Multi-User BBB Security of Public Permutations Based MAC

Yu Long Chen[1] and Avijit Dutta[2] and Mridul Nandi[3]

KU Leuven, Belgium
Institute for Advancing Intelligence, TCG-CREST, India
Indian Statistical Institute, Kolkata
yulong.chen@kuleuven.be,avirocks.dutta13@gmail.com,mridul.nandi@gmail.com

Abstract. At CRYPTO 2019, Chen et al. have shown a beyond the birthday bound secure n-bit to n-bit PRF based on public random permutations. Followed by the work, Dutta and Nandi have proposed a beyond the birthday bound secure nonce based MAC nEHtM_p based on public random permutation. In particular, the authors have shown that nEHtM_p achieves tight $2n/3$-bit security (*with respect to the state size of the permutation*) in the single-user setting, and their proven bound gracefully degrades with the repetition of the nonces. However, we have pointed out that their security proof is not complete (albeit it does not invalidate their security claim). In this paper, we propose a minor variant of nEHtM_p construction, called nEHtM_p^* and show that it achieves a tight $2n/3$ bit security in the multi-user setting. Moreover, the security bound of our construction also degrades gracefully with the repetition of nonces. Finally, we have instantiated our construction with the PolyHash function to realize a concrete beyond the birthday bound secure public permutation-based MAC, nEHtM_p^+ in the multi-user setting.

Keywords: Faulty Nonce, Mirror Theory, Public Permutation, Expectation Method

1 Introduction

The purpose of analyzing a cryptographic construction is not only to model the real-world settings, but also to accurately capture the practical limits imposed by the real-world environments, the desired security properties, and should also precisely assess the degradation of its security with its use. When a cryptographic algorithm is deployed in real-time protocols, then based on the requirement, the key size, the block size, and other various parameters of the construction are set to some fixed values. Therefore, to estimate the security of the construction, one needs to evaluate the adversarial success probability in breaking the system in terms of the adversary's resource. For example, the model that we rely on for analyzing the security of a block cipher is that the adversary is given access to an encryption and decryption oracle, keyed with a secret value chosen uniformly at random. The job of the adversary is to distinguish it from a random instance of a permutation. This is formally called the *indistinguishability setting*. For many practical purposes, the indistinguishability of a block cipher from a random instance of a permutation suffices for establishing its security. However, we estimate the adversarial success by analyzing the best-known attacks against the block cipher with respect to the adversary's computational complexity, which measures the cost of running the attack in terms of time, memory, and also the data complexity. The data complexity measures the amount of data the adversary transmitted during the interaction with the oracle. For example, the best-known attacks on full round AES-128 have computational complexity improving over the naive brute-force search by a factor 2 to 4, whereas increasing the data complexity does not help much to reduce computational complexity.

On the other extreme, assessing the adversarial success probability is different for modes of operations. These are cryptographic algorithms that repeatedly use block cipher calls to achieve some specific security properties beyond what a block cipher is supposed to provide on its own.

Message Authentication Code (MAC) is a symmetric key cryptographic algorithm used to provide both authenticity and integrity for any digital message transmitted over an insecure communication channel. When a sender wants to send a message m, it computes the tag t corresponding to message m by evaluating a secret keyed function F that takes as input m, the shared secret key k, and possibly an auxiliary input variable ν (called nonce). Then it sends (ν, m, t) to the receiver. Upon receiving, the receiver verifies the authenticity of (ν, m, t) by computing the function F with the received tuple (ν, m, t) and checks whether the computed tag t' matches with t.

One of the popular MAC algorithms is PMAC [12]. It uses the block cipher repeatedly to provide integrity and authenticity for a message. Its security property is formalized in a setting where the adversaries are given access to a keyed signing oracle and a verification oracle. The security of the construction is proved by reducing the advantage of the MAC adversary against the mode to the advantage of an adversary against the pseudo-randomness of the underlying block cipher. Therefore, assuming AES is a secure pseudorandom permutation (PRP), we prove that AES-PMAC is a secure MAC. However, the quality of this reduction from AES to AES-PMAC deteriorates with use. In fact, following the concrete security bound [4], this degradation has been quantified to be roughly $O(\ell q^2/2^{128})$ [44], where ℓ denotes the maximum number of message blocks and q denotes the number of queries. Therefore, the MAC security of AES-PMAC relies not only on the security of the underlying block cipher (e.g., AES) but also on the degradation of the security as a mode. Note that the security of AES degrades as we increase the computational resources of the adversary. Still, increased data complexity does not seem to affect the advantage of the attack. In contrast, the security of AES-PMAC degrades as data complexity increases, but increased computational complexity does not seem to have a role in its security.

1.1 Nonce Based MAC and Resilience to Faulty Nonce

Wegman-Carter MAC [47] is the first example of a nonce-based MAC that masks the hash value of the message with an encrypted nonce to generate the tag. Although it gives optimal security when the nonce is unique for every authenticated message, its security is compromised if the nonce repeats even once. Therefore, from the usability point of view, one needs to ensure the nonce's uniqueness for every authenticated message, which is challenging in practical contexts. For example, it is difficult to maintain the uniqueness of the nonce while implementing the cipher in a stateless device or in cases where the nonce is chosen randomly from a small set. The nonce may also accidentally repeat due to a faulty implementation of the cipher or due to the fault that occurred by resetting of the nonce itself [13]. Therefore, the guard from the nonce repetition attack is much desired from a nonce-based MAC.

To get rid of the nonce repetition problem, an immediate solution is to encrypt the output of the Wegman Carter MAC, resulted in Encrypted Wegman-Carter Shoup, or in short EWCS [22] MAC. Even though EWCS guarantees security even when the nonce repeats, its security drops to the birthday bound even when the nonce is unique. To facilitate both the features, EWCDM [22], DWCDM [26], and 1K-DWCDM [28] have been proposed that gives beyond the birthday bound security when the nonce is unique and birthday bound security when the nonce repeats. However, the disadvantage of both constructions is that their security falls to the birthday bound if the nonce ever repeats.

To mitigate the above problem, Dutta et al. [31] have proposed a nonce-based variant of EHtM [39] MAC, called nEHtM MAC. Similar to EWCDM and DWCDM, nEHtM gives beyond the birthday bound (resp. birthday bound) security when the nonce is unique (resp. the nonce repeats). But, unlike these two constructions, the security of nEHtM degrades gracefully with the repetition of the nonce. In other words, one can get adequate security from nEHtM if the repetition of the nonce occurs in a controlled way, a feature which is not present in EWCDM or DWCDM. This phenomenon is formally known as *faulty nonce model.* This notion was introduced in [31], which informally says that a nonce is *faulty* if it appears in a previous signing query. [1] The authors of [31] have shown that nEHtM gives $2n/3$-bit security under faulty nonce model, which is recently improved to $3n/4$-bit by Choi et al. [21]. However, the tightness of its security bound is still open.

1.2 Public Permutation Based MAC

The underlying primitive of all the MACs as mentioned above is a block cipher that is evaluated only in the forward direction (except DWCDM and 1K-DWCDM, which require the inverse call to the underlying block cipher). However, as most block ciphers are designed to be efficient in both the forward and the inverse direction, they seem to be an over-engineered primitive for such purpose [20]. In contrast to the block cipher, cryptographic permutations are mainly designed with the motive to be fast in the forward direction, but not necessarily in the inverse direction, e.g., Keccak [7], Gimli [6], SPONGENT [14]. Moreover, as permutations do not employ any round key, evaluating them is faster than evaluating a keyed block cipher due to the complexity of the underlying key scheduling algorithm.

In this regard, it would be apt to quote the statement of Bertoni et al. from their paper [8] regarding the efficiency of permutations over block ciphers.

" . . . the inverse mapping of block ciphers imposes a separation of the processing of the $n + k$ bits of the input. The key is processed in a key schedule and the data in the data path, and there can be no diffusion from the data path to the key schedule, which strongly limits the potential diffusion . . . Such a restriction is not present in the design of cryptographic permutations as they do not make a distinction between the processing of key and data input as there is no specific key input."

In this line of work, Dutta and Nandi [30] have shown a public permutation-based beyond the birthday bound secure nonce based MAC, which they refer to as nEHtM_p MAC. It is a permutation-based variant of nEHtM MAC and the first instance of a nonce based beyond the birthday bound secure permutation-based MAC whose security bound degrades gracefully with the repetition of the nonce.

$$\mathsf{nEHtM}_p(\nu, m) = \pi(0\|\nu \oplus k) \oplus \pi(1\|\nu \oplus \mathsf{H}_{k_h}(m)). \tag{1}$$

Authors have shown that the construction is secure roughly up to $2^{2n/3}$ signing queries. Independent to this work, Chakraborti et al. have also proposed a few other public permutation-based beyond the birthday bound secure nonce based MACs, which they refer to as PDM MAC and its related single keyed construction PDM^* MAC and $\mathsf{1K\text{-}PDM}^*$ MAC. Both these constructions, i.e., nEHtM_p and PDM^* MAC require two invocations of the permutation. But, PDM^* MAC requires the invertibility of the permutation, whereas nEHtM_p is free from that constraint. We note here that nonce-based MACs using public

[1] It has been stated [31] that faulty nonce model is a weaker notion than multi-collision of nonces – a natural and a popular metric to measure the misuse of the nonce.

permutations are usually designed in sponge mode. But the drawback of such designs are twofold: (i) they do not use the full size of the permutation for guaranteeing security and (ii) most of them attain only the birthday bound security in the size of its capacity c, i.e., $c/2$ bit security (exceptions are Bettle [16], [24] whose security bound is roughly the size of its capacity). Although security bound of $2^{c/2}$ is usually good in practice when they are instantiated with large size permutations (e.g., Keccak [7] of state size is 1600 bits), but they are not suitable for use in a resource-constrained environment. In such a scenario, one often prefers to use lightweight permutations such as SPONGENT [14] of state size 88 bits and PHOTON [34] of state size 100 bits. However, in such circumstances, using them as underlying primitives in birthday bound secure sponge type constructions renders them inadequate security. Therefore, it is preferable to design a public permutation-based nonce-based MAC whose security depends on the full size of the underlying permutation and gives adequate security when instantiated with light-weight permutation. In this regard, nEHtM_p and PDM offers good practical security when instantiated with small size permutation, and their security bound exploits the entire state size of the underlying permutation.

1.3 Public Permutation Based PRF

Designing PRFs out of public permutations was initiated by Chen et al. in [20], where they proposed two fixed-input and fixed-output length beyond birthday bound secure PRFs based on public permutations - one is in the parallel mode and the other is in the sequential mode. (i) For the parallel mode, they have shown that the sum of two independent instances of Even-Mansour [33] cipher, which they refer to as $\mathsf{SoEM22}$,

$$\mathsf{SoEM22}_{k_1,k_2}^{\pi_1,\pi_2}(x) \overset{\Delta}{=} \pi_1(x \oplus k_1) \oplus \pi_2(x \oplus k_2) \oplus k_1 \oplus k_2,$$

provides a tight $2n/3$-bit security. This construction was later extended by Bhattacharjee et al. [9], where they showed the beyond birthday bound security of the domain separated variant of $\mathsf{SoEM22}$. They have also proved that one cannot reduce the number of keys of $\mathsf{SoEM22}$ without degrading the security bound to the birthday limit. (ii) For the sequential mode, Chen et al. proposed two constructions $\mathsf{SoKAC1}$ and $\mathsf{SoKAC21}$, where

$$\begin{aligned} \mathsf{SoKAC1}_{k_1,k_2}^{\pi}(x) &\overset{\Delta}{=} \pi(\pi(x \oplus k_1) \oplus k_2) \oplus \pi(x \oplus k_1) \oplus k_2 \oplus k_1 \\ \mathsf{SoKAC21}_{k}^{\pi_1,\pi_2}(x) &\overset{\Delta}{=} \pi_2(\pi_1(x \oplus k) \oplus k) \oplus \pi_1(x \oplus k) \oplus k. \end{aligned}$$

Chen et al. have shown an $n/2$ bit attack on $\mathsf{SoKAC1}$ construction whereas $\mathsf{SoKAC21}$ has been proven to have a tight $2n/3$-bit security. However, later in [17], Chakraborti et al. claimed that the attack on $\mathsf{SoKAC1}$ is possibly wrong and shown a $2n/3$-bit attack on it. They also conjectured that this attack bound is indeed tight. On the other hand, Nandi [43] exhibited a birthday bound attack on $\mathsf{SoKAC21}$ and hence falsifying the security claim of the construction. In [17], Chakraborti et al. have proposed PDM MAC, a beyond birthday bound secure single permutation based fixed input and fixed output length PRF that opearates in sequential mode. The design of PDM MAC is motivated from the *Decrypted Davis-Meyer* (DDM) construction,

$$\mathsf{DDM}_k(x) \overset{\Delta}{=} \pi^{-1}(\pi(x) \oplus x).$$

PDM MAC requires an n-bit key k and an n-bit public permutation π to generate the output as follows:

$$\mathsf{PDM}_k^{\pi}(x) \overset{\Delta}{=} \pi^{-1}(\pi(x \oplus k) \oplus (x \oplus 3k)) \oplus 2k.$$

Although, minimally structured, PDM MAC requires the invertiblity of the permutation P (similar to the design of DWCDM [26]), which somewhat brings down one of the advantages of using cryptographic permutations in a mode, i.e, the efficiency of evaluating the permutation in forward direction. Recently, Dutta et al. [32] proposed another beyond birthday bound secure PRF, called pEDM, built from of public permutations and it does not require the inverse call like PDM MAC.

1.4 Comparison with Keyed Sponge Construction

The sponge construction consists of a sequential application of a permutation to a state of n-bits, which is splitted up into an r-bit rate (called the outer part) and a c-bit capacity (called the inner part). In the absorption phase, message blocks of size r-bits are absorbed by the outer part and the state is transformed using the permutation, while in the squeezing phase, r-bit digests are extracted from the outer part at a time. While sponge mode has essentially been used for designing lightweight hash functions, its keyed variants have become very popular modes of operation to build message authentication codes. Although the state size of a keyed sponge-based design is only n-bits, allowing them to achieve a smaller hardware footprint, the security bound of almost all such designs falls within the birthday bound of the capacity part of the state size of the permutation (except Bettle [16]). Hence, the full state size of the permutation is not used in the security bound of the construction, which is one of the major shortcomings of using these modes with smaller state permutations. On the other hand, although the state size of our construction is $2n$-bits compared to the n-bit state size for keyed sponge based mode, our construction achieves a comparitively better security bound (i.e., $2n/3$-bits) over the security bound of keyed sponge based constructions which roughly falls within $c/2$-bits, where $c < n$ is the capacity part of the construction.

1.5 Single-User vs Multi-user Security

Until now, we have discussed the security models for block ciphers, and mode of operations in which adversaries are given access to some keyed oracles for a single unknown randomly sampled key. Such model is known as the *single-user security model*, i.e., when the adversary interacts with a specific machine in which the cryptographic algorithm is deployed and tries to compromise its security. However, in practice, cryptographic algorithms are usually deployed in more than one machine. For example, AES-GCM is now widely used in the TLS protocol to protect web traffic and is currently used by billions of users daily. Thus, it is natural to ask that

To what extent the number of users will affect the security bound of MAC constructions?

In other words, it is paramount to investigate the security of the cryptographic algorithms in the *multi-key setting*, where adversaries are successful if they compromise the security of one out of many users. That means the adversary's winning condition is a disjunction of single key winning conditions. The notion of multi-user (mu) security is introduced by Biham [10] in symmetric cryptanalysis and by Bellare, Boldyreva, and Micali [3] in the context of public-key encryption. In the multi-user setting, attackers have access to multiple machines such that a particular cryptographic algorithm F is deployed in each machine with independent secret keys. An attacker can adaptively distribute its queries across multiple machines with independent keys. Multi-user security considers such attackers that succeed in compromising the security of at least one machine, among others.

Multi-user security for block ciphers is different than the multi-user security for modes. In the single-key setting, the best attacks against block cipher such as AES do not improve with increased data complexity. However, in the multi-key environment, they do, as first observed Biham [10] and later refined as a time-memory-data trade-off by Biryukov et al. [11]. These results say that one can take advantage of the fact that recovering a block cipher key out of a large group of keys is much easier than targeting a specific key. The same observation can be applied to any deterministic symmetric-key algorithm, as done for MACs by Chatterjee et al.[19]. A more general result guarantees that the *multi-user advantage of an adversary for a cryptographic algorithm is at most μ times its single user advantage.* Therefore, for any cryptographic algorithm, multi-user security bound involving a factor μ is easily established using a hybrid argument that shows the upper bound of the adversarial success probability roughly μ times its single user security advantage. Bellare and Tackmann [5] first formalized a multi-user secure authenticated encryption scheme and also analyzed countermeasures against multi-key attacks in the context of TLS 1.3. However, they derived a security bound that also contains the μ factor. Such a bound implies a significant security drop of the construction when the number of users is large, and in fact, this is precisely the situation faced in large-scale deployments of AES-GCM such as TLS.

As evident from [2, 5, 15, 35, 36, 37, 42], it is a challenging problem to study the security degradation of cryptographic primitives with the number of users, even when its security is known in the single-user setting. The study of the multi-user security of MACs are somewhat scarce in the literature except for the work of Chatterjee et al. [19], and very recently the work of Andrew et al. [41], and Bellare et al. [2]. The first two consider a generic reduction for MACs in which the security of the primitive in the multi-user setting is derived by multiplying the number of users u with the single-user security. Recently, Shen et al. [48] have shown that the multi-user security of DbHtS [25] MAC is at most $2^{2n/3}$. This bound is non-trivial and improved over the usual birthday bound, which would have been obtained from the generic multi-user to single-user reduction [48]. Although the multi-user security is analyzed over a block cipher based MAC, there are no known results that analyze the security of permutation-based MAC in the multi-user setting.

1.6 Our Contribution

Given the practical importance of analyzing the security of a construction in multi-user setting, in this paper, for the first time we analyze the multi-user security of a permutation based MAC under faulty nonce model [2]. In particular, we propose a nonce based MAC out of public permutations called nEHtM^*_p

$$\mathsf{nEHtM}^*_p(\nu, m) = \pi(0\|\nu \oplus k) \oplus \pi(1\|\nu \oplus k \oplus \mathsf{H}_{k_h}(m)) \qquad (2)$$

and shown that it is secured roughly up to $2^{2n/3}$ queries in the multi-user setting. Moreover, the bound degrades gracefully under the faulty nonce model. Following the definition of Dutta et al. [31], a signing query is said to be a *faulty query* in **single-user setting** if there exists a previous MAC query such that their corresponding nonces match. The nonce in a faulty query is called a *faulty nonce*. In the multi-user set up, we consider the total number of faulty queries over all users. We show the unforgeability of this construction through an extended distinguishing game and apply the expectation method to bound its distinguishing advantage. Moreover, we also instantiate the underlying hash function of the construction with Polyhash [40] function to realize a concrete instance of a permutation-based MAC nEtHtM^+_p, whose security bound degrades gracefully under the faulty nonce model.

1.6.1 Why nEHtM^*_p over nEHtM_p

At AFRICACRYPT'20, Dutta and Nandi [30] have shown that their proposed construction nEHtM_p achieves beyond birthday bound security under the faulty nonce model in the

[2] We would like to emphasize that repetition of nonce between two different users will not be considered as a faulty nonce

single-user setting. However, we have identified that their security analysis is not complete. The authors of [30] had used the tool of "*Expectation Method*" [35] to derive the security bound of their construction. As a part of the derivation process, they identified and bounded some bad events, followed by showing that the probability of realizing a transcript that does not satisfy the bad events is almost same in both the real and the ideal world.

We found that the authors did not consider all possible bad events. In particular, we found that some bad events related to the MAC and the verification queries are not present in the existing set of bad events of nEHtM_p. As a result, the analysis for the good transcript of the construction is also not complete. We would like to mention here that the analysis for the good transcript is highly dependent on the result of "*Extended Mirror Theory*" [31], where the authors estimated a lower bound on the number of solutions for a given set of bivariate affine equations and non-equations over the set $\{0,1\}^n$. Due to the requirement of the additional bad events to complete the proof, we also found out that the extended mirror theory used in [30] only contains bivariate affine equations and non-equations, which is not sufficient for the good transcript analysis of nEHtM_p. We also need to consider univariate affine non-equations, which were not considered in [30].

When we collected all the missing pieces, we started to fix the proof of nEHtM_p, we got stuck to bound one of the additional bad events for nEHtM_p, which is mentioned below:

$$\nu_i \oplus \mathsf{H}_{k_h}(M_i) = x, \nu_j \oplus \mathsf{H}_{k_h}(M_j) = x', t_i \oplus y = t_j \oplus y',$$

where $(\nu_i, t_i), (\nu_j, t_j)$ are the nonces and tags respectively for i-th and j-th query. Note that when $M_i \neq M_j$, then their hash value will also likely be distinct and in that case, the left hand side of the first two equations of the above event are not supposed to be identical. Hence the difficulty comes in bounding this event. To mitigate this problem and simplify the analysis, we make a small modification to the design of nEHtM_p, including the key k as an input to the second permutation call. This small change allows us to treat the modified hash function $\mathsf{H}'_{k_h}(M) \overset{\Delta}{=} \mathsf{H}_{k_h}(M) \oplus k$ as a pairwise independent hash function, which in turn enables us to bound the above bad event easily. Using this modified hash function, we not only provide a complete security proof of the construction nEHtM^*_p, but also extend its proof in the multi-user setting. The difference between nEHtM_p and nEHtM^*_p is depicted in Fig. 1.1.

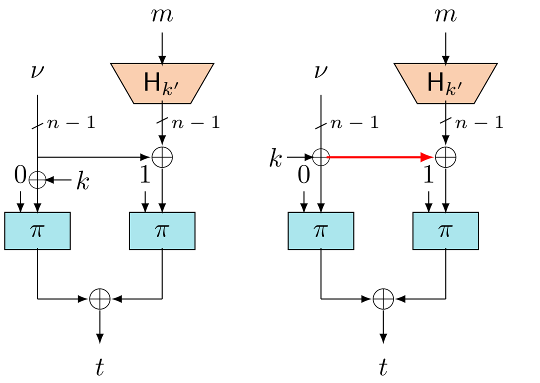

Figure 1.1: Left side of the figure is nEHtM_p MAC and the right side of the figure is nEHtM^*_p MAC. Difference between the two constructions is depicted by red colored edge.

