Forgery Attacks on Several Beyond-Birthday-Bound Secure MACs

Yaobin Shen[1], François-Xavier Standaert[1](✉), and Lei Wang[2](✉)

[1] UCLouvain, ICTEAM, Crypto Group, Louvain-la-Neuve, Belgium
yaobin.shen@uclouvain.be, fstandae@uclouvain.be
[2] Shanghai Jiao Tong University, Shanghai, China
wanglei_hb@sjtu.edu.cn

Abstract. At CRYPTO'18, Datta et al. proposed nPolyMAC and proved the security up to $2^{2n/3}$ authentication queries and 2^n verification queries. At EUROCRYPT'19, Dutta et al. proposed CWC+ and showed the security up to $2^{2n/3}$ queries. At FSE'19, Datta et al. proposed PolyMAC and its key-reduced variant 2k-PolyMAC, and showed the security up to $2^{2n/3}$ queries. This security bound was then improved by Kim et al. (EUROCRYPT'20) and Datta et al (FSE'23) respectively to $2^{3n/4}$ and in the multi-user setting. At FSE'20, Chakraborti et al. proposed $\mathsf{PDM}^*\mathsf{MAC}$ and $\mathsf{1k\text{-}PDM}^*\mathsf{MAC}$, and showed the security up to $2^{2n/3}$ queries. Recently, Chen et al. proposed nEHtM_p^+ and showed the security up to $2^{2n/3}$ queries. In this paper, we show forgery attacks on nPolyMAC, CWC+, PolyMAC, 2k-PolyMAC, $\mathsf{PDM}^*\mathsf{MAC}$, $\mathsf{1k\text{-}PDM}^*\mathsf{MAC}$ and nEHtM_p^+. Our attacks exploit some vulnerability in the underlying polynomial hash function Poly, and (i) require only one authentication query and one verification query; (ii) are nonce-respecting; (iii) succeed with probability 1. Thus, our attacks disprove the provable high security claims of these schemes. We then revisit their security analyses and identify what went wrong. Finally, we propose two solutions that can restore the beyond-birthday-bound security.

Keywords: Message authentication code · Beyond-birthday-bound security · Polynomial hash function · Forgery attack

1 Introduction

Message authentication codes (MAC) are symmetric cryptographic primitives that allow senders and receivers who share a common secret key to ensure integrity and authenticity of a transmitted message. A MAC is typically designed from block ciphers, from hash functions or from universal hash functions. In this paper, we focus on the third class. The most widely used schemes are designed following the Wegman-Carter paradigm [35]: the input message is first mapped to a fixed-length string using a universal hash function indexed by a secret key, and then the resulting string is masked with a one-time pad. The one-time pad is typically achieved by using a block cipher with a unique nonce each time, e.g., the mechanism used in GMAC/GCM [28,1,2,21,30]. This method is simple and efficient, yet the security vanishes when the nonces are misused since the hash key can then be recovered. On the other hand, when the nonces are unique as required, its security caps at the so-called birthday bound $2^{n/2}$, i.e., resisting against at most $2^{n/2}$ queries. Indeed, the outputs of a random permutation can be distinguished from random strings within roughly $2^{n/2}$ queries. This bound is not always satisfying in practical applications. For conventional platforms where block ciphers like the AES are used with $n = 128$, it implies that we need to renew the key when the number of authentication queries exceeds 2^{32} if we want to maintain the forgery advantage of an adversary below $1/2^{32}$. For resource-constrained environments, where lightweight block ciphers with 64-bit block or even shorter [8,9,3,10] are likely to implemented, the bound becomes 2^{32} and is vulnerable in certain applications [6].

To go beyond the birthday bound and resist against nonce misuse, Cogliati and Seurin [13] proposed a scheme called Encrypted Wegman-Carter with Davies-Meyer (EWCDM) that requires one universal hash function and two block cipher calls with independent keys. They instantiated the one-time pad with the Davies-Meyer construction and encrypted the output of the Wegman-Carter construction with another block cipher call. They showed that this scheme is provably secure up to $2^{2n/3}$ authentication queries and 2^n verification queries against nonce-respecting adversaries, and secure up to $2^{n/2}$ authentication and verification queries against nonce-misusing adversaries[3]. Later, Mennink and Neves [31] improved this security bound to the optimal 2^n in the nonce-respecting setting using the mirror theory. To reduce the number of keys, Datta el al. [16] then proposed Decrypted Wegman-Carter with Davies-Meyer (DWCDM), which is similar to EWCDM except that the outer encryption call is replaced by a decryption call. The advantage of DWCDM is that the two block cipher calls can use the same key. It even becomes a truly single-key MAC if the hash key is derived as $K_h = E_K(0^{n-1} \,\|\, 1)$. They proved that DWCDM can achieve the security up to $2^{2n/3}$ authentication queries and 2^n verification queries against nonce-respecting adversaries, and remains secure up to $2^{n/2}$ authentication queries and 2^n verification queries against nonce-misusing adversaries. They then proposed nPolyMAC, a concrete instance of DWCDM by realizing the universal hash function with a polynomial hash, and proved that nPolyMAC enjoys the same beyond-birthday-bound security as DWCDM. Recently, Chakraborti [11] proposed PDM*MAC and 1k-PDM*MAC, a permutation-based variant of DWCDM and its single-key version, and proved that these two schemes are both secure up to $2^{2n/3}$ queries against nonce-respecting adversaries, which is tight as illustrated with a matching attack.

Another popular approach to achieve the beyond-birthday-bound security and maintain security against nonce misuse is to use the nonce-based Enhanced Hash-then-Mask (nEHtM) paradigm. The Enhanced Hash-then-Mask (EHtM)

[3] An adversary is said to be nonce-respecting if she does not repeat nonces in authentication queries, and is said to be nonce-misusing if she repeats nonces in authentication queries.

method was originally proposed by Minematsu [32] to construct a probabilistic MAC with beyond-birthday-bound security. It requires an n-bit salt, two independent pseudorandom functions and a universal hash function, and is proved to achieve a tight $2^{3n/4}$ security [18]. Dutta et al. [19] turned this method into a nonce-based MAC named nEHtM where (i) the random salt is replaced by the nonce; (ii) the two independent pseudorandom functions are replaced by a single-key block cipher with domain separation. They showed that nEHtM has beyond-birthday-bound security that gracefully degrades under nonce misuse. They then proposed a nonce-based AE coined CWC+ by combining nEHtM with the encryption mode CENC [24]. They proved that CWC+ provides the authenticity up to $2^{2n/3}$ queries against nonce-respecting adversaries and maintains gracefully degrading security up to $2^{n/2}$ queries against nonce-misusing adversaries. Recently, Chen et al. [12] proposed nEHtM_p^+, a permutation-based variant of nEHtM, and proved that it is secure up to $2^{2n/3}$ authentication and verification queries in both single-user and multi-user settings.

There is also another approach called Double-block Hash-then-Sum (DbHtS) [14] to provide beyond-birthday-bound security without a nonce. It requires two n-bit universal hash functions, and thus is less efficient than the above two methods that require a single n-bit universal hash when nonces are available. Nevertheless, it enjoys high provable security guarantees. A notable example is PolyMAC that is built from two polynomial hash functions and two block cipher calls. A series of works showed that PolyMAC and its key-reduced variant 2k-PolyMAC are provably highly secure in both single-user and multi-user settings. Datta et al. [14] proved that both PolyMAC and 2k-PolyMAC can achieve $2^{2n/3}$ security. Kim et al. [27] improved the security bound of PolyMAC to $2^{3n/4}$. Recently, Datta et al. [15] further showed that 2k-PolyMAC can achieve $2^{3n/4}$ security in the multi-user setting.

Our contribution. In this paper, we show forgery attacks on beyond-birthday-bound secure schemes, including nPolyMAC [16], CWC+ [20], PolyMAC [14,27], 2k-PolyMAC [14,15], PDM*MAC [11], 1k-PDM*MAC [11], and nEHtM_p^+ [12]. Interestingly, all of these schemes use the same polynomial hash function called Poly [16,20,14,27,15,11,12] to handle arbitrary length messages. This polynomial hash function is supposed to hash a message efficiently and securely, and is backed up with security proofs. Yet, as we discovered, it has some vulnerability. Although Poly implicitly encodes the length of a message as a parameter in the polynomial by $M_i \cdot K_h^{\ell+1-i}$, these terms can be zeroed out if we choose $M_i = 0^n$. Hence, it allows length-extension attack by prepending arbitrary number of 0^n blocks while the hashed value remains the same. By exploiting this vulnerability, we thus mount forgery attacks against all of these schemes. Notably, our attacks (i) require only one authentication query and one verification query; (ii) are nonce-respecting; (iii) succeed with probability 1. Thus, our attacks disprove their high provable security claims.

