1} \oplus Y^j_{\ell_j-\alpha-1} = M_i[\ell_i - \alpha] \oplus M_j[\ell_j - \alpha] \\ Y^i_1 \oplus \cdots \oplus Y^i_{\ell_i-\alpha-1} \oplus Y^j_1 \oplus \cdots \oplus Y^j_{\ell_j-\alpha-1} = 0 \ . \end{cases}$$

If $\ell_i = \alpha + 1$, then the first equation cannot hold since $M_i[1] \oplus M_j[1] \neq 0$ (note that $Y^i_0 = Y^j_0 = 0$). If $\ell_i = \alpha + 2$, then the second equation degenerates to $Y^i_1 \oplus Y^j_1 = 0$ or $Y^i_1 = 0$, neither of which can hold. Therefore $\ell_i \ge \alpha + 2$. Due to $|\mathsf{Acc}(G_\pi^{M_i})| = 0$, all the variables $Y^i_1, \ldots, Y^i_{\ell_i-\alpha-1}$ are distinct, and $Y^i_{\ell_i-\alpha-2} \notin \{Y^j_1, \ldots, Y^j_{\ell_j-\alpha-1}\}$, otherwise it will induce one additional accident on the structure graph $G_\pi^{\{M_i,M_j\}}$. Hence variable $Y_{\ell_i-\alpha-2}$ is unique in the second equation and does not appear in the first equation. Therefore, the rank of these two equations is 2. The first subcase holds with probability at most

$$\Pr\left[\, \varSigma_i = \varSigma_j \wedge \varLambda_i = \varLambda_j \wedge \neg\mathsf{crash} \,\right] \le \frac{8\ell^2}{2^{2n}} \ .$$

Next, we analyze the subcase ii. This subcase is the same as

$$\begin{cases} Y^i_{\ell_i} = 2 \cdot (Y^j_1 \oplus \cdots \oplus Y^j_{\ell_j}) \\ 2 \cdot (Y^i_1 \oplus \cdots \oplus Y^i_{\ell_i}) = Y^j_{\ell_j} \ , \end{cases}$$

which is equivalent to

$$\begin{cases} Y^i_{\ell_i} \oplus 2 \cdot (Y^j_1 \oplus \cdots \oplus Y^j_{\ell_j}) = 0 \\ 2 \cdot (Y^i_1 \oplus \cdots \oplus Y^i_{\ell_i}) \oplus Y^j_{\ell_j} = 0 \ . \end{cases}$$

If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 0$, then the rank of above two equations is 2. From Lemma 9, we have

$$\Pr\left[\, \varSigma_i = \varLambda_j \wedge \varLambda_i = \varSigma_j \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 0 \,\right] \le \frac{1}{(2^n - 2\ell + 2)_2} \le \frac{4}{2^{2n}}$$

by assuming $\ell \le 2^{n-2}$. If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 1$, then this accident appears between the path of $M_i$ and $M_j$ since $|\mathsf{Acc}(G_\pi^{M_i})| = |\mathsf{Acc}(G_\pi^{M_j})| = 0$. Without loss of generality, assume $\ell_i \ge \ell_j$. Then there exists some variable $Y^i_a$ for $1 \le a \le \ell_i$ such that $Y^i_a \notin \{Y^j_1, \ldots, Y^j_{\ell_j}\}$. It can be seen that the rank of these two equations is 2, since $Y^i_a$ is unique and has different coefficients in each equation, and at least one of two equations contains a different variable $Y^j_b$ for $1 \le b \le \ell_j$. Hence from Lemma 9,

$$\Pr\left[\, \varSigma_i = \varLambda_j \wedge \varLambda_i = \varSigma_j \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 1 \,\right] \le \frac{1}{(2^n - 2\ell + 2)_2} \cdot \binom{2\ell}{2} \le \frac{8\ell^2}{2^{2n}}$$

where we assume $\ell \le 2^{n-2}$ and the number of structure graphs $G_\pi^{\{M_i,M_j\}}$ with one accident is at most $\binom{2\ell}{2}$ from Lemma 7. Thus the probability that subcase ii occurs is at most

$$\Pr\left[\, \varSigma_i = \varLambda_j \wedge \varLambda_i = \varSigma_j \wedge \neg\mathsf{crash} \,\right] \le \frac{4}{2^{2n}} + \frac{8\ell^2}{2^{2n}} \ .$$

By the union bound, and summing over at most $\binom{q}{2}$ pairs of $M_i$ and $M_j$,

$$\Pr\left[\,\mathsf{bad}_3 \wedge \neg\mathsf{crash}\,\right] \le \frac{8q^2\ell^2}{2^{2n}} + \frac{2q^2}{2^{2n}} \ . \tag{5}$$

<u>*Events 4 and 5.*</u> We then bound the probability of event 4. We begin by analyzing the first two equations. The equations $\Sigma_i = \Sigma_j$ or $\Sigma_i = \Lambda_j$ are the same as

$$Y^i_{\ell_i} = Y^j_{\ell_j} \textbf{ or } Y^i_{\ell_i} = 2 \cdot (Y^j_1 \oplus \cdots \oplus Y^j_{\ell_j}) \ .$$

If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 0$, then the first equation cannot hold since it requires one accident. For the second equation, all these variables are distinct and thus the rank of this equation is 1. By Lemma 9, this equation holds with probability at most $1/(2^n - \ell) \le 2/2^n$ by assuming $\ell \le 2^{n-1}$. If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| \ge 1$, then by Lemma 8, this condition itself holds with probability at most $4\ell^2/2^n$. For the last two equations $V_i = V_j$ or $V_i = U_j$, they are the same as

$$U_i \oplus T_i = V_j \textbf{ or } U_i \oplus T_i = U_j$$

which holds with probability at most $2/2^n$ since $T_i$ is a random string and independent of these queries. Summing over at most $\binom{q}{2}$ pairs of queries,

$$\Pr\left[\,\mathsf{bad}_4 \wedge \neg\mathsf{crash}\,\right] \le \binom{q}{2} \cdot \left(\frac{2}{2^n} + \frac{4\ell^2}{2^n}\right) \cdot \frac{2}{2^n} \le \frac{6q^2\ell^2}{2^{2n}} \ .$$

From similar arguments,

$$\Pr\left[\,\mathsf{bad}_5 \wedge \neg\mathsf{crash}\,\right] \le \binom{q}{2} \cdot \left(\frac{4}{2^n} + \frac{4\ell^2}{2^n}\right) \cdot \frac{2}{2^n} \le \frac{8q^2\ell^2}{2^{2n}}$$

by assuming $\ell \le 2^{n-2}$.

<u>*Event 6.*</u> Next, we bound the probability of event 6. This event consists of four subcases, namely (i)$\Sigma_i = \Sigma_j \wedge \Lambda_i = \Sigma_k$; (ii) $\Sigma_i = \Sigma_j \wedge \Lambda_i = \Lambda_k$; (iii)$\Sigma_i = \Lambda_j \wedge \Lambda_i = \Sigma_k$; (iv) $\Sigma_i = \Lambda_j \wedge \Lambda_i = \Lambda_k$. The first subcase is the same as

$$\begin{cases} Y^i_{\ell_i} = Y^j_{\ell_j} \\ 2 \cdot (Y^i_1 \oplus \cdots \oplus Y^i_{\ell_i}) = Y^k_{\ell_k} \end{cases} ,$$

which is equivalent to

$$\begin{cases} Y^i_{\ell_i} \oplus Y^j_{\ell_j} = 0 \\ 2 \cdot (Y^i_1 \oplus \cdots \oplus Y^i_{\ell_i}) \oplus Y^k_{\ell_k} = 0 \end{cases} .$$

If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 0$, then the first equation cannot hold since it requires one accident. If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 1$, then the first equation counts this accident. If $\ell_i = 1$, then obviously these two equations have rank 2 since $Y^i_1$ has different coefficients in each equation. If $\ell_i > 1$, then we can always find some $Y_a^i$ for $1 \le a < \ell_i$ such that $Y_a^i \neq Y_{\ell_i}^i$ since $|\mathsf{Acc}(G_\pi^{M_i})| = 0$. Hence the rank of these two equations is 2 since $Y_a^i$ only appears in the second equation. From Lemma 9,

$$\Pr\left[\, \Sigma_i = \Sigma_j \wedge \Lambda_i = \Sigma_k \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 1 \,\right] \le \frac{1}{(2^n - 3\ell + 2)_2} \cdot \binom{3\ell}{2} \le \frac{18\ell^2}{2^{2n}}$$

where we assume $\ell \le 2^{n-3}$ and the number of structure graphs $G_\pi^{\{M_i,M_j,M_k\}}$ with one accident is at most $\binom{3\ell}{2}$ from Lemma 7. On the other hand, if $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 2$, then again, the first equation counts one accident. Then the other accident will introduce a third equation $Y_a^\alpha \oplus Y_b^\beta = M_\alpha[a+1] \oplus M_\beta[b+1]$ which is linearly independent from the first equation. The second equation is always linearly independent from the first and the third equation due to the coefficient 2. Hence the rank of these three equations is 3. From Lemma 9,

$$\begin{aligned}&\Pr\left[\, \Sigma_i = \Sigma_j \wedge \Lambda_i = \Sigma_k \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 2 \,\right] \\ &\le \frac{1}{(2^n - 3\ell + 2)_3} \cdot \binom{3\ell}{2}^2 \le \frac{162\ell^4}{2^{3n}} \ ,\end{aligned}$$

where the number of structure graphs $G_\pi^{\{M_i,M_j,M_k\}}$ with two accidents is at most $\binom{3\ell}{2}^2$ from Lemma 7. Thus subcase i holds with probability at most

$$\Pr\left[\, \Sigma_i = \Sigma_j \wedge \Lambda_i \wedge \neg\mathsf{crash} \,\right] \le \frac{18\ell^2}{2^{2n}} + \frac{162\ell^4}{2^{3n}}$$