We would like to note here that although it seems nEHtM^*_p is just an incremental update of nEHtM_P, we currently do not know how to bound the above event without having the troubleshooting that we adopted here. In fact, it is an open problem to see that whether the proof of nEHtM_p can be completed or not. In the following table (tab. 1), we compare the several public permutation-based PRFs and MACs in terms of various parameters.

Table 1: Comparison table for $2n/3$ bit secure permutation-based PRFs and MACs. n denotes the state size of the permutation and the output size of PRFs or MACs. Inv denotes whether the construction requires an inverse call of the permutation. mu denotes whether the construction is secure in the multi-user setting. $\mathsf{graceful}$ denotes whether the security of the construction degrades gracefully with repetition of the nonce. By having a 'x' in a column, we mean that the construction has not been proven secure under the column description model so far. However, it does not immediately imply any attack on it under the corresponding setting. The last four constructions require a keyed hash function with at most ℓ blocks input. The number of keys for those constructions includes the hash keys as well.

Constructions	(perm, keys)	mu	graceful	Inv	i/p size
SoEM22 [20]	$(2,2)$	x	x	x	n
SoKAC1 [20]	$(1,2)$	x	x	x	n
PDMMAC [17]	$(1,1)$	x	x	✓	n
DS-SoEM [9]	$(1,2)$	x	x	x	$n-1$
pEDM [32]	$(1,2)$	x	x	x	n
nEHtM_p [30]	$(1,2)$	x	✓	x	$n-1+\ell n$
$\mathsf{PDM}^*\mathsf{MAC}$ [17]	$(1,2)$	x	x	✓	$n+\ell n$
$\mathsf{1K\text{-}PDM}^*\mathsf{MAC}$ [17]	$(1,1)$	x	x	✓	$n+\ell n$
nEHtM_p^* [This Paper]	$(1,2)$	✓	✓	x	$n-1+\ell n$

Remark 1. We would like to mention here that in the table above we compared our construction with various permutation based MACs and PRFs without considering keyed sponge designs. The objective of this paper is to have a theoretical study on the beyond birthday bound security of permutation based MACs that exploits the full state of the permutation. While we agree that for a concrete value of the state size of the permutation (n), $2n/3$ bit security is achievable from a keyed sponge construction at the cost of a slightly larger size permutation and a smaller state size. However, achieving beyond birthday bound security by exploiting the full state size of the permutation always comes at the cost of extra state size. A detail comparison of our construction with keyed sponges in terms of implementation is out of the scope of this paper.

Organization. In Sect. 2 we set up the background. In Sect. 3, we rigorously analyze the extended mirror theory with univariate affine non-equations for general $\xi_{\max}$. Sect. 4 recalls the construction nEHtM_p^* and state its multi-user MAC security bound. We have presented a complete and correct security proof of the construction in the multi-user setting under the faulty nonce model in Sect. 5. We recover the single-user security bound of the construction trivially from its multi-user bound by setting the value of $u = 1$.

2 Preliminaries

General Notations: For $n \in \mathbb{N}$, we denote the set of all binary strings of length n and the set of all binary strings of finite arbitrary length by $\{0,1\}^n$ and $\{0,1\}^*$ respectively. $\{0,1\}^+$ denotes the set of all non-empty binary strings of finite arbitrary length. We often refer the elements of $\{0,1\}^n$ as *block.* For an n-bit binary string $x = (x_{n-1} \dots x_0)$, $\mathsf{msb}(x)$ denotes the first bit of x in left to right ordering, i.e. $\mathsf{msb}(x) = x_{n-1}$. Moreover, $\mathsf{chop}_{\mathrm{msb}}(x) \overset{\Delta}{=} (x_{n-2} \dots, x_0)$, i.e., $\mathsf{chop}_{\mathrm{msb}}(x)$ returns the string x by dropping just its msb.

For any element $x \in \{0,1\}^*$, $|x|$ denotes the number of bits in x and for $x, y \in \{0,1\}^*$, $x \| y$ denotes the concatenation of x followed by y. We denote the bitwise xor operation of $x, y \in \{0,1\}^n$ by $x \oplus y$. We parse $x \in \{0,1\}^+$ as $x = x_1 \| x_2 \| \ldots \| x_l$ where for each $i = 1, \ldots, l-1$, x_i is a block and $1 \leq |x_l| \leq n$. For a sequence of elements $(x^1, x^2, \ldots, x^s) \in \{0,1\}^*$, x^i_a denotes the a-th block of i-th element x^i. For a value s, we denote by $t \leftarrow s$ the assignment of s to variable t. For any natural number $j \in \mathbb{N}$, $\langle j \rangle_s$ denotes the s bit binary representation of integer j. For $i \in \{0,1\}^n$, $\mathsf{left}_k(i)$ represents the leftmost k bits of i. Similarly, $\mathsf{right}_k(i)$ represents the rightmost k bits of i. For any finite set $\mathcal{X}$, $X \leftarrow_{\$} \mathcal{X}$ denotes that X is sampled uniformly at random from $\mathcal{X}$ and $X_1, \ldots, X_s \leftarrow_{\$} \mathcal{X}$ denotes that X_i's are sampled uniformly and independently from $\mathcal{X}$. $\mathbb{F}_{\mathcal{X}}(n)$ denotes the set of all functions from $\mathcal{X}$ to $\{0,1\}^n$. We often write $\mathbb{F}(n)$ when the domain is clear from the context. We denote the set of all permutations over $\{0,1\}^n$ by $\mathbb{P}(n)$. For integers $1 \leq b \leq a$, $(a)_b$ denotes the product $a(a-1)\ldots(a-b+1)$, where $(a)_0 = 1$ by convention and for $q \in \mathbb{N}$, $[q]$ refers to the set $\{1, \ldots, q\}$.

2.1 Public Permutation Based MAC in Multi-User Setting

Let $n, d \in \mathbb{N}$, let $\mathsf{F} : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \to \mathcal{T}$ be a keyed function based on d permutations that are independent and uniformly sampled from $\mathbb{P}(n)$. Here $\mathcal{K}, \mathcal{N}, \mathcal{M}$ and $\mathcal{T}$ are respectively the key space, the nonce space, the message space, and the tag space. For simplicity, we write $\mathsf{F}^{\boldsymbol{\pi}}_k$ to denote F with uniform k and uniform $\boldsymbol{\pi}$. Based on $\mathsf{F}^{\boldsymbol{\pi}}_k$, we define the nonce-based message authentication code $\mathcal{I} = (\mathcal{I}.\mathsf{KGen}, \mathcal{I}.\mathsf{Sign}, \mathcal{I}.\mathsf{Ver})$ build from public permutations. We define $\mathcal{I}$ as follows: for $k \in \mathcal{K}$, the signing algorithm $\mathcal{I}.\mathsf{Sign}_k$ takes as input $(\nu, m) \in \mathcal{N} \times \mathcal{M}$ and outputs $t \leftarrow \mathsf{F}^{\boldsymbol{\pi}}_k(\nu, m)$, and the verification algorithm $\mathcal{I}.\mathsf{Ver}_k$, takes as input $(\nu, m, t) \in \mathcal{N} \times \mathcal{M} \times \mathcal{T}$ and outputs 1 if $\mathsf{F}^{\boldsymbol{\pi}}_k(\nu, m) = t$ and 0 otherwise.

In the multi-user setting we assume there are μ users, such that the i-th user executes $\mathsf{F}^{\boldsymbol{\pi}}_{k_i}$. Moreover, the i-th user key k_i is independent of the keys of all other users. An adversary A has access to all the μ users as oracles. A makes signing queries of the form (i, ν, m) to the i-th user and obtains $t \leftarrow \mathsf{F}^{\boldsymbol{\pi}}_{k_i}(\nu, m)$. Similarly, A makes verification queries of the form (i, ν, m, t) to the i-th user and obtains either $\top$ (accepts) or $\bot$ (rejects). A signing query to the i-th user (i, ν, m) by an adversary A is called a **faulty query** if A has already queried to the signing algorithm of the i-th user with the same nonce but with a different message. Note that, (i, ν, m) *is not considered to be a faulty query*, if the nonce ν collides with the nonce of the j-th user for some $j \neq i$. A can also make queries to the tuple of underlying permutations $\boldsymbol{\pi}$ and their inverses $\boldsymbol{\pi^{-1}} = (\pi_1^{-1}, \ldots, \pi_d^{-1})$. Let A be a $(\eta, q_m, q_v, p, \mathsf{t})$-adversary against the unforgeability of $\mathcal{I}$ with oracle access to the signing algorithm $\mathcal{I}.\mathsf{Sign}_{k_i}$ and the verification algorithm $\mathcal{I}.\mathsf{Ver}_{k_i}$ for all μ users, and A also gets access to the d-tuple of permutations $\boldsymbol{\pi}$. A makes in total η faulty signing queries out of q_m signing queries distributed over its μ signing oracles, q_v verification queries distributed over its μ verification oracles, and p primitive queries with running time of A at most t. A is said to be *nonce respecting* (resp. nonce misuse) if $\eta = 0$ (resp. if $\eta \geq 1$). However, A may repeats nonces in its verification queries. Moreover, the primitive queries are interleaved with the signing and the verification queries. We assume that for any $i \in [\mu]$ A does not make a verification query (i, ν, m, t) such that t is obtained through any of the previous signing queries to the i-th user. A is said to *forge* $\mathcal{I}$ if for any $i \in [\mu]$, and for any of its verification queries (not obtained through a previous signing query), the verification algorithm returns 1. The forging advantage of A against the unforgeability of the nonce-based MAC $\mathcal{I}$ in the multi-user seting is defined as

$$\mathbf{Adv}^{\text{mu-nMAC}}_{\mathcal{I}}(\mathsf{A}) \triangleq \Pr\left[\mathsf{A}^{(\mathcal{I}.\mathsf{Sign}_{k_1}, \mathcal{I}.\mathsf{Ver}_{k_1}), \ldots, (\mathcal{I}.\mathsf{Sign}_{k_\mu}, \mathcal{I}.\mathsf{Ver}_{k_\mu}), \boldsymbol{\pi}, \boldsymbol{\pi^{-1}}} \text{ forges}\right],$$

where the randomness is defined over $k_1, \ldots, k_\mu \leftarrow_{\$} \mathcal{K}$, $\pi_1, \ldots, \pi_d \leftarrow_{\$} \mathbb{P}(n)$ and the randomness of the adversary (if any). We write

$$\mathbf{Adv}_{\mathcal{I}}^{\text{mu-nMAC}}(\eta, q_m, q_v, p, \mathsf{t}) \overset{\Delta}{=} \max_{\mathsf{A}} \mathbf{Adv}_{\mathcal{I}}^{\text{mu-nMAC}}(\mathsf{A}),$$

where the maximum is taken over all $(\eta, q_m, q_v, p, \mathsf{t})$-adversaries A. In this paper, we skip the time parameter of the adversary as we will assume throughout the paper that the adversary is computationally unbounded. This will render us to assume that the adversary is deterministic. When $\mu = 1$, then it renders to the single user forging advantage.

UPPER BOUND ON $\mathbf{Adv}_{\mathcal{I}}^{\text{mu-nMAC}}(\mathsf{A})$ ([29]). To obtain an upper bound for the forging advantage of $\mathcal{I}$ in the multi-user setting with respect to the adversary A, we consider an another adversary B that interacts either with μ pairs of oracles $((\mathcal{I}.\mathsf{Sign}_{k_1}, \mathcal{I}.\mathsf{Ver}_{k_1}), \ldots, (\mathcal{I}.\mathsf{Sign}_{k_\mu}, \mathcal{I}.\mathsf{Ver}_{k_\mu}))$ or μ pair of oracles $(\mathsf{RF}_1, \mathsf{Rej}_1), \ldots, (\mathsf{RF}_\mu, \mathsf{Rej}_\mu)$ such that on a signing query of the form (i, ν, m), RF_i samples the tag t independently and uniformly at random from $\{0,1\}^n$ for every nonce message pair (ν, m) queried to the i-th oracle. Moreover, for all $i \in [\mu]$, Rej_i oracle always returns $\perp$ for any (i, ν, m, t). Then, $\mathbf{Adv}_{\mathcal{I}}^{\text{mu-nMAC}}(\mathsf{A})$ is upper bounded by

$$\max_{\mathsf{B}} \left| \Pr\left[\mathsf{B}^{\boldsymbol{\mathcal{O}}_{\mathbf{re}}, \boldsymbol{\pi}, \boldsymbol{\pi}^{-1}} \Rightarrow 1\right] - \Pr\left[\mathsf{B}^{\boldsymbol{\mathcal{O}}_{\mathbf{id}}, \boldsymbol{\pi}, \boldsymbol{\pi}^{-1}} \Rightarrow 1\right] \right|, \tag{3}$$

where $\boldsymbol{\mathcal{O}}_{\mathbf{re}}$ denotes the oracle $((\mathcal{I}.\mathsf{Sign}_{k_1}, \mathcal{I}.\mathsf{Ver}_{k_1}), \ldots, (\mathcal{I}.\mathsf{Sign}_{k_\mu}, \mathcal{I}.\mathsf{Ver}_{k_\mu}))$, and $\boldsymbol{\mathcal{O}}_{\mathbf{id}}$ denotes $((\mathsf{RF}_1, \mathsf{Rej}_1), \ldots, (\mathsf{RF}_\mu, \mathsf{Rej}_\mu))$. Moreover, $\mathsf{B}^{\boldsymbol{\mathcal{O}}} \Rightarrow 1$ denotes that adversary B outputs 1 after interacting with its oracle $\boldsymbol{\mathcal{O}}$. When $\mu = 1$, then Eqn. (3) represents the upper bound of the forging advantage of the MAC in the single user setting.

2.2 Almost Xor Universal and Almost Regular Hash Function

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

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

Moreover, H is said to be an ϵ_{reg}-almost regular (ar) hash function, if for any $x \in \mathcal{X}$ and for any $\Delta \in \{0,1\}^n$,

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

2.3 Pairwise Independent Hash Function

Let $\mathcal{K}_h$ and $\mathcal{X}$ be two non-empty finite sets and H be a keyed function $\mathsf{H} : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$. Then, H is said to be an δ-pairwise independent hash function, if for any distinct $x, x' \in \mathcal{X}$ and for any $y, y' \in \{0,1\}^n$,

$$\Pr\left[K_h \leftarrow\!\!\$\, \mathcal{K}_h : \mathsf{H}_{K_h}(x) = y, \mathsf{H}_{K_h}(x') = y'\right] \leq \delta.$$

It is easy to see that, if H is an ϵ_{axu}-almost-xor universal hash function, then the function $\mathsf{H}'_{(k_h,k)} \overset{\Delta}{=} \mathsf{H}_{k_h} \oplus k$, where $k \in \{0,1\}^n$ is indepedently sampled over k_h, is $\epsilon_{\text{axu}}/2^n$-pairwise independent hash function. This is because for any $x \neq x'$ and for any $y, y' \in \{0,1\}^n$,

$$\begin{aligned}
& \Pr\left[K_h \leftarrow\!\!\$\, \mathcal{K}_h, K \leftarrow\!\!\$\, \{0,1\}^n : \mathsf{H}'_{(K_h,K)}(x) = y, \mathsf{H}'_{(K_h,K)}(x') = y'\right] \\
= \; & \Pr\left[K_h \leftarrow\!\!\$\, \mathcal{K}_h, K \leftarrow\!\!\$\, \{0,1\}^n : \mathsf{H}_{K_h}(x) \oplus K = y, \mathsf{H}_{K_h}(x') \oplus K = y'\right] \\
= \; & \Pr\left[K_h \leftarrow\!\!\$\, \mathcal{K}_h, K \leftarrow\!\!\$\, \{0,1\}^n : \mathsf{H}_{K_h}(x) \oplus K = y, \mathsf{H}_{K_h}(x) \oplus \mathsf{H}_{K_h}(x') = y \oplus y'\right] \\
\leq \; & \epsilon_{\text{axu}}/2^n.
\end{aligned}$$

2.4 Sum-Capture Lemma

We use the sum capture lemma due to Babai [1] and Steinberger [46]. Informally, the result states that for a random subset $\mathcal{S}$ of $\{0,1\}^n$ of size q and for any two arbitrary subsets $\mathcal{A}$ and $\mathcal{B}$ of $\{0,1\}^n$, the size of the set $\{(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} : s = a \oplus b\}$ is at most $q|\mathcal{A}||\mathcal{B}|/2^n$, except with negligible probabilty. In our setting, $\mathcal{S}$ is the set of tag values t_i, which are sampled with replacement from $\{0,1\}^n$. In this paper, we appeal to the result of sum-capture theorem by Cogliati and Seurin [23].

Lemma 1 (**Sum-Capture Lemma**). *Let $n, q \in \mathbb{N}$. Let $\mathcal{S} = \{t_1, \ldots, t_q\} \subseteq \{0,1\}^n$ such that t_i's are with replacement sample of $\{0,1\}^n$. Then, for any two subsets $\mathcal{A}$ and $\mathcal{B}$ of $\{0,1\}^n$, we have*

$$\Pr[|\{(t,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} : t = a \oplus b\}| \geq q|\mathcal{A}||\mathcal{B}|/2^n + \sqrt{3nq|\mathcal{A}||\mathcal{B}|}] \leq \frac{2}{2^n}, \tag{4}$$

where the randomness is defined over the set $\mathcal{S}$.

3 Solving a System of Affine (Non)-Equations

We prove the MAC security of nEHtM_p^* using the Expectation Method, where one is required to lower bound the probability of realizing a good transcript in the real and the ideal world. In order to compute this probability in the real world, we require to count the number of permutations π such that

$$\begin{cases} \pi(0\|\nu_1 \oplus k) \oplus \pi(1\|\nu_1 \oplus \mathsf{H}_1 \oplus k) = t_1 \\ \pi(0\|\nu_2 \oplus k) \oplus \pi(1\|\nu_2 \oplus \mathsf{H}_2 \oplus k) = t_2 \\ \qquad \vdots \\ \pi(0\|\nu_{q_m} \oplus k) \oplus \pi(1\|\nu_{q_m} \oplus \mathsf{H}_{q_m} \oplus k) = t_{q_m} \end{cases} \qquad \begin{cases} \pi(0\|\nu'_1 \oplus k) \oplus \pi(1\|\nu'_1 \oplus \mathsf{H}' \oplus k) \neq t'_1 \\ \pi(0\|\nu'_2 \oplus k) \oplus \pi(1\|\nu'_2 \oplus \mathsf{H}'_2 \oplus k) \neq t'_2 \\ \qquad \vdots \\ \pi(0\|\nu'_{q_v} \oplus k) \oplus \pi(1\|\nu'_{q_v} \oplus \mathsf{H}'_{q_v} \oplus k) \neq t'_{q_v}. \end{cases}$$

holds. Therefore, it boils down to count the number of solutions to the above system of equations and non-equations. This result is captured by the result of *Extended Mirror Theory* [31].

Consider an undirected edge-labelled graph $\mathcal{G} = (\mathcal{V} := \{P_1, \ldots, P_\beta\}, \mathcal{E} \sqcup \mathcal{E}', \mathcal{L})$ with edge labelling function $\mathcal{L} : \mathcal{E} \sqcup \mathcal{E}' \to \{0,1\}^n$, where the edge set is partitioned into two disjoint sets $\mathcal{E}$ and $\mathcal{E}'$. For an edge $\{Y_i, Y_j\} \in \mathcal{E}$, we write $\mathcal{L}(\{P_i, P_j\}) = \lambda_{ij}$ (and so $\lambda_{ij} = \lambda_{ji}$) and $\mathcal{L}(\{P_i, P_j\}) = \lambda'_{ij}$ for all $\{P_i, P_j\} \in \mathcal{E}'$. We call the edges of $\mathcal{E}$ as *equation edges* and the edges of $\mathcal{E}'$ as *non-equation edges*. Let $\mathcal{G}^= := (\mathcal{V}^=, \mathcal{E}, \mathcal{L}_{|\mathcal{E}})$ denotes the subgraph of $\mathcal{G}$, where $\mathcal{V}^=$ is the set of vertices of $\mathcal{V}$ such that they are incident on at least one edge of $\mathcal{E}$ and $\mathcal{L}_{|\mathcal{E}}$ is the function $\mathcal{L}$ restricted over the set $\mathcal{E}$. We say $\mathcal{G}$ is **good** if it satisfies the following three conditions:

1. $\mathcal{G}^=$ must be acylic.
2. $\mathcal{L}(P_{st}) \neq \mathbf{0}$ for all paths P_{st} in graph $\mathcal{G}^=$, where $\mathcal{L}(P_{st}) := \sum_{e \in P_{st}} \mathcal{L}(e) = P_s \oplus P_t$ and P_{st} is a path of $\mathcal{G}^=$ between vertices s and t.
3. If we consider a cycle C of length at least 2 in $\mathcal{G}$ such that the edge set of C contains exactly one non-equation edge $e' \in \mathcal{E}'$ and the remaining edges are the equation edges $e \in \mathcal{E}$, then $\mathcal{L}(C) \neq \mathbf{0}$, where $\mathcal{L}(C) := \sum_{e \in C} \mathcal{L}(e)$.

For such a good graph $\mathcal{G}$, we denote the set of components of $\mathcal{G}^=$ as $\mathsf{comp}(\mathcal{G}^=) = (\mathsf{C}_1, \ldots, \mathsf{C}_k)$, μ_i denotes the size of (i.e. the number of vertices in) the i-th component C_i and $\mu_{\max} = \max\{\mu_1, \ldots, \mu_k\}$ is the size of the largest component of $\mathcal{G}^=$. ρ_i the total number of vertices upto the i-th component with the convention that $\rho_0 = 0$. We denote the set of vertices of i-th component of $\mathcal{G}^=$ as $\mathcal{V}_i$. Now, we consider three types of non-equation edges in $\mathcal{E}'$: (a) first type of non-equation edges are those whose both end points belong to vertices of different components of $\mathcal{G}^=$. We color these edges with blue and these connect two vertices of different components of $\mathcal{G}^=$. (b) The second type of edges are those whose exactly one end point belongs to the set of vertices of $\mathcal{G}^=$. We color these edges with orange. (iii) The last type of edges are those whose both end points do not belong to the set of vertices of $\mathcal{G}^=$. We color these edges with red. Let $\mathcal{V}^{\neq} \subseteq \mathcal{V} \setminus \mathcal{V}^=$ be set of the isolated end points of red-colored and orange-colored edges. Let q_b be the number of blue colored edges, q_o be the number of orange colored edges and q_r be the number of red colored edges. We pictorially depict such graphs in Fig. 3.1.

