A Uniform Class of Weak Keys for Universal Hash Functions

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

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{kyzheng13,wp}@is.ac.cn
[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

Abstract. In this paper we investigate weak keys of universal hash functions (UHFs) from their combinatorial properties. We find that any UHF has a general class of keys, which makes the combinatorial properties totally disappear, and even compromises the security of the UHF-based schemes, such as the Wegman-Carter scheme, the UHF-then-PRF scheme, etc. By this class of keys, we actually get a general method to search weak-key classes of UHFs, which is able to derive all previous weak-key classes of UHFs found by intuition or experience. Moreover we give a weak-key class of the BRW polynomial function which was once believed to have no weak-key issue, and exploit such weak keys to implement a distinguish attack and a forgery attack against DTC - a BRW-based authentication encryption scheme. Furthermore in Grain-128a, with the linear structure revealed by weak-key classes of its UHF, we can recover any first $(32 + b)$ bits of the UHF key, spending no more than 1 encryption and $(2^{32} + b)$ decryption queries.

Keywords. Universal hash function, weak key, Wegman-Carter scheme, authenticated encryption, BRW polynomials, Grain-128a.

1 Introduction

UHFs. Universal hash functions (UHFs) were firstly introduced by Carter and Wegman [7,37], and have become common components in numerous cryptographic constructions, like message authentication codes [14,12,14,6], tweakable enciphering schemes [20,35,10] and authenticated encryption schemes [22,2], etc. A UHF is a keyed function. Compared with other primitives, such as pseudorandom permutations (PRPs), pseudorandom functions (PRFs), UHFs have no strength of pseudorandomness. The only requirement is some simple combinatorial properties, which makes UHFs high-performance but meanwhile brittle and vulnerable to weak-key analyses [15,27,25,40,1] and related-key attacks [34,36].

Weak-key analyses. Handschuh and Preneel [15] initiated the study of weak keys 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 this definition they analyzed the weak-key classes of several UHFs from the security of the upper message authentication code (MAC). More specifically, finding a UHF weak-key class is to show that it is easy to do forgery attacks once the key falls into the class and it is easy to detect whether the unknown key belongs to the class. Furthermore they investigated key recovery attacks based on weak-key classes.

The later studies followed this routine, analyzing weak-key classes of UHFs from the upper schemes, such as the authenticated encryption scheme GCM and the message authenticated code GMAC. All the analyses mainly focused on a special UHF - *polynomial evaluation function*, which evaluates a polynomial in the key with the data blocks as coefficients. Saarrinen [27] found that the keys satisfying $K^t = K$ formed a weak-key class in GCM. Procter and Cid [25] found that any subset $\mathcal{D}$ is a weak-key class in GCM and GMAC, if $|\mathcal{D}| \geq 3$ or $|\mathcal{D}| \geq 2$ and $0 \in \mathcal{D}$, exploiting the so-called forgery polynomial $q(x) = \Sigma_{H \in \mathcal{D}}(x - H)$. Zhu, Tan and Gong [40] pointed out that any subset $\mathcal{D}$ 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 evaluation functions. Abdelraheem, Beelen, Bogdanov and Tischhauser [1] further proposed twisted polynomials from Ore rings to construct sparse forgery polynomials, which greatly facilitate key recovery attacks.

Previous weak-key classes of UHFs were found by intuition or experience. There is no general method to find weak keys for UHFs.

Our contributions. In this paper, we investigate weak keys of UHFs firstly through their own combinatorial properties. In Section 3, we define a general class of keys for the almost-universal (AU) hash function and the almost-Delta-universal (AΔU) hash function, which makes the combinatorial properties totally disappear, and even compromises the security of the UHF-based schemes, such as the Wegman-Carter scheme (Section 3.2) and the UHF-then-PRF scheme (Section 3.3). With such general class of keys, we have a general way to find weak-key classes for any AU or AΔU hash function. To facilitate understanding, we give out a few general weak-key classes for some specific AΔU functions including polynomial evaluation function, dot product function, Square hash function, Pseudo-Dot-Product hash function, which shows that previous weak-key classes found by intuition or experience can be derived by this general method (Section 3.4).

In Section 4, we give a weak-key class of the BRW polynomial function which was once believed to have no weak-key issue. With such weak keys, more attacks can be played against the BRW-based schemes. For example, we display a weak-key distinguish attack and a weak-key forgery attack against DCT, a recent authenticated encryption scheme which suggests instantiating its UHF as BRW.

Moreover in Section 5, we develop a weak key attack against Grain-128a to recover its UHF key by exploiting some general weak-key classes. With no more than 1 encryption query and $(2^{32}+b)$ decryption queries, we can recover any first $(32+b)$ bits of the UHF key, and even recover the whole keystream produced by the underlying stream cipher. With the known keystream, we can forge any ciphertext that is not longer than the recovered one in the same IV. This is a great example illustrating that the weak integrity in authenticated encryption may damage the confidentiality.

2 Preliminaries

2.1 Notations

For a finite set $\mathcal{S}$, let $x \xleftarrow{\$} \mathcal{S}$ denote selecting an element x uniformly at random from the set 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 $H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ where $\mathcal{K}$ is a key space, and we often write $H(K, M)$ as $H_K(M)$, where $(K, M) \in \mathcal{K} \times \mathcal{D}$.

