IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 0, No. 0, pp. 0–0. DOI:10.46586/tosc.v0.i0.0-0

Small Stretch Problem of the DCT Scheme and How to Fix it

Yuchao Chen[1,2], Tingting Guo[3], Lei Hu[4,5], Lina Shang[6], Shuping Mao[4,5] and Peng Wang[7(✉)]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, China
[2] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
chenyuchao@mail.sdu.edu.cn
[3] Research Center for Data Hub and Security, Zhejiang lab
guotingting4633@gmail.com
[4] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, CAS
[5] School of Cyber Security, University of Chinese Academy of Sciences
{hulei,maoshuping}@iie.ac.cn
[6] Space Star Technology Co., Ltd.
sln-8108@163.com
[7] School of Cryptology, University of Chinese Academy of Sciences
w.rocking@gmail.com

Abstract. DCT is a beyond-birthday-bound (BBB) deterministic authenticated encryption (DAE) mode proposed by Forler et al. in ACISP 2016, ensuring integrity by redundancy. The instantiation of DCT employs the BRW polynomial, which is more efficient than the usual polynomial in GCM by reducing half of the multiplication operations. However, we show that DCT suffers from a small stretch problem similar to GCM. When the stretch length τ is small, choosing a special m-block message, we can reduce the number of queries required by a successful forgery to $\mathcal{O}(2^\tau/m)$. We emphasize that this attack efficiently balances space and time complexity but does not contradict the security bounds of DCT. Finally, we propose an improved scheme named Robust DCT (RDCT) with a minor change to DCT, which improves the security when τ is small and makes it resist the above attack.

Keywords: DCT · Deterministic Authenticated Encryption · AEAD · BRW polynomial · Forgery Attack · Stretch.

1 Introduction

Authenticated encryption (AE) schemes [Rog04] provide confidentiality and integrity simultaneously. AE achieves integrity by generating a tag or encoding some redundancy into the message, leading to ciphertext expansion, a.k.a. tag length or stretch. In the real world of cryptographic systems, such as RFID cards, sensor networks, or embedded devices, 128-bit tags may not be supported; instead, these embedded devices usually support tag sizes such as 32-bit or 64-bit. Generally, GCM [MV04a] has a variety of tag lengths for choice by truncation, such as $64, 96, 104, 112, 120$, and 128-bit, which fits all kinds of requirements. GCM is widely used in many applications, such as IPsec and TLS. NIST already standardizes GCM as SP 800-38D [Dwo07], and ISO includes GCM as a part of ISO/IEC 19772:2020 [II20].

However, GCM has a tag truncation problem due to the linear modification technique proposed by Ferguson [Fer05] in 2005. When using a small truncated tag, the adversary

can change the ciphertext by solving a system of linear equations (or a linear system) to obtain potential successful modifications with higher probability. For example, when GCM uses a 32-bit tag, and the adversary knows the ciphertext for a message consisting of 2^{17} blocks (about 2 MB), with Ferguson's technique, the probability of an adversary forging a 32-bit tag is 2^{-16} instead of optimal 2^{-32}. The authentication key can be recovered after a successful forgery, further compromising the security of DCT.

Many nonce-based AE schemes suffer catastrophic confidentiality and integrity failures when the nonce is repeated, including GCM [MV04a] and OCB [RBBK01]. Hence, Rogaway and Shrimpton introduced the notion of Deterministic Authenticated Encryption (DAE) [RS06] at EUROCRYPT 2006, with the primary objective of addressing the key-wrap issue, as well as the nonce-misuse issue.

Numerous DAE schemes have been proposed, including SIV [RS06], GCM-SIV [GL15], AES-GCM-SIV [GLL17], and Deoxys-II [JNPS21]. All these schemes combine a conventional IV-based encryption scheme (e.g., CTR mode) and a PRF-secure authentication scheme. The security of SIV does not depend on the freshness of the nonce. Since SIV requires two independent keys, it increases the key management overhead. In 2009, Iwata and Yasuda proposed a single-key mode named HBS [IY09b] to achieve the DAE goal. HBS also accelerated the speed by employing a polynomial universal hashing rather than blockcipher-based MAC. Subsequently, they proposed BTM [IY09a], which requires only one blockcipher key as HBS and does not require the decryption algorithm of the underlying blockcipher, whereas HBS does.

Schemes like SIV [RS06] and HBS [IY09b] suffer from the so-called birthday attack. Assuming that the block size of the underlying blockcipher is n-bit, after about $2^{n/2}$ queries, the adversary will obtain a successful attack with high probability. For example, Ferguson [Fer02] proposed a birthday-bound forgery attack on OCB. BBB schemes are secure for above $2^{n/2}$ queries. BBB security is, therefore, a desirable goal for DAE.

In 2016, Forler et al. [FLLW16] proposed a beyond-birthday-bound DAE scheme named DCT (Deterministic Counter in Tweak), which is inspired by the CTRT (*CounTeR in Tweak*) encryption scheme [PS16] and the BRW polynomial [Ber07]. DCT encodes τ-bit of redundancy in the message and then encrypts it using the Hash-Counter [Min16, DK22] approach to obtain a BBB DAE scheme. DCT can obtain different integrity strengths by selecting different values of τ. They also proposed an efficient implementation that requires only a single key.

DCT uses a BRW (Bernstein Rabin Winograd) polynomial to instantiate its universal hash function (UHF), which was proposed by Bernstein [Ber07] in 2007, based on the work of Rabin and Winograd [RW72]. The BRW polynomial is faster than the UHF used by GCM because the former requires only half as many multiplications over finite fields. The BRW polynomial is widely used in many schemes, including tweakable enciphering schemes [Sar09, Sar11], message authentication codes (MACs) [CGS17], universal hash functions [GS19], authenticated encryption schemes [FLLW16], etc.

Our contributions. In this paper, we describe several forgery attacks on the DCT scheme and give a modification. Our attack results are summarized in Table 1.

1) We propose a systematic technique to linearize the UHF employed by the instantiation of DCT, which is used in subsequent attacks. We extend this technique to the case where the message length is no longer limited to a special value.
2) We show that although DCT employs the BRW polynomial to instantiate its UHF, it still suffers from a small stretch problem similar to that of GCM. When τ is small, the adversary can query the encryption oracle so that the input to the UHF of DCT is a particular $(2^{u+2}-2)$-block messages where $2 \leq u \leq \tau$, and then attack the integrity of DCT with $2^{\tau-u}$ decryption queries. This attack can efficiently balance the space complexity $\mathcal{O}(2^{u+2})$ and the time complexity $\mathcal{O}(2^{\tau-u})$ for a user-selected parameter u. We extend this attack while the message length of input to the UHF

is no longer limited to $2^{u+2}-2$, making our attacks more general. Next, we find that $2^{u+2}-3$ is the minimal length needed to perform our forgery attack mentioned above when u is fixed.

3) To solve the above small stretch problem, we propose a variant of DCT named Robust DCT (RDCT) with minimal modification, and we prove the DAE security of RDCT. When τ is small, our proof shows that a successful forgery attack requires $\mathcal{O}(2^{\tau})$ decryption queries.

Table 1: Comparing the attack complexity among GCM, DCT, and RDCT schemes. n is the size of the message block. m is the maximum number of blocks of a query. q is the number of queries. τ is the number of bits in the GCM tag or the redundancy of DCT and RDCT. u is a user-selected parameter, $2 \leq u \leq \tau$. The query length is the input length of the underlying UHF.

Scheme	Provable security	Query complexity		Query length	Ref.
		Encryption	Decryption		
GCM	$\mathcal{O}(\frac{q^2m^2}{2^n}+\frac{qm}{2^{\tau}})$	1	$2^{\tau-u}$	2^{u+1}	[Fer05]
DCT	$\mathcal{O}(\frac{q^2m^2}{2^{2n}}+\frac{qm^2}{2^{\tau}})$	1	$2^{\tau-u}$	$2^{u+2}-3$	Sect. 5.4
RDCT	$\mathcal{O}(\frac{q^2m^2}{2^{2n}}+\frac{q}{2^{\tau}-q})$	0	2^{τ}	1	Sect. 6

Table 1 compares the complexity of the attack among GCM, DCT, and RDCT. GCM's attack requires one encryption query and $2^{\tau-u}$ decryption queries with 2^{u+1} blocks to the UHF. However, our attack on DCT requires a longer message length than the attack on GCM, but the query complexity for attacking both schemes is at the same magnitude. We remark that we only attack the DCT scheme with a small stretch length. When users select a longer, such as $2n$-bit stretch, the complexity of our attack is impractical.

Organization. The paper is structured as follows: after Section 3 reviews the linear modification technique and Section 4 introduces the DCT scheme, Section 5 discusses the small stretch problem of DCT, Section 6 presents our new scheme named RDCT. Section 7 gives a summary of this work.

2 Preliminaries

2.1 Notations

We define $\mathbb{Z}^+$ as the set of positive integers. We define $\{0,1\}^*$ as the set of arbitrary length strings, $\{0,1\}^{\geq u}$ as the set of strings with length no less than u. We use lowercase letters x, y as integers, uppercase letters X, Y as strings or functions, and curlicue uppercase letters $\mathcal{X}, \mathcal{Y}$ as sets. Let $X\|Y$ represent the concatenation of strings X and Y, and let $X \oplus Y$ represent the result of their bitwise XOR. We denote $\emptyset$ as the empty set, $|X|$ as the length of X, $x \xleftarrow{\$} \mathcal{X}$ as the element x chosen uniformly at random from the set $\mathcal{X}$. We define MSB_{τ} as the most significant τ bits and LSB_{τ} as the least significant τ bits. We define $\mathsf{Perm}(\mathcal{S})$ as the set of all permutations on $\mathcal{S}$; $\widetilde{\mathsf{Perm}}(\mathcal{T},\mathcal{S})$ as the set of all tweakable permutations on $\mathcal{S}$ with tweak space $\mathcal{T}$; Let $\widetilde{\mathsf{Func}}(\mathcal{X},\mathcal{Y})$ be all functions with domain space $(\mathcal{X},\mathcal{Y})$ and range space $\mathcal{Y}$, it is also the set of all tweakable functions on domain/range space $\mathcal{Y}$ with tweak space $\mathcal{X}$.