We remark that all of forgery attacks against these schemes follow the same principle. If we abstract these schemes by a single construction $\mathrm{MAC}(N, M) = \mathrm{F}(N, \mathsf{Poly}(M))$ where F is a function, Poly the polynomial hash function, M

the message, and N the nonce (there is no nonce in PolyMAC), then this attack principle can be summarized as follows: first query $T = \mathrm{MAC}(N, M) = \mathrm{F}(N, \mathsf{Poly}(M))$, and then $(N, (0^n)_i \| M, T)$ is a valid forgery against these schemes for any $i \geq 1$ as $\mathsf{Poly}(M) = \mathsf{Poly}((0^n)_i \parallel M)$ always holds, where $(0^n)_i$ denotes the i-time concatenation of string 0^n.

We then revisit their security analyses to see what went wrong. Their beyond-birthday-bound security analyses require the underlying polynomial hash to be (i) ϵ_1-regular, namely for any message, the probability that the hashed value equals to any constant value should be negligible; (ii) ϵ_2-almost-xor-universal, namely for any two distinct messages, the probability that the difference of these two hashed values equals to any constant value should be negligible; (iii) ϵ_3-3-way-regular, namely for any three distinct messages, the probability that the sum of these three hashed values equals to a non-zero constant value should be negligible, when the hash key is uniformly chosen from the key space. Note that the almost-xor-universal property is required in security analyses of all these constructions, while the regular property is needed in nPolyMAC, PDM*MAC, 1k-PDM*MAC and nEHtM$_p^+$[4], and the 3-way-regular property is needed in nPolyMAC, PDM*MAC and 1k-PDM*MAC. Apparently, Poly does not meet the second property since the difference of two hashed values is always 0^n for any two messages M and M' where M' is obtained by prepending arbitrary 0^n blocks to M. Thus, the proposition [16,17,15] that showed Poly meets these three properties is flawed and cannot be fixed. Consequently, the security analyses of the schemes nPolyMAC, CWC+, PolyMAC, 2k-PolyMAC, PDM*MAC, 1k-PDM*MAC and nEHtM$_p^+$ that rely on this result to prove the beyond-birthday-bound security are flawed.

Finally, we propose two polynomial hash functions called PolyX and GHASHX that both meet regular, almost-xor-universal and 3-way-regular properties. The first one, PolyX, is a variant of Poly by reversing the order of a message in the polynomial. By doing so, the length-dependent term $M_\ell 10^* \cdot K_h^\ell$ in the polynomial will never be zeroed out since $M_\ell 10^*$ is always a non-zero value. We then prove that PolyX is ϵ_1-regular, ϵ_2-almost-xor-universal, and ϵ_3-3-way-regular, where $\epsilon_1 = \epsilon_2 = \epsilon_3 = \ell_{\max}/2^n$ and $\ell_{\max}$ is the maximum number of n-bits blocks of a message. The second one, GHASHX, is a variant of GHASH [30,29,28] by replacing the 0^* padding with 10^*. Although GHASH is well-known to be a ϵ_2-almost-xor-universal hash where $\epsilon_2 = (\ell_{\max} + 1)/2^n$, it is not regular since for an empty message $M = \varepsilon$, the hashed value always equals to 0^n. Even worse, if we instantiate nPolyMAC with GHASH, then it will result in a forgery attack. The 10^* padding can avoid this issue as it always appends a 1 first. We then prove that GHASHX is ϵ_1-regular, ϵ_2-almost-xor-universal, and ϵ_3-3-way regular where $\epsilon_1 = \epsilon_2 = \epsilon_3 = (\ell_{\max} + 1)/2^n$. Hence, by instantiating nPolyMAC, CWC+, PolyMAC, 2k-PolyMAC, PDM*MAC, 1k-PDM*MAC and nEHtM$_p^+$ with either of PolyX and GHASHX, we can restore their beyond-birthday-bound security.

[4] The regular property is also required in the multi-user security analysis of 2k-PolyMAC [15] and is not mandatory in its single-user security analysis.

Organization. We first introduce notations and security notions in section 2. We then show forgery attacks on $\mathsf{nPolyMAC}$, $\mathsf{CWC+}$, $\mathsf{PolyMAC}$, $\mathsf{2k\text{-}PolyMAC}$, $\mathsf{PDM^*MAC}$, $\mathsf{1k\text{-}PDM^*MAC}$ and nEHtM_p^+ in section 3. Next, we discuss the issues in their security analyses and propose two solutions that can restore their beyond-birthday-bound security in section 4. Finally, we conclude the paper in section 5. We also provide an overview of how Poly is used in these schemes in Appendix A.

2 Preliminaries

Notation. Let ε denote the empty string. Let $\{0,1\}^*$ be the set of all finite bit strings including the empty string ε. For a finite set $\mathcal{X}$, we let $X \leftarrow\!\!\$\, \mathcal{X}$ denote the uniform sampling from $\mathcal{X}$ and assigning the value to X. Let $|X|$ denote the length of string X. Let $|X|_n$ denote the n-bit encoding of the length of string X. Concatenation of strings X and Y is written as $X \parallel Y$ or simply XY. $X10^*$ denotes the padding that appended with a single 1 and as few 0 bits so that the length of string to be a multiple of n. We let $Y \leftarrow \mathcal{A}(X_1, \ldots; r)$ denote running algorithm $\mathcal{A}$ with randomness r on inputs $X_1, \ldots$ and assigning the output to Y. We let $Y \leftarrow\!\!\$\, \mathcal{A}(X_1, \ldots)$ be the result of picking r at random and letting $Y \leftarrow \mathcal{A}(X_1, \ldots; r)$. Let $\mathrm{Perm}(n)$ denote the set of all permutations over $\{0,1\}^n$, and let $\mathrm{Func}(*, n)$ denote the set of all functions from $\{0,1\}^*$ to $\{0,1\}^n$. For a string $X \in \{0,1\}^*$, let $(X)_i$ denote concatenating X itself by i times, namely $(X)_i = X \parallel \ldots \parallel X$ where X repeats i times.

Block ciphers and PRFs. An adversary $\mathcal{A}$ is a probabilistic algorithm that has access to one or more oracles. Let $\mathcal{A}^{O_1, O_2, \cdots}$ denote an adversary $\mathcal{A}$ interacting with oracles $O_1, O_2, \ldots$, and $\mathcal{A}^{O_1, O_2, \cdots} = 1$ denote the event that $\mathcal{A}$ outputs 1 after interacting with $O_1, O_2, \ldots$. The resources of $\mathcal{A}$ are measured in terms of time and query complexities. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Let $\pi \leftarrow\!\!\$\, \mathrm{Perm}(n)$ be a random permutation. The advantage of $\mathcal{A}$ against the PRP security of E is defined as

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

where K is chosen uniformly at random from $\{0,1\}^k$. The block cipher E is said to be a (q, t, ϵ)-secure PRP if $\mathsf{Adv}_E^{\mathrm{prp}}(q, t) = \max_{\mathcal{A}} \mathsf{Adv}_E^{\mathrm{prp}}(\mathcal{A}) \leq \epsilon$ where the maximum is taken over all adversaries $\mathcal{A}$ that makes at most q queries and runs in time at most t.

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

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

where K is chosen uniformly at random from $\mathcal{K}$. The function F is said to be a (q, t, ϵ)-secure PRF if $\mathsf{Adv}_F^{\mathrm{prf}}(q, t) = \max_{\mathcal{A}} \mathsf{Adv}_F^{\mathrm{prf}}(\mathcal{A}) \leq \epsilon$ where the maximum is taken over all adversaries $\mathcal{A}$ that makes at most q queries and runs in time at most t.

Message authentication code. A message authentication code (MAC) scheme Π is a triplet of algorithms $(\mathrm{Gen}, \mathrm{Auth}, \mathrm{Ver})$, where Gen is the key-generation algorithm, Auth is the authentication algorithm, and Ver is the verification algorithm. The key-generation algorithm Gen samples a key K uniformly at random from the key space $\mathcal{K}$. The authentication algorithm Auth takes as input a key $K \in \mathcal{K}$ and a message $M \in \mathcal{M}$, and outputs a tag $T \in \{0,1\}^\tau$ where $T \leftarrow \mathrm{Auth}_K(M)$. The verification algorithm takes as input a key $K \in \mathcal{K}$, a message $M \in \mathcal{M}$ and a tag $T \in \{0,1\}^\tau$, and outputs 1 if $\mathrm{Auth}_K(M) = T$ and otherwise a symbol $\perp$ indicating invalidity.

An adversary $\mathcal{A}$ has oracle access to Auth_K and Ver_K, and attempts to forge a pair of message and tag against the MAC scheme Π. We say $\mathcal{A}$ forges successfully if she outputs a pair of (M, T) that can pass the verification oracle Ver_K and M has not been queried to Auth_K before. The advantage of $\mathcal{A}$ against the unforgeability of Π is defined as

$$\mathsf{Adv}_{\Pi}^{\mathrm{mac}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathrm{Auth}_K, \mathrm{Ver}_K} \text{ forges}\right]$$

where K is chosen uniformly at random from $\mathcal{K}$. The scheme Π is said to be a (q_m, q_v, t, ϵ)-secure MAC if $\mathsf{Adv}_{\Pi}^{\mathrm{mac}}(q_m, q_v, t) = \max_{\mathcal{A}} \mathsf{Adv}_{\Pi}^{\mathrm{mac}}(\mathcal{A}) \leq \epsilon$ where the maximum is taken over all adversaries $\mathcal{A}$ that makes at most q_m authentication queries, q_v verification queries, and runs in time at most t.

