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

Yuchao Chen¹,², Tingting Guo³, Lei Hu⁴,⁵, Lina Shang⁶, Shuping Mao⁴,⁵ and Peng Wang⁷

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
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
⁶ Space Star Technology Co., Ltd.
sln-81088163.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 \( \tau \) is small, choosing a special \( m \)-block message, we can reduce the number of queries required by a successful forgery to \( O(2^{\tau/m}) \).

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 \( \tau \) is small and makes it resist the above attack.

Keywords: DCT · DeterministicAuthenticated 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 $\tau$-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 $\tau$. 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 $\tau$ 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 $2^{7-u}$ decryption queries. This attack can efficiently balance the space complexity $O(2^{u+2})$ and the time complexity $O(2^{7-u})$ 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 $\tau$ is small, our proof shows that a successful forgery attack requires $O(2^\tau)$ 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. $\tau$ 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.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Provable security</th>
<th>Query complexity</th>
<th>Query length</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCM</td>
<td>$O(2^{m^2} + 2^q)$</td>
<td>Encryption: 1, Decryption: 2$^\tau-u$, 2$^u+1$</td>
<td>[\text{Sect. 5.4}]</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>$O(2^{m^2} + 2^{q/2})$</td>
<td>Encryption: 1, Decryption: 2$^\tau-u$, 2$^u+2 - 3$</td>
<td>[\text{Sect. 6}]</td>
<td></td>
</tr>
<tr>
<td>RDCT</td>
<td>$O(q^{m^2} + q^{\tau-q})$</td>
<td>0</td>
<td>2$^\tau$, 1</td>
<td>[\text{Sect. 6}]</td>
</tr>
</tbody>
</table>

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, $\{0, 1\}^{\leq u}$ 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 $X$, $x \leftarrow X$ as the element $x$ chosen uniformly at random from the set $X$. We define $\text{MSB}_e$ as the most significant $\tau$ bits and $\text{LSB}_e$ as the least significant $\tau$ bits. We define $\text{Perm}(S)$ as the set of all permutations on $S$; $\text{Perm}(T, S)$ as the set of all tweakable permutations on $S$ with tweak space $T$; Let $\text{Func}(\mathcal{X}, \mathcal{Y})$ 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 $A$ runs in time at most $t$, makes at most $q$ queries of at most $m$ blocks in total, and can interact
Adv with respect to an IV-based encryption scheme


The real oracle defined as

Let \( A \)

Definition 4

Let \( \tilde{A} \)

\( \sim \)

a non-empty set. Let

Definition 3

Let \( \pi \)

\( \equiv \)

be a tweakable blockcipher

2.2 Definition of Universal Hash Functions

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

2.3 Security of (Tweakable) Blockciphers

Definition 2 ((Strong) PRP Advantage). Fix integers \( n, k \geq 1 \). Let \( E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n \) be a blockcipher and \( A(\tilde{A}') \) be an adversary with access to an oracle (two oracles). Let \( K \sim \tilde{K} \subseteq \{0,1\}^k \) and \( \pi \sim \tilde{\pi} \sim \text{Perm}((\{0,1\}^n)^*) \). Then, the PRP and SPRP advantages of \( A \) with respect to \( E \) are defined as

\[
\text{Adv}_E^{\text{PRP}}(A) := \left| \Pr[A^{E,K} \Rightarrow 1] - \Pr[A^{\tilde{K}} \Rightarrow 1] \right|
\]

and

\[
\text{Adv}_E^{\text{SPRP}}(A) := \left| \Pr[A^{E,K,E_{\tilde{K}}^{-1}} \Rightarrow 1] - \Pr[A^{\tilde{K},\tilde{E}_{\tilde{K}}^{-1}} \Rightarrow 1] \right|
\]

respectively.

Definition 3 ((Strong) Tweakable PRP Advantage). Fix integers \( n, k \geq 1 \). Let \( T \) denote a non-empty set. Let \( \tilde{E} : \{0,1\}^k \times T \times \{0,1\}^n \rightarrow \{0,1\}^n \) be a tweakable blockcipher and \( A \) be an adversary with access to an oracle (two oracles). Let \( K \sim \tilde{K} \subseteq \{0,1\}^k \) and \( \tilde{\pi} \sim \text{Perm}((T,\{0,1\}^n)^*) \). Then, the TPRP and STPRP advantages of \( A \) with respect to \( \tilde{E} \) are defined as

\[
\text{Adv}_E^{\text{TPRP}}(A) := \left| \Pr[A^{\tilde{E},K} \Rightarrow 1] - \Pr[A^{\tilde{\pi}} \Rightarrow 1] \right|
\]

and

\[
\text{Adv}_E^{\text{STPRP}}(A) := \left| \Pr[A^{E,K,E_{\tilde{K}}^{-1}} \Rightarrow 1] - \Pr[A^{\tilde{K},\tilde{E}_{\tilde{K}}^{-1}} \Rightarrow 1] \right|
\]

respectively.