We define $\Pr[V]$ as the probability of event V. We assume that an adversary $\mathbf{A}$ runs in time at most t, makes at most q queries of at most m blocks in total, and can interact

with several given oracles as black boxes. We denote by $\mathbf{A}^{\mathcal{O}} \Rightarrow b$ where $b = 0$ or 1 the output of $\mathbf{A}$ after interacting with an oracle $\mathcal{O}$. We write

$$\mathbf{Adv}_E^{\mathrm{GOAL}}(\mathbf{A}) \coloneqq |\Pr[\mathbf{A}^R \Rightarrow 1] - \Pr[\mathbf{A}^I \Rightarrow 1]|,$$

as the advantage of $\mathbf{A}$ to distinguish between oracles R and I, where GOAL is the security goal, E is the attacking object, R is the real oracle and I is the ideal oracle. We define $\mathbf{Adv}_E^{\mathrm{GOAL}}(q, m, t) := \max_{\mathbf{A}} \left\{ \mathbf{Adv}_E^{\mathrm{GOAL}}(\mathbf{A}) \right\}$ as the maximum of $\mathbf{Adv}_E^{\mathrm{GOAL}}(\mathbf{A})$ over all adversaries against the GOAL-security of E that run in time at most t and make at most q queries of at most m blocks in total.

2.2 Definition of Universal Hash Functions

Definition 1 (Universal Hash Functions)**.** Let $\mathcal{X} \subseteq \{0,1\}^*, \mathcal{Y} \subseteq \{0,1\}^n$, and $\mathcal{K}$ is a key space. $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is *ϵ-almost-universal (ϵ-AU)*, if for all distinct elements $X, X' \in \mathcal{X}$, it holds that $\Pr[K \xleftarrow{\$} \mathcal{K} : H_K(X) = H_K(X')] \leq \epsilon$. H is *ϵ-almost-XOR-universal (ϵ-AXU)*, if for all distinct elements $X, X' \in \mathcal{X}$ and $Y \in \mathcal{Y}$, it holds $\Pr[K \xleftarrow{\$} \mathcal{K} : H_K(X) \oplus H_K(X') = Y] \leq \epsilon$.

2.3 Security of (Tweakable) Blockciphers

Definition 2 ((Strong) PRP Advantage)**.** Fix integers $n, k \geq 1$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and $\mathbf{A}(\mathbf{A}')$ be an adversary with access to an oracle (two oracles). Let $K \xleftarrow{\$} \{0,1\}^k$ and $\pi \xleftarrow{\$} \mathsf{Perm}(\{0,1\}^n)$. Then, the PRP and SPRP advantages of $\mathbf{A}$ with respect to E are defined as $\mathbf{Adv}_E^{\mathrm{PRP}}(\mathbf{A}) \coloneqq |\Pr[\mathbf{A}^{E_K} \Rightarrow 1] - \Pr[\mathbf{A}^{\pi} \Rightarrow 1]|$ and $\mathbf{Adv}_E^{\mathrm{SPRP}}(\mathbf{A}) \coloneqq |\Pr[\mathbf{A}^{E_K, E_K^{-1}} \Rightarrow 1] - \Pr[\mathbf{A}^{\pi, \pi^{-1}} \Rightarrow 1]|$, respectively.

Definition 3 ((Strong) Tweakable PRP Advantage)**.** Fix integers $n, k \geq 1$. Let $\mathcal{T}$ denote a non-empty set. Let $\widetilde{E} : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable blockcipher and $\mathbf{A}$ be an adversary with access to an oracle (two oracles). Let $K \xleftarrow{\$} \{0,1\}^k$ and $\widetilde{\pi} \xleftarrow{\$} \widetilde{\mathsf{Perm}}(\mathcal{T}, \{0,1\}^n)$. Then, the TPRP and STPRP advantages of $\mathbf{A}$ with respect to $\widetilde{E}$ are defined as $\mathbf{Adv}_{\widetilde{E}}^{\mathrm{TPRP}}(\mathbf{A}) \coloneqq |\Pr[\mathbf{A}^{\widetilde{E}_K} \Rightarrow 1] - \Pr[\mathbf{A}^{\widetilde{\pi}} \Rightarrow 1]|$ and $\mathbf{Adv}_{\widetilde{E}}^{\mathrm{STPRP}}(\mathbf{A}) \coloneqq |\Pr[\mathbf{A}^{\widetilde{E}_K, \widetilde{E}_K^{-1}} \Rightarrow 1] - \Pr[\mathbf{A}^{\widetilde{\pi}, \widetilde{\pi}^{-1}} \Rightarrow 1]|$, respectively.

2.4 Security of IV-Based Encryption Schemes

An *IV-based encryption scheme* [BDJR97] is a tuple $\Pi = (\mathcal{E}, \mathcal{D})$ of encryption $\mathcal{E} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \to \mathcal{C}$ and decryption $\mathcal{D} : \mathcal{K} \times \mathcal{IV} \times \mathcal{C} \to \mathcal{M}$ algorithms with IV space $\mathcal{IV}$, key space $\mathcal{K}$, and message/ciphertext space $\mathcal{M}, \mathcal{C} \subseteq \{0,1\}^*$. For any query M, encryption oracle samples uniformly at random $IV \xleftarrow{\$} \mathcal{IV}$ and computes the ciphertext $C \leftarrow \mathcal{E}_K^{IV}(M)$. The real oracle $\mathcal{E}_K$ outputs $IV \| C$, and the random oracle $\$^{\mathcal{E}}$ outputs a random string as long as $|IV \| \mathcal{E}_K^{IV}(M)|$.

Definition 4 (ivE Advantage)**.** $K \xleftarrow{\$} \mathcal{K}$, let $\Pi = (\mathcal{E}, \mathcal{D})$ be an IV-based encryption scheme. Let $\mathbf{A}$ be an adversary with access to an oracle. Then, the ivE advantage of $\mathbf{A}$ over Π is defined as $\mathbf{Adv}_{\Pi}^{\mathrm{ivE}}(\mathbf{A}) \coloneqq |\Pr[\mathbf{A}^{\mathcal{E}_K} \Rightarrow 1] - \Pr[\mathbf{A}^{\$^{\mathcal{E}}} \Rightarrow 1]|$.

3 Linear Modification Technique

Each element $U = u_0u_1 \cdots u_{127}$ in the Galois Field $\mathbb{GF}(2^{128})$ can be represented by an 127-degree polynomial $U(x) = u_0x^{127} + \cdots + u_{126}x + u_{127}$ over $\mathbb{GF}(2)$. We denote $\overline{U} = \begin{pmatrix} u_0 \\ \vdots \\ u_{127} \end{pmatrix}$ as the column vector of the coefficients of $U(x)$ over $\mathbb{GF}(2)$ and $\overline{U}^T$ as the transpose of $\overline{U}$.

3.1 Multiplication Operation

The function $F_C(U) = C \cdot U$ over $\mathbb{GF}(2^{128})$ with primitive polynomial $p(x)$ for a constant field element C is a linear function. For each C there exists a 128×128 matrix M_C over $\mathbb{GF}(2)$ such that

$$\overline{C \cdot U} = M_C\overline{U} = \left(M_C^0\ M_C^1\ \cdots\ M_C^{127}\right) \cdot \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{127} \end{pmatrix}$$

for all $U \in \mathbb{GF}(2^{128})$, where M_C^i represents the i-th column of $M_C, 0 \leq i \leq 127$. We can calculate M_C as follows:

$$\begin{cases} M_C^{127} = \overline{C \cdot x^0}, \\ M_C^{126} = \overline{C \cdot x^1}, \\ \qquad \vdots \\ M_C^0 = \overline{C \cdot x^{127}}. \end{cases}$$

$C \cdot x$ is the C left-shifted by 1 bit if the most significant bit of C is 0, further XORed by a constant otherwise. Therefore, each column vector of M_C is a linear combination of C.

3.2 Square Operation

Due to the fact that $\mathbb{GF}(2^{128})$ is a field of characteristic of 2, which implies that $(A+B)^2 = A^2 + B^2$ for any $A, B \in \mathbb{GF}(2^{128})$. Therefore, the function $F_S(U) = U^2$ over $\mathbb{GF}(2^{128})$ with primitive polynomial $p(x)$ is a linear function. Thus, there exists a fixed matrix M_S such that

$$\overline{U^2} = M_S\overline{U} = \left(M_S^0\ M_S^1\ \cdots\ M_S^{127}\right) \cdot \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{127} \end{pmatrix}$$

for all $U \in \mathbb{GF}(2^{128})$. We can calculate M_S as follows:

$$\begin{cases} M_S^{127} = \overline{x^0}, \\ M_S^{126} = \overline{x^2}, \\ \qquad \vdots \\ M_S^0 = \overline{(x^{127})^2}. \end{cases}$$

Note that M_S does not depend on anything except the chosen Galois Field. Therefore, M_S is a fixed matrix.

3.3 Concrete Attack Against GCM

GCM, proposed by McGrew and Viega, is a famous authenticated encryption scheme. GCM follows the Encrypt-then-MAC mechanism [NRS14] with a CTR-like encryption scheme and a polynomial-based UHF. For more details, please refer to [MV04b].

This section introduces Ferguson's linear modification technique [Fer05], which attacks the integrity of GCM by changing the ciphertext $C \coloneqq C_1\|C_2\|\cdots\|C_m$ without changing the tag. Since Ferguson's attack does not use the associated data, we will ignore it. Then, the authentication function can be denoted as

$$T \coloneqq R \oplus \sum_{i=1}^{m} C_i H^i,$$

where $H = E_K(0)$ is the authentication key and $R = E_K(N\|1)$ is a 128-bit string, which holds only for 96-bit nonces, $C_1 = |A|_{64}\||C|_{64}$ is length information of the inputs, where $|X|_m$ denotes the length of X by an m-bit string. The tag after truncation is the most significant τ-bit of the T, denoted by $MSB_\tau(T)$. The attacking goal is to find $D \coloneqq D_1\|D_2\|\cdots\|D_m$ such that

$$MSB_\tau\left(\sum_{i=1}^{m} D_i H^i\right) = 0^\tau.$$

So we can obtain the collision

$$MSB_\tau(T) = MSB_\tau\left(T \oplus \sum_{i=1}^{m} D_i H^i\right)$$

for any two ciphertexts C and $C \oplus D$ of equal length, which means if we query GCM to obtain a ciphertext triple (N, C, T), we can forge with $(N, C \oplus D, T)$ successfully. The concrete steps of searching D are as follows.

1. Adjust the search goal to the coefficients D_{2^i} $(i \geq 0)$, such that

   $$MSB_u(\sum_i D_{2^i} H^{2^i}) = 0^u,$$

   where the parameter $u \leq \tau$. We focus on D_{2^i}s for the reason that only $\sum_i D_{2^i} H^{2^i}$ is a linear function about multiplication and square operations (see Section 3.1 and Section 3.2, respectively). For coefficients where j is not a power of two, let $D_j = 0^n$. Since the first ciphertext block encodes the length information, we do not change it and let $D_1 = 0^n$.

