Ubiquitous Weak-key Classes of BRW-polynomial Function

Kaiyan Zheng[1,2,3] and Peng Wang[1,2,3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
[3] School of Cyber Security, University of Chinese Academic Science, Beijing 100049, China
`zhengkaiyan@iie.ac.cn,wp@is.ac.cn`

Abstract. BRW-polynomial function is suggested as a preferred alternative of polynomial function, owing to its high efficiency and seemingly non-existent weak keys. In this paper we investigate the weak-key issue of BRW-polynomial function as well as BRW-instantiated cryptographic schemes. Though, in BRW-polynomial evaluation, the relationship between coefficients and input blocks is indistinct, we give out a recursive algorithm to compute another $(2^{v+1}-1)$-block message, for any given $(2^{v+1}-1)$-block message, such that their output-differential through BRW-polynomial evaluation, equals any given s-degree polynomial, where $v \geq \lfloor \log_2(s+1) \rfloor$. With such algorithm, we illustrate that any non-empty key subset is a weak-key class in BRW-polynomial function. Moreover any key subset of BRW-polynomial function, consisting of at least 2 keys, is a weak-key class in BRW-instantiated cryptographic schemes like the Wegman-Carter scheme, the UHF-then-PRF scheme, DCT, etc. Especially in the AE scheme DCT, its confidentiality, as well as its integrity, collapses totally, when using weak keys of BRW-polynomial function, which are ubiquitous.

Keywords. Weak key, polynomial evaluation hash, BRW-polynomial, DCT, message authentication code, authenticated encryption.

1 Introduction

Universal hash function. Universal hash functions (short as UHFs) were firstly introduced by Carter and Wegman [8,37], and have become common components in numerous cryptographic constructions, like message authentication code (short as MAC) schemes [13,11,13,7], tweakable enciphering schemes [19,35,10] and authenticated encryption (short as AE) schemes [21,3], etc. A UHF is a keyed function. Compared with other primitives like pseudorandom permutations (short as PRPs) and pseudorandom functions (short as PRFs), UHFs have no strength of pseudorandomness. The only requirement is some simple combinatorial properties, which makes UHFs high-performance but brittle and vulnerable to weak-key analyses [14,27,25,39,1] and related-key attacks [34,36].

Weak-key analysis. Handschuh and Preneel [14] initiated the study of the weak-key issue of UHFs, as they pointed out that "in symmetric cryptology, a class of keys is called a weak-key class if for the members of that class the algorithm *behaves in an unexpected way* and if it is easy to *detect* whether a particular unknown key belongs to this class. Moreover, if a weak-key class is of size C, one requires that identifying that a key belongs to this class requires testing fewer than C keys by exhaustive search and fewer than C verification queries." Following such definition, they investigated several weak-key classes of UHFs in MACs. Later on the weak-key analyses of UHFs mainly focused on a specific UHF, i.e. polynomial function.

Polynomial function. Polynomial function, which evaluates a polynomial in the key with the data blocks as coefficients, is one of the most widely used UHFs [5,20,17,35,10,4,15]. However the weak-key issue of polynomial function in cryptographic schemes such as MACs was extensively studied and was found unavoidable, especially in the example of GCM/GMAC which uses polynomial function in its authentication component. Saarrinen [27] found that the keys of polynomial function satisfying $K^t = K$ formed a weak-key class in GCM. Procter and Cid [25] found that any subset $\mathcal{W}$ is a weak-key class in GCM and GMAC, if $|\mathcal{W}| \geq 3$ or $|\mathcal{W}| \geq 2$ and $0 \in \mathcal{W}$, exploiting the so-called forgery polynomial $q(K) = \Sigma_{H \in \mathcal{W}}(K - H)$. Zhu, Tan and Gong [39] pointed out that any subset $\mathcal{W}$ consisting of at least 2 keys is a weak-key class. Sun, Wang and Zhang [34] applied the above results to tweakable enciphering schemes based on polynomial function. Abdelraheem, Beelen, Bogdanov and Tischhauser [1] further proposed twisted polynomials from Ore rings to construct sparse forgery polynomials, which greatly facilitate key recovery attacks.

The weak-key issue casts shadow on the further application of polynomial function. For example, during the CAESAR competition, due to the weak-key issue of polynomial function in the AE scheme POET [2], the designers [3] decided to abandon the polynomial-function-based POET and retain the four-round-AES-based version.

BRW-polynomial function. Bernstein [6] proposed a variant of polynomial function, after the work of Rabin and Winograd [26], which is named as BRW (short for Bernstein-Rabin-Winograd) in [28]. BRW-polynomial function performs more highly-efficient than polynomial function, as it decreases nearly a half of multiplications in polynomial evaluation. BRW-polynomial function is widely-used in lots of cryptographic schemes, including authentication schemes [6,30], tweakable enciphering schemes [28,29,9], authenticated encryption schemes [12], etc.

Furthermore, unlike the case of polynomial function, the weak-key issue of BRW-polynomial function seems avoidable. By now, no weak-key problem of BRW-polynomial function has been found [14,12], which makes BRW-polynomial function an ideal UHF candidate in cryptographic schemes to alleviate the threat of weak keys. For example, the designers of DCT, a deterministic authenticated encryption scheme [12], suggested using BRW-polynomial function to instantiate its UHF to avoid the weak-key issue.

Our contributions. This work investigates the weak-key problem of BRW-polynomial function and BRW-instantiated schemes. Unlike polynomial function, in BRW-polynomial evaluation, the relationship between coefficients and input blocks is indistinct owing to its recursive definition. Nevertheless we give out a recursive algorithm -*SumBRWpoly*- which, for any given $(2^{v+1}-1)$-block message M and any given s-degree polynomial $q(K) = Q_0K^s+Q_1K^{s-1}+\cdots+Q_s$ that $v \geq \lfloor \log_2(s+1) \rfloor$, computes another $(2^{v+1}-1)$-block message M' such that $BRW_K(M') = BRW_K(M) + q(K)$.

With *SumBRWpoly*, we illustrate that any s-key subset $\mathcal{W} = \{H_0, \cdots, H_{s-1}\}$ is a weak-key class of BRW-polynomial function. Moreover similar to the case of polynomial function, any $\mathcal{W}$, as long as $s \geq 2$, is also a weak-key class in BRW-instantiated schemes, even when padding rules are taken into consideration, which negates the suggestion of substituting BRW-polynomial function for polynomial function to mitigate the weak-key threat.

For example, when instantiating with BRW-polynomial, both the Wegman-Carter scheme and the UHF-then-PRF scheme suffer the forgery attack if the UHF key falls into $\mathcal{W}$, and it is easy to detect if the unknown UHF key belongs to $\mathcal{W}$. Furthermore, the BRW-instantiated DCT, a deterministic AE scheme, suffers both the distinguishing attack and the forgery attack once its UHF key is in $\mathcal{W}$, implying that the confidentiality, as well as the integrity, of DCT totally collapses when using weak keys of BRW-polynomial, which are ubiquitous.

The remaining of the paper is structured as following: after reviewing the weak-key problem of polynomial-based MACs in Section 2, *SumBRWpoly* is illustrated in Section 3, together with ubiquitous weak keys of BRW-polynomial and BRW-instantiated MACs. Section 4 discuss weak-key classes of DCT, and Section 5 makes a simple conclusion of this work.