2.4 Security of IV-Based Encryption Schemes

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

Definition 4 (IV Advantage). \( K \sim \tilde{K} \), let \( \Pi = (E, D) \) be an IV-based encryption scheme. Let \( A \) be an adversary with access to an oracle. Then, the IV advantage of \( A \) over \( \Pi \) is defined as

\[
\text{Adv}_\Pi^{\text{IV}}(A) := \left| \Pr[A^{E_K} \Rightarrow 1] - \Pr[A^{\$C} \Rightarrow 1] \right|
\]
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 $U(x) = u_0 x^{127} + \cdots + u_{126} x + u_{127}$ over $\mathbb{GF}(2)$. We denote $\mathbf{U} = \begin{pmatrix} u_0 \\ \vdots \\ u_{127} \end{pmatrix}$ as the column vector of the coefficients of $U(x)$ over $\mathbb{GF}(2)$ and $\mathbf{U}^T$ as the transpose of $\mathbf{U}$.

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 $128 \times 128$ matrix $M_C$ over $\mathbb{GF}(2)$ such that $C \cdot U = M_C \mathbf{U} = \begin{pmatrix} M_C^0 \\ M_C^1 \\ \vdots \\ M_C^{127} \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{127} \end{pmatrix}$ 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:

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

$C \cdot x$ 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 $U^2 = M_S \mathbf{U} = \begin{pmatrix} M_S^0 \\ M_S^1 \\ \vdots \\ M_S^{127} \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{127} \end{pmatrix}$ for all $U \in \mathbb{GF}(2^{128})$. We can calculate $M_S$ as follows:

$$
\begin{cases}
M_S^{127} = x^0, \\
M_S^{126} = x^1, \\
\vdots \\
M_S^0 = (x^{127})^2.
\end{cases}
$$

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 $C := C_1||C_2|\cdots||C_m$ 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

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

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 $\tau$-bit of the $T$, denoted by $\text{MSB}_\tau(T)$. The attacking goal is to find $D := D_1||D_2|\cdots||D_m$ such that

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

So we can obtain the collision

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

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 $D_{2^j}$ $(i \geq 0)$, such that

$$\text{MSB}_u(\sum_i D_{2^j} H^{2^j}) = 0^u,$$

where the parameter $u \leq \tau$. We focus on $D_{2^j}$’s for the reason that only $\sum_i D_{2^j} H^{2^j}$ 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 $D_1 = 0^u$. Since the first ciphertext block encodes the length information, we do not change it and let $D_1 = 0^u$.

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

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

where the matrix $M_{D_{2^j}}$ represents the operation corresponding to multiplication with $D_{2^j}$, and the elements in $M_{D_{2^j}}$ are all linear combinations of bits of $D_{2^j}$. $M_S$ 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 $D_{2^j}$ such that $u$ rows of $\sum_i M_{D_{2^j}}(M_S)^2$ are completely zero. When the number of $D_{2^j}$ 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 \( 17 \times 128 \) 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 \( \sum_i M_{p_j}(M_S)^{2i} \) equal to zero by creating \( 16 \times 128 \) 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 \( \tau \), 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 \( \tilde{\Pi} = (\tilde{E}, \tilde{D}) \) of deterministic algorithms \( \tilde{E} : K \times A \times M \to C \) and \( \tilde{D} : K \times A \times C \to M \cup \{ \bot \} \) with key space \( K \), associated-data space \( A \), and message/ciphertext space \( M, C \subseteq \{ 0, 1 \}^* \). For \( K \in K, A \in A, M \in M, \tilde{E}_K \) maps \( (A, M) \) to an output \( C \) such that \( |C| = |M| + \tau \) for a fixed stretch length \( \tau \). \( \tilde{D}_K(A, C) \) outputs the corresponding message \( M \) if \( C \) is valid or \( \bot \) otherwise, where \( \bot \) 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 \( \text{detPriv} \), \( \text{detAuth} \) as the confidentiality and integrity of the DAE scheme, respectively, as follows.

**Definition 5 (Confidentiality and Integrity Advantages).** Let \( \tilde{\Pi} = (\tilde{E}, \tilde{D}) \) be a DAE scheme and \( K \subsetneq K \). The adversary \( A \) has access to one or two oracles \( O_1 \) and \( O_2 \). \( A \) is not allowed to ask \( O_2 \) with the result of querying \( O_1 \). \( A \) does not repeat a query. Then, the \( \text{detPriv} \) and \( \text{detAuth} \) advantages of \( A \) with respect to \( \tilde{\Pi} \), are defined as

\[
\begin{align*}
\text{Adv}^\text{detPriv}_{\tilde{\Pi}}(A) &:= \left| \Pr[A \tilde{E}_K \Rightarrow 1] - \Pr[A \tilde{E}'_K \Rightarrow 1] \right|, \\
\text{Adv}^\text{detAuth}_{\tilde{\Pi}}(A) &:= \Pr[A \tilde{E}_K, \tilde{D}_K \text{ forges }],
\end{align*}
\]

where “forges” means that \( A \) asks \( \tilde{D}_K \) with \( (A, C) \) and returns anything other than \( \bot \) 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 \( \text{detPriv} \) and deterministic authenticity \( \text{detAuth} \). The relation between it and \( \text{detPriv} \) and \( \text{detAuth} \) is as follows.
Theorem 1 (DAE Advantage [FLLW16]). Let $\Pi = (\mathcal{E}, \mathcal{D})$ be a DAE scheme and $K \leftarrow \mathcal{K}$. Let $A$ be a DAE adversary on $\Pi$ with access to two oracles $O_1$ and $O_2$, $A$ is not allowed to ask $O_2$ with the result of querying $O_1$. Then, the DAE advantages of $A$ with respect to $\Pi$ is defined as