Nonce-based MAC. A nonce-based MAC scheme Π takes an additional parameter called nonce $N \in \mathcal{N}$. The key-generation algorithm Gen samples a key K uniformly at random from the key space $\mathcal{K}$. The authentication algorithm Auth takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$ and a message $M \in \mathcal{M}$, and outputs a tag $T \in \{0,1\}^\tau$ where $T \leftarrow \mathrm{Auth}_K(N, M)$. The verification algorithm takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, a message $M \in \mathcal{M}$ and a tag $T \in \{0,1\}^\tau$, and outputs 1 if $\mathrm{Auth}_K(N, M, T) = T$ and otherwise a symbol $\perp$ indicating invalidity.

The adversary is said to be nonce-respecting if she does not repeat nonces in authentication queries, and is said to be nonce-misusing if she repeats nonces in authentication queries. However, in both cases, the adversary can always repeat nonces in verification queries, either using the same nonce in two verification queries or repeating the nonce between a verification query and a authentication query. We say $\mathcal{A}$ forges successfully if she outputs a tuple of (N, M, T) that can pass the verification oracle Ver_K and (N, M) has not been queried to Auth_K before. The advantage of $\mathcal{A}$ against the unforgeability of Π is defined as

$$\mathsf{Adv}_{\Pi}^{\mathrm{mac}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathrm{Auth}_K, \mathrm{Ver}_K} \text{ forges}\right]$$

where K is chosen uniformly at random from $\mathcal{K}$. The scheme Π is said to be a (q_m, q_v, t, ϵ)-secure MAC if $\mathsf{Adv}_{\Pi}^{\mathrm{mac}}(q_m, q_v, t) = \max_{\mathcal{A}} \mathsf{Adv}_{\Pi}^{\mathrm{mac}}(\mathcal{A}) \leq \epsilon$ where the maximum is taken over all adversaries $\mathcal{A}$ that makes at most q_m authentication queries, q_v verification queries, and runs in time at most t.

Authenticated encryption. An authenticated encryption (AE) scheme Π is a triplet of algorithms $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$, where Gen is the key-generation algorithm, Enc the encryption algorithm and Dec the decryption algorithm. The key-generation algorithm Gen samples a key K uniformly at random from the key space $\mathcal{K}$. The encryption algorithm Enc takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, an associated data $A \in \{0,1\}^*$ and a message $M \in \mathcal{M}$, and returns a pair of ciphertext and tag $(C,T) \in \{0,1\}^{|M|+\tau}$ where $(C,T) \leftarrow \mathrm{Enc}_K(N,A,M)$. The decryption algorithm takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, an associated data $A \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$ and a tag $T \in \{0,1\}^\tau$, and returns either a message $M \in \{0,1\}^*$ or a symbol $\perp$ indicating invalidity. For correctness, we assume that if $(C,T) \leftarrow \mathrm{Enc}_K(N,A,M)$, then $M \leftarrow \mathrm{Dec}_K(N,A,C,T)$. Note that the message M is encrypted and authenticated simultaneously, while the associated data A is only authenticated.

Similarly, the adversary is said to be nonce-respecting if she does not repeat nonces in encryption queries, and is said to be nonce-misusing if she repeats nonces in encryption queries. In both cases, the adversary can always repeat nonces in decryption queries. An AE scheme Π should provide both authenticity and confidentiality. In this paper, we only consider the adversary against the authenticity of Π. We say $\mathcal{A}$ forges successfully if she outputs a tuple of (N,A,C,T) that can pass the decryption oracle Dec_K and (N,A,C,T) has not been obtained from queries to Enc_K before. The advantage of $\mathcal{A}$ against the authenticity of Π is defined as

$$\mathsf{Adv}_{\Pi}^{\mathrm{Auth}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathrm{Enc}_K,\mathrm{Dec}_K} \text{ forges}\right]$$

where K is chosen uniformly at random from $\mathcal{K}$. The scheme Π is said to be a (q_e,q_d,t,ϵ)-secure authenticator if $\mathsf{Adv}_{\Pi}^{\mathrm{Auth}}(q_e,q_d,t) = \max_{\mathcal{A}} \mathsf{Adv}_{\Pi}^{\mathrm{Auth}}(\mathcal{A}) \leq \epsilon$ where the maximum is taken over all adversaries $\mathcal{A}$ that makes at most q_e encryption queries, q_d decryption queries, and runs in time at most t.

3 Forgery Attacks on Polynomial-Based Constructions

In this section, we show forgery attacks on several polynomial-based MACs that are claimed to achieve beyond-birthday-bound security, including nPolyMAC, CWC+, PolyMAC, 2k-PolyMAC, PDM*MAC, 1k-PDM*MAC and nEHtM_p^+. Our attacks (i) require only one authentication query and one verification query; (ii) are nonce-respecting; (iii) succeed with the probability 1.

All of these attacks are due to the polynomial hash function chosen in those schemes and follow the attack principle outlined in introduction: assuming a construction $\mathrm{MAC}(N,M) = \mathrm{F}(N,\mathsf{Poly}(M))$ where F is a function, Poly the polynomial hash function, M the message and N the nonce (there is no nonce in PolyMAC), the attack works by querying $T = \mathrm{MAC}(N,M) = \mathrm{F}(N,\mathsf{Poly}(M))$ and observing that $(N,(0^n)_i \parallel M,T)$ is then a valid forgery for any $i \geq 1$ as $\mathsf{Poly}(M) = \mathsf{Poly}((0^n)_i \parallel M)$ always holds.

We do an exhaustive description of how this attack principle results in forgery attacks against these schemes and postpone a more general discussion about the source of these attacks and how to fix them to the next section. We also provide a brief summary of how the polynomial hash function Poly is used in these schemes in Appendix A.

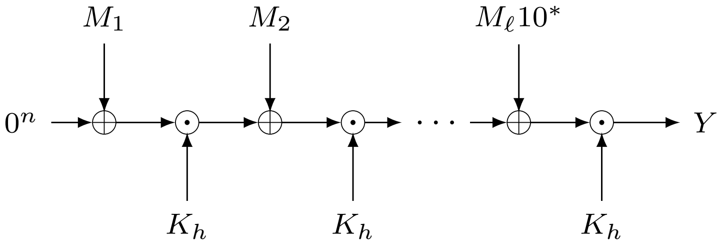

Fig. 1: The polynomial hash function Poly for a message $M = M_1 \parallel \ldots \parallel M_\ell$ with a hash key K_h.

3.1 Attack on nPolyMAC

nPolyMAC is an instance of DWCDM construction that is proved to achieve beyond-birthday-bound security [16,17]. It is built from a polynomial hash function Poly : $\mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ and a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, and can authenticate a message $M \in \{0,1\}^*$ of variable length. Given a message $M = M_1 \parallel M_2 \parallel \ldots \parallel M_\ell$ where $|M_i| = n$ and $0 \le M_\ell \le n-1$, the 10^* padding[5] is first applied to make the total string length of M a multiple of n. The polynomial hash Poly is defined as

$$\mathsf{Poly}_{K_h}(M) = M_1 \cdot K_h^\ell \oplus M_2 \cdot K_h^{\ell-1} \oplus \ldots \oplus M_\ell 10^* \cdot K_h \ , \tag{1}$$

where $K_h \in \mathcal{K}_h$ is the hash key. See Figure 1 for a pictorial illustration of Poly [6]. Then nPolyMAC is defined as

$$\mathsf{nPolyMAC}[\mathsf{Poly}, E](N, M) = E_K^{-1}(E_K(N) \oplus N \oplus \mathsf{Poly}_{K_h}(M))$$

where $N \in \{0,1\}^n$ is the nonce, $M \in \{0,1\}^*$ is the message.

In Theorem 4 of [16,17], it is proved that nPolyMAC is secure up to $2^{2n/3}$ authentication queries and 2^n verification queries against nonce-respecting adversaries, and remains secure up to $2^{n/2}$ authentication queries and 2^n verification queries against nonce-misusing adversaries. In the following, we show a forgery attack against nPolyMAC in the nonce-respecting setting that requires only one authentication query and one verification query, and succeeds with probability 1. Thus, the attack disproves the security claim of nPolyMAC.

The adversary can mount an attack against nPolyMAC as follows. She chooses an arbitrary message $M \in \{0,1\}^*$ and a nonce N, and queries (N, M) to obtain the tag T where

$$T = E_K^{-1}(E_K(N) \oplus N \oplus \mathsf{Poly}_{K_h}(M)) \ ,$$

[5] The 10^* padding is explicitly used as an injective padding method for Poly in [16,17,20,19,14,15,11,12].