2. Represent the linear function $T_1 = \sum_i D_{2^i} H^{2^i}$ in terms of H by matrix over the finite field. Consider T_1 as bit vector $\overline{T_1}$ and H as $\overline{H}$:

   $$\overline{T_1} = \sum_i M_{D_{2^i}} (M_S)^{2^i} \overline{H},$$

   where the matrix $M_{D_{2^i}}$ represents the operation corresponding to multiplication with D_{2^i}, and the elements in $M_{D_{2^i}}$ are all linear combinations of bits of D_{2^i}. M_S is a fixed matrix that represents the square operation. To force u-bit of $\overline{T_1}$ to zero, we need to create $u \times 128$ linear equations about D_{2^i} such that u rows of $\sum_i M_{D_{2^i}} (M_S)^{2^i}$ are completely zero. When the number of D_{2^i} is $u+1$, which corresponds to $(u+1) \times 128$ free variables (or unknowns), the number of unknowns exceeds the number of equations, we can obtain non-zero solutions of D, and the size of the solution set of the linear system is at least 2^{128}.

3. Continue to perform about $2^{\tau-u}$ decryption queries in search of the remaining $(\tau-u)$-bit tag corresponding to the modified ciphertext $C \oplus D$, until leading to a successful forgery $(N, C \oplus D, T)$.

For example, when GCM uses a 32-bit tag, the attack consists of the following steps: First, assume that the adversary can obtain the ciphertext for a message of $m = 2^{17}$ blocks (about 2 MB) by encryption queries, which corresponds to 17×128 unknowns. Second, suppose that the number of unknowns is greater than the number of linear equations; The adversary can calculate non-zero solutions of $D_{2^1}, D_{2^2}, \cdots, D_{2^{17}}$ that make the 16 rows of the $\sum_i M_{D_{2^i}}(M_S)^{2^i}$ equal to zero by creating 16×128 constraint equations about $D_{2^i}, 1 \leq i \leq 17$. Third, the adversary continues to perform about $q = 2^{16}$ decryption queries in search of the remaining 16-bit tag corresponding to the modified ciphertext, leading to a successful forgery.

More generally, assuming that the length of the truncated tag is τ, the adversary knows the ciphertext of a message of $m = 2^{u+1}$ blocks and can successfully obtain a forge with $q = 2^v$ queries, which $u + v = \tau$. This technique can efficiently balance the number of message blocks m selected by the adversary and the number of queries q needed for the forgery.

4 A DAE Scheme: DCT

In this section, we introduce the notion of DAE [RS06] and the DCT scheme [FLLW16].

4.1 DAE Scheme

A DAE scheme [RS06] is a tuple $\widetilde{\Pi} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ of deterministic algorithms $\widetilde{\mathcal{E}} : \mathcal{K} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C}$ and $\widetilde{\mathcal{D}} : \mathcal{K} \times \mathcal{A} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$ with key space $\mathcal{K}$, associated-data space $\mathcal{A}$, and message/ciphertext space $\mathcal{M}, \mathcal{C} \subseteq \{0,1\}^*$. For $K \in \mathcal{K}, A \in \mathcal{A}, M \in \mathcal{M}$, $\widetilde{\mathcal{E}}_K$ maps (A, M) to an output C such that $|C| = |M| + \tau$ for a fixed stretch length τ. $\widetilde{\mathcal{D}}_K(A, C)$ outputs the corresponding message M if C is valid or $\perp$ otherwise, where $\perp$ is a symbol of decryption failures. If we need a nonce in DAE, we can take a part of A as the nonce.

A DAE scheme achieves both confidentiality and integrity. Confidentiality means that the adversary cannot obtain any information about plaintext from the corresponding ciphertext except the length; Integrity means that the adversary cannot generate a fresh pair of ciphertext and tag (not previously generated by the encryption oracle) to pass the decryption verification. We define detPriv, detAuth as the confidentiality and integrity of the DAE scheme, respectively, as follows.

Definition 5 (Confidentiality and Integrity Advantages)**.** Let $\widetilde{\Pi} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ be a DAE scheme and $K \xleftarrow{\$} \mathcal{K}$. The adversary $\mathbf{A}$ has access to one or two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. $\mathbf{A}$ is not allowed to ask $\mathcal{O}_2$ with the result of querying $\mathcal{O}_1$. $\mathbf{A}$ does not repeat a query. Then, the detPriv and detAuth advantages of $\mathbf{A}$ with respect to $\widetilde{\Pi}$, are defined as

$$\mathbf{Adv}^{\text{detPriv}}_{\widetilde{\Pi}}(\mathbf{A}) \coloneqq \left| \Pr\left[\mathbf{A}^{\widetilde{\mathcal{E}}_K} \Rightarrow 1\right] - \Pr\left[\mathbf{A}^{\$^{\widetilde{\mathcal{E}}}} \Rightarrow 1\right] \right|,$$
$$\mathbf{Adv}^{\text{detAuth}}_{\widetilde{\Pi}}(\mathbf{A}) \coloneqq \Pr\left[\mathbf{A}^{\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K} \text{ forges } \right],$$

where "forges" means that $\mathbf{A}$ asks $\widetilde{\mathcal{D}}_K$ with (A, C) and returns anything other than $\perp$ for at least one query among multiple decryption queries.

Rogaway and Shrimpton [RS06] introduced the "all-in-one" definition, which is equivalent to the two-part notion that requires deterministic confidentiality detPriv and deterministic authenticity detAuth. The relation between it and detPriv and detAuth is as follows.

Theorem 1 (DAE Advantage [FLLW16]). *Let* $\widetilde{\Pi} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ *be a* DAE *scheme and* $K \xleftarrow{\$} \mathcal{K}$. *Let* $\mathbf{A}$ *be a* DAE *adversary on* $\widetilde{\Pi}$ *with access to two oracles* $\mathcal{O}_1$ *and* $\mathcal{O}_2$, $\mathbf{A}$ *is not allowed to ask* $\mathcal{O}_2$ *with the result of querying* $\mathcal{O}_1$. *Then, the* DAE *advantages of* $\mathbf{A}$ *with respect to* $\widetilde{\Pi}$ *is defined as*

$$\mathbf{Adv}^{\mathrm{DAE}}_{\widetilde{\Pi}}(\mathbf{A}) \coloneqq \left| \Pr\left[\mathbf{A}^{\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K} \Rightarrow 1\right] - \Pr\left[\mathbf{A}^{\$^{\widetilde{\mathcal{E}}}, \perp} \Rightarrow 1\right] \right|,$$

where $\mathbf{A}$ *runs in time at most* t *and asks* q_e *queries to its left oracles, and asks* q_d *queries to its right oracles,* $\mathbf{A}$ *asks at most* m *blocks in total. Then, there exists a* detPriv *adversary* $\mathbf{A}_1$ *and a* detAuth *adversary* $\mathbf{A}_2$ *both against* $\widetilde{\Pi}$, *such that*

$$\mathbf{Adv}^{\mathrm{DAE}}_{\widetilde{\Pi}}(\mathbf{A}) \leq \mathbf{Adv}^{\mathrm{detPriv}}_{\widetilde{\Pi}}(\mathbf{A}_1) + \mathbf{Adv}^{\mathrm{detAuth}}_{\widetilde{\Pi}}(\mathbf{A}_2),$$

where $\mathbf{A}_1$ *make at most* q_e *queries with a maximum of* m *blocks,* $\mathbf{A}_2$ *make at most* $q = q_e + q_d$ *queries with a maximum of* m *blocks and they both run in time* $\mathcal{O}(t)$.

4.2 The DCT Scheme

Forler et al. [FLLW16] proposed the DCT scheme, a beyond-birthday-bound DAE scheme. We show the encryption of DCT in Figure 1.

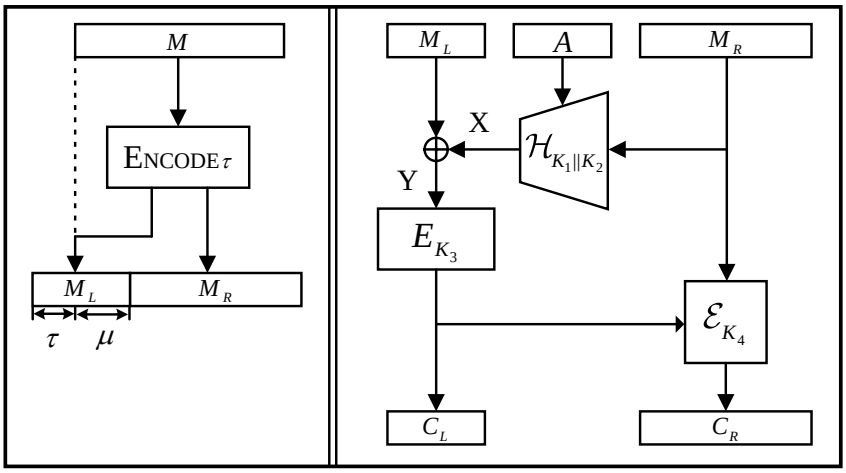

Figure 1: The $\textsc{Encode}_\tau$ (**left**) process and the encryption process of DCT (**right**).

Fix the parameters $n, \tau \geq 1$ with $\tau \leq 2n$. Let $\mu = 2n - \tau$. Let $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ and $\mathcal{K}_4$ be non-empty key spaces and $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \mathcal{K}_3 \times \mathcal{K}_4$. Let $\mathcal{A} \subseteq \{0,1\}^*, \mathcal{M} \subseteq \{0,1\}^{\geq \mu}, \mathcal{C} \subseteq \{0,1\}^{\geq 2n}$ denote the associated-data space, message space, and ciphertext space, respectively. Let $\mathcal{H} : \mathcal{K}_1 \times \mathcal{K}_2 \times \mathcal{A} \times \{0,1\}^* \to \{0,1\}^{2n}$ be an AXU hash function with key space $\mathcal{K}_1 \times \mathcal{K}_2$. Let $E : \mathcal{K}_3 \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ be a blockcipher with key space $\mathcal{K}_3$. Let $\Pi_1 = (\mathcal{E}, \mathcal{D})$ be an IV-based encryption scheme with key space $\mathcal{K}_4$ and IV space $\mathcal{IV} = \{0,1\}^{2n}$. Let $\Pi_2 = (\textsc{Encode}_\tau, \textsc{Decode}_\tau)$ be an encode scheme with encoding function and decoding function

$$\textsc{Encode}_\tau : \mathcal{M} \to \{0,1\}^{2n} \times \{0,1\}^{|M|-\mu},$$
$$\textsc{Decode}_\tau : \{0,1\}^{2n} \times \{0,1\}^{|M|-\mu} \to \mathcal{M} \cup \{\perp\},$$

where $\textsc{Encode}_\tau$ is an injection, encodes τ-bit redundancy into the input. The τ-bit redundancy is fully contained in the left part of the output. For example, $\textsc{Encode}_\tau(M) = (M_L, M_R)$ where $M_L = 0^\tau \| MSB_\mu(M)$ and $M_R = LSB_{|M|-\mu}(M)$. The decoding function

returns a unique $M \in \mathcal{M}$ such that $\textsc{Encode}_\tau(M) = (X, Y)$ for $(X, Y) \in \{0,1\}^{2n} \times \{0,1\}^{|M|-\mu}$ if such an M exists; otherwise, it returns $\perp$. Then, the DCT scheme $\mathrm{DCT}_{\mathcal{H},E,\Pi_1,\Pi_2} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ based on $\mathcal{H}, E, \Pi_1$ and Π_2 is in Algorithm 1.