2.2 Universal hash functions

Two commonly-used UHFs are almost-universal (AU) hash function and almost-Delta-universal (AΔU) 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 $M, M' \in \mathcal{D}$, $M \neq M'$,*

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

When ϵ is negligible we say that H is AU. We can take $\epsilon = \max_{M \neq M'} \Pr[K \xleftarrow{\$} \mathcal{K} : H_K(M) = H_K(M')]$.

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

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

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

When ϵ is negligible we say that H is AΔU. We can take $\epsilon = \max_{M \neq M', C} \Pr[K \xleftarrow{\$} \mathcal{K} : H_K(M) - H_K(M') = C]$.

For a special abelian group $(\mathcal{R}, \oplus)$ where the addition is exclusive-OR (XOR), we also say H is an almost XOR universal (AXU) hash function [19]. Clearly, if H is ϵ-AΔU, it is also ϵ-AU, for ϵ-AU is a special case of ϵ-AΔU when $C = 0$.

3 A uniform method to find weak-key classes of UHFs

3.1 A general class of keys for any UHF

We investigate weak keys of UHFs firstly through their combinatorial properties, rather than the security of the upper schemes based on them. The combinatorial properties are characterized by $\max_{M \neq M'} \Pr[K \xleftarrow{\$} \mathcal{K} : H_K(M) = H_K(M')]$ or $\max_{M \neq M', C} \Pr[K \xleftarrow{\$} \mathcal{K} : H_K(M) - H_K(M') = C]$ in the definition of UHFs, which is negligible in AU/AΔU hash function. We find a general class of keys that makes the above probability equal the maximal value 1.

Observation 1 *For any universal hash function $H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$, any $M, M' \in \mathcal{D}$, $M \neq M'$, we define*

$$\mathcal{C}_{M,M'} = \{K \in \mathcal{K} : H_K(M) = H_K(M')\}.$$

If $\mathcal{C}_{M,M'} \neq \emptyset$ then

$$\Pr[K \xleftarrow{\$} \mathcal{C}_{M,M'} : H_K(M) = H_K(M')] = 1.$$

Observation 2 *For any universal hash function $H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$, any $M, M' \in \mathcal{D}$, $M \neq M'$, and any $C \in \mathcal{R}$, we define*

$$\mathcal{D}_{M,M',C} = \{K \in \mathcal{K} : H_K(M) - H_K(M') = C\}.$$

If $\mathcal{D}_{M,M',C} \neq \emptyset$ then

$$\Pr[K \xleftarrow{\$} \mathcal{D}_{M,M',C} : H_K(M) - H_K(M') = C] = 1.$$

The proof is straightforward, take $\mathcal{C}_{M,M'}$ as an example,

$$\begin{aligned}\Pr[K \xleftarrow{\$} \mathcal{C}_{M,M'} : H_K(M) = H_K(M')] &= \frac{\#\{K \in \mathcal{C}_{M,M'} : H_K(M) = H_K(M')\}}{\#\mathcal{C}_{M,M'}} \\ &= \frac{\#\mathcal{C}_{\mathcal{M},\mathcal{M}'}}{\#\mathcal{C}_{M,M'}} = 1.\end{aligned}$$

$\mathcal{C}_{M,M'}$. Although the output-collision probability of any two different inputs is negligible, it is *average* over all keys. For any key K in $\mathcal{C}_{M,M'}$, the collision always happens, i.e. $H_K(M) = H_K(M')$. Therefore $\mathcal{C}_{M,M'}$ is a class of keys which makes the combinatorial properties totally disappear for any AU hash function, and so is any nonempty subset of $\mathcal{C}_{M,M'}$.

$\mathcal{D}_{M,M',C}$. Although the output-differential distribution of any two different inputs is negligible, it is also *average* over all keys. For any key K in $\mathcal{D}_{M,M',C}$, it always happens that $H_K(M) - H_K(M') = C$ which makes the distribution totally biased. Therefore $\mathcal{D}_{M,M',C}$ is a class of keys which makes the combinatorial

properties totally disappear for any AΔU hash function, and so is any nonempty subset of $\mathcal{D}_{M,M',C}$.

Relationship with the upper UHF-based schemes. In section 3.2 and 3.3, we will show that generally $\mathcal{C}_{M,M'}$ or $\mathcal{D}_{M,M',C}$ is also a weak-key class in the Wegman-Carter scheme (Section 3.2) or UHF-then-PRF scheme (Section 3.3), as long as the size of the class is no less than 2. In section 3.4 we show that this general class of weak keys includes nearly all weak-key classes of UHFs ever found, and can direct us to search more weak-key classes rather than by intuition or experience as before.

Size of the weak-key class. For any ϵ-AU (ϵ-AΔU) hash function, $\#\mathcal{C}_{M,M'} \leq \epsilon \cdot \#\mathcal{K}$ ($\#\mathcal{D}_{M,M',C} \leq \epsilon \cdot \#\mathcal{K}$). Therefore $\mathcal{C}_{M,M'}$ ($\mathcal{D}_{M,M',C}$) constitutes only a negligible portion of the key space for any AU (AΔU) hash function.

Enumerability of the weak-key class. The above results show that weak-key classes are ubiquitous. In order to make use of weak keys in the attacks against the UHF-based schemes, such as key recovery attack, we have to know what exactly the members are in the classes. Fortunately, most of UHFs are very simple and it is easy to enumerate the members in $\mathcal{C}_{M,M'}$ or $\mathcal{D}_{M,M',C}$.