We then bound the probability of subcase ii. This subcase is the same as

$$\begin{cases} Y_{\ell_i}^i = Y_{\ell_j}^j \\ 2 \cdot (Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i) = 2 \cdot (Y_1^k \oplus \cdots \oplus Y_{\ell_k}^k) \ , \end{cases}$$

which is equivalent to

$$\begin{cases} Y_{\ell_i-1}^i \oplus Y_{\ell_j-1}^j = M_i[\ell_i] \oplus M_j[\ell_j] \\ Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i \oplus Y_1^k \oplus \cdots \oplus Y_{\ell_k}^k = 0 \ . \end{cases}$$

If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 0$, then the first equation cannot hold since it requires one accident. If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 1$, then the first equation counts this accident. If $\ell_i = 1$, then $\ell_k \neq 1$ otherwise the second equation cannot hold since $M_i$ and $M_k$ are two distinct messages. Hence we can always find some $Y_a^k$ for $1 \le a \le \ell_k$ such that $Y_a^k \neq Y_1^i$. Then $Y_a^k$ only appears in the second equation, and thus the rank of these two equations is 2. If $\ell_i > 1$, then we can always find some $Y_a^i$ for $1 \le a \le \ell_i$ such that $Y_a^i \neq Y_{\ell_i-1}^i$ since $|\mathsf{Acc}(G_\pi^{M_i})| = 0$. Then $Y_a^i$ only appears in the second equation, and thus the rank of these two equations is 2. From Lemma 9,

$$\Pr\left[\, \Sigma_i = \Sigma_j \wedge \Lambda_i = \Lambda_k \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 1 \,\right] \le \frac{1}{(2^n - 3\ell + 2)_2} \cdot \binom{3\ell}{2} \le \frac{18\ell^2}{2^{2n}} \ .$$

On the other hand, if $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 2$, then the first equation counts one accident. The other accident will introduce a third equation $Y_a^\alpha \oplus Y_b^\beta = M_\alpha[a+1] \oplus M_\beta[b+1]$ which is linearly independent from the first equation. Obviously $(\alpha, \beta) \neq (i, j)$ otherwise $|\mathsf{Acc}(G_\pi^{\{M_i,M_j\}})| = 2$ which contradicts $\neg\mathsf{crash}$. We discuss two cases here, namely $(\alpha, \beta) = (i, k)$ or $(\alpha, \beta) = (j, k)$. For $(\alpha, \beta) = (i, k)$, the third equation is $Y_a^i \oplus Y_b^k = M_i[a+1] \oplus M_k[b+1]$. If $\ell_i = \ell_k = 1$, then the second equation cannot hold since $M_i$ and $M_k$ are two distinct messages. If $\ell_k = 1$ and $\ell_i = 2$, then if $a = 1$, $Y_2^i$ only appears in the second equation, and thus the rank of these three equations is 3; and if $a = 0$, then $Y_2^i$ also only appears in the second equation and the rank of these three equations is 3. If $\ell_k = 1$ and $\ell_i \geq 3$, then we can always find some $Y_c^i \notin \{Y_{\ell_i-1}^i, Y_a^i\}$ so that $Y_c^i$ only appears in the second equation, and thus the rank of these three equations is 3. If $\ell_k > 1$, then we can always find some $Y_c^k \neq Y_b^k$ such that $Y_c^k$ only appears in the second equation. Thus the rank of these three equations is 3. On the other hand, for the case of $(\alpha, \beta) = (j, k)$, we can analyze it similarly. Hence the rank of these three equations is 3. From Lemma 9,

$$\Pr\left[\, \Sigma_i = \Sigma_j \wedge \Lambda_i = \Lambda_k \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 2 \,\right] \leq \frac{1}{(2^n - 3\ell + 2)_3} \cdot \binom{3\ell}{2}^2 \leq \frac{162\ell^4}{2^{3n}} \ .$$

Thus,

$$\Pr\left[\, \Sigma_i = \Sigma_j \wedge \Lambda_i = \Lambda_k \wedge \neg\mathsf{crash} \,\right] \leq \frac{18\ell^2}{2^{2n}} + \frac{162\ell^4}{2^{3n}} \ .$$

Next, we bound the probability of subcase iii. This subcase is the same as

$$\begin{cases} Y_{\ell_i}^i = 2 \cdot (Y_1^j \oplus \cdots \oplus Y_{\ell_j}^j) \\ 2 \cdot (Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i) = Y_{\ell_k}^k \end{cases} ,$$

which is equivalent to

$$\begin{cases} Y_{\ell_i}^i \oplus 2 \cdot (Y_1^j \oplus \cdots \oplus Y_{\ell_j}^j) = 0 \\ 2 \cdot (Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i) \oplus Y_{\ell_k}^k = 0 \end{cases} .$$

If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 0$, then the rank of above two equations is 2 due to the coefficient 2. From Lemma 9, we have

$$\Pr\left[\, \Sigma_i = \Lambda_j \wedge \Lambda_i = \Sigma_k \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 0 \,\right] \leq \frac{1}{(2^n - 3\ell + 2)_2} \leq \frac{4}{2^{2n}}$$

by assuming $\ell \leq 2^{n-3}$. If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 1$, then this accident appears between two paths of $M_i$, $M_j$ and $M_k$. Suppose this accident introduces a third equation $Y_a^\alpha \oplus Y_b^\beta = M_\alpha[a+1] \oplus M_\beta[b+1]$ for $\alpha \neq \beta$. Then these two equations are linearly independent from this third equation due to the coefficient 2 (note that $Y \oplus 2 \cdot Y = 3 \cdot Y$). Thus the rank of these three equations is at least 2. From Lemma 9, we have

$$\Pr\left[\, \Sigma_i = \Lambda_j \wedge \Lambda_i = \Sigma_k \wedge |\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 1 \,\right] \leq \frac{1}{(2^n - 3\ell + 2)_2} \cdot \binom{3\ell}{2} \leq \frac{18\ell^2}{2^{2n}}$$

where we assume $\ell \leq 2^{n-3}$ and the number of structure graphs $G_\pi^{\{M_i,M_j,M_k\}}$ with one accident is at most $\binom{3\ell}{2}$ from Lemma 7. If $|\mathsf{Acc}(G_\pi^{\{M_i,M_j,M_k\}})| = 2$, then it introduces two linearly independent equations: $Y_a^\alpha \oplus Y_b^\beta = M_\alpha[a+1] \oplus M_\beta[b+1]$ and $Y_c^\gamma \oplus Y_d^\delta = M_\gamma[c+1] \oplus M_\delta[d+1]$ where $\alpha, \beta, \gamma, \delta \in \{i, j, k\}$ and $\alpha \neq \beta, \gamma \neq \delta, (\alpha, \beta) \neq (\gamma, \delta)$. Then these two accidental equations are linearly independent from the above two equations due to the coefficient 2. Thus the rank of these four equations is at least 3. From Lemma 9,

$$\begin{aligned}&\Pr\Big[\, \Sigma_i = \Lambda_j \wedge \Lambda_i = \Sigma_k \wedge |\mathsf{Acc}(G_\pi^{\{M_i, M_j, M_k\}})| = 2 \,\Big] \\ &\leq \frac{1}{(2^n - 3\ell + 2)_3} \cdot \binom{3\ell}{2}^2 \leq \frac{162\ell^4}{2^{3n}} \enspace .\end{aligned}$$

Thus subcase iii holds with probability at most

$$\Pr[\, \Sigma_i = \Lambda_j \wedge \Lambda_i = \Sigma_k \wedge \neg\mathsf{crash} \,] \leq \frac{18\ell^2}{2^{2n}} + \frac{4}{2^{2n}} + \frac{162\ell^4}{2^{3n}} \enspace .$$

Next, we bound the probability of subcase iv. This subcase is the same as

$$\begin{cases} Y_{\ell_i}^i = 2 \cdot (Y_1^j \oplus \cdots \oplus Y_{\ell_j}^j) \\ 2 \cdot (Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i) = 2 \cdot (Y_1^k \oplus \cdots \oplus Y_{\ell_k}^k) \enspace , \end{cases}$$

which is equivalent to

$$\begin{cases} Y_{\ell_i}^i \oplus 2 \cdot (Y_1^j \oplus \cdots \oplus Y_{\ell_j}^j) = 0 \\ Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i \oplus Y_1^k \oplus \cdots \oplus Y_{\ell_k}^k = 0 \enspace . \end{cases}$$

Then analogously to the analysis in subcase iii,

$$\Pr[\, \Sigma_i = \Lambda_j \wedge \Lambda_i = \Lambda_k \wedge \neg\mathsf{crash} \,] \leq \frac{18\ell^2}{2^{2n}} + \frac{4}{2^{2n}} + \frac{162\ell^4}{2^{3n}} \enspace .$$

By the union bound, and summing over at most $\binom{q}{3}$ triplets of $(M_i, M_j, M_k)$,

$$\Pr[\, \mathsf{bad}_6 \wedge \neg\mathsf{crash} \,] \leq \frac{12q^3\ell^2}{2^{2n}} + \frac{2q^3}{2^{2n}} + \frac{108q^3\ell^4}{2^{3n}} \enspace . \tag{6}$$

<u>*Events 7 and 8.*</u> Bounding the probability of event 7 is similar to handling event 4, except that now there are at most $q^3$ triplets of queries and the probability of $|\mathsf{Acc}(G_\pi^{\{M_i, M_j, M_k\}})| \geq 1$ is bounded by $9\ell^2/2^n$. Hence,

$$\Pr[\, \mathsf{bad}_7 \wedge \neg\mathsf{crash} \,] \leq q^3 \cdot (\frac{2}{2^n} + \frac{9\ell^2}{2^n}) \cdot \frac{2}{2^n} \leq \frac{22q^3\ell^2}{2^{2n}} \enspace .$$

Similarly,

$$\Pr[\, \mathsf{bad}_8 \wedge \neg\mathsf{crash} \,] \leq q^3 \cdot (\frac{4}{2^n} + \frac{9\ell^2}{2^n}) \cdot \frac{2}{2^n} \leq \frac{26q^3\ell^2}{2^{2n}} \enspace .$$

Summing up,

$$\begin{aligned}&\sum_{j=2}^{8} \Pr[\, \mathsf{bad}_j \,] \leq \Pr[\, \mathsf{crash} \,] + \sum_{j=2}^{8} \Pr[\, \mathsf{bad}_j \wedge \neg\mathsf{crash} \,] \\ &\leq \frac{60q^3\ell^2}{2^{2n}} + \frac{2q^3}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} + \frac{22q^2\ell^2}{2^{2n}} + \frac{108q^3\ell^4}{2^{3n}} + \frac{8q^2\ell^4}{2^{2n}} + \frac{2q^2}{2^{2n}} + \frac{q\ell^2}{2^n} + \frac{2q}{2^n}\end{aligned}$$

and conclude the proof of Lemma 3.

