Security of XCB and HCTR

Dissertation Submitted In Partial Fulfilment Of
The Requirements For The Degree Of

Master of Technology
in
Computer Science

by

Manish Kumar
[Roll No: CS-1616]

Under the Guidance of

Dr. Debrup Chakraborty
Associate Professor & Head
Cryptology and Security Research Unit (CSRU)

Indian Statistical Institute
Kolkata-700108, India

July 2018

Declaration

I hereby declare that the dissertation report entitled **"Security of XCB and HCTR"** submitted to Indian Statistical Institute, Kolkata, is a *bona fide* record of work carried out in partial fulfilment for the award of the degree of **Master of Technology in Computer Science**. The work has been carried out under the guidance of **Dr. Debrup Chakraborty**, Associate Professor, CSRU, Indian Statistical Institute, Kolkata.

I further declare that this work is original, composed by myself. The work contained herein is my own except where stated otherwise by reference or acknowledgement, and that this work has not been submitted to any other institution for award of any other degree or professional qualification.

Place : Kolkata
Date : July 13, 2018

Manish Kumar
Roll No: CS-1616
Indian Statistical Institute
Kolkata - 700108 , India.

CERTIFICATE

This is to certify that the dissertation entitled **"Security of XCB and HCTR"** submitted by **Manish Kumar** to Indian Statistical Institute, Kolkata, in partial fulfilment for the award of the degree of **Master of Technology in Computer Science** is a *bona fide* record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Debrup Chakraborty
Associate Professor & Head,
Cryptology and Security Research Unit,
Indian Statistical Institute,
Kolkata-700108, India.

“There exist attackers who follow non-violence.”

Acknowledgements

I would like to take this opportunity to thank people who are behind my success in this project.

Prima facie, I would like to thank my parents, family members and teachers who supported me in every walk of my life.

I would like to show my highest gratitude to my adviser, *Prof. Debrup Chakraborty* of Cryptology and Security Research Unit, for his guidance and continuous support and encouragement. His zeal and method of teaching are highly motivating.

I would also like to thank *A V S D Bharadwaj*, student research group of CSRU and “M.Tech. Crypto Cluster” for their valuable suggestions and discussions.

My deepest thanks to all the professor of Indian Statistical Institute, for their valuable suggestions which added an important dimension to my research work.

Last but not the least, I would like to thank all of my friends for their help. I would also like to thank all those, whom I have missed out from the above list.

Manish Kumar
Indian Statistical Institute
Kolkata - 700108 , India.

To my family and supervisor

Abstract

Tweakable Enciphering Scheme (TES) is a length preserving scheme which provides confidentiality and admissible integrity. XCB (Extended Code Book) is a TES which was introduced in 2004. In 2007, it was modified and security bound was provided. Later, these two versions were referred to as XCBv1 and XCBv2 respectively. XCBv2 was proposed as the IEEE-std 1619.2 2010 for encryption of sector oriented storage media. In 2013, first time Security bound of XCBv1 was given and XCBv2's security bound was enhanced. A constant of 2^{22} appears in the security bounds of the XCBv1 and XCBv2.

We showed that this constant of 2^{22} can be reduced to 2^5. Further, we modified the XCB (MXCB) scheme such that it gives better security bound compared to the present XCB scheme. We also analysed some weak keys attack on XCB and a type of TES known as HCTR (proposed in 2005). We performed distinguishing attack and the hash key recovery attack on HCTR. Next we analysed the dependency of the two different keys in HCTR.

Keywords: Disk encryption · IEEE-std 1619.2 2010 · Tweakable enciphering scheme · XCB · MXCB · Weak keys · HCTR.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Day-by-day we are becoming more reliable on the data which can be personal, organisational, top secret, anything or everything. Often we come across the news of data breach and fraud. Recently (in 2018), Cambridge Analytica was in news due to data breach of facebook users. In India, Unique Identification Authority of India (UIDAI) which is responsible for AADHAR scheme is facing data breach on daily basis. In 2012, we generated approx 2.5 exabytes of data every day. Majority (90%) of all the present data is generated in last few years. As data is asset of the 21^{st} century and it is on continuous threat, question arises whether we can make the data secure or not? If yes then up-to what extent? As a wise man said, "The hardest thing of all is to find a black cat in a dark room, especially if there is no cat". Providing security is more or less like that.

At any instant, data can be in two states - stored or in transit. Generally, we are interested in the confidentiality and integrity of the data in both of these states and the techniques involved in these two different scenarios are different. In most schemes which provide integrity/confidentiality in a strong sense a length expansion of the original data takes place, i.e., the transformed data occupies more space than the original data. Though such expansion can be easily tolerated for most scenarios, there are specific application areas where such length expansion cannot be tolerated. One such area is the application of low level disk encryption. Disk encryption ensures confidentiality and integrity of the stored data. Even if the hardware containing the disk is stolen the data stored in it would be unreadable to the adversary, also the adversary would be unable to change the contents of the disk in a meaningful way.

In low level disk encryption, the encryption/decryption algorithm resides on the disk controller and sees the disk as a bare collection of sectors. It encrypts the data before storing it into the sector and decrypts it after reading a sector and before sending it to high level applications. As each sector is of fixed length (4096 bytes, in modern disks), so length expansion after encryption cannot be tolerated. A well accepted solution for the problem of disk encryption is a cryptographic object called a Tweakable Enciphering Scheme (TES).

A TES (Wide block encryption) is a block-cipher mode of operation with the following properties:

1. *Length preserving:* The length of the cipher text is same as the length of the plain text.

2. *Ciphertext Variability:* A TES takes as input a key, a plaintext and a special quantity called the *tweak*. If the same message is encrypted with the same key but different tweaks then unrelated ciphertexts are obtained. In the application of disk encryption, sector addresses are considered as tweaks. This property ensures that even if the same data is stored in two different sectors the cipher texts would look different.

3. *Confidentiality:* The cipher texts produced by a TES are indistinguishable from random strings to any computationally bounded adversary. Which means that any practical adversary would see the cipher texts as random strings and would be thus unable to derive any information regarding the plain text which produced this cipher text.

4. *Integrity:* If a single bit of a valid ciphertext produced by a TES is changed, then this altered ciphertext on decryption will produce a random looking plaintext. This property ensures that an adversary would not be able to alter a ciphertext so that it gets decrypted to something meaningful.

In designing a TES, we have certain goals. We can list them as follows:

1. First and foremost goal is TES's correctness i.e. decryption should undo encryption for every message in the message space.

2. Our next aim, TES should be as efficient as possible. Some of the important dimensions of efficiency expected of a TES are the following:

   - Less running time for encryption/decryption.
   - Small circuit area when implemented in hardware.
   - Low power consumption when implemented in power constrained devices.
   - Small code size and low memory usage in memory constrained software implementations.

3. In TES, we are using tweak for getting variability in the output for the same input. So to change the tweak should be cheaper than changing the key. In most of the block cipher changing the encryption key is relatively expensive since there is need to perform "key setup" operation.

4. TES should be secure i.e. even the adversary has control of the tweak input than also the scheme should be secure.

Designing efficient TESs which provide the required security in provable terms is a challenging problem. In the last two decades there has been some intense work in designing and proving the security of TES. Some of the existing constructions are PEP [5], HCTR[20], HCH[4], TET [6], HEH [18], CMC[7], XCB[11, 12] and EME[8]. TES has been standardised because of its practical application in disk encryption.

1.1 XCB and HCTR

In this dissertation we study two TES called XCB and HCTR.

XCB (Extended Code Book) and HCTR are Hash-Counter-Hash Tweakable Enciphering Schemes i.e. both the schemes use first a layer of universal hash function then a counter (CTR) mode and then again another layer of universal hash function for encryption and decryption. The universal hash functions used in both constructions is a variant of polynomial evaluation hash [21]. Let us consider a message M which is parsed into m blocks $M_1, M_2, \ldots, M_m$, each of length n bits. To hash M using a polynomial hash with a key n bit key H, the polynomial $h_H(M) = \sum_i M_i H^i$ is computed. Variants of this polynomial evaluation hash has been widely used to construct message authentication codes (MAC) [1, 13, 21], authenticated encryption (AE), TES [11, 4, 20] and other cryptographic schemes. CTR mode uses the block cipher to generate the key stream used in the message encryption: $E_K(S_i), i = 1, 2, \cdots$, where K is the key of block cipher and S_i is the number generated by a counter. The main difference between HCTR and XCB is: HCTR has two master keys, one for counter mode and other for universal hash function while XCB has only one master key from that other keys are generated. XCB and HCTR use different variants of the Counter mode and the polynomial evaluation hash [19].

In 2004, McGraw and Fluherer proposed Tweakable Enciphering Scheme (TES) named as XCB in [11] without providing a proof. Later in 2007, they made changes in original construction and proved security of the updated construction in [12]. Authors claim that the changes were made for the improvement of performance of XCB and make it easier to analyse. Later Chakraborty, Hernandez-Jimenez and Sarkar [2] did a detailed analysis of two versions of XCB as described in [11] and [12]. The study in [2] names the version of XCB in [11] as XCBv1 and the one in [12] as XCBv2, we will also follow the same nomenclature. The analysis in [2] concludes that the security claims regarding XCBv2 as presented in [12] are largely erroneous. XCBv2 is completely insecure for certain types of messages, in particular, there is an easy distinguishing attack on XCBv2 if it is used on messages whose length is not a multiple of the block length n of the underlying block cipher. Though XCBv2 is secure for other messages, the proof and the security bound was shown to be incorrect. In [2] a correct security bound for XCBv2 (message for which it is secure) was derived and also a proof for XCBv1 was provided. That proof was based on the analyses done in [10]. In [10] the security bounds of an authenticated encryption scheme called GCM were analyzed, as GCM and XCB shares almost the same hash function. Hence, the techniques used and analyze GCM in [10] could be adopted to analyze XCB in [2]. Further, analysis of the GCM bound was done in [15]. We use the analyses done in [15] to give an improved security bound on XCBv1 and XCBv2. We also modify the XCB (MXCB say) and give its security bound. Further, we compare the improved security bound and MXCB with some existing TES mode having parameter of practical value followed by some weak keys analysis on XCB.

We also do some analysis on HCTR. It was proposed by Wang, Feng and Wu in 2005. It is a mode of operation which provides a tweakable strong pseudorandom permutation [20]. We show how the hash function is insecure. We perform distinguishing attack and the hash key recovery attack on HCTR. Next we analyse the dependency of the two different keys in HCTR. In particular, we analyse the following scenario. Suppose HCTR with keys K and h has been used for some time, and K gets compromised. We show that only changing K would rise to a completely insecure scheme.

1.2 Outline of dissertation

In chapter 2, we discuss and formalise the notion of security for Tweakable Enciphering Scheme. In chapter 3, we formalise the XCB and prove a lemma. Then gives the security proof of XCBv1 followed by security bound for MXCB and comparison with some existing TES. Also, weak keys analysis on XCB. In chapter 4, we discuss the construction of HCTR, distinguishing and key recovery attack on the existing HCTR scheme followed by key dependency of the master keys . In chapter 5, we conclude the discussion.

Chapter 2

Preliminaries

Following are the notation which we will use in subsequent chapter.

2.1 Notation

The set of all n-bit strings will be denoted by $\{0,1\}^n$. For a binary string X, $|X|$ will represent the size of the string in bits. We will use $X\|Y$ for concatenating binary string X and Y; for $r \leq |X|$, r left most and r right most bits of X would be denoted by $\mathrm{msb}_r(X)$ and $\mathrm{lsb}_r(X)$ respectively. By $\mathrm{int}(X)$ we denote the integer represented by the binary string X, $\mathsf{bin}_n(i)$ will denote the n-bit binary representation of i, where the leftmost bit is the most significant bit and i is non-negative such that $i \leq 2^n - 1$. For $X, Y \in \{0,1\}^n, X \oplus Y$ and XY will respectively denote addition and multiplication in $GF(2^n)$. We denote $parse_n(X)$ by $(X_1, X_2, \ldots, X_m)$ where each X_i is of n-bit except last one while $1 \leq |X_m| \leq n$ and cardinality of X would be denoted by $\#X$.