3.2 $\mathcal{D}_{M,M',C}$ is a weak-key class in the Wegman-Carter scheme

In the following, we discuss $\mathcal{D}_{M,M',C}$ in the Wegman-Carter scheme in which the message is first compressed into a fixed-length string and then encrypted into a tag by one-time-pad encryption. In reality the random string in one-time-pad encryption is often generated by pseudorandom functions. So we can write the Wegman-Carter scheme as a nonce-based MAC:

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

where $H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is the UHF, $F : \mathcal{K}' \times \mathcal{N} \to \mathcal{R}$ is the PRF, M is the message and N is a nonce that never repeats.

More specifically, the Wegman-Carter scheme consists of two algorithms:

- Tag-generation $\mathcal{T}(N,M)$: Calculate $T = H_K(M) + F_{K'}(N)$ and return T.
- Verification $\mathcal{V}(N,M,T)$: If $T = H_K(M) + F_{K'}(N)$, return 1; else return 0.

Two parties share the secret key (K,K') before communicating. For a message M, the sender generates a tag $T = \mathcal{T}(N,M)$ where N is the nonce that never repeats, and sends the triple (N,M,T) to the receiver. After receiving (N',M',T'), the receiver uses the verification algorithm $\mathcal{V}(N',M',T')$ to detect whether the triple is modified by the adversary. If the output is 1, the message is valid; else it is not.

Security of the Wegman-Carter scheme. The adversary **A** can query both the tag-generation oracle $\mathcal{T}(\cdot,\cdot)$ and the verification oracle $\mathcal{V}(\cdot,\cdot,\cdot)$. We assume that **A** never repeats the nonce in querying $\mathcal{T}$, and never queries $\mathcal{V}(N,M,T)$ after querying $\mathcal{T}(N,M)$ to get T. Once the oracle $\mathcal{V}$ outputs 1 during the queries, we say that **A** makes a successful forgery. It has been proved that the Wegman-Carter scheme is secure if H is an AΔU and F is a PRF [19].

$\mathcal{D}_{M,M',C}(\#\mathcal{D}_{M,M',C} \geq 2)$ is a weak-key class in the Wegman-Carter scheme. Following the definition given by Handschuh and Preneel [15] for MACs, a weak-key class is a subset of the key space satisfying the following two conditions.

1) Detectability. If the weak-key class is of size s, and identifying that a key belongs to this class requires testing fewer than s keys by exhaustive search and fewer than s verification queries.
2) Forgeability. The forgery probability for this class of keys is substantially larger than the average.

Theorem 1. *If $\#\mathcal{D}_{M,M',C} \geq 2$, $\mathcal{D}_{M,M',C} = \{K : H_K(M) - H_K(M') = C\}$ is a weak-key class in any Wegman-Carter scheme using H as the underlying universal hash function.*

Proof. 1) Detectability. First query $\mathcal{T}(N, M)$ to get T, so $T = H_K(M)+F_{K'}(N)$. Then query $\mathcal{V}(N, M', T-C)$. The answer is 1, if and only if $T-C = H_K(M')+F_{K'}(N)$, or $H_K(M) - H_K(M') = C$. Therefore the answer is 1, if and only if $K \in \mathcal{D}_{M,M',C}$. We only use 1 $\mathcal{T}$-query and 1 $\mathcal{V}$-query to identify if K belongs to $\mathcal{D}_{M,M',C}$.

2) Forgeability. First query $\mathcal{T}(N, M)$ to get T, that is $T = H_K(M)+F_{K'}(N)$. Then query $\mathcal{V}(N, M', T - C)$. From the above analysis, if $K \in \mathcal{D}_{M,M',C}$, the answer is 1. Therefore we only use 1 $\mathcal{T}$-query and 1 $\mathcal{V}$-query to make a successful forgery.

So if $\#\mathcal{D}_{M,M',C} \geq 2$, $\mathcal{D}_{M,M',C}$ is a weak-key class in any Wegman-Carter Scheme.

3.3 $\mathcal{C}_{M,M'}$ is a weak-key class in the UHF-then-PRF scheme

The Wegman-Carter scheme is the most popular way to construct MACs based on UHFs. The other way is to use a pseudorandom function (PRF) to process the outputs of the UHF:

$$\mathrm{UTP}_{K,K'}(M) = F_{K'}(H_K(M)).$$

It has been proved [31] that if H is an AU function and F is a PRF, then UTP is also a PRF, which can be used as a secure MAC.

Similar to the Wegman-Carter scheme, we also have the following result:

Theorem 2. *If $\#\mathcal{C}_{M,M'} \geq 2$, $\mathcal{C}_{M,M'} = \{K : H_K(M) = H_K(M')\}$ is a weak-key class in any UHF-then-PRF scheme using H as the underlying universal hash function.*

We omit the detailed proof of Theorem 2 since it is almost the same as that of Theorem 1.

3.4 Weak-key classes of other specific UHFs

In this section, we take some commonly-used AΔU hash functions as examples, to give out a few weak-key classes in the type of $\mathcal{D}_{M,M',C}$, which take various M, M' and C values. Actually most of previous weak-key classes, found by intuition or experience, are just specific examples of the general weak-key class. With $\mathcal{C}_{M,M'}$ and $\mathcal{D}_{M,M',C}$, we can find weak-key classes in a general way, some of which even break the whole key space into some clear structures like the linear partition.

Polynomial evaluation function. Polynomial evaluation function is the most explored UHF, appearing in GCM [21], XCB [17], HCTR [35], HCH [10], COBRA [4], Enchilada [16], POET [2], etc. Polynomial evaluation function is defined as

$$Poly_K(M) = \sum_{i=0}^{m-1} M_i K^{m-i}$$

where $K \in GF(2^n), M_i \in GF(2^n)$ for $i = 0, \cdots, m-1$.