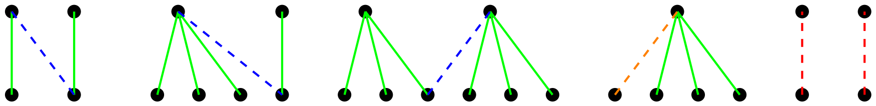

Figure 3.1: Dashed edges denotes the verification non-equation with three types (a) blue edges, (b) orange edges and (c) red edges.

Having such a good graph $\mathcal{G}$, the induced system of equations and non-equations is defined as:

$$\mathbb{E}_{\mathcal{G}} = \begin{cases} P_i \oplus P_j & = \lambda_{ij} \ \forall \ \{P_i, P_j\} \in \mathcal{E}, \\ P_i \oplus P_j & \neq \lambda'_{ij} \ \forall \ \{P_i, P_j\} \in \mathcal{E}', \end{cases}$$

Now, we define the *injective solution* of a system of bivariate affine equations and non-equations as follows:

Definition 1 (Injective Solution). With respect to the system of equations and non-equations $\mathbb{E}_{\mathcal{G}}$ (as defined above), an injective function $\Phi : \mathcal{V} \to \{0,1\}^n$ is said to be an *injective solution* if $\Phi(P_i) \oplus \Phi(P_j) = \lambda_{ij}$ for all $\{P_i, P_j\} \in \mathcal{E}$ and $\Phi(P_i) \oplus \Phi(P_j) \neq \lambda'_{ij}$ for all $\{P_i, P_j\} \in \mathcal{E}'$.

Theorem 1. *Let $\mathcal{U} = \{u_1, \ldots, u_\sigma\}$ be a non-empty finite subset of $\{0,1\}^n$ for some $\sigma \geq 0$ and let $\mathsf{R} = (r_1, \ldots, r_{q_v^*})$ be an ordered tuple of n-bit strings. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E} \sqcup \mathcal{E}', \mathcal{L})$ be a good graph with $|\mathcal{V}| = \beta, |\mathcal{E}| = q_m$ and $|\mathcal{E}'| = q_v$ with q_b blue edges, q_o orange edges and q_r red edges such that $q_v = q_v^* + q_b + q_o + q_r$. Let the subgraph $\mathcal{G}^=$ has k components, μ_i be the size of the i-th component and $\rho_i = (\mu_1 + \cdots + \mu_i)$. Moreover, we assume that*

1. *for all $i \in [k]$, with $|\mathcal{V}_i| = \alpha_i \geq 0$, $v_j \in \mathcal{V}_i$ cannot be assigned with a value $r_{\alpha_1 + \ldots + \alpha_{i-1} + j}$ for $j \in [\alpha_i]$ with $\alpha_0 = 0$.*
2. *there exists $\mathcal{V}^* \subseteq \mathcal{V}^{\neq}$, with $|\mathcal{V}^*| = \alpha^{\neq} := (q_v^* - \alpha_1 - \ldots - \alpha_k)$ such that $v_j \in \mathcal{V}^*$ cannot be assigned a value $r_{\alpha_1 + \ldots + \alpha_k + j}$ for $j \in [\alpha^{\neq}]$.*

Then the total number of injective solutions, chosen from a set $\mathcal{Z} = \{0,1\}^n \setminus \mathcal{U}$ of size $2^n - \sigma$, for the induced system of equations and non-equations $\mathbb{E}_{\mathcal{G}}$ is at least:

$$\frac{(2^n - \sigma)_\alpha}{2^{nq_m}} \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2 \binom{\mu_i}{2}}{2^{2n}} - \frac{2q_v}{2^n} \right), \tag{5}$$

provided $\rho'_k \mu_{\max} \leq 2^n/4$ where $\rho'_i = \rho_i + \sigma$.

Proof. We proceed the proof by counting the number of solutions in each of the k components. Let $\tilde{\mu}_{ij}$ denotes the number of blue edges between i-th and j-th component of $\mathcal{G}^=$ and μ'_i to be the number of edges incident on $v_i \in \mathcal{V} \setminus \mathcal{G}^=(\mathcal{V})$. Therefore,

$$\sum_{v_i \in \mathcal{V} \setminus \mathcal{G}^=(\mathcal{V})} \mu'_i = q_o + q_r.$$

For the first component, the number of solutions is at least exactly $(2^n - \mu_1\sigma - \alpha_1)$. We fix such a solution and count the number of solutions for the second component. which is $(2^n - \mu_1\mu_2 - \tilde{\mu}_{1,2} - \mu_2\sigma - \alpha_2)$. This is because, let $Y_{i_{\mu_1+1}}$ be an arbitrary vertex of the second component and let $y_{i_{\mu_1+1}}$ be a solution of it. This solution is valid if the following conditions hold:

- $y_{i_{\mu_1+1}} \notin \mathcal{U}$.
- $y_{i_{\mu_1+1}}$ does not take μ_1 values $(y_{i_1}, \ldots, y_{i_{\mu_1}})$ from the first component.
- It must discard $\mu_1(\mu_2 - 1)$ values $(y_{i_1} \oplus \mathcal{L}(P_j), \ldots, y_{i_{\mu_1}} \oplus \mathcal{L}(P_j))$ for all possible paths P_j from a fixed vertex to any other vertex in the second component.
- It must discard $p(\mu_2 - 1)$ values as $(y_{i_{\mu_1+1}} \oplus \mathcal{L}(P_j)) \notin \mathcal{Y}$ for all possible paths P_j from $Y_{i_{\mu_1+1}}$ to any other vertices in the second component.
- $y_{i_{\mu_1+1}}$ does not take $\tilde{\mu}_{12}$ values to compensate for the fact that the set of values is no longer a group.
- $y_{i_{\mu_1+1}}$ does not take α_2 values.

Summing up all the conditions, the number of solutions for the second component is at least $(2^n - \mu_1\mu_2 - \mu_2\sigma - \tilde{\mu}_{12} - \alpha_2)$. In general, the total number of solutions for the i-th component is at least

$$\prod_{i=1}^{k} \left(2^n - \rho_{i-1}\mu_i - \mu_i\sigma - \sum_{j=1}^{i-1} \tilde{\mu}_{ij} - \alpha_i\right).$$

Suppose $|\mathcal{V}^{\neq}| = k'$, where recall that $\mathcal{V}^{\neq}$ is the set of end points of red colored edges. Fix such a vertex Y_{ρ_k+i} and let us assume that μ'_{ρ_k+i} red dashed edges are incident on it. If y_{ρ_k+i} is a valid solution to the variable Y_{ρ_k+i}, then we must have (a) y_{ρ_k+i} should be distinct from the previous ρ_k assigned values, (b) y_{ρ_k+i} should be distinct from the $(i-1)$ values assigned to the variables that do not belong to the set of vertices of the subgraph $G^{=}(\mathcal{V})$, (c) y_{ρ_k+i} should be distinct from the values of $\mathcal{U}$, (d) y_{ρ_k+i} should not take those μ'_{ρ_k+i} values and (e) y_{ρ_k+i} should not take $\alpha_i^{\neq}$ values, where

$$\sum_{i\in[k']} \alpha_i^{\neq} = \alpha^{\neq}.$$

Therefore, the total number of solutions is at least

$$h_\alpha \geq \prod_{i=1}^{k} \left(2^n - \rho_{i-1}\mu_i - \mu_i\sigma - \sum_{j=1}^{i-1} \tilde{\mu}_{ij} - \alpha_i\right) \cdot \prod_{i\in[k']} (2^n - \rho_k - \sigma - i + 1 - \mu'_{\rho_k+i} - \alpha^{\neq}). \quad (6)$$

Let $\chi_i \overset{\Delta}{=} (\tilde{\mu}_{i1} + \ldots + \tilde{\mu}_{i,i-1}), q''_v \overset{\Delta}{=} (\mu'_{\rho_k+1} + \ldots + \mu'_{\rho_k+k'})$ and $\rho'_i = \rho_i + \sigma$. After a simple algebraic calculation on Eqn. (6), we obtain

$$h_\beta \frac{2^{nq_m}}{(2^n - \sigma)_\beta} \geq \underbrace{\prod_{i=1}^{k} \frac{(2^n - \rho'_{i-1}\mu_i - \chi_i - \alpha_i)2^{n(\mu_i-1)}}{(2^n - \rho'_{i-1})_{\mu_i}}}_{\text{D.1}} \underbrace{\prod_{i=1}^{k'} \frac{(2^n - \rho'_k - i + 1 - \mu'_{\rho_k+i} - \alpha^{\neq})}{(2^n - \rho'_k - i + 1)}}_{\text{D.2}}.$$

By expanding $(2^n - \rho'_{i-1})_{\mu_i}$ we have $(2^n - \rho'_{i-1})_{\mu_i} \leq 2^{n\mu_i} - 2^{n(\mu_i-1)}\left(\rho'_{i-1}\mu_i + \binom{\mu_i}{2}\right) + 2^{n(\mu_i-2)}A_i$, where $A_i = \left(\binom{\mu_i}{2}(\rho'_{i-1})^2 + \binom{\mu_i}{2}(\mu_i - 1)\rho'_{i-1} + \binom{\mu_i}{2}\frac{(\mu_i-2)(3\mu_i-1)}{12}\right)$.

Bounding D.1. With a simplification on the expression of $\mathsf{D.1}$, we have

$$\begin{aligned}\mathsf{D.1} \quad &\geq \quad \prod_{i=1}^{k}\left(1-\frac{A_i}{2^{2n}-2^n(\rho'_{i-1}\mu_i+\binom{\mu_i}{2})+A_i}-\frac{2^n(\chi_i+\alpha_i)}{2^{2n}-2^n(\rho'_{i-1}\mu_i+\binom{\mu_i}{2})+A_i}\right)\\ &\overset{(4)}{\geq} \quad \prod_{i=1}^{k}\left(1-\frac{2A_i}{2^{2n}}-\frac{2\chi_i}{2^n}-\frac{2\alpha_i}{2^n}\right) \overset{(5)}{\geq} \left(1-\sum_{i=1}^{k}\frac{6(\rho'_{i-1})^2\binom{\mu_i}{2}}{2^{2n}}-\frac{2q_b}{2^n}-\sum_{i=1}^{k}\frac{2\alpha_i}{2^n}\right),\end{aligned}$$

where (4) follows from the fact that $2^n(\rho'_{i-1}\mu_i+\binom{\mu_i}{2})-A_i \leq 2^{2n}/2$, which holds true when $\rho'_k\mu_{\max} \leq 2^n/4$, (5) holds true due to the fact that $A_i \leq 3(\rho'_{i-1})^2\binom{\mu_i}{2}$ and $(\chi_1+\ldots+\chi_k) = q_b$, the total number of blue edges across the components of $\mathcal{G}^=$ and $\mu_1+\ldots+\mu_k \leq \beta$.

Bounding D.2. For bounding $\mathsf{D.2}$, we have

$$\mathsf{D.2} \quad \geq \quad \prod_{i=1}^{k'}\left(1-\frac{\mu'_{\rho_k+i}+\alpha_i^{\neq}}{(2^n-\rho'_k-i+1)}\right) \overset{(6)}{\geq} \left(1-\frac{2q_o}{2^n}-\frac{2q_r}{2^n}-\frac{2\alpha^=}{2^n}\right),$$

where (6) follows due to the fact that $(\rho'_k+i-1) \leq 2^n/2$ and we denote $(\mu'_{\rho_k+1}+\ldots+\mu'_{\rho_k+k'}) = q_o+q_r$, the total number of red and orange edges incident on the vertices outside of the set $\mathcal{V}^=$ and $\alpha^{\neq} = \alpha_1^{\neq}+\ldots+\alpha_{k'}^{\neq}$.

Combining D.1 and D.2. Finally, by combining the expression of $\mathsf{D.1}$ and $\mathsf{D.2}$, we have

$$\begin{aligned}h_\beta\frac{2^{nq_m}}{(2^n-\sigma)_\beta} \quad &\geq \quad \left(1-\sum_{i=1}^{k}\frac{6(\rho'_{i-1})^2\binom{\mu_i}{2}}{2^{2n}}-\frac{2q_b}{2^n}-\sum_{i=1}^{k}\frac{2\alpha_i}{2^n}-\frac{2q_0}{2^n}-\frac{2q_r}{2^n}-\frac{2\alpha^{\neq}}{2^n}\right)\\ &\geq \quad \left(1-\sum_{i=1}^{k}\frac{6(\rho'_{i-1})^2\binom{\mu_i}{2}}{2^{2n}}-\frac{2q_v}{2^n}\right),\end{aligned}$$

where recall that $q_v^* = \alpha_1+\ldots+\alpha_k+\alpha^{\neq}$ and $q_v = q_b+q_r+q_o+q_v^*$, the total number of non-equation edges in $\mathcal{G}$. □

4 Multi-User Security of nEHtM$_p^*$

Dutta and Nandi [30] have proven that nEHtM_p is $2n/3$-bit secure in public permutaion model and the bound is tight. Moreover, the bound degrades gracefully with the repetition of the nonce. However, as mentioned earlier, the security proof of nEHtM_p in [30], which is based on the Expectation method [35], is not complete as the authors missed some bad events. In this paper, we propose a slight variant of nEHtM_p, called nEHtM_p^*, which is defined in Eqn. (2) and depicted in the right side of Fig. 1.1. We have shown that the construction is secured roughly upto $2^{2n/3}$ signing queries and $2^{2n/3}$ verification queries. Moreover, we have analyzed the security of this construction in the multi-user setting and have proven similar level of security using the Expectation method. Attack on nEHtM_p^* is exactly similar to the attack of nEHtM_p [30] and hence we omit it.

4.1 Security Theorem of nEHtM$_p^*$

We prove nEHtM_p^* is secure against all adversaries that make roughly $2^{2n/3}$ queries in the multi-user setting. Moreover, our proven bound degrades gracefully with the repetition of nonces in faulty nonce model. Similar to nEHtM_p, the construction posses a birthday bound forging attack when the number of faulty nonces reaches to an order of $2^{n/2}$ [31].

Theorem 2. *Let $\mathcal{M}$ and $\mathcal{K}_h$ be two finite and non-empty sets. Let $\pi \leftarrow\!\!\$\, \mathbb{P}(n)$ be an n-bit public random permutation and $\mathsf{H} : \mathcal{K}_h \times \mathcal{M} \to \{0,1\}^{n-1}$ be an $(n-1)$-bit ϵ-almost xor universal and ϵ-almost regular hash function* [3]*. Moreover, let $\mathsf{K} = (k_1, \ldots, k_\mu) \leftarrow\!\!\$\, \{0,1\}^{n-1}$* [4] *be a set of $n-1$-bit random keys for u users and η, ξ be two fixed parameters. Then the forging advantage for any $(\eta, q_m, q_v, 2p)$-adversary against the construction $\mathsf{nEHtM}^*_p[\pi, \mathsf{H}, \mathsf{K}]$ that makes at most η faulty queries out of q_m signing, q_v veritication and altogether $2p$ primitive queries, is given by*

$$\begin{aligned}\mathbf{Adv}^{\text{mu-nMAC}}_{\mathsf{nEHtM}^*_p}(\eta, q_m, q_v, 2p) \leq\ & \frac{196p^2q_m}{2^{2n}} + \frac{196pq_m^2}{2^{2n}} + \frac{7pq_m^2\epsilon}{2^n} + \frac{4p^2q_m\epsilon}{2^n} + \frac{4q_m^2p^2\epsilon^2}{2^n} + \frac{52q_m^2p^2\epsilon}{2^{2n}} \\ & + \frac{6q_m^2p^2}{2^{3n}} + \frac{q_m^2}{2^{2n}} + p\sqrt{6nq_m}\left(\epsilon + \frac{2}{2^n}\right) + \frac{2q_m^3\epsilon}{2^n} + \frac{q_m^2\epsilon}{2^{n+1}} + \frac{q_m^2\epsilon}{2\xi} \\ & + \frac{q_m + 2q_v + 5}{2^n} + \frac{2pq_mq_v\epsilon}{2^n} + \frac{2p^2q_v\epsilon}{2^n} + q_v\epsilon + \frac{q_mq_vp^2\epsilon^2}{2^n} + \frac{2q_mq_v\epsilon}{2^n} \\ & + \frac{12q_m^4\epsilon}{2^{2n}} + \frac{48q_m^3}{2^{2n}} + \frac{48pq_m^3\epsilon}{2^{2n}} + \eta\left(2p\epsilon + \frac{2}{2^n} + \frac{24q_vp^2\epsilon}{2^{2n}}\right) \\ & + \eta^2\left(\epsilon + \frac{12q_m^2}{2^{2n}} + \frac{48pq_m}{2^{2n}} + \frac{48p^2}{2^{2n}}\right).\end{aligned}$$

We defer the proof of this theorem in Sect. 5.

Interpretation of the bound: By assuming the almost-xor-universal advantage and the almost-regular advantage $\epsilon \approx 2^{-n}$, the security of the construction is valid as long as the number of primitive queries p, the number of MAC queries q_m and the number of verification queries q_v is at most $2^{2n/3}$ for all $\eta \ll 2^{n/2}$. In other words, as long as the overall number of faulty queries is within a birthday limit of the block size of the permutation, the construction gives $2n/3$-bit MAC security.

4.2 Instantiation of nEHtM^*_p with PolyHash Function

We instantiate the underlying almost-xor-universal hash function of nEHtM^*_p using the Polyhash function to realize a permutation based multi-user secure nonce based MAC nEHtM^+_p. Let $\mathsf{Poly} : \{0,1\}^n \times (\{0,1\}^n)^* \to \{0,1\}^n$ be a hash function defined as follows: For a fixed key $k_h \in \{0,1\}^n$ and for a fixed message m, we first apply an injective padding such as 10^* i.e., pad 1 followed by minimum number of zeros so that the total number of bits in the padded message becomes multiple of n. Let the padded message be $m^* = m_1 \| m_2 \| \ldots \| m_l$ where for each i, $|m_i| = n$. Then we define

$$\mathsf{Poly}_{k_h}(m) = m_l \cdot k_h \oplus m_{l-1} \cdot k_h^2 \oplus \ldots \oplus m_1 \cdot k_h^l, \tag{7}$$

where l is the number of n-bit blocks. Then, the almost-regular (ϵ_{reg}) and the almost-xor universal (ϵ_{axu}) advantage of Poly is $\ell/2^n$ [27]. By plugging-in the regular advantage of

[3] For the sake of simplicity of the security bound, we choose $\epsilon_{\text{axu}} = \epsilon_{\text{reg}} = \epsilon$.

[4] K denotes the tuple of user keys; yet the construction takes only a single key.

and the almost-xor-universal advantage of the Polyhash function in Theorem 2, we have

$$\begin{aligned}\mathbf{Adv}^{\text{mu-nMAC}}_{\mathsf{nEHtM}_p^+}(\eta, q_m, q_v, 2p) \le{}& \frac{196p^2q_m}{2^{2n}} + \frac{196pq_m^2}{2^{2n}} + \frac{7pq_m^2\ell}{2^{2n}} + \frac{4p^2q_m\ell}{2^{2n}} + \frac{4q_m^2p^2\ell^2}{2^{3n}} + \frac{52q_m^2p^2\ell}{2^{3n}} \\ &+ \frac{6q_m^2p^2}{2^{3n}} + \frac{q_m^2}{2^{2n}} + p\sqrt{6nq_m}\left(\frac{\ell+2}{2^n}\right) + \frac{2q_m^3\ell}{2^{2n}} + \frac{q_m^2\ell}{2^{2n+1}} + \frac{q_m^2\ell}{2^n \cdot 2\xi} \\ &+ \frac{q_m+2q_v+5}{2^n} + \frac{2pq_mq_v\ell}{2^{2n}} + \frac{2p^2q_v\ell}{2^{2n}} + \frac{q_v\ell}{2^n} + \frac{q_mq_vp^2\ell^2}{2^{2n}} + \frac{2q_mq_v\ell}{2^{2n}} \\ &+ \frac{12q_m^4\ell}{2^{3n}} + \frac{48q_m^3}{2^{2n}} + \frac{48pq_m^3\ell}{2^{3n}} + \eta\left(\frac{2p\ell}{2^n} + \frac{2}{2^n} + \frac{24q_vp^2\ell}{2^{3n}}\right) \\ &+ \eta^2\left(\frac{\ell}{2^n} + \frac{12q_m^2}{2^{2n}} + \frac{48pq_m}{2^{2n}} + \frac{48p^2}{2^{2n}}\right).\end{aligned}$$

5 Proof of Theorem 2

Due to Eqn. (3), we bound the distinguishing advantage instead of bounding the forging advantage of nEHtM_p^*. For this, we consider any information theoretic deterministic distinghisher A that has access to the following oracles in either the real world or the ideal world: in the real world, the distinguisher A has access to the following pair of oracles $(\mathsf{nEHtM}_p^*.\mathsf{Sig}^{\pi}_{(k^u,k_h^u)}, \mathsf{nEHtM}_p^*.\mathsf{Ver}^{\pi}_{(k^u,k_h^u)})_{u=1}^{\mu}$ and $\pi^{\pm}$; in the ideal world, it has access to $(\mathsf{RF}_i, \mathsf{Rej}_i)_{u=1}^{\mu}$ and $\pi^{\pm}$. Here, each MAC query is made for user index $u_i \in \{1, \ldots, \mu\}$, for $i = 1, \ldots, q_m$; and each verification query is made for user index $u'_a \in \{1, \ldots, \mu\}$, for $a = 1, \ldots, q_v$. We summarize the interactions of the distinguisher with its oracle in a transcript $\tau_m \cup \tau_v$, where $\tau_m \stackrel{\Delta}{=} \{(u_1, \nu_1, m_1, t_1), \ldots, (u_{q_m}, \nu_{q_m}, m_{q_m}, t_{q_m})\}$ is the MAC transcript, and $\tau_v \stackrel{\Delta}{=} \{(u'_1, \nu'_1, m'_1, t'_1, b'_1), \ldots, (u'_1, \nu'_{q_v}, m'_{q_v}, t'_{q_v}, b'_{q_v})\}$ is the verification transcript. Primitives queries to π are summarized in two disjoint lists in the form of $\tau_p^{(0)} \stackrel{\Delta}{=} \{(x_1^0, y_1^0), \ldots, (x_p^0, y_p^0)\}$ and $\tau_p^{(1)} \stackrel{\Delta}{=} \{(x_1^1, y_1^1), \ldots, (x_p^1, y_p^1)\}$, where $\mathsf{msb}(x_i^b) = b$ for $b \in \{0, 1\}$. We assume that none of the transcripts contain any duplicate elements. After the interaction, we reveal the keys $k^1, \ldots, k^{\mu}, k_h^1, \ldots, k_h^{\mu}$ to the distinguisher (before it output its decision), which happens to be the keys used in the construction for the real world, and uniformly sampled dummy keys for the ideal world. These keys are summarized in a transcript $\tau_k \stackrel{\Delta}{=} \{k^1, \ldots, k^{\mu}, k_h^1, \ldots, k_h^{\mu}\}$. The complete view is denoted by $\tau = (\tau_m, \tau_v, \tau_p^{(0)}, \tau_p^{(1)}, \tau_k)$.