2 Preliminaries

2.1 Notations

For a finite set $\mathcal{S}$, let $x \overset{\$}{\leftarrow} \mathcal{S}$ denote selecting an element x uniformly at random from the set $\mathcal{S}$ and $\#\mathcal{S}$ denote the number of members in $\mathcal{S}$. For $b \in \{0,1\}$, b^m denotes m bits of b. For a function $H : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ where $\mathcal{K}$ is a key space, we often write $H(K, M)$ as $H_K(M)$, where $(K, M) \in \mathcal{K} \times \mathcal{D}$. Without loss of generality, most of operations, such as additions, multiplications, in the remaining are defined over the finite filed $\mathbb{GF}(2^n)$. Let $\|$ denote the concatenation of two bit-strings, and $\Longleftrightarrow$ means if and only if. $M = M_0 \cdots M_{m-1}$ is a m-block message where $M_i \in \mathbb{GF}(2^n)$ for $i = 0, \cdots, m-1$.

2.2 Universal hash functions

Two commonly-used UHFs are almost-universal (AU) hash function and almost-XOR-universal (AXU) hash function. Both UHFs satisfy some simple combinatorial properties for *any* two different inputs.

For AU hash function, the output-collision probability of any two different inputs is negligible.

Definition 1 (AU [32]). *$H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is an ϵ-almost-universal (ϵ-AU) hash function, if for any $M, M' \in \mathcal{D}$, $M \neq M'$,*

$$\Pr[K \overset{\$}{\leftarrow} \mathcal{K} : H_K(M) = H_K(M')] = \frac{\#\{K \in \mathcal{K} : H_K(M) = H_K(M')\}}{\#\mathcal{K}} \leq \epsilon.$$

When ϵ is negligible we say that H is AU. Generally, $\epsilon = \max_{M \neq M'} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : H_K(M) = H_K(M')]$.

For AXU hash function, the output-differential distribution of any two different inputs is almost uniform.

Definition 2 (AXU [33]). *Let $(\mathcal{R}, +)$ be an abelian group where the addition is exclusive-OR (XOR). $H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is an ϵ-almost-xor-universal (ϵ-AXU), if for any $M, M' \in \mathcal{D}$, $M \neq M'$, and any $C \in \mathcal{R}$,*

$$\Pr[K \overset{\$}{\leftarrow} \mathcal{K} : H_K(M) + H_K(M') = C] = \frac{\#\{K \in \mathcal{K} : H_K(M) + H_K(M') = C\}}{\#\mathcal{K}} \leq \epsilon.$$

When ϵ is negligible we say that H is AXU. Generally, $\epsilon = \max_{M \neq M', C} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : H_K(M) + H_K(M') = C]$.

Clearly, if H is ϵ-AXU, it is also ϵ-AU, for ϵ-AU is a special case of ϵ-AXU when $C = 0$.

2.3 UHF-based MACs

One popular design of UHF-based MACs is to firstly compress the variable-length input message into a fixed-length digest by a UHF and secondly encrypt it into a tag. For example, the Wegman-Carter scheme [37,18,31] masks the digest with the keystream of a block-cipher, while the UHF-then-PRF scheme [31] maps the digest into a tag by a PRF.

More specifically, let $H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a UHF and $E : \mathcal{K}' \times \mathcal{R} \to \mathcal{R}$ be a secure block-cipher. Two common UHF-based MACs are as following:

- The Wegman-Carter scheme $\mathrm{WC} : (\mathcal{K} \times \mathcal{K}') \times \mathcal{N} \times \mathcal{D} \to \mathcal{R}$, for $M \in \mathcal{D}, N \in \mathcal{N}$ and $K \| K' \overset{\$}{\leftarrow} \mathcal{K} \times \mathcal{K}'$,

$$\mathrm{WC}_{K,K'}(N, M) = E_{K'}(N) + H_K(M).$$

- The UHF-Then-PRF scheme UTP : $(\mathcal{K} \times \mathcal{K}') \times \mathcal{D} \to \mathcal{R}$, for $M \in \mathcal{D}$ and $K \| K' \xleftarrow{\$} \mathcal{K} \times \mathcal{K}'$,
$$\mathrm{UTP}_{K,K'}(M) = E_{K'}(H_K(M)).$$

The Security of MACs. Without loss of generality, assuming that the key is uniform-randomly chosen, i.e. $K \| K' \xleftarrow{\$} \mathcal{K} \times \mathcal{K}'$, the MAC scheme $\mathcal{O}$ often consists of two algorithms: (let $\mathcal{O} \in \{\mathrm{WC}, \mathrm{UTP}\}$)

- Tag-generation $\mathcal{T}^{\mathcal{O}}$: When $\mathcal{O} = \mathrm{WC}$, on the input (N, M) where N is non-repeated nonce, calculate $T = \mathrm{WC}_{K,K'}(N, M)$; otherwise on the input M, calculate $T = \mathrm{UTP}_{K,K'}(M)$. Return T.
- Verification $\mathcal{V}^{\mathcal{O}}$: When $\mathcal{O} = \mathrm{WC}$, on the input (N, M, T), compute $T' = \mathrm{WC}_{K,K'}(N, M)$; otherwise on the input (M, T) compute $T' = \mathrm{UTP}_{K,K'}(M)$. If $T' = T$, return 1; else return 0.

During the communication between two parties who have shared a secret key (K, K'), the sender generates tags of his messages by the tag-generation algorithm $\mathcal{T}^{\mathcal{O}}$ and transmits the message-tag pairs, while the receiver validates the received message-tag pairs when the verification algorithm $\mathcal{V}^{\mathcal{O}}$ returns 1.

The security goal of MACs is to resist the forgery attack. More specifically, any adversary who has access to both the tag-generation oracle $\mathcal{T}^{\mathcal{O}}$ and the verification oracle $\mathcal{V}^{\mathcal{O}}$, is said to have made a successful forgery, once it outputs a new message-tag pair, i.e. a triple (N, M, T) when $\mathcal{O} = \mathrm{WC}$ or a duplet (M, T) when $\mathcal{O} = \mathrm{UTP}$, which is not produced by $\mathcal{T}^{\mathcal{O}}$ but is validated by $\mathcal{V}^{\mathcal{O}}$.

It has been proved that the Wegman-Carter scheme is secure if H is an AXU and E is a PRP [18], and that the UHF-then-PRF scheme is secure if H is an AU and E is a PRP [31].

2.4 Weak keys of polynomial function and polynomial-based MACs

Polynomial function. Polynomial function is defined as

$$Poly_K(M) = M_0 K^{m-1} + M_1 K^{m-2} + \cdots + M_{m-1}$$

where $K \in \mathbb{GF}(2^n)$, $M = M_0 M_1 \cdots M_{m-1}$, $M_i \in \mathbb{GF}(2^n)$ for $i = 0, 1, \cdots, m-1$. Obviously $Poly_K(M)$ determines a polynomial in $\mathbb{GF}(2^n)[K]$.

It is easy to deduce that $Poly_K(\cdot)$ is a $(m-1)/2^n$-AU, and that $K \cdot Poly_K(\cdot)$ is a $m/2^n$-AXU. Because for any distinct M, M' and any $C \in \mathbb{GF}(2^n)$, the equation $Poly_K(M') = Poly_K(M)$ has at most $(m-1)$ roots in $\mathbb{GF}(2^n)$, while the equation $K \cdot Poly_K(M) + K \cdot Poly_K(M') = C$ has at most m roots.

