Tweakable Enciphering Schemes Using Only the Encryption Function of a Block Cipher

Palash Sarkar

Applied Statistics Unit
Indian Statistical Institute
203, B.T. Road, Kolkata
India 700108.
email: palash@isical.ac.in

Abstract. A new construction of block cipher based tweakable enciphering schemes (TES) is described. The major improvement over existing TES is that the construction uses only the encryption function of the underlying block cipher. Consequently, this leads to substantial savings in the size of hardware implementation of TES applications such as disk encryption. This improvement is achieved without loss in efficiency of encryption and decryption compared to the best previously known schemes.
Keywords: modes of operations, tweakable encryption, strong pseudorandom permutation, disk encryption.

1 Introduction

Modes of operations of a block cipher are important from both a practical and a theoretical point of view. From a practical viewpoint, for a block cipher to be useful for a particular application, one needs to deploy it in an appropriate mode of operation. In this paper, we consider modes of operations for constructing length preserving strong pseudo-random permutation (SPRP) on variable and arbitrary length strings. Additionally, the construction should support a tweak, which provides flexibility in applications. Such a primitive is called a tweakable enciphering scheme (TES).

The known constructions of TESs can be broadly divided into two types: encrypt-mix-encrypt type [5, 6, 3, 19] and the hash-encrypt-hash type [11, 12, 21, 2, 4, 18]. The first type uses a block cipher while the second type additionally requires a finite field multiplier. The feature common to all previous constructions is that they require both the encryption and the decryption functions of the underlying block cipher.

This paper describes new hash-encrypt-hash type of TES constructions. The major innovation of the current work is that the decryption function of the block cipher is not required. Let us clarify what we mean by this. Each block cipher has an encryption and a decryption function. Similarly, a TES also has an encryption and a decryption algorithm. We describe TES constructions whose both encryption and decryption algorithms can be constructed using only the encryption function of the underlying block cipher.

This is achieved without loss in efficiency compared to the best previously known schemes [18]. From a practical point of view this is significant, since the decryption function of the block cipher need not be implemented. The major application of TES is disk encryption where the implementation of TES is done in hardware which typically resides just above the disk controller. Thus, for disk encryption application the new constructions lead to substantial savings in hardware.

1.1 Previous and Related Works

The first construction of an SPRP from a pseudo-random function (PRF) is due to Luby and Rackoff [9] which uses an n-bit to n-bit PRF to construct a $2n$-bit SPRP. A long line of research

([10, 15, 16, 7] and other papers) has refined and polished the construction and currently several variants are known. Naor and Reingold [14, 13] described a construction of SPRP using a block cipher. The notion of tweakable block cipher was introduced in [8]. Halevi and Rogaway [5] provided the first construction of a TES from a block cipher. As mentioned earlier, subsequent research has yielded several constructions of TES from a block cipher.

2 Construction

We follow the notation used in [18].

Choice of finite field. Let $\mathbb{F} = GF(2^n)$ be the finite field of 2^n elements. The addition operation over $\mathbb{F}$ will be denoted by $\oplus$. Elements of $\mathbb{F}$ can also be considered to be n-bit strings.

XOR-universal hash function. Let $h : \mathcal{K} \times \mathbb{F}^m \to \mathbb{F}$ be a function such that for $\mathbf{X}, \mathbf{X}' \in \mathbb{F}^m$, $\mathbf{X} \neq \mathbf{X}'$ and for any $\gamma \in \mathbb{F}$, $\Pr_\tau[h_\tau(\mathbf{X}) \oplus h_\tau(\mathbf{X}') = \gamma] \leq \epsilon$, where $h_\tau(\mathbf{X}) \stackrel{\Delta}{=} h(\tau, \mathbf{X})$ and the probability is over uniform random choice of τ. Such an h is called an ϵ-almost XOR universal hash function. Note that ϵ could depend on m (correspondingly, we write ϵ_m) and indeed does so in the examples below.

1. Usual polynomial hashing is quite well known, where

$$h_\tau(X_1, \ldots, X_m) = \tau \mathsf{Poly}_\tau(X_1, \ldots, X_m) \stackrel{\Delta}{=} \tau \left(\tau^{m-1} X_1 \oplus \cdots \oplus \tau X_{m-1} \oplus X_m \right).$$

   Using Horner's rule $\mathsf{Poly}_\tau(X_1, \ldots, X_m)$ can be evaluated using m multiplications and is $m/2^n$-almost XOR universal.
2. A faster option is to use a class of polynomials introduced by Bernstein [1] based on earlier work by Rabin and Winograd [17] which we call the BRW polynomials. For the exact definition of these polynomials see [1]. Here we note that

$$h_\tau(X_1, \ldots, X_m) = \tau \mathsf{BRW}(X_1, \ldots, X_m)$$

   is $2m/2^n$-almost XOR universal and $\mathsf{BRW}_\tau(X_1, \ldots, X_m)$ can be evaluated using $\lfloor m/2 \rfloor$ multiplications (assuming that $\tau, \tau^2, \tau^4, \ldots$ are pre-computed and available).

The number of multiplications required by BRW is half that required by Poly; on the other hand, the processing of Poly can be made faster by using a pre-computed multiplication table for τ, while this cannot be done for BRW. Other XOR universal hash functions such as those defined in [20] can also be used with our construction.

XOR Universal Function from a Linear Map. Let $\phi : \mathbb{F} \times \{0, \ldots, 2^n - 2\} \to \mathbb{F}$ be a map such that for $0 \leq i < j \leq 2^n - 2$, and for any element $\gamma \in \mathbb{F}$, $\Pr_\beta[\phi_\beta(i) \oplus \phi_\beta(j) = \gamma] = 1/2^n$ where $\phi_\beta(i) \stackrel{\Delta}{=} \phi(\beta, i)$. A simple implementation of such a map is $\phi_\beta(i) = \alpha^i \beta$, where α is a primitive element of $\mathbb{F}$. Using a tower field representation of $\mathbb{F}$ it is possible to obtain a faster implementation of ϕ (see [19]).

A new hash function. Let $h_\tau : \cup_{i \geq 1} \mathbb{F}^i \to \mathbb{F}$ be a XOR universal hash function, such that for $\mathbf{X}, \mathbf{X}' \in \mathbb{F}^m$ with $\mathbf{X} \neq \mathbf{X}'$, $\Pr_\tau[h_\tau(\mathbf{X}) = h_\tau(\mathbf{X}')] \leq \epsilon_m$. Define $\Upsilon_\tau : \cup_{i \geq 3} \mathbb{F}^i \to \mathbb{F}^2$ as follows

$$\Upsilon_\tau(X_1, X_2, X_3, \ldots, X_m) = (X_1 \oplus Z, X_2 \oplus Z) \tag{1}$$

where $Z = h_\tau(X_3, \ldots, X_m)$. It is not difficult to show that for $\mathbf{X} = (X_1, \ldots, X_m) \neq (X'_1, \ldots, X'_m) = \mathbf{X}'$, $\Pr_\tau[\Upsilon_\tau(\mathbf{X}) = \Upsilon_\tau(\mathbf{X}')] \leq \epsilon_{m-2}$.