[6] Part of this figure is inspired by IACR TikZ [26]

and

$$\mathsf{Poly}_{K_h}(M) = M_1 \cdot K_h^{\ell} \oplus M_2 \cdot K_h^{\ell-1} \oplus \ldots \oplus M_\ell 10^* \cdot K_h \ .$$

Then the tuple (N, M', T) where $M' = (0^n)_i \parallel M$ is a valid forgery for any $i \geq 1$ since the equation $\mathsf{Poly}_{K_h}(M') = \mathsf{Poly}_{K_h}(M)$ always holds as

$$\mathsf{Poly}_{K_h}(M') = 0^n \cdot K_h^{\ell+i} \oplus \ldots \oplus 0^n \cdot K_h^{\ell+1} \oplus M_1 \cdot K_h^{\ell} \oplus \ldots \oplus M_\ell 10^* \cdot K_h = \mathsf{Poly}_{K_h}(M) \ .$$

This attack thus invalidates the beyond-birthday-bound security claim of nPoly-MAC.

Remark 1. The reason why this attack works is that (i) the finite field multiplication has a fixed point 0^n, namely for any K_h, the result of $0^n \cdot K_h^{\ell+i}$ is always 0^n; (ii) although the design of Poly_{K_h} implicitly encodes the length of the messages as a parameter by $M_i \cdot K_h^{\ell+1-i}$, these terms will be canceled out if we choose M_i to be 0^n. Thus, we can prepend arbitrary number of 0^n blocks to a message while the hash value of this message remains the same. This attack looks simple but can be harmful, since the adversary can choose any message M that may contain some information that is unwilling to repeat again, and extend the number of 0^n blocks so that the message is always regarded as new and accepted by the receiver with probability 1.

3.2 Attack on CWC+

CWC+ [20] is a nonce-based authenticated encryption following the Encrypt-then-MAC paradigm [4]. The encryption of CWC+ is based on a variant of CENC [25] encryption scheme called $\mathsf{CENC}_{\max}$ [7]. Taking as input a block cipher key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^n$, and a length parameter $\ell < 2^{n/4}$, $\mathsf{CENC}_{\max}$ outputs a sequence of key stream blocks $(S_1, \ldots, S_\ell)$, where the i-th key stream block is defined as

$$S_i = E_K(N) \oplus E_K(N + i) \ .$$

The authentication of CWC+ is built from a beyond-birthday-bound secure MAC algorithm called nEHtM [20]. Taking as input a block cipher key $K \in \{0,1\}^k$, a hash key $K_h \in \mathcal{K}_h$, a nonce $N \in \{0,1\}^{n-1}$ and a message $M \in \{0,1\}^*$, nEHtM is defined as

$$\mathsf{nEHtM}[E, H](N, M) = E_K(0 \parallel N) \oplus E_K(1 \parallel (N \oplus H_{K_h}(M))) \ .$$

The $(n-1)$-bit hash function H is realized by truncating the first bit of polynomial hash Poly defined in Equation 1. The specification of CWC+ is given by combining $\mathsf{CENC}_{\max}$, nEHtM and Poly that is illustrated in Figure 2.

The Theorem 2 of [20,19] shows that CWC+ provides security up to $2^{2n/3}$ queries for both authenticity and confidentiality against nonce-respecting adversaries, and maintains graceful birthday-bound security against nonce-misusing adversaries. In the following, we show a forgery attack against the authenticity of CWC+ in the nonce-respecting setting that requires only one encryption query

```
procedure Enc$(K, N, A, M)$
$L \leftarrow E_K(0^n); N' \leftarrow N \parallel 0^{n/4-1}$
$\ell \leftarrow \lceil |M|/n \rceil$
$S \leftarrow \mathsf{CENC}_{\max}(K, 0 \parallel N', \ell)$
$C \leftarrow M \oplus \mathsf{first}(S, |M|)$
$\widetilde{T} \leftarrow \mathsf{nEHtM}[E, \mathsf{Poly}_L](N', C \parallel A)$
$T \leftarrow \mathsf{chop}_\tau \lceil \widetilde{T} \rceil$; return $(C, T)$
```

```
procedure Dec$(K, N, A, C, T)$
$L \leftarrow E_K(0^n); N' \leftarrow N \parallel 0^{n/4-1}$
$\ell \leftarrow \lceil |C|/n \rceil$
$\widetilde{T}' \leftarrow \mathsf{nEHtM}[E, \mathsf{Poly}_L](N', C \parallel A)$
if $\mathsf{chop}_\tau \lceil \widetilde{T}' \rceil \neq T$ then return $\perp$
$S \leftarrow \mathsf{CENC}_{\max}(K, 0 \parallel N', \ell)$
$M \leftarrow C \oplus \mathsf{first}(S, |C|)$; return $M$
```

Fig. 2: Encryption and decryption procedures of CWC+. Here the nonce N is a $3n/4$-bit string. $\mathsf{first}(S, |M|)$ denotes the first $|M|$ bits of the string S. $\mathsf{chop}_\tau \lceil \cdot \rceil$ is a function that truncates the last $n - \tau$ bits of its input.

and one decryption query and succeeds with probability 1. Thus, this attack disproves the security claim of CWC+ regarding the authenticity.

Since the adversary can arbitrarily choose a message M and an associated data A, she can simply set the message M to the empty string ε and choose an arbitrary associated data $A \in \{0,1\}^*$. Then the attack idea is similar to the one for nPolyMAC. In detail, the adversary can mount an attack against the authenticity of CWC+ as follows. She sets the message M to the empty string ε and chooses an arbitrary associated data $A \in \{0,1\}^*$ and a nonce $N \in \{0,1\}^{3n/4}$. She queries (N, A, ε) to obtain the tag T where

$$T = E_K(0 \parallel N') \oplus E_K(1 \parallel (N' \oplus \mathsf{chop}_{n-1} \lfloor \mathsf{Poly}_L(A) \rfloor)) \enspace ,$$

and $N' = N \parallel 0^{n/4-1}$, $L = E_K(0^n)$, $\mathsf{chop}_{n-1} \lfloor \cdot \rfloor$ is a function that truncates the first bit of its input. Then the tuple (N, A', ε, T) where $A' = (0^n)_i \parallel A$ is a valid forgery against CWC+ for any $i \geq 1$ since $\mathsf{Poly}_L(A) = \mathsf{Poly}_L(A')$ always holds. Note that similar forgery attack also applies to nEHtM [20] when the hash function is instantiated with Poly.

3.3 Attacks on PolyMAC and 2k-PolyMAC

PolyMAC is a MAC algorithm built from the polynomial hash $\mathsf{Poly} : \mathcal{K}_h \times \{0,1\}^* \rightarrow \{0,1\}^n$ defined in Equation 1 and two block ciphers $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ as follows

$$\mathsf{PolyMAC}[\mathsf{Poly}, E](M) = E_{K_1}(\mathsf{Poly}_{K_{h_1}}(M)) \oplus E_{K_2}(\mathsf{Poly}_{K_{h_2}}(M)) \enspace .$$

Its key-reduced variant 2k-PolyMAC is defined as

$$\mathsf{2k\text{-}PolyMAC}[\mathsf{Poly}, E](M) = E_K(\mathsf{fix}_0(\mathsf{Poly}_{K_{h_1}}(M))) \oplus E_K(\mathsf{fix}_1(\mathsf{Poly}_{K_{h_2}}(M))) \enspace ,$$

where the domain separating functions fix_0 and fix_1 fix the least significant bit of a string to be 0 and 1 respectively.

A series of works show that PolyMAC and its key-reduced variant 2k-PolyMAC enjoy provably high security in both single-user and multi-user settings. Datta et al. [14] proved that both PolyMAC and 2k-PolyMAC can achieve $2^{2n/3}$ security. Later, Kim et al. [27] improved the security bound of PolyMAC to be $2^{3n/4}$ by assuming an injective padding method. Recently, Datta et al. [15] further showed that 2k-PolyMAC can achieve $2^{3n/4}$ security in the multi-user setting. In the following, we show a forgery attack against PolyMAC that requires only one authentication query and one verification query and succeeds with probability 1. Similar attack also applies to 2k-PolyMAC. Thus, our attack disproves the security claim of both PolyMAC and 2k-PolyMAC.

The adversary can mount an attack against PolyMAC as follows. She first chooses an arbitrary message $M \in \{0,1\}^*$. She queries M to obtain T where

$$T = E_{K_1}(\mathsf{Poly}_{K_{h_1}}(M)) \oplus E_{K_2}(\mathsf{Poly}_{K_{h_2}}(M)) \ .$$

Then the pair of (M', T) is a valid forgery against PolyMAC where $M' = (0^n)_i \| M$ for any $i \geq 1$, since equations $\mathsf{Poly}_{K_{h_1}}(M') = \mathsf{Poly}_{K_{h_1}}(M)$ and $\mathsf{Poly}_{K_{h_2}}(M') = \mathsf{Poly}_{K_{h_2}}(M)$ always hold.