Weak-key classes of *Poly* and *Poly*-based MACs. Unfortunately, the weak-key issue of polynomial function is unavoidable. As shown in [25,39,34], any subset $\mathcal{W}$, as long as $|\mathcal{W}| \geq 2$, is a weak-key class of polynomial function in GCM and GMAC, both *Poly*-based schemes. We just give it a brief explanation in the following, and more details refer to [14,25,39,34].

For any key subset $\mathcal{W} = \{H_0, H_1, \cdots, H_{s-1}\}$ that $s \geq 2$, define

$$q(K) = (K - H_0)(K - H_1) \cdots (K - H_{s-1}) = Q_0 K^s + Q_1 K^{s-1} + \cdots + Q_s,$$

where $Q_0 = 1$. It is obvious that

$$K \in \mathcal{W} \iff q(K) = 0. \tag{1}$$

In polynomial function, each coefficient corresponds exactly each input block, and it is easy to find message pairs whose output-differential after polynomial evaluating equals $q(K)$. Specifically, for arbitrary m-block M that $m \geq s$, computes

$$\begin{aligned} Poly_K(M) + q(K) &= M'_0 K^{m-1} + M'_1 K^{m-2} + \cdots + M'_{m-1}, \\ K \cdot Poly_K(M) + q(K) &= K \cdot \left(M''_0 K^{m-1} + M''_1 K^{m-2} + \cdots + M''_{m-1}\right) + Q_s. \end{aligned}$$

Let $M' = M'_0 M'_1 \cdots M'_{m-1}$ and $M'' = M''_0 M''_1 \cdots M''_{m-1}$, and by (1),

$$K \in \mathcal{W} \iff Poly_K(M) = Poly_K(M'), \tag{2}$$

$$K \in \mathcal{W} \bigcup \{0\} \iff K \cdot Poly_K(M) = K \cdot Poly_K(M'') + Q_s. \tag{3}$$

By (2) (3), it is trivial that $\Pr\left[K \xleftarrow{\$} \mathcal{W} : Poly_K(M) = Poly_K(M')\right] = 1$ and $\Pr\left[K \xleftarrow{\$} \mathcal{W} : K \cdot Poly_K(M) = K \cdot Poly_K(M'') + Q_s\right] = 1$, which implies that the AU property of $Poly_K(\cdot)$, as well as the AXU property of $K \cdot Poly_K(\cdot)$, totally disappears in the key subset $\mathcal{W}$.

Furthermore, once the key of *Poly* falls into $\mathcal{W}$, the security of *Poly*-based schemes also collapses, and it is easy to detect whether the unknown key of *Poly* belongs to $\mathcal{W}$. Thus $\mathcal{W}$ ($|\mathcal{W}| \geq 2$) is a weak-key class of *Poly* in *Poly*-based schemes. Take two common *Poly*-based MACs, i.e. UTP and WC, as examples, that is, $\mathrm{UTP}_{K,K'}(M) = E_{K'}(Poly_K(M))$, and $\mathrm{WC}_{K,K'}(N, M) = E_{K'}(N) + K \cdot Poly_K(M)$. Since $E_{K'}$ is a PRP, according to (2) (3), it is easy to deduce that

$$K \in \mathcal{W} \iff \mathrm{UTP}_{K,K'}(M) = \mathrm{UTP}_{K,K'}(M'), \tag{4}$$

$$K \in \mathcal{W} \bigcup \{0\} \iff \mathrm{WC}_{K,K'}(N, M) = \mathrm{WC}_{K,K'}(N, M'') + Q_s, \tag{5}$$

which means that when $K \in \mathcal{W}$, neither UTP nor WC can resist the forgery attack.

From above, it is crucial that, for arbitrary key subset $\mathcal{W}$, it is easy to find message pairs whose output-differential after polynomial evaluating equals $q(K)$, the so-called forgery polynomial defined by $\mathcal{W}$. To deal with variable-length inputs in real applications, inputs to polynomial function are padded firstly. However even when padding rules are taken into consideration, such message pairs are easy to find, and examples include GCM and GMAC [27,25,39,1].

3 Weak keys of BRW-polynomial function and BRW-instantiated MACs

3.1 The description of BRW-polynomial function

BRW-polynomial function [6,23] is defined recursively, just as follows:

- $BRW_K(\varepsilon) = 0^n$;
- $BRW_K(M_0) = M_0$;
- $BRW_K(M_0M_1) = M_0K + M_1$;
- $BRW_K(M_0M_1M_2) = (M_0 + K)(M_1 + K^2) + M_2$;
- $BRW_K(M_0 \cdots M_{m-1}) = BRW_K(M_0 \cdots M_{t-2})(K^t + M_{t-1}) + BRW_K(M_t \cdots M_{m-1})$ for $t \in \{4, 8, 16, 32, \cdots\}$ and $t \leq m < 2t$ (i.e. $t = 2^{\lfloor \log_2 m \rfloor}$);

where ε is an empty string, $K \in \mathbb{GF}(2^n), M_i \in \mathbb{GF}(2^n)$ for $i = 0, \cdots, m-1$. When $m > 3$, let $t = 2^{\lfloor \log_2 m \rfloor}$, $BRW_K(\cdot)$ is a monic polynomial with the degree of $(2t-1)$. And it is easy to conclude that $BRW_K(\cdot)$ is $(2t-1)/2^n$-AU and $K \cdot BRW_K(\cdot)$ is $2t/2^n$-AXU [28].

Unlike the case of polynomial function, in BRW-polynomial evaluation, each input block may affect multiple coefficients in the meantime, and it's difficult to track the coefficients after modifying input blocks. However, even though the relationship between input blocks and coefficients is not so obvious as that in polynomial function, there are efficient methods to find message pairs whose output-differential after BRW-polynomial evaluating equals some given polynomial, and BRW-polynomial function suffers the same weak-key issue as the original polynomial function.

In the following, we firstly give out a recursive algorithm, *SumBRWpoly* in Algorithm 1, which finds another new $(2^{v+1}-1)$-block message for any given $(2^{v+1}-1)$-block message such that their output-differential after BRW-polynomial evaluating equals any given s-degree polynomial, where $v \geq \lfloor \log_2(s+1) \rfloor$. Secondly, we study the weak-key problem of BRW-polynomial function and BRW-instantiated MACs, i.e. *BRW*-based UTP and WC, with the recursive algorithm.

3.2 The description of *SumBRWpoly*.

Given any s-degree polynomial $q(K) = Q_0K^s + Q_1K^{s-1} + \cdots + Q_s$ and any m-block message M that $m = 2^{v+1} - 1$ and $v \geq \lfloor \log_2(s+1) \rfloor$, *SumBRWpoly*, exploiting the observations about the BRW-polynomial evaluation of the specific $(2^{v+1}-1)$-block inputs, computes another new m-block message M' such that $BRW_K(M')$ is exactly the sum of $BRW_K(M)$ and $q(K)$.

In this section, we first introduce the observations about the BRW-polynomial evaluation of $(2^{v+1}-1)$-block inputs, and then explain how *SumBRWpoly* works.