## 6 Security Analysis of n1kf9 Construction

In this section, we prove Theorem 2 that states the beyond-birthday-bound security of $\mathsf{n1kf9}$ (illustrated in Fig. 7).

Overview of the proof. The proof idea of $\mathsf{n1kf9}$ mainly follows from the one of $\mathsf{n2kf9}$. Yet, since $\mathsf{n1kf9}$ only requires one key that is both used in the hash part and final encryption, there are some points that are different and non-trivial. This is also the reason that the bound of $\mathsf{n1kf9}$ is slightly worse than the bound of $\mathsf{n2kf9}$. First, the simulation oracle used in the ideal world is adjusted to take into account the relation between the hash part and final encryption. The calculation of good transcripts is changed accordingly. In addition, more bad events emerge since $\Sigma_i$ and $\Lambda_i$ may collide with previous inputs of hash part. Moreover, to mitigate the influence of length $\ell$ on the bound, a fine-grained analysis is again required.

Remark. It may be interesting to summarize some property of enhanced f9 hash for generalized proof. However, as far as we can see, the analysis of single-key $2n$-bit hash function is case dedicated and requires many insights on the concrete construction.

### 6.1 Game Description

*Proof.* Without loss of generality, we assume that the adversary never repeats a prior query since otherwise it will receive the same answer. The $2n$-bit hash function $H$ of $\mathsf{n1kf9}$ consists of two $n$-bit hash functions $H^1$ and $H^2$ where $H^1_K(M) = Y_\ell$ and $H^2_K(M) = 2 \cdot (Y_0 \oplus Y_1 \oplus \cdots \oplus Y_\ell)$, and thus $\mathsf{n1kf9}[E](K, M) = E_K(H^1_K(M)) \oplus E_K(H^2_K(M))$. As usual, we first replace the blockcipher $E_K$ with a random permutation $\pi \leftarrow\!\$\, \mathrm{Perm}(n)$, and from the standard argument,

$$\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n1kf9}[E]}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n1kf9}[\pi]}(\mathcal{A}) \ ,$$

where $\mathcal{B}$ is an adversary against the PRP security of the blockcipher $E_K$ that runs in time at most $t = t + O(q(\ell+3))$ and makes at most $q(\ell+3)$ queries. We will use the H-coefficient technique to bound $\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n1kf9}[\pi]}(\mathcal{A})$, even when $\mathcal{A}$ is computationally unbounded. The real system and ideal system correspond to the game when $\mathcal{A}$ is interacting with the scheme $\mathsf{n1kf9}[\pi]$ and a random function $\mathcal{R} \leftarrow\!\$\, \mathrm{Func}(*, n)$, respectively.

Setup. After the adversary $\mathcal{A}$ finishes querying, it obtains a sequence of query-answer entries $(M_1, T_1), \ldots, (M_q, T_q)$ that records the interaction with its oracle, where $T_i = \mathsf{n1kf9}[\pi](M_i)$ in the real world and $T_i = \mathcal{R}(M_i)$ in the ideal world. In the real world, let $\Sigma_i = H^1(M_i)$ and $\Lambda_i = H^2(M_i)$ be the internal outputs of $H$ for entry $(M_i, T_i)$. Let $U_i = \pi(\Sigma_i)$ and $V_i = \pi(\Lambda_i)$ be the outputs of permutation $\pi$ after the hash part. After the interaction, we reveal the random permutation $\pi$ to the adversary, and grant it all the internal values $U_i$ and $V_i$. In the ideal world, we instead give the adversary a fresh random permutation $\pi$ that is independent of its queries, and grant it $q$ pairs of dummy values $U_i$ and $V_i$ sampled as follows: the simulation oracle $\textsc{Off}(q)$ is invoked which is illustrated in Fig. 12 of Appendix D and returns $(U_i, V_i)$ to the adversary. These additional information can only help the adversary. In addition, the internal values $\Sigma_i$ and $\Lambda_i$ (and also $Y^i_0, \ldots, Y^i_{\ell_i}$) appearing during the computation of $\textsc{Off}(q)$ are uniquely determined by message $M_i$ and permutation $\pi$. Hence a transcript consists of the query-answer pairs $(M_i, T_i)$, the permutation $\pi$, and the internal values $(U_i, V_i)$.

### 6.2 Bad Transcripts

Defining bad transcripts. We now give the definition of bad transcripts. The goal is to ensure that for each query, the corresponding pair of $(\Sigma_i, \Lambda_i)$ is always cover-free. Formally, we say a transcript is *bad* if at least one of the following conditions is triggered:

(1) There exists an entry $(M_i, T_i)$ such that $T_i = 0^n$. This will force $U_i = V_i$ in the real world, while there is no such constraint in the ideal world.
(2) There exists an entry $(M_i, T_i)$ such that $\Sigma_i = \Lambda_i$. This will force $T_i = 0^n$, while there is no such constraint in the ideal world.
(3) There exists an entry $(M_i, T_i)$ such that $\Sigma_i \in \{|M_i|_n, Y_0^i \oplus M_i[1], \ldots, Y_{\ell_i-1}^i \oplus M_i[\ell_i]\}$ and $\Lambda_i \in \{|M_i|_n, Y_0^i \oplus M_i[1], \ldots, Y_{\ell_i-1}^i \oplus M_i[\ell_i]\}$. That is, both $\Sigma_i$ and $\Lambda_i$ collide with previous inputs of permutation $\pi$ for the same query. This guarantees that for each query of all good transcripts, at least one of $\Sigma_i$ and $\Lambda_i$ is fresh, and thus at least one of corresponding outputs $U_i$ and $V_i$ has fresh randomness in the real world.
(4) There exists a pair of entries $(M_i, T_i)$ and $(M_j, T_j)$ such that $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$ and $\Lambda_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$. That is, both $\Sigma_i$ and $\Lambda_i$ collide with previous inputs of permutation $\pi$ for another entry $(M_j, T_j)$. Again, this guarantees that for each query of good transcripts, at least one of $\Sigma_i$ and $\Lambda_i$ is fresh.
(5) There exists a pair of entries $(M_i, T_i)$ and $(M_j, T_j)$ such that $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$ and $V_i \in \{Y_0^j, \ldots, Y_{\ell_j}^j, U_j, V_j\}$. This guarantees that the outputs of permutation $\pi$ in the simulation oracle $\textsc{Off}(q)$ are compatible with a permutation for all good transcripts, namely when the inputs are distinct, then the corresponding outputs should also be distinct.
(6) There exists a pair of entries $(M_i, T_i)$ and $(M_j, T_j)$ such that $\Lambda_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$ and $U_i \in \{Y_0^j, \ldots, Y_{\ell_j}^j, U_j, V_j\}$. Again, this guarantees that the outputs of permutation $\pi$ in the simulation oracle $\textsc{Off}(q)$ are compatible with a permutation for all good transcripts.
(7) There exists a triplet of entries $(M_i, T_i)$, $(M_j, T_j)$ and $(M_k, T_k)$ such that $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$ and $\Lambda_i \in \{|M_k|_n, Y_0^k \oplus M_k[1], \ldots, Y_{\ell_k-1}^j \oplus M_k[\ell_k], \Sigma_k, \Lambda_k\}$. That is, $\Sigma_i$ and $\Lambda_i$ collide with previous inputs of permutation $\pi$ for two different entries $(M_j, T_j)$ and $(M_k, T_k)$.
(8) There exists a triplet of entries $(M_i, T_i)$, $(M_j, T_j)$ and $(M_k, T_k)$ such that $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$ and $V_i \in \{Y_0^k, \ldots, Y_{\ell_k}^k, U_k, V_k\}$. This guarantees that the outputs of permutation $\pi$ in the simulation oracle $\textsc{Off}(q)$ are compatible with a permutation for all good transcripts, namely distinct inputs produce distinct outputs (and conversely).
(9) There exists a triplet of entries $(M_i, T_i)$, $(M_j, T_j)$ and $(M_k, T_k)$ such that $\Lambda_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$ and $U_i \in \{Y_0^k, \ldots, Y_{\ell_k}^k, U_k, V_k\}$. Again, this guarantees that the outputs of permutation $\pi$ in the simulation oracle $\textsc{Off}(q)$ are compatible with a permutation for all good transcripts.

If none of above conditions is met, then we say it is a *good* transcript. Denote by $X_1$ and $X_0$ the random variables for the transcript distribution in the real and ideal worlds respectively. The probability of bad transcripts in the ideal world is bounded by the following lemma; the proof is in Appendix C.

**Lemma 4.** *For any adversary that makes at most $q$ queries of block length at most $\ell \leq 2^{n-3} - 2$,*

$$\Pr[\, X_0 \text{ is bad} \,] \leq \frac{5q^3(\ell+3)^3}{2^{2n}} + \frac{3q^3(\ell+3)^2}{2^{2n}} + \frac{24q^2(\ell+2)^4}{2^{2n}} + \frac{122q^3(\ell+2)^6}{2^{3n}} + \frac{7q^3(\ell+3)^5}{2^{3n}} + \frac{q(\ell+2)^2}{2^n} + \frac{3q}{2^n} \ .$$

### 6.3 Good Transcripts

Transcript ratio. Let $\tau$ be a good transcript. Similarly to the arguments in Section 4.3, the set $\mathcal{N}$ in $\textsc{Off}(q)$ (illustrated in Fig. 12) is empty. In the set of $\Sigma_i$ and $\Lambda_i$, there are $q+|\mathcal{F}|$ fresh values and $q-|\mathcal{F}|$ non-fresh values. For the entries that are recorded by the set $\mathcal{G}$, suppose there are $g$ sampled values.