$$\text{Adv}^\text{DAE}_{\Pi}(A) := \Pr[A(\tilde{\mathcal{E}}_K, \tilde{\mathcal{D}}_K) = 1] - \Pr[A(\mathcal{E}^{\mathcal{K}}, \mathcal{D}) = 1],$$

where $A$ runs in time at most $t$ and asks $q_e$ queries to its left oracles, and asks $q_d$ queries to its right oracles, $A$ asks at most $m$ blocks in total. Then, there exists a DETPRIV adversary $A_1$ and a DETAUTH adversary $A_2$ both against $\Pi$, such that

$$\text{Adv}^\text{DAE}_{\Pi}(A) \leq \text{Adv}^\text{DETPRIV}_{\Pi}(A_1) + \text{Adv}^\text{DETAUTH}_{\Pi}(A_2),$$

where $A_1$ make at most $q_e$ queries with a maximum of $m$ blocks, $A_2$ make at most $q = q_e + q_d$ queries with a maximum of $m$ blocks and they both run in time $O(t)$.

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 Encode$_\tau$ (left) process and the encryption process of DCT (right).](image-url)

Fix the parameters $n, \tau \geq 1$ with $\tau \leq 2n$. Let $\mu = 2n - \tau$. Let $K_1, K_2, K_3$ and $K_4$ be non-empty key spaces and $K = K_1 \times K_2 \times K_3 \times K_4$. Let $A \subseteq \{0, 1\}^*$, $M \subseteq \{0, 1\}^{2n}$, $C \subseteq \{0, 1\}^{2n}$ denote the associated-data space, message space, and ciphertext space, respectively. Let $\mathcal{H} : K_1 \times K_2 \times A \times \{0, 1\}^* \rightarrow \{0, 1\}^{2n}$ be an AXU hash function with key space $K_1 \times K_2$. Let $E : K_3 \times \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$ be a blockcipher with key space $K_3$. Let $\Pi_1 = (\mathcal{E}, \mathcal{D})$ be an IV-based encryption scheme with key space $K_4$ and IV space $\mathcal{I}_V = \{0, 1\}^{2n}$. Let $\Pi_2 = (\text{Encode}_\tau, \text{Decode}_\tau)$ be an encode scheme with encoding function and decoding function

$$\text{Encode}_\tau : M \rightarrow \{0, 1\}^{2n} \times \{0, 1\}^{|M| - \mu},$$
$$\text{Decode}_\tau : \{0, 1\}^{2n} \times \{0, 1\}^{|M| - \mu} \rightarrow M \cup \{\bot\},$$

where $\text{Encode}_\tau$ is an injection, encodes $\tau$-bit redundancy into the input. The $\tau$-bit redundancy is fully contained in the left part of the output. For example, $\text{Encode}_\tau(M) = (M_L, M_R)$ where $M_L = 0^\tau|\text{MSB}_\mu(M)$ and $M_R = \text{LSB}_{|M| - \mu}(M)$. The decoding function
returns a unique \( M \in \mathcal{M} \) such that \( \text{Encode}_r(M) = (X,Y) \) for \((X,Y) \in \{0,1\}^{2n} \times \{0,1\}^{|M|-\mu}\) if such an \( M \) exists; otherwise, it returns \( \bot \). Then, the DCT scheme \( \text{DCT}_{H,E,\Pi_1,\Pi_2} = (\tilde{E},\tilde{D}) \) based on \( H,E,\Pi_1 \) and \( \Pi_2 \) is in Algorithm 1.

**Algorithm 1** Encryption and decryption of DCT

1: function \( \tilde{E}_{K_1,K_2,K_3,K_4}(A,M) \)
2: \( (M_L,M_R) \leftarrow \text{Encode}_r(M) \)
3: \( C_L \leftarrow E_{K_3}\left(M_L \oplus H_{K_1\|K_2}\left(A,M_R\right)\right) \)
4: \( C_R \leftarrow \mathcal{E}_{K_2}(C_L,M_R) \)
5: return \( (C_L\|C_R) \)
6: end function

7: function \( \tilde{D}_{K_1,K_2,K_3,K_4}(A,C) \)
8: \( (C_L,C_R) \leftarrow C \)
9: \( M_R \leftarrow \tilde{D}_{K_4}(C_L,C_R) \)
10: \( M_L \leftarrow E_{K_3}\left(C_L\right) \oplus H_{K_1\|K_2}\left(A,M_R\right) \)
11: return \( \text{Decode}_r(M_L,M_R) \)
12: end function

### 4.3 Instantiation of DCT

The instantiation of DCT employs a CTR-like encryption scheme as \( \mathcal{E}_{x_1}\left(x_2\right) = x_1\left(x_2\right) \) where \( x_1 \) generates an \( |M| \)-bit string. Due to the IV length of the \( \Pi_1 \) being 2\( n \)-bit, DCT employs a 2\( n \)-bit permutation as \( E \). \( \Pi_2 \) encodes the \( \tau \)-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 \( 2^\mu - 1 \) 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:

\[
\mathcal{H}_{K_1\|K_2}(X_1,X_2) = KBRW_{K_1}(M)||KBRW_{K_3}(M),
\]