In standard of XCB, field $GF(2^{128})$ is represented by the irreducible polynomial $x^{128}+x^7+x^2+x+1$. Note that selection of irreducible polynomial doesn't affect the security of scheme. Therefore, proofs and attacks are irrespective of the irreducible polynomial.

2.2 Tweakable Enciphering Schemes (TES)

A Tweakable Enciphering Scheme is a pair of functions (**E, D**) where **E** and **D** are the encryption and decryption functions respectively of the enciphering scheme. Here, encryption $\mathbf{E} : \mathcal{K}\times\mathcal{T}\times\mathcal{M} \to \mathcal{M}$ and decryption $\mathbf{D} : \mathcal{K}\times\mathcal{T}\times\mathcal{M} \to \mathcal{M}$, where $\mathcal{K}$ and $\mathcal{T}$ are non-empty sets, and they denote the key space and the tweak space respectively. The message and the cipher space $\mathcal{M} \subseteq \bigcup_{i\geq 1}\{0,1\}^i$. We will denote $\mathbf{E}(K, T, \cdot)$ by $\mathbf{E}_K^T(\cdot)$ and $\mathbf{D}(K, T, \cdot)$ by $\mathbf{D}_K^T(\cdot)$. Encryption and decryption are length preserving i.e. for every $K \in \mathcal{K}$, $M \in \mathcal{M}$ and $T \in \mathcal{T}$ such that $|\mathbf{E}_K^T(X)| = |X|$. For the correction purpose, $X = \mathbf{D}_K^T(Y)$ if and only if $\mathbf{E}_K^T(X) = Y$ where $\mathbf{D} = \mathbf{E}^{-1}$.

2.3 Security of TES

Discussion of this section is based on [8]. An n-bit block-cipher is a function $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{K} \neq \phi$ is the key space for any key $K \in \mathcal{K}$ and $E(K,\cdot)$ is a permutation.

An adversary A is a probabilistic algorithm which can access two oracles and gives output either 0 or 1. The notation $A^{\mathcal{O}_1,\mathcal{O}_2} \Rightarrow 1$ denotes the events that the adversary A, interacts with the oracles $\mathcal{O}_1, \mathcal{O}_2$ and finally output the bit 1. Event of choosing X uniformly at random from the finite set S is represented by $X \xleftarrow{\$} S$.

Let $\text{Perm}(n)$ denotes the set of all permutation on $\{0,1\}^n$. The advantage of A in breaking the strong pseudo randomness of E is defined as

$$\mathbf{Adv}_E^{\pm prp}(A) = \Bigg| Pr\Bigg[K \xleftarrow{\$} \mathcal{K} : A^{E_K(\cdot),E_K^{-1}(\cdot)} \Rightarrow 1\Bigg] \\ - Pr\Bigg[\pi \xleftarrow{\$} \text{Perm}(n) : A^{\pi(\cdot),\pi^{-1}(\cdot)} \Rightarrow 1\Bigg]\Bigg|.$$

Let $Perm^{\mathcal{T}}(\mathcal{M})$ denote the set of all functions $\boldsymbol{\pi} : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\boldsymbol{\pi}(T,\cdot)$ is a length preserving permutation on $\mathcal{M}$ and $\boldsymbol{\pi} \in Perm^{\mathcal{T}}(\mathcal{M})$ is known as indexed permutation. For a Tweakable Encipher Scheme $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, we define the advantage an adversary A in distinguishing $\mathbf{E}$ and its inverse from a random tweak indexed permutation and its inverse in the following way:

$$\mathbf{Adv}_{\mathbf{E}}^{\pm \widetilde{prp}}(A) = \Bigg| Pr\Bigg[K \xleftarrow{\$} \mathcal{K} : A^{\mathbf{E}_K(\cdot,\cdot),\mathbf{E}_K^{-1}(\cdot,\cdot)} \Rightarrow 1\Bigg] \\ - Pr\Bigg[\boldsymbol{\pi} \xleftarrow{\$} Perm^{\mathcal{T}}(\mathcal{M}) : A^{\boldsymbol{\pi}(\cdot,\cdot),\pi^{-1}(\cdot,\cdot)} \Rightarrow 1\Bigg]\Bigg|. \quad (2.1)$$

We define $\mathbf{Adv}_{\mathbf{E}}^{\pm \widetilde{prp}}(q,\sigma_n)$ by $max_A \mathbf{Adv}_{\mathbf{E}}^{\pm \widetilde{prp}}(A)$ where maximum is taken over all adversaries which makes at most q queries having at most σ_n many blocks. For a computational advantage we define $\mathbf{Adv}_{\mathbf{E}}^{\pm \widetilde{prp}}(q,\sigma_n,t)$ by $max_A \mathbf{Adv}_{\mathbf{E}}^{\pm \widetilde{prp}}(A)$. In addition to the previous restrictions on A, he can run in time at most t.

Chapter 3

Security of XCB

In 2004, McGraw and Fluhrer proposed Tweakable Enciphering Scheme (TES) named as XCB in [11] without providing a proof. Later in 2007, they made changes in original construction and proved security of the updated construction in [12]. Authors claim that the changes were made for the improvement of performance of XCB and make it easier to analyse. Later Chakraborty, Hernandez-Jimenez and Sarkar [2] did a detailed analysis of two versions of XCB as described in [11] and [12]. The study in [2] names the version of XCB in [11] as XCBv1 and the one in [12] as XCBv2, we will also follow the same nomenclature. The analysis in [2] concludes that the security claims regarding XCBv2 as presented in [12] are largely erroneous. XCBv2 is completely insecure for certain types of messages, in particular, there is an easy distinguishing attack on XCBv2 if it is used on messages whose length is not a multiple of the block length n of the underlying block cipher. Though XCBv2 is secure for other messages, the proof and the security bound was shown to be incorrect. In [2] a correct security bound for XCBv2 (message for which it is secure) was derived and also a proof for XCBv1 was provided. That proof was based on the analyses done in [10]. In [10] the security bounds of an authenticated encryption scheme called GCM were analyzed, as GCM and XCB shares almost the same hash function. Hence, the techniques used and analyze GCM in [10] could be adopted to analyze XCB in [2]. Further, analysis of the GCM bound was done in [15]. We use the analyses done in [15] to give an improved security bound on XCBv1 and XCBv2.

In this chapter, we give the improved security bounds on XCBv1 and XCBv2 (with full block). Further, we modify the XCB (MXCB say) and give its security bound. Also, we compare the improved security bound and MXCB with some existing TES mode having parameter of practical value. After that in last section of the chapter, we show some weak keys analysis on XCB.

3.1 Description of XCB

XCB is hash-counter-hash scheme which use hash function and counter mode as the basic building blocks of the scheme. The construction of XCBv1 and XCBv2 are shown in the Figure 3.1 and Figure 3.2 respectively, and encryption algorithms

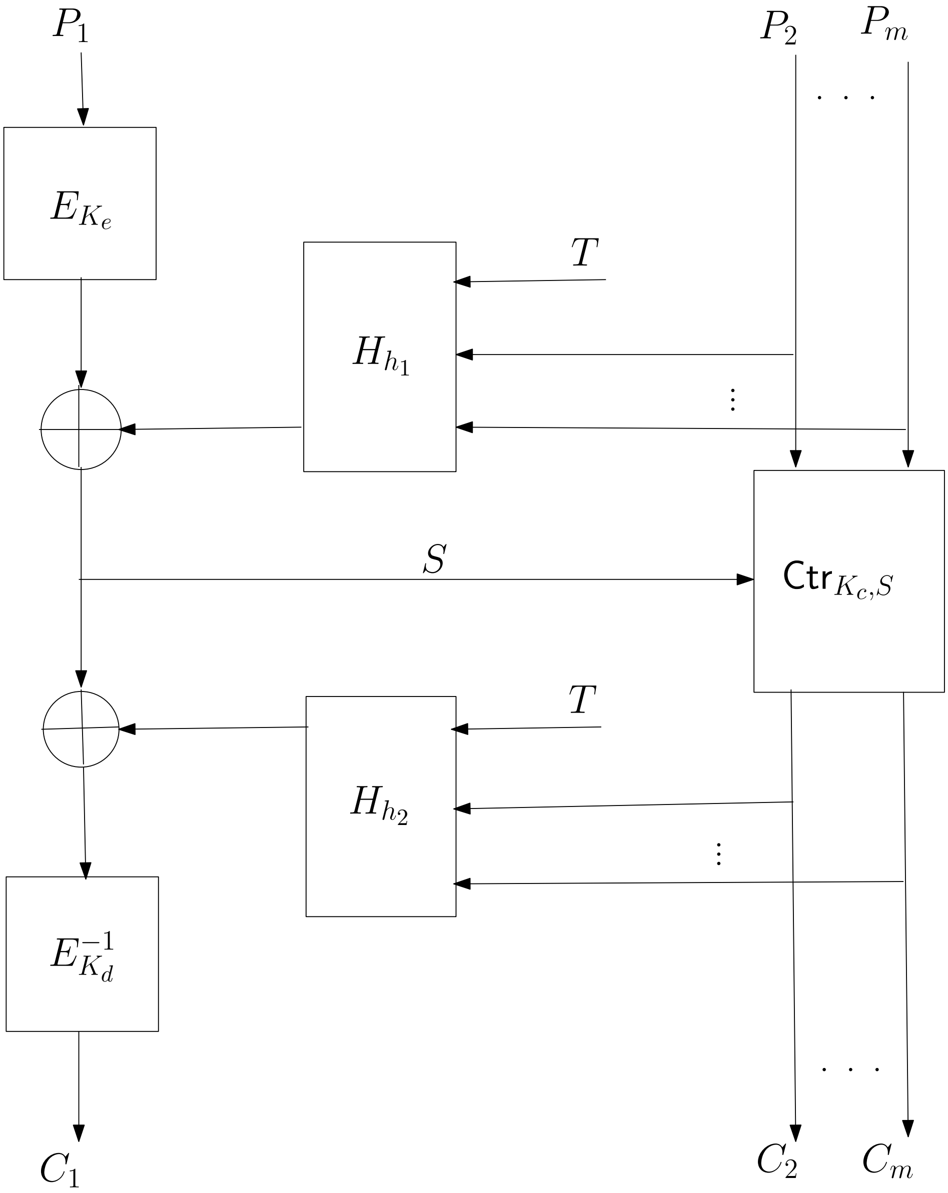

Figure 3.1: Encryption of XCBv1

for XCBv1 and XCBv2 are shown in the Figure 3.3.

Following definition of hash function and Ctr mode are as defined in [2].

Hash function $H : \{0,1\}^n \times \mathcal{X} \times \mathcal{Y} \rightarrow \{0,1\}^n$, where $\mathcal{X}, \mathcal{Y}$ are non-empty subsets of $\{0,1\}^*$. For $T \in \mathcal{Y}$ and $X \in \mathcal{X}$, we write $H_h(X,T)$ instead of $H(h,X,T)$. The hash function H is defined as

$$H_h(X,T) = X_1 h^{m+p+1} \oplus X_2 h^{m+p} \oplus \ldots \oplus \mathsf{pad}(X_m) h^{p+2} \oplus T_1 h^{p+1} \\ \oplus T_2 h^p \oplus \ldots \oplus \mathsf{pad}(T_p) h^2 \oplus (\mathsf{bin}_{\frac{n}{2}}(|X|) \| \mathsf{bin}_{\frac{n}{2}}(|T|)) h, \quad (3.1)$$

where $(X_1, X_2, \ldots, X_m) = parse_n(X)$, $(T_1, T_2, \ldots, T_p) = parse_n(T)$. The pad function is defined as $\mathsf{pad}(X_m) = X_m \| 0^r$ where $r = n - |X_m|$. Thus, $|\mathsf{pad}(X_m)| = n$.