Encryption function of a block cipher. Let $E_K : \{0,1\}^n \to \{0,1\}^n$ be the encryption function of a block cipher. The key K is from a suitable key space and there is no restriction on this key space. Importantly, we do not require the decryption function, i.e., the inverse of E_K. In the security proof we assume E_K to be a PRF. The invertibility of E_K is not required in either the construction or the proof.

Definition of the basic modes of operations.

ECB: $\mathrm{ECB}_K(X_1, \ldots, X_m) = (E_K(X_1), \ldots, E_K(X_m))$.
MCtr: for $\beta, S \in GF(2^n)$, define $\mathsf{MCtr}_{K,\beta,S}(X_1, \ldots, X_m) = \mathrm{ECB}_K(S_1, \ldots, S_m) \oplus (X_1, \ldots, X_m)$, where $S_i = \phi_\beta(i-1) \oplus S$.
OFB: for $S \in GF(2^n)$, define $\mathsf{OFB}_{K,S}(X_1, \ldots, X_m) = (X_1, \ldots, X_m) \oplus (S_1, \ldots, S_m)$, where $S_i = E_K^i(S)$, i.e., $S_1 = E_K(S)$ and $S_i = E_K(S_{i-1})$.

The tweakable enciphering scheme will be defined using either the modified counter mode or the OFB mode. The description of the algorithm is given in a unified manner using the notation $\mathsf{Mode}_{K,\beta_1,S}(X_1, \ldots, X_m)$. The definition of this notation is given below.

Counter: $\mathsf{Mode}_{K,\beta_1,S}(X_1, \ldots, X_m) = \mathsf{MCtr}_{K,\beta_1,S}(X_1, \ldots, X_m)$.
OFB: $\mathsf{Mode}_{K,\beta_1,S}(X_1, \ldots, X_m) = \mathsf{OFB}_{K,S}(X_1, \ldots, X_m)$. The quantity β_1 is not used.

Parsing. Let the length of the plaintext or the ciphertext be ℓ bits. Write $\ell = (m-1)n + r$, where $1 \le r \le n$. There are a total of m blocks $X_1, \ldots, X_m$, with $X_1, \ldots, X_{m-1}$ being full blocks and the length of the last block is r which is a possible partial block. *We require $\ell > 2n$ so that $m \ge 3$.*

Notation.

$\mathsf{First}_r(Z)$: the most significant r bits of the n-bit binary string Z.
$\mathsf{pad}_i(Z)$: append i zero bits to the string Z.
$\mathsf{drop}_i(Z)$: drop i zero bits from the end of the string Z.

The details of the encryption and the decryption algorithms for the tweakable enciphering scheme are shown in Table 1. The required definition of the Feistel structure is shown in Table 2 while the definition of the hashing keys is shown in Table 3.

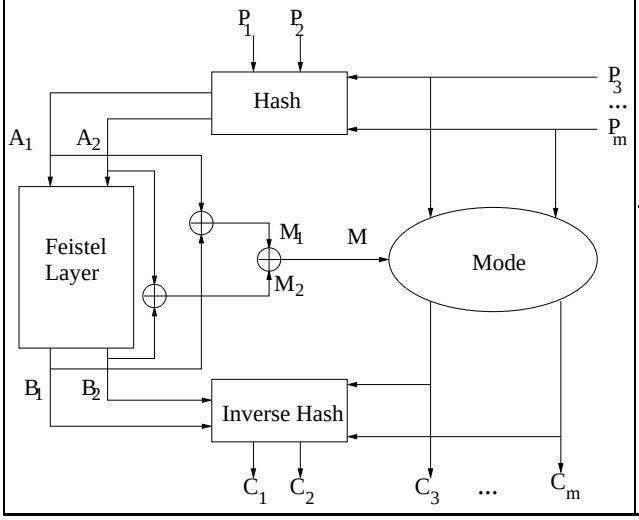
The notation $\mathsf{TES}[\mathsf{Mode}, \mathsf{Hash}, \mathsf{KeyDef}]$ denotes the tweakable enciphering scheme obtained by plugging in the appropriate mode, the hash function h and the suitable key definition. For example $\mathsf{TES}[\mathsf{MCtr}, \mathsf{BRW}, \mathsf{KeyDef3}]$ denotes the tweakable enciphering scheme obtained by using the modified counter mode, the BRW polynomials for h and $\mathsf{KeyDef3}$ for the key definition.

2.1 Intuition behind the construction

The basic idea is quite simple. The hash function Υ hashes the message to produce the input to the Feistel layer. So, for distinct messages, the inputs to the Feistel layer are distinct. The 4-round Feistel layer constructs a strong pseudo-random permutation. The XOR of the input and the output of the Feistel layer is used to form the initialization vector for the mode of operation.

In previous constructions based on the counter (or the OFB) mode, the XOR of the input and output of only the first block was used to initialize the counter mode. (This technique was first used in [21] and later followed in [2, 18].) Since a single block is involved, inverting this block requires the decryption function of the block cipher.

Table 1. Encryption and decryption algorithms. The block cipher key is K; and the hash key is $(\tau, \tau', \beta_1, \beta_2)$. Definitions of τ, β_1 and β_2 are given in Table 3. Let $\mathbf{K} = (K, \tau, \tau', \beta_1, \beta_2)$.



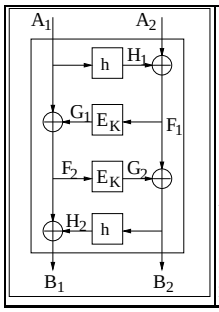
Algorithm $\mathsf{Encrypt}^T_{\mathbf{K}}(P_1, \ldots, P_m)$
1. $M_m = \mathsf{pad}_{n-r}(P_m)$;
2. $(A_1, A_2) = \Upsilon_\tau(P_1, \ldots, P_{m-1}, M_m) \oplus (\beta_1, \beta_1)$;
3. $(B_1, B_2) = \mathsf{Feistel}_{K,\tau'}(A_1, A_2)$;
4. $M_1 = A_1 \oplus B_1$; $M_2 = A_2 \oplus B_2$; $M = M_1 \oplus M_2$;
5. $(C_3, \ldots, C_{m-1}, D_m)$
   $= \mathsf{Mode}_{K,\beta_1,M}(P_3, \ldots, P_{m-1}, M_m)$;
6. $C_m = \mathsf{drop}_{n-r}(D_m)$; $U_m = \mathsf{pad}_{n-r}(C_m)$;
7. $(C_1, C_2) = \Upsilon_\tau(B_1, B_2, C_3, \ldots, C_{m-1}, U_m) \oplus (\beta_2, \beta_2)$;

return $(C_1, \ldots, C_m)$.