Two classes of weak keys for polynomial evaluation function are as following: ($C \in GF(2^n)$, $A_i \in GF(2^n)$ for $i = 0, \cdots, m$, and $H_i \in GF(2^n)$ for $i = 0, \cdots, m-1$)

- $\mathcal{D}_{M,M',0} = \{K : K^m = K\}$, where $M_0 = 1, M_j = 0$ for $j = 1, \cdots, m-1, M'_l = 0$ for $l = 0, \cdots, m-2$,$M'_{m-1} = 1$. [27].
- $\mathcal{D}_{M,M-A,C} = \{H_0, H_1, \cdots, H_{m-1}\}$, where M is any arbitrary input, $A = A_0 \cdots A_{m-1}$ and $C = -A_m$ are the coefficients of the so-called forgery polynomial $q(x) = \Sigma_{i=0}^{m-1}(x - H_i) = A_0 x^m + A_1 x^{m-1} + \cdots + A_{m-1}x + A_m$ [25]. (Note that $A_0 = 1$.)

Dot product function. Replace K^i in polynomial evaluation function by independent parts of the key, we get dot product function as

$$Dotp_K(M) = \sum_{i=0}^{m-1} M_i K_i$$

where $M_i, K_i \in GF(2^n)$ for $i = 0, \cdots, m-1$. Dot product function can be seen as a simple version of MMH [14] which uses modular arithmetic for efficiency.

Here we give a general weak-key class for any nonzero input-differential value $A = A_0 \cdots A_{m-1} \neq 0$ and any output-differential value C, regardless of the inputs: ($A_i \in GF(2^n)$ for $i = 0, \cdots, m-1$, $C \in GF(2^n)$)

- $\mathcal{D}_{M,M-A,C} = \{K : \sum_{i=0}^{m-1} A_i K_i = C\}$, where $A_i \in GF(2^n)$ for $i = 0, 1, \cdots, m-1$, and $C \in GF(2^n)$. With suitable values of A, C, $\mathcal{D}_{M,M-A,C}$ can form a linear partition of the whole key space $GF(2^{nm})$.

Square hash function. Square hash function [12] is defined as

$$SQH_K(M) = \sum_{i=0}^{m-1} (M_i + K_i)^2$$

where $M_i, K_i \in GF(p)$ for $i = 0, 1, \cdots, m-1$.

In the following we give two classes of weak keys for Square hash function: ($A_i \in GF(p)$ for $i = 0, 1, \cdots, m-1$, $C, \Delta \in GF(p)$)

- $\mathcal{D}_{M,M-A,\Delta} = \{K : \sum_{i=0}^{m-1} A_i K_i = C\}$, where M is any arbitrary input, $A = A_0 \cdots A_{m-1} \neq 0$ is any nonzero input-differential that $A_i \in GF(p)(i = 0, 1, \cdots, m-1)$, $C \in GF(p)$, and $\Delta = 2C + \sum_{i=0}^{m-1} A_i(2M_i - A_i)$ is the output-differential value. Similarly when taking suitable values of A, C, $\mathcal{D}_{M,M-A,\Delta}$ can break the key space of Square hash function into a linear partition.
- $\mathcal{D}_{M,M-A,0} = \{K : K_i = K_j\}$ for fixed distinct i, j $(i, j = 0, \cdots, m-1)$ is an example of the former general class by taking specific A, C, M, i.e. $A_i = -A_j \neq 0$, $A_l = 0$ for $l \in \{0, \cdots, m-1\} \setminus \{i, j\}$, $C = 0$, $M \in \{M : M_i - M_j = A_i\}$. Moreover $\bigcup_{i \neq j, i, j = 0, \cdots, m-1} (\mathcal{D}_{M,M-A,0}) = \{K : K_i = K_j, i \neq j, i, j = 0, 1, \cdots, m-1\}$ is exactly the *type II* weak-key class given in [15].

Pseudo-Dot-Product hash function. Pseudo-Dot-Product hash function is defined as

$$PDP_K(M) = \sum_{i=0}^{m/2-1} (M_{2i} + K_{2i})(M_{2i+1} + K_{2i+1})$$

where m is even and $M_j, K_j \in GF(2^n)$ for $j = 0, \cdots, m-1$.

Pseudo-Dot-Product hash function can be regarded as a simple version of NMH [14,37], NH [6], WH [18], etc. Similarly we give some linear weak-key classes for Pseudo-Dot-Product hash functions in the following:
($A_j \in GF(2^n)$ for $j = 0, \cdots, m-1$, $C, \Delta \in GF(2^n)$)

- $\mathcal{D}_{M,M',\Delta} = \{K : \sum_{j=0}^{m-1} A_j K_j = C\}$, where $A = A_0 \cdots A_{m-1} \neq 0$ is any nonzero input-differential, M is any arbitrary input while M' is defined as $M'_{2i} = M_{2i} - A_{2i+1}$, $M'_{2i+1} = M_{2i+1} - A_{2i}$ for $i = 0, \cdots, m/2-1$, and $\Delta = C + \sum_{i=0}^{m/2-1} (M_{2i}A_{2i} + M_{2i+1}A_{2i+1} - A_{2i}A_{2i+1})$ is the output-differential value. Again, with suitable values of A, C, $\mathcal{D}_{M,M',\Delta}$ can reveal a linear structure of the whole key space.
- $\mathcal{D}_{M,M',0} = \{K : K_{2i} = K_{2i+1}\}$ for fixed i $(i = 0, \cdots, m/2-1)$ is an example of the former general class by taking specific A, C, M, i.e. $A_{2i} = -A_{2i+1} \neq 0$, $A_{2l} = A_{2l+1} = 0$ for $l \in \{0, \cdots, m/2-1\} \setminus \{i\}$, $C = 0$, $M \in \{M : M_{2i} + A_{2i} = M_{2i+1}\}$ and M' is defined the same as the former class. Moreover $\bigcup_{i=0, \cdots, m-1} (\mathcal{D}_{M,M',0}) = \{K : K_{2i} = K_{2i+1}, i = 0, 1, \cdots, m/2-1\}$ is exactly the *type III* weak-key class given in [15].