Counter mode: Ctr with key K, counter value S and message $A_1, \ldots, A_m$ is defined as:

$$\mathsf{Ctr}_{K,S}(A_1, \ldots, A_m) = (A_1 \oplus E_K(\mathrm{inc}^0(S)), \ldots, A_m \oplus E_K(\mathrm{inc}^{m-1}(S)).$$

If the last block A_m is incomplete then the quantity $A_m \oplus E_K(\mathrm{inc}^{m-1}(S))$ is replaced by the quantity $A_m \oplus \mathsf{drop}_r(E_K(\mathrm{inc}^{m-1}(S))$ where $r = n - |A_m|$ and $\mathsf{drop}_r(E_K(\mathrm{inc}^{m-1}(S))$ is the first $(n-r)$ bits of $E_K(\mathrm{inc}^{m-1}(S)$. In the definition of Ctr, for a bit string $X \in \{0,1\}^n, \mathrm{inc}(X)$ treats the last significant 32 bits of X as a non-negative integer and increment this value modulo 2^{32} i.e.

$$\mathrm{inc}(X) = \mathrm{msb}_{n-32}(X) \| \mathsf{bin}_{32}(\mathrm{int}(\mathrm{lsb}_{32}(X)) + 1 \mod 2^{32}).$$

For $r \geq 0$, we write $\mathrm{inc}^r(X)$ to denote the r times iterative applications of inc on X. We use the convention that $\mathrm{inc}^0(X) = X$. For this specific structure of inc both XCBv1 and XCBv2 can only be used with block ciphers where the block length $n \geq 32$, which does not amount to a practical constraint.

XCBv1 and XCBv2 has length constraints with respect to plaintext and tweak as $n \leq |P| \leq 2^{39}$ and $0 \leq |T| \leq 2^{39}$ respectively. XCBv2 is specified in the standard IEEE 1619.2 2010. In the standard, AES is fixed as the block cipher and message length is always multiple of 8 bits.

3.2 Differences between XCBv1 and XCBv2

The main difference between XCBv1 with XCBv2 is that later version uses only one hash key whereas the earlier one uses two hash keys which was made to reduce the cost of additional hash keys. There are some other differences which we can list as follows:

1. Keys generating by the same master key are different for XCBv1 and XCBv2.
2. In XCBv1, length of the key is fixed to 128 bits while XCBv2's key length is variable which can be 128, 192 or 256 bits as per requirement.
3. As mentioned earlier, XCBv1 uses two hash keys as compare to XCBv2 which use only one hash key.
4. In XCBv1, first block of the message is encrypted (in line 6) and then XOR with hash function (in line 7) while XCBv2 perform same operation with the last block of the plaintext.
5. Padding is the part of XCBv2 hash function while XCBv1's hash function has no padding.
6. XCBv2 append the string of 0^n before tweak and after message in hash function (in line 108) while XCBv1 is simple hash function without appending any string with tweak and message.
7. In XCBv1, both hash function use the different hash key and same definition of hash function while XCBv2 uses same hash key and different hash function in the scheme (in line 108 and 110).

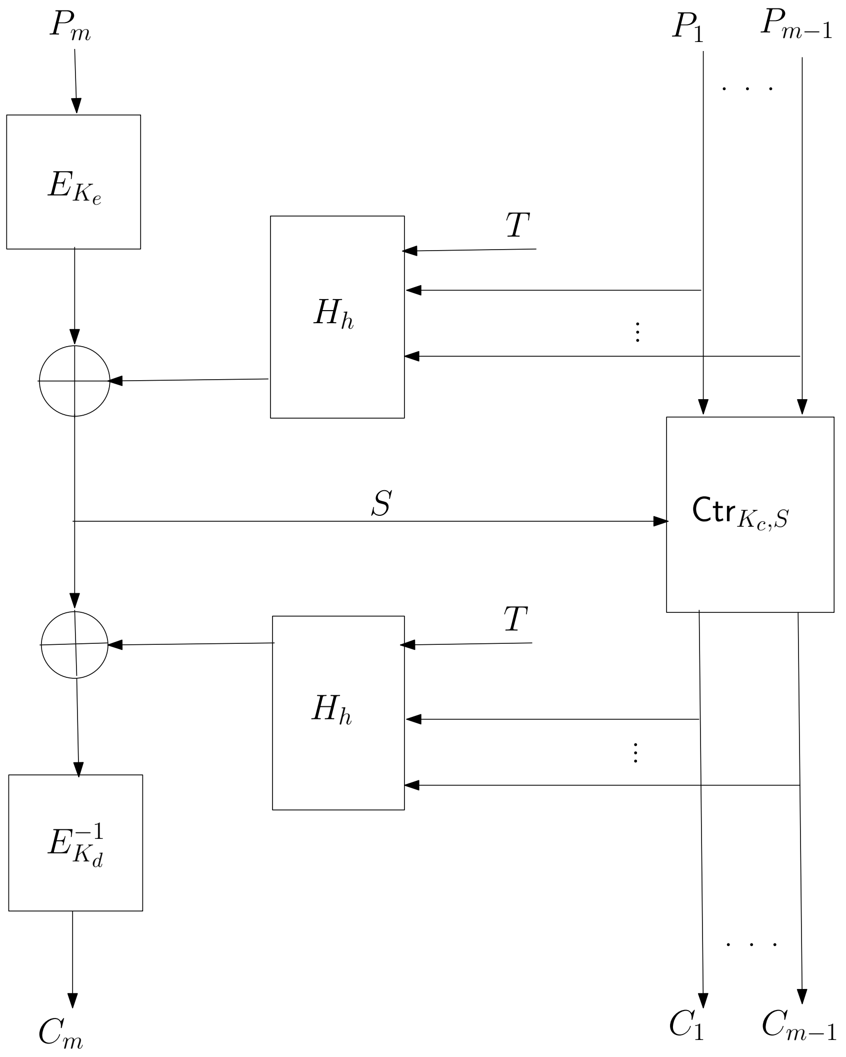

Figure 3.2: Encryption of XCBv2

All keys are supposed to be random by the security of AES.

3.3 Security Claims

In this section, we state both security bounds as claimed in [2] and our updated security bounds for both XCBv1 and XCBv2. We are interested in so called information theoretic security bound, i.e., we only state the bound for the schemes where the block ciphers are replaced by true random permutations. In both versions of XCB three block ciphers with different keys are used. We replace these block ciphers with three independent random permutations on $\{0,1\}^n$, we call the resulting construction as XCBvl[3Perm(n)] where $\mathrm{l} \in \{1,2\}$. For XCBv1 security bound as stated in [2] is:

Theorem 3.3.1. Let A be an arbitrary adversary which queries only with messages/ciphers whose lengths are multiples of n and A asks a total of q queries of

```
Encryption under XCBv1 : $\mathbf{E}_K^T(P)$

$(P_1, \ldots, P_m) \leftarrow \mathrm{parse}_n P$
$h_1 \leftarrow E_K(0^{n-3}\|001)$
$h_2 \leftarrow E_K(0^{n-3}\|011)$
$K_e \leftarrow E_K(0^n)$
$K_d \leftarrow E_K(0^{n-3}\|100)$
$K_c \leftarrow E_K(0^{n-3}\|010)$
$CC \leftarrow E_{K_e}(P_1)$
$S \leftarrow CC \oplus H_{h_1}(P_2\| \ldots \|P_{m-1}\|P_m, T)$
$(C_2, \ldots, C_m) \leftarrow \mathsf{Ctr}_{K_c,S}(P_2, \ldots, P_m)$
$MM \leftarrow S \oplus H_{h_2}(C_2\| \ldots \|C_{m-1}\|C_m, T)$
$C_1 \leftarrow E_{K_d}^{-1}(MM)$
$\mathbf{return}(C_1, C_2, \ldots, C_m)$
```

```
Encryption under XCBv2 : $\mathbf{E}_K^T(P)$

$P_m \leftarrow \mathrm{lsb}_n(P)$
$A \leftarrow \mathrm{msb}_{|P|-n}(P)$
$(P_1, P_2, \ldots, P_{m-2}, P_{m-1}) \leftarrow \mathrm{parse}_n(A)$
$\mathrm{h} \leftarrow E_K(0^n)$
$K_e \leftarrow \mathrm{msb}_{|K|}(E_{|K|}(0^{n-3}\|001)\|E_{|K|}(0^{n-3}\|010))$
$K_d \leftarrow \mathrm{msb}_{|K|}(E_{|K|}(0^{n-3}\|011)\|E_{|K|}(0^{n-3}\|100))$
$K_c \leftarrow \mathrm{msb}_{|K|}(E_{|K|}(0^{n-3}\|101)\|E_{|K|}(0^{n-3}\|110))$
$CC \leftarrow E_{K_e}(P_m)$
$S \leftarrow CC \oplus H_h(0^n\|T, P_1\| \ldots \|P_{m-2}\|\mathsf{pad}(P_{m-1})\|0^n)$
$(C_1, \ldots, C_{m-1}) \leftarrow \mathsf{Ctr}_{K_c,S}(P_1, \ldots, P_{m-2}, P_{m-1})$
$MM \leftarrow S \oplus H_h(T\|0^n, C_1\| \ldots \|\mathsf{pad}(C_{m-1})\|\mathsf{bin}_{\frac{n}{2}}(|T\|0^n|)\|\mathsf{bin}_{\frac{n}{2}}(|C_1\| \ldots \|C_{m-2}\|C_{m-1}|))$
$C_m \leftarrow E_{K_d}^{-1}(MM)$
$\mathbf{return}(C_1, C_2, \ldots, C_m)$
```

Figure 3.3: Encryption using XCBv1 and XCBv2

overall query complexity σ_n where each query is at most ℓ blocks long (each block of n bits). Then,

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{XCBv1[3Perm}(n)]}(A) \leq \frac{(3 + 2^{22})\ell q \sigma_n}{2^n}.$$

For XCBv2 the security bound given in [2] is:

Theorem 3.3.2. Let A be an arbitary adversary which queries only with messages/ciphers whose lengths are multiples of n and A asks a total of q queries of overall query complexity σ_n where each query is at most ℓ blocks long (each block of n bits). Then,

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{XCBv2fb[3Perm}(n)]}(A) \leq \frac{(5 + 2^{22})\ell q \sigma_n}{2^n}.$$

We show that the bound of $\frac{(3+2^{22})\ell q\sigma_n}{2^n}$ and $\frac{(5+2^{22})\ell q\sigma_n}{2^n}$ in both the above theorems can be improved to $\frac{(3+2^5)\ell q\sigma_n}{2^n}$ and $\frac{(5+2^5)\ell q\sigma_n}{2^n}$ for XCBv1 and XCBv2 respectively. Specifically, we prove the following two theorems.
For XCBv1, our security bound is as follows:

Theorem 3.3.3. Consider an arbitrary adversary A which queries only with messages/ciphers whose lengths are multiples of n and A asks a total of q queries of overall query complexity σ_n where each query is at most ℓ blocks long (each block of n bits). Then,

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{XCBv1[3Perm}(n)]}(A) \leq \frac{(3+2^5)\ell q\sigma_n}{2^n}. \tag{3.2}$$

and for XCBv2 our security bound is given as:

Theorem 3.3.4. Consider an arbitrary adversary A which queries only with messages/ciphers whose lengths are multiples of n and A asks a total of q queries of overall query complexity σ_n where each query is at most ℓ blocks long (each block of n bits). Then,

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{XCBv2fb[3Perm}(n)]}(A) \leq \frac{(5+2^5)\ell q\sigma_n}{2^n}. \tag{3.3}$$

The complete proof of these theorems are presented in the following section. The proof heavily uses a technique from [15] where an improved bound on GCM was proved.

3.4 Some useful lemmas

We start with a few useful lemmas which would be used later.
The following Version of the Schwartz-Zippel lemma is from [14].

Lemma 1. Let $\mathbb{F}$ be a field and $p \in \mathbb{F}[x_1, x_2, \ldots, x_r]$ be a r-variate, non-zero polynomial with total degree d. Let S be finite subset of $\mathbb{F}$, and $x_1, x_2, \ldots, x_r$ be selected uniformly at random from S. Then

$$\Pr[p(x_1, x_2, \ldots, x_r) = 0] \leq \frac{d}{|S|}.$$

For $0 \leq r \leq 2^{32}-1$, $\mathbb{Y}_r \overset{def}{=} \{\mathsf{bin}_{32}(\mathrm{int}(Y) + r \mod 2^{32}) \oplus Y | Y \in \{0,1\}^{32}\}$ and $\mathrm{inc}^r(X) = (X \oplus 0^{n-32} \| Y)$ for some $Y \in \mathbb{Y}_r$.
From [15] we define $\mathbb{W}_r \subseteq \{0,1\}^{32}$, for $0 \leq r \leq 2^{32}-1$, as $\mathbb{W}_0 \overset{def}{=} \mathbb{Y}_0$ and $\mathbb{W}_r \overset{def}{=} \mathbb{Y}_r \backslash \bigcup_{i=0}^{r-1} \mathbb{Y}_i$ and $r \geq 1$. We denote cardinality as $w_r \overset{def}{=} \#\mathbb{W}_r$ and $w_{\max} \overset{def}{=} max\{w_r \mid 0 \leq r \leq 2^{32}-1\}$ and it was shown in [15] that $w_{\max} \leq 32$.