Let D_{re} (resp. D_{id}) be the random variable that takes a transcript resulting from the interaction between A and the oracles of the real world (resp. the ideal world). A transcript τ is said to be *attainble* if $\Pr[\mathsf{D}_{\text{id}} = \tau] > 0$. Let Θ denotes the set of all attainable transcripts. Let $\Phi : \Theta \to [0, \infty)$ be a non-negative function which maps any attainable transcript to a non-negative real value. Suppose there is a set of good transcripts $\mathsf{GoodT} \subseteq \Theta$ such that for any $\tau \in \mathsf{GoodT}$,

$$\frac{\Pr[\mathsf{D}_{\text{re}} = \tau]}{\Pr[\mathsf{D}_{\text{id}} = \tau]} \ge 1 - \Phi(\tau). \tag{8}$$

Then, the Expectation Method by Hoang and Tessaro [35] says that the statistical distance between D_{re} and D_{id} can be bounded as

$$\Delta(\mathsf{D}_{\text{re}}, \mathsf{D}_{\text{id}}) \le \mathbf{E}[\Phi(\mathsf{D}_{\text{id}})] + \Pr[\mathsf{D}_{\text{id}} \in \mathsf{BadT}], \tag{9}$$

where $\mathsf{BadT} \stackrel{\Delta}{=} \Theta \setminus \mathsf{GoodT}$ is the set of all bad transcripts. In other words, the advantage of A in distinguishing $\mathbf{O}_{\text{re}}$ from $\mathbf{O}_{\text{id}}$ is bounded by $\mathbf{E}[\Phi(\mathsf{D}_{\text{id}})] + \Pr[\mathsf{D}_{\text{id}} \in \mathsf{BadT}]$. In the rest of the paper, we write Θ, GoodT, and BadT to denote the set of attainable, set of good, and set of bad transcripts, respectively. Note that the expectation method is a generic tool over H-Coefficient technique [45] for bounding the distinguishing advantage of two random systems as the latter can be derived as a simple corollary of the former when Φ is taken to be a constant function.

5.1 Definition of Bad Transcripts

In this section, we define bad transcripts. For the notational simplicity, we denote $\mathsf{H}_{k_h^{u_i}}(m_i) = \mathsf{H}_i^{u_i}$, and let $\hat{x}^b$ denotes $\mathsf{chop}_{\mathrm{msb}}(x^b)$ for $b = 0, 1$. We also define a multiset and two sets as follows: (a) $\mathcal{T} \overset{\Delta}{=} \{t_i : (u_i, \nu_i, m_i, t_i) \in \tau_m\}$, multiset of responses corresponding to construction queries, (b) $\mathcal{Y}_0 \overset{\Delta}{=} \{y^0 : (x^0, y^0) \in \tau_p^{(0)}\}$ and (c) $\mathcal{Y}_1 \overset{\Delta}{=} \{y^1 : (x^1, y^1) \in \tau_p^{(1)}\}$.

Rationale of Bad Transcripts: For a transcript $\tau' = (\tau_m, \tau_v, \tau_p^{(0)}, \tau_p^{(1)}, \tau_k)$ and an n-bit permutation π, we have the following system of equations and non-equations, along with $\pi(\hat{x}_i^b) = y_i^b$, for $b \in \{0, 1\}$ and $i \in [q_p]$. We call the equations of block $\mathbb{E}^=$ as "MAC equations" and the equations of block $\mathbb{E}^{\neq}$ as "verification non-equations". For the notational simplicity, let us denote $\nu \oplus k$ as α and $\nu \oplus k \oplus \mathsf{H}$ as β

$$\mathbb{E}^= = \begin{cases} \pi(0\|\alpha_1) \oplus \pi(1\|\beta_1) = t_1 \\ \pi(0\|\alpha_2) \oplus \pi(1\|\beta_2) = t_2 \\ \qquad \vdots \\ \pi(0\|\alpha_{q_m}) \oplus \pi(1\|\beta_{q_m}) = t_{q_m} \end{cases} \qquad \mathbb{E}^{\neq} = \begin{cases} \pi(0\|\alpha'_1) \oplus \pi(1\|\beta'_1) \neq t'_1 \\ \pi(0\|\alpha'_2) \oplus \pi(1\|\beta'_2) \neq t'_2 \\ \qquad \vdots \\ \pi(0\|\alpha_{q_v}) \oplus \pi(1\|\beta_{q_v}) \neq t'_{q_v}. \end{cases}$$

For a good transcript, we are required to count the number of permutations π that satisfy each of the equations in block $\mathbb{E}^=$ and $\mathbb{E}^{\neq}$. Therefore, we identify the events, we call them **bad**, that do not let us to do our job. We enlist such events in two parts: in the first part, we are concerned only with the MAC equations and in the other part, we include the verification non-equations.

MAC Equations.

1. For any equation of block $\mathbb{E}^=$, let both inputs be non-fresh. An input becomes non-fresh in either of the two ways: (a) if it collides with the input of some earlier MAC query or (b) if it collides with the input of some primitive query. These cases are depicted in B.1-B.4 of Fig. 5.1.

2. For any equation of block $\mathbb{E}^=$, let exactly one of the inputs be non-fresh and the other be fresh. This condition actually determines the permutation output corresponding to the fresh input. But bad event happens if the determined value becomes non-fresh as depicted in B.5-B.8 of Fig. 5.1.

3. For any two equations of block $\mathbb{E}^=$, if exactly one of the inputs of each equations collides with the input of some primitive query, then the permutation output of their fresh-input counterparts become determined. Bad events happen when these two determined outputs collide as depicted in B.9-B.11 of Fig. 5.1.

For the part including the verification queries, a bad event happens when the adversary makes a valid forging attempt. Hence, we need to identify the cases where the adversary can potentially make a valid forging attempt.

Verification Non-Equations.

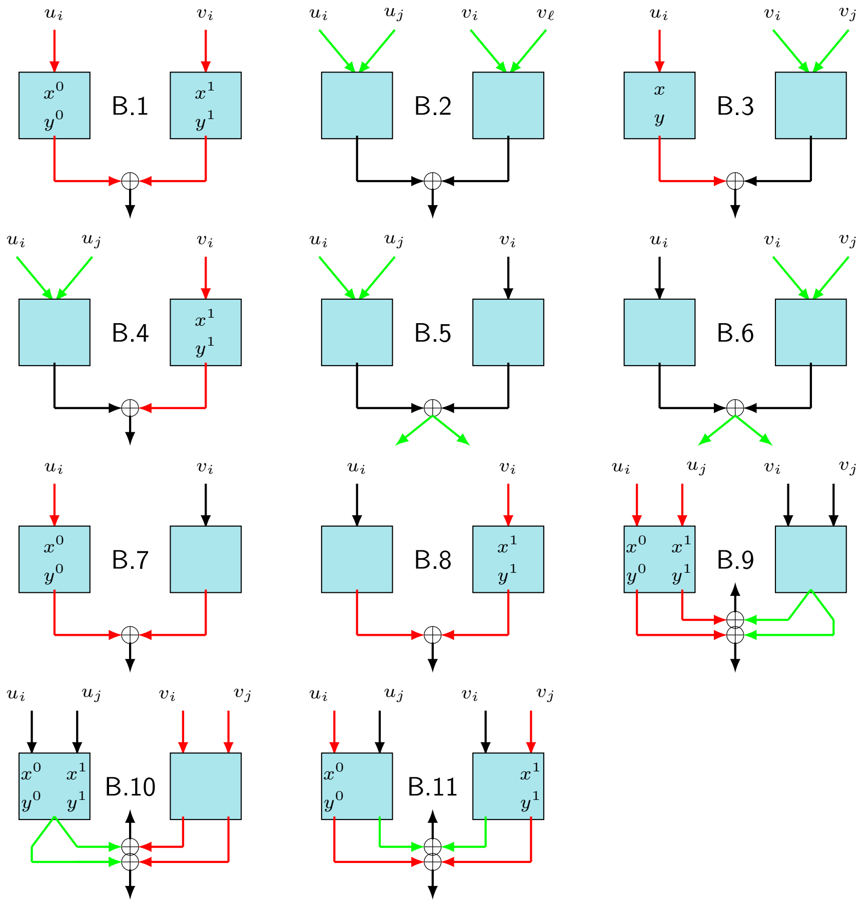

Figure 5.1: Different cases of bad events regarding the MAC and primitive queries. Red edge denotes the input / output collides with prmitive input / output. Green edge denotes that input collides with the input of some other construction queries or output of some other construction queries.

4. For any non-equation of block $\mathbb{E}^{\neq}$, if both inputs are non-fresh and the corresponding tag is set to the appropriate value. This gives rise to the following subcases:

   – Both inputs of the a-th non-equation of block $\mathbb{E}^{\neq}$ collide with the inputs of two different primitive queries, and the corresponding tag t'_a is set to the xor of the outputs of those two primitive queries. This case is depicted in B.12 of Fig. 5.1.

   – Both inputs of the a-th non-equation of block $\mathbb{E}^{\neq}$ collide with the corresponding inputs of the i-th equation of block $\mathbb{E}^{=}$, and the corresponding tag t'_a is set to t_i. This case is depicted in B.13 of Fig. 5.1.

   – One of the inputs of the a-th non-equation of block $\mathbb{E}^{\neq}$ collides with the input of some primitive query, the other input of this non-equation collides with the corresponding input of the i-th equation of block $\mathbb{E}^{=}$, the remaining input of the i-th equation of block $\mathbb{E}^{=}$ collides with the input of some different primitive query, and finally the xor of tags $t'_a \oplus t_i$ is set to the xor of the outputs of those two primitive queries. These cases are depicted in B.14 and B.15 of Fig. 5.1.

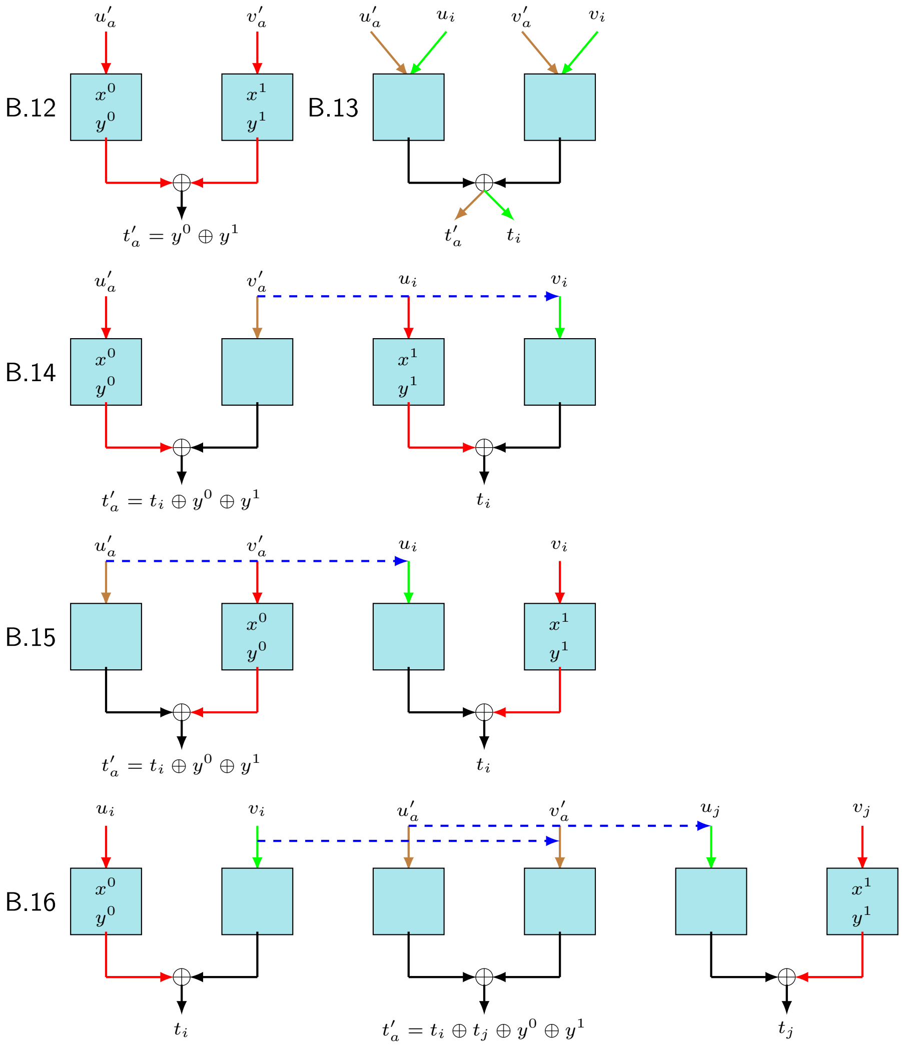

Figure 5.2: Different cases of bad events regarding the verification queries. Red edge denotes the input / output collides with prmitive input / output. Orange edge denotes the input of the verification query. Green edge denotes the input of some MAC queries and the blue dotted edge denotes the corresponding two inputs have collided.

- Nonce input of the a-th equation of block $\mathbb{E}^{\neq}$ collides with the nonce input of j-th equation of block $\mathbb{E}^{=}$, and the other input of this non-equation collides with the corresponding input of i-th equation of block $\mathbb{E}^{=}$. Moreover, the other two inputs of i-th and j-th equation of block $\mathbb{E}^{=}$ collide with the input of two different primitive queries, and finally the xor of tags $t'_a \oplus t_i \oplus t_j$ is set to the xor of the outputs of those two primitive queries. This case depicted in B.16 of Fig. 5.1.

Note that in the work of Dutta and Nandi [30], they did not consider the bad events related to point 3 and the last two cases of point 4.

Definition 2 (Bad Transcript for nEHtM_p^*). Given a parameter $\xi \in \mathbb{N}$, where $\xi \geq \eta$, an attainable transcript $\tau' = (\tau_m, \tau_v, \tau_p^{(0)}, \tau_p^{(1)}, \tau_k)$ is called a **bad** transcript if any one of the following holds:

1. Both the inputs to the permutation are non-fresh -- condn (1).

   - B.1 : $\exists\ i \in [q_m], (\hat{x}^0, \hat{y}^0), (\hat{x}^1, \hat{y}^1)$ such that $\nu_i \oplus k^{u_i} = \hat{x}^0, \nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1$.
   - B.2 : $\exists\ i, j, \ell \in [q_m], i \neq j, j \neq \ell$ such that $\nu_i \oplus k^{u_i} = \nu_j \oplus k^{u_j}, \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j} = \nu_\ell \oplus k^{u_\ell} \oplus \mathsf{H}_\ell^{u_\ell}$.
   - B.3 : $\exists\ i \neq j \in [q_m], (\hat{x}^0, \hat{y}^0)$ such that $\nu_i \oplus k^{u_i} = \hat{x}^0, \nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j}$.
   - B.4 : $\exists\ i \neq j \in [q_m], (\hat{x}^1, \hat{y}^1)$ such that $\nu_i \oplus k^{u_i} = \nu_j \oplus k^{u_j}, \nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1$.

2. Exactly one i/p is non-fresh and the o/p of the fresh input is non-fresh -- condn (2).

   - B.5 : $\exists\ i \neq j \in [q_m]$ such that $\nu_i \oplus k^{u_i} = \nu_j \oplus k^{u_j}, t_i = t_j$.
   - B.6 : $\exists\ i \neq j \in [q_m]$ such that $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j}, t_i = t_j$.
   - B.7 : $\exists\ i \in [q_m], (\hat{x}^0, \hat{y}^0)$ such that $\nu_i \oplus k^{u_i} = \hat{x}^0, \hat{y}^0 \oplus t_i \in \mathcal{Y}_0 \cup \mathcal{Y}_1$.
   - B.8 : $\exists\ i \in [q_m], (\hat{x}^1, \hat{y}^1)$ such that $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1, \hat{y}^1 \oplus t_i \in \mathcal{Y}_0 \cup \mathcal{Y}_1$.

3. For two construction queries, exactly one of the inputs collides with the input of some primitive query and the permutation output of their fresh-input counterpart collides -- condn (3).

   - B.9 : $\exists\ i \neq j \in [q_m], (\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)$ such that $\nu_i \oplus k^{u_i} = \hat{x}^0, \nu_j \oplus k^{u_j} = \hat{x}'^0, t_i \oplus \hat{y}^0 = t_j \oplus \hat{y}'^0$.
   - B.10 : $\exists\ i \neq j \in [q_m], (\hat{x}^1, \hat{y}^1), (\hat{x}'^1, \hat{y}'^1)$ such that $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1, \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j} = \hat{x}'^1, t_i \oplus \hat{y}^1 = t_j \oplus \hat{y}'^1$.
   - B.11 : $\exists\ i \neq j \in [q_m], (\hat{x}^0, \hat{y}^0), (\hat{x}^1, \hat{y}^1)$ such that $\nu_i \oplus k^{u_i} = \hat{x}^0, \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j} = \hat{x}^1, t_i \oplus \hat{y}^0 = t_j \oplus \hat{y}^1$.

4. Both the inputs of the verification query are non-fresh -- condn (4).

   - B.12 $\exists\ a \in [q_v], (\hat{x}^0, \hat{y}^0), (\hat{x}^1, \hat{y}^1)$ such that $\nu'_a \oplus k^{u'_a} = \hat{x}^0$, $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a = \hat{x}^1, t'_a = \hat{y}^0 \oplus \hat{y}^1$.
   - B.13 $\exists\ a \in [q_v]$, $\exists\ i \in [q_m]$ such that $\nu_i \oplus k^{u_i} = \nu'_a \oplus k^{u'_a}$, $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a, t_i = t'_a$.
   - B.14 $\exists\ a \in [q_v]$, $\exists\ i \in [q_m], (\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)$ such that $\nu_i \oplus k^{u_i} = \hat{x}^0, \nu'_a \oplus k^{u'_a} = \hat{x}'^0$, $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a, t'_a = \hat{y}^0 \oplus \hat{y}'^0 \oplus t_i$.
   - B.15 $\exists\ a \in [q_v]$, $\exists\ i \in [q_m], (\hat{x}^1, \hat{y}^1), (\hat{x}'^1, \hat{y}'^1)$ such that $\nu_i \oplus k^{u_i} = \nu'_a \oplus k^{u'_a}, \nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1, \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a = \hat{x}'^1, t'_a = \hat{y}^1 \oplus \hat{y}'^1 \oplus t_i$.
   - B.16 $\exists\ a \in [q_v], \exists\ i \neq j \in [q_m], (\hat{x}^0, \hat{y}^0), (\hat{x}^1, \hat{y}^1)$ such that $\nu_i \oplus k^{u_i} = \hat{x}^0, \nu_i \oplus k^{u_i} = \nu'_a \oplus k^{u'_a}, \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j} = \hat{x}^1, \nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a, t'_a = \hat{y}^0 \oplus \hat{y}^1 \oplus t_i \oplus t_j$.

5. Additional Bad Events.

- $\mathsf{B.17}$: $\{i_1, \ldots, i_{\xi+1}\} \subseteq [q_m]$ such that $\nu_{i_1} \oplus k_{i_1} \oplus \mathsf{H}_{i_1} = \nu_{i_2} \oplus k_{i_2} \oplus \mathsf{H}_{i_2} = \ldots = \nu_{i_{\xi+1}} \oplus k_{\xi+1} \oplus \mathsf{H}_{i_{\xi+1}}$ (the optimal value of ξ shall be determined later in the proof).
- $\mathsf{B.18}$ $\exists\, i \in [q_m]$ such that $t_i = 0^n$.

Lemma 2. *Let* D_{id} *and* BadT *be defined as above. Then*

$$\begin{aligned}\Pr[\mathsf{D}_{\mathrm{id}} \in \mathsf{BadT}] \leq\ & \frac{4p^2 q_m}{2^{2n}} + \frac{4pq_m^2}{2^{2n}} + \frac{7pq_m^2\epsilon}{2^n} + \frac{4p^2q_m\epsilon}{2^n} + \frac{4q_m^2p^2\epsilon^2}{2^n} + \frac{4q_m^2p^2\epsilon}{2^{2n}} + \frac{6q_m^2p^2}{2^{3n}} + \frac{q_m^2}{2^{2n}} \\ & + p\sqrt{6nq_m}\left(\epsilon + \frac{2}{2^n}\right) + \frac{2q_m^3\epsilon}{2^n} + \frac{q_m^2\epsilon}{2^{n+1}} + \frac{q_m^2\epsilon}{2\xi} + \frac{q_m+5}{2^n} + \frac{2pq_mq_v\epsilon}{2^n} + \frac{2p^2q_v\epsilon}{2^n} \\ & + q_v\epsilon + \frac{q_mq_vp^2\epsilon^2}{2^n} + \frac{2q_mq_v\epsilon}{2^n} + \eta^2\epsilon + \eta\left(2p\epsilon + \frac{2}{2^n} + \frac{24q_vp^2\epsilon}{2^{2n}}\right).\end{aligned}$$

We defer the proof of the lemma in Sect. C.