where \( K_1,K_2 \in \{0,1\}^n \) are independent keys, \( M = \text{PAD}_n(X_1)||\text{PAD}_n(X_2)||L \), \( L \) is an \( n \)-bit block containing the length information of the inputs, \( \text{PAD}_n \) 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 \( M = (M_1,\ldots,M_m) \), \( M_i \in \{0,1\}^n \), the polynomial \( KBRW_K(M) \) is defined as follows:

\[
\begin{align*}
KBRW_K(\varepsilon) &= 0^n; \\
KBRW_K(M_1) &= M_1K; \\
KBRW_K(M_1,M_2) &= M_1K^2 \oplus M_2K; \\
KBRW_K(M_1,M_2,M_3) &= K^4 \oplus M_1K^3 \oplus M_2K^2 \oplus (M_1M_2 \oplus M_3)K; \\
KBRW_K(M_1,\ldots,M_m) &= KBRW_K(M_1,\ldots,M_{t-1})(K^t \oplus M_t) \oplus KBRW_K(M_{t+1},\ldots,M_m) \text{ if } t \leq m < 2t \text{ for } t = 2^i, i \geq 2.
\end{align*}
\]
Let $\varepsilon$ represent the empty string. All operations in the KBRW polynomial are performed over $GF(2^n)$, 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:

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

We can query the decryption of DCT with $C_L \parallel C_R'$ 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:

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

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 $m = 2^n - 1$ for the sake of convenience. When $u = 2$, $M = (M_1, M_2, M_3)$ and

$$KBRW_K(M) = K^4 \oplus M_1 K^3 \oplus M_2 K^2 \oplus (M_1 M_2 \oplus M_3) K. \quad (1)$$

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 $KBRW_K(M \oplus KBRW_K(M \oplus D) = D_2 K^2 \oplus (M_1 D_2 \oplus D_3) K$ 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

$$KBRW_K(M) = K^6 \oplus M_1 K^5 \oplus M_2 K^4 \oplus (M_1 M_2 \oplus M_3) K^3 \oplus (M_2 M_4 \oplus M_6) K^2$$

$$\oplus (M_1 M_2 M_4 \oplus M_5 M_4 \oplus M_5 M_6 \oplus M_7) K. \quad (2)$$

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 $M_1 M_4 \oplus M_5$. 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: $V_0^u$, $V_1^u$, $A_0^u$, $A_1^u$, $F_0^u$ and $F_1^u$ according to the subscript of the message blocks, which specifies how these blocks should be valued and modified.

- $V_0^u$ and $V_1^u$ 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 $V_1^u$, otherwise in $V_0^u$;
- $A_0^u$ and $A_1^u$ are sets of blocks that can be chosen arbitrarily but not modified by unknowns. The difference between $A_0^u$ and $A_1^u$ is the same as the above;
- $F_0^u$ and $F_1^u$ are sets of blocks that are fixed as 0 and 1 respectively and not modified by unknowns. The difference between $F_0^u$ and $F_1^u$ is the same as above.
For example, when \( u = 2 \), consider Equation (1). Let \( F_1^2 = F_2^2 = V_1^2 = A_2^2 = \emptyset \), \( V_0^2 = \{ M_2, M_3 \} \) and \( A_0^2 = \{ M_1 \} \), which means that the value of \( M_1 \) should remain invariant 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 \( V_0^5 \) by the case with \( u = 2 \), and they originate from the subscript plus 4 of elements in \( V_0^2 \), as shown in Figure 2, so that \( V_0^5 = \{ M_{i+2} : |M_i \in V_0^2 \} = \{ M_6, M_7 \} \). Thus, for general \( u > 2 \), we have \( V_0^u = \{ M_{i+2^{u-1}} : |M_i \in V_0^{u-1} \} \).

Note that the term \( (M_4 \oplus 1)K^4 \) in Equation (2), we choose \( V_1^3 = \{ M_4 \} \). 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 \( F_0^3 = A_0^3 = \{ M_1 \} \) \( (M_4 = D_4 = 0) \) to make sure \( M_4 \) as an unknowns. Thus, for general \( u > 2 \), we have \( F_0^u = F_0^{u-1} \cup \{M_{i+2^{u-1}} : |M_i \in F_0^{u-1} \} \cup A_0^{u-1} \). So that \( KBRW_K(M) \oplus KBRW_K(M \oplus D) = D_4K^4 \oplus (M_2D_4 \oplus D_6)K^2 \oplus (M_1D_4 \oplus M_5D_6 \oplus D_7)K \) is a linear function of \( K \).

\[
\begin{align*}
    m = 3 \ (u = 2) : & \quad M_2 \quad M_3 \\
    m = 7 \ (u = 3) : & \quad M_4 \quad M_6 \quad M_7 \\
    m = 15 \ (u = 4) : & \quad M_8 \quad M_{12} \quad M_{14} \quad M_{15} \\
    m = 31 \ (u = 5) : & \quad M_{16} \quad M_{24} \quad M_{28} \quad M_{30} \quad M_{31} \\
    m = 63 \ (u = 6) : & \quad M_{32} \quad M_{48} \quad M_{56} \quad M_{60} \quad M_{62} \quad M_{63}
\end{align*}
\]

**Figure 2:** Calculating the set \( V_0^u \) and \( V_1^u \) recursively. The different colored arrow indicates that the subscript of the elements plus with different \( t \). E.g., the elements in \( V_0^u \) originate from the subscript plus \( t = 2^{u-1} \) of elements in \( V_0^{u-1} \). The elements in the orange dashed box belong to the set \( V_0^u \), and the rest belong to the set \( V_1^u \).