We now proceed to compute the transcript ratio. In the ideal world, since $\tau$ is good, the event $X_0 = \tau$ is the composition of the following independent events:

- When we sample a random permutation $\pi \leftarrow_{\$} \mathrm{Perm}(n)$, we use exactly $|\mathcal{H}|$ values which appear in $\tau$. This happens with probability $1/(2^n)_{|\mathcal{H}|}$.
- The answers of these $q$ queries are the same as the values defined in $\tau$. This happens with probability $2^{-qn}$. On the other hand, the internal values $(U_i, V_i)_{1\leq i\leq q}$ from $\textsc{Off}(q)$ are the same as the values defined in $\tau$. This happens with probability $1/|\mathcal{S}| \cdot 1/(2^n - |\mathcal{H}| - 2|\mathcal{F}|)_g$: the variables $(U_i, V_i)_{i\in\mathcal{F}}$ are uniformly at random sampled from the set $\mathcal{S}$, and there are $g$ variables sampled without replacement from the remaining $2^n - |\mathcal{H}| - 2|\mathcal{F}|$ elements for the rest $(U_i, V_i)_{i\in\mathcal{G}}$.

Therefore,

$$\Pr[\, X_0 = \tau \,] = \frac{1}{(2^n)_{|\mathcal{H}|}} \cdot \frac{1}{2^{qn}} \cdot \frac{1}{|\mathcal{S}|} \cdot \frac{1}{(2^n - |\mathcal{H}| - 2|\mathcal{F}|)_g} \ .$$

On the other hand, in the real world, the probability of the event $X_1 = \tau$ only comes from the computation of the random permutation $\pi$:

- First we draw $|\mathcal{H}|$ values from $\pi$ to compute the internal $Y$ states values.
- To compute the $(U_i, V_i)_{1\leq i\leq q}$, the number of permutation outputs required is exactly $q+|\mathcal{F}|+g$, because we totally have $q+|\mathcal{F}|$ fresh input-output tuples, and for each class in $\mathcal{G}$, we have one additional input-output tuple.

Hence,

$$\Pr[\, X_1 = \tau \,] = \frac{1}{(2^n)_{|\mathcal{H}|+q+|\mathcal{F}|+g}} \ .$$

Therefore,

$$\begin{aligned}
&\frac{\Pr[\, X_1 = \tau \,]}{\Pr[\, X_0 = \tau \,]} = \frac{2^{qn} \cdot |\mathcal{S}| \cdot (2^n - |\mathcal{H}| - 2|\mathcal{F}|)_g}{(2^n - |\mathcal{H}|)_{q+|\mathcal{F}|+g}} \\
&\geq \frac{2^{(q-|\mathcal{F}|)n} \cdot (2^n - |\mathcal{H}|)_{2|\mathcal{F}|} \cdot (2^n - |\mathcal{H}| - 2|\mathcal{F}|)_g}{(2^n - |\mathcal{H}|)_{q+|\mathcal{F}|+g}} \cdot \Big(1 - \frac{4|\mathcal{F}||\mathcal{H}|^2 + 8|\mathcal{F}|^2|\mathcal{H}| + 6|\mathcal{F}|^3}{2^{2n}}\Big) \\
&\geq \frac{2^{(q-|\mathcal{F}|)n}}{(2^n - |\mathcal{H}| - 2|\mathcal{F}| - g)_{q-|\mathcal{F}|}} \cdot \Big(1 - \frac{4|\mathcal{F}||\mathcal{H}|^2 + 8|\mathcal{F}|^2|\mathcal{H}| + 6|\mathcal{F}|^3}{2^{2n}}\Big) \\
&\geq 1 - \frac{4q(\ell+2)^2 + 8q^2(\ell+2) + 6q^3}{2^{2n}} \ , \qquad (7)
\end{aligned}$$

where the first inequality comes from Lemma 2.

### 6.4 Conclusion

Wrapping up. From Lemma 1 and combining Lemma 4 and Equation (7), we obtain

$$\begin{aligned}\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{n1kf9}[\pi]}(\mathcal{A}) \leq \frac{8q^3(\ell+3)^3}{2^{2n}} + \frac{129q^3(\ell+2)^6}{2^{3n}} + \frac{36q^2(\ell+2)^4}{2^{2n}} \\ + \frac{6q^3}{2^{2n}} + \frac{q(\ell+2)^2}{2^n} + \frac{3q}{2^n}\end{aligned}$$

and conclude the proof of Theorem 2.

## Acknowledgments

We thank the reviewers for their useful comments. This work has been funded in parts by the European Union through the ERC consolidator grant SWORD (num. 724725), and by National Key Research and Development Program of China (No. 2019YFB2101601).

## References

1. Computer data authentication. National Bureau of Standards, NIST FIPS PUB 113, U.S. Department of Commerce, 1985.
2. 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.
3. 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.
4. R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. `https://eprint.iacr.org/2013/404`.
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. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.
7. 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.
8. 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.
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. 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.
11. 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.
12. 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.

13. 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.
14. 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.
15. N. Datta, A. Dutta, M. Nandi, G. Paul, and L. Zhang. Building single-key beyond birthday bound message authentication code. Cryptology ePrint Archive, Report 2015/958, 2015. `https://eprint.iacr.org/2015/958`.
16. 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.
17. C. De Cannière, O. Dunkelman, and M. Knežević. KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers. In C. Clavier and K. Gaj, editors, *CHES 2009*, volume 5747 of *LNCS*, pages 272–288. Springer, Heidelberg, Sept. 2009.
18. Y. Dodis, K. Pietrzak, and P. Puniya. A new mode of operation for block ciphers and length-preserving MACs. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 198–219. Springer, Heidelberg, Apr. 2008.
19. Y. Dodis and J. P. Steinberger. Message authentication codes from unpredictable block ciphers. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 267–285. Springer, Heidelberg, Aug. 2009.
20. P. Gaži, K. Pietrzak, and M. Rybár. The exact PRF-security of NMAC and HMAC. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 113–130. Springer, Heidelberg, Aug. 2014.
21. T. Guo and P. Wang. A note on the security framework of two-key dbhts macs. *IACR Cryptol. ePrint Arch.*, page 375, 2022.
22. V. T. Hoang and Y. Shen. Security analysis of NIST CTR-DRBG. In D. Micciancio and T. Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 218–247. Springer, Heidelberg, Aug. 2020.
23. 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.
24. T. Iwata and T. Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 427–445. Springer, Heidelberg, Feb. 2004.
25. 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.
26. A. Jha and M. Nandi. Revisiting structure graph and its applications to CBC-MAC and EMAC. Cryptology ePrint Archive, Report 2016/161, 2016. `https://eprint.iacr.org/2016/161`.
27. S. Kim, B. Lee, and J. Lee. Tight security bounds for double-block hash-then-sum MACs. In A. Canteaut and Y. Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 435–465. Springer, Heidelberg, May 2020.
28. 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.
29. Y. Naito. Blockcipher-based MACs: Beyond the birthday bound without message length. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 446–470. Springer, Heidelberg, Dec. 2017.
30. Y. Naito. Improved security bound of LightMAC_Plus and its single-key variant. In N. P. Smart, editor, *CT-RSA 2018*, volume 10808 of *LNCS*, pages 300–318. Springer, Heidelberg, Apr. 2018.
31. M. Nandi. A unified method for improving PRF bounds for a class of blockcipher based MACs. In S. Hong and T. Iwata, editors, *FSE 2010*, volume 6147 of *LNCS*, pages 212–229. Springer, Heidelberg, Feb. 2010.
32. 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.
33. E. Petrank and C. Rackoff. CBC MAC for real-time data sources. *Journal of Cryptology*, 13(3):315–338, June 2000.
34. B. Preneel and P. C. van Oorschot. MDx-MAC and building fast MACs from hash functions. In D. Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 1–14. Springer, Heidelberg, Aug. 1995.
35. Y. Shen, L. Wang, D. Gu, and J. Weng. Revisiting the security of DbHtS MACs: Beyond-birthday-bound in the multi-user setting. In T. Malkin and C. Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 309–336, Virtual Event, Aug. 2021. Springer, Heidelberg.
36. Y. Shen, L. Wang, and J. Weng. Revisiting the security of dbhts macs: Beyond-birthday-bound in the multi-user setting. *IACR Cryptol. ePrint Arch.*, page 1523, 2020.

37. G. T. . v 3.1.1. Specification of the 3gpp confidentiality and integrity algorithms, document 1: f8 and f9 specification. https://www.3gpp.org/DynaReport/35-series.htm.
38. G. T. . v 3.1.1. Specification of the 3gpp confidentiality and integrity algorithms, document 2: Kasumi specification. http://www.3gpp.org/ftp/Specs/html-info/35-series.htm.
39. G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong. The simeck family of lightweight block ciphers. In T. Güneysu and H. Handschuh, editors, *CHES 2015*, volume 9293 of *LNCS*, pages 307–329. Springer, Heidelberg, Sept. 2015.
40. 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.
41. 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.
42. 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.

## A Multi-User Security of n2kf9

In this section, we show that $\mathsf{n2kf9}$ is secure beyond the birthday-bound in the multi-user (mu) setting. The proof is done in the ideal-cipher model which is common in most analyses for the mu security. The mu security of $\mathsf{n2kf9}$ is formalized by the following theorem.

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

$$\mathsf{Adv}^{\mathrm{mprf}}_{\mathsf{n2kf9}}(\mathcal{A}) \le \frac{52q^3\ell^2}{2^{2n}} + \frac{40q^3}{2^{2n}} + \frac{q(3q+p)(6q+2p)}{2^{2k}} + \frac{20q^2\ell^4}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}} + \frac{18q^2\ell^3}{2^{n+k}} + \frac{2qp\ell^2}{2^{n+k}} + \frac{10qp\ell}{2^{n+k}} + \frac{q\ell^2}{2^n} + \frac{2q}{2^k} + \frac{3q}{2^n}$$

*by assuming that $\ell \le 2^{n-3}$ and $p + q\ell \le 2^{n-1}$.*

To prove this theorem, we first recall the security framework by Shen et al. [35,36] for two-key $\mathsf{DbHtS}$ construction that is useful in our analysis.