Lemma 2. 1. Let $X, Y, X', Y' \in \{0,1\}^*$, such that $(X, Y) \neq (X', Y')$. Let $C, C' \in \{0,1\}^n$ and $h \xleftarrow{\$} \{0,1\}^n, S = C \oplus H_h(X, Y)$, and $S' = C' \oplus H_h(X', Y')$, where $H_h(\cdot)$ is defined in (3.1). Then,

$$\Pr\left[\bigvee_{i=0}^{m^s-2}\left(\mathrm{inc}^i(S) \oplus S' = 0\right)\right] \leq \frac{w_{max}\ell(m^s-1)}{2^n}.$$

2. Let $X, Y, X', Y' \in \{0,1\}^*$, $C, C' \in \{0,1\}^n$ and $h_1, h_2 \xleftarrow{\$} \{0,1\}^n, S = C \oplus H_{h_1}(X,Y)$, and $S' = C' \oplus H_{h_2}(X',Y')$. Then,

$$\Pr\left[\bigvee_{i=0}^{m^s-2}\left(\mathrm{inc}^i(S) \oplus S' = 0\right)\right] \leq \frac{w_{max}\ell(m^s-1)}{2^n}.$$

In both cases ℓ is the $\max\{\ell^s, \ell^{s'}\}$ where ℓ^s and $\ell^{s'}$ are the degrees of the two polynomials S and S' respectively. In the first case probability is taken over the random choice of h, and in the second case it is taken over the random choice of h_1, h_2.

Proof. **Case 1.** The proof uses technique given in [15].

Let's figure out the upper bound on collision with two distinct pair (X,Y) and (X',Y') with $0 \leq r < r' \leq 2^{32}-1$ i.e.

$$\Pr[(\mathrm{inc}^r(S) = S') \vee (\mathrm{inc}^{r'}(S) = S')]$$

Also we obtain the following upper sum bound

$$\begin{aligned}\Pr[(\mathrm{inc}^r(S) = S') \vee (\mathrm{inc}^{r'}(S) = S')] \leq & \sum_{Y \in \mathbb{Y}_r} \Pr[S \oplus (0^{n-32}\|Y) = S'] \\ & + \sum_{Y' \in \mathbb{Y}_{r'}} \Pr[S \oplus (0^{n-32}\|Y') = S'] \quad (3.4)\end{aligned}$$

since $\mathrm{inc}^r(X) = (X \oplus 0^{n-32}\|Y)$ for some $Y \in \mathbb{Y}_r$.

Claim: For $0 \leq r' < r \leq 2^{32}-1$, and $Y \in \{0,1\}^{32}$ such that $Y \in \mathbb{Y}_r$ and $Y \in \mathbb{Y}_{r'}$. Then there does not exist $X \in \{0,1\}^n$ such that $\mathrm{inc}^r(X) = X \oplus (0^{n-32}\|Y)$ and $\mathrm{inc}^{r'}(X) = X \oplus (0^{n-32}\|Y)$ simultaneously.

Proof: Proof by contradiction. Let's such r and r' exist, without loss of generality $r' < r$. Therefore, $\mathrm{inc}^r(X) = \mathrm{inc}^{r'}(X)$ which imply $\mathrm{inc}^{r-r'}(X) = X$ which is not possible. Hence, r and r' are not distinct.

So, we can conclude with an upper bound i.e.

$$\begin{aligned}\Pr[(\mathrm{inc}^r(S) = S') \vee (\mathrm{inc}^{r'}(S) = S')] \leq & \sum_{Y \in \mathbb{Y}_r} \Pr[S \oplus (0^{n-32}\|Y) = S'] \\ & + \sum_{Y' \in \mathbb{Y}_{r'} \setminus \mathbb{Y}_r} \Pr[S \oplus (0^{n-32}\|Y') = S']. \quad (3.5)\end{aligned}$$

After generalisation for m^s-2, we get

$$\Pr\left[\bigvee_{i=0}^{m^s-2}\left(\mathrm{inc}^i(S) \oplus S' = 0\right)\right] \leq \sum_{0 \leq i \leq m^s-2} \sum_{Y \in \mathbb{Y}_i \setminus \bigcup_{j=0}^{i-1} \mathbb{Y}_j} \Pr[S \oplus (0^{n-32}\|Y) = S']. \tag{3.6}$$

Now, by using Schwartz-Zippel Lemma, we know

$$\Pr[S \oplus (0^{n-32} \| Y) = S'] \leq \frac{\ell}{2^n}.$$

Therefore,

$$\sum_{0 \leq i \leq m^s - 2} \sum_{Y \in \mathbb{Y}_i \setminus \bigcup_{j=0}^{i-1} \mathbb{Y}_j} \Pr[S \oplus (0^{n-32} \| Y) = S'] \leq \sum_{0 \leq i \leq m^s - 2} \frac{w_{max} \ell}{2^n}. \tag{3.7}$$

From equations (3.6) and (3.7), we can conclude

$$\Pr\left[\bigvee_{i=0}^{m^s-2} \left(\mathrm{inc}^i(S) \oplus S' = 0\right)\right] \leq \frac{w_{max}\ell(m^s - 1)}{2^n}.$$

□

Case 2. h_1, h_2 are selected independently and uniformly at random from $\{0,1\}^n$, and $S = H_{h_1}(X, Y)$ and $S' = H_{h_2}(X', Y')$. According to the definition of $H(\cdot)$, both S and S' are non-zero polynomials. So, the result will be same as of Case 1 by Lemma 1.

3.5 Repairing XCB Security proofs

Proof of the theorem is heavily based on the proof given in [2]. As stated earlier, in place of three block cipher given in line 6,8 and 10 of XCBv1, we use the three different permutation on n-bit string. The encryption and decryption of the scheme of XCBv1[3Perm(n)] by $\mathbf{E}_{\tilde{\pi},\tilde{h}}$ and $\mathbf{D}_{\tilde{\pi},\tilde{h}}$ respectively, where $\tilde{\pi} = (\pi_1, \pi_2, \pi_3)$ and π_1, π_2, π_3 are three permutation selected uniformly and independently and $\tilde{h} = (h_1, h_2)$ where h_1 and h_2 are two hash keys selected uniformly and independently to other variables.

For proving (3.2), we need to consider an adversary's advantage in distinguishing XCBv1[3Perm(n)] from an oracle which simply returns random bit strings. This advantage defined in following way :

$$\mathbf{Adv}^{\pm rnd}_{\mathrm{XCBv1[3Perm}(n)]}(A) = \left| \Pr\left[\tilde{\pi} \xleftarrow{\$} \mathrm{3Perm}(n), \tilde{h} \xleftarrow{\$} \{0,1\}^n : A^{\mathbf{E}_{\tilde{\pi},\tilde{h}}, \mathbf{D}_{\tilde{\pi},\tilde{h}}} \Rightarrow 1\right] - \Pr\left[A^{\$(\cdot,\cdot),\$(\cdot,\cdot)} \Rightarrow 1\right] \right|, \tag{3.8}$$

where $\$(\cdot, M)$ or $\$(\cdot, C)$ returns independently distributed random bits of length $|M|$ or $|C|$ respectively. The basic idea of proving (3.2) is as follows.

$$\begin{aligned}
\mathbf{Adv}^{\pm\widetilde{prp}}_{\text{XCBv1[3Perm}(n)]}(A) &= \left(\Pr\left[\tilde{\pi} \xleftarrow{\$} \text{3Perm}(n), \tilde{h} \xleftarrow{\$} \{0,1\}^n : A^{\mathbf{E}_{\tilde{\pi},\tilde{h}},\mathbf{D}_{\tilde{\pi},\tilde{h}}} \Rightarrow 1\right]\right. \\
&\quad \left. - \Pr\left[\pi \xleftarrow{\$} \text{Perm}^{\mathcal{T}}(\mathcal{M}) : A^{\pi(\cdot,\cdot),\pi^{-1}(\cdot,\cdot)} \Rightarrow 1\right]\right) \\
&= \left(\Pr\left[\tilde{\pi} \xleftarrow{\$} \text{3Perm}(n), \tilde{h} \xleftarrow{\$} \{0,1\}^n : A^{\mathbf{E}_{\tilde{\pi},\tilde{h}},\mathbf{D}_{\tilde{\pi},\tilde{h}}} \Rightarrow 1\right]\right. \\
&\quad \left. - \Pr\left[A^{\$(\cdot,\cdot),\$(\cdot,\cdot)} \Rightarrow 1\right]\right) \\
&\quad + \left(\Pr\left[A^{\$(\cdot,\cdot),\$(\cdot,\cdot)} \Rightarrow 1\right]\right. \\
&\quad \left. - \Pr\left[\pi \xleftarrow{\$} Perm^{\mathcal{T}}(\mathcal{M}) : A^{\pi(\cdot,\cdot),\pi^{-1}(\cdot,\cdot)} \Rightarrow 1\right]\right) \\
&\leq \mathbf{Adv}^{\pm rnd}_{\text{XCBv1[3Perm}(n)]}(A) + \binom{q}{2}\frac{1}{2^n}. \qquad (3.9)
\end{aligned}$$

where q is the number of queries made by the adversary. For proof of the last inequality see [8]. Thus, the main task of the proof now reduces to obtaining an upper bound on $\mathbf{Adv}^{\pm rnd}_{\text{XCBv1[3Perm}(n)]}(A)$. We prove this by the usual techniques of sequence of games which are in games XCB1 (Figure 3.4), RAND1 (Figure 3.4) and RAND2 (Figure 3.5).

Game XCB1 is same as in Figure 3.4 except the algorithm of XCBv1 uses three independent random permutation π_1, π_2, π_3 instead of the block cipher implementation. We denote this as follows:

$$\Pr\left[A^{\mathbf{E}_{\tilde{\pi},h},\mathbf{D}_{\tilde{\pi},h}} \Rightarrow 1\right] = \Pr\left[A^{\text{XCB1}} \Rightarrow 1\right]. \qquad (3.10)$$

Game RAND1 is also described in Figure 3.4 with the boxed entries removed. In this game it is not guaranteed that $\pi_i (i = 1, 2, 3)$ are permutation as though we do the consistency checks but we don't reset the values of Y (in Ch-π_i) and X (in Ch-π_i^{-1}). Thus, the games XCB1 and RAND1 are identical apart from what happens when the bad flag is set. By the fundamental lemma of game-ploting or difference lemma, we have

$$\left|\Pr\left[A^{\text{XCB1}} \Rightarrow 1\right] - \Pr\left[A^{RAND1} \Rightarrow 1\right]\right| \leq \Pr\left[A^{RAND1} \text{ set } \mathsf{bad}\right]. \qquad (3.11)$$

Here, we see RAND1 gives the random string in response of encryption and decryption queries. So,

$$\Pr\left[A^{RAND1} \Rightarrow 1\right] = \Pr\left[A^{(\cdot,\cdot),(\cdot,\cdot)} \Rightarrow 1\right]. \qquad (3.12)$$

```
Subroutine Ch-$\pi_i(X)(i = 1, 2, 3)$
$Y \xleftarrow{\$} \{0,1\}^n$; if $Y \in Range_i$ then bad ← true; $\boxed{Y \xleftarrow{\$} \overline{Range_i}}$; end if
if $X \in Domain_i$ then bad ← true; $\boxed{Y \leftarrow \pi_i(X)}$; end if
$\pi_i(X) \leftarrow Y; Domain_i \leftarrow Domain_i \cup \{X\}$; $Range_i \leftarrow Range_i \cup (Y)$; return (Y);

Subroutine Ch-$\pi_i^{-1}(Y)$
$X \xleftarrow{\$} \{0,1\}^n$; if $X \in Domain_i$ then bad ← true; $\boxed{X \xleftarrow{\$} \overline{Domain_i}}$; end if
if $Y \in Range_i$ then bad ← true; $\boxed{X \leftarrow \pi_i^{-1}(Y)}$; end if
$\pi_i(X) \leftarrow Y; Domain_i \leftarrow Domain_i \cup \{X\}$; $Range_i \leftarrow Range_i \cup (Y)$; return (X);

Initialization:
for all $X \in \{0,1\}^n \pi_i(X)$ = undef end for
bad = false
$h_1, h_2 \xleftarrow{\$} \{0,1\}^n$
```