5.2 Analysis of Good Transcripts

In this section, we show that for a good transcript $\tau' = (\tau_m, \tau_v, \tau_p^{(0)}, \tau_p^{(1)}, \tau_k)$, realizing τ' is almost as likely in the real world as in the ideal world. For the simplicity of the notation, we write C_u^π to denote the construction $\mathsf{nEHtM}_p^{*\pi}$ using keys k^u and k_h^u.

Lemma 3 (Good Lemma). *Let* $\tau' = (\tau_m, \tau_v, \tau_p^{(0)}, \tau_p^{(1)}, \tau_k)$ *be a good transcript. Then, we have*

$$\frac{\Pr[\mathsf{D}_{\mathrm{re}} = \tau']}{\Pr[\mathsf{D}_{\mathrm{id}} = \tau']} \geq \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2\binom{\mu_i}{2}}{2^{2n}} - \frac{q_v}{2^n}\right),$$

Proof. Since the ideal world always rejects the verification attempt, for a good transcript τ', the ideal interpolation probability is

$$\Pr[\mathsf{D}_{\mathrm{id}} = \tau'] = \frac{1}{|\mathcal{K}_h|^\mu} \cdot \frac{1}{2^{\mu(n-1)}} \cdot \frac{1}{2^{nq_m}} \cdot \frac{1}{(2^n)_{2p}}. \tag{10}$$

We must now lower bound the probability of getting τ' in the real world, i.e., we lower bound the following:

$$\begin{aligned}\Pr[\mathsf{D}_{\mathrm{re}} = \tau'] &= \frac{1}{|\mathcal{K}_h|^\mu} \cdot \frac{1}{2^{\mu(n-1)}} \cdot \Pr[\pi \leftarrow\!\!{\scriptstyle\$}\, \mathbb{P}(n) : (\mathsf{C}_u^\pi)_{u=1}^\mu \mapsto \tau_m \cup \tau_v \wedge \pi \mapsto \tau_p^{(0)} \cup \tau_p^{(1)}] \\ &= \frac{1}{|\mathcal{K}_h|^\mu} \cdot \frac{1}{2^{\mu(n-1)}(2^n)_{2p}} \cdot \Pr[\pi \leftarrow\!\!{\scriptstyle\$}\, \mathbb{P}(n) : (\mathsf{C}_u^\pi)_{u=1}^\mu \mapsto \tau_m \cup \tau_v \mid \pi \mapsto \tau_p^{(0)} \cup \tau_p^{(1)}].\end{aligned}$$

Therefore, by taking the ratio of real to ideal interpolation probability, we have

$$\frac{\Pr[\mathsf{D}_{\mathrm{re}} = \tau']}{\Pr[\mathsf{D}_{\mathrm{id}} = \tau']} = 2^{nq_m} \cdot \underbrace{\Pr[\pi \leftarrow\!\!{\scriptstyle\$}\, \mathbb{P}(n) : (\mathsf{C}_u^\pi)_{u=1}^\mu \mapsto \tau_m \cup \tau_v \mid \pi \mapsto \tau_p^{(0)} \cup \tau_p^{(1)}]}_{\mathsf{Z}}. \tag{11}$$

Now, our goal is to lower bound Z. We say that a permutation π that is fixed on $2p$ input-output pairs (due to $\tau_p^{(0)} \cup \tau_p^{(1)}$), is compatible with τ_m if

$$\pi(0\|\nu_i \oplus k^{u_i}) \oplus \pi(1\|\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i}) = t_i, \quad i \in \{1, \ldots, q_m\},$$

and it is compatible with τ_v if

$$\pi(0\|\nu'_a \oplus k^{u'_a}) \oplus \pi(1\|\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a) \neq t'_a, \quad a \in \{1, \ldots, q_v\}.$$

We say that a permutation π is compatible with τ' if it is compatible with both τ_m and τ_v. We denote $\mathsf{Comp}(\tau_m), \mathsf{Comp}(\tau_v)$ and $\mathsf{Comp}(\tau')$ the set of permutations that are compatible with τ_m, τ_v and τ', respectively. Then, one can easily check that

$$\mathsf{Z} = \Pr[\pi \leftarrow\!\!\$\, \mathbb{P}(n) : \pi \in \mathsf{Comp}(\tau')].$$

<u>Bounding Z.</u> We first consider the probability that a random permutation π that is compatible with $2p$ input-output pairs is also compatible with the MAC and the verification transcript $\tau_m \cup \tau_v$. From now onwards, we write τ_p to denote $\tau_p^{(0)} \cup \tau_p^{(1)}$. Let U_b (resp. V_b) denotes the set of inputs (resp. outputs) of the primitive queries of $\tau_p^{(b)}$ for $b \in \{0,1\}$, i.e., for $b \in \{0,1\}$,

$$U_b = \{\hat{x}^b : (\hat{x}^b, \hat{y}^b) \in \tau_p^{(b)}\}, \quad V_b = \{\hat{y}^b : (\hat{x}^b, \hat{y}^b) \in \tau_p^{(b)}\}.$$

We now partition the transcript τ_m as follows:

$$\begin{aligned}
\mathcal{Q}_0 &= \{(u_i, \nu_i, m_i, t_i) \in \tau_m : \nu_i \oplus k \in U_0 \\
&\quad \wedge \forall (u'_a, \nu'_a, m'_a, t'_a) \in \tau_v, \nu'_a \oplus k^{u'_a} \neq \nu_i \oplus k^{u_i} \wedge \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a \neq \nu_i \oplus k^{u_i} \oplus \mathsf{H}^{u_i}_i\} \\
\mathcal{Q}_1 &= \{(u_i, \nu_i, m_i, t_i) \in \tau_m : \nu_i \oplus k \oplus \mathsf{H_i} \in U_1 \\
&\quad \wedge \forall (u'_a, \nu'_a, m'_a, t'_a) \in \tau_v, \nu'_a \oplus k^{u'_a} \neq \nu_i \oplus k^{u_i} \wedge \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a \neq \nu_i \oplus k^{u_i} \oplus \mathsf{H}^{u_i}_i\} \\
\mathcal{Q}_2 &= \tau_m \setminus (\mathcal{Q}_0 \cup \mathcal{Q}_1).
\end{aligned}$$

Note that, $\mathcal{Q}_0 \cap \mathcal{Q}_2 = \phi = \mathcal{Q}_1 \cap \mathcal{Q}_2$ follows from the definition. Moreover, $\mathcal{Q}_0 \cap \mathcal{Q}_1 = \phi$ due to $\overline{\mathsf{B.1}}$. Let E_b denotes the event $(\mathsf{C}^\pi_u)^\mu_{u=1} \mapsto \mathcal{Q}_b$, for $b = \{0,1,2\}$. Therefore, we have

$$\begin{aligned}
\mathsf{Z} &= \Pr[\pi \leftarrow\!\!\$\, \mathbb{P}(n) : \mathsf{E}_0 \wedge \mathsf{E}_1 \wedge \mathsf{E}_2 \wedge ((\mathsf{C}^\pi_u)^\mu_{u=1} \mapsto \tau_v) \mid \pi \mapsto \tau_p] \\
&= \underbrace{\Pr[\pi \leftarrow\!\!\$\, \mathbb{P}(n) : \mathsf{E}_0 \wedge \mathsf{E}_1 \mid \pi \mapsto \tau_p]}_{\mathsf{Z.1}} \cdot \underbrace{\Pr[\mathsf{E}_2 \wedge ((\mathsf{C}^\pi_u)^\mu_{u=1} \mapsto \tau_v) \mid \mathsf{E}_0 \wedge \mathsf{E}_1 \wedge \pi \mapsto \tau_p]}_{\mathsf{Z.2}}
\end{aligned}$$

<u>Bounding Z.1:</u> Conditioned on $(\pi \mapsto \tau_p)$, π is fixed on exactly $2p$ values. For each $(u, \nu, m, t) \in \mathcal{Q}_0$, there is a unique $(x^0, y^0) \in \tau_p^{(0)}$ such that $\nu \oplus k = \hat{x}^0$, so that $\pi(0 \| \nu \oplus k)$ is well defined (and equal to y^0). Similarly, for each $(u, \nu, m, t) \in \mathcal{Q}_1$, there is a unique $(x^1, y^1) \in \tau_p^{(1)}$ such that $\nu \oplus k \oplus \mathsf{H} = \hat{x}^1$, so that $\pi(1 \| \nu \oplus k \oplus \mathsf{H})$ is well defined (and equal to y^1). In the following, we let

$$\begin{aligned}
\mathcal{D}_0 &= \{1 \| (\nu_i \oplus k^{u_i} \oplus \mathsf{H}^{u_i}_i) : (u_i, \nu_i, m_i, t_i) \in \mathcal{Q}_0\}, \quad \mathcal{R}_0 = \{\pi(\nu_i \oplus k^{u_i}) \oplus t_i : (u_i, \nu_i, m_i, t_i) \in \mathcal{Q}_0\} \\
\mathcal{D}_1 &= \{0 \| (\nu_i \oplus k^{u_i}) : (u_i, \nu_i, m_i, t_i) \in \mathcal{Q}_1\}, \quad \mathcal{R}_1 = \{\pi(\nu_i \oplus k^{u_i} \oplus \mathsf{H}^{u_i}_i) \oplus t_i : (u_i, \nu_i, m_i, t_i) \in \mathcal{Q}_1\}.
\end{aligned}$$

Note that, the elements of $\mathcal{D}_0$ are all distinct (otherwise satisfy $\mathsf{B.3}$) and they have not collided with the input of any primitive query (otherwise satisfy $\mathsf{B.1}$). Similarly, elements of $\mathcal{D}_1$ are all distinct (otherwise satisfy $\mathsf{B.4}$) and they have not collided with the input of any primitive query (otherwise satisfy $\mathsf{B.1}$). Also, note that the elements of $\mathcal{R}_0$ are all distinct (otherwise satisfy $\mathsf{B.9}$) and they have not collided with the output of any primitive query (otherwise satisfy $\mathsf{B.7}$). Similarly, elements of $\mathcal{R}_1$ are all distinct (otherwise satisfy $\mathsf{B.10}$) and they have not collided with the output of any primitive query (otherwise satisfy $\mathsf{B.8}$). Moreover, $\mathcal{R}_0 \cap \mathcal{R}_1 = \phi$ (otherwise satisfy $\mathsf{B.11}$). Let $|\mathcal{Q}_0| = \alpha_0$ and $|\mathcal{Q}_1| = \alpha_1$. Then $\mathsf{E}_0 \wedge \mathsf{E}_1$ is equivalent to $\alpha_0 + \alpha_1$ new and distinct equations on π so that

$$\Pr[\mathsf{E}_0 \wedge \mathsf{E}_1 \mid \pi \mapsto \tau_p] = \frac{1}{(2^n - 2p)_{\alpha_0 + \alpha_1}}. \tag{12}$$

5.3 Lower Bounding Z.2

Now, we are only required to lower bound Z.2. To do this, recall that we denoted $\nu \oplus k$ as α and $\nu \oplus k \oplus \mathsf{H}$ as β. Moreover, we define $\tau' \leftarrow \mathcal{Q}_0 \cup \tau_v \cup \tau_p^{(0)} \cup \tau_p^{(1)}$. Now, we consider the following system of bivariate affine equations and non-equations associated to τ'

$$\mathbb{E}^{=}_{\tau'} = \begin{cases} \pi(0\|\alpha_1) \oplus \pi(1\|\beta_1) = t_1 \\ \pi(0\|\alpha_2) \oplus \pi(1\|\beta_2) = t_2 \\ \qquad\vdots \\ \pi(0\|\alpha_{q_m}) \oplus \pi(1\|\beta_{q_m}) = t_{q_m} \end{cases} \qquad \mathbb{E}^{\neq}_{\tau'} = \begin{cases} \pi(0\|\alpha'_1) \oplus \pi(1\|\beta'_1) \neq t'_1 \\ \pi(0\|\alpha'_2) \oplus \pi(1\|\beta'_2) \neq t'_2 \\ \qquad\vdots \\ \pi(0\|\alpha'_{q_v}) \oplus \pi(1\|\beta'_{q_v}) \neq t'_{q_v}. \end{cases}$$

Since $\tau = (\tau_m, \tau_v, \tau_p^{(0)}, \tau_p^{(1)}, \tau_k)$ is a good transcript, τ' is also a good transcript. For the good transcript τ', we define a *equation graph*, denoted as $\mathcal{G}_{\tau'}$, corresponding to the system of bivariate affine equations and non-equations $\mathbb{E}^{=}_{\tau'} \cup \mathbb{E}^{\neq}_{\tau'}$ as follows: the set of vertices of $\mathcal{G}_{\tau'}$ corresponds to the variable of $\mathbb{E}^{=}_{\tau'} \cup \mathbb{E}^{\neq}_{\tau'}$, which we denote as $\mathcal{V}$. Moreover, for each equation

$$\pi(0\|\alpha_i) \oplus \pi(1\|\beta_i) = t_i$$

of $\mathbb{E}^{=}_{\tau'}$, we assign a green colored edge between the corresponding vertices in $\mathcal{G}$ with the label t_i and we denote the set of such green colored edges as $\mathcal{E}^{=}_{\tau'}$. We denote $\mathcal{G}^{=}_{\tau'}$ to be the subgraph of $\mathcal{G}_\tau$ induced by the edges of $\mathcal{E}^{=}_{\tau'}$ and $\mathcal{V}^{=} \subseteq \mathcal{V}$ denotes the set of vertices of $\mathcal{G}^{=}_{\tau'}$. We assume that $\mathcal{G}^{=}_{\tau'}$ has k components. Since, τ' is a good transcript, $\mathcal{G}^{=}_{\tau'}$ is acyclic and each of its component is either a **star component** or a **single-edge component** as depicted in Fig. 5.3. It easily follows from the definition of τ' that

1. for **star component**, the only vertex that can collide with the output of any primitive query, is the centre vertex of the graph.

2. For a **single-edge component**, both of its vertices can collide with the output of any primitive query.

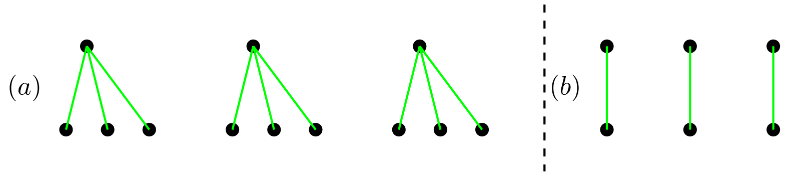

Figure 5.3: (a) Each component is a star type graph and (b) each component is a single edge type graph.

For each non-equation

$$\pi(0\|\alpha'_a) \oplus \pi(1\|\beta'_a) \neq t'_a$$

of $\mathbb{E}^{\neq}_{\tau'}$, we assign three types of edges:

- if $\nu'_a \oplus k^{u'_a} = \nu_i \oplus k^{u_i}, \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a = \nu_j \oplus k^{u_j} \oplus \mathsf{H}^{u_j}_j$ such that $\nu_i \oplus k^{u_i} \neq \nu_j \oplus k^{u_j}$ and $\nu_i \oplus k^{u_i} \oplus \mathsf{H}^{u_i}_i \neq \nu_j \oplus k^{u_j} \oplus \mathsf{H}^{u_j}_j$, i.e., the non-equation connects two vertices of different components of $\mathcal{G}^{=}_{\tau'}$. In this case, we assign a blue colored edge between the corresponding vertices with label t'_a.

- if $\nu'_a \oplus k^{u'_a} = \nu_i \oplus k^{u_i}$ but $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a$ is fresh, i.e., we have a non-equation of the form $\pi(0\|\alpha_i) \oplus \pi(1\|\beta'_a) \neq t'_a$. Similarly, if $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a = \nu_i \oplus k^{u_i} \oplus \mathsf{H}^{u_i}_i$ but

$\nu'_a \oplus k^{u'_a}$ is fresh, i.e., we have a non-equation of the form $\pi(0\|\alpha'_a) \oplus \pi(1\|\beta'_i) \neq t'_a$. For each of the above cases, we assign a orange colored edge between the corresponding vertices with label t'_a. Note that the above cases correspond to the situation where exactly one of the vertices of an non-equation edge coincides with the vertex of $\mathcal{G}^=$.

- if both $\nu'_a \oplus k^{u'_a}$ and $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a$ are fresh, i.e., we have a non-equation of the form $\pi(0\|\alpha'_a) \oplus \pi(1\|\beta'_a) \neq t'_a$. In this case, we assign a red colored edge between the corresponding vertices with label t'_a.

Identifying univariate affine non-equation. Now, we are interested to study that which of the above cases lead to a univarate affine non-equation of the form $\pi(0\|\alpha) \neq c$ or $\pi(1\|\beta) \neq c$; or $\pi(0\|\alpha') \neq c$ or $\pi(1\|\beta') \neq c$.

Note that a univariate affine non-equation arises due to the collision of a vertex of a non-equation edge with one of the primitive queries. We begin with blue edges that connect two components and check when can it lead to an univariate affine non-equation.

1. We first consider that the blue edge connects two **single-edge** components. Note that a univariate affine non-equation arises when exactly one vertex of one of the edges collides with a primitive query. Such a situation can arise if one of the following four conditions holds as depicted in Fig. 5.4:

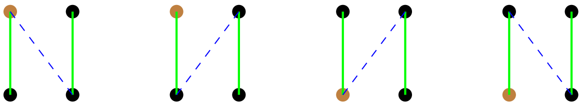

Figure 5.4: Brown colored vertex represents that the node collides with a primitive query. Dashed blue edge denotes the verification non-equation.

2. We consider that the blue edge connects a **star** component with a **single-edge** component. Note that a univariate affine non-equation arises when either (a) the center vertex of the star component collides with a primitive query or (b) one end point of the single-edge component collides with a primitive query. Such a situation can arise if one of the following six conditions holds as depicted in Fig. 5.5.

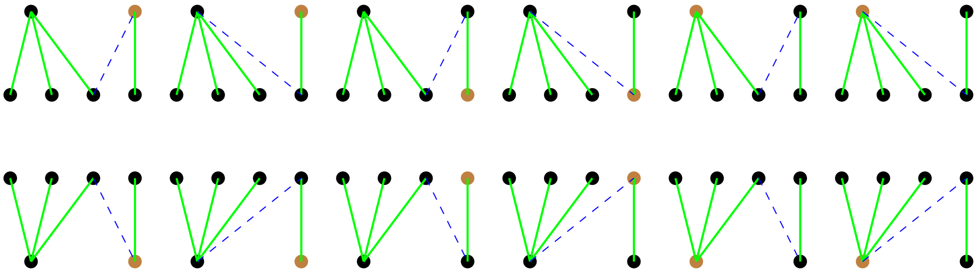

Figure 5.5: Brown colored vertex represents that the node collides with a primitive query. Dahsed blue edge denotes the verification non-equation.

3. We consider that the blue edge connects two **star** components. Note that a univariate affine non-equation arises when exactly one of the center vertex of the star components collides with a primitive query. Such a situation can arise if one of the following four conditions holds as depicted in Fig. 5.6:

We have listed out the set of univariate affine non-equations corresponding to each of the above three cases in Supplementary Sect A.

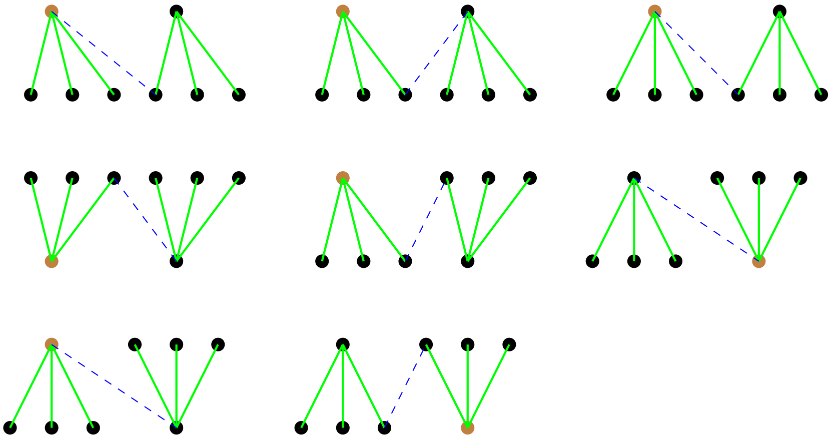

Figure 5.6: Brown colored vertex represents that the node collides with a primitive query. Dashed blue edge denotes the verification non-equation.

Let there are total q_v^* verification queries that leads to the above set of univariate affine non-equations. Apart from the above cases, we have the other cases as follows:

4. $\nu'_a \oplus k^{u'_a} = \nu_i \oplus k^{u_i}$ but $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a$ is fresh
5. $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'_a = \nu_i \oplus k^{u_i} \oplus \mathsf{H}^{u_i}_i$ but $\nu'_a \oplus k^{u'_a}$ is fresh
6. both $\nu'_a \oplus k^{u'_a}$ and $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a$ are fresh

Note that case (4) and (5) can also lead to the following set of univariate affine non-equations. Because one of the vertices of a orange edge can either coincide with a vertex of an edge whose one of the end point collides with a primitive query or the vertex of the orange edge can coincide with a vertex of a star component whose center vertex collides with a primitive query as depicted in Fig. 5.7. We have listed out the set of univariate affine non-equations corresponding to conditions (4), (5) and (6) in Supplementary Sect B.

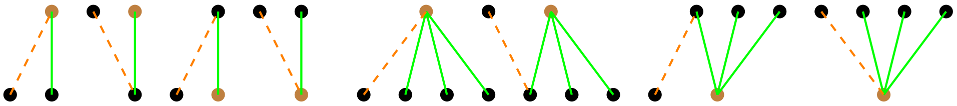

Figure 5.7: Brown colored vertex represents that the node collides with a primitive query. Dashed orange edge denotes the verification non-equation.