3.4 Attack on PDM*MAC and 1k-PDM*MAC

PDM*MAC [11] is a permutation-based Davis-Meyer MAC that is proved to achieve beyond-birthday-bound security. Given a key $K \in \{0,1\}^n$, a hash key $K_h \in \mathcal{K}_h$, an n-bit nonce N and a message $M \in \{0,1\}^*$, it computes a tag as follows

$$\mathsf{PDM^*MAC}[H, \pi] = \pi^{-1}(\pi(K \oplus N) \oplus 3K \oplus N \oplus H_{K_h}(M)) \oplus 2K$$

where $H : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^n$ is a hash function and π is a public permutation over $\{0,1\}^n$. The hash function H is instantiated with Poly as defined in Equation 1.

Theorem 2 of [11] shows that PDM*MAC is secure up to $2^{2n/3}$ queries against nonce-respecting adversaries and this security bound is tight illustrated with a matching attack. In the following, we show a forgery attack against PDM*MAC. Our attack requires only one authentication query and one verification query and succeeds with probability 1. Thus, our attack disproves the security claim of PDM*MAC instantiated with Poly.

The adversary can mount an attack against PDM*MAC as follows. She first chooses an arbitrary message $M \in \{0,1\}^*$. She asks M to PDM*MAC and obtains the tag T that is computed as

$$T = \pi^{-1}(\pi(K \oplus N) \oplus 3K \oplus N \oplus \mathsf{Poly}_{K_h}(M)) \oplus 2K \ .$$

Then the pair of (M', T) is a valid forgery against PDM*MAC where $M' = (0^n)_i \parallel M$ for any $i \geq 1$ since $\mathsf{Poly}_{K_h}(M') = \mathsf{Poly}_{K_h}(M)$ always holds. This attack also applies to 1k-PDM*MAC instantiated with Poly in [11], which is a single-key version of PDM*MAC and is proved in Theorem 3 [11] to achieve beyond-birthday-bound security.

3.5 Attack on $\mathbf{nEHtM}_p^+$

nEHtM_p^+[12] is a concrete instance of a permutation-based MAC called nEHtM_p^* that is proved to achieve beyond-birthday-bound security. It is built from a $(n-1)$-bit hash function $H : \mathcal{K}_h \times \{0,1\}^* \to \{0,1\}^{n-1}$ and a public permutation π over $\{0,1\}^n$ as follows:

$$\mathsf{nEHtM}_p^+[H, \pi](N, M) = \pi(0 \parallel N \oplus K) \oplus \pi(1 \parallel N \oplus K \oplus H_{K_h}(M))$$

where $K \in \{0,1\}^n$ is the key, $N \in \{0,1\}^{n-1}$ the nonce, $M \in \{0,1\}^*$ the message, and $H_{\mathcal{K}_h}(M)$ is instantiated by truncating the first bit of $\mathsf{Poly}_{K_h}(M)$ as defined in Equation 1.

It is proved in [12] that nEHtM_p^+ is secure up to $2^{2n/3}$ authentication queries and $2^{2n/3}$ verification queries in both single-user and multi-user settings. In the following we show a forgery attack that disproves this security claim.

Similarly to previous attacks, the adversary can mount an attack against nEHtM_p^+ as follows. She first chooses an arbitrary message $M \in \{0,1\}^*$ and a nonce N. She queries (N, M) to obtain the tag T where

$$T = \pi(0 \parallel N \oplus K) \oplus \pi(1 \parallel N \oplus K \oplus \mathsf{chop}_{n-1}\lfloor \mathsf{Poly}_{K_h}(M) \rfloor)$$

and $\mathsf{chop}_{n-1}\lfloor \cdot \rfloor$ is a function that truncates the first bit of its input. Then the tuple (N, M', T) is a valid forgery against nEHtM_p^+ where $M' = (0^n)_i \parallel M$ for any $i \geq 1$ since $\mathsf{Poly}_{K_h}(M') = \mathsf{Poly}_{K_h}(M)$ always holds.

Remark 2. Our attacks do not apply to EWCDM [13] or EHtM [32] since they assumed using an almost-xor-universal hash and didn't propose concrete instance of the hash function. Their schemes are secure as claimed when instantiating with a proper hash function.

4 Issues in Previous Analyses and Possible Fixes

In this section, we first revisit the properties of the underlying hash function that are required in security analyses of constructions nPolyMAC, CWC+, PolyMAC, 2k-PolyMAC, PDM*MAC, 1k-PDM*MAC and nEHtM_p^+, and show that the polynomial hash Poly fails to meet some of these properties. The failure of Poly is the source reason that their security analyses are flawed since all of their beyond-birthday-bound proofs rely on these properties. We then propose two polynomial hash functions called PolyX and GHASHX, and prove that they satisfy these properties. By instantiating these constructions with either of these two hash functions, their beyond-birthday-bound security can be restored.

4.1 Properties of the Hash Function

There are three properties that are required for the underlying hash function, namely regular, almost xor universal and 3-way regular properties. The almost-xor-universal property is required in security analyses of all constructions, i.e.,

$\mathsf{nPolyMAC}$, $\mathsf{CWC+}$, $\mathsf{PolyMAC}$, $\mathsf{2k\text{-}PolyMAC}$, $\mathsf{PDM^*MAC}$, $\mathsf{1k\text{-}PDM^*MAC}$ and nEHtM_p^+, while the regular property is needed in $\mathsf{nPolyMAC}$, $\mathsf{PDM^*MAC}$, $\mathsf{1k\text{-}PDM^*MAC}$, nEHtM_p^+ and multi-user $\mathsf{2k\text{-}PolyMAC}$, and the 3-way-regular property is needed in $\mathsf{nPolyMAC}$, $\mathsf{PDM^*MAC}$ and $\mathsf{1k\text{-}PDM^*MAC}$. We introduce them as follows.

Definition 1 (regular). *Let $\mathcal{K}_h$ and $\mathcal{X}$ be two non-empty finite sets. A keyed hash function $H : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$ is said to be ϵ_1-regular, if for any $X \in \mathcal{X}$ and $\mathit{\Delta} \in \{0,1\}^n$,*

$$\Pr\left[\, K_h \leftarrow_{\$} \mathcal{K}_h : H_{K_h}(X) = \mathit{\Delta} \,\right] \le \epsilon_1 \ \ .$$

Definition 2 (almost xor universal). [7] *Let $\mathcal{K}_h$ and $\mathcal{X}$ be two non-empty finite sets. A keyed hash function $H : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$ is said to be ϵ_2-almost-xor-universal, if for any distinct $X_1, X_2 \in \mathcal{X}$ and for any $\mathit{\Delta} \in \{0,1\}^n$,*

$$\Pr\left[\, K_h \leftarrow_{\$} \mathcal{K}_h : H_{K_h}(X_1) \oplus H_{K_h}(X_2) = \mathit{\Delta} \,\right] \le \epsilon_2 \ \ .$$

Definition 3 (3-way regular). *Let $\mathcal{K}_h$ and $\mathcal{X}$ be two non-empty finite sets. A keyed hash function $H : \mathcal{K}_h \times \mathcal{X} \to \{0,1\}^n$ is said to be ϵ_3-3-way-regular, if for any distinct $X_1, X_2, X_3 \in \mathcal{X}$ and for any non-zero $\mathit{\Delta} \in \{0,1\}^n$,*

$$\Pr\left[\, K_h \leftarrow_{\$} \mathcal{K}_h : H_{K_h}(X_1) \oplus H_{K_h}(X_2) \oplus H_{K_h}(X_3) = \mathit{\Delta} \,\right] \le \epsilon_3 \ \ .$$

We then recall a proposition in [16,17,14,15] that erroneously shows that Poly meets all of these three properties and discuss what is wrong.

Proposition 1. [16,17][8] *Let $\mathsf{Poly} : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ be a hash function defined as follows: For a key $K_h \in \{0,1\}^n$ and a message $M \in \{0,1\}^*$, 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 $M^* = M_1 \parallel M_2 \parallel \ldots \parallel M_\ell$ where $|M_i| = n$ for each i. Then we define*

$$\mathsf{Poly}_{K_h}(M) = M_1 \cdot K_h^{\ell} \oplus M_2 \cdot K_h^{\ell-1} \oplus \ldots \oplus M_\ell \cdot K_h \ \ ,$$

where ℓ is the number of n-bit blocks. Then, Poly is ϵ_1-regular, ϵ_2-almost-xor-universal, and ϵ_3-3-way-regular where $\epsilon_1 = \epsilon_2 = \epsilon_3 = \ell_{\max}/2^n$ and $\ell_{\max}$ denotes the maximum number of n-bit blocks of a message.