Respond to the s^{th} query as follows:

```
Encipher query: $Enc(T^s; P_1^s, P_2^s, \ldots, P_{m_s}^s)$

if $P_1^s = P_1^{s'}$ for $s' < s$ then
    $CC^s \leftarrow CC^{s'}$
else
    $CC^s \leftarrow$ Ch-$\pi_1(P_1^s)$
end if
$S^s \leftarrow CC^s \oplus H_{h_1}(P_2^s \| \ldots \| P_{m^s}^s, T^s)$
for $i = 1$ to $m^s$ -1
    $Z_i^s \leftarrow$ Ch-$\pi_2(\mathrm{inc}^i(S^s))$
    $C_{i+1}^s \leftarrow P_{i+1}^s \oplus Z_i^s$
end for
$MM^s \leftarrow S^s \oplus H_{h_2}(C_2^s \| \ldots \| C_{m^s}^s, T^s)$
$C_1^s \leftarrow Ch - \pi_3^{-1}(MM^s)$
return $(C_1^s, C_2^s, \ldots, C_{m^s}^s)$
```

```
Decipher query: $Dec(T^s; C_1^s, C_2^s, \ldots, C_{m^s}^s)$

if $C_1^s = C_1^{s'}$ for s' < s then
    $MM^s \leftarrow MM^{s'}$
else
    $MM^s \leftarrow$ Ch-$\pi_3(C_{m^s}^s)$
end if
$S^s \leftarrow MM^s \oplus H_{h_2}(C_2^s \| \ldots \| C_{m^s}^s, T^s)$
for $i = 1$ to $m^s$-1
    $C_{i+1}^s \leftarrow P_{i+1}^s \oplus Z_i^s$
    $P_{i+1}^s \leftarrow C_{i+1}^s \oplus Z_i^s$
end for
$CC^s \leftarrow S^s \oplus H_{h_1}(P_2^s \| \ldots \| P_{m^s}^s, T^s)$
$P_1^s \leftarrow Ch - \pi_1^{-1}(CC^s)$
return $(P_1^s, P_2^s, \ldots, P_{m^s}^s)$
```

Figure 3.4: Games XCB1 and RAND1 : In RAND1 the boxed entries are removed.

```
Respond to the $s^{th}$ adversary query as follows:
ENCIPHER QUERY $\mathbf{Enc}(T^s; P^s)$
10. $(P_1^s, P_2^s, \ldots, P_{m^s}^s) \leftarrow \mathrm{parse}_n(P^s)$
11. $ty^s = \mathbf{Enc}$
12. $C_1^s \| C_2^s \| \ldots \| C_{m^s-1}^s \| D_{m^s}^s \xleftarrow{\$} \{0,1\}^{nm^s}$
13. $C_{m^s}^s \leftarrow \mathsf{drop}_{n-r^s}(D_{m^s}^s)$
14. return $C_1^s \| C_2^s \| \ldots \| C_{m^s}^s$
DECIPHER QUERY $\mathbf{Dec}(T^s; C^s)$
20. $(C_1^s, C_2^s, \ldots, C_{m^s-1}^s, C_{m^s}^s) \leftarrow \mathrm{parse}_n(C^s)$
21. $ty^s = \mathbf{Dec}$
22. $P_1^s \| P_2^s \| \ldots \| P_{m^s-1}^s \| V_{m^s}^s \xleftarrow{\$} \{0,1\}^{nm^s}$
23. $P_{m^s}^s \leftarrow \mathsf{drop}_{n-r^s}(V_{m^s})$
24. return $P_1^s \| P_2^s \| \ldots \| P_{m^s}^s$
Finalization:
001. $h_1 \xleftarrow{\$} \{0,1\}^n$;      002. $h_2 \xleftarrow{\$} \{0,1\}^n$
for s = 1 to q
    if $ty^s = \mathbf{Enc}$ then
    101. if $P_1^s = P_1^{s'}$ for s' < s then
    102.     $CC^s \leftarrow CC^{s'}$
    103. else
    104.     $CC^s \xleftarrow{\$} \{0,1\}^n$
    105.     $\mathcal{D}_1 \leftarrow \mathcal{D}_1 \cup \{P_1^s\}$
    106.     $\mathcal{R}_1 \leftarrow \mathcal{R}_1 \cup \{CC^s\}$
    107. end if
    108. $S^s \leftarrow CC^s \oplus H_{h_1}(P_2^s \| \ldots \| P_{m^s}^s, T^s)$
    109. $MM^s \leftarrow S^s \oplus H_{h_2}(C_2^s \| \ldots \| C_{m^s}^s, T^s)$
    110. $\mathcal{D}_3 \leftarrow \mathcal{D}_3 \cup \{C_1^s\}$
    111. $\mathcal{R}_3 \leftarrow \mathcal{R}_3 \cup \{MM^s\}$
    112. for $i = 0$ to $m^s - 3$,
    113.     $Y_i^s \leftarrow C_{i+2}^s \oplus P_{i+2}^s$
    114.     $\mathcal{D}_2 \leftarrow \mathcal{D}_2 \cup \{\mathrm{inc}^i(S^s)\}$
    115.     $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup \{Y_i^s\}$
    116. end for
    117. $Y_{m^s-2}^s \leftarrow \mathsf{pad}(P_{m^s}^s) \oplus D_{m^s}^s$
    118. $\mathcal{D}_2 \leftarrow \mathcal{D}_2 \cup \{\mathrm{inc}^{m^s-2}(S^s)\}$
    119. $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup \{Y_{m^s-2}^s\}$
    else if $ty^s = \mathbf{Dec}$:
    201. if $C_1^s = C_1^{s'}$ for s' < s then
    202. $MM^s \leftarrow MM^{s'}$
    203. else
    204.     $MM^s \xleftarrow{\$} \{0,1\}^n$
    205.     $\mathcal{D}_3 \leftarrow \mathcal{D}_3 \cup \{C_1^s\}$
    206.     $\mathcal{R}_3 \leftarrow \mathcal{R}_3 \cup \{MM^s\}$
    207. end if
    208. $S^s \leftarrow MM^s \oplus H_{h_2}(C_2^s \| \ldots \| C_{m^s}^s, T^s)$
    209. $CC^s \leftarrow S^s \oplus H_{h_1}(P_2^s \| \ldots \| P_{m^s}^s, T^s)$
    210. $\mathcal{D}_1 \leftarrow \mathcal{D}_1 \cup \{P_1^s\}$
    211. $\mathcal{R}_1 \leftarrow \mathcal{R}_1 \cup \{CC^s\}$
    212. for $i = 0$ to $m^s - 3$,
    213.     $Y_i^s \leftarrow C_{i+2}^s \oplus P_{i+2}^s$
    214.     $\mathcal{D}_2 \leftarrow \mathcal{D}_2 \cup \{\mathrm{inc}^i(S^s)\}$
    215.     $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup \{Y_i^s\}$
    216. end for
    217. $Y_{m^s-2}^s \leftarrow \mathsf{pad}(C_{m^s}^s) \oplus V_{m^s}^s$
    218. $\mathcal{D}_2 \leftarrow \mathcal{D}_2 \cup \{\mathrm{inc}^{m^s-2}(S^s)\}$
    219. $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup \{Y_{m^s-2}^s\}$
    end if
end for
SECOND PHASE
bad = false;
if(some value occurs more than once in $\mathcal{D}_i, i = 1,2,3$)then bad = true end if;
if(some value occurs more than once in $\mathcal{R}_i, i = 1,2,3$)then bad = true end if.
```

Figure 3.5: Game RAND2 for XCBv1

Now, by using the definition

$$\begin{aligned}\mathbf{Adv}^{\pm rnd}_{\text{XCBv1[3Perm}(n)]}(A) &= \Bigg|\Pr\left[A^{\mathbf{E}_{\pi_1,\pi_2,\pi_3},\mathbf{D}_{\pi_1,\pi_2,\pi_3}} \Rightarrow 1\right] \\ &\quad - \Pr\left[A^{(\cdot,\cdot),(\cdot,\cdot)} \Rightarrow 1\right]\Bigg| \\ &= \Bigg|\Pr\left[A^{\text{XCB1}} \Rightarrow 1\right] \\ &\quad - \Pr[A^{RAND1} \Rightarrow 1\Bigg]\Bigg| \\ &\leq \Pr\left[A^{RAND1} \text{ set } \mathsf{bad}\right]. \end{aligned} \tag{3.13}$$

Game RAND2 is slightly different from RAND1, in this permutation is not maintained, just a random string of appropriate length in response of an encryption/decryption query is returned. In the finalisation step of game, the internal variable are adjusted and the appropriate variables are inserted in the multi sets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ and $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$. If collision occurs in these multi sets then the bad flag is set.

Games RAND1 and RAND2 are indistinguishable to the adversary as both returns the random strings in response to queries. And also for both the cases, probability for which RAND1 and RAND2 have bad flag set is same. Therefore, we can write:

$$\Pr[A^{RAND1} \text{ set } \mathsf{bad}] = \Pr[A^{RAND2} \text{ set } \mathsf{bad}]. \tag{3.14}$$

Thus, from equations (3.13) and (3.14)

$$\mathbf{Adv}^{\pm rnd}_{\text{XCBv1[3Perm}(n)]}(A) \leq \Pr[A^{RAND2} \text{ set } \mathsf{bad}]. \tag{3.15}$$

So, our goal is to bound $\Pr[A^{RAND2}$ set $\mathsf{bad}]$. If there is a collision in these multi sets in Game RAND2 then the bad flag is set. So if COLLD_i and COLLR_i denote the events of a collision in $\mathcal{D}_i$ and $\mathcal{R}_i$ respectively then we have

$$\Pr[A^{RAND2} \text{ set } \mathsf{bad}] \leq \sum_{1\leq i\leq 3} (\Pr[\mathsf{COLLR}_i] + \Pr[\mathsf{COLLD}_i]). \tag{3.16}$$

In the rest of the section we analyze the collision probabilities in the sets $\mathcal{D}_i$ and $\mathcal{R}_i$. After q queries of the adversary where the s^{th} query has m^s blocks of plaintext or ciphertext and t^s blocks of tweak, then the sets $\mathcal{D}_i$ and $\mathcal{R}_i$ can be written as follows:

$\mathcal{D}_1 = \{P_1^s : 1 \leq s \leq q\}$,
$\mathcal{D}_2 = \bigcup_{s=1}^{q}\{inc^j(S^s) : 0 \leq j \leq m^s - 2\}$,
$\mathcal{D}_3 = \{C_1^s : 1 \leq s \leq q\}$,

$\mathcal{R}_1 = \{CC^s : 1 \leq s \leq q\}$,

$\mathcal{R}_2 = \bigcup_{s=1}^{q}\{Y_j^s = C_{j+2}^s \oplus P_{j+2}^s : 0 \le j \le m^s - 3\}$,
$\mathcal{R}_3 = \{MM^s : 1 \le s \le q\}$.

Following are the points which will help in the analysis:

1. For the s^{th} query $ty^s \in \{enc, dec\}$ will denote whether the query is an encryption or a decryption query.
2. In each query, the adversary specifies a tweak T^s, we consider $t^s = \lceil (|T^s|/n) \rceil$. Thus, for any s, H_{h_1} and H_{h_2} in line 108 and 109 respectively (H_{h_1} and H_{h_2} in line 209 and 208 respectively for decryption) for encryption of game RAND2 has degree at most $m^s + t^s$. We denote $\sigma_n = \sum_s t^s + \sum_s m^s$. We denote $\ell = \max\{m^s + t^s, m^{s'} + t^{s'}\}$.
3. In game RAND2 the hash key h_1 and h_2 are selected uniformly at random from $\{0,1\}^n$.
4. For an encryption query, the response received by A is $(C_1^s, C_2^s, \ldots, C_{m^s}^s)$ and for a decryption query the response received is $(P_1^s, P_2^s, \ldots, P_{m^s}^s)$. Both these responses are uniformly distributed and independent of other variables.