Let us assume that there are total q_b blue edges, q_o orange edges and q_r red edges. Therefore, $q_v = q_v^* + q_b + q_o + q_r$. Moreover, note that the variables of $\mathbb{E}^= \cup \mathbb{E}^{\neq}$ are not supposed to be distinct, as they collide with other variables and thus $\mathbb{E}^= \cup \mathbb{E}^{\neq}$ is defined over β many distinct variables. This implies that the number of vertices in the equation graph $\mathcal{G}$ has β many vertices with q'_m green edges and q_b blue edges, q_o orange edges and q_r red edges. Moreover, there are q_v^* univariate affine non-equations. Since, $\tau_m \cup \tau_v$ is a good transcript (as τ' is good), $\mathcal{G}$ is a good graph. Now, it is evident that to lower bound the conditional probability of the event $\mathsf{E}_2 \wedge (\mathsf{C}_u^\pi)_{u=1}^\mu \mapsto \tau_v$ conditioned on $\mathsf{E}_0 \wedge \mathsf{E}_1 \wedge \pi \mapsto \tau_p$ is equivalent to bounding the probability of the event that $\mathbb{E}^= \cup \mathbb{E}^{\neq}$ holds. Therefore, we have

$$\Pr[\mathbb{E}^= \cup \mathbb{E}^{\neq} \text{ holds}] = \frac{h_\beta}{(2^n - 2p - \alpha_0 - \alpha_1)_\beta}, \tag{13}$$

where h_β denotes the number of solutions of the good equation graph $\mathcal{G}$ with β vertices, q'_m green edges, q_b blue edges, q_o orange edges and q_r red edges with q_v^* univariate affine non-equations. Since, $\mathcal{G}$ is good, following Theorem 1 with where k be the total number of components of $\mathcal{G}$, we have

$$h_\beta \geq \frac{(2^n - 2p - \alpha_0 - \alpha_1)_\beta}{2^{nq'_m}} \cdot \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2 \binom{\mu_i}{2}}{2^{2n}} - \frac{2q_v}{2^n}\right). \tag{14}$$

Therefore, from Eqn. (13) and Eqn. (14), we have

$$\Pr[\mathbb{E}^{=} \cup \mathbb{E}^{\neq} \text{ holds } \mid \mathsf{E}_0 \wedge \mathsf{E}_1 \wedge \pi \mapsto \tau_p] \geq \frac{1}{2^{nq'_m}} \cdot \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2 \binom{\mu_i}{2}}{2^{2n}} - \frac{2q_v}{2^n}\right),$$

and thus, we have

$$\Pr[\mathsf{E}_2 \wedge (\mathsf{C}_u^\pi)_{u=1}^{\mu} \mapsto \tau_v \mid \mathsf{E}_0 \wedge \mathsf{E}_1 \wedge \pi \mapsto \tau_p] \geq \frac{1}{2^{nq'_m}} \cdot \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2 \binom{\mu_i}{2}}{2^{2n}} - \frac{2q_v}{2^n}\right). \tag{15}$$

Finally, from Eqn. (12), Eqn. (12) and Eqn. (15), we have

$$\mathbb{Z} \geq \frac{1}{(2^n - 2p)_{\alpha_0 + \alpha_1}} \cdot \frac{1}{2^{nq'_m}} \cdot \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2 \binom{\mu_i}{2}}{2^{2n}} - \frac{2q_v}{2^n}\right). \tag{16}$$

<u>Final Step.</u> In the final step of the proof, we combine Eqn. (11) and Eqn. (16) to obtain

$$\begin{aligned} \frac{\Pr[\mathsf{D}_{\text{re}} = \tau']}{\Pr[\mathsf{D}_{\text{id}} = \tau']} &\geq \underbrace{\frac{(2^n)^{\alpha_0 + \alpha_1}}{(2^n - 2p)_{\alpha_0 + \alpha_1}}}_{\Delta} \cdot \frac{2^{nq'_m}}{2^{nq'_m}} \cdot \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2 \binom{\mu_i}{2}}{2^{2n}} - \frac{2q_v}{2^n}\right) \\ &= \left(1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2 \binom{\mu_i}{2}}{2^{2n}} - \frac{2q_v}{2^n}\right), \end{aligned} \tag{17}$$

where we use the obvious inequality that $\Delta \geq 1$ and $q_m = q'_m + \alpha_0 + \alpha_1$. This concludes the proof of the good lemma. □

<u>Applying the Expectation Method.</u> From Lemma 3, we have

$$\frac{\Pr[\mathsf{D}_{\text{re}} = \tau']}{\Pr[\mathsf{D}_{\text{id}} = \tau']} \overset{(1)}{\geq} 1 - \underbrace{\left(\sum_{i=1}^{k} \frac{24(q'_m + q'_p)^2 \binom{\mu_i}{2}}{2^{2n}} + \frac{2q_v}{2^n}\right)}_{\Phi(\tau')},$$

where the simplification for (1) follows from the fact $\rho'_{i-1} = \alpha + q'_p \leq 2(q'_m + q'_p)$. Now, from Sect.6.2 of [31] we have

$$\mathbf{E}\left[\sum_{i=1}^{k} \binom{\mu_i}{2}\right] \leq (q'_m)^2 \epsilon/2 + \eta^2/2 + 2q'_m. \tag{18}$$

By applying the expectation method of on Eqn. (18), we have

$$\mathbf{E}[\Phi(\mathsf{D}_{\text{id}})] \leq \frac{12(q'_m + q'_p)^2}{2^{2n}} \left((q'_m)^2 \epsilon + \eta^2 + 4q'_m\right) + \frac{2q_v}{2^n}. \tag{19}$$

By doing a simple algebra on Eqn. (19) and by assuming $q'_m \leq q_m, q'_p \leq 4p$, we have

$$\begin{aligned}\mathbf{E}[\Phi(\mathsf{D}_{\mathrm{id}})] \quad &\leq \quad \Bigg(\frac{12q_m^4\epsilon}{2^{2n}} + \frac{12\eta^2 q_m^2}{2^{2n}} + \frac{48q_m^3}{2^{2n}} + \frac{48pq_m^3\epsilon}{2^{2n}} + \frac{48\eta^2 pq_m}{2^{2n}} + \frac{192pq_m^2}{2^{2n}} \\ &\qquad + \frac{48p^2q_m^2\epsilon}{2^{2n}} + \frac{48\eta^2p^2}{2^{2n}} + \frac{192p^2q_m}{2^{2n}} + \frac{2q_v}{2^n}\Bigg). \qquad (20)\end{aligned}$$

Finalizing the proof. We have assumed that $\xi \geq \eta$ and from the condition of Theorem 1, we have $\xi \leq 2^n/(8q'_m + 2q'_p) \leq 2^n/8q'_m$. By assuming $\eta \leq 2^n/8q'_m$ (otherwise the bound becomes vacuously true) we choose $\xi = 2^n/8q'_m$. Hence, the result follows by applying Eqn. (9), Lemma 2, Eqn. (20) and $\xi = 2^n/8q'_m$.

6 Conclusion and Future Works

In this paper, we have proposed a public permutation based nonce based MAC that offers beyond the birthday bound security in the multi-user setting. We have also instantiated our construction with Polyhash function. In this regard, one might wonder if it is possible to design a permutation based almost-xor-universal hash and use it as a hash instantiation of the construction. One easy approach is to take any block cipher based hash function (e.g., Hash function of PMAC [12] or LightMAC [38]) and replace each block cipher call with one round Even Mansour [33] cipher. This leads to having a $p\ell/2^n$ almost-xor-universal bound. By substituting the value of $p = 2^{2n/3}$, its axu advantage reduces to $O(2^{n/3})$. Even worse, composing this hash with nEHtM_p^* makes the security of the resultant construction to the birthday bound. Therefore, it is desired to design a permutation based hash function with 2^{-n} almost-xor-universal advantage. We believe that coming up with an efficient permutation based hash function with 2^{-n} axu advantage is a challenging problem and we leave it open for further research. We would like to mention here that the natural hardness of the problem comes from the fact that to achieve $2^{kn/k+1}$-axu advantage, one needs to require k invocations of independent permutations and k independent keys. We also believe that unlike this generic independent result of permutation based hash function, it is possible to come up with a dedicated nonce based MAC (similar to permutation variant of LightMAC [38]) and prove its graceful degradation of beyond birthday bound security in faulty nonce model.

References

[1] László Babai. The fourier transform and equations over finite abelian groups (lecture notes, version 1.3). 2002.

[2] Mihir Bellare, Daniel J. Bernstein, and Stefano Tessaro. Hash-function based prfs: AMAC and its multi-user security. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 566–595. Springer, 2016.

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

[4] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 394–403. IEEE Computer Society, 1997.

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

[6] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 299–320, 2017.

[7] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 313–314, 2013.

[8] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Cryptol. ePrint Arch.*, 2016:1188, 2016.

[9] Arghya Bhattarcharjee, Avijit Dutta, Eik List, and Mridul Nandi. CENCPP - beyond-birthday-secure encryption from public permutations. *IACR Cryptol. ePrint Arch.*, 2020:602, 2020.

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

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

[12] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397. Springer, 2002.

[13] Hanno Böck, Aaron Zauner, Sean Devlin, Juraj Somorovsky, and Philipp Jovanovic. Nonce-disrespecting adversaries: Practical forgery attacks on GCM in TLS. In *10th USENIX Workshop on Offensive Technologies, WOOT 16, Austin, TX, USA, August 8-9, 2016.*, 2016.

[14] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. SPONGENT: the design space of lightweight cryptographic hashing. *IEEE Trans. Computers*, 62(10):2041–2053, 2013.

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

[16] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.

[17] Avik Chakraborti, Mridul Nandi, Suprita Talnikar, and Kan Yasuda. On the composition of single-keyed tweakable even-mansour for achieving BBB security. *IACR Trans. Symmetric Cryptol.*, 2020(2):1–39, 2020.

[18] Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the security of sponge-type authenticated encryption modes. *IACR Trans. Symmetric Cryptol.*, 2020(2):93–119, 2020.

[19] Sanjit Chatterjee, Alfred Menezes, and Palash Sarkar. Another look at tightness. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 293–319. Springer, 2011.

[20] Yu Long Chen, Eran Lambooij, and Bart Mennink. How to build pseudorandom functions from public random permutations. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, pages 266–293, 2019.

[21] Wonseok Choi, ByeongHak Lee, Yeongmin Lee, and Jooyoung Lee. Improved security analysis for nonce-based enhanced hash-then-mask macs. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 697–723. Springer, 2020.

[22] Benoît Cogliati and Yannick Seurin. EWCDM: an efficient, beyond-birthday secure, nonce-misuse resistant MAC. In *CRYPTO 2016, Proceedings, Part I*, pages 121–149, 2016.

[23] Benoît Cogliati and Yannick Seurin. Analysis of the single-permutation encrypted davies-meyer construction. *Des. Codes Cryptogr.*, 86(12):2703–2723, 2018.

[24] Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-state keyed duplex with built-in multi-user support. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 606–637. Springer, 2017.

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

[26] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based MAC. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference,*

Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I, pages 631–661, 2018.

[27] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based mac. Cryptology ePrint Archive, Report 2018/500, 2018.

[28] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. sfdwcdm+: A BBB secure nonce based MAC. *Adv. Math. Commun.*, 13(4):705–732, 2019.

[29] Avijit Dutta, Ashwin Jha, and Mridul Nandi. Tight security analysis of ehtm MAC. *IACR Trans. Symmetric Cryptol.*, 2017(3):130–150, 2017.

[30] Avijit Dutta and Mridul Nandi. BBB secure nonce based MAC using public permutations. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, volume 12174 of *Lecture Notes in Computer Science*, pages 172–191. Springer, 2020.

[31] Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Beyond birthday bound secure MAC in faulty nonce model. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, pages 437–466, 2019.

[32] Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Permutation based edm: An inverse free bbb secure prf. Cryptology ePrint Archive, Report 2021/679, 2021.

[33] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptology*, 10(3):151–162, 1997.

[34] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 222–239, 2011.

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

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

[37] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 575–605. Springer, 2017.

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

[39] Kazuhiko Minematsu. How to thwart birthday attacks against macs via small randomness. In *Fast Software Encryption, FSE 2010*, pages 230–249, 2010.

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

[41] Andrew Morgan, Rafael Pass, and Elaine Shi. On the adaptive security of macs and prfs. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 724–753. Springer, 2020.

[42] Nicky Mouha and Atul Luykx. Multi-key security: The even-mansour construction revisited. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2015.

[43] Mridul Nandi. Mind the composition: Birthday bound attacks on EWCDMD and sokac21. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 203–220. Springer, 2020.

[44] Mridul Nandi and Avradip Mandal. Improved security analysis of PMAC. *J. Math. Cryptol.*, 2(2):149–162, 2008.

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

[46] John P. Steinberger. Counting solutions to additive equations in random sets. *CoRR*, abs/1309.5582, 2013.

[47] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[48] Yaobin Shen; Lei Wang; Jian Weng. Revisiting the security of dbhts macs: Beyond-birthday-bound in the multi-user setting. Cryptology ePrint Archive, Report 2020/1523, 2020.

Supplementary Materials

A Univariate Affine Non-Equations For (1), (2) and (3)

A.1 Univariate Affine Non-Equations For (1)

(a) $\nu_i \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_i, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_j \oplus k \oplus \mathsf{H}_j$

(b) $\nu_i \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_j, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_i \oplus k \oplus \mathsf{H}_i$ (symmetric to (a))

(c) $\nu_i \oplus k \oplus \mathsf{H}_i = \hat{x}^1_\alpha, \nu'_a = \nu_j, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_i \oplus k \oplus \mathsf{H}_i$

(d) $\nu_i \oplus k \oplus \mathsf{H}_i = \hat{x}^1_\alpha, \nu'_a = \nu_i, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_j \oplus k \oplus \mathsf{H}_j$ (symmetric to (d))

Note that the above conditions give rise to the following non-equations:

(a) $\pi(1\|v_j) \neq t'_a \oplus y^0_\alpha, \pi(0\|u_j) \neq t'_a \oplus y^0_\alpha \oplus t_j$

(b) $\pi(1\|v_j) \neq t'_a \oplus y^0_\alpha \oplus t_i \oplus t_j, \pi(0\|u_j) \neq t'_a \oplus y^0_\alpha \oplus t_i$

(c) $\pi(1\|v_j) \neq t'_a \oplus y^1_\alpha \oplus t_j, \pi(0\|u_j) \neq t'_a \oplus y^1_\alpha$

(d) $\pi(1\|v_j) \neq t'_a \oplus y^1_\alpha \oplus t_i, \pi(0\|u_j) \neq t'_a \oplus y^1_\alpha \oplus t_i \oplus t_j$

A.2 Univariate Affine Non-Equations For (2)

(a) $\nu_\ell \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_\ell, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1}$

(b) $\nu_\ell \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_{i_1} = \nu_{i_2} = \ldots = \nu_{i_p}, \nu'_a \oplus \mathsf{H}'_a = \nu_\ell \oplus \mathsf{H}_\ell$

(c) $\nu_\ell \oplus k \oplus \mathsf{H}_\ell = \hat{x}^1_\alpha, \nu'_a = \nu_\ell, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1}$

(d) $\nu_\ell \oplus k \oplus \mathsf{H}_\ell = \hat{x}^1_\alpha, \nu'_a = \nu_{i_1} = \nu_{i_2} = \ldots = \nu_{i_p}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_\ell \oplus k \oplus \mathsf{H}_\ell$

(e) $\nu_{i_1} \oplus k = \nu_{i_2} \oplus k = \ldots = \nu_{i_p} \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_\ell, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1}$

(f) $\nu_{i_1} \oplus k = \nu_{i_2} \oplus k = \ldots = \nu_{i_p} \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_{i_1} = \nu_{i_2} = \ldots = \nu_{i_p}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_\ell \oplus k \oplus \mathsf{H}_\ell$

Note that the above conditions give rise to the following non-equations:

(a) $\pi(1\|v_{i_1}) \neq t'_a \oplus y^0_\alpha, \pi(0\|u_{i_1}) = \pi(0\|u_{i_2}) = \ldots = \pi(0\|u_{i_p}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1}, \pi(1\|u_{i_j}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1} \oplus t_{i_j}, \forall j \in \{2, \ldots, p\}$

(b) $\pi(0\|u'_a) = \pi(0\|u_{i_1}) = \pi(0\|u_{i_2}) = \ldots = \pi(0\|u_{i_p}) \neq t'_a \oplus y^0_\alpha \oplus t_\ell, \pi(1\|u_{i_j}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_j} \oplus t_\ell, \forall j \in \{1, \ldots, p\}$

(c) $\pi(1\|v_{i_1}) \neq t'_a \oplus y^1_\alpha \oplus t_\ell, \pi(0\|u_{i_1}) = \pi(0\|u_{i_2}) = \ldots = \pi(0\|u_{i_p}) \neq t'_a \oplus y^1_\alpha \oplus t_\ell \oplus t_{i_1}, \pi(1\|u_{i_j}) \neq t'_a \oplus y^1_\alpha \oplus t_{i_1} \oplus t_{i_j}, \forall j \in \{2, \ldots, p\}$

(d) $\pi(0\|u'_a) = \pi(0\|u_{i_1}) = \pi(0\|u_{i_2}) = \ldots = \pi(0\|u_{i_p}) \neq t'_a \oplus y^1_\alpha, \pi(1\|u_{i_j}) \neq t'_a \oplus y^1_\alpha \oplus t_{i_j}, \forall j \in \{1, \ldots, p\}$

(e) $\pi(0\|u_\ell) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1}, \pi(1\|v_\ell) \neq t'_a \oplus y^0_\alpha \oplus t_\ell \oplus t_{i_1}$

(f) $\pi(0\|u_\ell) \neq t'_a \oplus y^0_\alpha \oplus t_\ell, \pi(1\|v_\ell) \neq t'_a \oplus y^0_\alpha$

A.3 Univariate Affine Non-Equations For (3)

(a) $\nu_{i_1} \oplus k = \nu_{i_2} \oplus k = \ldots = \nu_{i_p} \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_{i_1} = \ldots = \nu_{i_p}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1}$

(b) $\nu_{i_1} \oplus k = \nu_{i_2} \oplus k = \ldots = \nu_{i_p} \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_{j_1} = \ldots = \nu_{j_s}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1}$

(c) $\nu_{j_1} \oplus k = \nu_{j_2} \oplus k = \ldots = \nu_{j_s} \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_{i_1} = \ldots = \nu_{i_p}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1}$

(d) $\nu_{j_1} \oplus k = \nu_{j_2} \oplus k = \ldots = \nu_{j_s} \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_{j_1} = \ldots = \nu_{j_s}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1}$

(e) $\nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} = \nu_{i_2} \oplus k \oplus \mathsf{H}_{i_2} = \ldots = \nu_{i_p} \oplus k \oplus \mathsf{H}_{i_p} = \hat{x}^1_\alpha, \nu'_a = \nu_{j_1}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} = \ldots = \nu_{i_p} \oplus k \oplus \mathsf{H}_{i_p}$

(f) $\nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} = \nu_{i_2} \oplus k \oplus \mathsf{H}_{i_2} = \ldots = \nu_{i_p} \oplus k \oplus \mathsf{H}_{i_p} = \hat{x}^1_\alpha, \nu'_a = \nu_{i_1}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \ldots = \nu_{i_s} \oplus k \oplus \mathsf{H}_{i_s}$

(g) $\nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \nu_{j_2} \oplus k \oplus \mathsf{H}_{j_2} = \ldots = \nu_{j_s} \oplus k \oplus \mathsf{H}_{j_s} = \hat{x}^1_\alpha, \nu'_a = \nu_{j_1}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} = \ldots = \nu_{i_p} \oplus k \oplus \mathsf{H}_{i_p}$

(h) $\nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \nu_{j_2} \oplus k \oplus \mathsf{H}_{j_2} = \ldots = \nu_{j_s} \oplus k \oplus \mathsf{H}_{j_s} = \hat{x}^1_\alpha, \nu'_a = \nu_{i_1}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \ldots = \nu_{j_s} \oplus k \oplus \mathsf{H}_{j_s}$

(i) $\nu_{i_1} = \nu_{i_2} = \ldots = \nu_{i_p} = \hat{x}^0_\alpha, \nu'_a = \nu_{j_1}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1}$

(j) $\nu_{i_1} = \nu_{i_2} = \ldots = \nu_{i_p} = \hat{x}^0_\alpha, \nu'_a = \nu_{i_1}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \ldots = \nu_{j_s} \oplus k \oplus \mathsf{H}_{j_s}$

(k) $\nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \nu_{j_2} \oplus k \oplus \mathsf{H}_{j_2} = \ldots = \nu_{j_s} \oplus k \oplus \mathsf{H}_{j_s} = \hat{x}^1_\alpha, \nu'_a = \nu_{j_1}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1}$

(l) $\nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \nu_{j_2} \oplus k \oplus \mathsf{H}_{j_2} = \ldots = \nu_{j_s} \oplus k \oplus \mathsf{H}_{j_s} = \hat{x}^1_\alpha, \nu'_a = \nu_{i_1} = \ldots = \nu_{i_p}, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{j_1} \oplus k \oplus \mathsf{H}_{j_1} = \ldots = \nu_{j_s} \oplus k \oplus \mathsf{H}_{j_s}$

Note that the above conditions give rise to the following non-equations:

(a) $\pi(1\|v'_a) = \pi(1\|v_{j_1}) \neq t'_a \oplus y^0_\alpha, \pi(0\|u_{j_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{j_1}, \forall \ell \in \{1, \ldots, s\}, \pi(1\|v_{j_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{j_1} \oplus t_{j_\ell}, \forall \ell \in \{2, \ldots, s\}$

(b) $\pi(0\|u'_a) = \pi(0\|u_{j_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1}, \pi(1\|v_{j_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1} \oplus t_{j_\ell}, \forall \ell \in \{1, \ldots, s\}$

(c) $\pi(0\|u'_a) = \pi(0\|u_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1}, \pi(1\|v_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1} \oplus t_{i_\ell}, \forall \ell \in \{1, \ldots, p\}$

(d) $\pi(1\|v'_a) = \pi(1\|v_{i_1}) \neq t'_a \oplus y^0_\alpha, \pi(0\|u_{i_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1}, \forall \ell \in \{1, \ldots, p\}, \pi(1\|v_{i_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1} \oplus t_{i_\ell}, \forall \ell \in \{2, \ldots, p\}$