BRW-polynomial evaluation of $(2^{v+1}-1)$-block inputs. When $v \geq 2$, let $m = 2^{v+1} - 1$ and $t = 2^{\lfloor \log_2(m) \rfloor} = 2^v$. To arbitrary m-block message M,

$$BRW_K(M_0 \cdots M_{t-2} M_{t-1} M_t \cdots M_{2t-2}) =$$
$$BRW_K(M_0 \cdots M_{t-2}) \cdot K^t + M_{t-1} \cdot BRW_K(M_0 \cdots M_{t-2}) + BRW_K(M_t \cdots M_{2t-2}),$$

and the observations exploited in *SumBRWpoly* are as following:

(1) $BRW_K(M)$ is a monic polynomial with the degree of $(2t-1)$;
(2) Both $BRW_K(M_0 \cdots M_{t-2})$ and $BRW_K(M_t \cdots M_{2t-2})$ are monic polynomials with the degree of $(t-1)$;

(3) The last $(t-1)$ blocks of M, i.e. $M_t \cdots M_{2t-2}$, only affect the terms with a degree lower than $(t-1)$;

(4) Only the first $(t-1)$ blocks of M, i.e. $M_0 \cdots M_{t-2}$, affect the terms with a degree greater than t.

(5) Moreover, the last block of M, i.e. M_{2t-2}, only affects the constant term.

Note that when $v = 0, 1$ and $m = 1, 3$ respectively, the evaluation of $BRW_K(M)$ is simple.

How *SumBRWpoly* works. The description of *SumBRWpoly* is shown in Algorithm 1. It is required that $m > s$. Otherwise there is no such m-block message pair M, M' satisfying $BRW_K(M') = BRW_K(M) + q(K)$ since both $BRW_K(M')$ and $BRW_K(M)$ are monic polynomials with the degree of m. Besides, m is often expected to be as small as possible to make the attacks efficient. For any s, the shortest messages dealt by *SumBRWpoly* is $m = 2^{\lfloor \log_2(s+1) \rfloor + 1} - 1$, i.e. the smallest m is no larger than $2s$.

When $s = 0$. Note that $s = 0$ when $q(K)$ has only a constant term, i.e. $q(K) = Q_0, Q_0 \in \mathbb{GF}(2^n)$. To be simple, let $M'_{m-1} = M_{m-1} + Q_0$, as the last block of the $(2^{v+1}-1)$-block message only affect the constant term in BRW-polynomial evaluation for $v \geq 0$.

When $v = 1$ and $s = 1, 2$. In this case, the specific message that *SumBRWpoly* processes is of 3 blocks, i.e $m = 3$. According to

$$\begin{cases} BRW_K(M'_0 M'_1 M'_2) = K^3 + M'_0 K^2 + (M'_1 + M'_0)K + M'_2 \\ BRW_K(M_0 M_1 M_2) = K^3 + M_0 K^2 + (M_1 + M_0)K + M_2 \end{cases},$$

it is easy to define M' satisfying $BRW_K(M') = BRW_K(M) + q(K)$ for $s = 1, 2$. One simple way to define M' is given in Algorithm 1.

When $v \geq 2$. In this case, *SumBRWpoly* runs in a recursive way by exploiting the observations about the BRW-polynomial evaluation of $(2^{v+1}-1)$-block inputs.

If $s < t-1$, because the last $(t-1)$ input blocks only affect the terms with the degree lower than $(t-1)$ in BRW-polynomial evaluation, to be simple, *SumBRWpoly*$(q(K), M)$ keeps the first t blocks of M' the same as that of M, and computes the remaining $(t-1)$ blocks of M' by making a recursive call of *SumBRWpoly*$(q(K), M_t \cdots M_{2t-2})$. That is,

$$\mathit{SumBRWpoly}\,(q(K), M) = M_0 \cdots M_{t-1} \| \mathit{SumBRWpoly}\,(q(K), M_t \cdots M_{2t-2})\,.$$

However when $s \geq t-1$, the problem is a bit complex. Rewrite the terms of $q(K)$ into three parts as following:

$$\begin{aligned} q(K) &= \left(Q_0 K^s + \cdots + Q_{s-t} K^t\right) + Q_{s-t+1} K^{t-1} + \left(Q_{s-t+2} K^{t-2} + \cdots + Q_s\right) \\ &= q_1(K) \cdot K^t + Q_{s-t+1} K^{t-1} + \left(Q_{s-t+2} K^{t-2} + \cdots + Q_s\right) \end{aligned} \tag{6}$$

where $q_1(K) = Q_0 K^{s-t} + Q_1 K^{s-t-1} + \cdots + Q_{s-t}$ when $s \geq t$, or $q_1(K) = \varepsilon$ when $s = t-1$. As $m = 2t-1$ and $s < m$, the degree of $q_1(K)$ is smaller than $(t-1)$.

Algorithm 1: The description of *SumBRWpoly*

```
Input: $q(K) = Q_0K^s + Q_1K^{s-1} + \cdots + Q_s$, $M = M_0 \cdots M_{m-1}$, where
       $m = 2^{v+1} - 1$ and $v \geq \lfloor \log_2(s+1) \rfloor$.
Output: $M' = M'_0 \cdots M'_{m-1}$.
if $s = 0$ then
    $M'_0 \cdots M'_{m-2} = M_0 \cdots M_{m-2}$;
    $M'_{m-1} = M_{m-1} + Q_s$;
else
    $v = \lfloor \log_2 m \rfloor$;
    $t = 2^v$;
    if $v = 1$ then
        if $s = 1$ then
            $M'_0 = M_0$;
            $M'_1 = M_1 + Q_0$;
            $M'_2 = M_2 + Q_1$;
        if $s = 2$ then
            $M'_0 = M_0 + Q_0$;
            $M'_1 = M_1 + Q_0 + Q_1$;
            $M'_2 = M_2 + Q_2$;
    else
        if $s < t - 1$ then
            $M'_0 \cdots M'_{t-1} = M_0 \cdots M_{t-1}$;
            $M'_t \cdots M'_{2t-2} = \mathit{SumBRWpoly}(q(K), M_t \cdots M_{2t-2})$;
        if $s \geq t - 1$ then
            if $s \geq t$ then
                $q_1(K) = \sum_{i=0}^{s-t} Q_{s-t-i}K^i$;
                $M'_0 \cdots M'_{t-2} = \mathit{SumBRWpoly}\,(q_1(K), M_0 \cdots M_{t-2})$;
            else
                $q_1(K) = \varepsilon$;
                $M'_0 \cdots M'_{t-2} = M_0 \cdots M_{t-2}$;
            $M'_{t-1} = M_{t-1} + Q_{s-t+1}$;
            $q_2(K) = \sum_{i=0}^{t-2} Q_{s-i}K^i + Q_{s-t+1} \cdot \left(BRW_K(M_0 \cdots M_{t-2}) + K^{t-1}\right) +$
              $(M_{t-1} + Q_{s-t+1}) \cdot q_1(K)$;
            $M'_t \cdots M'_{2t-2} = \mathit{SumBRWpoly}\,(q_2(K), M_t \cdots M_{2t-2})$;
return $M'$
```

Because only the first $(t-1)$ input blocks affect the terms whose degree is greater than t in BRW-polynomial evaluation, $SumBRWpoly(q(K), M)$ first calls $SumBRWpoly(q_1(K), M_0 \cdots M_{t-2})$ to computes $M'_0 \cdots M'_{t-2}$ if $q_1(K) \neq \varepsilon$, otherwise let $M'_0 \cdots M'_{t-2} = M_0 \cdots M_{t-2}$.