Algorithm 1 Encryption and decryption of DCT

1: **function** $\widetilde{\mathcal{E}}_{K_1,K_2,K_3,K_4}(A, M)$
2: $\quad(M_L, M_R) \leftarrow \textsc{Encode}_\tau(M)$
3: $\quad C_L \leftarrow E_{K_3}\left(M_L \oplus \mathcal{H}_{K_1\|K_2}(A, M_R)\right)$
4: $\quad C_R \leftarrow \mathcal{E}_{K_4}(C_L, M_R)$
5: $\quad$ **return** $(C_L \| C_R)$
6: **end function**
7: **function** $\widetilde{\mathcal{D}}_{K_1,K_2,K_3,K_4}(A, C)$
8: $\quad(C_L, C_R) \leftarrow C$
9: $\quad M_R \leftarrow \mathcal{D}_{K_4}(C_L, C_R)$
10: $\quad M_L \leftarrow E^{-1}_{K_3}(C_L) \oplus \mathcal{H}_{K_1\|K_2}(A, M_R)$
11: $\quad$ **return** $\textsc{Decode}_\tau(M_L, M_R)$
12: **end function**

4.3 Instantiation of DCT

The instantiation of DCT employs a CTR-like encryption scheme as $\mathcal{E}_{K_4}(IV, M) = e_{K_4}(IV) \oplus M$ where e_{K_4} generates an $|M|$-bit string. Due to the IV length of the Π_1 being $2n$-bit, DCT employs a $2n$-bit permutation as E. Π_2 encodes the τ-bit of zero into the message. DCT uses the BRW polynomial to instantiate its underlying UHF.

The attack in Section 5 focuses on the authentication security of the DCT instantiation scheme. The following section shows that DCT suffers from the so-called small stretch problem.

5 Attacks on DCT with Small Stretch

In Section 5.1, we briefly introduce the UHF used by DCT and the detailed steps of our forgery attack. In Section 5.2, we linearize the KBRW polynomial with $2^u - 1$ blocks. Next, Section 5.3 considers the minimal length problem. In Section 5.4, we analyze how to attack UHF of the instantiation of DCT, which is the so-called small stretch problem of DCT. Finally, in Section 5.5, we briefly analyze the security bounds of DCT.

5.1 The Universal Hash Function of DCT

In DCT, the universal hash function based on the BRW polynomial can be denoted as:

$$\mathcal{H}_{K_1\|K_2}(X_1, X_2) = KBRW_{K_1}(M) \| KBRW_{K_2}(M),$$

where $K_1, K_2 \in \{0,1\}^n$ are independent keys, $M = \mathrm{PAD}_n(X_1) \| \mathrm{PAD}_n(X_2) \| L$, L is an n-bit block containing the length information of the inputs, PAD_n padding the minimal number of "0" bits to make the length of the padded message a multiple of n, and $KBRW_K(M) = K \cdot BRW_K(M)$ is defined directly as follows.

Definition 6 (KBRW polynomial [FLLW16])**.** Given an m-block message $M = (M_1, \cdots, M_m)$, $M_i \in \{0,1\}^n$, the polynomial $KBRW_K(M)$ is defined as follows:

$$\begin{aligned}
&KBRW_K(\varepsilon) = 0^n;\\
&KBRW_K(M_1) = M_1 K;\\
&KBRW_K(M_1, M_2) = M_1 K^2 \oplus M_2 K;\\
&KBRW_K(M_1, M_2, M_3) = K^4 \oplus M_1 K^3 \oplus M_2 K^2 \oplus (M_1 M_2 \oplus M_3) K;\\
&KBRW_K(M_1, \cdots, M_m) = KBRW_K(M_1, \cdots, M_{t-1})(K^t \oplus M_t) \oplus\\
&\qquad KBRW_K(M_{t+1}, \cdots, M_m) \text{ if } t \le m < 2t \text{ for } t = 2^i, i \ge 2.
\end{aligned}$$

Let ε represent the empty string. All operations in the KBRW polynomial are performed over $\mathbb{GF}(2^n)$, the Galois Field with a given primitive polynomial $p(x)$ of degree n. For $n = 128$, $p(x) = x^{128} + x^7 + x^2 + x + 1$.

Since the associated data are irrelevant to our attacks, we ignore them for convenience. For DCT with $\tau \leq 2n$ and $A = \varepsilon$, when we obtain the ciphertext (C_L, C_R) corresponding to (M_L, M_R), we have the following equation:

$$MSB_\tau(M_L) = MSB_\tau(E_{K_3}^{-1}(C_L) \oplus KBRW_K(M_R)) = 0^\tau.$$

We can query the decryption of DCT with $C_L \| C'_R$ where only the value of C_R is modified to C'_R. Note that the IV-based encryption scheme in DCT is a CTR-like encryption scheme. The forgery is successful if and only if the following equation is established:

$$MSB_\tau(M_L) = MSB_\tau(E_{K_3}^{-1}(C_L) \oplus KBRW_K(M_R \oplus C_R \oplus C'_R)) = 0^\tau.$$

So the forgery attack is reduced to the problem of looking for a modification string $D = C_R \oplus C'_R = M_R \oplus M'_R$ while keeping $MSB_\tau(KBRW_K(M)) = MSB_\tau(KBRW_K(M \oplus D))$, the same problem in Section 3 but with a different universal hash function.

5.2 Linearizing KBRW with Special Length Message

Assume the adversary queries the KBRW polynomial with the message of length $m = 2^u - 1$ for the sake of convenience. When $u = 2, M = (M_1, M_2, M_3)$ and

$$KBRW_K(M) = K^4 \oplus M_1K^3 \oplus M_2K^2 \oplus (M_1M_2 \oplus M_3)K. \tag{1}$$

Let M_1 remain invariable ($D_1 = 0$), and only modify M_2 and M_3 by unknowns D_2 and D_3, respectively, so that $KBRW_K(M) \oplus KBRW_K(M \oplus D) = D_2K^2 \oplus (M_1D_2 \oplus D_3)K$ is a linear function of K, where $D = (D_1, D_2, D_3)$, and we can calculate D_2 and D_3 by the technique outlined in Section 3.

When $u = 3$, the situation becomes complicated as

$$\begin{aligned} KBRW_K(M) = {} & K^8 \oplus M_1K^7 \oplus M_2K^6 \oplus (M_1M_2 \oplus M_3)K^5 \\ & \oplus (M_4 \oplus 1)K^4 \oplus (M_1M_4 \oplus M_5)K^3 \oplus (M_2M_4 \oplus M_6)K^2 \\ & \oplus (M_1M_2M_4 \oplus M_3M_4 \oplus M_5M_6 \oplus M_7)K. \end{aligned} \tag{2}$$

We choose M_4, M_6, and M_7 as blocks modified by unknowns. Still, some message blocks cannot be chosen arbitrarily: for example, the coefficient of K^4 is $M_4 \oplus 1$, and the coefficient of K^3 is $M_1M_4 \oplus M_5$. To obtain the linear function, M_1 and D_1 must be 0.

It is easy to find that each block value depends on its location. For $u \geq 2$, to linearize $KBRW_K(M) \oplus KBRW_K(M \oplus D)$, we divide the message blocks into six disjoint sets: $\mathcal{V}_0^u$, $\mathcal{V}_1^u$, $\mathcal{A}_0^u$, $\mathcal{A}_1^u$, $\mathcal{F}_0^u$ and $\mathcal{F}_1^u$ according to the subscript of the message blocks, which specifies how these blocks should be valued and modified.

- $\mathcal{V}_0^u$ and $\mathcal{V}_1^u$ are sets of blocks that can be chosen arbitrarily and modified by unknowns. If the block M_i appears as the coefficient of a power term in the form of $M_i \oplus 1$, we put it in $\mathcal{V}_1^u$, otherwise in $\mathcal{V}_0^u$;

- $\mathcal{A}_0^u$ and $\mathcal{A}_1^u$ are sets of blocks that can be chosen arbitrarily but not modified by unknowns. The difference between $\mathcal{A}_0^u$ and $\mathcal{A}_1^u$ is the same as the above;

- $\mathcal{F}_0^u$ and $\mathcal{F}_1^u$ are sets of blocks that are fixed as 0 and 1 respectively and not modified by unknowns. The difference between $\mathcal{F}_0^u$ and $\mathcal{F}_1^u$ is the same as above.

For example, when $u = 2$, consider Equation (1). Let $\mathcal{F}_0^2 = \mathcal{F}_1^2 = \mathcal{V}_1^2 = \mathcal{A}_1^2 = \emptyset$, $\mathcal{V}_0^2 = \{M_2, M_3\}$ and $\mathcal{A}_0^2 = \{M_1\}$, which means that the value of M_1 should remain invariable in our forgery attacks ($D_1 = 0$). So that $KBRW_K(M) \oplus KBRW_K(M \oplus D) = D_2K^2 \oplus (M_1D_2 \oplus D_3)K$ is a linear function of K.

For $u > 2$, we intend to obtain these sets recursively. Equation (2) can also be denoted as:

$$KBRW_K(M) = KBRW_K(M_1, M_2, M_3)(K^4 \oplus M_4) \oplus KBRW_K(M_5, M_6, M_7). \tag{3}$$

Similarly to the analysis of $KBRW_K(M_1, M_2, M_3)$, in $KBRW_K(M_5, M_6, M_7)$ we choose $\mathcal{V}_0^3$ by the case with $u = 2$, and they originate from the subscript plus 4 of elements in $\mathcal{V}_0^2$, as shown in Figure 2, so that $\mathcal{V}_0^3 = \{M_{i+2^2} | M_i \in \mathcal{V}_0^2\} = \{M_6, M_7\}$. Thus, for general $u > 2$, we have $\mathcal{V}_0^u = \{M_{i+2^{u-1}} | M_i \in \mathcal{V}_0^{u-1}\}$.