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 \( V_0^u = \{ M_2, M_3 \} \), \( A_0^u = \{ M_1 \} \), and initialize the remaining set to \( \emptyset \). We can obtain the following recursions:

\[
\begin{align*}
    V_0^u &= \{ M_{i+2^{u-1}} : |M_i \in V_0^{u-1} \}, \\
    V_1^u &= \{ M_{2^{u-1}} \} \cup \{ M_{i+2^{u-1}} : |M_i \in V_0^{u-1} \}, \\
    A_0^u &= V_0^{u-1} \cup \{ M_{i+2^{u-1}} : |M_i \in A_0^{u-1} \}, \\
    A_1^u &= V_1^{u-1} \cup \{ M_{i+2^{u-1}} : |M_i \in A_1^{u-1} \}, \\
    F_0^u &= F_0^{u-1} \cup \{ M_{i+2^{u-1}} : |M_i \in F_0^{u-1} \} \cup A_0^{u-1}, \\
    F_1^u &= F_1^{u-1} \cup \{ M_{i+2^{u-1}} : |M_i \in F_1^{u-1} \} \cup A_1^{u-1},
\end{align*}
\]

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:

\[
\begin{align*}
V_0^3 &= \{M_{i+2^2}|M_i \in V_0^2\} = \{M_6, M_7\}, \\
V_1^3 &= \{M_{2^2}\} \cup \{M_{i+2^2}|M_i \in V_1^2\} = \{M_4\}, \\
A_0^3 &= V_0^2 \cup \{M_{i+2^2}|M_i \in A_0^2\} = \{M_2, M_3, M_5\}, \\
A_1^3 &= V_1^2 \cup \{M_{i+2^2}|M_i \in A_1^2\} = \emptyset, \\
F_0^3 &= F_0^2 \cup \{M_{i+2^2}|M_i \in F_0^2\} \cup A_0^2 = \{M_1\}, \\
F_1^3 &= F_1^2 \cup \{M_{i+2^2}|M_i \in F_1^2\} \cup A_1^2 = \emptyset,
\end{align*}
\]

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 \( u \):

\[
KBRW_K(M_1, \cdots, M_m) = KBRW_K(M_1, \cdots, M_{2u-3})(K^{2u-1} \oplus M_{2u-1}) \oplus KBRW_K(M_{2u-1+1}, \cdots, M_m).
\]

Since the subscript of coefficients in \( KBRW_K(M_{2u-1+1}, \cdots, M_m) \) is larger than \( KBRW_K(M_1, \cdots, M_{2u-1}) \) by \( 2^u-1 \), therefore, as shown in Figure 2, we have \( V_0^u = \{M_{i+2^u-1}|M_i \in V_0^{u-1}\} \) and \( \{M_{i+2^u-1}|M_i \in V_1^{u-1}\} \subseteq V_1^u \).

To linearize the above two functions, we need to assign 0 and 1 to each block in \( F_0^{u-1} \cup \{M_{i+2^u-1}|M_i \in F_0^{u-1}\} \) and \( F_1^{u-1} \cup \{M_{i+2^u-1}|M_i \in F_1^{u-1}\} \), respectively. Thus, we have:

\[
\begin{align*}
F_0^{u-1} \cup \{M_{i+2^u-1}|M_i \in F_0^{u-1}\} &\subseteq F_0^u, \\
F_1^{u-1} \cup \{M_{i+2^u-1}|M_i \in F_1^{u-1}\} &\subseteq F_1^u.
\end{align*}
\]

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

\[
L_K(M_1, \cdots, M_m) = L_K(M_1, \cdots, M_{2u-3})(K^{2u-1} \oplus M_{2u-1}) \oplus L_K(M_{2u-1+1}, \cdots, M_m),
\]

the leading term of \( L_K(M_1, \cdots, M_{2u-3}) \) is \( M_{2u-3} \) raised to the degree of each term by \( 2^u-1 \), so none of the degrees of each term in this function are powers of two except the leading term \( K^{2^u} \). To linearize it, each block in \( V_0^u \cup \{M_{i+2^u-1}|M_i \in A_0^{u-1}\} \) and \( V_1^u \cup \{M_{i+2^u-1}|M_i \in A_1^{u-1}\} \) should keep unchanged when forgery. Thus, we have:

\[
A_0^u = V_0^{u-1} \cup \{M_{i+2^u-1}|M_i \in A_0^{u-1}\}, \\
A_1^u = V_1^{u-1} \cup \{M_{i+2^u-1}|M_i \in A_1^{u-1}\}.
\]