After that $SumBRWpoly$ figures out how $q_1(K)$ affects the remaining lower-degree terms. Moreover let $M'_{t-1} = M_{t-1} + Q_{s-t+1}$, and then

$$\begin{aligned} & BRW_K(M'_0 \cdots M'_{t-2}) \cdot \left(K^t + M'_{t-1}\right) \\ = & \left(BRW_K(M_0 \cdots M^{t-2}) + q_1(K)\right) \cdot \left(K^t + M_{t-1} + Q_{s-t+1}\right) \\ = & BRW_K(M_0 \cdots M^{t-2}) \cdot \left(K^t + M_{t-1}\right) + q_1(K) \cdot K^t \\ & \quad + (M_{t-1} + Q_{s-t+1}) \cdot q_1(K) + Q_{s-t+1} \cdot BRW_K(M_0 \cdots M_{t-2}) \\ = & BRW_K(M_0 \cdots M^{t-2}) \cdot \left(K^t + M_{t-1}\right) + q_1(K) \cdot K^t \boldsymbol{+ Q_{s-t+1} K^{t-1}} \\ & \quad + (M_{t-1} + Q_{s-t+1}) \cdot q_1(K) + Q_{s-t+1} \cdot \left(BRW_K(M_0 \cdots M_{t-2}) \boldsymbol{+ K^{t-1}}\right). \end{aligned} \tag{7}$$

To deal with the lower-degree terms, by (6) (7), let

$$\begin{aligned} q_2(K) = {} & Q_{s-t+2}K^{t-2} + \cdots + Q_s \\ & + (M_{t-1} + Q_{s-t+1}) \cdot q_1(K) + Q_{s-t+1} \cdot \left(BRW_K(M_0 \cdots M_{t-2}) + K^{t-1}\right), \end{aligned}$$

and the degree of $q_2(K)$ is smaller than $(t-1)$, since the degree of both $q_1(K)$ and $\left(BRW_K(M_0 \cdots M_{2^t-2}) + K^{t-1}\right)$ is smaller than $(t-1)$.

Thus the remaining blocks $M'_t \cdots M'_{2t-2}$ can be computed by making another recursive call of $SumBRWpoly(q_2(K), M_t \cdots M_{2t-2})$, and then

$$BRW_K(M'_t \cdots M'_{2t-2}) = BRW_K(M_t \cdots M_{2t-2}) + q_2(K). \tag{8}$$

Therefore when $s \geq t-1$, by (7) (8),

$$\begin{aligned} & BRW_K(M'_0 \cdots M'_{t-2} M'_{t-1} M'_t \cdots M'_{2t-2}) \\ = & BRW_K(M'_0 \cdots M'_{t-2}) \cdot \left(K^t + M'_{t-1}\right) + BRW_K(M'_t \cdots M'_{2t-2}) \\ = & BRW_K(M_0 \cdots M^{t-2}) \cdot \left(K^t + M_{t-1}\right) + BRW_K(M_t \cdots M_{2t-2}) \\ & \quad + q_1(K) \cdot K^t + Q_{s-t+1}K^{t-1} + q_2(K) \\ & \quad\quad + (M_{t-1} + Q_{s-t+1}) \cdot q_1(K) + Q_{s-t+1} \cdot \left(BRW_K(M_0 \cdots M_{t-2}) + K^{t-1}\right) \\ = & BRW_K(M_0 \cdots M_{t-2} M_{t-1} M_t \cdots M_{2t-2}) + q(K). \end{aligned}$$

3.3 Weak keys of BRW-polynomial in MACs

Weak keys in BRW-polynomial function are found ubiquitous, which also threats BRW-based schemes. In this section, we explain how a key subset of BRW-polynomial function turns out to be a weak-key class, and then briefly discuss the weak-key issue of BRW-instantiated MACs.

For any key subset $\mathcal{W} = \{H_0, H_1, \cdots, H_{s-1}\}$ that $s \geq 1$, define

$$q(K) = (K - H_0)(K - H_1) \cdots (K - H_{s-1}) = Q_0 K^s + Q_1 K^{s-1} + \cdots + Q_s$$

where $Q_0 = 1$, similarly. Moreover let $\overline{q}(K) = Q_0 K^{s-1} + Q_1 K^{s-2} + \cdots + Q_{s-1}$ and then $q(K) = K \cdot \overline{q}(K) + Q_s$.

Choose arbitrary m-block message M where $m = 2^{v+1} - 1$ and $v = \lfloor \log_2(s+1) \rfloor$, i.e. $s < m < 2s$. Compute M' and M'' by calling *SumBRWpoly*, that is $M' = \mathit{SumBRWpoly}\,(q(K), M)$ and $M'' = \mathit{SumBRWpoly}\,(\overline{q}(K), M)$. By

$$\begin{aligned} BRW_K(M') &= BRW_K(M) + q(K), \\ K \cdot BRW_K(M'') &= K \cdot BRW_K(M) + K \cdot \overline{q}(K), \end{aligned}$$

it is obvious that

$$K \in \mathcal{W} \iff BRW_K(M') = BRW_K(M), \tag{9}$$

$$K \in \mathcal{W} \bigcup \{0\} \iff K \cdot BRW_K(M') = K \cdot BRW_K(M) + Q_s. \tag{10}$$

Thus the AU property of $BRW_K(\cdot)$, as well as the AXU property of $K \cdot BRW_K(\cdot)$, totally disappears in $\mathcal{W}$, as $\Pr\left[K \xleftarrow{\$} \mathcal{W} : BRW_K(M) = BRW_K(M')\right] = 1$ and $\Pr\left[K \xleftarrow{\$} \mathcal{W} : BRW_K(M) = BRW_K(M'') + Q_s\right] = 1$.

Besides, once the key of BRW falls into $\mathcal{W}$, the security of the BRW-based scheme also collapses, and it is easy to detect whether the unknown key of BRW belongs to $\mathcal{W}$. So $\mathcal{W}$ is a weak-key class of BRW in the BRW-based schemes.

Take two BRW-instantiated MACs, i.e. UTP and WC, as examples, any $\mathcal{W}$ is a weak-key class, as long as $|\mathcal{W}| \geq 2$, because that:

- $\mathrm{UTP}_{K,K'}(M) = E_{K'}(BRW_K(M))$
  1) Forgery attack. Make a single tag-generation query of M and get its tag T. Once $K \in \mathcal{W}$, (M', T) is a successful forgery, since $E_{K'}$ is a PRP and then $T = E_{K'}\left(BRW_K(M)\right) = E_{K'}\left(BRW_K(M')\right)$ according to (9).
  2) Detection. Simply make a tag-generation query of M to get its tag T, and one more verification query of (M', T). If 1 is returned, $BRW_K(M) = BRW_K(M')$ since $E_{K'}$ is a PRP, and thus $K \in \mathcal{W}$ according to (9), otherwise $K \notin \mathcal{W}$.
- $\mathrm{WC}_{K,K'}(N, M) = E_{K'}(N) + K \cdot BRW_K(M)$
  1) Forgery attack. Make a single tag-generation query of (N, M) and get its tag T. Once $K \in \mathcal{W}$, $(N, M'', T + Q_s)$ is a successful forgery, since $T + Q_s = E_{K'}(N) + K \cdot BRW_K(M) + Q_s = E_{K'}(N) + K \cdot BRW_K(M'')$ according to (10).
  2) Detection. Make a single tag-generation query of (N, M) to get its T, and one more verification query of $(N, M'', T + Q_s)$. If 1 is returned, $K \in \mathcal{W} \bigcup \{0\}$ according to (10), otherwise $K \notin \mathcal{W} \bigcup \{0\}$. Besides it is trivial to check if $K = 0$ when $0 \notin \mathcal{W}$.