Note that the term $(M_4 \oplus 1)K^4$ in Equation (2), we choose $\mathcal{V}_1^3 = \{M_4\}$. Because the degree of $M_1M_4K^3$ is not a power of two in $KBRW_K(M_1, M_2, M_3)M_4$ of Equation (3), we must force $\mathcal{F}_0^3 = \mathcal{A}_0^2 = \{M_1\}$ ($M_1 = D_1 = 0$) to make sure M_4 as an unknowns. Thus, for general $u > 2$, we have $\mathcal{F}_0^u = \mathcal{F}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_0^{u-1}\} \bigcup \mathcal{A}_0^{u-1}$. So that $KBRW_K(M) \oplus KBRW_K(M \oplus D) = D_4K^4 \oplus (M_2D_4 \oplus D_6)K^2 \oplus (M_3D_4 \oplus M_5D_6 \oplus D_7)K$ is a linear function of K.

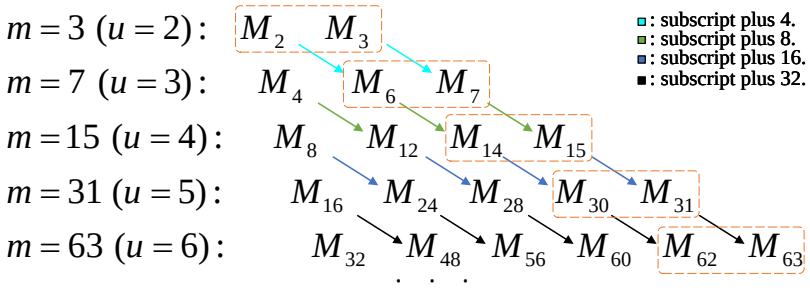

Figure 2: Calculating the set $\mathcal{V}_0^u$ and $\mathcal{V}_1^u$ recursively. The different colored arrow indicates that the subscript of the elements plus with different t. E.g., the elements in $\mathcal{V}_0^u$ originate from the subscript plus $t = 2^{u-1}$ of elements in $\mathcal{V}_0^{u-1}$, $u \geq 2$. The elements in the orange dashed box belong to the set $\mathcal{V}_0^u$, and the rest belong to the set $\mathcal{V}_1^u$.

Consider the general case when $u \geq 2$, we formalize the above deduction as the following theorem.

Theorem 2. *For the KBRW polynomial, assume $m = 2^u - 1$, $u \geq 2$. Let $\mathcal{V}_0^2 = \{M_2, M_3\}$, $\mathcal{A}_0^2 = \{M_1\}$, and initialize the remaining set to $\emptyset$. We can obtain the following recursions:*

$$\begin{aligned}
\mathcal{V}_0^u &= \{M_{i+2^{u-1}} | M_i \in \mathcal{V}_0^{u-1}\}, \\
\mathcal{V}_1^u &= \{M_{2^{u-1}}\} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{V}_1^{u-1}\}, \\
\mathcal{A}_0^u &= \mathcal{V}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{A}_0^{u-1}\}, \\
\mathcal{A}_1^u &= \mathcal{V}_1^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{A}_1^{u-1}\}, \\
\mathcal{F}_0^u &= \mathcal{F}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_0^{u-1}\} \bigcup \mathcal{A}_0^{u-1}, \\
\mathcal{F}_1^u &= \mathcal{F}_1^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_1^{u-1}\} \bigcup \mathcal{A}_1^{u-1},
\end{aligned} \tag{4}$$

where $i \in \mathbb{Z}^+$. Then, after assigning the message blocks according to the recursions above, $KBRW_K(M) \oplus KBRW_K(M \oplus D)$ is a linear function of K.

Proof. When $u = 3$, by analyzing $KBRW_K(M_1, \cdots, M_7)$, we can obtain the following conclusions:

$$\begin{aligned}
&\mathcal{V}_0^3 = \{M_{i+2^2} | M_i \in \mathcal{V}_0^2\} = \{M_6, M_7\},\\
&\mathcal{V}_1^3 = \{M_{2^2}\} \bigcup \{M_{i+2^2} | M_i \in \mathcal{V}_1^2\} = \{M_4\},\\
&\mathcal{A}_0^3 = \mathcal{V}_0^2 \bigcup \{M_{i+2^2} | M_i \in \mathcal{A}_0^2\} = \{M_2, M_3, M_5\},\\
&\mathcal{A}_1^3 = \mathcal{V}_1^3 \bigcup \{M_{i+2^2} | M_i \in \mathcal{A}_1^2\} = \emptyset,\\
&\mathcal{F}_0^3 = \mathcal{F}_0^2 \bigcup \{M_{i+2^2} | M_i \in \mathcal{F}_0^2\} \bigcup \mathcal{A}_0^2 = \{M_1\},\\
&\mathcal{F}_1^3 = \mathcal{F}_1^2 \bigcup \{M_{i+2^2} | M_i \in \mathcal{F}_1^2\} \bigcup \mathcal{A}_1^2 = \emptyset,
\end{aligned}$$

which means when M_2, M_3 and M_5 remain invariable ($D_2 = D_3 = D_5 = 0$), and let $M_1 = 0$, $KBRW_K(M) \oplus KBRW_K(M \oplus D)$ is a linear function of K, we can calculate D_4, D_6 and D_7 using the linear modification technique stated in Section 3. Therefore, the conclusion is true.

Suppose that the conclusion is true in the case $u - 1$. Next, we consider the KBRW polynomial in case u:

$$\begin{aligned}
KBRW_K(M_1, \cdots, M_m) = &KBRW_K(M_1, \cdots, M_{2^{u-1}-1})(K^{2^{u-1}} \oplus M_{2^{u-1}})\\
&\oplus KBRW_K(M_{2^{u-1}+1}, \cdots, M_m).
\end{aligned} \tag{5}$$

Since the subscript of coefficients in $KBRW_K(M_{2^{u-1}+1}, \cdots, M_m)$ is larger than $KBRW_K(M_1, \cdots, M_{2^{u-1}-1})$ by 2^{u-1}, therefore, as shown in Figure 2, we have $\mathcal{V}_0^u = \{M_{i+2^{u-1}} | M_i \in \mathcal{V}_0^{u-1}\}$ and $\{M_{i+2^{u-1}} | M_i \in \mathcal{V}_1^{u-1}\} \subseteq \mathcal{V}_1^u$.

To linearize the above two functions, we need to assign 0 and 1 to each block in $\mathcal{F}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_0^{u-1}\}$ and $\mathcal{F}_1^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_1^{u-1}\}$, respectively. Thus, we have:

$$\begin{aligned}
&\mathcal{F}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_0^{u-1}\} \subseteq \mathcal{F}_0^u,\\
&\mathcal{F}_1^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_1^{u-1}\} \subseteq \mathcal{F}_1^u.
\end{aligned}$$

Assume the linearized Equation (5) can be denoted as:

$$\begin{aligned}
L_K(M_1, \cdots, M_m) = &L_K(M_1, \cdots, M_{2^{u-1}-1})(K^{2^{u-1}} \oplus M_{2^{u-1}})\\
&\oplus L_K(M_{2^{u-1}+1}, \cdots, M_m),
\end{aligned} \tag{6}$$

the leading term of $L_K(M_1, \cdots, M_{2^{u-1}-1})M_{2^{u-1}}$ and $L_K(M_{2^{u-1}+1}, \cdots, M_m)$ are $M_{2^{u-1}}K^{2^{u-1}}$ and $K^{2^{u-1}}$, respectively. Thus, the coefficient of $K^{2^{u-1}}$ in Equation (6) is $M_{2^{u-1}} \oplus 1$. Therefore, we have $\mathcal{V}_1^u = \{M_{2^{u-1}}\} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{V}_1^{u-1}\}$.

We continue linearizing Equation (6). Since $L_K(M_1, \cdots, M_{2^{u-1}-1})$ times $K^{2^{u-1}}$ raises the degree of each term by 2^{u-1}, so none of the degrees of each term in this function are powers of two expect the leading term K^{2^u}. To linearize it, each block in $\mathcal{V}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{A}_0^{u-1}\}$ and $\mathcal{V}_1^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{A}_1^{u-1}\}$ should keep unchanged when forgery. Thus, we have:

$$\begin{aligned}
&\mathcal{A}_0^u = \mathcal{V}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{A}_0^{u-1}\},\\
&\mathcal{A}_1^u = \mathcal{V}_1^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{A}_1^{u-1}\}.
\end{aligned}$$

Because $M_{2^{u-1}} \in \mathcal{V}_1^u$, when linearizing $L_K(M_1, \cdots, M_{2^{u-1}-1})M_{2^{u-1}}$, we need to assign each block in $\mathcal{A}_0^{u-1}$ and $\mathcal{A}_1^{u-1}$ as 0 or 1, respectively. So we have:

$$\begin{aligned}
&\mathcal{F}_0^u = \mathcal{F}_0^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_0^{u-1}\} \bigcup \mathcal{A}_0^{u-1},\\
&\mathcal{F}_1^u = \mathcal{F}_1^{u-1} \bigcup \{M_{i+2^{u-1}} | M_i \in \mathcal{F}_1^{u-1}\} \bigcup \mathcal{A}_1^{u-1}.
\end{aligned}$$

Therefore, the conclusion is true in case u. □

5.3 Linearizing KBRW with General Length Message

In Section 5.2, we fix the message length to $m = 2^u - 1$. Users usually select other-length messages. Therefore, this section considers how to linearize the KBRW polynomial with general-length messages.

For general m, we define six disjoint sets of message blocks as V_0^m, V_1^m, A_0^m, A_1^m, F_0^m and F_1^m, they are similar to the sets defined in Section 5.2, except that the former targets messages of general length, but the latter targets messages of length $m = 2^u - 1$. Next, we generalize the above conclusion to Theorem 3.

Theorem 3. *For the KBRW polynomial, assuming the message length is m, $t \leq m < 2t, t = 2^u, u \geq 2$. Let $V_0^1 = \{M_1\}$, $V_0^2 = \{M_1, M_2\}$, $V_0^3 = \{M_2, M_3\}$, $A_0^3 = \{M_1\}$ and initialize the remaining set to $\emptyset$. We can obtain the following recursions when $m \geq 4$:*

$$\begin{aligned}
A_0^m &= V_0^{t-1} \bigcup \{M_{i+t} | M_i \in A_0^{m-t}\},\\
A_1^m &= V_1^{t-1} \bigcup \{M_{i+t} | M_i \in A_1^{m-t}\},\\
F_0^m &= F_0^{t-1} \bigcup \{M_{i+t} | M_i \in F_0^{m-t}\} \bigcup A_0^{t-1},\\
F_1^m &= F_1^{t-1} \bigcup \{M_{i+t} | M_i \in F_1^{m-t}\} \bigcup A_1^{t-1}.
\end{aligned}$$

Furthermore, we can obtain the following recursions when $m \geq 7$:

$$V_0^m = \begin{cases} \{M_{i+t} | M_i \in V_0^{m-t}\} \bigcup \{M_t\}, & m < \frac{3t}{2} \\ \{M_{i+t} | M_i \in V_0^{m-t}\}, & otherwise \end{cases}$$

$$V_1^m = \begin{cases} \{M_{i+t} | M_i \in V_1^{m-t}\}, & m < \frac{3t}{2} \\ \{M_{i+t} | M_i \in V_1^{m-t}\} \bigcup \{M_t\}, & otherwise, \end{cases}$$

where $i \in \mathbb{Z}^+$. Then, after assigning the message blocks according to the above recursions, $KBRW_K(M) \oplus KBRW_K(M \oplus D)$ is a linear function of K.