However, Poly is not almost xor universal as shown by the following counterexample. For any two distinct messages M and M' such that $M \in \{0,1\}^*$ and $M' = (0^n)_i \parallel M$ for any $i \ge 1$, the equation $\mathsf{Poly}_{K_h}(M) \oplus \mathsf{Poly}_{K_h}(M') = 0^n$ always holds since

$$\mathsf{Poly}_{K_h}(M') = 0^n \cdot K_h^{\ell+i} \oplus \ldots \oplus 0^n \cdot K_h^{\ell+1} \oplus M_1 \cdot K_h^{\ell} \oplus \ldots \oplus M_\ell 10^* \cdot K_h = \mathsf{Poly}_{K_h}(M) \ \ .$$

[7] In [27], it only requires the polynomial hash function to be universal, namely the probability that $H_{K_h}(X_1) = H_{K_h}(X_2)$ is negligible for two different messages X_1 and X_2. The almost xor universal implies universal since we can choose $\mathit{\Delta} = 0^n$.

[8] This proposition appears as Lemma 4 in [15] without the ϵ_3-3-way-regular property.

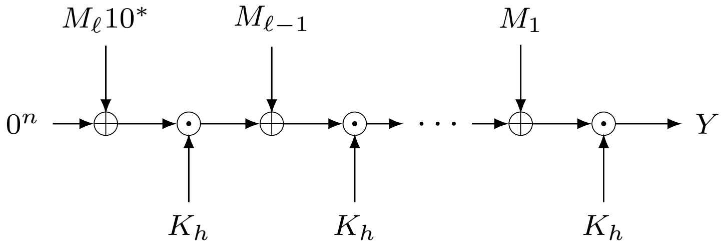

Fig. 3: The polynomial hash function PolyX for a message $M = M_1 \parallel \ldots \parallel M_\ell$ with a hash key K_h.

In the proof of Proposition 1 [16,17,15], the authors argued that the equation $\mathsf{Poly}_{K_h}(M) \oplus \mathsf{Poly}_{K_h}(M') \oplus \Delta = 0^n$ is a non-trivial polynomial of K_h with degree at most ℓ and thus the maximum number of roots of this polynomial is ℓ. They then claimed that the almost-xor-universal advantage of Poly is $\ell/2^n$. However, they overlooked the fact that the multiplication has a fixed point 0^n. By prepanding arbitrary 0^n blocks to the message M to become another message M', the above equation becomes trivial and always holds for $\Delta = 0^n$ regardless of the value of K_h. All the security analyses of constructions, i.e., nPolyMAC, CWC+, PolyMAC, 2k-PolyMAC, PDM*MAC, 1k-PDM*MAC and nEHtM_p^+, rely on this result to prove the beyond-birthday-bound security, and thus are flawed.

Remark 3. There is another polynomial hash function proposed by Minematsu and Iwata [33] that is different from Poly. Given a message $M = M_1 \| M_2 \| \ldots \| M_\ell$ where $|M_i| = n$ for $1 \le i \le \ell - 1$ and $0 \le |M_\ell| \le n$, it is defined as

$$\mathsf{Poly}'_{K_h}(M) = M_1 \cdot K_h^\ell \oplus M_2 \cdot K_h^{\ell-1} \oplus \ldots \oplus M_\ell 0^* \cdot K_h$$

where 0^* denotes the padding that appends as few 0 bits so that the length of string to be a multiple of n. As emphasized by the authors [33], this hash function is $\ell/2^n$-almost-xor-universal only for fixed-length inputs. Thus, it cannot be used for variable-length messages.

4.2 Two Possible Fixes

We now propose two polynomial hash functions and prove that they meet the above three properties. Thus, by using either of these two hash functions, the beyond-birthday-bound security of these constructions can be restored.

The first hash function called PolyX is a variant of Poly that reverses the order of a message in the polynomial. Let the message be $M = M_1 \| M_2 \| \ldots \| M_\ell$ where $|M_i| = n$ for $1 \le i \le \ell - 1$ and $0 \le |M_\ell| \le n - 1$. Then given a key $K_h \in \{0,1\}^n$ and using 10^* padding, we define

$$\mathsf{PolyX}_{K_h}(M) = M_1 \cdot K_h \oplus M_2 \cdot K_h^2 \oplus \ldots \oplus M_\ell 10^* \cdot K_h^\ell \ . \tag{2}$$

A pictorial illustration of PolyX is given in Figure 3. Compared to Poly, a notable difference is that the length-dependent term $M_\ell 10^* \cdot K_h^\ell$ will never be zeroed out since $M_\ell 10^*$ is always a non-zero value. Hence, it prevents the attacks shown in section 3 since extending the number of 0^n blocks implicitly changes the length of a message and thus the value of $M_\ell 10^* \cdot K_h^\ell$. The following lemma shows that indeed PolyX meets regular, almost xor universal and 3-way regular properties.

Lemma 1. *Let* $\mathsf{PolyX} : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ *be defined by Equation 2. Then* PolyX *is* ϵ_1*-regular,* ϵ_2*-almost-xor-universal, and* ϵ_3*-3-way-regular where* $\epsilon_1 = \epsilon_2 = \epsilon_3 = \ell_{\max}/2^n$ *and* $\ell_{\max}$ *is the maximum number of n-bit blocks of a message.*

Proof. We first consider the regular property. Given a message $M \in \{0,1\}^*$ and a constant value $\Delta \in \{0,1\}^n$, it requires the equation $\mathsf{PolyX}_{K_h}(M) \oplus \Delta = 0^n$ holds, namely

$$M_1 \cdot K_h \oplus M_2 \cdot K_h^2 \oplus \ldots \oplus M_\ell 10^* \cdot K_h^\ell \oplus \Delta = 0^n \ .$$

This is a non-trivial polynomial of K_h of degree ℓ, since the coefficient $M_\ell 10^*$ of K_h^ℓ is always non-zero. Hence the maximum number of roots of this polynomial is ℓ. Thus, the regular advantage becomes $\ell/2^n \leq \ell_{\max}/2^n$ since the hash key K_h is chosen uniformly at random from the set $\{0,1\}^n$.

We then analyze the almost xor universal property. Given two distinct messages $M, M' \in \{0,1\}^*$ and a constant value $\Delta \in \{0,1\}^n$, it implies the equation $\mathsf{PolyX}_{K_h}(M) \oplus \mathsf{PolyX}_{K_h}(M') \oplus \Delta = 0^n$, i.e.,

$$M_1 \cdot K_h \oplus \ldots \oplus M_\ell 10^* \cdot K_h^\ell \oplus M_1' \cdot K_h \oplus \ldots \oplus M_{\ell'}' 10^* \cdot K_h^{\ell'} \oplus \Delta = 0^n \ .$$

If $\ell = \ell'$, then either there exists some $1 \leq i \leq \ell - 1$ such that $M_i \neq M_i'$ or $M_\ell 10^* \neq M_{\ell'}' 10^*$. Hence this is a non-trivial polynomial of K_h of degree at most ℓ, since either the coefficient $M_i \oplus M_i'$ of K_h^i or the coefficient $M_\ell 10^* \oplus M_\ell' 10^*$ of K_h^ℓ is non-zero. Thus, the almost xor universal advantage is at most $\ell/2^n \leq \ell_{\max}/2^n$. If $\ell \neq \ell'$, without loss of generality, we assume $\ell > \ell'$. Then the coefficient $M_\ell 10^*$ of K_h^ℓ is non-zero that implies this is a non-trivial polynomial of K_h of degree ℓ. Hence the almost xor universal advantage is again at most $\ell/2^n \leq \ell_{\max}/2^n$.

Finally we consider the 3-way regular property. Given three distinct messages $M, M', M'' \in \{0,1\}^*$ and a *non-zero* constant $\Delta \in \{0,1\}^n$, it requires the equation $\mathsf{PolyX}_{K_h}(M) \oplus \mathsf{PolyX}_{K_h}(M') \oplus \mathsf{PolyX}_{K_h}(M'') = \Delta$ holds. If the left part reduces to a zero polynomial such that each coefficient is zero, which is possible by certain choice of messages, e.g., $M = M' \oplus M''$, then the 3-way regular advantage is 0 since $\Delta \neq 0^n$. Otherwise, the left part is a non-trivial polynomial since there is at least one K_h^i whose coefficient is non-zero. Hence, the 3-way regular advantage is at most $\max\{\ell, \ell', \ell''\}/2^n \leq \ell_{\max}/2^n$.