4 Application to the BRW polynomial function

In [5], Bernstein gave a variant of polynomial evaluation function, by making small changes to the function first published by Rabin and Winograd [26], which is defined as BRW (short for Bernstein-Rabin-Winograd) polynomial function formally in [23]. Its recursive definition is as following:

- $BRW_K(\varepsilon) = 0^n$;
- $BRW_K(M_0) = M_0$;
- $BRW_K(M_0 M_1) = M_0 K + M_1$;
- $BRW_K(M_0 M_1 M_2) = (M_0 + K)(M_1 + K^2) + M_2$;

– $BRW_K(M_0 \cdots M_{m-1}) = BRW_K(M_0 \cdots M_{t-2})(K^t + M_{t-1}) + BRW_K(M_t \cdots M_{m-1})$ if $t \in \{4, 8, 16, 32, \cdots\}$ and $t \leq m < 2t$;

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

No weak-key class for the BRW polynomial function was found before due to its recursive definition which Handschuh and Preneel believed "could alleviate concerns related to weak keys" [15]. The seemingly avoidance of the weak-key issue, as well as the high-performance owing to the decreasing number of multiplications, makes the BRW polynomial function widely-used in lots of cryptographic schemes, such as authentication schemes [5,30], tweakable enciphering schemes [28,29,8,9], authenticated encryption schemes [13]. Unfortunately, there exists weak keys for BRW polynomial function.

Weak-key class of BRW. It is obvious that in $BRW_K(M_0 \cdots M_{t-2})$ where $t = 2^s (s \geq 2)$, M_{t-2} only appears in the constant term. For any message with the specific length $(t+2)$-block, say $M = M_0 \cdots M_{t-3} M_{t-2} M_{t-1} M_t M_{t+1}$,

$$\begin{aligned} BRW_K(M) &= BRW_K(M_0 \cdots M_{t-2})(K^t + M_{t-1}) + M_t K + M_{t+1} \\ &= (f_K(M_0 \cdots M_{t-3}) + M_{t-2})(K^t + M_{t-1}) + M_t K + M_{t+1}, \end{aligned}$$

where $f_K(M_0 \cdots M_{t-3})$ is a polynomial in K determined by $M_0, \cdots, M_{t-3}$. Define $M' = M'_0 \cdots M'_{t+1}$ as

$$\begin{cases} M'_i & = M_i, i = 0, 1, \cdots, t-3, t-1 \\ M'_{t-2} & = M_{t-2} - 1 \\ M'_t & = M_t + 1 \\ M'_{t+1} & = M_{t-1} + M_{t+1} \end{cases} \tag{1}$$

and $\mathcal{C}_{M,M'} = \{K : BRW_K(M) = BRW_K(M')\} = \{K : K^t = K\}$. If s is a factor of n, $\{K : K^t = K\}$ is exactly the subfield of $GF(2^n)$ which contains 2^s elements. This weak-key class is similar to the one found in [27].

When the BRW polynomial function is used in the Wegman-Carter scheme or the UHF-then-PRF scheme, just as shown in Section 3.2 and 3.3, the adversary can firstly make a query of M to the tag-generation oracle and get the tag T, and secondly make a query of the forgery pair (M', T) to the verification oracle. The forgery is verified successfully, if and only if $K \in \mathcal{C}_{M,M'}$. Therefore $\mathcal{C}_{M,M'} = \{K : K^t = K\}$ is a class of weak keys in the upper authentication schemes based on BRW polynomial function.

Besides, $\mathcal{C}_{M,M'}$ is also a weak-key class in other BRW-based schemes. Here we take DCT, a recent authenticated encryption scheme designed by Forler et. al [13], as an example. The reason why they chose the BRW polynomial function to instantiate the UHF component in DTC is that there seems no weak-key issue in BRW. Unfortunately, it is not the case.

Description of 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.

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

Table 1. Encryption and decryption of DCT.

The decryption of DCT takes the input of (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. $\mathcal{E}$ is instantiated by the stream cipher mode CTRT [24]. If the left τ bits of P_L are zeroes, $Decode_\tau(P_L,P_R)$ deletes the zeroes and returns the rest bits, otherwise $Decode_\tau$ returns $\perp$ indicating the verification is failed.

In DCT, the UHF is

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

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 $len(X)$ is an $n/2$-bit variable representing the bit length of X.