REMARK. Recently Guo and Wang [21] pointed out a miscalculation in the proof by Shen et al. [35]. The authors consequently updated the security framework in ePrint [36] to fix this issue. Here we will adopt the updated framework to analyze the multi-user security of $\mathsf{n2kf9}$.

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 where $\mathcal{K}_h$ is the key space and $\mathcal{M}$ is the message space. The $2n$-bit hash function $H$ consists of two $n$-bit hash functions $H^1$ and $H^2$, and $H_{K_h}(M) = (H^1_{K_{h,1}}(M), H^2_{K_{h,2}}(M))$ where $K_h = (K_{h,1}, K_{h,2})$. Given a $2n$-bit hash function $H$ and a blockcipher $E$, the $\mathsf{DbHtS}$ construction is defined as follows

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

Algorithm $\mathsf{n2kf9}$ can be regarded as one of $\mathsf{DbHtS}$ constructions, where $H_{K_h}(M) = (H^1_L(M), H^2_L(M))$, $H^1_L(M) = Y_\ell$ and $H^2_L(M) = 2 \cdot (Y_1 \oplus \cdots \oplus Y_\ell)$, and thus $\mathsf{n2kf9}[E](L,K,M) = E_K(H^1_L(M)) \oplus E_K(H^2_L(M))$.

We recall two useful properties with respect to a hash function. For a hash function $H : \mathcal{K}_h \times \mathcal{M} \to \{0,1\}^n$, we say it is $\epsilon_1$-*regular* if for any $M \in \mathcal{M}$ and $Z \in \{0,1\}^n$,

$$\Pr[\, K_h \leftarrow_{\$} \mathcal{K}_h : H_{K_h}(M) = Z \,] \le \epsilon_1 \ ,$$

and say it is $\epsilon_2$-*almost universal* if for any two distinct strings $M_1, M_2 \in \{0,1\}^n$,

$$\Pr[\, K_h \leftarrow_{\$} \mathcal{K}_h : H_{K_h}(M_1) = H_{K_h}(M_2) \,] \le \epsilon_2 \ .$$

MULTI-USER SECURITY. For an adversary $\mathcal{A}$, let

$$\mathsf{Adv}^{\mathrm{mprf}}_{\mathsf{DbHtS}}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\mathrm{mprf}}_{\mathsf{DbHtS}}(\mathcal{A})] - 1$$

be the advantage of the adversary against the multi-user PRF security of $\mathsf{DbHtS}$ construction, where game $\mathbf{G}^{\mathrm{mprf}}_{\mathsf{DbHtS}}$ is defined in Fig. 9. We now recall the multi-user security framework of $\mathsf{DbHtS}$ construction, which is characterized by the following lemma.

| | |
|---|---|
| **procedure Initialize** | **procedure** $\textsc{Eval}(i, M)$ |
| $(K_h^1, K_1), (K_h^2, K_2), \cdots, \leftarrow\$\ \mathcal{K}_h \times \mathcal{K}$ | $T_1 \leftarrow \mathsf{DbHtS}[H, E](K_h^i, K_i, M)$ |
| $f_1, f_2, \cdots, \leftarrow\$\ \mathrm{Func}(\mathcal{M}, \{0,1\}^n)$ | $T_0 \leftarrow f_i(M)$ |
| $b \leftarrow\$\ \{0,1\}$ | **return** $T_b$ |
| **procedure** $\textsc{Prim}(J, X)$ | **procedure Finalize**$(b')$ |
| **if** $X = (+, x)$ **then return** $E_J(x)$ | **return** $(b' = b)$ |
| **if** $X = (-, y)$ **then return** $E_J^{-1}(y)$ | |

Fig. 9: Game $\mathbf{G}_{\mathsf{DbHtS}}^{\mathrm{mprf}}$ defining the multi-user prf security of construction $\mathsf{DbHtS}$.

**Lemma 5.** [36] *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that is assumed to be ideal. Suppose that each $n$-bit hash function of $H = (H^1, H^2)$ is $\epsilon_1$-regular and $\epsilon_2$-almost universal. Then for any adversary $\mathcal{A}$ that makes at most $q$ MAC queries and $p$ ideal-cipher queries,*

$$\begin{aligned}\mathsf{Adv}_{\mathsf{DbHtS}}^{\mathrm{mprf}}(\mathcal{A}) \le \frac{q}{2^n} + \frac{2q}{2^k} + \frac{q(3q+p)(6q+2p)}{2^{2k}} + \frac{2qp\ell}{2^{n+k}} + \frac{2qp\epsilon_1}{2^k} + \frac{4qp}{2^{n+k}} \\ + q\epsilon_1 + \frac{4q^2\epsilon_1}{2^k} + \frac{2q^2\ell\epsilon_1}{2^k} + 2q^3(\epsilon_2^2 + 2\epsilon_1\epsilon_2) + q^2\epsilon_3 + q^3\epsilon_4 \\ + \frac{8q^3(\epsilon_1+\epsilon_2)}{2^n} + \frac{6q^3}{2^{2n}} \ ,\end{aligned}$$

*by assuming $p + q\ell \le 2^{n-1}$, where $\ell$ is the maximal block length among these $q$ MAC queries.*

Note that the terms $\epsilon_3$ and $\epsilon_4$ are used to capture two joint events of hash function $H^1$ and $H^2$ that are $\Sigma_i^a = \Lambda_i^b \wedge \Lambda_i^a = \Sigma_i^b$ and $\Sigma_i^a = \Lambda_i^b \wedge \Lambda_i^a = \Sigma_i^c$ respectively. These two special joint events cannot be simply bounded by $\epsilon_1$-regular and $\epsilon_2$-almost universal of hash functions as pointed out by Guo and Wang [21]. The values $\epsilon_3$ and $\epsilon_4$ will become clear in the following analysis. We refer the reader to [36] for more detailed discussions about this framework. Hence to apply this lemma, we need to show that each $n$-bit hash function of $H_{K_h} = (H_L^1, H_L^2)$ of $\mathsf{n2kf9}$ is $\epsilon_1$-regular and $\epsilon_2$-almost universal, and analyze the relation between these two $n$-bit hash function.

**Lemma 6.** *Assume that $\ell \le 2^{n/4}$. Then $H_L^1$ is $(\frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}})$-regular and $(\frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}})$-almost universal, and $H_L^2$ is $(\frac{2}{2^n} + \frac{\ell^2}{2^n})$-regular and $(\frac{2}{2^n} + \frac{4\ell^2}{2^n})$-almost universal.*

*Proof.* For $H_L^1$ which is exactly the $\mathsf{CBC\text{-}MAC}$ algorithm, Bellare et al. [8] and Jha and Nandi [26] show that

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

for any two distinct messages $M_1$ and $M_2$ of maximal block length $\ell$. This directly implies that $H_L^1$ is $\epsilon_2^1$-almost universal where $\epsilon_2^1 = \frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}}$. Equipping with this result, Shen et al. [36] also show that $H_L^1$ is $\epsilon_1^1$-regular where $\epsilon_1^1 = \frac{2\sqrt{\ell}}{2^n} + \frac{16\ell^4}{2^{2n}}$. For $H_L^2$, we first show that it meets the regular property. The equation $H_L^2(M) = Z$ is the same as

$$2 \cdot (Y_1 \oplus \cdots \oplus Y_\ell) = Z \ .$$

Recall that $\mathsf{Acc}(G_E^M)$ denotes the set of accidents of structure graph $G_E^M$. If $|\mathsf{Acc}(G_E^M)| = 0$, then obviously the rank of this equation is 1. From Lemma 9, we have

$$\Pr\Big[\, H_L^2(M) = Z \wedge |\mathsf{Acc}(G_E^M)| = 0 \,\Big] \leq \frac{1}{2^n - \ell + 1} \leq \frac{2}{2^n}$$

by assuming $\ell \leq 2^{n-1}$. According to Lemma 8, the probability that there is at least 1 accident, that is $|\mathsf{Acc}(G_E^M)| \geq 1$, is at most $\ell^2/2^n$. Hence $H_L^2$ is $\epsilon_1^2$-regular where $\epsilon_1^2 = \frac{2}{2^n} + \frac{\ell^2}{2^n}$. Similarly, we can prove that $H_L^2$ is $\epsilon_2^2$-almost universal where $\epsilon_2^2 = \frac{2}{2^n} + \frac{4\ell^2}{2^n}$ by assuming $\ell \leq 2^{n-2}$.

In the following, let $\epsilon_1 = \max\{\epsilon_1^1, \epsilon_1^2\} = \frac{2}{2^n} + \frac{\ell^2}{2^n}$, and $\epsilon_2 = \max\{\epsilon_2^1, \epsilon_2^2\} = \frac{2}{2^n} + \frac{4\ell^2}{2^n}$. However, until now we cannot simply apply Lemma 5 to $\mathsf{n2kf9}$ since it assumes that $K_{h,1}$ and $K_{h,2}$ are two independent keys while $K_{h,1}$ and $K_{h,2}$ are identical in $\mathsf{n2kf9}$, and the terms $q^2\epsilon_3$ and $q^3\epsilon_4$ still remain to be analyzed. Look more closely at the framework, the terms relying on the independence of these two keys are $q\epsilon_1$ and $2q^3(\epsilon_2^2 + 2\epsilon_1\epsilon_2) + q^2\epsilon_3 + q^3\epsilon_4$, which are the probabilities of three bad events in the multi-user setting, namely for some user $a$

i. there exists an entry $(M_i^a, T_i^a)$ such that $\Sigma_i^a = \Lambda_i^a$;
ii. there exists a pair of entries $(M_i^a, T_i^a)$ and $(M_j^a, T_j^a)$ such that $\Sigma_i^a = \Sigma_j^a$ and $\Lambda_i^a = \Lambda_j^a$, or $\Sigma_i^a = \Lambda_j^a$ and $\Lambda_i^a = \Sigma_j^a$;
iii. there exists a triplet of entries $(M_i^a, T_i^a)$, $(M_j^a, T_j^a)$ and $(M_k^a, T_k^a)$ such that $\Sigma_i^a = \Sigma_j^a$ or $\Sigma_i^a = \Lambda_j^a$, and $\Lambda_i^a = \Sigma_k^a$ or $\Lambda_i^a = \Lambda_k^a$.