The second hash function called GHASHX is a variant of GHASH [30,29,28] by replacing the 0^* padding with the 10^* padding. Before that, we first recall

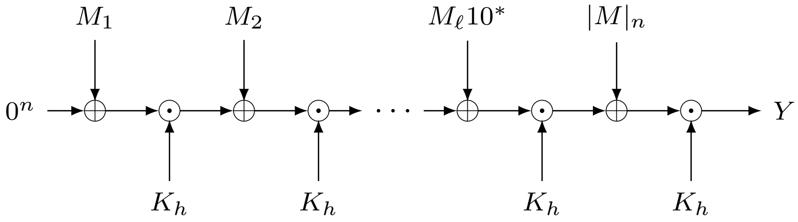

Fig. 4: The polynomial hash function GHASHX for a message $M = M_1 \| \ldots \| M_\ell$ with a hash key K_h.

the definition of GHASH and show that it cannot meet the regular property, and thus cannot be used in nPolyMAC, PDM*MAC, 1k-PDM*MAC, nEHtM$_p^+$ and multi-user 2k-PolyMAC to restore the beyond-birthday-bound security. Let the message be $M = M_1 \| M_2 \| \ldots \| M_\ell$ where $|M_i| = n$ for $1 \leq i \leq \ell - 1$ and $0 \leq |M_\ell| \leq n$. Given a hash key $K_h \in \{0,1\}^n$, GHASH is defined as follows:

$$\mathsf{GHASH}_{K_h}(M) = M_1 \cdot K_h^{\ell+1} \oplus M_2 \cdot K_h^{\ell} \oplus \ldots \oplus M_\ell 0^* \cdot K_h^2 \oplus |M|_n \cdot K_h \quad (3)$$

where 0^* is the padding method that appends as few zeros to make the total string length a multiple of n [9], and $|M|_n$ is the n-bit encoding of the length of message M. Note that GHASH explicitly multiplies the n-bit encoding of the length of a message $|M|_n$ by K_h. Hence, it can prevent the attack of prepending arbitrary 0^n blocks in section 3 since then the length will change. McGrew and Viega [29,28] showed that GHASH is a ϵ_2-almost-xor-universal hash where $\epsilon_2 = (\ell_{\max} + 1)/2^n$ and $\ell_{\max}$ is the maximum number of n-bit blocks of a message. Hence, GHASH can be used in nEHtM, CWC+, PolyMAC, and 2k-PolyMAC to restore their beyond-birthday-bound security since for these constructions, they only require the underlying hash function to be almost-xor-universal. However, for nPolyMAC, PDM*MAC, 1k-PDM*MAC, nEHtM$_p^+$ and multi-user 2k-PolyMAC, their security analyses require that the underlying hash function should be also regular. Apparently, GHASH is not regular since for an empty string $M = \varepsilon$, $\mathsf{GHASH}_{K_h}(\varepsilon) = 0$ always holds. Hence, we cannot use GHASH in these three constructions otherwise it will violate their security analyses. Even worse, if we use GHASH in nPolyMAC, then there is a forgery attack since the tuple $(N, M, T) = (0^n, \varepsilon, 0^n)$ can always pass the decryption oracle without queried before.

Now we define GHASHX that is a variant of GHASH by replacing the 0^* padding with 10^* padding. Given a hash key $K_h \in \{0,1\}^n$ and a message $M = M_1 \| M_2 \| \ldots \| M_\ell$ where $|M_i| = n$ for $1 \leq i \leq \ell - 1$ and $0 \leq |M_\ell| \leq n - 1$, GHASHX is defines as follows:

$$\mathsf{GHASHX}_{K_h}(M) = M_1 \cdot K_h^{\ell+1} \oplus M_2 \cdot K_h^{\ell} \oplus \ldots \oplus M_\ell 10^* \cdot K_h^2 \oplus |M|_n \cdot K_h \quad (4)$$

[9] The number of zeros padded is 0 if the length of original message is already a multiple of n.

where 10^* is the padding method that first appends a single 1 followed by as few zeros to make the total string length a multiple of n. A pictorial illustration of GHASHX is given in Figure 4. The following lemma shows that GHASHX meets regular, almost-xor-universal and 3-way-regular properties, and thus can be instantiated in all of these constructions to restore their beyond-birthday-bound security. The proof of this lemma is similar to the one of Lemma 2. For the sake of completeness, we provide it in Appendix B.

Lemma 2. *Let* $\mathsf{GHASHX} : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ *be defined by Equation 4. Then* GHASHX *is* ϵ_1*-regular,* ϵ_2*-almost-xor-universal, and* ϵ_3*-3-way-regular where* $\epsilon_1 = \epsilon_2 = \epsilon_3 = (\ell_{\max} + 1)/2^n$ *and* $\ell_{\max}$ *is the maximum number of n-bit blocks of a message.*

Remark 4. Compared to GHASHX, PolyX uses the reverse order of a message to implicitly encode the length of a message as a parameter in the polynomial that will not be zeroed out. While GHASHX explicitly multiplies the length of a message by K_h, and thus requires one additional multiplication. On the other hand, GHASHX can be computed efficiently in the on-the-fly manner by using Honer's rule [34]. But if we want to compute PolyX efficiently by using Honer's rule, then we need to wait until the last block of a message arrives.

Remark 5. Note that both PolyX and GHASHX are 1-key n-multiplication polynomial hash functions over $\mathrm{GF}(2^n)$, the same as Poly. There are other types of polynomial hash functions requiring multiple keys, using $n/2$ multiplications, or over prime fields. We refer to [5,22] for a detailed discussion of these polynomial hash functions. Since our main focus is to propose some possible fixes by using 1-key n-multiplication polynomial hash function over $\mathrm{GF}(2^n)$ as Poly, we leave it as the future work to investigate whether these hash functions can meet all of these three properties that are discussed in this section. Moreover, as the security and the performance of polynomial hash functions could have a significant impact on actual deployment, we leave it as another interesting and important future work to comparing these two newly proposed polynomial hash functions with existing ones (e.g., [5,22] or the international standard ISO/IEC 9797-3:2011 [23]).

5 Conclusion

In this paper, we demonstrate forgery attacks on several polynomial-hash-based MACs with provably beyond-birthday-bound security, namely nPolyMAC, CWC+, PolyMAC, 2k-PolyMAC, PDM*MAC, 1k-PDM*MAC and nEHtM_p^+. Our attacks exploit vulnerabilities in the underlying polynomial hash function, and require only one authentication query and one verification query with succeed probability of 1. Thus, our attacks disprove their high security claims. We then propose two new polynomial hash functions called PolyX and GHASHX, and prove that using either of two hash functions can fix the issues in these schemes, and thus can restore their beyond-birthday-bound security.

Acknowledgments

François-Xavier Standaert is senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work has been funded in parts by the ERC consolidator grant 724725 (acronym SWORD), the ERC Advanced Grant 101096871 (acronym BRIDGE), and by National Key Research and Development Program of China (No. 2019YFB2101600). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

A Overview of the Usage of Polynomial Hash Function

In this section, we briefly recall how the polynomial hash function Poly defined in Equation 1 is used in schemes nPolyMAC [16], CWC+ [20], PolyMAC [14,27], 2k-PolyMAC [14,15], PDM*MAC [11], 1k-PDM*MAC [11], and nEHtM$_p^+$ [12]. In [16], the authors first showed the beyond-birthday-bound security of construction DWCDM. They then proposed nPolyMAC as a concrete instance of DWCDM by using Poly as the underlying hash function. They proved the beyond-birthday-bound security of nPolyMAC by combining the result of DWCDM and the properties of Poly. The required properties of Poly are ϵ_1-regular, ϵ_2-almost-xor-universal and ϵ_3-3-way-regular, and are proved in [16,17, Proposition 1]. In [20], the polynomial hash Poly was directly integrated into CWC+. The beyond-birthday-bound security analysis of CWC+ relies on the ϵ_2-almost-xor-universal property of Poly. In both PolyMAC and 2k-PolyMAC [14,27,15], the polynomial hash Poly was the main component of these two schemes. The beyond-birthday-bound security analyses of PolyMAC and 2k-PolyMAC require Poly to be ϵ_2-almost-xor-universal, while the multi-user beyond-birthday-bound security analysis of 2k-PolyMAC additionally requires Poly to be ϵ_1-regular. The authors [15, Lemma 4] proved that Poly was ϵ_1-regular and ϵ_2-almost-xor-universal. In [11], the authors first proved the beyond-birthday-bound security of PDM*MAC and 1k-PDM*MAC. They then explicitly used Poly to instantiate these two schemes to achieve the beyond-birthday-bound security. The required properties of Poly are ϵ_1-regular, ϵ_2-almost-xor-universal and ϵ_3-3-way-regular. In [12], the authors first showed the beyond-birthday-bound security of nEHtM$_p^*$. They then proposed nEHtM$_p^+$ as a concrete instance of nEHtM$_p^*$ by using Poly as the underlying hash function. They proved the beyond-birthday-bound security of nEHtM$_p^+$ by combining the result of nEHtM$_p^*$ and the ϵ_2-almost-xor-universal property of Poly.

B Proof of Lemma 2

We first consider the regular property. Given a message $M \in \{0,1\}^*$ and a constant value $\Delta \in \{0,1\}^n$, it requires the equation $\mathsf{GHASHX}_{K_h}(M) \oplus \Delta = 0^n$ holds, namely

$$M_1 \cdot K_h^{\ell+1} \oplus M_2 \cdot K_h^{\ell} \oplus \ldots \oplus M_\ell 10^* \cdot K_h^2 \oplus |M| \cdot K_h \oplus \Delta = 0^n \ .$$

This is a non-trivial polynomial of K_h of degree at most $\ell+1$ because the coefficient $M_\ell 10^*$ of K_h^2 is always non-zero. The number of roots of this polynomial is at most $\ell+1$. Hence, the regular advantage is $(\ell+1)/2^n \le (\ell_{\max}+1)/2^n$ since the hash key K_h is chosen uniformly at random from the set $\{0,1\}^n$.