In real applications, inputs are often padded firstly to deal with variable-length inputs. However even when padding rules are taken into consideration, *SumBRWpoly* still works by some tricks, such as the one used in the weak-key discussion of DCT (Section 4.2), and more refer to [27,25,39,1].

DCT.$enc_{K_1,K_2,K_3}(A,P)$	DCT.$dec_{K_1,K_2,K_3}(A,C)$
$P_L \| P_R = Encode_\tau(P)$	$C_L \| C_R = C$
$X = H_{K_1}(A, P_R)$	$P_R = \mathcal{D}_{K_3}(C_L, C_R)$
$Y = P_L + X$	$X = H_{K_1}(A, P_R)$
$C_L = E_{K_2}(Y)$	$Y = E^{-1}_{K_2}(C_L)$
$C_R = \mathcal{E}_{K_3}(C_L, P_R)$	$P_L = Y - X$
return $C_L \| C_R$	**return** $Decode_\tau(P_L, P_R)$

Table 1. The encryption and decryption of DCT.

4 Weak keys of BRW-polynomial in DCT

DCT [12], short for Deterministic Counter in Tweak, is a Beyond-Birthday-Bound-secure AE scheme, which is constructed from an efficient UHF, a CCA-secure PRP and a Beyond-Birthday-Bound-secure encryption scheme. Forler et al., the designers of DCT, suggest instantiating the underlying UHF with BRW-polynomial function, rather than polynomial function, to avoid the weak-key issue. Pitifully, BRW-polynomial function suffers the same weak-key problem, which can be extended to DCT when instantiating with BRW-polynomial function.

4.1 A brief introduction to DCT.

The encryption of DCT takes the input (A, P), where A is the associated data and P is the plaintext, and outputs the ciphertext C. The decryption of DCT takes the input (A, C), and outputs the plaintext P if the verification is passed.

The encryption and decryption of DCT are illustrated in Table 1. The block length is n-bit. $Encode_\tau(P)$ puts 0^τ on the left of P and then partitions the data into two part $P_L \| P_R$ where $|P_L| = n$. E is a block cipher. $\mathcal{E}$ is an encryption scheme and $\mathcal{D}$ is its inverse. If the left τ bits of P_L are zeroes, $Decode_\tau(P_L, P_R)$ deletes the zeroes and returns the rest bits of $P_L \| P_R$, otherwise $Decode_\tau$ returns $\perp$ indicating the verification is failed.

In DCT, $\mathcal{E}$ is instantiated by the stream-cipher mode CTRT [24]. For simplicity, let $CTRT.Gen_{K_3}(C_L, l)$ be the function that outputs l-bit keystream in the key K_3, and once C_L is new, the l-bit keystream looks random at all. And then

$$\begin{cases} \mathcal{E}_{K_3}(C_L, P_R) = P_R + CTRT.Gen_{K_3}(C_L, |P_R|) \\ \mathcal{D}_{K_3}(C_L, C_R) = C_R + CTRT.Gen_{K_3}(C_L, |C_R|) \end{cases}.$$

The underlying UHF is defined as

$$H_{K_1}(A, P_R) = K_1 \cdot BRW_{K_1}(pad(A) \| pad(P_R) \| L)$$

where the function $pad(X)$ pads X with the minimal number of trailing zeroes such that its length after padding are multiples of n, $L = len(A) \| len(P_R)$ that

$\mathit{len}(X)$ is an $(n/2)$-bit variable representing the bit length of X. Note that the UHF description here is a bit different from the original design in [12], but it doesn't affect the weak-key discussion in the following.

4.2 Weak keys of BRW-polynomial function in DCT.

When instantiating with BRW-polynomial function, which is suggested by its designers, DCT suffers the unavoidable weak-key problem, owing to ubiquitous weak keys of its BRW-polynomial UHF component, and the details are given out in the following.

AE schemes are designed to provide both the confidentiality of plaintexts and the integrity of plaintexts and associated data. However when weak keys are used, at least one of the security goal is broken. For example, GCM, one of the standardized AE schemes, fails to provide the integrity when using weak keys of its polynomial-function UHF, which is proved by the forgery attacks given in [14,27,25,39,1]. Another example is the robust AE scheme AEZ [16], which, when using weak keys given in [22], fails to offer the confidentiality, as its ciphertexts can be distinguished from random bits efficiently. As for BRW-instantiated DCT, both its confidentiality and integrity collapse, when using weak keys of BRW-polynomial. Besides it is easy to detect if the unknown key of BRW-polynomial belongs to some weak-key class.

Inherited from BRW-polynomial function, any subset $\mathcal{W} = \{H_0,\cdots,H_{s-1}\}$ is a weak-key class of DCT, as long as $s \geq 2$. That is, once $K_1 \in \mathcal{W}$, the confidentiality, as well as the integrity, of DCT collapses totally, and it is easy to detect whether $K_1 \in \mathcal{W}$.

Construct message pairs. The crux is how to construct distinct message pairs, say $(A, P), (A', P')$, for any $\mathcal{W}$, satisfying that

$$K_1 \in \mathcal{W} \iff H_{K_1}(A', P'_R) = H_{K_1}(A, P_R).$$

In the following, we explain how to find such pairs with *SumBRWpoly* and a little trick to deal with the padding rule.

For any s-key subset $\mathcal{W}$, let $m = 2^{v+1} - 1$ and $v = \lfloor \log_2(s+1) \rfloor$, i.e. $s < m < 2s$. Let A be arbitrary m-block message, i.e. $A = M_0\cdots M_{m-1}$ where $M_i \in \{0,1\}^n$ for $i = 0, \cdots, m-1$. Let $A' = M'_0\cdots M'_{m-1} = \mathit{SumBRWpoly}(q(K_1), M_0\cdots M_{m-1})$ where $q(K_1) = (K_1 - H_0)\cdots(K_1 - H_{s-1})$, and

$$\mathit{BRW}_{K_1}(M'_0\cdots M'_{m-1}) = \mathit{BRW}_{K_1}(M_0\cdots M_{m-1}) + q(K_1). \tag{11}$$

Besides, let $P_R = P'_R = 0^n\|U$ where $U \in \bigcup_{l=0}^{(m-2)n}\{0,1\}^l$, and

$$\begin{cases} \mathit{pad}(A)\|\mathit{pad}(P_R)\|L & = & M_0\cdots M_{m-1} & \| & 0^n\|\mathit{pad}(U)\|L \\ \mathit{pad}(A')\|\mathit{pad}(P'_R)\|L' & = & M'_0\cdots M'_{m-1} & \| & 0^n\|\mathit{pad}(U)\|L' \end{cases} \tag{12}$$

where $L = \mathit{len}(A)\|\mathit{len}(P_R)$, $L' = \mathit{len}(A')\|\mathit{len}(P'_R)$ and $L = L'$. Obviously, $m + 2 \leq \mid \mathit{pad}(A)\|\mathit{pad}(P_R)\|L \mid \leq 2m - 1$, i.e. no larger than $4s$ blocks, and $\mid \mathit{pad}(A)\|\mathit{pad}(P_R)\|L \mid = \mid \mathit{pad}(A')\|\mathit{pad}(P'_R)\|L' \mid$.