These three bad events are exactly the same as $\mathsf{bad}_2$, $\mathsf{bad}_3$ and $\mathsf{bad}_4$ in the single-user setting. We can similarly define the $\mathsf{crash}$ event in the multi-user setting:

$$\mathsf{crash} = |\mathsf{Acc}(G_\pi^{M_i^a})| \geq 1 \textbf{ or } |\mathsf{Acc}(G_\pi^{\{M_i^a, M_j^a\}})| \geq 2 \textbf{ or } |\mathsf{Acc}(G_\pi^{\{M_i^a, M_j^a, M_k^a\}})| \geq 3$$

Suppose that the adversary issues totally $q$ queries to $u$ users, and denote by $q_i$ the number of queries to the $i$-th user. Then from Equations (3), (4), (5) and (6),

$$\Pr[\,\mathsf{crash}\,] \leq \sum_{i=1}^{u} \frac{q_i\ell^2}{2^n} + \frac{8q_i^2\ell^4}{2^{2n}} + \frac{122q_i^3\ell^6}{2^{3n}} \leq \frac{q\ell^2}{2^n} + \frac{8q^2\ell^4}{2^{2n}} + \frac{122q^3\ell^6}{2^{3n}}$$

and

$$\Pr[\,\text{event i} \wedge \neg\mathsf{crash}\,] \leq \sum_{i=1}^{u} \frac{2q_i}{2^n} \leq \frac{2q}{2^n}$$

and

$$\Pr[\,\text{event ii} \wedge \neg\mathsf{crash}\,] \leq \sum_{i=1}^{u} \frac{8q_i^2\ell^2}{2^{2n}} + \frac{2q_i^2}{2^{2n}} \leq \frac{8q^2\ell^2}{2^{2n}} + \frac{2q^2}{2^{2n}}$$

and

$$\Pr[\,\text{event iii} \wedge \neg\mathsf{crash}\,] \leq \sum_{i=1}^{u} \frac{12q_i^3\ell^2}{2^{2n}} + \frac{2q_i^3}{2^{2n}} + \frac{108q_i^3\ell^4}{2^{3n}} \leq \frac{12q^3\ell^2}{2^{2n}} + \frac{2q^3}{2^{2n}} + \frac{108q^3\ell^4}{2^{3n}} \quad .$$

Hence from Lemma 5, Lemma 6 and above analysis, we can obtain the multi-user security of $\mathsf{n2kf9}$.

## B Structure Graph of CBCMAC

The structure graph of CBC-MAC has been rigorously studied by Bellare et al. [8] and Jha and Nandi [26], and is extensively used to prove security bounds of several cryptographic algorithms [20,22]. Here we briefly recall the structure graph whose presentation partially follows from [22], and highlight some results that will be helpful in our analysis.

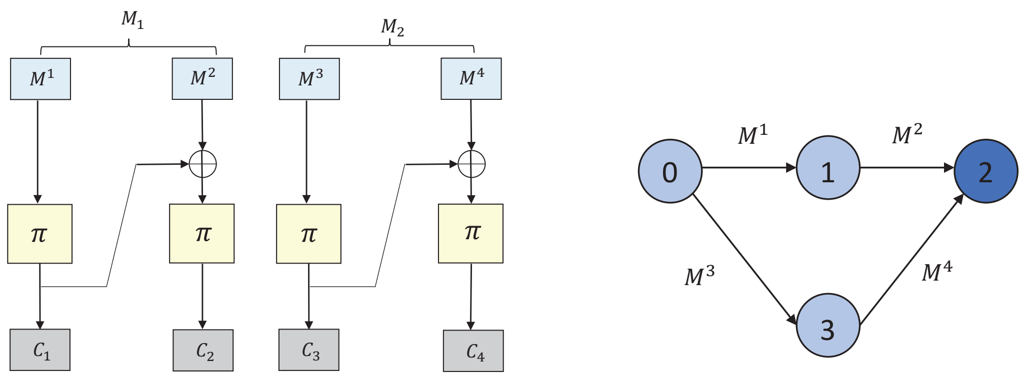


Fig. 10: **Left:** Illustration of the blocks $C_1, \ldots, C_4$ for the case of two messages $M_1$ and $M_2$, and there are $\sigma = 4$ blocks. **Right:** The corresponding structure graph of these two messages, assuming that the blocks $C_1, C_2, C_3$ are distinct and $C_4 = C_2$.

Structure graph of CBC-MAC. For $1 \le i \le q$, let $\ell_i$ be the block length of message $M_i$, and $\sigma_i = \sum_{j=1}^{i} \ell_j$ be the total block length of the first $i$ messages $M_1, \ldots, M_i$, and $\sigma_0 = 0$ for completeness. For simplicity, let $\sigma = \sum_{i=1}^{q} \ell_i$ and thus $\sigma = \sigma_q$. For $q$ distinct messages $M_1, \ldots, M_q$, let $\boldsymbol{M} = M_1 \parallel \cdots \parallel M_q$ be the concatenation of these messages, and split it as $M^1 \cdots M^\sigma$ where each $M^i$ is an $n$-bit block. Let $C_0 = 0^n$, and for each $i \in \{1, \ldots, \sigma\}$, define

$$C_i = \begin{cases} \pi(C_{i-1} \oplus M^i) & \text{if } i \notin \{\sigma_0 + 1, \ldots, \sigma_{q-1} + 1\} \\ \pi(M^i) & \text{otherwise.} \end{cases}$$

For each $i \in \{0, \ldots, \sigma\}$, define the mapping $\mathsf{map}(i) = \min\{j : C_j = C_i\}$, and let $\mathsf{pmap}(i) = \mathsf{map}(i)$ if $i \notin \{\sigma_0, \ldots, \sigma_{q-1}\}$ and $\mathsf{pmap}(i) = 0$ otherwise. Then given $\boldsymbol{M}$ and a permutation $\pi \in \mathrm{Perm}(n)$, the corresponding *structure graph* $G_\pi^{\boldsymbol{M}} = (V, E, L)$ of CBC-MAC is a directed graph where node set $V = \{\mathsf{map}(i) \mid 0 \le i \le \sigma\}$, edge set $E = \{e_i = (\mathsf{pmap}(i-1), \mathsf{map}(i)) \mid 1 \le i \le \sigma\}$, and edge-labeling function $L(e_i) = M^i$. See Figure 10 for an illustration of two messages $M_1$ and $M_2$.

Suppose a structure graph $G_\pi^{\boldsymbol{M}}$ is exposed edge by edge that the edge $e_i$ appears in the $i$th step. A *collision* occurs at the step $i$ if the exposed edge $e_i$ points to a node which is already in the graph. We classify collisions into two types, namely *induced collisions* and *accidents*. Informally, an induced collision at the $i$-th step is a collision that is implied by the collisions from previous steps, while an accident is an accidental collision. We refer the reader to [8] for a rigorous treatment of two types of collisions.

Take two messages $M_1$ and $M_2$ in Figure 11 as an example. Here $M_1 = M^1 \cdots M^5$ is a five-block message where $M^4 = M^2$, and $M_2 = M^6$ is a one-block message where $M^6 = M^1 \oplus M^3 \oplus M^5$. We assume that $C_3 = C_1$, and thus

$$C_2 \oplus M^3 = C_0 \oplus M^1 \ .$$

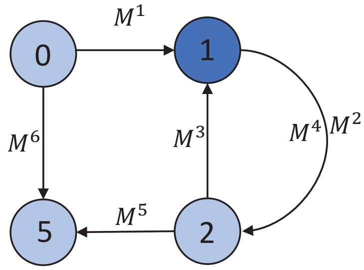


Fig. 11: Illustration of accident and induced collision.

Then

$$C_6 = \pi(C_0 \oplus M^6) = \pi(C_0 \oplus M^1 \oplus M^3 \oplus M^5) = \pi(C_2 \oplus M^5) = C_5 \ .$$

Thus the collision $C_3 = C_1$ is an accident while the collision $C_6 = C_5$ is an induced collision.

Let $\mathcal{G}(\boldsymbol{M})$ be the set of structure graphs of $\boldsymbol{M}$ for all $\pi \in \mathrm{Perm}(n)$, let $\mathsf{Acc}(G_\pi^{\boldsymbol{M}})$ be the set of accidents of $G_\pi^{\boldsymbol{M}}$. Bellare et al. [8] showed that $G \in \mathcal{G}(\boldsymbol{M})$ is uniquely determined by $\mathsf{Acc}(G)$ and $\boldsymbol{M}$ alone.

For a collision that is formed by an edge $e_i$ pointing to a prior node $j$, if there is no prior edge $e_k$ pointing to $j$, then $j$ must be the node 0. A flag $\mathsf{zero}(G_\pi^{\boldsymbol{M}})$ will be set to 1 and an equation $C_i = 0^n$ will appear if there exist an edge $e_i$ pointing node 0, and otherwise the flag will remain to be 0. If there is a prior edge $e_k$ that points to the same node $j$, then we say this is a true collision. For a true collision that is formed from two edges $e_i = (u, j)$ and $e_k = (v, j)$, it implies that $C_u \oplus M^i = C_v \oplus M^k$ since both of them are equal to $\pi^{-1}(C_j)$, and thus

$$C_u \oplus C_v = M^i \oplus M^k$$

where $C_u$ and $C_v$ are variables, and $M^i$ and $M^k$ are two $n$-bit strings. Let $\mathsf{rank}(G_\pi^{\boldsymbol{M}})$ be the rank of the system of linear equations induced by the true collisions of $G_\pi^{\boldsymbol{M}}$. Then $|\mathsf{Acc}(G_\pi^{\boldsymbol{M}})| = \mathsf{rank}(G_\pi^{\boldsymbol{M}}) + \mathsf{zero}(G_\pi^{\boldsymbol{M}})$, which is the definition of Jha and Nandi [26] for the number of accidents. We now recall the known results on structure graph.