Proof. Assuming the linearized $KBRW_K(M_1, \cdots, M_m)$ can be denoted as:

$$\begin{aligned}
L_K(M_1, \cdots, M_m) = {} & L_K(M_1, \cdots, M_{t-1})(K^t \oplus M_t) \\
& \oplus L_K(M_{t+1}, \cdots, M_m),\ t \leq m < 2t.
\end{aligned} \tag{7}$$

The coefficient of K^t in $L_K(M_1, \cdots, M_{t-1})(K^t \oplus M_t)$ is M_t, $t = 2^u, u \geq 2$. Therefore, the coefficient of K^t is $M_t \oplus 1$ or M_t in Equation (7) depends on whether the leading term of $L_K(M_{t+1}, \cdots, M_m)$ is K^t or not, and it's true only when $m - t \geq t/2$. Then we have $\{M_t\} \subseteq F_1^m$, otherwise $\{M_t\} \subseteq F_0^m$. Thus, when $m \geq 7$, we have:

$$V_0^m = \begin{cases} \{M_{i+t} | M_i \in V_0^{m-t}\} \bigcup \{M_t\}, & m < \frac{3t}{2} \\ \{M_{i+t} | M_i \in V_0^{m-t}\}, & otherwise \end{cases}$$

$$V_1^m = \begin{cases} \{M_{i+t} | M_i \in V_1^{m-t}\}, & m < \frac{3t}{2} \\ \{M_{i+t} | M_i \in V_1^{m-t}\} \bigcup \{M_t\}, & otherwise. \end{cases}$$

The rest of the proof is the same as Theorem 2 and will not repeat here. □

As we can see, Theorem 2 is a particular case of Theorem 3. Next, we analyze our attack's complexity by finding the minimal m satisfying $|V_0^m| + |V_1^m| = u$ for a fixed u ($u \geq 2$).

Theorem 4. *For the KBRW polynomial, $m = 2^u - 2$ is the minimal message length that satisfies $|V_0^m| + |V_1^m| = u, u \geq 2$. Then, after assigning the message blocks according to the above recursions, $KBRW_K(M) \oplus KBRW_K(M \oplus D)$ is a linear function of K.*

Proof. Define $V^m = V_0^m \bigcup V_1^m$, we have $|V^2| = |V^3| = 2$. Therefore, $m = 2^2 - 2 = 2$ is the minimal message length in case $u = 2$.

According to Theorem 3, we have $|V^m| = |V^{m-t}| + 1$. To obtain one more block that can be chosen arbitrarily and modified by unknowns, set $m - t = 2$, solving for the minimal m and t as 6 and 4, respectively. Therefore, $m = 2^3 - 2 = 6$ is the minimal message length in case $u = 3$.

Assume $m = 2^u - 2$ is the minimal message length in case u. To obtain one more block that we want, let $m' - t' = m$ and solve this equation:

$$m' - t' = 2^{u+1} - 2 - 2^u = 2^u - 2 = m,$$

the minimal m' and t' are $2^{u+1} - 2$ and 2^u, respectively. Therefore, $m' = 2^{u+1} - 2$ is the optimal solution in case $u + 1$. □

For a particular $m = 2^u - 2, u \geq 2$, define new sets $A^m = A_0^m \bigcup A_1^m$, and $F^m = F_0^m \bigcup F_1^m$, and we can calculate the size of V^m, A^m and F^m by the following expression:

$$V(m) = u, A(m) = \sum\nolimits_{i=2}^{u} i, F(m) = m - u - \sum\nolimits_{i=2}^{u} i.$$

5.4 Attacking the Instantiation of DCT

In this section, consider the universal hash function $\mathcal{H}$ of DCT defined in Section 5.1, then the adversary can query $\widetilde{\mathcal{E}}$ (defined in Algorithm 1) to get $C_L \| C_R$ and control the inputs of $\mathcal{H}$ by modifying C_R.

The core of our attack is to choose a particular message M and set some blocks of D as unknown, making $KBRW_K(M) \oplus KBRW_K(M \oplus D)$ a linear function of K. The generic steps of the attack are as follows.

1. Select a particular message M to query the encryption of DCT and obtain the corresponding ciphertext $C_L \| C_R$. Assume that $\text{ENCODE}_\tau(M) = (M_L, M_R)$ and M_R consists of $m = 2^{u+2} - 2$ blocks.

2. Using the linear modification technique, determine the value of each message block and modification block according to the six sets in Section 5.2, to make $KBRW_K(M_R) \oplus KBRW_K(M_R \oplus D)$ a linear function of K. Then calculate a set of solutions $\mathcal{D}$ satisfying
$$MSB_u(KBRW_K(M_R) \oplus KBRW_K(M_R \oplus D)) = 0^u,$$
where $u \leq \tau$.

3. Select a D from $\mathcal{D}$ and query the decryption of DCT with $C_L \| (C_R \oplus D)$. Repeat the step until passing the decryption verification. After about $2^{\tau - u}$ queries, we obtain a successful forgery.

Since $\mathcal{H}$ will first attach the length information at the end and then process it with two independent KBRW polynomial, we choose a special $(2^{u+2} - 2)$-block message, and after adding one 128-bit length information block at the end of the message, the input length of the KBRW polynomial is $2^{u+2} - 1$.

Theorem 2 is a partition method of message blocks. For a fixed u, the message blocks in $\mathcal{V}_0^{u+2}$ and $\mathcal{V}_1^{u+2}$ can be chosen arbitrarily, and the corresponding modification blocks are set to unknown. The message blocks in $\mathcal{A}_0^{u+2}$ and $\mathcal{A}_1^{u+2}$ can be chosen arbitrarily, and the corresponding modification blocks are set to zero. The message blocks in $\mathcal{F}_0^{u+2}$ and

$\mathcal{F}_1^{u+2}$ must be fixed as 0 or 1, respectively, and the corresponding modification blocks are set to zero. After assigning the message and modification blocks according to these rules, $KBRW_K(M) \oplus KBRW_K(M \oplus D)$ is a linear function of K.

Because the last block is the length information, which is also just right one element belonging to the set $\mathcal{V}_0^{u+2}$, we cannot change its value. Therefore, we can create a linear system $\mathcal{S}$ containing $(u+1) \times 128$ unknowns and $u \times 128$ linear equations to calculate a set of solutions $\mathcal{D}$ using the technique stated in Section 3. For each solution $D \in \mathcal{D}$, it satisfies $MSB_u(KBRW_K(M) \oplus KBRW_K(M \oplus D)) = 0^u, u \leq \tau$. The size of $\mathcal{D}$ is at least 2^{128}.

Next, consider the rest of the stretch. For each $D \in \mathcal{D}$, we can represent the linearized KBRW polynomial L_K by matrix over finite fields:

$$\overline{L_K} = \sum_i M_{D_i}(M_S)^i \overline{K} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \\ \alpha_{u+1,0} & \alpha_{u+1,1} & \cdots & \alpha_{u+1,127} \\ \vdots & \vdots & & \vdots \\ \alpha_{\tau,0} & \alpha_{\tau,1} & \cdots & \alpha_{\tau,127} \\ \vdots & \vdots & & \vdots \end{pmatrix} \cdot \begin{pmatrix} k_0 \\ \vdots \\ k_{u+1} \\ \vdots \\ k_\tau \\ \vdots \\ k_{127} \end{pmatrix}, \quad (8)$$

where $\alpha_{i,j}$ are all linear combinations of D, $0 \leq i, j \leq 127$.

To make the least significant $(\tau - u)$-bit of the stretch all zeros, we can list a series of new linear equations about $\overline{K}$ from the $(u+1)$-th row to the τ-th row of $\sum_i M_{D_i}(M_S)^i$ in Equation (8):

$$\begin{cases} \alpha_{u+1,0}k_0 + \alpha_{u+1,1}k_1 + \cdots + \alpha_{u+1,127}k_{127} = 0, \\ \alpha_{u+2,0}k_0 + \alpha_{u+2,1}k_1 + \cdots + \alpha_{u+2,127}k_{127} = 0, \\ \cdots \\ \alpha_{\tau,0}k_0 + \alpha_{\tau,1}k_1 + \cdots + \alpha_{\tau,127}k_{127} = 0, \end{cases} \quad (9)$$

where the size of the equations is $\tau - u < 128$. When the key is fixed, $\alpha_{i,0}k_0 + \alpha_{i,1}k_1 + \cdots + \alpha_{i,127}k_{127}$ in each rows of Equation (9) becomes linear combinations of $\alpha_{i,j}, u+1 \leq i \leq \tau, 0 \leq j \leq 127$, it is also linear combinations of D, and must be in the set $\mathcal{D}$.

Equation (9) are $\tau - u$ linear constraints on D, and we add these equations into the aforementioned linear system $\mathcal{S}$. As a result, the new linear system $\mathcal{S}'$ has non-zero solutions since the number of equations is $s \times 128 + (\tau - u)$, which is fewer than the number of unknowns $(s+1) \times 128$. The size of the new solution set $\mathcal{D}'$ is at least $2^{128-(\tau-u)}$. The above analysis indicates that our forgery attack is effective. For each $K \in \mathcal{K}$, the adversary can always find a suitable solution $D' \in \mathcal{D}'$ to satisfies $MSB_\tau(KBRW_K(M) \oplus KBRW_K(M \oplus D')) = 0^\tau$ with probability $\frac{2^{128-(\tau-u)}}{2^{128}} = 2^{-(\tau-u)}$.

For $\tau = 32$, our forgery attack on $\mathcal{H}$ is as follows. First, assume the adversary knows the ciphertext for a message (M_L, M_R), which M_R consisting of $m = 2^{18} - 2$ blocks (about 4 MB), corresponding to 17×128 unknowns. Second, by applying the technique stated above, the adversary can calculate a solution set $\mathcal{D}$ while keeping $MSB_{16}(\mathcal{H}_{K_1 \| K_2}(A, M_R)) = MSB_{16}(\mathcal{H}_{K_1 \| K_2}(A, M_R \oplus D))$ for each $D \in \mathcal{D}$. Third, the adversary continues to make 2^{16} decryption queries for the modified ciphertext $C_L \| (C_R \oplus D)$ by selecting a new difference D from $\mathcal{D}$ to match the remaining 16-bit of the stretch. Note that the adversary needs to perform about $q = \mathcal{O}(2^{16})$ queries to obtain a successful forgery. Therefore, our attack can break the integrity with fewer queries when τ is small.