(e) $\pi(0\|u'_a) = \pi(0\|u_{j_1}) \neq t'_a \oplus y^1_\alpha, \pi(1\|v_{j_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1}, \forall \ell \in \{1, \ldots, s\}, \pi(0\|u_{j_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1} \oplus t_{j_\ell}, \forall \ell \in \{2, \ldots, s\}$

(f) $\pi(1\|v'_a) = \pi(1\|v_{j_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{i_1}, \pi(0\|u_{j_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{i_1} \oplus t_{j_\ell}, \forall \ell \in \{1, \ldots, s\}$

(g) $\pi(1\|v'_a) = \pi(1\|v_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1}, \pi(0\|u_{j_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1} \oplus t_{i_\ell}, \forall \ell \in \{1, \ldots, p\}$

(h) $\pi(0\|u'_a) = \pi(0\|u_{i_1}) \neq t'_a \oplus y^1_\alpha, \pi(1\|v_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{i_1}, \forall \ell \in \{1, \ldots, p\}, \pi(0\|u_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{i_1} \oplus t_{i_\ell}, \forall \ell \in \{2, \ldots, p\}$

(i) $\pi(0\|u'_a) = \pi(0\|u_{j_1}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1}, \pi(1\|v_{j_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1} \oplus t_{j_1}, \forall \ell \in \{1, \ldots, s\}, \pi(0\|u_{j_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1} \oplus t_{j_1} \oplus t_{j_\ell}, \forall \ell \in \{2, \ldots, s\}$

(j) $\pi(1\|v'_a) = \pi(1\|v_{j_\ell}) \neq t'_a \oplus y^0_\alpha, \pi(0\|u_{j_\ell}) \neq t'_a \oplus y^0_\alpha \oplus t_{j_\ell}, \forall \ell \in \{1, \ldots, s\}$

(k) $\pi(1\|v'_a) = \pi(1\|u_{i_1}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1}, \pi(0\|u_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1} \oplus t_{i_1}, \forall \ell \in \{1, \ldots, p\}, \pi(1\|v_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{j_1} \oplus t_{i_1} \oplus t_{i_\ell}, \forall \ell \in \{2, \ldots, p\}$

(l) $\pi(0\|u'_a) = \pi(0\|u_{i_\ell}) \neq t'_a \oplus y^1_\alpha, \pi(1\|u_{i_\ell}) \neq t'_a \oplus y^1_\alpha \oplus t_{i_\ell}, \forall \ell \in \{1, \ldots, p\}$

B Univariate Affine Non-Equations For (4), (5) and (6)

(a) $\nu_i \oplus k = \hat{x}^0_\alpha, \nu'_a = \nu_i \Rightarrow \pi(1\|v'_a) \neq t'_a \oplus y^0_\alpha$

(b) $\nu_i \oplus k = \hat{x}^0_\alpha, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_i \oplus k \oplus \mathsf{H}_i \Rightarrow \pi(0\|u'_a) \neq t'_a \oplus t_i \oplus y^0_\alpha$

(c) $\nu_i \oplus k \oplus \mathsf{H}_i = \hat{x}^1_\alpha, \nu'_a = \nu_i \Rightarrow \pi(1\|v'_a) \neq t'_a \oplus t_i \oplus y^1_\alpha$

(d) $\nu_i \oplus k \oplus \mathsf{H}_i = \hat{x}^1_\alpha, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_i \oplus k \oplus \mathsf{H}_i \Rightarrow \pi(0\|u'_a) \neq t'_a \oplus y^1_\alpha$

Note that the above conditions give rise to the following non-equations:

(a) $\nu_{i_1} = \nu_{i_2} = \ldots = \nu_{i_p} = \hat{x}^0_\alpha, \nu'_a = \nu_{i_1} \Rightarrow \pi(1\|v'_a) \neq t'_a \oplus y^0_\alpha$

(b) $\nu_{i_1} = \nu_{i_2} = \ldots = \nu_{i_p} = \hat{x}^0_\alpha, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} \Rightarrow \pi(0\|u'_a) \neq t'_a \oplus y^0_\alpha \oplus t_{i_1}$

(c) $\nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} = \nu_{i_2} \oplus k \oplus \mathsf{H}_{i_2} = \ldots = \nu_{i_p} \oplus k \oplus \mathsf{H}_{i_p} = \hat{x}^1_\alpha, \nu'_a = \nu_{i_1} \Rightarrow \pi(1\|v'_a) \neq t'_a \oplus y^1_\alpha \oplus t_{i_1}$

(d) $\nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} = \nu_{i_2} \oplus k \oplus \mathsf{H}_{i_2} = \ldots = \nu_{i_p} \oplus k \oplus \mathsf{H}_{i_p} = \hat{x}^1_\alpha, \nu'_a \oplus k \oplus \mathsf{H}'_a = \nu_{i_1} \oplus k \oplus \mathsf{H}_{i_1} \Rightarrow \pi(0\|u'_a) \neq t'_a \oplus y^1_\alpha$

C Proof of Lemma 2

In the following, we bound the probabilities of all the bad events individually. The lemma will follow by adding the individual bounds using the union bound.

Before that, we state the following basic result which we will repeatedly use in bounding the following bad events.

Proposition 1. *Let L_1 be a set of elements of $\{0,1\}^n$ having size q_1 and $\mathsf{L}_2, \mathsf{L}_3$ and L_4 be the multisets of $\{0,1\}^n$ such that the number of elements in L_i (includes repetition of elements) is q_i, for $i = 2,3,4$. Then, we have*

$$\mid \{(a,b,c,d) \in \mathsf{L}_1 \times \mathsf{L}_2 \times \mathsf{L}_3 \times \mathsf{L}_4 : a \oplus b \oplus c \oplus d = 0\} \mid \leq \prod_{i=2}^{4} q_i.$$

Proof. For each choice of elements $(b,c,d) \in \mathsf{L}_2 \oplus \mathsf{L}_3 \oplus \mathsf{L}_3$, there can be at most one $a \in \mathsf{L}_1$ such that $a = b \oplus c \oplus d$. As the number of choices of triplets (b,c,d) is $q_2 q_3 q_4$, the result follows. □

Bounding B.1. For any possible MAC query $(u_i, \nu_i, m_i, t_i) \in \tau_m$ and a pair of any possible primitive queries $(\hat{x}^0, \hat{y}^0) \in \tau_p^{(0)}$ and $(\hat{x}^1, \hat{y}^1) \in \tau_p^{(1)}$, we rely on the randomness of k^{u_i} in the equation $\nu_i \oplus k^{u_i} = \hat{x}^0_j$, and on the randomness of the hash key $k_h^{u_i}$ in the equation $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1_\ell$. In the ideal world, k^{u_i} and $k_h^{u_i}$ are dummy keys sampled uniformly and independently from their respective space. Therefore, for a fixed choice of $i, (\hat{x}^0, \hat{y}^0)$ and $(\hat{x}^1, \hat{y}^1)$, the probability of the event is $\epsilon/2^{n-1}$. Summing over all possible choices of $i, (\hat{x}^0, \hat{y}^0)$ and $(\hat{x}^1, \hat{y}^1)$, we have

$$\Pr[\mathsf{B.1}] \leq \frac{2p^2 q_m \epsilon}{2^n}. \tag{21}$$

Bounding B.2. For this bad event, we need to consider the following cases, namely when (a) $u_i = u_j$ (the i-th and the j-th query are made to the same user oracle), and when (b) $u_i \neq u_j$ (the i-th and the j-th query are made to the different user oracles).

1. For case (a), since the i-th and the j-th user are the same, we have $k^{u_i} = k^{u_j}$, and the first equation becomes $\nu_i = \nu_j$. Let $\mathcal{N}$ be the set of all MAC query indices pairs (i, j) such that $\nu_i = \nu_j$. Event $\mathsf{B.2}$ occurs if $\nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j} = \nu_\ell \oplus k^{u_\ell} \oplus \mathsf{H}_\ell^{u_\ell}$ for some $\ell \neq j$. For any such fixed i, j, ℓ, the probability of the event is at most ϵ (using the randomness of the hash key $k_h^{u_j}$ (or $k_h^{u_\ell}$)). The number of such choices of (i, j, ℓ) is at most $(\eta + 1)^2$. Hence,
$$\Pr[\mathsf{B.2}(a)] \leq \eta^2 \epsilon. \tag{22}$$

2. For case (b), the keys k^{u_i} and k^{u_j} are generated independently of each other. For MAC queries $(u_i, \nu_i, m_i, t_i) \neq (u_j, \nu_j, m_j, t_j) \in \tau_m$ and $(u_j, \nu_j, m_j, t_j) \neq (u_\ell, \nu_\ell, m_\ell, t_\ell) \in \tau_m$ such that $u_i \neq u_j$, we rely on the randomness of k^{u_i} (or k^{u_j}) in the equation $\nu_i \oplus k^{u_i} = \nu_j \oplus k^{u_j}$, and on the randomness of $k_h^{u_j}$ (or $k_h^{u_\ell}$) in the equation $\nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j} = \nu_\ell \oplus k^{u_\ell} \oplus \mathsf{H}_\ell^{u_\ell}$. In the ideal world, k^{u_i} (or k^{u_j}) and $k_h^{u_j}$ (or $k_h^{u_\ell}$) are dummy keys sampled uniformly and independently from their respective space. Therefore, for a fixed choice of (i, j, ℓ), the probability of the event is at most $\epsilon/2^{n-1}$. The number of such choices of (i, j, ℓ) is at most q_m^3. Hence,
$$\Pr[\mathsf{B.2}(b)] \leq \frac{2q_m^3 \epsilon}{2^n}. \tag{23}$$

Putting (22) and (23) together, we have
$$\Pr[\mathsf{B.2}] \leq \eta^2 \epsilon + \frac{2q_m^3 \epsilon}{2^n}. \tag{24}$$

Bounding B.3. For any two MAC queries $(u_i, \nu_i, m_i, t_i) \neq (u_j, \nu_j, m_j, t_j) \in \tau_m$ and a primitive query $(\hat{x}^0, \hat{y}^0) \in \tau_p^{(0)}$, we rely on the randomness of k^{u_i} in the equation $\nu_i \oplus k^{u_i} = \hat{x}_\ell^1$, and on the randomness of the hash key $k_h^{u_i}$ (or $k_h^{u_j}$) in the equation $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j}$. In the ideal world, k^{u_i} and $k_h^{u_i}$ (or $k_h^{u_j}$) are dummy keys, sampled uniformly and independently from their respective space. Therefore, for a fixed choice of i, j and $(\hat{x}^0, \hat{y}^0)$, the probability of the event is $\epsilon/2^{n-1}$. Summing over all possible choices of i, j and $(\hat{x}^0, \hat{y}^0)$ we have
$$\Pr[\mathsf{B.3}] \leq \frac{pq_m^2 \epsilon}{2^n}. \tag{25}$$

Bounding B.4. For this bad event, we need to consider the following cases, namely when (a) $u_i = u_j$ (the i-th and the j-th query are made to the same user oracle), and when (b) $u_i \neq u_j$ (the i-th and the j-th query are made to the different user oracles).

1. For case (a), since the i-th and the j-th user are the same, we have $k^{u_i} = k^{u_j}$, and the first equation becomes $\nu_i = \nu_j$. For any two MAC queries $(u_i, \nu_i, m_i, t_i) \neq (u_j, \nu_j, m_j, t_j) \in \tau_m$ and a primitive query $(\hat{x}^1, \hat{y}^1) \in \tau_p^{(1)}$, the randomness in the equation $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1$ is $k_h^{u_i}$. Therefore, for a fixed choice of i, j and $(\hat{x}^1, \hat{y}^1)$, the probability of the event is ϵ. The number of choices of $i \neq j \in [q_m]$ such that $\nu_i = \nu_j$ is at most 2η and the number of choices of $(\hat{x}^1, \hat{y}^1)$ is at most p. Hence
$$\Pr[\mathsf{B.4}(a)] \leq 2\eta p \epsilon. \tag{26}$$

2. For case (b), the keys k^{u_i} and k^{u_j} are generated independently of each other. For any two MAC queries $(u_i, \nu_i, m_i, t_i) \neq (u_j, \nu_j, m_j, t_j) \in \tau_m$ such that $u_i \neq u_j$, and a primitive query $(\hat{x}^1, \hat{y}^1) \in \tau_p^{(1)}$, we rely on the randomness of k^{u_i} (or k^{u_j}) in the equation $\nu_i \oplus k^{u_i} = \nu_j \oplus k^{u_j}$, and on the randomness of $k_h^{u_i}$ in the equation $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \hat{x}^1$. In the ideal world, k^{u_i} (or k^{u_j}) and $k_h^{u_i}$ are dummy keys,

sampled uniformly and independently from their respective space. Therefore, for a fixed choice of i, j and $(\hat{x}^1, \hat{y}^1)$, the probability of the event is $\epsilon/2^{n-1}$. The number of choices of i, j and $(\hat{x}^1, \hat{y}^1)$ is at most pq_m^2. Hence,

$$\Pr[\mathsf{B.4}(b)] \leq \frac{2pq_m^2\epsilon}{2^n}. \tag{27}$$

Putting (26) and (27) together, we have

$$\Pr[\mathsf{B.4}] \leq 2\eta p\epsilon + \frac{2pq_m^2\epsilon}{2^n}. \tag{28}$$

Bounding B.5. For this bad event, we need to consider the following cases, namely when (a) $u_i = u_j$ (the i-th and the j-th query are made to the same user oracle), and when (b) $u_i \neq u_j$ (the i-th and the j-th query are made to the different user oracles).

1. For case (a), since the i-th and the j-th user are the same, we have $k^{u_i} = k^{u_j}$, and the first equation becomes $\nu_i = \nu_j$. For a fixed choice of indices i and j, the probability of the event $t_i = t_j$ is at most $1/2^n$. Number of choices of i and j such that $\nu_i = \nu_j$ is at most 2η. Summing over all possible choices of i and j, we have

$$\Pr[\mathsf{B.5}(a)] \leq \frac{2\eta}{2^n}. \tag{29}$$

2. For case (b), the keys k^{u_i} and k^{u_j} are generated independently of each other. For any two MAC queries $(u_i, \nu_i, m_i, t_i) \neq (u_j, \nu_j, m_j, t_j) \in \tau_m$ such that $u_i \neq u_j$, the probability that the event happens for a fixed choice of indices i and j is at most $2/2^{2n}$, as $\nu_i \oplus k^{u_i} = \nu_j \oplus k^{u_j}$ is independent of $t_i = t_j$. The number of choices of (i, j) is at most $q_m^2/2$. Hence,

$$\Pr[\mathsf{B.5}(b)] \leq \frac{q_m^2}{2^{2n}}. \tag{30}$$

Putting (29) and (30) together, we have

$$\Pr[\mathsf{B.5}] \leq \frac{2\eta}{2^n} + \frac{q_m^2}{2^{2n}}. \tag{31}$$

Bounding B.6. Similar to B.5(b), for a fixed choice of indices i and j, the probability that the event happens is at most $\epsilon/2^n$, as the event $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu_j \oplus k^{u_j} \oplus \mathsf{H}_j^{u_j}$ is independent over $t_i = t_j$. The number of choices of (i, j) is at most $q_m^2/2$. Hence,

$$\Pr[\mathsf{B.6}] \leq \frac{q_m^2\epsilon}{2^{n+1}}. \tag{32}$$

Bounding B.7. To deal with this event, we define the following event:

$$\mathsf{E}_0 \overset{\Delta}{=} \#\{(t_i, \hat{y}^0, \delta) \in \mathcal{T} \times \mathcal{Y}_0 \times (\mathcal{Y}_0 \cup \mathcal{Y}_1) : t_i \oplus \hat{y}^0 = \delta\} \geq 2p^2q_m/2^n + p\sqrt{6nq_m}.$$

Note that, the cardinality of $\mathcal{Y}_0 \cup \mathcal{Y}_1$ is $2p$ and the event E_0 is bounded using Lemma 1, where we take $\mathcal{A} = \mathcal{Y}_0$ and $\mathcal{B} = \mathcal{Y}_0 \cup \mathcal{Y}_1$. Therefore, from Lemma 1, we have $\Pr[\mathsf{E}_0] \leq 2/2^n$. Now, we write the probability of the event B.7 as follows:

$$\Pr[\mathsf{B.7}] \leq \Pr[\mathsf{B.7} \mid \overline{\mathsf{E}}_0] + \frac{2}{2^n}.$$

Now, it remains to bound the first term of the above equation. Note that, for a fixed choice of indices $i, (\hat{x}^0, \hat{y}^0)$, and δ such that

$$\nu_i \oplus k^{u_i} = \hat{x}^0, t_i \oplus \hat{y}^0 = \delta$$

holds with probability at most $2/2^n$, using the randomness of k^{u_i}. However, the number of choices of $i, (\hat{x}^0, \hat{y}^0)$, and δ is restricted to at most $2p^2 q_m/2^n + p\sqrt{6nq_m}$. Therefore, by summing over all possible choices of $i, (\hat{x}^0, \hat{y}^0)$, and δ, we have

$$\Pr[\mathsf{B.7}] \leq \frac{4p^2 q_m}{2^{2n}} + \frac{2p\sqrt{6nq_m}}{2^n} + \frac{2}{2^n}. \tag{33}$$

Bounding B.8. Bounding B.8 is identical to that of B.7. As before, where we define

$$\mathsf{E}_1 \overset{\Delta}{=} \#\{(t_i, \hat{y}^1, \delta) \in \mathcal{T} \times \mathcal{Y}_1 \times (\mathcal{Y}_0 \cup \mathcal{Y}_1) : t_i \oplus \hat{y}^1 = \delta\} \geq 2p^2 q_m/2^n + p\sqrt{6nq_m},$$

and using Lemma 1, we bound the probability of the event E_1 to $2/2^n$. As a result, using the similar manner, we have

$$\Pr[\mathsf{B.8}] \leq \Pr[\mathsf{B.8} \mid \overline{\mathsf{E}}_1] + \frac{2}{2^n} \leq \frac{2p^2 q_m \epsilon}{2^n} + p\epsilon\sqrt{6nq_m} + \frac{2}{2^n}. \tag{34}$$

Bounding B.9. For a fixed choice of indices, the event can be expressed as the following three linear equations:

$$\begin{cases} k^{u_i} = \nu_i \oplus \hat{x}^0 \\ k^{u_j} = \nu_j \oplus \hat{x}'^0 \\ t_i \oplus \hat{y}^0 = t_j \oplus \hat{y}'^0. \end{cases}$$

We bound this event under the following cases: (a) when one of the MAC queries (i, j) appears after the primitive queries $(\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)$, and (b) when one of the primitive queries $(\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)$ appears after the MAC queries (i, j).

1. For case (a), assume without loss of generality that $i > j$, we first consider the subcase when $k^{u_i} = k^{u_j}$ (the i-th and the j-th query are made to the same user oracle). Then till the point of making the i-th query but before observing the response, we can condition all the random variables obtained so far and thus define the following set:

   $$\mathcal{I}_1 = \{(i, j, (\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)) \mid i > j \wedge (\hat{x}^0, \hat{y}^0) \neq (\hat{x}'^0, \hat{y}'^0) \wedge \nu_i \oplus \nu_j = \hat{x}^0 \oplus \hat{x}'^0\}.$$

   Note that $|\mathcal{I}_1| \leq q_m^2 p$ due to Proposition 1, as we can freely choose (i, j) in at most q_m^2 ways and $(\hat{x}^0, \hat{y}^0)$ in at most p ways. This choice of i, j and $(\hat{x}^0, \hat{y}^0)$ uniquely determines the choice of $(\hat{x}'^0, \hat{y}'^0)$ to be at most 1. Now, to bound the probability of the event B.9, it is enough to bound the probability of the following event:

   $$\exists (i, j, (\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)) \in \mathcal{I}_1 : k^{u_i} = \nu_i \oplus \hat{x}^0, t_i = t_j \oplus \hat{y}^0 \oplus \hat{y}'^0.$$

   For a fixed choice of such indices, the probability that the above event holds is $2/2^{2n}$, using the randomness of k^{u_i} and t_i. Therefore, by varying over all possible such of indices of $\mathcal{I}_1$, this subcase happens with probability at most $2q_m^2 p/2^{2n}$.

   However, for the subcase when $k^{u_i} \neq k^{u_j}$ (the i-th and the j-th query are made to the different user oracles), the probability that the event holds for a fixed choice of indices is $4/2^{3n}$. Since the randomness of the first equation comes from k^{u_i}, the randomness of the second equation comes from k^{u_j}, and the randomness of the third equations comes from t_i. Therefore, by varying over all possible indices, this subcase happens with probability at most $4q_m^2 p^2/2^{3n}$.