Therefore, by (11)(12),

$$
\begin{aligned}
&BRW_{K_1}(pad(A')\|pad(P'_R)\|L')\\
&= BRW_{K_1}(M'_0\cdots M'_{m-1})\cdot\left(K_1^{m+1}+0^n\right)+BRW_{K_1}(pad(U)\|L')\\
&= (BRW_{K_1}(M_0\cdots M_{m-1})+q(K_1))\cdot\left(K_1^{m+1}+0^n\right)+BRW_{K_1}(pad(U)\|L')\\
&= BRW_{K_1}(M_0\cdots M_{m-1})\cdot\left(K_1^{m+1}+0^n\right)+BRW_{K_1}(pad(U)\|L)+q(K_1)\cdot K_1^{m+1}\\
&= BRW_{K_1}(pad(A)\|pad(P_R)\|L)+q(K_1)\cdot K_1^{m+1},
\end{aligned}
$$

and thus

$$K_1\in\mathcal{W}\bigcup\{0\}\iff H_{K_1}(A',P'_R)=H_{K_1}(A,P_R). \tag{13}$$

Moreover, with $H_{K_1}(A',P'_R)=H_{K_1}(A,P_R)$, let $P=V\|P_R, P'=V\|P'_R$ where $V\in\{0,1\}^{n-\tau}$, and thus

$$C'_L=C_L, \tag{14}$$

where $C'_L\|C'_R=\text{DCT}.enc_{K_1,K_2,K_3}(A',P')$, $C_L\|C_R=\text{DCT}.enc_{K_1,K_2,K_3}(A,P)$.

Weak-key classes in DCT. For any key subset $\mathcal{W}$ of BRW-polynomial function, with the message pair $(A,P),(A',P')$ that satisfy (13) (14) found, both confidentiality and integrity of DCT collapse when $K_1\in\mathcal{W}$. More specifically, when $K_1\in\mathcal{W}$, the following attacks are successful:

- **Distinguishing attack.** Make two encryption queries of $(A,P),(A',P')$, and denote the ciphertexts as $C_L\|C_R, C'_L\|C'_R$ respectively. According to (14), $C_L=C'_L$ is always true in DCT, while happens with the small probability of 2^{-n} in the random case.
- **Forgery attack.** Make a single encryption query of (A,P) to get its ciphertext $C_L\|C_R$, and forge the ciphertext of (A',P') as $C_L\|\left(P'_R+P_R+C_R\right)$, where P_R+C_R is the keystream which is produced by the CTRT encryption component $\mathcal{E}$, i.e. $P_R+C_R=CTRT.Gen_{K_3}(C_L,|P_R|)$.
  More specifically, denote $C'_L\|C'_R=\text{DCT}.enc_{K_1,K_2,K_3}(A',P')$. By (14), $C'_L=C_L$, and then $CTRT.Gen_{K_3}(C'_L,|P'_R|)=CTRT.Gen_{K_3}(C_L,|P_R|)$ since $|P_R|=|P'_R|$. Thus $C'_R=P'_R+CTRT.Gen_{K_3}(C'_L,|P'_R|)=P'_R+P_R+C_R$.

Moreover it is easy to detect whether $K_1\in\mathcal{W}$. Simply make two encryption queries of $(A,P),(A',P')$ and denote the ciphertexts as $C_L\|C_R, C'_L\|C'_R$ respectively. Once $C'_L=C_L$, $H_{K_1}(A',P'_R)=H_{K_1}(A,P_R)$ as the block-cipher E is a PRP, and by (13), $K_1\in\mathcal{W}\bigcup\{0\}$. Besides, if $K_1=0$, the UHF outputs 0 for arbitrary input, and then it is easy to detect either $K_1=0$ or $K_1\in\mathcal{W}$ when $0\notin\mathcal{W}$.

Thus, any key subset $\mathcal{W}$ of BRW-polynomial function that $|\mathcal{W}|\geq 2$ is a weak-key class in BRW-instantiated DCT.

5 Conclusions

This work studies the weak-key problem of BRW-polynomial function and BRW-instantiated schemes. It is found that weak keys in BRW-polynomial function are ubiquitous, and that any key subset of BRW-polynomial which consists of at least 2 keys is a weak-key class in BRW-based cryptographic schemes like the Wegman-Carter scheme, the UHF-then-PRF scheme, DCT, etc. Similar weak-key classes also exist in more BRW-instantiated schemes [6,30,28,29,9]. Although the weak-key attack seems impossible to break the provable security of these schemes, the ubiquity of weak keys is a potential security risk. Furthermore refer to [1], the discussion of twisted polynomials from Ore rings can be applied to BRW-polynomial function to facilitate weak-key attacks.

References

1. Abdelraheem, M.A., Beelen, P., Bogdanov, A., Tischhauser, E.: Twisted polynomials and forgery attacks on GCM. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 762–786. Springer (2015), http://dx.doi.org/10.1007/978-3-662-46800-5_29 2, 6, 11, 13, 15
2. Abdelraheem, M.A., Bogdanov, A., Tischhauser, E.: Weak-key analysis of poet. Cryptology ePrint Archive, Report 2014/226 (2014), http://eprint.iacr.org/2014/226 2
3. Abed, F., Fluhrer, S., Foley, J., Forler, C., List, E., Lucks, S., McGrew, D., , Wenzel, J.: The POET family of on-line authenticated encryption schemes (2014), http://competitions.cr.yp.to/caesar-submissions.html 1, 2
4. Andreeva, E., Bogdanov, A., Lauridsen, M.M., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: AES-COBRA (2014), http://competitions.cr.yp.to/caesar-submissions.html 2
5. Bernstein, D.J.: The poly1305-AES message-authentication code. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3557, pp. 32–49. Springer (2005), http://dx.doi.org/10.1007/11502760_3 2
6. Bernstein, D.J.: Polynomial evaluation and message authentication (2011), http://cr.yp.to/papers.html#pema 2, 6, 15
7. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: fast and secure message authentication. In: Wiener [38], pp. 216–233, http://dx.doi.org/10.1007/3-540-48405-1_14 1
8. Carter, L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. 18(2), 143–154 (1979) 1
9. Chakraborty, D., Mancillas-López, C.: Double ciphertext mode: a proposal for secure backup. IJACT 2(3), 271–287 (2012), http://dx.doi.org/10.1504/IJACT.2012.045588 2, 15
10. Chakraborty, D., Sarkar, P.: HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In: Barua, R., Lange, T. (eds.) Progress in Cryptology - INDOCRYPT 2006. Lecture Notes in Computer Science, vol. 4329, pp. 287–302. Springer (2006), http://dx.doi.org/10.1007/11941378_21 1, 2
11. Etzel, M., Patel, S., Ramzan, Z.: SQUARE hash: fast message authenication via optimized universal hash functions. In: Wiener [38], pp. 234–251, http://dx.doi.org/10.1007/3-540-48405-1_15 1
12. Forler, C., List, E., Lucks, S., Wenzel, J.: Efficient beyond-birthday-bound-secure deterministic authenticated encryption with minimal stretch. In: Liu, J.K., Steinfeld, R. (eds.) Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part