In the following claims we bound the required collision probabilities.

Claim 1. $\Pr[\mathsf{COLLD}_1] \le \binom{q}{2}/2^n$.

Proof. In case of encryption i.e. $ty^s = ty^{s'} = enc$ then $\Pr[P_1^s = P_1^{s'}] = 0$ because of the condition in line 101 of RAND2. In case of at least one decryption. Without loss of generality, if $ty^s = dec$, then P_1^s is a uniform n-bit string, hence $\Pr[P_1^s = P_1^{s'}] = 1/2^n$.
For q queries, $\Pr[\mathsf{COLLD}_1] \le \binom{q}{2}/2^n$. □

Claim 2. $\Pr[\mathsf{COLLD}_2] \le (\ell q \sigma_n) w_{max}/2^n$.

Proof. $\mathcal{D}_2 = Z_1 \cup Z_2 \cup \ldots \cup Z_q$, where
$Z_s = \{\mathrm{inc}^j(S^s) : 0 \le j \le m^s - 2\}$, for $1 \le s \le q$ and

$$S^s = \begin{cases} CC^s \oplus H_{h_1}(P_2^s \| \ldots \| P_{m^s}^s, T^s) & \text{if } ty^s = enc, \\ MM^s \oplus H_{h_2}(C_2^s \| \ldots \| C_{m^s}^s, T^s) & \text{if } ty^s = dec. \end{cases}$$

As Z_s contains distinct elements, if $\forall x_1, x_2 \in Z_s$ then $\Pr[x_1 = x_2] = 0$. We need to bound collision between $x1$ and $x2$ when these are in different set then without loss of generality $x_1 \in Z_s$ and $x_2 \in Z_{s'}$, for $s \ne s'$. For $s \ne s'$, we define $\mathsf{COLL}(Z_s, Z_{s'})$ as the event that at least one element of Z_s collide with one element of $Z_{s'}$. Hence, we have

$$\Pr[\mathsf{COLLD}_2] \le \sum_{1 \le s < s' \le q} \Pr[\mathsf{COLL}(Z_s, Z_{s'})]. \tag{3.17}$$

We also have

$$\mathrm{inc}^{j}(S^{s}) \oplus \mathrm{inc}^{j'}(S^{s'}) = \begin{cases} \mathrm{inc}^{j-j'}(S^{s}) \oplus S^{s'} & \text{if } j \geq j'. \\ S^{s} \oplus \mathrm{inc}^{j'-j}(S^{s'}) & \text{if } j < j'. \end{cases}$$

Now we define the following events

$$U_i \equiv \mathrm{inc}^{i}(S^{s}) \oplus S^{s'} = 0, \quad \text{for } 0 \leq i \leq m^{s} - 2, \tag{3.18}$$

$$V_i \equiv S^{s} \oplus \mathrm{inc}^{i}(S^{s'}) = 0 \quad \text{for } 0 \leq i \leq m^{s'} - 2. \tag{3.19}$$

Hence,

$$\mathsf{COLL}(Z_s, Z_{s'}) = \left(\bigvee_{i=0}^{m^{s}-2} U_i\right) \bigvee \left(\bigvee_{i=0}^{m^{s'}-2} V_i\right). \tag{3.20}$$

Now we will bound $\Pr[U_i]$. We assume $s < s'$. For computing $\Pr[\mathrm{inc}^{i}(S^{s}) \oplus S^{s'} = 0]$, In case of encryption of both message, where $P_1^{s} \neq P_1^{s'}$. CC^{s} is chosen randomly hence for any i

$$\Pr[\mathrm{inc}^{i}(S^{s}) \oplus S^{s'} = 0] = \frac{1}{2^{n}}. \tag{3.21}$$

Similarly, in case of decryption of both the message where MM^{s} is chosen randomly hence the same probability of (3.21) holds.

If one of them is decryption and other one is encryption. Without loss of generality, if $ty^{s} = dec$ then both S^{s} and $S^{s'}$ are polynomials of h_1 or h_2 of degree at most $\ell = \max\{m^{s} + t^{s}, m^{s'} + t^{s'}\}$ Then using Lemma 2, we have

$$\Pr\left[\bigvee_{i=0}^{m^{s}-2} U_i\right] \leq \frac{w_{max}\ell(m^{s} - 1)}{2^{n}}. \tag{3.22}$$

Similarly, we have

$$\Pr\left[\bigvee_{i=0}^{m^{s}-2} V_i\right] \leq \frac{w_{max}\ell(m^{s} - 1)}{2^{n}}. \tag{3.23}$$

Thus, using equations (3.22) and (3.23), we have

$$\Pr[\mathsf{COLL}(Z_s, Z_{s'})] \leq \frac{w_{max}(m^{s} + m^{s'} - 2)\ell}{2^{n}}. \tag{3.24}$$

Using equations (3.24) and (3.17) we have

$$\begin{aligned} \Pr[\mathsf{COLLD}_2] &\leq \sum_{1 \leq s < s' \leq q} \frac{w_{max}(m^{s} + m^{s'} - 2)\ell}{2^{n}} \\ &\leq \frac{\ell w_{max}}{2^{n}} \sum_{1 \leq s < s' \leq q} (m^{s} + m^{s'}) \\ &\leq \frac{\ell w_{max} q \sigma_n}{2^{n}}. \end{aligned}$$

□

Claim 3. $\Pr[\mathsf{COLLD}_3] \leq \binom{q}{2}/2^n$.
The proof is similar to the proof of the Claim 1.

Claim 4. $\Pr[\mathsf{COLLR}_1] \leq \frac{(q-1)\sigma_n}{2^n}$.

Proof. In case $P_1^s \neq P_1^{s'}$ and $ty^s = ty^{s'} = enc$ then CC^s and $CC^{s'}$ are selected uniformly at random from $\{0,1\}^n$. Then, $\Pr[CC^s = CC^{s'}] = 1/2^n$.
In case If $ty^s = dec$, then

$$CC^s = MM^s \oplus H_{h_2}(R^s, T^s) \oplus H_{h_1}(Q^s, T^s),$$

where $Q^s = P_2^s \| P_3^s \| \ldots \| P_{m^s}^s$ and $R^s = C_2^s \| C_3^s \| \ldots \| C_{m^s}^s$.
Now, we have the following two cases to solve:

Case I: When $ty^{s'} = enc$ then CC^s is selected uniformly and independently from $\{0,1\}^n$, then $\Pr[CC^s = CC^{s'}] \leq 1/2^n$.

Case II: When $ty^{s'} = dec$ and $MM^s = MM^{s'}$. In this case we have

$$CC^s \oplus CC^{s'} = \big[H_{h_2}(R^s, T^s) \oplus H_{h_2}(R^{s'}, T^{s'})\big] \oplus \big[H_{h_1}(Q^s, T^s) \oplus H_{h_1}(Q^{s'}, T^{s'})\big].$$

Let

$$\begin{aligned} H_2^{s,s'} &= H_{h_2}(R^s, T^s) \oplus H_{h_2}(R^{s'}, T^{s'}), \\ H_1^{s,s'} &= H_{h_1}(Q^s, T^s) \oplus H_{h_1}(Q^{s'}, T^{s'}). \end{aligned}$$

Note that $H_1^{s,s'} \oplus H_2^{s,s'}$ is non-zero bivariabe polynomial on h_1, h_2 with (total) degree ℓ. Hence, from Schwartz-Zippel Lemma

$$\Pr[CC^s \oplus CC^{s'} = 0] \leq \frac{\ell}{2^n}.$$

Therefore, we have

$$\begin{aligned} \Pr[\mathsf{COLLR}_1] &\leq \sum_{1 \leq s \leq s' \leq q} \frac{\ell}{2^n} \\ &\leq \frac{(q-1)\sigma_n}{2^n}. \end{aligned} \tag{3.25}$$

□

Claim 5. $\Pr[\mathsf{COLLR}_2] \leq \binom{\sum_s m^s - q}{2}/2^n$.

Proof. From the game RAND2, we have for $1 \leq s \leq q$, $Y_j^s = C_{j+2}^s \oplus P_{j+2}^s$, where $0 \leq j \leq m^s - 3$, and

$$Y_{m^s-2}^s = \begin{cases} \mathsf{pad}(P_{m^s}^s) \oplus D_{m^s}^s & \text{if } s = enc, \\ \mathsf{pad}(C_{m^s}^s) \oplus V_{m^s}^s & \text{if } s = dec. \end{cases}$$

Hence, there are $\sum_s m^s - q$ uniformly and independently generated n-bit strings. Hence, the Claim follows. □

Claim 6. $\Pr[\mathsf{COLLR}_3] \leq \frac{(q-1)\sigma_n}{2^n}$.
Proof of the Claim is similar to the Claim 4.

Now by using Claim 1 to 6 and equation 3.13, we get :

$$\begin{aligned}\mathbf{Adv}^{\pm rnd}_{\mathrm{XCBv1[3Perm}(n)]}(A) &\leq \frac{2}{2^n}\binom{q}{2} + \frac{\ell q \sigma_n w_{max}}{2^n} + \frac{1}{2^n}\binom{\sum_{s=1}^{q} m^s - q}{2} + \frac{2(q-1)\sigma_n}{2^n} \\ &\leq \frac{2\sigma_n^2}{2^n} + \frac{\ell q \sigma_n w_{max}}{2^n}. \qquad (3.26)\end{aligned}$$

By using above bound, equation (3.9) and the fact $w_{max} \leq 2^5$ as stated in [15]. We get the bound of XCBv1 as stated in Theorem 3.2 i.e.

$$\mathbf{Adv}^{\pm \widetilde{prp}}_{\mathrm{XCBv1[3Perm}(n)]}(A) \leq \frac{(3+2^5)\ell q \sigma_n}{2^n}.$$

Similarly, we can prove the bound for XCBv2 as stated in Theorem 3.3.

3.6 Security of MXCB

In Claim 2, if we change the Ctr mode used in XCBv1 to

$$\mathsf{Ctr}_{K,S}(P_1,\ldots,P_m) = (P_1 \oplus E_K(S \oplus 1), \ldots, P_m \oplus E_K(S \oplus m)),$$

where S is the counter and K is the key, let's say this is "new counter mode". Thus, we have a modified XCBv1, say, MXCBv1. Then MXCBv1's security bound will be different from the original due to "new counter mode". Proof of the security bound of this MXCBv1 will be similar to the previous, but we notice the change in the security bound due to "new counter mode". All the games are similar as we have shown, in the original XCBv1 we only replace Counter mode to "new Counter mode". The collision probability of all the multi sets will be same except $\mathcal{D}_2$. Let us rename COLLD_2 as $\mathsf{COLLD}_{\mathsf{mod}}$ which we calculate as follows:

Claim 7. $\Pr[\mathsf{COLLD}_{\mathsf{mod}}] \leq \binom{\sigma_n - q}{2}\frac{1}{2^n}$.