Weak-key class of DCT. We show that $\{K : K^t = K\}(t = 2^s)$ for $s \geq 2)$ is a class of weak keys for DCT. The crux is to construct two distinct inputs $(A,P),(A',P')$ that for any $K_1 \in \{K : K^t = K\}$, $H_{K_1}(A,P_R) = H_{K_1}(A',P'_R)$ or

$$BRW_{K_1}(pad(A)\|pad(P_R)\|L) = BRW_{K_1}(pad(A')\|pad(P'_R)\|L').$$

Let M, M' denote $pad(A)\|pad(P_R)\|L, pad(A')\|pad(P'_R)\|L'$ respectively. According to $\mathcal{C}_{M,M'}$ discussed before, what we have to find are two distinct inputs $(A,P),(A',P')$ such that M, M' satisfies (1). Moreover $M_{t+1} = L, M'_{t+1} = L'$ are defined by the former blocks, which has to be dealt carefully. The process is as following:

- Choose some (A,P) that makes $M = pad(A)\|pad(P_R)\|L$ be exactly $(t+2)$ blocks. Here $M_{t+1} = L$ is decided by $M_0 \cdots M_t$.
- Find a meaningful M' according to (1) where "meaningful" refers to there exists (A',P') satisfies $M' = pad(A')\|pad(P'_R)\|L'$. More specifically, let $M'_i = M_i$ for $i = 0,1,\cdots,t-3$, $M'_{t-2} = M_{t-2} - 1$, $M'_{t-1} = M_{t-1}$, $M'_t = M_t + 1$, which is great likely to contradict with $M'_{t+1} = M_{t-1} + M_{t+1}$ since both M_{t+1}, M'_{t+1} are defined by their former blocks. In this case, we can modify M_{t-1} carefully to find a special M_{t-1} that satisfies $L' = M_{t-1} + L$. (Note that different M_{t-1} may get different L, L'.)

– Get (A', P'_R) from the meaningful M' inversely, and let $P'_L = P_L$.
– Return $(A, P), (A', P')$.

After getting two inputs, say $(A, P), (A', P')$, that satisfies both $H_{K_1}(A, P_R) = H_{K_1}(A', P'_R)(K_1 \in \{K : K^t = K\})$ and $P_L = P'_L$, we can apply the following attacks:

1) Distinguish attack. Make encryption queries of (A, P) and (A', P') respectively. In DCT, the leftmost n bits of the two ciphertexts are the same, which distinguish DCT from a random oracle within only two encryption queries.

2) Forgery attack. Make a single encryption query of (A, P) and get $C = (C_L, C_R)$. The forgery ciphertext C' is defined as $C'_L = C_L$, $C'_R = P'_R \oplus P_R \oplus C_R$, where $C_R \oplus P_R$ is the keystream produced by the encryption component $\mathcal{E}$ which is instantiated by CTRT. Thus (A', C') is a valid forgery, which can be successfully decrypted to P'.

5 Application to Grain-128a

Grain-128a [3] is a stream cipher with optional authentication. Here we only focus on Grain-128a with mandatory authentication. The massage authentication code (MAC) in Grain-128a that adopts the Wegman-Carter scheme is defined as

$$MAC_{K,R}(M) = H_K(M) \oplus R$$

where $H : \mathcal{K} \times \mathcal{D} \to \{0,1\}^{32}$ is a UHF, $K \in \mathcal{K}$ and $R \in \{0,1\}^{32}$ is a random string generated by the underlying stream cipher denoted as G.

Suppose that the message is $M = m_0 m_1 \cdots m_{l-1}$ where $m_i \in \{0,1\}$ for $i = 0, 1, \cdots, l-1$, and the key is $K = k_0 k_1 \cdots k_{l+31}$ where $k_i \in \{0,1\}$ for $i = 0, 1, \cdots, l+31$. The UHF calculates as

$$H_K(M) = \sum_{i=0}^{l-1} m_i(k_i k_{i+1} \cdots k_{i+31}) + k_l k_{l+1} \cdots k_{l+31},$$

where $m_i(k_i k_{i+1} \cdots k_{i+31})$ equals $k_i k_{i+1} \cdots k_{i+31}$ if $m_i = 1$, or 0^{32} otherwise.

Actually the UHF is based on *Toeplitz hashing* and we can write it as $H_K(M) = [M](k_0, k_1, \cdots, k_{l+31})^T$, where $[M]$ is a $32 \times (l+32)$ matrix

$$[M] = \begin{pmatrix} m_0 & m_1 & \cdots & m_{l-1} & 1 & 0 & \cdots & 0 \\ 0 & m_0 & m_1 & \cdots & m_{l-1} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & m_0 & m_1 & \cdots & m_{l-1} & 1 \end{pmatrix}.$$

Before encryption or decryption, the underlying stream cipher G generates random bits. We denote it as

$$G(IV, MK) \to r_0 r_1 r_2 r_3 \cdots$$

where IV is an initial vector and MK is the key of the stream cipher. And R, K and U used in the encryption and decryption algorithms are initialized respectively as following:

- $R = r_0 r_1 \cdots r_{31}$.
- $K = k_0 k_1 \cdots k_{l+31}$, where $k_0 k_1 \cdots k_{31} = r_{32} r_{33} \cdots r_{63}$, $k_{32+i} = r_{64+2i+1}$, $i = 0, 1, \cdots, l-1$.
- $U = u_0 u_1 \cdots u_{l-1}$, where $u_i = r_{64+2i}$, $i = 0, 1, \cdots l-1$.

The encryption and decryption algorithms in Grain-128a are as following:

- Encryption $\mathcal{E}(IV, M)$: Calculate $C = M \oplus U$ and $T = H_K(M) \oplus R$. Return (IV, C, T).
- Decryption $\mathcal{D}(IV, C, T)$: Calculate $M = C \oplus U$ and $T' = H_K(M) \oplus R$. If $T = T'$, return M; else return $\top$.