   Hence for case (a), we have

   $$\Pr[\mathsf{B.9}(a)] \leq \frac{2q_m^2 p}{2^{2n}} + \frac{4q_m^2 p^2}{2^{3n}}. \tag{35}$$

2. For case (b), assume without loss of generality that $(\hat{x}^0, \hat{y}^0)$ appears after $(\hat{x}'^0, \hat{y}'^0)$. We first consider the case when $(\hat{x}^0, \hat{y}^0)$ is the forward query. The analysis is the same as for case (a), but here we use the randomness $\hat{y}^0$ for our third equation. Therefore, we have

$$\Pr[\mathsf{B.9}(b, \mathsf{forward})] \leq \frac{4q_m^2 p}{2^{2n}} + \frac{6q_m^2 p^2}{2^{3n}}. \tag{36}$$

However, if the primitive query $(\hat{x}^0, \hat{y}^0)$ is backward, then till the point of making the backward primitive query $(\hat{x}^0, \hat{y}^0)$ but before observing the response, we can condition all the random variables obtained so far and thus define the following set:

$$\mathcal{I}_2 = \{(i, j, (\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)) \mid i \neq j \wedge (\hat{x}^0, \hat{y}^0) \neq (\hat{x}'^0, \hat{y}'^0) \wedge t_i \oplus t_j = \hat{y}^0 \oplus \hat{y}'^0\}.$$

Note that $|\mathcal{I}_2| \leq q_m^2 p$ due to Proposition 1, as the number of choices for (i, j) is at most q_m^2 and the number of choices for $(\hat{x}'^0, \hat{y}'^0)$ is at most p. Then, the number of choices for $(\hat{x}^0, \hat{y}^0)$ is 1 as it has to satisfy $t_j = \hat{y}^0 \oplus \hat{y}'^0$. Now, to bound the probability of the event $\mathsf{B.9}$, it is enough to bound the probability of the following event:

$$\exists (i, j, (\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)) \in \mathcal{I}_2 : \hat{x}^0_\alpha = \nu_i \oplus k, k = \nu_j \oplus \hat{x}^0_\beta.$$

For a fixed choice of such indices, the probability that the above event holds is $2/2^{2n}$, using the independent randomness of k^{u_j} and $\hat{x}^0$ (Note that for this particular subcase, we can also use the independent randomness of k^{u_i} and k^{u_j} when $u_i \neq u_j$). Therefore, by varying over all possible such of indices of $\mathcal{I}_2$, we have

$$\Pr[\mathsf{B.9}(b, \mathsf{backward})] \leq \frac{2q_m^2 p}{2^{2n}}. \tag{37}$$

Therefore, by combining Eqn. (35), Eqn. (36) and Eqn. (37), we have

$$\Pr[\mathsf{B.9}] \leq \frac{4q_m^2 p}{2^{2n}} + \frac{6q_m^2 p^2}{2^{3n}}. \tag{38}$$

Bounding B.10. For a fixed choice of indices, the event can be expressed as the following three linear equations:

$$\begin{cases} \mathsf{H}_i^{u_i} \oplus k^{u_i} = \nu_i \oplus \hat{x}^1 \\ \mathsf{H}_j^{u_j} \oplus k^{u_j} = \nu_j \oplus \hat{x}'^1 \\ t_i \oplus \hat{y}^1 = t_j \oplus \hat{y}'^1. \end{cases}$$

We bound this event in a similar way as we bounded $\mathsf{B.9}$. We again split the analysis in two cases: (a) when one of the MAC queries (i, j) appears after the primitive queries $(\hat{x}^1, \hat{y}^1), (\hat{x}'^1, \hat{y}'^1)$, and (b) when one of the primitive queries $(\hat{x}^1, \hat{y}^1), (\hat{x}'^1, \hat{y}'^1)$ appears after the MAC queries (i, j). The analysis of these cases are exactly the same as that of $\mathsf{B.9}$, except that we now rely on the randomness of the hash keys $h_h^{u_i}$ and $h_h^{u_j}$ in the first two equations, instead of h^{u_i} and h^{u_j} (ϵ instead of $2/2^n$). Therefore, we have

$$\Pr[\mathsf{B.10}] \leq \frac{2q_m^2 p \epsilon}{2^n} + \frac{4q_m^2 p^2 \epsilon^2}{2^n}. \tag{39}$$

However, for the particular subcase when $u_i = u_j$ and $m_i \neq m_j$, the variables in the left hand side of the first two equations are not supposed to be identical. Hence it differs slightly from the rest of the analysis, that is the main reason for the modification in the construction.

1. For case (a), assume without loss of generality that $i > j$, Till the point of making the i-th query but before observing the response, we can condition all the random variables obtained so far. Now, for a fixed choice of indices, the probability of the equations hold is $2\epsilon/2^{2n}$. Since the randomness of the first equation comes from the hash key $k_h^{u_i}$, the randomness of the second equation comes from k^{u_i}, and the randomness of the third equations comes from t_i. Therefore, this particular subcase happens with probability at most $2q_m^2 p^2 \epsilon/2^{2n}$.

2. For case (b), assume without loss of generality that $(\hat{x}^0, \hat{y}^0)$ appears after $(\hat{x}'^0, \hat{y}'^0)$. We only consider the case that when $(\hat{x}^0, \hat{y}^0)$ is the forward query, since the case of backward query is identical to the analysis of $\mathsf{B.10}$. As done for case (a), here we use the randomness $\hat{y}^0$ for our third equation. Therefore, this particular subcase happens with probability at most $4q_m^2 p^2 \epsilon/2^{2n}$.

Therefore, by combining Eqn. (39) with the subcases when $u_i = u_j$ and $m_i \neq m_j$, we have

$$\Pr[\mathsf{B.10}] \leq \frac{2q_m^2 p\epsilon}{2^n} + \frac{4q_m^2 p^2 \epsilon^2}{2^n}. \tag{40}$$

Bounding B.11. This event can be bounded in the similarly way as $\mathsf{B.9}$ or $\mathsf{B.10}$. For a fixed choice of indices, the event can be expressed as the following three linear equations:

$$\begin{cases} k^{u_i} = \nu_i \oplus \hat{x}^0 \\ \mathsf{H}_j^{u_j} \oplus k^{u_j} = \nu_j \oplus \hat{x}^1 \\ t_i \oplus \hat{y}^0 = t_j \oplus \hat{y}^1. \end{cases}$$

Using the similar analysis, we bound the event (a) when one of the MAC queries (i, j) appears after the primitive queries $(\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)$, and (b) when one of the primitive queries $(\hat{x}^0, \hat{y}^0), (\hat{x}'^0, \hat{y}'^0)$ appears after the MAC queries (i, j).

1. For case (a), assume without loss of generality that $i > j$. Till the point of making the i-th query but before observing the response, we can condition all the random variables obtained so far. Then, for a fixed choice of indices, we bound the probability of the event to $2\epsilon/2^{2n}$ as the randomness of the first equation comes from k^{u_i}, the randomness of the second equation comes from the hash key $k_h^{u_i}$, and the randomness of the third equations comes from t_i. Moreover, the number of possible choices of $i, j, (\hat{x}^0, \hat{y}^0)$, and $(\hat{x}^1, \hat{y}^1)$ is at most $q_m^2 p^2$. Hence

$$\Pr[\mathsf{B.11}(a)] \leq \frac{2q_m^2 p^2 \epsilon}{2^{2n}}. \tag{41}$$

2. For case (b), assume without loss of generality that $(\hat{x}^0, \hat{y}^0)$ appears after $(\hat{x}^1, \hat{y}^1)$. We first consider the case that when $(\hat{x}^0, \hat{y}^0)$ is the forward query. The analysis is the same as for case (a), but here the randomness of the third equations comes from $\hat{y}^0$. Hence

$$\Pr[\mathsf{B.11}(b, \mathsf{forward})] \leq \frac{4q_m^2 p^2 \epsilon}{2^{2n}}. \tag{42}$$

The analysis of subcase when the primitive query $(\hat{x}^0, \hat{y}^0)$ is a backward query can be performed in the similarly way as that of $\mathsf{B.9}$ or $\mathsf{B.10}$. Hence, we have

$$\Pr[\mathsf{B.11}(b, \mathsf{backward})] \leq \frac{2q_m^2 p\epsilon}{2^n}. \tag{43}$$

Therefore, by combining Eqn. (41), Eqn. (42) and Eqn. (43), we have

$$\Pr[\mathsf{B.11}] \leq \frac{4q_m^2 p^2 \epsilon}{2^{2n}} + \frac{2q_m^2 p \epsilon}{2^n}. \tag{44}$$

Bounding B.12. For any possible verification query $(u'_a, \nu'_a, m'_a, t'_a) \in \tau_v$ and a pair of any possible primitive queries $(\hat{x}^0, \hat{y}^0) \in \tau_p^{(0)}$ and $(\hat{x}^1, \hat{y}^1) \in \tau_p^{(1)}$, we rely on the randomness of $k^{u'_a}$ in the equation $\nu'_a \oplus k^{u'_a} = \hat{x}^0$, and on the randomness of the hash key $k_h^{u'_a}$ in the equation $\nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a = \hat{x}^1$. In the ideal world, $k^{u'_a}$ and $k_h^{u'_a}$ are dummy keys, sampled uniformly and independently from their respective spaces. Therefore, for a fixed choice of $a, (\hat{x}^0, \hat{y}^0)$ and $(\hat{x}^1, \hat{y}^1)$, the probability of the event is $\epsilon/2^{n-1}$. The number of choices of a is q_v, and the choice for $(x_j^0, y_j^0) \in \tau_p^{(0)}$ and $(\hat{x}^1, \hat{y}^1) \in \tau_p^{(1)}$ is p^2. Hence,

$$\Pr[\mathsf{B.12}] \leq \frac{2q_v p^2 \epsilon}{2^n}. \tag{45}$$

Bounding B.13. For this bad event, we need to consider the following cases, namely when (a) $u_i = u'_a$ (the i-th MAC query and the a-th verification query are made to the same), and when (b) $u_i \neq u'_a$ (the i-th MAC query and the a-th verification query are made to the different users).

1. For case (a), we have $k^{u_i} = k^{u'_a}$, and the first equation becomes $\nu_i = \nu'_a$. For some $a \in [q_v]$ and $i \in [q_m]$, if $\nu_i = \nu'_a$, $\nu_i \oplus \mathsf{H}_i^{u_i} = \nu'_a \oplus \mathsf{H}'^{u'_a}_a$ and $t_i = t'_a$, then we must have $m_i \neq m'_a$ (as the distinguisher is non-trivial). Now, for a fixed choice of a, the number of choices for i is at most 1 due to $\overline{\mathsf{B.5}}$. Therefore, for such choice of (i, a), the probability that $\nu_i \oplus \mathsf{H}_i^{u_i} = \nu'_a \oplus \mathsf{H}'^{u'_a}_a$ holds is ϵ. The number of choices of a is at most q_v. Hence,
$$\Pr[\mathsf{B.13(a)}] \leq q_v \epsilon. \tag{46}$$

2. For case (b), the keys k^{u_i} and $k^{u'_a}$ are generated independently of each other. For any MAC query $(u_i, \nu_i, m_i, t_i) \in \tau_m$ and verification query $(u'_a, \nu'_a, m'_a, t'_a) \in \tau_v$ such that $u_i \neq u'_a$, we rely on the randomness of k^{u_i} (or $k^{u'_a}$) in the equation $\nu_i \oplus k^{u_i} = \nu'_a \oplus k^{u'_a}$, and on the randomness of the hash key $k_h^{u_i}$ (or $k_h^{u'_a}$) in the equation $\nu_i \oplus k^{u_i} \oplus \mathsf{H}_i^{u_i} = \nu'_a \oplus k^{u'_a} \oplus \mathsf{H}'^{u'_a}_a$. In the ideal world, k^{u_i} (or $k^{u'_a}$) and $k_h^{u_i}$ (or $k_h^{u'_a}$) are dummy keys, sampled uniformly and independently from their respective space. Therefore, for a fixed choice of (i, a), the probability that the event happens is $\epsilon/2^{n-1}$. The number of choices of i is at most q_m, and the number of choices of a is at most q_v. Hence
$$\Pr[\mathsf{B.13}] \leq \frac{2q_m q_v \epsilon}{2^n}. \tag{47}$$

Putting (46) and (47) together, we have

$$\Pr[\mathsf{B.13}] \leq q_v \epsilon + \frac{2q_m q_v \epsilon}{2^n}. \tag{48}$$

Bounding B.14. For a fixed choice of indices, the event can be expressed as the following linear equations:

$$\begin{cases} k^{u_i} = \nu_i \oplus \hat{x}^0 \\ k^{u'_a} = \nu'_a \oplus \hat{x}'^0 \\ \mathsf{H}_i^{u_i} \oplus \mathsf{H}'^{u'_a}_a \oplus k^{u_i} \oplus k^{u'_a} = \nu_i \oplus \nu'_a \\ t_i \oplus t'_a = \hat{y}^0 \oplus \hat{y}'^0. \end{cases}$$

Note that since our goal is the prove $2n/3$-bit security, it is sufficient the only focus on the first and the third equations. Hence we define the following event:

$$\mathsf{S} \overset{\Delta}{=} \exists i \in [q_m], a \in [q_v], (\hat{x}^0, \hat{y}^0) : k^{u_i} = \nu_i \oplus \hat{x}^0, \mathsf{H}_i^{u_i} \oplus \mathsf{H}_a'^{u_a'} \oplus k^{u_i} \oplus k^{u_a'} = \nu_i \oplus \nu_a'.$$

Now, we bound the probability of the event $\mathsf{B.14}$ as follows:

$$\Pr[\mathsf{B.14}] \leq \Pr[\mathsf{B.14} \mid \overline{\mathsf{S}}] + \Pr[\mathsf{S}].$$

Note that the probability of the event $\mathsf{B.14} \mid \overline{\mathsf{S}}$ is zero. It is easy to see that for a fixed choice of indices, the probability that the event S happens is $2\epsilon/2^n$, by using the randomness of the secret key k^{u_i} and the hash key $k_h^{u_i}$. Number of choices for (i, a) and $(\hat{x}^0, \hat{y}^0)$ is at most pq_mq_v. Hence, we have

$$\Pr[\mathsf{B.14}] \leq \frac{pq_mq_v\epsilon}{2^n}. \tag{49}$$

Bounding B.15. For a fixed choice of indices, the event can be expressed as the following linear equations:

$$\begin{cases} \nu_i \oplus k^{u_i} = \nu_a' \oplus k^{u_a'} \\ \mathsf{H}_i^{u_i} \oplus k^{u_i} = \nu_i \oplus \hat{x}^1 \\ \mathsf{H}_a'^{u_a'} \oplus k^{u_a'} = \nu_a' \oplus \hat{x}'^1 \\ t_a' \oplus t_i = \hat{y}^1 \oplus \hat{y}'^1. \end{cases}$$

For this bad event, we need to consider the following cases, namely when (a) $u_i = u_a'$ (the i-th MAC query and the a-th verification query belong to the same user), and when (b) $u_i \neq u_a'$ (the i-th MAC query and the a-th verification query belong to the different users).

1. For case (a), we have $k^{u_i} = k^{u_a'}$ and $k_h^{u_i} = k_h^{u_a'}$, the first equation becomes $\nu_i = \nu_a'$. We bound this event under the assumption that an adversary makes the verification attempt after all MAC and primitive queries. This assumption is sound in the sense that the forging advantage of an adversary who is making verification attempts after all MAC and primitive queries is identical to the forging advantage of an adversary who is making verification attempts interleaved with MAC and primitive queries. This assumption leads us to analyze this event when the verification query is the latest. Let us define the following random variable: let $z_{j\ell}$ be the random variable representing the sum of $y_j^1 \oplus y_\ell^1$, where y_j^1 (resp. y_ℓ^1) be the response of the primitive query x_j^1 (resp. x_ℓ^1), for $j \in [p], \ell \in [p]$. Now, we consider a tuple $\widetilde{Z} = (z_{j\ell})_{j,\ell \in [p]}$ of length p^2 and define multi-collision of the tuple $\widetilde{Z}$ as follows: an *r-multicollision* is said to occur in the tuple $\widetilde{Z}$ if there exist a finite set $\{i_1, i_2, \ldots, i_r\} \subseteq [p^2]$ such that $z_{i_1} = z_{i_2} = \ldots = z_{i_r}$. Thus, *at most r multicollisions* occur in $\widetilde{Z}$, denoted as $\mathbf{mc}(\widetilde{Z})$, if for any subset of indices $\{i_1, i_2, \ldots, i_r, i_{r+1}\}$ of $[p^2]$, $z_{i_1} = z_{i_2} = \ldots = z_{i_r} = z_{i_{r+1}}$ implies there exist $j \neq k$ in $[r+1]$ such that $i_j = i_k$. Let E denotes the event that $\mathsf{mc}(\widetilde{Z}) \geq \rho$, where $\rho \overset{\Delta}{=} \max\left\{n, \frac{12p^2}{2^n}\right\}$. Then, from [18] we have

$$\Pr[\mathbf{mc}(\widetilde{Z}) \geq \rho] \leq \frac{\binom{p^2}{\rho}}{2^{n(\rho-1)}} \overset{(1)}{\leq} 2^n \frac{p^{2\rho}}{2^{n\rho}\rho!} \overset{(2)}{\leq} 2^n \left(\frac{p^2 e}{2^n \rho}\right)^\rho \leq 2^n \left(\frac{3p^2}{2^n \rho}\right)^n \overset{(3)}{\leq} \frac{1}{2^n},$$

where (1) follows as $\binom{q}{\rho} \leq q^\rho/\rho!$. (2) follows as $e^\rho \geq \rho^\rho/\rho!$ and finally (3) follows as $\rho = \max\{n, 12p^2/2^n\}$. Now, we write the event $\mathsf{B.15}$ as follows:

$$\Pr[\mathsf{B.15}] \leq \Pr[\mathsf{B.15} \mid \overline{\mathsf{E}}] + \Pr[\mathsf{E}]. \tag{50}$$

To bound the first term of Eqn. (50), we use the randomness of k^{u_i} from the second equation and the randomness of the hash key $k_h^{u_a'}$ from the third equation. However, the number of choices of the verification query is at most q_v and for each such choices, number of choice of i is η. Therefore, by varying over all possible choice of indices, we have

$$\Pr[\mathsf{B.15}] \leq \frac{24\eta q_v p^2\epsilon}{2^{2n}} + \frac{1}{2^n}. \tag{51}$$

2. For case (b), the keys k^{u_i}, $k^{u'_a}$, $k_h^{u_i}$, and $k_h^{u'_a}$ are generated independently of each other. For any MAC query $(u_i, \nu_i, m_i, t_i) \in \tau_m$ and verification query $(u'_a, \nu'_a, m'_a, t'_a) \in \tau_v$ such that $u_i \neq u'_a$, we rely on the randomness of k^{u_i} (or $k^{u'_a}$) in the equation $\nu_i \oplus k^{u_i} = \nu'_a \oplus k^{u'_a}$, on the randomness of the hash key $k_h^{u_i}$ in the equation $\mathsf{H}_i^{u_i} \oplus k^{u_i} = \nu_i \oplus \hat{x}^1$, and on the randomness of the hash key $k_h^{u'_a}$ in the equation $\mathsf{H}'^{u'_a}_a \oplus k^{u'_a} = \nu'_a \oplus \hat{x}'^1$. In the ideal world, k^{u_i} (or $k^{u'_a}$), $k_h^{u_i}$, and $k_h^{u'_a}$ are dummy keys, sampled uniformly and independently from their respective space. Therefore, for a fixed choice of $i, a, (\hat{x}^1, \hat{y}^1)$, and $(\hat{x}'^1, \hat{y}'^1)$, the probability of the event is $\epsilon^2/2^{n-1}$. The number of choice of i is at most q_m, the number of choice of a is at most q_v, the number of choice of $(\hat{x}^1, \hat{y}^1)$ and $(\hat{x}'^1, \hat{y}'^1)$ is at most $p(p-1)/2$. Hence

$$\Pr[\mathsf{B.15}(b)] \leq \frac{q_m q_v p^2 \epsilon^2}{2^n}. \tag{52}$$

Putting (51) and (52) together, we have

$$\Pr[\mathsf{B.15}] \leq \frac{24\eta q_v p^2\epsilon}{2^{2n}} + \frac{q_m q_v p^2 \epsilon^2}{2^n} + \frac{1}{2^n}. \tag{53}$$

Bounding B.16. For a fixed choice of indices, the event can be expressed as the following linear equations:

$$\begin{cases} k^{u_i} = \nu_i \oplus \hat{x}^0 \\ k^{u_i} \oplus k^{u'_a} = \nu_j \oplus \nu'_a \\ \mathsf{H}_j^{u_j} \oplus k^{u_j} = \nu_j \oplus \hat{x}^1 \\ \mathsf{H}_i^{u_i} \oplus \mathsf{H}'^{u'_a}_a \oplus k^{u_i} \oplus k^{u'_a} = \nu_i \oplus \nu'_a \\ t'_a \oplus t_i \oplus t_j = \hat{y}^0 \oplus \hat{y}^1. \end{cases}$$

Note that since our goal is the prove $2n/3$-bit security, it is sufficient the only focus on the first and the fourth equations. Hence we define the following event:

$$\mathsf{S} \triangleq \exists i \in [q_m], a \in [q_v], (\hat{x}^0, \hat{y}^0) : k^{u_i} = \nu_i \oplus \hat{x}^0, \mathsf{H}_i^{u_i} \oplus \mathsf{H}'^{u'_a}_a \oplus k^{u_i} \oplus k^{u'_a} = \nu_i \oplus \nu'_a.$$

We bound the probability of the event $\mathsf{B.16}$ as follows:

$$\Pr[\mathsf{B.16}] \leq \Pr[\mathsf{B.16} \mid \overline{\mathsf{S}}] + \Pr[\mathsf{S}].$$

Note that the probability of the event $\mathsf{B.16} \mid \overline{\mathsf{S}}$ is zero. It is easy to see that for a fixed choice of indices, the probability that the event S happens is at most $2\epsilon/2^n$, by using the randomness of the secret key k^{u_i} and the hash key $k_h^{u_i}$. Number of choices for (i, a) and $(\hat{x}^0, \hat{y}^0)$ is at most $pq_m q_v$. Hence, we have

$$\Pr[\mathsf{B.16}] \leq \frac{pq_m q_v \epsilon}{2^n}. \tag{54}$$

Bounding B.17. Note that since the keys for different users are generated independently of each other, it is sufficient to consider the single user case when $k_{i_1} = k_{i_2} = \ldots = k_{i_{\xi+1}}$. Then, the event $\mathsf{B.17}$ occurs if there exist $\xi+1$ distinct signing query indices $\{i_1, \ldots, i_{\xi+1}\} \subseteq [q_m]$ such that $\nu_{i_1} \oplus \mathsf{H}_{i_1} = \ldots = \nu_{i_{\xi+1}} \oplus \mathsf{H}_{i_{\xi+1}}$. This event is thus a $(\xi+1)$-multicollision on the ϵ-universal hash function [5] mapping (ν, m) to $\nu \oplus \mathsf{H}_{k_h}(m)$ (as H_{k_h} is an ϵ-almost-xor universal). Therefore, by applying the multicollision theorem of universal hash function (Theorem 1) of [31], we have

$$\Pr[\mathsf{B.17}] \leq q_m^2 \epsilon / 2\xi. \tag{55}$$

Bounding B.18. For a fixed choice of i, the probability that $t_i = 0^n$ is exactly 2^{-n}. Summing over all possible choices of i we have

$$\Pr[\mathsf{B.18}] \leq \frac{q_m}{2^n}. \tag{56}$$

The proof follows from Eqn. (21)-Eqn. (56). □

[5] A hash function H_{k_h} is said to be an ϵ-universal hash function if for all $x \neq x'$, $\Pr[\mathsf{H}_{k_h}(x) = \mathsf{H}_{k_h}(x')] \leq \epsilon$.