Algorithm $\mathsf{Decrypt}^T_{\mathbf{K}}(P_1, \ldots, P_m)$
1. $U_m = \mathsf{pad}_{n-r}(C_m)$;
2. $(B_1, B_2) = \Upsilon_\tau(C_1, \ldots, C_{m-1}, U_m) \oplus (\beta_2, \beta_2)$;
3. $(A_1, A_2) = \mathsf{Feistel}^{-1}_{K,\tau'}(B_1, B_2)$;
4. $M_1 = A_1 \oplus B_1$; $M_2 = A_2 \oplus B_2$; $M = M_1 \oplus M_2$;
5. $(P_3, \ldots, P_{m-1}, V_m)$
   $= \mathsf{Mode}_{K,\beta_1,M}(C_3, \ldots, C_{m-1}, U_m)$;
6. $P_m = \mathsf{drop}_{n-r}(V_m)$; $M_m = \mathsf{pad}_{n-r}(P_m)$;
7. $(P_1, P_2) = \Upsilon_\tau(A_1, A_2, P_3, \ldots, P_{m-1}, M_m) \oplus (\beta_1, \beta_1)$;

return $(P_1, \ldots, P_m)$.

Table 2. A four-round Feistel construction required in Figure 1. Note that if the two input blocks are swapped then decryption can be done using the encryption module (as is usual for Feistel construction).



$\mathsf{Feistel}_{K,\tau'}(A_1, A_2)$
1. $H_1 = h_{\tau'}(A_1)$;
2. $F_1 = H_1 \oplus A_2$;
3. $F_2 = A_1 \oplus E_K(F_1)$;
4. $B_2 = F_1 \oplus E_K(F_2)$;
5. $H_2 = h_{\tau'}(B_2)$;
6. $B_1 = H_2 \oplus F_2$;

return (B_1, B_2).

$\mathsf{Feistel}^{-1}_{K,\tau'}(B_1, B_2)$
1. $H_2 = h_{\tau'}(B_2)$;
2. $F_2 = B_1 \oplus H_2$;
3. $F_1 = B_2 \oplus E_K(F_2)$;
4. $A_1 = F_2 \oplus E_K(F_1)$;
5. $H_1 = h_{\tau'}(A_1)$;
6. $A_2 = H_1 \oplus F_1$;

return (A_1, A_2).

Table 3. Different definitions of the hashing keys τ, τ', β_1 and β_2. Here T is an n-bit tweak and ℓ is the length of the message (or ciphertext) in bits. The block cipher key K is chosen uniformly at random from the appropriate key space. Also, in KeyDef2 and KeyDef3, τ and τ' are chosen uniformly at random from $\{0,1\}^n$.

KeyDef1	KeyDef2	KeyDef3
BC key: K	BC key: K; hash keys: τ, τ'	BC key: K; hash keys: τ, τ'
1. $\gamma = E_K(T)$; 2. $\beta_1 = E_K(\gamma \oplus \mathsf{bin}_n(\ell))$; 3. $\beta_2 = \phi_{\beta_1}(1)$; 4. $\tau = \gamma$; 5. $\tau' = E_K(\phi_{\beta_1}(2))$.	1. $\gamma = E_K(T)$; 2. $\beta_1 = E_K(\gamma \oplus \mathsf{bin}_n(\ell))$; 3. $\beta_2 = \phi_{\beta_1}(1)$.	1. $\beta_1 = E_K(T)$; 2. $\beta_2 = \phi_{\beta_1}(1)$.

Table 4. Performance of the new modes of operations. We assume that there are m (full or partial) blocks.

	KeyDef1	KeyDef2	KeyDef3
# keys	1[BK]	1[BK]+2[AK]	1[BK]+2[AK]
# BC calls	$m+3$	$m+2$	$m+1$

	Poly	BRW
# mults.	$2(m-1)$	$4+2\lfloor (m-2)/2 \rfloor$

We have modified this idea to separate the processing into two parts – the first two blocks and the rest of the message. An invertible map on the first two blocks is created using the Feistel structure. This involves only the encryption function of the block cipher and due to the propery of the Feistel structure, during decryption also, only the encryption function is required. At the same time, we should mention that even though the idea is simple, it takes a bit of effort to work out the polished form and the exact details of the construction and the proof.

Note that the hash key used in the Feistel structure is different from the hash key used outside this structure, even though the basic XOR-universal hash function h is the same in both cases. Using a single key would have been more convenient, but, then the probability computation in Lemma 2 that we perform later would not go through. This fact forced us to use separate and independent keys.

Alternative structures. The Feistel structure that we have used makes two multiplications and two block cipher calls. There are several other Feistel based constructions with different costs all of which convert an n-bit to n-bit PRF to a $2n$-bit SPRP. Conceptually, it is possible to use these structures to replace the Feistel structure that we use; however, one has to be careful with the actual details. Our choice of the particular Feistel structure was based upon simplicity and efficiency.

2.2 Avoiding Decryption: Inefficient Approaches

From a conceptual point of view, there is a generic idea to avoid using the decryption function of the block cipher. As mentioned earlier, Feistel based constructions convert an n-bit to n-bit PRF to a $2n$-bit SPRP. Given such an SPRP, previously known constructions can be used to obtain TESs. Any such TES would not require the decryption function of the block cipher. However, such a conversion would also be inefficient. For the encrypt-mix-encrypt type conversions, the cost would be between 4[BC] to 2[BC]+2[M] per block, where [BC] is a block cipher call and [M] is a multiplication over $GF(2^n)$. On the other hand, the hash-encrypt-hash type constructions will require $2n$-bit multiplications which is about 3 n-bit multiplications, leading to a cost of about 1[BC]+1.5[M] per block.

2.3 Efficiency and Comparison

Performance of the new constructions is given in Table 4. For BRW polynomials, the efficiency comes to about 1[BC]+1[M] per block, which is comparable to the best previously known efficiency [18]. The main advantage of the new constructions is that the decryption function of the block function is not required at all. This is an advantage for hardware implementation, since it results in substantial savings in the size of the hardware. One of the main application of TES is disk encryption and the implementation is typically in hardware. For this application, the new construction offers a significant advantage.

The only restriction is that the message length must be greater than $2n$ bits while in the previous constructions the length was required to be at least n bits. There is no upper bound on the length and messages of varying lengths including partial blocks can be handled quite efficiently. The restriction of more than $2n$ bits on the message length is not a concern for disk encryption and other practical applications.