Weak-key classes in Grain-128a. Using the general method, we get the weak-key class of H as following:

$$\mathcal{D}_{M, M \oplus A, Z} = \{K : \bigoplus_{i=0}^{l-1} a_i (k_i k_{i+1} \cdots k_{i+31}) = Z\}$$

where $A = a_0 a_1 \cdots a_{l-1} \neq 0$ and $a_i \in \{0, 1\}$ for $i = 0, \cdots, l-1$, $Z \in \{0,1\}^{32}$. This class of keys is only determined by A and Z, so we denote it as $W_{A,Z}$ which is an affine space. With suitable values of A, Z, $W_{A,Z}$ can make a linear partition to the key space. For example, for any fixed A, $\mathcal{K} = \bigcup_{Z \in \{0,1\}^{32}} W_{A,Z}$, which is exploited in the following attack.

It is easy to prove that arbitrary $W_{A,Z}$ is weak in Grain-128a. We can detect whether the unknown key belongs to the class and do forgery attacks once the key falls into the class. The proof is almost the same as the one in Theorem 1, so we omit it.

Key recovery attack. When the key of H falls into any weak-key class $W_{A,Z}$, we can detect which one it does and predict the tag value for a new message by doing forgeries, since $W_{A,Z}$ is also weak in Grain-128a. Using such property, we can recover the key of H by two steps: firstly the leftmost 32 bits and then the remaining ones bit-by-bit.

1) Recovery of $k_0 k_1 \cdots k_{31}$, given 1 encryption query and 2^{32} decryption queries.

Let $A = 10^{l-1}$, then $W_{A,Z} = \{K : k_0 k_1 \cdots k_{31} = Z\}$, and we get a complete disjoint coverage for the key space of H, i.e. $\mathcal{K} = \bigcup_{Z \in \{0,1\}^{32}} \{K : k_0 k_1 \cdots k_{31} = Z\}$. We can recover $k_0 k_1 \cdots k_{31}$ according to the property that whether the key of H belongs to some class, say $K \in W_{10^{l-1}, Z}$, can be detected by doing forgeries, and then $k_0 k_1 \cdots k_{31} = Z$. The process is as following:

- Get (IV, C, T) by intercepting or by querying the encryption oracle $\mathcal{E}$.
- Flip the first bit of C to get $C^0 = C \oplus 10^{l-1}$.
- Query the decryption oracle $\mathcal{D}$ with (IV, C^0, T'), where T' goes through all 2^{32} values until the decryption oracle verifies successfully. Denote the verified tag as T^0 and the plaintext returned as M^0.

In the decryption oracle query, the keystream is the same as the one in the encryption since the initial value is the same. Only the first bit of C is flipped, so $M^0 = M \oplus 10^{l-1}$, and M is recovered when (IV, C, T) is intercepted. Since

$T = H_K(M) \oplus R$ and $T^0 = H_K(M^0) \oplus R$, then $H_K(M) \oplus H_K(M^0) = T \oplus T^0 = Z$, that is $k_0 k_1 \cdots k_{31} = T \oplus T^0$.

2) Recovery of $k_{32}, k_{33}, \cdots, k_{l+30}$, given 1 decryption query per bit.

We show that after knowing the value of $k_0 k_1 \cdots k_{31}$, we can recover the remaining part of the key *bit-by-bit* with the decryption oracle. The main idea is that, to recover k_{i+31} for $i = 1, \cdots, l-1$, let $A = 0^i 10^{l-i-1}$, then $W_{A,Z} = \{K : k_i k_{i+1} \cdots k_{i+31} = Z\}$, and Z has only two possible values since $k_i k_{i+1} \cdots k_{i+30}$ is already recovered. Thus we detect either $W_{A,Z}$ the unknown key belongs to by a single forgery.

At first we show how to recover k_{32} by one query to $\mathcal{D}$. Let $A = 010^{l-2}$ and $W_{A,Z}$ turns out to be $\{K : k_1 k_2 \cdots k_{32} = Z\}$ where only k_{32} is unknown. Assume that $k_{32} = 1$, and we can detect whether the unknown key belongs to $W_{010^{l-2},Z}$ where $Z = k_1 \cdots k_{31} 1$ with a single forgery. Flip the second bit of C to get $C^1 = C \oplus 010^{l-2}$ and query the decryption oracle $\mathcal{D}(IV, C^1, T^1)$, where $T^1 = T \oplus k_1 k_2 \cdots k_{31} 1$. If the decryption query is verified successfully, we know that T^1 is the correct tag for $M^1 = M \oplus 010^{l-2}$, and $Z = T \oplus T^1$ since $T = H_K(M) \oplus R$ and $T^1 = H_K(M^1) \oplus R$, that is the unknown key belongs to $W_{010^{l-2},Z}$ where $Z = k_1 \cdots k_{31} 1$. Therefore if the decryption query is verified successfully, $k_{32} = 1$; otherwise $k_{32} = 0$.

Generally for $i = 1, \cdots, l-1$, we flip the i^{th} bit of C to get $C^i = C \oplus 0^i 10^{l-i-1}$ and query the decryption oracle $\mathcal{D}$ with (IV, C^i, T^i), where $T^i = T \oplus k_i k_{i+1} \cdots k_{i+30} 1$. If the decryption query is verified successfully, then $k_{i+31} = 1$; else $k_{i+31} = 0$.