Because \( M_{2u-1} \in V_1^u \), when linearizing \( L_K(M_1, \cdots, M_{2u-3}) \), we need to assign each block in \( A_0^{u-1} \) and \( A_1^{u-1} \) as 0 or 1, respectively. So we have:

\[
\begin{align*}
F_0^u &= F_0^{u-1} \cup \{M_{i+2^u-1}|M_i \in F_0^{u-1}\} \cup A_0^{u-1}, \\
F_1^u &= F_1^{u-1} \cup \{M_{i+2^u-1}|M_i \in F_1^{u-1}\} \cup A_1^{u-1}.
\end{align*}
\]
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 \( V_0^m, V_1^m, A^m, F_0^m \) 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 \( V_0^1 = \{M_1\}, V_0^2 = \{M_1, M_2\}, V_0^3 = \{M_2, M_3\}, A_3 = \{M_1\} \) and initialize the remaining set to \( \emptyset \). We can obtain the following recursions when \( m \geq 4 \):

\[
A_0^m = V_0^{m-1} \cup \{M_{i+t} | M_i \in A_0^{m-1}\},
\]

\[
A_1^m = V_1^{m-1} \cup \{M_{i+t} | M_i \in A_1^{m-1}\},
\]

\[
F_0^m = F_0^{m-1} \cup \{M_{i+t} | M_i \in F_0^{m-1}\} \cup A_0^{m-1},
\]

\[
F_1^m = F_1^{m-1} \cup \{M_{i+t} | M_i \in F_1^{m-1}\} \cup A_1^{m-1}.
\]

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

\[
V_0^m = \begin{cases} 
\{M_{i+t} | M_i \in V_0^{m-1}\} \cup \{M_i\}, & m < \frac{3t}{2} \\
\{M_{i+t} | M_i \in V_0^{m-1}\}, & \text{otherwise} 
\end{cases}
\]

\[
V_1^m = \begin{cases} 
\{M_{i+t} | M_i \in V_1^{m-1}\}, & m < \frac{3t}{2} \\
\{M_{i+t} | M_i \in V_1^{m-1}\} \cup \{M_i\}, & \text{otherwise} 
\end{cases}
\]

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:

\[
L_K(M_1, \cdots, M_m) = L_K(M_1, \cdots, M_{t-1})(K^t \oplus M_t) \oplus L_K(M_{t+1}, \cdots, M_m), \quad t \leq m < 2t. \tag{7}
\]

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 \( \{M_i\} \subseteq F_1^m \), otherwise \( \{M_i\} \subseteq F_0^m \). Thus, when \( m \geq 7 \), we have:

\[
V_0^m = \begin{cases} 
\{M_{i+t} | M_i \in V_0^{m-1}\} \cup \{M_i\}, & m < \frac{3t}{2} \\
\{M_{i+t} | M_i \in V_0^{m-1}\}, & \text{otherwise} 
\end{cases}
\]

\[
V_1^m = \begin{cases} 
\{M_{i+t} | M_i \in V_1^{m-1}\}, & m < \frac{3t}{2} \\
\{M_{i+t} | M_i \in V_1^{m-1}\} \cup \{M_i\}, & \text{otherwise} 
\end{cases}
\]

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, \( \text{KBRW}_K(M) \oplus \text{KBRW}_K(M \oplus D) \) is a linear function of \( K \).

Proof. Define \( V^m = V_0^m \cup V_1^m \), 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 \( |V^m| = |V^{m-1}| + 1 \). 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 \( A^m = A_0^m \cup A_1^m \) and \( F^m = F_0^m \cup F_1^m \), and we can calculate the size of \( V^m, A^m \) and \( F^m \) by the following expression:

\[
V(m) = u, A(m) = \sum_{i=2}^{u} i, F(m) = m - u - \sum_{i=2}^{u} i.
\]

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 \( \mathcal{E} \) (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 \( \text{KBRW}_K(M) \oplus \text{KBRW}_K(M \oplus D) \) 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 \( \text{KBRW}_K(M_R) \oplus \text{KBRW}_K(M_R \oplus D) \) a linear function of \( K \). Then calculate a set of solutions \( D \) satisfying

\[
\text{MSB}_u(\text{KBRW}_K(M_R) \oplus \text{KBRW}_K(M_R \oplus D)) = 0^u,
\]

where \( u \leq \tau \).

3. Select a \( D \) from \( D \) 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 \( V_0^{u+2} \) and \( V_1^{u+2} \) can be chosen arbitrarily, and the corresponding modification blocks are set to unknown. The message blocks in \( A_0^{u+2} \) and \( A_1^{u+2} \) can be chosen arbitrarily, and the corresponding modification blocks are set to zero. The message blocks in \( F_0^{u+2} \) and
\( F_{i}^{u+2} \) 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 \( V_{\alpha}^{u+2} \), we cannot change its value. Therefore, we can create a linear system \( S \) containing \((u+1) \times 128 \) unknowns and \( u \times 128 \) linear equations to calculate a set of solutions \( D \) using the technique stated in Section 3. For each solution \( D \in D \), it satisfies \( MSB_{\alpha}(KBRW_{K}(M) \oplus KBRW_{K}(M \oplus D)) = 0^{u}, u \leq \tau \). The size of \( D \) is at least \( 2^{128} \).

Next, consider the rest of the stretch. For each \( D \in D \), we can represent the linearized KBRW polynomial \( L_{K} \) by matrix over finite fields:

\[
\vec{L}_{K} = \sum_{i} M_{D_{i}}(M_{S}) \cdot \vec{K} = \begin{pmatrix}
0 & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 0
\end{pmatrix}
\begin{pmatrix}
k_{0} \\
k_{u+1} \\
k_{\tau}
\end{pmatrix}, \quad (8)
\]