Similarly, we can select the minimal message length according to Theorem 4 to attack $\mathcal{H}$. For a similar reason, the adversary can select a message of $m = 2^{u+2} - 3$ blocks M_R and successfully execute our attack with $q = 2^v$ decryption queries $(u + v = \tau)$.

5.5 Analysis of the Security Bounds with DCT Instantiation Scheme

Let $\widetilde{\Pi}$ denote the DCT scheme and let E denote the blockcipher used by the encryption scheme Π_1, $K \xleftarrow{\$} \mathcal{K}$. Theorem 3 in [FLLW16] describes the DAE security of $\widetilde{\Pi}$. Let $\mathbf{A}$ be an adversary against $\widetilde{\Pi}$ that asks at most q queries of at most m blocks in total and runs in time at most t. Then, the $\mathbf{Adv}^{\mathrm{DAE}}_{\widetilde{\Pi}}(\mathbf{A})$ is upper bounded by

$$\frac{3q^2\epsilon}{2} + \frac{2q^2}{2^{2n}} + \frac{3q\epsilon \cdot 2^{2n}}{2^{\tau}} + 3 \cdot \mathbf{Adv}^{\mathrm{SPRP}}_{E}(q, \mathcal{O}(t)) + 2 \cdot \mathbf{Adv}^{\mathrm{ivE}}_{\Pi_1}(q, m, \mathcal{O}(t)).$$

For small τ, the security of DCT depends mainly on the leading term $\frac{q\epsilon \cdot 2^{2n}}{2^{\tau}}$ mostly, which is related not only to q but also to ϵ. When DCT is implemented using the BRW polynomial with a bound of $\epsilon = \mathcal{O}(\frac{m^2}{2^{2n}})$ [FLLW16], the provable bounds of DCT are $\mathcal{O}(\frac{q^2m^2}{2^{2n}} + \frac{qm^2}{2^{\tau}})$, and we only focus on the term $\frac{qm^2}{2^{\tau}}$. Let $u + v = \tau$, when the adversary makes $q = \mathcal{O}(2^v)$ decryption queries of $m = \mathcal{O}(2^{u+2})$ blocks, $\frac{qm^2}{2^{\tau}} > 1$. We remark that the security bounds between the above attack and the proof are not contradictory.

As a result, the above analysis shows that DCT has a similar problem as GCM. The attack succeeds because the stretch part no longer holds the ϵ-AXU property well when τ is small. It is possible to recover the authentication key using the technique described in Section 5 of [Fer05], which would further compromise the security of DCT.

In the following section, we propose Robust DCT (RDCT), a variant of the DCT scheme in which the security bound is better than DCT, and the above attack is invalid for RDCT.

6 How to Fix It: Robust DCT

Our attack works because of the way DCT deals with the stretch. DCT encrypts M_L by XORing it with the result of $\mathcal{H}$, which does not prevent manipulation of the stretch. To make M_L unpredictable, in this section, we slightly modify DCT to avoid the problem in Section 5 by simply XORing the output of $\mathcal{H}$ to the output of the keyed permutation E. Therefore, encryption (resp. decryption) of it will lead to a random output. We call the new scheme Robust DCT (RDCT). We illustrate the encryption of RDCT in Figure 3 and more details in Algorithm 2.

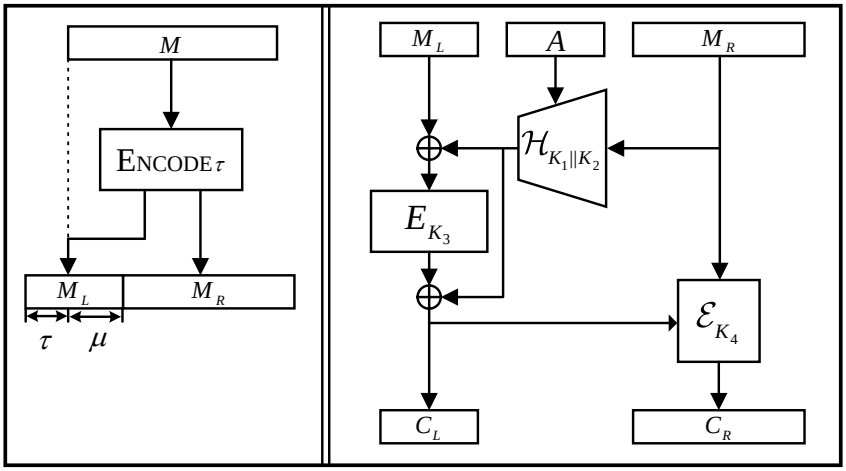

Figure 3: The ENCODE$_\tau$ process (**left**) and the encryption process of RDCT (**right**).

In fact, the modification forms a tweakable blockcipher $\widetilde{E}$ based on $\mathcal{H}_{K_1\|K_2}$ and E_{K_3}:

$$\widetilde{E}_{K_1,K_2,K_3}((A,M_R),M_L) \coloneqq E_{K_3}\left(M_L \oplus \mathcal{H}_{K_1\|K_2}(A,M_R)\right) \oplus \mathcal{H}_{K_1\|K_2}(A,M_R). \quad (10)$$

The idea is similar to the paper by Ashur et al. [ADL17], which introduces minor tweaks, such as an additional XOR, to obtain a tweakable blockcipher. Moreover, as a result, the core of RDCT is actually an instantiation of UIV construction [DK22]. $\widetilde{E}$ is an 2n-bit STPRP [LRW02]. The adversary does not repeat a query, so the input to $\widetilde{E}^{-1}$ rarely repeats. Therefore, decryption queries to RDCT will lead to a random left output. That is why the modification enhances the integrity of the scheme.

Algorithm 2 Encryption and decryption of RDCT

function $\widetilde{\mathcal{E}}_{K_1,K_2,K_3,K_4}(A,M)$
  $(M_L,M_R) \leftarrow \textsc{Encode}_\tau(M)$
  $C_L \leftarrow E_{K_3}\left(M_L \oplus \mathcal{H}_{K_1\|K_2}(A,M_R)\right) \oplus \mathcal{H}_{K_1\|K_2}(A,M_R)$
  $C_R \leftarrow \mathcal{E}_{K_4}(C_L,M_R)$
  return $(C_L\|C_R)$
end function

function $\widetilde{\mathcal{D}}_{K_1,K_2,K_3,K_4}(A,C)$
  $(C_L,C_R) \leftarrow C$
  $M_R \leftarrow \mathcal{D}_{K_4}(C_L,C_R)$
  $M_L \leftarrow E^{-1}_{K_3}\left(C_L \oplus \mathcal{H}_{K_1\|K_2}(A,M_R)\right) \oplus \mathcal{H}_{K_1\|K_2}(A,M_R)$
  return $\textsc{Decode}_\tau(M_L,M_R)$
end function

In the following, we show the security bounds of RDCT.

Lemma 1 (Confidentiality Advantage of RDCT)**.** *Let* $\widetilde{\Pi} = \mathrm{RDCT}_{\mathcal{H},E,\Pi_1,\Pi_2}$ *be as defined in* Algorithm 2*. Let* $\mathbf{A}$ *be a* detPriv *adversary on* $\widetilde{\Pi}$ *that submits at most* q_e *encryption queries of at most* m *blocks in total and runs in time at most* t*. Then*

$$\mathbf{Adv}^{\textsc{detPriv}}_{\widetilde{\Pi}}(\mathbf{A}) \le 3q_e^2\epsilon + \frac{q_e(q_e-1)}{2^{2n+1}} + \mathbf{Adv}^{\mathrm{PRP}}_{E}(q_e,\mathcal{O}(t+q_e)) + \mathbf{Adv}^{\mathrm{IvE}}_{\Pi_1}(q_e,m,\mathcal{O}(t)).$$

Proof. Firstly, we regard $\widetilde{E}$ as a tweakable random permutation $\widetilde{\pi} \xleftarrow{\$} \widetilde{\mathsf{Perm}}\left(\{0,1\}^{|A|+|M_R|},\{0,1\}^{|M_L|}\right)$ with TPRP advantage bounded by $\mathbf{Adv}^{\mathrm{PRP}}_{E}(q_e,\mathcal{O}(t+q_e)) + 3q_e^2\epsilon$ [LRW02]. We can also regard $\widetilde{\pi}$ with a (tweakable) random function $\rho \xleftarrow{\$} \widetilde{\mathsf{Func}}\left(\{0,1\}^{|A|+|M_R|},\{0,1\}^{|M_L|}\right)$ further with advantage bounded by $\frac{q_e(q_e-1)}{2^{2n+1}}$ by TPRP-TPRF switching lemma [HR04]. No adversary repeats an encryption query. So the input $((A,M_R),M_L)$ of the function ρ is different from all other $((A,M_R),M_L)$s deriving from previous encryption queries. Then, by the randomness of the function ρ, it always samples a fresh output C_L, a fresh random IV for encryption scheme $\mathcal{E}_{K_4}$. Then the probability between resulting $C_L\|\mathcal{E}_{K_4}(C_L,C_R)$ and $|C_L\|\mathcal{E}_{K_4}(C_L,C_R)|$-bit random string is bounded by ivE advantage. Summing all the advantages above, we get the detPriv advantage of $\widetilde{\Pi}$. □

Lemma 2 (Integrity Advantage of RDCT)**.** *Let* $\widetilde{\Pi} = \mathrm{RDCT}_{\mathcal{H},E,\Pi_1,\Pi_2}$ *be as defined in* Algorithm 2*. Let* $\mathbf{A}$ *be a* detAuth *adversary on* $\widetilde{\Pi}$ *that submits at most* q_e *encryption queries and* q_d *decryption queries of at most* m *blocks in total, and runs in time at most* t*. Then*

$$\mathbf{Adv}^{\textsc{detAuth}}_{\widetilde{\Pi}}(\mathbf{A}) \le 3q^2\epsilon + \frac{q(q-1)}{2^{2n+1}} + \frac{q_d}{2^\tau} + \mathbf{Adv}^{\mathrm{SPRP}}_{E}(q,\mathcal{O}(t+q)),$$

where $q = q_e + q_d$*.*

Proof. Firstly, we can regard the tweakable blockcipher $\widetilde{E}$ with a bidirectional tweakable random permutation $\widetilde{\pi}^{\pm} \xleftarrow{\$} \widetilde{\mathsf{Perm}}\left(\{0,1\}^{|A|+|M_R|}, \{0,1\}^{|M_L|}\right)$ with STPRP advantage bounded by $\mathbf{Adv}_{E}^{\mathrm{SPRP}}(q, \mathcal{O}(t+q)) + 3q^2\epsilon$ [LRW02]. We can also regard $\widetilde{\pi}^{\pm}$ with a bidirectional random function $\widetilde{\rho}^{\pm} \xleftarrow{\$} \widetilde{\mathsf{Func}}\left(\{0,1\}^{|A|+|M_R|}, \{0,1\}^{|M_L|}\right)$ further with advantage bounded by $\frac{q(q-1)}{2^{2n+1}}$ by STPRP-STPRF switching lemma [HR03]. Any adversary does not repeat a decryption query. So the input $((A, M_R), C_L)$ of the function $\widetilde{\rho}^{-1}$ is different from all other $((A, M_R), C_L)$s deriving from previous decryption queries. Then, by the randomness of the function $\widetilde{\rho}^{-1}$, it always samples a fresh output M_L. Then the probability of the left τ-bit of M_L of any q_d times decryption queries being 0^{τ} is bounded by $\frac{q_d}{2^{\tau}}$. Summing all the advantages above, we get the detAuth advantage of $\widetilde{\Pi}$. □