All in all, in order to recover the first $(32 + b)$ bits of the UHF key, we need only no more than 1 encryption query and $(2^{32} + b)$ decryption queries, where $b \leq l - 2$ (k_{l+31} is unknown yet). This key recovery attack based on weak-key classes also applies to lots of schemes that adopt Toeplitz hashing strategy, such as 128-EIA3 [11], LFSR-based Toeplitz hashing MAC [19], Sablier [39], etc.

After recovering the leftmost $(l + 30)$ key bits of H, all random bits generated by $G(IV, MK)$ can be recovered. More specifically, during the recovery of $k_0 k_1 \cdots k_{31}$, M is easy to know and then $U = C \oplus M$. Since k_{l+31} is still unknown, let us guess $k_{l+31} = 1$ and compute $R = T \oplus H_K(M)$ where the first 31 bits of R are recovered while the last bit stays unsure. To be simple, denote the 1-bit message $M'' = m_0$, i.e. the first bit of M, and its corresponding ciphertext $C'' = c_0$, i.e. the first bit of C. Compute $T'' = H_K(M'') \oplus R$, and query the decryption oracle $\mathcal{D}$ with (IV, C'', T''). If the decryption query is verified successfully, the last bit of R is right and $k_{l+31} = 1$, otherwise $R = R \oplus 0^{31}1$ and $k_{l+31} = 0$. By so far, R, K, U are all recovered, and we can forge any ciphertext that is not longer than l in the same IV.

Though the key recovery attack against Grain-128a is a great example that exploits the structure of the key space revealed by weak-key classes, the main problem exists in Grain-128a is that it adopts a *short* tag strategy in its MAC algorithm, and the adversary can traverse all possible tag values, i.e. 2^{32} in Grain-128a, with the decryption oracle to find the correct tag for its forgery, since there is no restriction in how many queries in the same IV can be made to the

decryption oracle. Moreover, for any triple (IV, C, T) intercepted, the adversary can recover the plaintext by simply implementing the first step of the above key recovery attack, which illustrates that the weak integrity in authenticated encryption may damage the confidentiality. To fix this problem, Grain-128a can either adopts a long tag, say 64 or 128 bits, or restricts the number of decryption queries for a single IV value.

6 Conclusions

In this paper we investigate weak keys of UHFs firstly from their simple combinatorial properties. By giving a class of keys which makes the combinatorial properties totally disappear and even compromises the security of the UHF-based schemes, lots of new weak-key classes for various UHFs can be found in a general way easily, indicating that weak keys of UHFs are ubiquitous. We stress that any weak-key class found only accounts for a negligible portion of the UHF key spaces, and the probability that the secret key falls into the class is also negligible. Furthermore all the weak-key analyses to UHFs do not contradict with the provable security results of the upper schemes based on UHFs. However weak-key analyses of UHFs may reveal some clear structure of the key space, which can be exploited to develop various attacks like key recovery attack, once the attack complexity goes beyond the provable security bound.

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 1, 2
2. 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, 7
3. Ågren, M., Hell, M., Johansson, T., Meier, W.: Grain-128a: a new version of Grain-128 with optional authentication. IJWMC 5(1), 48–59 (2011) 11
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 7
5. Bernstein, D.J.: Polynomial evaluation and message authentication (2011), http://cr.yp.to/papers.html#pema 8, 9
6. 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
7. Carter, L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. 18(2), 143–154 (1979) 1
8. 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 9

9. Chakraborty, D., Mancillas-López, C., Rodríguez-Henríquez, F., Sarkar, P.: Efficient hardware implementations of BRW polynomials and tweakable enciphering schemes. IEEE Trans. Computers 62(2), 279–294 (2013), http://dx.doi.org/10.1109/TC.2011.227 9
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, 7
11. ETSI/SAGE: Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. Document 1: 128-EEA3 and 128-eia3 specification. Version 1.7, July 2012 (2012) 13
12. 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, 7
13. 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 9
14. 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, 7, 8
15. 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) 1, 6, 8, 9
16. Harris, S.: The Enchilada authenticated ciphers (2014), http://competitions.cr.yp.to/caesar-submissions.html 7
17. IEEE Std 1619.2-2010: IEEE standard for wide-block encryption for shared storage media (2011) 7
18. Kaps, J.P., Yüksel, K., Sunar, B.: Energy scalable universal hashing. IEEE Trans. Computers 54(12), 1484–1495 (2005) 8
19. 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 3, 5, 13
20. 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
21. McGrew, D.A., Viega, J.: The Galois/Counter mode of operation (GCM) (2004), http://csrc.nist.gov/groups/ST/toolkit/BCM/ 7
22. 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
23. Morales-Luna, G.: On formal expressions of BRW-polynomials. IACR Cryptology ePrint Archive 2013, 3 (2013), http://eprint.iacr.org/2013/003 8
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 10
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 1, 2, 7
26. Rabin, M.O., Winograd, S.: Fast evaluation of polynomials by rational preparation. Communications on Pure and Applied Mathematics 25(4), 433–458 (1972) 8
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 1, 2, 7, 9
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 9
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 9
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 9
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 6
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_53
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 3
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 1, 2
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, 7
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 1
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, 8
38. Wiener, M.J. (ed.): Advances in Cryptology - CRYPTO '99, Lecture Notes in Computer Science, vol. 1666. Springer (1999) 14, 15

39. Zhang, B., Shi, Z., Xu, C., Yao, Y., Li, Z.: Sablier v1 (2014), http://competitions.cr.yp.to/caesar-submissions.html 13
40. 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 1, 2