**Lemma 7.** [8] *The number of structure graphs of $\boldsymbol{M} = M_1 \parallel \cdots \parallel M_q$ with $a$ accidents is at most $\binom{\sigma}{2}^a$. In particular, there exists exactly one structure graph with no accident.*

**Lemma 8.** [26,8] *For a given $\boldsymbol{M} = M_1 \parallel \cdots \parallel M_q$ of total block length $\sigma$ and any integer $a \geq 1$,*

$$\Pr_{\pi \leftarrow\$ \mathrm{Perm}(n)}\Big[|\mathsf{Acc}(G_\pi^{\boldsymbol{M}})| \geq a\Big] \leq \left(\frac{\sigma^2}{2^n}\right)^a \ .$$

We also recall a known result on the rank of variables in the structure graph. For these $\sigma$ variables $(C_1, \ldots, C_\sigma)$ in the structure graph $G_\pi^{\boldsymbol{M}}$, we say two variables $C_i$ and $C_j$ are in the same class if $C_i = C_j$, and suppose these $\sigma$ variables can be divided into $v$ classes $\mathcal{C}_1, \ldots, \mathcal{C}_v$. Let $x_i = \min\{k : C_k \in \mathcal{C}_i\}$ be the minimal index of variables among each class $\mathcal{C}_i$. A normalized variable of class $\mathcal{C}_i$ is the variable that has the minimal index among class $\mathcal{C}_i$, e.g., $C_{x_i}$ is the normalized variable of class $\mathcal{C}_i$. For $1 \leq i \leq s$, let $E_i$ denote a linear equation over variables $(C_1, \ldots, C_\sigma)$ of the form $a_{i,1}C_1 \oplus \cdots \oplus a_{i,\sigma}C_\sigma \oplus c_i = 0$ where $a_{i,j}, c_i \in \{0,1\}^n$. Let $\mathcal{E} = \{E_1, \ldots, E_s\}$ be the system of these linear equations. We replace each variable $C_i$ in $\mathcal{E}$ by the corresponding normalized variable as follows: if $C_i \in \mathcal{C}_j$, then replace $C_i$ with $C_{x_j}$. After that, we will obtain the normalized version of $\mathcal{E}$ that is denoted by $\widetilde{\mathcal{E}}$. Note that the linear equations in $\mathcal{E}$ are equivalent to those in $\widetilde{\mathcal{E}}$. We will use the following algebraic result for $\mathcal{E}$, the proof of which can be found in the paper of Datta et al. [14, Lemma 6].

**Lemma 9.** *Let $G_\pi^{\boldsymbol{M}}$ be a structure graph that is realized through variables $(C_1, \dots, C_\sigma)$. Suppose there are $v$ distinct classes among these variables. Let $\mathcal{E}$ be the system of linear equations over these $\sigma$ variables, and $\widetilde{\mathcal{E}}$ the normalized version of $\mathcal{E}$. If the rank of $\widetilde{\mathcal{E}}$ is $r$, then*

$$\Pr\left[\, \mathcal{E} \textit{ holds } \right] \le \frac{1}{(2^n - v + r)_r} \enspace .$$

## C Proof of Lemma 4

In this section, we bound the probability that $X_0$ is bad in the ideal world. For $1 \le i \le 9$, denote by $\mathsf{bad}_i$ the event when the $i$th condition is violated. We first consider event $\mathsf{bad}_1$. Recall that in the ideal world, each $T_i$ is a random $n$-bit string. Hence the probability that $T_i = 0^n$ is exactly $1/2^n$. Summing over at most $q$ queries,

$$\Pr\left[\, \mathsf{bad}_1 \,\right] = \frac{q}{2^n} \enspace .$$

The analyses of events from 2 to 9 are based on the structure graph of CBC-MAC that is recalled in Section B. In the rest of this section, let $\overline{M}_i = |M_i|_n \parallel M_i 10^* \parallel 0^n$.

Restricting the accidents. Before bounding the probability, it is useful to bound the number of accidents that arise within any single, pair or triplet of messages. Define the following event for any three distinct messages $M_i$, $M_j$ and $M_k$:

$$\mathsf{crash} = |\mathsf{Acc}(G_\pi^{\overline{M}_i})| \ge 1 \textbf{ or } |\mathsf{Acc}(G_\pi^{\{\overline{M}_i, \overline{M}_j\}})| \ge 2 \textbf{ or } |\mathsf{Acc}(G_\pi^{\{\overline{M}_i, \overline{M}_j, \overline{M}_k\}})| \ge 3$$

From Lemma 8,

$$\Pr\left[\, \mathsf{crash} \,\right] \le \frac{q(\ell+2)^2}{2^n} + \frac{8q^2(\ell+2)^4}{2^{2n}} + \frac{122q^3(\ell+2)^6}{2^{3n}} \enspace .$$

We then analyze the probability of events from 2 to 9 only in conjunction with $\neg\mathsf{crash}$. That is only when there is no accident within any single message, and at most one accident within any pair of messages, and at most two accidents within any triplet of messages.

Proof ideas of these events. Note that proof ideas of these events are essentially similar to those in section 5, except that we need to consider the impact of $\Sigma_i$ and $\Lambda_i$ colliding with internal input values of CBC-MAC.

Probability analyses of these events. For the event 2, it is exactly the same as event 2 of Lemma 3, hence

$$\Pr\left[\, \mathsf{bad}_2 \wedge \neg\mathsf{crash} \,\right] \le \frac{2q}{2^n} \enspace .$$

We then bound the probability of event 3. If this event happens, then it requires an accident on the structure graph $G_\pi^{\overline{M}_i}$, which contradicts with the event $\neg\mathsf{crash}$. Hence,

$$\Pr\left[\,\mathsf{bad}_3 \wedge \neg\mathsf{crash}\,\right] = 0\ .$$

Next, we bound the probability of event 4. This event can be decomposed into two subcases: (4-i) $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j\}$ and $\Lambda_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$; (4-ii) $\Sigma_i \in \{\Lambda_j\}$ and $\Lambda_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$. The first subcase requires one accident that is determined by the first set. Hence we only need to consider it when $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 1$. Similarly to the event 3 in Lemma 3, the rank of any two equations when variables $\Sigma_i$ and $\Lambda_i$ equal to values from above two sets is 2: for the two equations from $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j\}$ and $\Lambda_i \in \{\Lambda_j\}$, we can justify it by similar arguments as in the subcase (i) of event 3; for the two equations from $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j\}$ and $\Lambda_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j\}$, we can justify it by the coefficient 2 that only appears in the second equation and will not be canceled out. Hence from Lemma 9,

$$\begin{aligned}&\Pr\left[\text{event (4-i)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 1\right]\\ &\leq \frac{1}{(2^n-2\ell-2)_2} \cdot (\ell+2)(\ell+3) \leq \frac{16(\ell+3)^2}{2^{2n}}\end{aligned}$$

by assuming $\ell \leq 2^{n-2}-2$. We then analyze subcase (4-ii). Following analogous arguments from subcase (ii) of event 3 in Lemma 3, if $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 0$, then the rank of any two equations from above two sets is 2. From Lemma 9,

$$\begin{aligned}&\Pr\left[\text{event (4-ii)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 0\right]\\ &\leq \frac{1}{(2^n-2\ell-2)_2} \cdot (\ell+3) \leq \frac{4(\ell+3)}{2^{2n}}\ .\end{aligned}$$

On the other hand, if $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 1$, then similarly, the rank of these two equations is also 2: for the two equations from $\Sigma_i \in \{\Lambda_j\}$ and $\Lambda_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j\}$, we can justify it by similar arguments as in the subcase (ii) of event 3 in Lemma 3; for the two equations from $\Sigma_i \in \{\Lambda_j\}$ and $\Lambda_i \in \{\Lambda_j\}$, we can justify it by the coefficient 2 since it will be canceled out in the second equation and only appears in the first equation. Again from Lemma 9,

$$\begin{aligned}&\Pr\left[\text{event (4-ii)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 1\right]\\ &\leq \frac{1}{(2^n-2\ell-2)_2} \cdot \binom{2\ell+4}{2} \cdot (\ell+3) \leq \frac{8(\ell+3)^3}{2^{2n}}\ .\end{aligned}$$

From the union bound and summing over at most $\binom{q}{2}$ pairs of $M_i$ and $M_j$,

$$\Pr\left[\,\mathsf{bad}_4 \wedge \neg\mathsf{crash}\,\right] \leq \frac{8q^2(\ell+3)^3}{2^{2n}} + \frac{2q^2(\ell+3)}{2^{2n}}\ .$$

We then bound the probability of event 5. For the case $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j, \Lambda_j\}$, if $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 0$, then $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j\}$ cannot happen since it requires at least one accident, and $\Sigma_i \in \{\Lambda_j\}$ happens with probability $1/(2^n - \ell - 2) \leq 2/2^n$ by assuming $\ell \leq 2^{n-1} - 2$. If $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j\}})| = 1$, then from Lemma 8, this condition itself holds with probability at most $4(\ell+2)^2/2^n$. For the case $V_i \in \{Y_0^j, \ldots, Y_{\ell_j}^j, U_j, V_j\}$, since $V_i = U_i \oplus T_i$ is a random $n$-bit string, it holds with probability at most $(\ell+3)/2^n$. Summing over at most $\binom{q}{2}$ pairs of queries,

$$\Pr[\,\mathsf{bad}_5 \wedge \neg\mathsf{crash}\,] \leq \binom{q}{2} \cdot \left(\frac{2}{2^n} + \frac{4(\ell+2)^2}{2^n}\right) \cdot \frac{\ell+3}{2^n} \leq \frac{3q^2(\ell+3)^3}{2^{2n}} \quad .$$