The other drawback compared to the best previously known constructions [18] is that if the key of the TES is only the block cipher key, then an extra block cipher invocation is required to produce the key for the hash function in the Feistel structure (KeyDef1); otherwise an extra n-bit

key is required (KeyDef2 and KeyDef3). We consider this to be a minor trade-off compared to the fact that it is not required to implement the decryption module at all.

3 Security

We consider information theoretic security and hence without loss of generality the adversary $\mathcal{A}$ is considered to be a deterministic algorithm. See [5] for other notions of security and the relation between computational and information theoretic security.

An adversary $\mathcal{A}$ interacts with the encryption and the decryption oracles of the tweakable enciphering scheme and finally outputs either 0 or 1. Oracles are written as superscripts. The encryption oracle $\mathbf{\Pi}$ takes as input (T, P), where T is an n-bit string and P is a string of length greater than $2n$ and returns C which is of length equal to that of P. Similarly, the decryption oracle $\mathbf{\Pi}^{-1}$ takes as input (T, C) and returns P. The notation $\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1$ denotes the event that the adversary $\mathcal{A}$, interacts with the oracles $\mathbf{\Pi}$ and $\mathbf{\Pi}^{-1}$, and finally outputs the bit 1.

Pointless queries: We assume that an adversary never repeats a query, i.e., it does not ask the encryption oracle with a particular value of (T, P) more than once and neither does it ask the decryption oracle with a particular value of (T, C) more than once. Furthermore, an adversary never queries its deciphering oracle with (T, C) if it got C in response to an encipher query (T, P) for some P. Similarly, the adversary never queries its enciphering oracle with (T, P) if it got P as a response to a decipher query of (T, C) for some C. These queries are called *pointless* as the adversary knows what it would get as responses to such queries.

We consider an adversary's advantage in distinguishing a tweakable enciphering scheme from an oracle which simply returns random bit strings. This advantage is defined in the following manner.

$$\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathbf{\Pi}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(.,.),\$(.,.)} \Rightarrow 1\right]$$

where $\$(., M)$ returns random bits of length $|M|$.

The query complexity σ_n of an adversary is defined to be the total number of n-bit blocks it provides in all its encryption and decryption queries. This includes the plaintext and ciphertext blocks as well as the n-bit tweak. By $\mathbf{Adv}(\sigma_n)$ (with suitable sub and super-scripts) we denote the maximum advantage of any adversary with query complexity σ_n.

Theorem 1. *Fix n and σ_n to be positive integers. Suppose that an adversary uses a total of σ_n blocks in all its queries, where each block is an n-bit string. Then*

$$\mathbf{Adv}^{\pm\mathrm{rnd}}_{\mathbf{\Pi}}(\sigma_n) \leq \frac{14\sigma_n^2}{2^n} \tag{2}$$

Here $\mathbf{\Pi}$ is TES[Mode, Hash, KeyDef] *with* Mode, Hash *and* KeyDef *being instantiated by any of the constructions described in Section 2.*

The proof of Theorem 1 is described in the next two subsections.

3.1 Internal Variables

A variable ty is used to indicate whether a call to $\mathbf{\Pi}$ or $\mathbf{\Pi}^{-1}$ is made. If the call is to $\mathbf{\Pi}$, then ty = enc otherwise, ty = dec.

There are two block cipher calls in the Feistel structure and $(m-2)$ block cipher calls in the "mode of operation" part. Denote the inputs and outputs of the Feistel calls by F_1, F_2 and G_1, G_2 respectively and denote the inputs and outputs of the other $(m-2)$ calls by $F_3, \ldots, F_m$ and $G_3, \ldots, G_m$ respectively.

In addition to these, several encryption calls are made in the key definition part. Define

$$J_0 = T; J_1 = \gamma \oplus \mathsf{bin}_n(\ell); J_2 = \phi_{\beta_1}(2); L_0 = \gamma; L_1 = \beta_1; L_2 = \tau'.$$

The quantities $F_1, \ldots, F_m$ and $G_1, \ldots, G_m$ can be expressed in terms of the plaintext and ciphertext blocks. We show how this can be done.

The variables F_1, F_2 and G_1, G_2.

$$\begin{array}{ll} F_1 = h_{\tau'}(A_1) \oplus A_2; & F_2 = B_1 \oplus h_{\tau'}(B_2); \\ G_1 = A_1 \oplus F_2 = A_1 \oplus B_1 \oplus h_{\tau'}(B_2); & G_2 = B_2 \oplus F_1 = B_2 \oplus A_2 \oplus h_{\tau'}(A_1); \end{array}$$

where $M_m = \mathsf{pad}_{n-r}(P_m)$, $U_m = \mathsf{pad}_{n-r}(C_m)$ and

$$\begin{array}{ll} A_1 = \beta_1 \oplus P_1 \oplus h_\tau(P_3, \ldots, P_{m-1}, M_m); & A_2 = \beta_1 \oplus P_2 \oplus h_\tau(P_3, \ldots, P_{m-1}, M_m); \\ B_1 = \beta_2 \oplus C_1 \oplus h_\tau(C_3, \ldots, C_{m-1}, U_m); & B_2 = \beta_2 \oplus C_2 \oplus h_\tau(C_3, \ldots, C_{m-1}, U_m). \end{array}$$

The variables $F_3, \ldots, F_m$ and $G_3, \ldots, G_m$. For both the modified counter and the OFB modes, the expressions for $G_3, \ldots, G_m$ are same but the expressions for $F_3, \ldots, F_m$ depend on the actual mode that is used. First we provide the expressions for the G_j's and then the expressions for the F_i's.

$$\begin{array}{rll} G_i = P_i \oplus C_i & & \text{for } 3 \le i \le m-1; \\ G_m = \left\{ \begin{array}{ll} D_m \oplus (P_m || 0^{n-r}) & \text{if } \mathsf{ty} = \mathsf{enc}; \\ V_m \oplus (C_m || 0^{n-r}) & \text{if } \mathsf{ty} = \mathsf{dec}; \end{array} \right\} & \text{for } i = m. \end{array}$$

Note that the first r bits of G_m is $P_m \oplus C_m$ irrespective of whether ty is enc or dec. For $3 \le i \le m$, the F_i's are defined as follows.

MCtr: $F_i = M \oplus \phi_{\beta_1}(i-3) = A_1 \oplus A_2 \oplus B_1 \oplus B_2 \oplus \phi_{\beta_1}(i-3)$.
OFB: $F_3 = M$ and for $4 \le i \le m$, $F_i = P_{i-1} \oplus C_{i-1}$.

Below we prove some results on the probability of certain kinds of collisions.