Proof. $\mathcal{D}_2 = Z_1 \cup Z_2 \cup \ldots \cup Z_q$, where
$Z_s = \{S^s \oplus \mathsf{bin}_n(j) : 1 \leq j \leq m^s - 1\}$, for $1 \leq s \leq q$ and

$$S^s = \begin{cases} CC^s \oplus H_{h_1}(P_2^s \| \ldots \| P_{m^s}^s, T^s) & \text{if } ty^s = enc, \\ MM^s \oplus H_{h_2}(C_2^s \| \ldots \| C_{m^s}^s, T^s) & \text{if } ty^s = dec. \end{cases}$$

It is easy to see for $x_1, x_2 \in Z_s$ then $\Pr[x_1 = x_2] = 0$. So our main task is to figure out $\Pr[x_1 = x_2]$ when $x_1 \in Z_s$ and $x_2 \in Z_{s'}$, where $s \neq s'$.

$$\Pr[\mathsf{COLLD}_{\mathsf{mod}}] \leq \sum_{1 \leq s < s' \leq q} \Pr[\mathsf{COLL}(Z_s, Z_{s'})].$$

In case of $ty^s = ty^{s'} = enc$ and $P_1^s \neq P_1^{s'}$, CC^s is chosen randomly and independently, for any i and j

$$\Pr[S_i^s = S_j^{s'}] = \frac{1}{2^n}.$$

Similarly, for the case $ty^s = ty^{s'} = dec$ and $C_1^s \neq C_1^{s'}$, for any i and j

$$\Pr[S_i^s = S_j^{s'}] = \frac{1}{2^n}.$$

In rest of the case where $ty^s = enc$ and $ty^{s'} = dec$, for i^{th} and j^{th} block, where $1 \leq i \leq m^s - 1$ and $1 \leq j \leq m^{s'} - 1$,
$Z_s^i = Z_{s'}^j$ implies

$$\mathsf{bin}_n(i) \oplus CC^s \oplus H_{h_1}(P_2^s \| \ldots \| P_{m^s}^s, T^s) = \mathsf{bin}_n(j) \oplus MM^{s'} \oplus H_{h_2}(C_2^{s'} \| \ldots \| C_{m^{s'}}^{s'}, T^{s'})$$

Let $s \leq s'$ and $(s, i) \neq (s', j)$. Thus, either CC^s (in case s^{th} is encryption query) or MM^s (in case s'^{th} is decryption query) is uniformly and independently distributed with all other variables. Thus, collision probability is $1/2^n$.
Hence, for $\sigma_n - q$ messages

$$\Pr[\mathsf{COLLD}_{\mathsf{mod}}] \leq \binom{\sigma_n - q}{2} \frac{1}{2^n}.$$

□

Now we can calculate the security bound for the MXCBv1 which is similar to the XCBv1. Difference in the security bound is due to the Claim 2 which we will replace by the Claim 7. Thus, Except the Claim 2, by using the Claim 1 to 7, we get:

$$\begin{aligned}
\mathbf{Adv}^{\pm rnd}_{\mathrm{MXCBv1[3Perm}(n)]}(A) &\leq \frac{2}{2^n}\binom{q}{2} + \frac{2}{2^n}\binom{\sum_{s=1}^{q} m^s - q}{2} + \frac{2(q-1)\sigma_n}{2^n} \\
&\leq \frac{q^2}{2^n} + \frac{(\sigma_n - q)^2}{2^n} + \frac{2\sigma_n q}{2^n} \\
&\leq \frac{2q^2 + \sigma_n^2}{2^n}.
\end{aligned}$$

By using above bound and equation (3.9). We get the bound of MXCBv1 i.e.

$$\mathbf{Adv}^{\pm\widetilde{prp}}_{\mathrm{MXCBv1[3Perm}(n)]}(A) \leq \frac{2.5q^2 + \sigma_n^2}{2^n}.$$

As we stated earlier, similarly we can state the security bound for the MXCBv2 (modified XCBv2). Thus, we can state the security bound of MXCBv1 and MXCBv2fb. For MXCBv1, security bound is as follows:

Theorem 3.6.1. Consider an arbitrary adversary A which queries only with messages/ciphers whose lengths are multiples of n and A asks a total of q queries of overall query complexity σ_n where each query is at most ℓ blocks long (each block of n bits). Then,

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{MXCBv1[3Perm}(n)]}(A) \leq \frac{2.5q^2 + \sigma_n^2}{2^n}. \tag{3.27}$$

Similarly, for MXCBv2 our security bound is given as:

Theorem 3.6.2. Consider an arbitrary adversary A which queries only with messages/ciphers whose lengths are multiples of n and A asks a total of q queries of overall query complexity σ_n where each query is at most ℓ blocks long (each block of n bits). Then,

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathrm{MXCBv2fb[3Perm}(n)]}(A) \leq \frac{3.5q^2 + \sigma_n^2}{2^n}. \tag{3.28}$$

3.7 Comparison of TES Security bounds

In this section, we compare our derived security bound with the bound compared in [2] with the same practical values of the parameter as taken in [2]. So, Available ciphertext/plaintext to adversary is 2^{42}. Also, the block length is 16 bytes. So, total ciphertext/plaintext is $2^{42}/2^4 = 2^{38}$ blocks. And sector size is 4 KB i.e. 2^{12} bytes. Thus, we take message length $2^{12}/2^4 = 2^8$ blocks. Therefore, we have total number of message is $2^{38}/2^8 = 2^{30}$. So, the total query complexity of the adversary is $2^{38} + 2^{30} = 2^{38.006}$, where 2^{30} is the tweak for adversary.

Hence, q is number of queries i.e. 2^{30}, ℓ is maximum query length i.e. $2^8 + 1$ and σ_n query complexity i.e $2^{38.006}$.

In the Table 3.1, we can see even repaired XCB gives the worst security bound as compare to listed TES scheme while MXCB gives the best bound.

3.8 Weak keys analysis of XCB

In 2008, Handschuh and Preneel [9] gave the following definition of weak keys:

> In symmetric cryptology, a class of keys is called a weak key class if for the members of that class the algorithm behaves in an unexpected way and if it is easy to detect whether a particular unknown key belongs to this class. For a MAC algorithm, the unexpected behaviour can be that the forgery probability for this key is substantially larger than average. Moreover, if a weak key class is of size C, one requires that identifying that a key belongs to this class requires testing fewer than C keys by exhaustive search and fewer than C verification queries.

According to this definition, a lot of work has been proposed by Handschuh and Preenel and Saarinen in MAC, AE based on polynomial hash function. Initially, Handschuh and Preenel considered 0 as the only weak key for all polynomial hash function. Later, in 2012 Marakku-Juhani [17], in 2015 Gorden Procter et al.[16] increased the number of weak keys for GCM.

3.8.1 Saarinen's Cycling attacks

Saarinen's cycling attacks was proposed in 2012 in [17] against GCM and other polynomial-based MACs and hashes. The main idea is, if a hash key h lies in a subgroup of order r, then $h^r = 1$. Hence, for any i, j message block M_i and M_{i+jr}

List of some TES			
TES mode	Source	Claimed Bound	Numerical Value
TET	[6]	$\frac{3\sigma_n^2}{2\phi(2^n-1)}$	$2^{-50.40}$
HCTR	[3]	$\frac{4.5\sigma_n^2}{2^n}$	$2^{-49.81}$
CMC	[7]	$\frac{7\sigma_n^2}{2^n}$	$2^{-49.18}$
EME	[8]	$\frac{7\sigma_n^2}{2^n}$	$2^{-49.18}$
HEH, HMCH	[18]	$\frac{20\sigma_n^2}{2^n}$	$2^{-47.66}$
XCB	[12]	$\frac{8q^2(\ell+2)^2}{2^n}$	$2^{-48.96}$
XCBv2fb	[2]	$\frac{(5+2^{22})\ell q\sigma_n}{2^n}$	$2^{-29.98}$
XCBv1	[2]	$\frac{(5+2^{22})\ell q\sigma_n}{2^n}$	$2^{-29.98}$
Repaired XCBv2fb	This Dissertation	$\frac{(5+2^{5})\ell q\sigma_n}{2^n}$	$2^{-46.78}$
Repaired XCBv1	This Dissertation	$\frac{(3+2^{5})\ell q\sigma_n}{2^n}$	$2^{-46.87}$
MXCBv2fb	This Dissertation	$\frac{3.5q^2+\sigma_n^2}{2^n}$	$2^{-51.99}$
MXCBv1	This Dissertation	$\frac{2.5q^2+\sigma_n^2}{2^n}$	$2^{-51.99}$

Table 3.1: Comparison of the Bounds, here ϕ is Euler's totient.

can be swapped without changing the value of hash function. Saarinen also talk about the specific bit swapping instead of whole block swapping while the other condition of M_i and M_{i+jr} will remain same. The forgery technique is successful if the hash key is an element of low order subgroup with order dividing the distance between the swapped message space. This method identifies whether the hash key is in that class or not, by one valid message, tag pair and single verification query. Saarinen observes that any r that divides $2^{128} - 1$ can be used and that swapping of M_i and M_{i+r} will give successful forgery with probability at least $\frac{r+1}{2^{128}}$.

3.8.2 Saarinen's Cycling attacks on XCB

As XCB contain hash function, a relevant question which arise about XCB - Is it contains weak keys? Some of weak keys attack on TES are present in [19], here we will present how Saarinen's cycling attack (2012) which was done against GCM and hashes, can be done in XCBv1 and XCBv2.

In XCBv2, line 108 in Figure 3.3, if a hash key h is of order t, then $h^t = 1$. Therefore, we can swap a plain text block P_i and P_{i+jt} (for some i, j) without changing the original value of the hash function.

$$108. \quad S \leftarrow CC \oplus H_h(0^n \| T, P_1 \| \dots \| P_{m-2} \| \mathsf{pad}(P_{m-1}) \| 0^n).$$

For example, if $h^3 = 1$ then we can swap plaintext P_i and P_{i+3j}, where $1 \leq i, i + 3j \leq m - 1$ hence the plaintext P will change but not the hash function value. Therefore, In line 109, counter mode will be same after swapping the plaintext block. Before swapping, let ciphertext C, plaintext P and counter mode $\mathsf{Ctr}_{K_c,S}$ and after swapping ciphertext C', plaintext P' and counter mode will be same i.e. $\mathsf{Ctr}_{K_c,S}$. Now as we know the plaintext and ciphertext before swapping, therefore we can determine the Counter mode from line 109 i.e $C \oplus P = \mathsf{Ctr}_{K_c,S}$ (here $\oplus$ is block wise XOR, as per XCB scheme). As counter mode before and after swapping is same, so we can easily figure out the C' i.e. $C' = P' \oplus \mathsf{Ctr}_{K_c,S}$. Thus, the cycling attack took place.

$$109. \quad (C_1, \dots, C_{m-1}) \leftarrow \mathsf{Ctr}_{K_c,S}(P_1, \dots, P_{m-2}, P_{m-1}).$$

Similarly, we can perform the weak keys attack on XCBv1.

Chapter 4

Security of HCTR

HCTR was proposed by Wang, Feng and Wu in 2005. It is a mode of operation which provides a tweakable strong pseudorandom permutation [20].

In this chapter, we show how the hash function is insecure. We perform distinguishing attack and the hash key recovery attack on HCTR. Next we analyse the dependency of the two different keys in HCTR. In particular, we analyse the following scenario. Suppose HCTR with keys K and h has been used for some time, and K gets compromised. We show that only changing K would rise to a completely insecure scheme.

4.1 Description of HCTR

HCTR comprises a block cipher E and hash function H. It is a Tweakable Encipher Scheme with two master key : hash key h and block cipher key K. HCTR and XCB have the similar structure i.e. these are hash-counter-hash Tweakable Enciphering Scheme. But HCTR's hash function and Ctr mode definition is different from the XCB.

Following is the definition of the HCTR as defined in [20]:

Polynomial hash function in HCTR : Let P be the plaintext such that $|P| = n(m-1) + r$ for $1 \leq r \leq n$. Partition of P into $P_1||P_2||\ldots||P_m$, where $|P_i| = n$ for $1 \leq i \leq m-1$, 0^{n-r} will append at the end of P to complete the block such that all block size will be n.

$$H_h(P) : \{0,1\}^n \times \{0,1\}^* \rightarrow \{0,1\}^n,$$

and the hash function H is defined as

$$H_h(P) = \begin{cases} h, & \text{if } P \text{ is an empty string,} \\ P_1 h^{n+1} \oplus \cdots \oplus (P_n||0^{n-r})h^2 \oplus ((|P|)h), & \text{otherwise.} \end{cases}$$

Counter mode in HCTR: Given an n-bit string S, the counter mode Ctr is defined as: $\mathsf{Ctr}_{K,S}(A_1, \ldots, A_m) = (A_1 \oplus E_K(S \oplus 1), \ldots, A_m \oplus E_K(S \oplus m)$, where S is the counter and K is the key.