Similarly,

$$\Pr[\,\mathsf{bad}_6 \wedge \neg\mathsf{crash}\,] \leq \binom{q}{2} \cdot \left(\frac{2(\ell+3)}{2^n} + \frac{4(\ell+2)^2}{2^n}\right) \cdot \frac{\ell+3}{2^n} \leq \frac{3q^2(\ell+3)^3}{2^{2n}} \quad .$$

Next, we bound the probability of event 7. This event can be decomposed into two subcases: (7-i) $\Sigma_i \in \{|M_j|_n, Y_0^j \oplus M_j[1], \ldots, Y_{\ell_j-1}^j \oplus M_j[\ell_j], \Sigma_j\}$ and $\Lambda_i \in \{|M_k|_n, Y_0^k \oplus M_k[1], \ldots, Y_{\ell_k-1}^j \oplus M_k[\ell_k], \Sigma_k, \Lambda_k\}$; (7-ii) $\Sigma_i \in \{\Lambda_j\}$ and $\Lambda_i \in \{|M_k|_n, Y_0^k \oplus M_k[1], \ldots, Y_{\ell_k-1}^j \oplus M_k[\ell_k], \Sigma_k, \Lambda_k\}$. Similarly to subcases (i) and (ii) of event 6 in Lemma 3, if $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})| = 0$, then subcase (7-i) cannot happen since it requires at least one accident. If $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})| = 1$, then the accident is determined by the first set and the rank of any two equations from above two sets is 2. From Lemma 9,

$$\Pr\left[\text{event (7-i)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})| = 1\right]$$
$$\leq \frac{1}{(2^n - 3\ell - 4)_2} \cdot (\ell+2)(\ell+3) \leq \frac{12(\ell+3)^2}{2^{2n}}$$

by assuming $\ell \leq 2^{n-3} - 2$. If $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})| = 2$, then similarly we have

$$\Pr\left[\text{event (7-i)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})| = 2\right]$$
$$\leq \frac{1}{(2^n - 3\ell - 4)_3} \cdot \binom{3\ell+6}{2} \cdot (\ell+2)(\ell+3) \leq \frac{36(\ell+3)^4}{2^{3n}}$$

Similarly to subcases (iii) and (iv) of event 6 in Lemma 3, for the subcase (7-ii), if $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})| = 0$, then the rank of any two equations from above two sets is 2. From Lemma 9,

$$\Pr\left[\text{event (7-ii)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})| = 0\right]$$
$$\leq \frac{1}{(2^n - 3\ell - 4)_2} \cdot (\ell+3) \leq \frac{4(\ell+3)}{2^{2n}} \quad .$$

If $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})|=1$, then similarly the rank of any two equations from above two sets is also 2. From Lemma 9,

$$\Pr\left[\text{event (7-ii)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})|=1\right]$$
$$\le \frac{1}{(2^n-3\ell-4)_2}\cdot\binom{3\ell+6}{2}\cdot(\ell+3)\le\frac{6(\ell+3)^3}{2^{2n}} \quad .$$

If $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})|=2$, then the rank of these two equations plus two accidental equations is at least 3. Hence from Lemma 9,

$$\Pr\left[\text{event (7-ii)} \wedge |\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})|=2\right]$$
$$\le \frac{1}{(2^n-3\ell-4)_3}\cdot\binom{3\ell+6}{2}^2\cdot(\ell+3)\le\frac{6(\ell+3)^5}{2^{3n}} \quad .$$

From the union bound and summing over at most $\binom{q}{3}$ triplets of $(M_i,M_j,M_k)$,

$$\Pr\left[\,\mathsf{bad}_7\wedge\neg\mathsf{crash}\,\right]\le\frac{q^3(\ell+3)^3}{2^{2n}}+\frac{2q^3(\ell+3)^2}{2^{2n}}+\frac{q^3(\ell+3)}{2^{2n}}+\frac{q^3(\ell+3)^5}{2^{3n}}+\frac{6q^3(\ell+3)^4}{2^{3n}} \quad .$$

Bounding the probability of event 8 is similar to handling event 5, with the difference that there are at most $\binom{q}{3}$ triplets of queries and the probability that $|\mathsf{Acc}(G_\pi^{\{\overline{M}_i,\overline{M}_j,\overline{M}_k\}})|\ge 1$ is bounded by $9(\ell+2)^2/2^n$. Hence,

$$\Pr\left[\,\mathsf{bad}_8\wedge\neg\mathsf{crash}\,\right]\le\binom{q}{3}\cdot\left(\frac{2}{2^n}+\frac{9(\ell+2)^2}{2^n}\right)\cdot\frac{\ell+3}{2^n}\le\frac{2q^3(\ell+3)^3}{2^{2n}} \quad .$$

Similarly,

$$\Pr\left[\,\mathsf{bad}_9\wedge\neg\mathsf{crash}\,\right]\le\binom{q}{3}\cdot\left(\frac{2(\ell+3)}{2^n}+\frac{9(\ell+2)^2}{2^n}\right)\cdot\frac{\ell+3}{2^n}\le\frac{2q^3(\ell+3)^3}{2^{2n}} \quad .$$

Summing up,

$$\sum_{j=2}^{9}\Pr\left[\,\mathsf{bad}_j\,\right]\le\Pr\left[\,\mathsf{crash}\,\right]+\sum_{j=2}^{9}\Pr\left[\,\mathsf{bad}_j\wedge\neg\mathsf{crash}\,\right]$$
$$\le\frac{5q^3(\ell+3)^3}{2^{2n}}+\frac{3q^3(\ell+3)^2}{2^{2n}}+\frac{24q^2(\ell+2)^4}{2^{2n}}+\frac{122q^3(\ell+2)^6}{2^{3n}}$$
$$+\frac{7q^3(\ell+3)^5}{2^{3n}}+\frac{q(\ell+2)^2}{2^n}+\frac{2q}{2^n} \quad .$$

## D Offline Oracle in n1kf9

$$
\begin{array}{l}
\textbf{procedure } \textsc{Off}(q) \\
\textbf{for } i \leftarrow 1 \textbf{ to } q \textbf{ do} \\
\quad M_i[1] \parallel \ldots \parallel M_i[\ell_i] \leftarrow M;\ Y_0^i \leftarrow \pi(|M_i|_n) \\
\quad \textbf{for } j \leftarrow 1 \textbf{ to } \ell_i \textbf{ do} \\
\quad\quad Y_j^i \leftarrow \pi(Y_{j-1}^i \oplus M_i[j]) \\
\quad \varSigma_i = Y_{\ell_i}^i;\ \varLambda_i = 2 \cdot (Y_0^i \oplus Y_1^i \oplus \cdots \oplus Y_{\ell_i}^i) \\
\quad \mathcal{H} = \mathcal{H} \cup \{Y_0^i, \ldots, Y_{\ell_i}^i, \varLambda_i\} \\
\mathcal{F} = \{i : \text{ both } \varSigma_i \text{ and } \varLambda_i \text{ are fresh in } \mathcal{H}\} \\
\mathcal{G} = \{i : \text{ only one of } \varSigma_i \text{ and } \varLambda_i \text{ is fresh in } \mathcal{H}\} \\
\mathcal{N} = \{i : \text{ neither } \varSigma_i \text{ nor } \varLambda_i \text{ is fresh in } \mathcal{H}\} \\
\mathcal{I}\text{: set of tuples of } 2\,|\mathcal{F}| \text{ distinct values from } \{0,1\}^n \setminus \mathcal{H} \\
\mathcal{S} = \{(W_i, X_i)_{i \in \mathcal{F}} \in \mathcal{I} : W_i \oplus X_i = T_i\} \\
(U_i, V_i)_{i \in \mathcal{F}} \leftarrow\!\!\$\ \mathcal{S} \\
\forall i \in \mathcal{F} : (\pi(\varSigma_i), \pi(\varLambda_i)) \leftarrow (U_i, V_i) \\
\forall\, i \in \mathcal{G} : \\
\quad \textbf{if } \varSigma_i \text{ is not fresh in } \mathcal{H} \textbf{ then} \\
\quad\quad \textbf{if } \varSigma_i \notin \mathrm{Dom}(\pi) \\
\quad\quad\quad \textbf{then } U_i \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathrm{Rng}(\pi);\ \pi(\varSigma_i) \leftarrow U_i \\
\quad\quad \textbf{else } U_i \leftarrow \pi(\varSigma_i) \\
\quad\quad V_i \leftarrow T_i \oplus U_i;\ \pi(\varLambda_i) \leftarrow V_i \\
\quad \textbf{else} \\
\quad\quad \textbf{if } \varLambda_i \notin \mathrm{Dom}(\pi) \\
\quad\quad\quad \textbf{then } V_i \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathrm{Rng}(\pi);\ \pi(\varLambda_i) \leftarrow V_i \\
\quad\quad \textbf{else } V_i \leftarrow \pi(\varLambda_i) \\
\quad\quad U_i \leftarrow T_i \oplus V_i;\ \pi(\varSigma_i) \leftarrow U_i \\
\exists\, i \in \mathcal{N} : \textbf{return } \perp \\
\textbf{return } (U_i, V_i)_{1 \le i \le q}
\end{array}
$$

Fig. 12: **Offline oracle used in the proof of** `n1kf9`**.** Here the set $\mathcal{H}$ is initialized to be empty at the beginning. Variables $\varSigma_i$ and $\varLambda_i$ are inputs of a random permutation, and $U_i$ and $V_i$ are corresponding outputs of this random permutation. Sets $\mathrm{Dom}(\pi)$ and $\mathrm{Rng}(\pi)$ start off to be empty and automatically grow when point is applied by permutation $\pi$, e.g., if $y \leftarrow \pi(x)$, then $\mathrm{Dom}(\pi) = \mathrm{Dom}(\pi) \cup \{x\}$ and $\mathrm{Rng}(\pi) = \mathrm{Rng}(\pi) \cup \{y\}$.