Lemma 1. *Let τ' and β_1 be chosen independently and uniformly at random from $\mathbb{F}$. For any $(P_1, \ldots, P_{m-1}, M_m)$ and $(C_1, \ldots, C_{m-1}, U_m)$, $\Pr_{\tau,\beta_1}[F_1 = F_2] = 1/2^{n-1}$.*

Proof. $A_1 = \beta_1 \oplus \mathsf{rest}$ and $B_2 = \beta_2 \oplus \mathsf{rest}_1$, where rest and rest_1 are independent of both β_1 and β_2. So, $A_1 = B_2$ holds if and only if $\beta_1 \oplus \beta_2 = \mathsf{rest} \oplus \mathsf{rest}_1$. Since $\beta_2 = \phi_{\beta_1}(1)$, by the XOR universality of ϕ, the last equation holds with probability at most $1/2^n$. In other words, $\Pr[A_1 = B_2] \le 1/2^n$.

$$\begin{aligned} \Pr[F_1 = F_2] &= \Pr_{\tau',\beta_1}[h_{\tau'}(A_1) \oplus A_2 = h_{\tau'}(B_2) \oplus B_1] \\ &= \Pr_{\tau',\beta_1}[(h_{\tau'}(A_1) \oplus A_2 = h_{\tau'}(B_2) \oplus B_1) \wedge ((A_1 = B_2) \vee (A_1 \ne B_2))] \\ &= \Pr_{\tau',\beta_1}[(h_{\tau'}(A_1) \oplus A_2 = h_{\tau'}(B_2) \oplus B_1) \,|\, (A_1 = B_2)] \Pr_{\beta_1}[A_1 = B_2] \\ &\quad + \Pr_{\tau',\beta_1}[(h_{\tau'}(A_1) \oplus A_2 = h_{\tau'}(B_2) \oplus B_1) \,|\, (A_1 \ne B_2)] \Pr_{\beta_1}[A_1 \ne B_2] \\ &\le \Pr_{\beta_1}[A_1 = B_2] + \Pr_{\tau'}[(h_{\tau'}(A_1) \oplus A_2 = h_{\tau'}(B_2) \oplus B_1) \,|\, (A_1 \ne B_2))] \\ &\le \frac{2}{2^n}. \end{aligned}$$

The last inequality follows from the XOR-universal property of h. □

Lemma 2. *Suppose that τ and τ' are chosen independently and uniformly at random from $\mathbb{F}$.*

1. *If $(P_1, \ldots, P_{m-1}, M_m) \neq (P'_1, \ldots, P'_{m-1}, M'_m)$, then $\Pr_{\tau,\tau'}[F_1 = F'_1] \leq \epsilon_{m-2} + 1/2^n$.*
2. *If $(C_1, \ldots, C_{m-1}, U_m) \neq (U'_1, \ldots, U'_{m-1}, U'_m)$, then $\Pr_{\tau,\tau'}[F_2 = F'_2] \leq \epsilon_{m-2} + 1/2^n$.*

Proof. We prove the first item, the proof of the other one being similar. $F_1 = h_{\tau'}(A_1) \oplus A_2$ and $F'_1 = h_{\tau'}(A'_1) \oplus A'_2$. Suppose that $(P_3, \ldots, P_{m-1}, M_m) = (P'_3, \ldots, P'_{m-1}, M'_m)$. Then $(P_1, P_2) \neq (P'_1, P'_2)$. If $P_1 = P'_1$ (and so, $P_2 \neq P'_2$), then $A_1 = A'_1$. $P_2 \neq P'_2$ implies $A_2 \neq A'_2$ whereby we have $F_1 \neq F'_1$. If $P_2 = P'_2$ (and so, $P_1 \neq P'_1$), then $A_2 = A'_2$. $P_1 \neq P'_1$ implies $A_1 \neq A'_1$ whereby $F_1 = F'_1$ implies $h_\tau(A_1) = h_\tau(A'_1)$. Since $A_1 \neq A'_1$, by the XOR-universality of h, the last condition holds with probability $1/2^n$.

Suppose now that $(P_3, \ldots, P_{m-1}, M_m) \neq (P'_3, \ldots, P'_{m-1}, M'_m)$. By the XOR-universality of h, $\Pr_\tau[A_1 = A'_1] \leq \epsilon_{m-2}$. Also, note that A_1, A_2, A'_1 and A'_2 are independent of τ'.

$$\begin{aligned}\Pr_{\tau,\tau'}[F_1 = F'_1] &= \Pr_{\tau,\tau'}[h_{\tau'}(A_1) \oplus A_2 = h_{\tau'}(A'_1) \oplus A'_2] \\ &\leq \Pr_{\tau}[A_1 = A'_1] + \Pr_{\tau'}[h_{\tau'}(A_1) \oplus A_2 = h_{\tau'}(A'_1) \oplus A'_2 | (A_1 \neq A'_1)] \\ &\leq \epsilon_{m-2} + \frac{1}{2^n}.\end{aligned}$$

The last relation holds due to the XOR-universality of h. □

3.2 Collision Analysis

The adversary makes a total of q queries of possibly different lengths. We use the superscript $^{(s)}$ to denote quantities corresponding to the s-th query. For example, the length is $\ell^{(s)}$; the number of blocks is $m^{(s)}$; the tweak is $T^{(s)}$; the plaintext blocks are $P_1^{(s)}, \ldots, P_{m^{(s)}}^{(s)}$ and the ciphertext blocks are $C_1^{(s)}, \ldots, C_{m^{(s)}}^{(s)}$. Similarly, we denote the internal variables.

A query can be either an encryption or a decryption query. The variable $\mathsf{ty}^{(s)}$ denotes the type of the query, i.e., if the query is an encryption query, then $\mathsf{ty}^{(s)} = \mathsf{enc}$, and if the query is a decryption query, then $\mathsf{ty}^{(s)} = \mathsf{dec}$.

The responses to the queries are as follows.

Encryption query: $(\mathsf{ty}^{(s)} = \mathsf{enc}, T^{(s)}, P_1^{(s)}, \ldots, P_{m^{(s)}-1}^{(s)}, P_{m^{(s)}}^{(s)})$,
where $\ell^{(s)} = (m^{(s)} - 1)n + r^{(s)}$ and $|P_{m^{(s)}}^{(s)}| = r^{(s)}$.

1. Choose $C_1^{(s)}, \ldots, C_{m^{(s)}-1}^{(s)}, D_{m^{(s)}}^{(s)}$ independently and uniformly at random from $\{0,1\}^n$.
2. Set $C_{m^{(s)}}^{(s)} = \mathsf{First}_{r^{(s)}}(D_{m^{(s)}}^{(s)})$.
3. Return $(C_1^{(s)}, \ldots, C_{m^{(s)}-1}^{(s)}, C_{m^{(s)}}^{(s)})$.

Decryption query: $(\mathsf{ty}^{(s)} = \mathsf{dec}, T^{(s)}, C_1^{(s)}, \ldots, C_{m^{(s)}-1}^{(s)}, C_{m^{(s)}}^{(s)})$,
where $\ell^{(s)} = (m^{(s)} - 1)n + r^{(s)}$ and $|C_{m^{(s)}}^{(s)}| = r^{(s)}$.