HCTR's encryption algorithm is shown in Figure 4.1 and construction in the Figure 4.2 with plaintext $|P| \geq n$ and tweak $|T| \geq 0$.

```
Encryption under HCTR : $\mathbf{E}_K^T(P)$

Partition $P$ into $P_1, \ldots, P_m$
$CC \leftarrow P_1 \oplus H_h(P_2, \ldots, P_m || T)$
$S \leftarrow CC \oplus E_K(CC)$
$C_2, \ldots, C_m \leftarrow \mathsf{Ctr}_{K,S}(P_2, \ldots, P_m)$
$C_1 \leftarrow E_K(CC) \oplus H_h(C_2, \ldots, C_m || T)$
return $(C_1, \ldots, C_m)$
```

Figure 4.1: Encryption using HCTR

4.2 Insecurity of the hash function

In 2005, the author showed a cubic security bound for HCTR in [20]. Later in 2008, Chakraborty and Nandi gave the quadratic Security bound of HCTR [3] which showed distinguishing advantage of an adversary in distinguishing HCTR and its inverse from a random permutation and its inverse is bounded above by $\frac{4.5\sigma^2}{2^n}$ where n the block-length of the block-cipher and σ is the number of n-block queries made by the adversary (including the tweak). In this section, we show how the above claim is contradictory.

4.2.1 The Case of Empty Message

Here, we give distinguish attack on the HCTR. In distinguish attack on the construction of HCTR, an adversary distinguish between the scheme and random permutation with high probability. Distinguish attack on HCTR which we show here generate the same internal value (i.e. CC) after performing hash function for two different messages. In HCTR's hash function if P is empty string (as an input of hash function) or $P = 0$ (only single 0 as an input of hash function) then the $H_h(P)$ return the same counter value i.e. h, which is collision in hash function. Below we explain how distinguish attack can be performed on HCTR.

Suppose an adversary makes two queries $P^{(1)}$ and $P^{(2)}$ such that $C^{(1)} = \text{HCTR}_{K,h}(P^{(1)})$ and $C^{(2)} = \text{HCTR}_{K,h}(P^{(2)})$ for empty tweak. Assume that $P^{(1)} = x$ and $P^{(2)} = (x \| 0)$ where $x \in \{0,1\}^n$ is arbitrary. Now,

1. Due to collision in hash function for empty input and 0 input, internal value $CC^{(1)}$ and $CC^{(2)}$ have the same value i.e. $x \oplus h$. And so $E_K(CC^{(1)}) = E_K(CC^{(2)})$, say, $E_K(CC)$.

2. If $C_2^{(2)}$ has output 0 then $C_1^{(1)} = C_1^{(2)}$ i.e. $E_K(CC) \oplus h$.

3. Also, $\Pr[C_2^{(2)} = 0] = 1/2$.

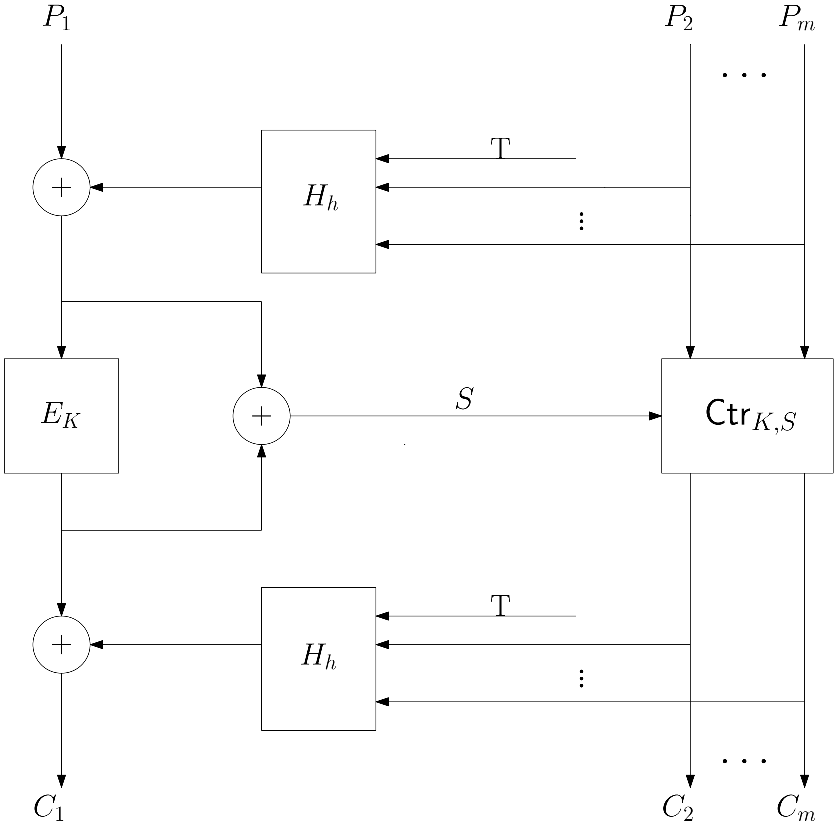

Figure 4.2: Encryption of HCTR

Thus, the advantage of the adversary is $(\frac{1}{2} - \frac{1}{2^n})$ which is very high. Therefore, the bound proved in [3] is contradictory.

Not only distinguish attack, adversary can perform hash key recovery attack. If we make the same set-up as above and $C_2^{(2)}$ is 1 instead of 0 then

$$\begin{aligned} C_1^{(1)} \oplus C_1^{(2)} &= E_K(CC) \oplus h \oplus E_K(CC) \oplus h^2 \oplus h \\ &= h^2. \end{aligned}$$

After getting h^2, adversary can easily retrieve the hash key h. Therefore, in k iteration where $k \geq 1$, adversary have a high probability of retrieving hash key h i.e.

$$\Pr\left[C_2^{(2)} = 1\right] = \left(1 - \frac{1}{2^k}\right).$$

Thus, in the above discussion adversary not only perform the distinguishing attack but also retrieve the hash key h with almost surety from HCTR scheme.

4.2.2 Comment about the Attack

In above two attacks, the weak point is definition of the hash function. Therefore, we can avoid the above attacks by putting some restriction either on the input query or hash function, or both. Here are the following way which can help us to avoid these attacks:

1. We can avoid these attacks if we do not define the hash function for empty string and exclude message of length n-bits or less from the message space of HCTR i.e. $|P| > n$.

2. Above technique put restriction on the message. We can prevent the above attack without changing the message size of HCTR just by modifying the definition of hash function i.e increase the size of the input by one. So, the new hash $H^{'}$ is defined as: $H^{'}_h(P) = H_h(P||1)$.

4.3 Key Dependency in HCTR

In definition of HCTR, we didn't talk about the two master keys whether they have some relation or not. Suppose in an implementation, Two master keys h_1 (hash key) and K_1 (block cipher key) were used and encryption, decryption took place. Suppose after some time, block key K_1 gets compromised and it changed from K_1 to K_2. Also, we have sufficient amount of data available corresponding to the old key pair. Here, we show how one can extract the hash key i.e. h_1 with certainty and the whole scheme is compromised i.e. hash key recovery attack is possible. Thus, the scheme will not be secure any-more.

Suppose an adversary has a plaintext ciphertext pair (P, C) such that $C = \text{HCTR}_{K_1,h_1}(P)$ for empty tweak. Assume that $P = x\|x$ where $x \in \{0,1\}^n$ is arbitrary. Now,

$$C_1 = E_{K_1}(CC) \oplus H_{h_1}(C_2), \tag{4.1}$$

where $CC = x \oplus H_{h_1}(x)$.

$$C_2 = E_{K_1}(S \oplus \mathsf{bin}_n(1)) \oplus x,$$

Here,

$$\begin{aligned} S &= CC \oplus E_{K_1}(CC) \\ &= x \oplus H_{h_1}(x) \oplus E_{K_1}(CC). \end{aligned} \tag{4.2}$$

Now, as we know C_2, x and K_1, we have

$$S = E^{-1}_{K_1}(C_2 \oplus x) \oplus \mathsf{bin}_n(1). \tag{4.3}$$

By using equations (4.1), (4.2) and 4.3, we get

$$\begin{aligned} C_1 \oplus S \oplus x &= E_{K_1}(CC) \oplus H_{h_1}(C_2) \oplus x \oplus H_{h_1}(x) \oplus E_{K_1}(CC) \oplus x, \\ &= (x \oplus C_2){h_1}^2. \end{aligned} \tag{4.4}$$

Here equation (4.4) is a quadratic equation in h_1 which we can solve easily and retrieve the hash key h_1.

To prevent from this attack, we should change both the keys simultaneously. Note we did not comment on changing only the hash key and keeping block cipher key as it is.

Chapter 5

Conclusion

In this dissertation, we improved the security bound of a Tweakable Enciphering Scheme (TES) known as XCB (Extended Code Book). Also, we proposed a Modified XCB (MXCB) and showed that the security bound of MXCB has better numerical value than many other popular TES like HCTR, HCH, TET, HEH, CMC, XCB and EME. We also analysed some weak keys attack on XCB.

Further, we analysed a TES known as HCTR. We performed distinguishing and key recovery attack on the existing HCTR and also showed how it can be avoided easily. We also showed why both the master keys of HCTR should be changed simultaneously or otherwise it could be a serious attack on the construction of HCTR.

Bibliography

[1] Gilles Brassard. On computationally secure authentication tags requiring short secret shared keys. In *Advances in Cryptology*, pages 79–86. Springer, 1983.

[2] Debrup Chakraborty, Vicente Hernandez-Jimenez, and Palash Sarkar. Another look at xcb. *Cryptography and Communications*, 7(4):439–468, 2015.

[3] Debrup Chakraborty and Mridul Nandi. An improved security bound for hctr. In *International Workshop on Fast Software Encryption*, pages 289–302. Springer, 2008.

[4] Debrup Chakraborty and Palash Sarkar. Hch: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In *Indocrypt*, volume 4329, pages 287–302. Springer, 2006.

[5] Debrup Chakraborty and Palash Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. In *International Workshop on Fast Software Encryption*, pages 293–309. Springer, 2006.

[6] Shai Halevi. Invertible universal hashing and the tet encryption mode. In *Annual International Cryptology Conference*, pages 412–429. Springer, 2007.

[7] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In *Annual International Cryptology Conference*, pages 482–499. Springer, 2003.

[8] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In *Cryptographers' Track at the RSA Conference*, pages 292–304. Springer, 2004.

[9] Helena Handschuh and Bart Preneel. Key-recovery attacks on universal hash function based mac algorithms. In *Annual International Cryptology Conference*, pages 144–161. Springer, 2008.

[10] Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and repairing gcm security proofs. In *Advances in Cryptology–CRYPTO 2012*, pages 31–49. Springer, 2012.

[11] David A McGrew and Scott R Fluhrer. The extended codebook (xcb) mode of operation. *IACR Cryptology ePrint Archive*, 2004:278, 2004.

[12] David A McGrew and Scott R Fluhrer. The security of the extended codebook (xcb) mode of operation. In *Selected Areas in Cryptography*, volume 4876, pages 311–327. Springer, 2007.

[13] David A McGrew and John Viega. The security and performance of the galois/counter mode (gcm) of operation. In *International Conference on Cryptology in India*, pages 343–355. Springer, 2004.

[14] Rajeev Motwani and Prbhakar Raghavan. In *Randomized Algorithm.* Cambridge University Press, 2007.

[15] Yuichi Niwa, Keisuke Ohashi, Kazuhiko Minematsu, and Tetsu Iwata. Gcm security bounds reconsidered. In *International Workshop on Fast Software Encryption*, pages 385–407. Springer, 2015.

[16] Gordon Procter and Carlos Cid. On weak keys and forgery attacks against polynomial-based mac schemes. *Journal of Cryptology*, 28(4):769–795, 2015.

[17] Markku-Juhani Olavi Saarinen. Cycling attacks on gcm, ghash and other polynomial macs and hashes. In *Fast Software Encryption*, pages 216–225. Springer, 2012.

[18] Palash Sarkar. Efficient tweakable enciphering schemes from (block-wise) universal hash functions. *IEEE Transactions on Information Theory*, 55(10):4749–4760, 2009.

[19] Zhelei Sun, Peng Wang, and Liting Zhang. Weak-key and related-key analysis of hash-counter-hash tweakable enciphering schemes. In *Australasian Conference on Information Security and Privacy*, pages 3–19. Springer, 2015.

[20] Peng Wang, Dengguo Feng, and Wenling Wu. Hctr: A variable-input-length enciphering mode. In *International Conference on Information Security and Cryptology*, pages 175–188. Springer, 2005.

[21] Mark N Wegman and J Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of computer and system sciences*, 22(3):265–279, 1981.