II. Lecture Notes in Computer Science, vol. 9723, pp. 317–332. Springer (2016), http://dx.doi.org/10.1007/978-3-319-40367-0_20 2, 3, 12, 13

13. Halevi, S., Krawczyk, H.: MMH: software message authentication in the gbit/second rates. In: Biham, E. (ed.) Fast Software Encryption 1997. Lecture Notes in Computer Science, vol. 1267, pp. 172–189. Springer (1997), http://dx.doi.org/10.1007/BFb0052345 1
14. Handschuh, H., Preneel, B.: Key-recovery attacks on universal hash function based MAC algorithms. In: Wagner, D. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 5157, pp. 144–161. Springer (2008) 2, 5, 13
15. Harris, S.: The Enchilada authenticated ciphers (2014), http://competitions.cr.yp.to/caesar-submissions.html 2
16. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 15–44. Springer (2015), https://doi.org/10.1007/978-3-662-46800-5_2 13
17. IEEE Std 1619.2-2010: IEEE standard for wide-block encryption for shared storage media (2011) 2
18. Krawczyk, H.: LFSR-based hashing and authentication. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94. Lecture Notes in Computer Science, vol. 839, pp. 129–139. Springer (1994), http://dx.doi.org/10.1007/3-540-48658-5_15 4, 5
19. McGrew, D.A., Fluhrer, S.R.: The extended codebook (XCB) mode of operation. IACR Cryptology ePrint Archive 2004, 278 (2004), http://eprint.iacr.org/2004/278 1
20. McGrew, D.A., Viega, J.: The Galois/Counter mode of operation (GCM) (2004), http://csrc.nist.gov/groups/ST/toolkit/BCM/ 2
21. McGrew, D.A., Viega, J.: The security and performance of the Galois/Counter mode of operation (full version). IACR Cryptology ePrint Archive 2004, 193 (2004), http://eprint.iacr.org/2004/193 1
22. Mennink, B.: Weak keys for aez, and the external key padding attack. In: Handschuh, H. (ed.) Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10159, pp. 223–237. Springer (2017), https://doi.org/10.1007/978-3-319-52153-4_13 13
23. Morales-Luna, G.: On formal expressions of BRW-polynomials. IACR Cryptology ePrint Archive 2013, 3 (2013), http://eprint.iacr.org/2013/003 6
24. Peyrin, T., Seurin, Y.: Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 33–63. Springer (2016), http://dx.doi.org/10.1007/978-3-662-53018-4_2 12
25. Procter, G., Cid, C.: On weak keys and forgery attacks against polynomial-based MAC schemes. In: Moriai, S. (ed.) Fast Software Encryption - 20th International Workshop, FSE 2013. Lecture Notes in Computer Science, vol. 8424, pp. 287–304. Springer (2013), http://dx.doi.org/10.1007/978-3-662-43933-3_15 2, 5, 6, 11, 13
26. Rabin, M.O., Winograd, S.: Fast evaluation of polynomials by rational preparation. Communications on Pure and Applied Mathematics 25(4), 433–458 (1972) 2

27. Saarinen, M.O.: Cycling attacks on GCM, GHASH and other polynomial MACs and Hashes. In: Canteaut, A. (ed.) Fast Software Encryption - 19th International Workshop, FSE 2012. Lecture Notes in Computer Science, vol. 7549, pp. 216–225. Springer (2012), http://dx.doi.org/10.1007/978-3-642-34047-5_13 2, 6, 11, 13
28. Sarkar, P.: Efficient tweakable enciphering schemes from (block-wise) universal hash functions. IEEE Trans. Information Theory 55(10), 4749–4760 (2009), http://dx.doi.org/10.1109/TIT.2009.2027487 2, 7, 15
29. Sarkar, P.: Tweakable enciphering schemes using only the encryption function of a block cipher. Inf. Process. Lett. 111(19), 945–955 (2011), http://dx.doi.org/10.1016/j.ipl.2011.06.014 2, 15
30. Sarkar, P.: Modes of operations for encryption and authentication using stream ciphers supporting an initialisation vector. Cryptography and Communications 6(3), 189–231 (2014), http://dx.doi.org/10.1007/s12095-013-0097-7 2, 15
31. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. IACR Cryptology ePrint Archive 2004, 332 (2004), http://eprint.iacr.org/2004/332 4, 5
32. Stinson, D.R.: Universal hashing and authentication codes. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91. Lecture Notes in Computer Science, vol. 576, pp. 74–85. Springer (1991), http://dx.doi.org/10.1007/3-540-46766-1_5 4
33. Stinson, D.R.: On the connections between universal hashing, combinatorial designs and error-correcting codes. Electronic Colloquium on Computational Complexity (ECCC) 2(52) (1995), http://eccc.hpi-web.de/eccc-reports/1995/TR95-052/index.html 4
34. Sun, Z., Wang, P., Zhang, L.: Weak-key and related-key analysis of hash-counter-hash tweakable enciphering schemes. In: Foo, E., Stebila, D. (eds.) Information Security and Privacy - 20th Australasian Conference, ACISP 2015. Lecture Notes in Computer Science, vol. 9144, pp. 3–19. Springer (2015), http://dx.doi.org/10.1007/978-3-319-19962-7_1 2, 5
35. Wang, P., Feng, D., Wu, W.: HCTR: A variable-input-length enciphering mode. In: Feng, D., Lin, D., Yung, M. (eds.) Information Security and Cryptology, CISC 2005. Lecture Notes in Computer Science, vol. 3822, pp. 175–188. Springer (2005), http://dx.doi.org/10.1007/11599548_15 1, 2
36. Wang, P., Li, Y., Zhang, L., Zheng, K.: Related-key almost universal hash functions: definitions, constructions and applications. In: Peyrin, T. (ed.) Fast Software Encryption - 23rd International Conference, FSE 2016. Lecture Notes in Computer Science, vol. 9783, pp. 514–532. Springer (2016), http://dx.doi.org/10.1007/978-3-662-52993-5_26 2
37. 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) 1, 4
38. Wiener, M.J. (ed.): Advances in Cryptology - CRYPTO '99, Lecture Notes in Computer Science, vol. 1666. Springer (1999) 15
39. Zhu, B., Tan, Y., Gong, G.: Revisiting MAC forgeries, weak keys and provable security of Galois/Counter mode of operation. In: Abdalla, M., Nita-Rotaru, C., Dahab, R. (eds.) Cryptology and Network Security - 12th International Conference, CANS 2013. Lecture Notes in Computer Science, vol. 8257, pp. 20–38. Springer (2013), http://dx.doi.org/10.1007/978-3-319-02937-5_2 2, 5, 6, 11, 13