1. Choose $P_1^{(s)}, \ldots, P_{m^{(s)}-1}^{(s)}, U_{m^{(s)}}^{(s)}$ independently and uniformly at random from $\{0,1\}^n$.
2. Set $P_{m^{(s)}}^{(s)} = \mathsf{First}_{r^{(s)}}(U_{m^{(s)}}^{(s)})$.
3. Return $(P_1^{(s)}, \ldots, P_{m^{(s)}-1}^{(s)}, P_{m^{(s)}}^{(s)})$.

For either encrypt or decrypt query, the internal variables in the key definitions are chosen as in Table 5. Note that $J_0^{(s)} = T^{(s)}$, $J_1^{(s)} = \gamma^{(s)} \oplus \mathsf{bin}_n(\ell^{(s)})$, $J_2^{(s)} = \phi_{\beta_1^{(s)}}(2)$; and $L_0^{(s)} = \gamma^{(s)}$, $L_1^{(s)} = \beta_1^{(s)}$, $L_2^{(s)} = \tau'^{(s)}$. Also, the internal variables $F_1^{(s)}, \ldots, F_{m^{(s)}}^{(s)}$ and $G_1^{(s)}, \ldots, G_{m^{(s)}}^{(s)}$ are defined from the P_i's and the C_j's as in Section 3.1.

Let $\mathcal{D}$ and $\mathcal{R}$ be two sets of random variables where $\mathcal{D}$ (resp $\mathcal{R}$) is the collection of all random variables which occur as an input to (resp. output of) an encryption. These sets are defined as shown in Table 5.

We define $\mathsf{Coll}(\mathcal{D})$ to be the event that two distinct random variables in $\mathcal{D}$ take the same value which we call a collision. Similarly, we define the event $\mathsf{Coll}(\mathcal{R})$ and $\mathsf{Coll} = \mathsf{Coll}(\mathcal{D}) \vee \mathsf{Coll}(\mathcal{R})$. Further, $\overline{\mathsf{Coll}}$ denotes the event that there is no collision. For any adversary $\mathcal{A}$, we clearly have

$$\Pr[\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1 | \overline{\mathsf{Coll}}] = \Pr[\mathcal{A}^{\$,\$} \Rightarrow 1].$$

In other words, if Coll does not occur, then in the real game, i.e. during the interaction with $\mathbf{\Pi}, \mathbf{\Pi}^{-1}$, the adversary gets independent and uniform random strings as responses which is the same as it would get while interacting with oracles that return independent and uniform random strings.

$$\begin{aligned}
\Pr[\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1] &= \Pr[(\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1) \wedge (\mathsf{Coll} \vee \overline{\mathsf{Coll}})] \\
&= \Pr[(\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1) | \mathsf{Coll}] \Pr[\mathsf{Coll}] + \Pr[(\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1) | \overline{\mathsf{Coll}}] \Pr[\overline{\mathsf{Coll}}] \\
&\leq \Pr[\mathsf{Coll}] + \Pr[\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1 | \overline{\mathsf{Coll}}] \\
&= \Pr[\mathsf{Coll}] + \Pr[\mathcal{A}^{\$,\$} \Rightarrow 1].
\end{aligned}$$

This gives $\mathsf{Adv}_{\mathbf{\Pi}}(\mathcal{A}) = \Pr[\mathcal{A}^{\mathbf{\Pi},\mathbf{\Pi}^{-1}} \Rightarrow 1] - \Pr[\mathcal{A}^{\$,\$} \Rightarrow 1] \leq \Pr[\mathsf{Coll}]$. We next show that for any adversary with query complexity σ, $\Pr[\mathsf{Coll}] \leq 14\sigma^2/2^n$ which in turn would show that $\mathsf{Adv}_{\mathbf{\Pi}}(\sigma) \leq 14\sigma^2/2^n$ and so would complete the proof of Theorem 1.

The actual collision analysis to upper bound $\Pr[\mathsf{Coll}(\mathcal{D})]$ depends on the key definition and the mode of operation. We perform this task for $\mathsf{KeyDef1}$ and the modified counter mode of operation. Analysis of the other cases are similar. Consider the s-th and the r-th query. There are two possibilities.

$s = r$. In this case, we need to consider collision between the following pairs.

$$(J_i^{(s)}, J_j^{(s)}),\ 0 \leq i < j \leq 2;\ (F_i^{(s)}, F_j^{(s)}),\ 1 \leq i < j \leq m;\ (J_i^{(s)}, F_j^{(s)}),\ 0 \leq i \leq 2,\ 1 \leq j \leq m.$$

From the definition of the J_i's and Table 5 it is easy to see that $\Pr[J_i^{(s)} = J_j^{(s)}] = 1/2^n$ and $\Pr[J_i^{(s)} = F_j^{(s)}] = 1/2^n$. If $\mathsf{ty}^{(s)} = \mathsf{enc}$, then $C_1^{(s)}$ is chosen uniformly at random. $F_1^{(s)}$ is independent of C_1, while $F_2^{(s)}$ can be written as $C_1^{(s)} \oplus X$, where X is independent of $C_1^{(s)}$. So, $\Pr[F_1^{(s)} = F_2^{(s)}] = 1/2^n$. Similarly, if $\mathsf{ty}^{(s)} = \mathsf{dec}$, then $P_1^{(s)}$ is chosen uniformly at random and we again have $\Pr[F_1^{(s)} = F_2^{(s)}] = 1/2^n$. For $3 \leq i < j \leq m$, $F_i^{(s)}$ and $F_j^{(s)}$ are distinct. Further, $\Pr[F_1^{(s)} = F_i^{(s)}]$ for $i \geq 3$ can easily be seen to be $1/2^n$ and similarly for $\Pr[F_2^{(s)} = F_i^{(s)}]$.

$s \neq r$. In this case, we need to consider collision between the following pairs.

$$\begin{array}{ll}
(J_i^{(s)}, J_j^{(r)}),\ 0 \leq i, j \leq 2; & (F_i^{(s)}, F_j^{(r)}),\ 1 \leq i, j \leq m; \\
(J_i^{(s)}, F_j^{(r)}),\ 0 \leq i \leq 2,\ 1 \leq j \leq m; & (F_i^{(s)}, J_j^{(r)}),\ 1 \leq i \leq m,\ 0 \leq j \leq 2.
\end{array}$$

Table 5. Definition of the internal random variables in the key definition and the updation of the sets $\mathcal{D}$ and $\mathcal{R}$ for the s-th query. Initially, $\mathcal{D} = \mathcal{R} = \emptyset$. In RAND2-Sub1 and RAND3-Sub3, τ is chosen uniformly at random at the start of RAND2 and does not change during the game.