Theorem 5 (DAE Advantage of RDCT)**.** *Let* $\widetilde{\Pi} = \mathrm{RDCT}_{\mathcal{H},E,\Pi_1,\Pi_2}$ *be as defined in* Algorithm 2. *Let* $\mathbf{A}$ *be a* DAE *adversary on* $\widetilde{\Pi}$ *that asks at most* q_e *encryption queries and* q_d *decryption queries of at most* m *blocks in total and runs in time at most* t. *Then,* $\mathbf{Adv}_{\widetilde{\Pi}}^{\mathrm{DAE}}(\mathbf{A})$ *is upper bounded by*

$$\mathbf{Adv}_{\widetilde{\Pi}}^{\mathrm{DAE}}(\mathbf{A}) \leq 6q^2\epsilon + \frac{q^2}{2^{2n}} + \frac{q_d}{2^{\tau}} + 2\mathbf{Adv}_{E}^{\mathrm{SPRP}}(q, \mathcal{O}(t+q)) + \mathbf{Adv}_{\Pi_1}^{\mathrm{IVE}}(q_e, m, \mathcal{O}(t)),$$

where $q = q_e + q_d$.

The proof of Theorem 5 follows from Theorem 1 and the individual bounds for the detPriv and detAuth security in Lemma 1 and 2. For small τ, the security of RDCT depends mainly on the leading term $\frac{q_d}{2^{\tau}}$. At this point, it is independent of ϵ. However, the provable security of DCT is $\mathcal{O}(\frac{q^2\epsilon}{2} + \frac{q^2}{2^{2n}} + \frac{q\epsilon \cdot 2^{2n}}{2^{\tau}})$ [FLLW16]. For small τ, it depends mainly on the leading term $\frac{q\epsilon \cdot 2^{2n}}{2^{\tau}}$. Therefore, it is related to not only q but also ϵ. We will show how it affects when DCT and RDCT are implemented using the BRW polynomial with a bound of $\epsilon = \mathcal{O}(\frac{m^2}{2^{2n}})$ [FLLW16]. Now the provable securities of DCT and RDCT are $\mathcal{O}(\frac{q^2m^2}{2^{2n}} + \frac{qm^2}{2^{\tau}})$ and $\mathcal{O}(\frac{q^2m^2}{2^{2n}} + \frac{q_d}{2^{\tau}})$, respectively. For small τ, we focus only on $\frac{qm^2}{2^{\tau}}$ and $\frac{q_d}{2^{\tau}}$, respectively. Note that the security of DCT depends on the length of the query. However, the security of RDCT is not affected by it.

The attack on DCT requires only $\mathcal{O}(2^v)$ queries with $\mathcal{O}(2^u)$ blocks, where $v \leq \tau$ and $u = \tau - v$. Compared to DCT, to break the integrity of RDCT, the adversary has to make $\mathcal{O}(2^{\tau})$ decryption queries. Therefore, the minor change made by RDCT improves security.

7 Conclusions

We show that DCT suffers from a small stretch problem similar to that of GCM. Although the BRW polynomial is more complicated than the usual polynomial, the ideas of Ferguson's linear modification technique still work. To obtain a successful forgery of the DCT scheme, we must choose the length of the message m and the number of queries q flexibly with trade-off $mq = \mathcal{O}(2^{\tau})$ for small τ. Both GCM and DCT use the Wegman-Carter [WC81, Sho96] framework to authenticate. If we replace the UHF in GCM with KBRW or use KBRW in Wegman-Carter MACs, our method still works.

To solve the small stretch problem, we propose Robust DCT (RDCT). The adversary has to make $\mathcal{O}(2^{\tau})$ queries to obtain a successful forgery. Our fixing method is similar to that of [ADL17] to boost the security of GCM: XORing the result of the AXU function before and after the blockcipher. The core of RDCT actually follows the UIV construction, which is one of the rugged pseudorandom permutations suggested by Degabriele et al. [DK22].

Acknowledgments

The authors thank the anonymous reviewers for many helpful comments. The work of this paper was supported by the National Key Research and Development Program of China (Grant No. 2023YFB3105802, 2018YFA0704704, 2018YFA0704702), the National Natural Science Foundation of China (Grant No. 62032014, U2336207).

References

[ADL17] Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting Authenticated Encryption Robustness With Minimal Modifications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017*, volume 10403 of *LNCS*, pages 3–33. Springer, 2017. 16, 17

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A Concrete Security Treatment of Symmetric Encryption. In *FOCS '97*, pages 394–403. IEEE Computer Society, 1997. 3

[Ber07] Daniel J Bernstein. Polynomial evaluation and message authentication. *URL: https://cr.yp.to/antiforgery/pema-20071022.pdf. Citations in this document*, 2, 2007. 1

[CGS17] Debrup Chakraborty, Sebati Ghosh, and Palash Sarkar. A Fast Single-Key Two-Level Universal Hash Function. *FSE 2017*, 2017(1):106–128, 2017. 1

[DK22] Jean Paul Degabriele and Vukasin Karadzic. Overloading the Nonce: Rugged PRPs, Nonce-Set AEAD, and Order-Resilient Channels. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022*, volume 13510 of *LNCS*, pages 264–295. Springer, 2022. 1, 16, 17

[Dwo07] Morris Dworkin. *SP 800-38D. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. NIST, 2007. 0

[Fer02] Niels Ferguson. Collision attacks on OCB. *Comments submitted to NIST Modes of Operation Process*, pages 1–13, 2002. 1

[Fer05] Niels Ferguson. Authentication weaknesses in GCM. *Comments submitted to NIST Modes of Operation Process*, pages 1–10, 2005. 0, 2, 5, 15

[FLLW16] Christian Forler, Eik List, Stefan Lucks, and Jakob Wenzel. Efficient Beyond-Birthday-Bound-Secure Deterministic Authenticated Encryption with Minimal Stretch. In *ACISP 2016*, volume 9723 of *LNCS*, pages 317–332. Springer, 2016. 1, 6, 7, 8, 15, 17

[GL15] Shay Gueron and Yehuda Lindell. GCM-SIV: Full Nonce Misuse-Resistant Authenticated Encryption at Under One Cycle per Byte. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *CCS 2015*, pages 109–119. ACM, 2015. 1

[GLL17] Shay Gueron, Adam Langley, and Yehuda Lindell. AES-GCM-SIV: Specification and Analysis. *IACR Cryptol. ePrint Arch.*, page 168, 2017. 1

[GS19] Sebati Ghosh and Palash Sarkar. Evaluating Bernstein-Rabin-Winograd Polynomials. *DCC*, 87(2-3):527–546, 2019. 1

[HR03] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–499. Springer, 2003. 17

[HR04] Shai Halevi and Phillip Rogaway. A Parallelizable Enciphering Mode. In *CT-RSA 2004*, volume 2964 of *LNCS*, pages 292–304. Springer, 2004. 16

[II20] ISO and IEC. *ISO/IEC 19772:2020 Information security — Authenticated encryption*. ISO, 2020. 0

[IY09a] Tetsu Iwata and Kan Yasuda. BTM: A Single-Key, Inverse-Cipher-Free Mode for Deterministic Authenticated Encryption. In *SAC 2009*, volume 5867 of *LNCS*, pages 313–330. Springer, 2009. 1

[IY09b] Tetsu Iwata and Kan Yasuda. HBS: A Single-Key Mode of Operation for Deterministic Authenticated Encryption. In *FSE 2009*, volume 5665 of *LNCS*, pages 394–415. Springer, 2009. 1

[JNPS21] Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. The Deoxys AEAD Family. *JoC*, 34(3):31, 2021. 1

[LRW02] Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable Block Ciphers. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 31–46. Springer, 2002. 16, 17

[Min16] Kazuhiko Minematsu. Authenticated Encryption with Small Stretch (or, How to Accelerate AERO). In Joseph K. Liu and Ron Steinfeld, editors, *ACISP 2016*, volume 9723 of *LNCS*, pages 347–362. Springer, 2016. 1

[MV04a] David A. McGrew and John Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 343–355. Springer, 2004. 0, 1

[MV04b] David A McGrew and John Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation (Full Version). Cryptology ePrint Archive, Paper 2004/193, 2004. https://eprint.iacr.org/2004/193. 5

[NRS14] Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering Generic Composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, 2014. 5

[PS16] Thomas Peyrin and Yannick Seurin. Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. In *CRYPTO 2016*, volume 9814 of *LNCS*, pages 33–63. Springer, 2016. 1

[RBBK01] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption. In Michael K. Reiter and Pierangela Samarati, editors, *CCS 2001*, pages 196–205. ACM, 2001. 1

[Rog04] Phillip Rogaway. Nonce-Based Symmetric Encryption. In *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, 2004. 0

[RS06] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, 2006. 1, 6

[RW72] Michael O Rabin and Shmuel Winograd. Fast Evaluation of Polynomials by Rational Preparation. *Communications on Pure and Applied Mathematics*, 25(4):433–458, 1972. 1

[Sar09] Palash Sarkar. Efficient Tweakable Enciphering Schemes from (Block-Wise) Universal Hash Functions. *IEEE TIT*, 55(10):4749–4760, 2009. 1

[Sar11] Palash Sarkar. Tweakable Enciphering Schemes Using Only the Encryption Function of a Block Cipher. *IPL*, 111(19):945–955, 2011. 1

[Sho96] Victor Shoup. On Fast and Provably Secure Message Authentication Based on Universal Hashing. In Neal Koblitz, editor, *CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 313–328. Springer, 1996. 17

[WC81] Mark N. Wegman and Larry Carter. New Hash Functions and Their Use in Authentication and Set Equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981. 17