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 \( \vec{K} \) from the \((u+1)\)-th row to the \( \tau \)-th row of \( \sum_{i} M_{D_{i}}(M_{S}) \) in Equation (8):

\[
\begin{array}{l}
\alpha_{n+1.0}k_{0} + \alpha_{n+1.1}k_{1} + \cdots + \alpha_{n+1.127}k_{127} = 0, \\
\alpha_{n+2.0}k_{0} + \alpha_{n+2.1}k_{1} + \cdots + \alpha_{n+2.127}k_{127} = 0, \\
\vdots \\
\alpha_{\tau.0}k_{0} + \alpha_{\tau.1}k_{1} + \cdots + \alpha_{\tau.127}k_{127} = 0,
\end{array} \quad (9)
\]

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

Equation (9) is \( \tau - u \) linear constraints on \( D \), and we add these equations into the aforementioned linear system \( S \). As a result, the new linear system \( S' \) 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 \( D' \) 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 \( D' \in D' \) 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 \times 128 unknowns. Second, by applying the technique stated above, the adversary can calculate a solution set \( D \) while keeping \( MSB_{16}(H_{K}||K_{2}(A, M_{R})) = MSB_{16}(H_{K}||K_{2}(A, M_{R} \oplus D)) \) for each \( D \in D \). 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 \( D \) to match the remaining 16-bit of the stretch. Note that the adversary needs to perform about \( q = O(2^{16}) \) queries to obtain a successful forgery. Therefore, our attack can break the integrity with fewer queries when \( \tau \) 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 \( q = 2^{u} \) decryption queries \( u + v = \tau \).
5.5 Analysis of the Security Bounds with DCT Instantiation Scheme

Let \( \tilde{\Pi} \) denote the DCT scheme and let \( E \) denote the blockcipher used by the encryption scheme \( \Pi_1 \). K \( \leftarrow \mathcal{K} \). Theorem 3 in [FLLW16] describes the DAE security of \( \tilde{\Pi} \). Let \( A \) be an adversary against \( \tilde{\Pi} \) that asks at most \( q \) queries of at most \( m \) blocks in total and runs in time at most \( t \). Then, the \( \text{Adv}_{\tilde{\Pi}}^{\text{DAE}}(A) \) is upper bounded by

\[
\frac{3q^2\epsilon}{2} + \frac{2q^2}{2^2n} + \frac{3q \cdot 2^{2n}}{2^2} + 3 \cdot \text{Adv}_{E}^{\text{SPRP}}(q, O(t)) + 2 \cdot \text{Adv}_{\Pi_1}^{\text{iv}}(q, m, O(t)).
\]

For small \( \tau \), the security of DCT depends mainly on the leading term \( \frac{q^2 \cdot 2^{2n}}{2} \) mostly, which is related not only to \( q \) but also to \( \epsilon \). When DCT is implemented using the BRW polynomial with a bound of \( \epsilon = O\left(\frac{m^2}{2^{n}}\right) \) [FLLW16], the provable bounds of DCT are \( O\left(\frac{q^2 m^2}{2^{n}} + \frac{q m^2}{2^{n}}\right) \), and we only focus on the term \( \frac{q m^2}{2^{n}} \). Let \( u + v = \tau \), when the adversary makes \( q = O(2^u) \) decryption queries of \( m = O(2^{u+2}) \) blocks, \( \frac{q m^2}{2^{n}} > 1 \). 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 \( \epsilon \)-AXU property well when \( \tau \) 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 \( H \), 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 \( H \) 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 process (left) and the encryption process of RDCT (right).](image)
In fact, the modification forms a tweakable blockcipher \( \tilde{E} \) based on \( \mathcal{H}_{K_1 || K_2} \) and \( E_{K_3} \):

\[
\tilde{E}_{K_1, K_2, K_3}(A, M_R, M_L) := E_{K_3} \left( M_L \oplus \mathcal{H}_{K_1 || K_2} (A, M_R) \right) \oplus \mathcal{H}_{K_1 || K_2} (A, M_R).
\]

(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]. \( \tilde{E} \) is an 2n-bit STPRP [LRW02]. The adversary does not repeat a query, so the input to \( \tilde{E}^{-1} \) 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

1. function \( \tilde{E}_{K_1, K_2, K_3, K_4}(A, M) \)
2. \( (M_L, M_R) \leftarrow \text{ENCODE}_r(M) \)
3. \( C_L \leftarrow E_{K_3} \left( M_L \oplus \mathcal{H}_{K_1 || K_2} (A, M_R) \right) \oplus \mathcal{H}_{K_1 || K_2} (A, M_R) \)
4. \( C_R \leftarrow \tilde{E}_{K_4}(C_L, M_R) \)
5. return \( (C_L, C_R) \)
6. end function

7. function \( \tilde{D}_{K_1, K_2, K_3, K_4}(A, C) \)
8. \( (C_L, C_R) \leftarrow C \)
9. \( M_R \leftarrow \tilde{D}_{K_3}(C_L, C_R) \)
10. \( M_L \leftarrow E^{-1}_{K_3}(C_L \oplus \mathcal{H}_{K_1 || K_2} (A, M_R) \oplus \mathcal{H}_{K_1 || K_2} (A, M_R) \)
11. return \( \text{DECODE}_r(C_L, M_R) \)
12. end function

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

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

\[
\text{Adv}_{\tilde{\Pi}}^{\text{DETPriv}}(A) \leq 3q_e \epsilon + \frac{q_e(q_e - 1)}{2^{2n+1}} + \text{Adv}_{E}^{\text{PRP}}(q_e, O(t + q_e)) + \text{Adv}_{\Pi_1}^{\text{ivE}}(q_e, m, O(t)).
\]