RAND2-Sub1(T_s, ℓ_s)	RAND2-Sub2(T_s, ℓ_s)	RAND3-Sub3(T_s)

if $T^{(s)} = T^{(r)}$ for some $r < s$
then
$\tau^{(s)} = \tau^{(r)}$;
if $\ell^{(s)} = \ell^{(r)}$
then $\beta_1^{(s)} = \beta_1^{(r)}$; $\tau'^{(s)} = \tau'^{(r)}$;
else
$\beta_1^{(s)} \xleftarrow{\$} \{0,1\}^n$; $\tau'^{(s)} \xleftarrow{\$} \{0,1\}^n$;
$\mathcal{D} = \mathcal{D} \cup \{\tau^{(s)} \oplus \mathsf{bin}(\ell^{(s)}), \phi_{\beta_1^{(s)}}(2)\}$;
$\mathcal{R} = \mathcal{R} \cup \{\beta_1^{(s)}, \tau'^{(s)}\}$;
endif
else
$\tau^{(s)} \xleftarrow{\$} \{0,1\}^n$; $\beta_1^{(s)} \xleftarrow{\$} \{0,1\}^n$; $\tau'^{(s)} \xleftarrow{\$} \{0,1\}^n$;
$\mathcal{D} = \mathcal{D} \cup \{T^{(s)}, \tau^{(s)} \oplus \mathsf{bin}_n(\ell^{(s)}), \phi_{\beta_1^{(s)}}(2)\}$;
$\mathcal{R} = \mathcal{R} \cup \{\tau^{(s)}, \beta_1^{(s)}, \tau'^{(s)}\}$;
endif;
$\beta_2^{(s)} = \phi_{\beta_1^{(s)}}(1)$.
	if $T^{(s)} = T^{(r)}$ for some $r < s$
	then
	$\gamma^{(s)} = \gamma^{(r)}$;
	if $\ell^{(s)} = \ell^{(r)}$
	then $\beta_1^{(s)} = \beta_1^{(r)}$;
	else
	$\beta_1^{(s)} \xleftarrow{\$} \{0,1\}^n$;
	$\mathcal{D} = \mathcal{D} \cup \{\gamma^{(s)} \oplus \mathsf{bin}(\ell^{(s)})\}$;
	$\mathcal{R} = \mathcal{R} \cup \{\beta_1^{(s)}\}$;
	endif
	else
	$\gamma^{(s)} \xleftarrow{\$} \{0,1\}^n$;
	$\mathcal{D} = \mathcal{D} \cup \{T^{(s)}\}$; $\mathcal{R} = \mathcal{R} \cup \{\gamma^{(s)}\}$;
	$\beta_1^{(s)} \xleftarrow{\$} \{0,1\}^n$;
	$\mathcal{D} = \mathcal{D} \cup \{\gamma^{(s)} \oplus \mathsf{bin}(\ell^{(s)})\}$;
	$\mathcal{R} = \mathcal{R} \cup \{\beta_1^{(s)}\}$;
	endif;
	$\beta_2^{(s)} = \phi_{\beta_1^{(s)}}(1)$.
		if $T^{(s)} = T^{(r)}$ for some $r < s$
		then $\beta_1^{(s)} = \beta_1^{(r)}$;
		else
		$\beta_1^{(s)} \xleftarrow{\$} \{0,1\}^n$;
		$\mathcal{D} = \mathcal{D} \cup \{T^{(s)}\}$;
		$\mathcal{R} = \mathcal{R} \cup \{\beta_1^{(s)}\}$;
		endif;
		$\beta_2^{(s)} = \phi_{\beta_1^{(s)}}(1)$.

$\mathcal{D} = \mathcal{D} \cup \{F_1^{(s)}, \ldots, F_{m^{(s)}}^{(s)}\}$; $\mathcal{R} = \mathcal{D} \cup \{G_1^{(s)}, \ldots, G_{m^{(s)}}^{(s)}\}$		

If $T^{(s)} = T^{(r)}$, then $J_0^{(s)} = J_0^{(r)}$, *but* $J_0^{(s)}$ is not put in $\mathcal{D}$. Similarly, for $J_1^{(s)}$ and $J_2^{(s)}$. Thus, there are no trivial collisions in $\mathcal{D}$ and also similarly for $\mathcal{R}$. The analysis of collisions of the type $(F_i^{(s)}, F_j^{(r)})$ is non-trivial, the other types of collisions can be easily seen to hold with probability at most $1/2^n$.

First note that if $(T^{(s)}, \ell^{(s)}) \neq (T^{(r)}, \ell^{(r)})$, then $\beta_1^{(s)}, \tau'^{(s)}, \beta_1^{(r)}$ and $\tau'^{(r)}$ are independent and uniform random n-bit strings from which it is possible to argue that $\Pr[F_i^{(s)} = F_j^{(r)}] = 1/2^n$.

So we consider the situation when $(T^{(s)}, \ell^{(s)}) = (T^{(r)}, \ell^{(r)})$. In this case, we have $\tau^{(s)} = \tau^{(r)} = \tau$ (say), $\beta_1^{(s)} = \beta_1^{(r)} = \beta_1$ (say) and $\tau'^{(s)} = \tau'^{(r)} = \tau'$ (say). Also, since $(T^{(s)}, \ell^{(s)}) = (T^{(r)}, \ell^{(r)})$, if $\mathsf{ty}^{(s)} = \mathsf{enc}$ then $(P_1^{(s)} \ldots, P_{m^{(s)}}^{(s)}) \neq (P_1^{(r)}, \ldots, P_{m^{(r)}}^{(r)})$; and if $\mathsf{ty}^{(s)} = \mathsf{dec}$ then $(C_1^{(s)}, \ldots, C_{m^{(s)}}^{(s)}) \neq (C_1^{(r)}, \ldots, C_{m^{(r)}}^{(r)})$. We consider different cases.

$(F_1^{(s)}, F_1^{(r)})$, $(F_2^{(s)}, F_2^{(r)})$: if at least one (say the s-th one) is a decrypt query, then using the randomness of $P_2^{(s)}$ it follows that $\Pr[F_1^{(s)} = F_1^{(r)}] = 1/2^n$. Suppose that both are encrypt queries. Since $(P_1^{(s)} \ldots, P_{m^{(s)}}^{(s)}) \neq (P_1^{(r)}, \ldots, P_{m^{(r)}}^{(r)})$, using Lemma 2, we have $\Pr[F_1^{(s)} = F_1^{(r)}] = \epsilon_{m-2} + 1/2^n$. The collision analysis for the pair $(F_2^{(s)}, F_2^{(r)})$ is similar.