We then analyze the almost xor universal property. Given two distinct messages $M, M' \in \{0,1\}^*$ and a constant value $\Delta \in \{0,1\}^n$, it implies the equation $\mathsf{GHASHX}_{K_h}(M) \oplus \mathsf{GHASHX}_{K_h}(M') \oplus \Delta = 0^n$, i.e.,

$$M_1 \cdot K_h^{\ell+1} \oplus \ldots \oplus M_\ell 10^* \cdot K_h^2 \oplus |M| \cdot K_h \oplus M'_1 \cdot K_h^{\ell'+1} \oplus \ldots \oplus M'_{\ell'} 10^* \cdot K_h^2 \oplus |M'| \cdot K_h \oplus \Delta = 0^n \quad .$$

If $\ell = \ell'$, then either there exists some $1 \le i \le \ell - 1$ such that $M_i \ne M'_i$ or $M_\ell 10^* \ne M'_{\ell'} 10^*$. Hence this is a non-trivial polynomial of K_h of degree at most $\ell+1$, since either the coefficient $M_i \oplus M'_i$ of $K_h^{\ell+2-i}$ or the coefficient $M_\ell 10^* \oplus M'_{\ell'} 10^*$ of K_h^2 is non-zero. Thus, the almost xor universal advantage is at most $(\ell+1)/2^n \le (\ell_{\max}+1)/2^n$. If $\ell \ne \ell'$, then the coefficient $|M|_n \oplus |M'|_n$ of K_h is non-zero that implies this is a non-trivial polynomial of K_h of degree at most $\ell+1$. Hence the almost xor universal advantage is again at most $(\ell+1)/2^n \le (\ell_{\max}+1)/2^n$.

Finally we consider the 3-way regular property. Given three distinct messages $M, M', M'' \in \{0,1\}^*$ and a *non-zero* constant $\Delta \in \{0,1\}^n$, it requires the equation $\mathsf{GHASHX}_{K_h}(M) \oplus \mathsf{GHASHX}_{K_h}(M') \oplus \mathsf{GHASHX}_{K_h}(M'') = \Delta$ holds. If the left part reduces to a zero polynomial, then the 3-way regular advantage is 0 since $\Delta \ne 0^n$. Otherwise, the left part is a non-trivial polynomial since there is at least one K_h^i whose coefficient is non-zero. Hence, the 3-way regular advantage is at most $\max\{\ell+1, \ell'+1, \ell''+1\}/2^n \le (\ell_{\max}+1)/2^n$.

References

1. Iso/iec 9797-3:2011 information technology – security techniques – message authentication codes (macs) – part 3: Mechanisms using a universal hash-function. International Standard ISO/IEC 9797-3 (2011)
2. Iso/iec 19772:2020 information security – authenticated encryption. International Standard ISO/IEC 19772 (2020)
3. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. pp. 321–345 (2017), https://doi.org/10.1007/978-3-319-66787-4_16
4. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. pp. 531–545 (2000), https://doi.org/10.1007/3-540-44448-3_41
5. Bernstein, D.J.: Polynomial evaluation and message authentication. URL: http://cr. yp. to/papers. html# pema. ID b1ef3f2d385a926123e 1517392e20f8c. Citations in this document **2** (2007)

6. Bhargavan, K., Leurent, G.: On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and openvpn. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 456–467 (2016), https://doi.org/10.1145/2976749.2978423
7. Bhattacharya, S., Nandi, M.: Revisiting variable output length XOR pseudorandom function. IACR Trans. Symmetric Cryptol. **2018**(1), 314–335 (2018), https://doi.org/10.13154/tosc.v2018.i1.314-335
8. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings. pp. 450–466 (2007), https://doi.org/10.1007/978-3-540-74735-2_31
9. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. pp. 208–225 (2012), https://doi.org/10.1007/978-3-642-34961-4_14
10. Cannière, C.D., Dunkelman, O., Knezevic, M.: KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers. In: Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings. pp. 272–288 (2009), https://doi.org/10.1007/978-3-642-04138-9_20
11. Chakraborti, A., Nandi, M., Talnikar, S., Yasuda, K.: On the composition of single-keyed tweakable even-mansour for achieving BBB security. IACR Trans. Symmetric Cryptol. **2020**(2), 1–39 (2020), https://doi.org/10.13154/tosc.v2020.i2.1-39
12. Chen, Y.L., Dutta, A., Nandi, M.: Multi-user BBB security of public permutations based MAC. Cryptogr. Commun. **14**(5), 1145–1177 (2022), https://doi.org/10.1007/s12095-022-00571-w
13. Cogliati, B., Seurin, Y.: EWCDM: an efficient, beyond-birthday secure, nonce-misuse resistant MAC. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. pp. 121–149 (2016), https://doi.org/10.1007/978-3-662-53018-4_5
14. Datta, N., Dutta, A., Nandi, M., Paul, G.: Double-block hash-then-sum: A paradigm for constructing BBB secure PRF. IACR Trans. Symmetric Cryptol. **2018**(3), 36–92 (2018), https://doi.org/10.13154/tosc.v2018.i3.36-92
15. Datta, N., Dutta, A., Nandi, M., Talnikar, S.: Tight multi-user security bound of sfdbhts. IACR Trans. Symmetric Cryptol. **2023**(1), 192–223 (2023), https://doi.org/10.46586/tosc.v2023.i1.192-223
16. Datta, N., Dutta, A., Nandi, M., Yasuda, K.: 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. pp. 631–661 (2018), https://doi.org/10.1007/978-3-319-96884-1_21
17. Datta, N., Dutta, A., Nandi, M., Yasuda, K.: Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based MAC. IACR Cryptol. ePrint Arch. p. 500 (2018), https://eprint.iacr.org/2018/500

18. Dutta, A., Jha, A., Nandi, M.: Tight security analysis of ehtm MAC. IACR Trans. Symmetric Cryptol. **2017**(3), 130–150 (2017), https://doi.org/10.13154/tosc.v2017.i3.130-150
19. Dutta, A., Nandi, M., Talnikar, S.: Beyond birthday bound secure MAC in faulty nonce model. IACR Cryptol. ePrint Arch. p. 127 (2019), https://eprint.iacr.org/2019/127
20. Dutta, A., Nandi, M., Talnikar, S.: 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. pp. 437–466 (2019), https://doi.org/10.1007/978-3-030-17653-2_15
21. Dworkin, M.: Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. NIST Special Publication 800-38D (2007)
22. Handschuh, H., Preneel, B.: Key-recovery attacks on universal hash function based MAC algorithms. In: Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. pp. 144–161 (2008), https://doi.org/10.1007/978-3-540-85174-5_9
23. Information technology Security techniques Message Authentication Codes (MACs) Part 3: Mechanisms using a universal hash-function. Iso/iec 9797-3:2011 (2011)
24. Iwata, T.: New blockcipher modes of operation with beyond the birthday bound security. In: Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers. pp. 310–327 (2006), https://doi.org/10.1007/11799313_20
25. Iwata, T.: New blockcipher modes of operation with beyond the birthday bound security. IACR Cryptol. ePrint Arch. p. 188 (2006), http://eprint.iacr.org/2006/188
26. Jean, J.: TikZ for Cryptographers. https://www.iacr.org/authors/tikz/ (2023)
27. Kim, S., Lee, B., Lee, J.: Tight security bounds for double-block hash-then-sum macs. In: Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I. pp. 435–465 (2020), https://doi.org/10.1007/978-3-030-45721-1_16
28. McGrew, D.A., Viega, J.: The security and performance of the galois/counter mode (GCM) of operation. In: Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings. pp. 343–355 (2004), https://doi.org/10.1007/978-3-540-30556-9_27
29. McGrew, D.A., Viega, J.: The security and performance of the galois/counter mode of operation (full version). IACR Cryptol. ePrint Arch. p. 193 (2004), http://eprint.iacr.org/2004/193
30. McGrew, D.A., Viega, J.: The use of galois message authentication code (GMAC) in ipsec ESP and AH. RFC **4543**, 1–14 (2006), https://doi.org/10.17487/RFC4543
31. Mennink, B., Neves, S.: Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In: Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. pp. 556–583 (2017), https://doi.org/10.1007/978-3-319-63697-9_19

32. Minematsu, K.: How to thwart birthday attacks against macs via small randomness. In: Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers. pp. 230–249 (2010), https://doi.org/10.1007/978-3-642-13858-4_13
33. Minematsu, K., Iwata, T.: 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. pp. 391–412 (2011), https://doi.org/10.1007/978-3-642-25516-8_24
34. Shoup, V.: On fast and provably secure message authentication based on universal hashing. In: Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. pp. 313–328 (1996), https://doi.org/10.1007/3-540-68697-5_24
35. Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. **22**(3), 265–279 (1981), https://doi.org/10.1016/0022-0000(81)90033-7