**Proof.** Firstly, we regard \( \tilde{E} \) as a tweakable random permutation \( \tilde{\pi} \leftarrow \tilde{\text{FR}} \left( \{0, 1\}^{2^n}, \{0, 1\}^{2^n} \right) \) with TPRP advantage bounded by \( \text{Adv}_{E}^{\text{PRP}}(q_e, O(t + q_e)) + 3q_e^2 \epsilon \) [LRW02]. We can also regard \( \tilde{\pi} \) with a (tweakable) random function \( \rho \leftarrow \tilde{\text{Func}} \left( \{0, 1\}^{2^n}, \{0, 1\}^{2^n} \right) \) further with advantage bounded by \( \frac{q_e(q_e - 1)}{2^{2n+1}} \) by TPRP-TPRP switching lemma [HR04]. No adversary repeats an encryption query. So the input \((A, M_R, M_L)\) of the function \( \rho \) is different from all other \((A, M_R, M_L)\) deriving from previous encryption queries. Then, by the randomness of the function \( \rho \), it always samples a fresh output \( C_L \), a fresh random IV for encryption scheme \( E_{K_3} \). Then the probability to existing \( C_L \| \mathcal{E}_{K_4}(C_L, C_R) \) and \( |C_L| \mathcal{E}_{K_1}(C_L, C_R) \)-bit random string is bounded by \( \text{ivE} \) advantage. Summing all the advantages above, we get the \( \text{DETPriv} \) advantage of \( \tilde{\Pi} \). \( \square \)

**Lemma 2** (Integrity Advantage of RDCT). Let \( \tilde{\Pi} = \text{RDCT}_{\mathcal{H}, E, \Pi_1, \Pi_2} \) be as defined in Algorithm 2. Let \( A \) be a \( \text{DETAuth} \) adversary on \( \tilde{\Pi} \) 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

\[
\text{Adv}_{\tilde{\Pi}}^{\text{DETAuth}}(A) \leq 3q^2 \epsilon + \frac{q(q - 1)}{2^{2n+1}} + \frac{qd}{2^2} + \text{Adv}_{E}^{\text{PRP}}(q, O(t + q)),
\]

where \( q = q_e + q_d \).
Proof. Firstly, we can regard the tweakable blockcipher $\tilde{E}$ with a bidirectional tweakable random permutation $\tilde{\pi}^R \overset{\$}{\leftarrow} \text{Perm} \left( \{0, 1\}^{|A|+|M_R|}, \{0, 1\}^{|M_L|} \right)$ with STPRP advantage bounded by $\text{Adv}^{\text{SPRP}}_E(q, \mathcal{O}(t+q)) + 3q^2 \epsilon$ [LRW02]. We can also regard $\tilde{\pi}^L$ with a bidirectional random function $\tilde{\pi}^L \overset{\$}{\leftarrow} \text{Func} \left( \{0, 1\}^{|A|+|M_R|}, \{0, 1\}^{|M_L|} \right)$ further with advantage bounded by $\frac{q(q-1)}{2}$ 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 $\tilde{\rho}^{-1}$ is different from all other $((A, M_R), C_L)$ is deriving from previous decryption queries. Then, by the randomness of the function $\tilde{\rho}^{-1}$, it always samples a fresh output $M_L$. Then the probability of the left $\tau$-bit of $M_L$ of any $q_d$ times decryption queries being $0^\tau$ is bounded by $\frac{q_d}{2^\tau}$. Summing all the advantages above, we get the DETAUTH advantage of $\tilde{\Pi}$.

\[ \text{Adv}^{\text{DAE}}_{\tilde{\Pi}}(A) \leq 6q^2 \epsilon + q^2 \frac{2^\tau}{2^n} + \frac{q_d}{2^\tau} + 2\text{Adv}^{\text{SPRP}}_{\tilde{E}}(q, \mathcal{O}(t+q)) + \text{Adv}^{\text{SPRP}}_{\tilde{E}}(q_e, m, \mathcal{O}(t)), \]

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 $\tau$, the security of RDCT depends mainly on the leading term $\frac{q_d}{2^\tau}$. At this point, it is independent of $\epsilon$. However, the provable security of DCT is $O\left(\frac{q^2}{2^n} + \frac{q^2}{2^n} + \frac{q^2}{2^{2n}}\right)$ [FLLW16]. For small $\tau$, it depends mainly on the leading term $\frac{q_d}{2^\tau}$. Therefore, it is related to not only $q$ but also $\epsilon$. We will show how it affects when DCT and RDCT are implemented using the BRW polynomial with a bound of $\epsilon = O\left(\frac{q^2}{2^\tau}\right)$ [FLLW16]. Now the provable securities of DCT and RDCT are $O\left(\frac{q^2}{2^\tau} + \frac{q^2}{2^n}\right)$ and $O\left(\frac{q^2}{2^\tau} + \frac{q^2}{2^n}\right)$, respectively. For small $\tau$, we focus only on $\frac{q^2}{2^n}$ and $\frac{q^2}{2^\tau}$, 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 $O(2^u)$ queries with $O(2^u)$ blocks, where $v \leq \tau$ and $u = \tau - v$. Compared to DCT, to break the integrity of RDCT, the adversary has to make $O(2^\tau)$ 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 $mq = O(2^\tau)$ for small $\tau$. 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 $O(2^\tau)$ 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


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


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