$(F_1^{(s)}, F_2^{(r)})$, $(F_2^{(s)}, F_1^{(r)})$: Using Lemma 1, $\Pr[F_1^{(s)} = F_2^{(r)}] = \Pr[F_1^{(r)} = F_2^{(s)}] = 1/2^{n-1}$.

$(F_1^{(s)}, F_i^{(r)})$, $(F_2^{(s)}, F_i^{(r)})$, $i \geq 3$: let $X = P_1^{(r)} \oplus P_2^{(r)} \oplus C_1^{(r)} \oplus C_2^{(r)}$ and note that $F_i^{(r)} = X \oplus \mathsf{rest}$ and $F_1^{(s)} = A_2^{(s)} \oplus h_\tau(A_1^{(s)})$ where X is independent of both rest and $F_1^{(s)}$. Irrespective of whether the r-th query is an encryption or a decryption query, we have X to be a uniformly distributed random variable and so $\Pr[F_1^{(s)} = F_i^{(r)}] = 1/2^n$. Similarly for the pair $(F_2^{(s)}, F_i^{(r)})$.

$(F_i^{(s)}, F_j^{(r)})$ **with** $3 \leq i, j \leq m$**:** in this case, $M^{(s)}$ and $M^{(r)}$ are independent and uniform random quantities so that $\Pr[F_i^{(s)} = F_j^{(r)}] = 1/2^n$.

To summarize, if $(T^{(s)}, \ell^{(s)}) = (T^{(r)}, \ell^{(r)})$ then the collision probability of pairs of the type $(F_1^{(s)}, F_1^{(r)})$, $(F_2^{(s)}, F_2^{(r)})$ is at most $\epsilon_{m-2} + 1/2^n$; collision probability of pairs of the type $(F_1^{(s)}, F_2^{(r)})$, $(F_2^{(s)}, F_1^{(r)})$ is at most $1/2^{n-1}$; and for all other possible pairs in $\mathcal{D}$, the collision probability is at most $1/2^n$. Suppose that there are q_i queries having m_i blocks, $1 \leq i \leq t$, where $q = q_1 + \cdots + q_t$ and the number of elements in $\mathcal{D}$ is at most $\sum_{i=1}^{t} q_i(m_i + 2) = \sigma + q$ where $\sigma = \sum_{i=1}^{t} q_i(m_i + 1)$.

The number of pairs of the type $(F_1^{(s)}, F_1^{(r)})$, $(F_2^{(s)}, F_2^{(r)})$ in $\mathcal{D}$ is at most $\sum_{i=1}^{t} q_i(q_i - 1)$; and the number of pairs of the type $(F_1^{(s)}, F_2^{(r)})$, $(F_2^{(s)}, F_1^{(r)})$ is also $\sum_{i=1}^{t} q_i(q_i - 1)$. So, the maximum probability of a collision in $\mathcal{D}$ is

$$\sum_{i=1}^{t} i \left(\frac{1}{2^n} \epsilon_{m_i - 2} \right) q_i(q_i - 1) + \frac{1}{2^{n-1}} \sum_{i=1}^{t} q_i(q_i - 1) + \frac{1}{2^n} \binom{\sigma + q}{2} \leq \frac{7\sigma^2}{2^n}.$$

In the above we have assumed that $\epsilon_m \leq 2m/2^n$ so that $\epsilon_{m_i - 2} + 1/2^n < 2m_i/2^n$. This corresponds to the bound for BRW polynomials which is weaker than the bound for Poly ($\epsilon_m \leq m/2^n$). Consequently, the constant 7 holds for both hash functions.

A similar analysis for the random variables in $\mathcal{R}$ shows that the probability of $\mathsf{Coll}(\mathcal{R})$, i.e., the probability of a collision in $\mathcal{R}$ is also upper bounded by $7\sigma^2/2^n$. So, the probability of Coll, i.e., the probability of a collision in either $\mathcal{D}$ or $\mathcal{R}$ is at most $14\sigma^2/2^n$. □

Note. The constant 14 can be reduced by a more careful analysis of the inequalities. Since 14 itself is a small enough constant we did not perform this analysis.

References

1. Daniel J. Bernstein. Polynomial evaluation and message authentication, 2007. `http://cr.yp.to/papers.html#pema`.
2. Debrup Chakraborty and Palash Sarkar. HCH: A new tweakable enciphering scheme using the hash-counter-hash approach. *IEEE Transactions on Information Theory*, 54(4):1683–1699, 2008.
3. Shai Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 315–327. Springer, 2004.
4. Shai Halevi. Invertible universal hashing and the TET encryption mode. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 412–429. Springer, 2007.
5. Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 2003.
6. Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 2004.
7. Tetsu Iwata and Kaoru Kurosawa. How to construct super-pseudorandom permutations with short keys. *IEICE Transactions*, 90-A(1):2–13, 2007.
8. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.
9. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
10. Stefan Lucks. Faster Luby-Rackoff ciphers. In Dieter Gollmann, editor, *FSE*, volume 1039 of *Lecture Notes in Computer Science*, pages 189–203. Springer, 1996.
11. David A. McGrew and Scott R. Fluhrer. The extended codebook (XCB) mode of operation. Cryptology ePrint Archive, Report 2004/278, 2004. `http://eprint.iacr.org/`.
12. David A. McGrew and Scott R. Fluhrer. The security of the extended codebook (XCB) mode of operation. In *Selected Areas in Cryptography*, Lecture Notes in Computer Science. Springer, 2007. To appear.

13. Moni Naor and Omer Reingold. A pseudo-random encryption mode. Manuscript available from `www.wisdom.weizmann.ac.il/~naor`.
14. Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *J. Cryptology*, 12(1):29–66, 1999.
15. Jacques Patarin. How to construct pseudorandom and super pseudorandom permutations from one single pseudorandom function. In *EUROCRYPT*, pages 256–266, 1992.
16. Sarvar Patel, Zulfikar Ramzan, and Ganapathy S. Sundaram. Luby-Rackoff ciphers: Why XOR is not so exclusive. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 271–290. Springer, 2002.
17. Michael O. Rabin and Shmuel Winograd. Fast evaluation of polynomials by rational preparation. *Communications on Pure and Applied Mathematics*, 25:433–458, 1972.
18. Palash Sarkar. Efficient tweakable enciphering schemes from (block-wise) universal hash functions). *IEEE Transactions on Information Theory*. to appear.
19. Palash Sarkar. A general mixing strategy for the ECB-Mix-ECB mode of operation. *Inf. Process. Lett.*, 109(2):121–123, 2008.
20. Palash Sarkar. A new universal hash function and other cryptographic algorithms suitable for resource constrained devices. Cryptology ePrint Archive, Report 2008/216, 2008. `http://eprint.iacr.org/`.
21. Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In Dengguo Feng, Dongdai Lin, and Moti Yung, editors, *CISC*, volume 3822 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2005.