A Simple Security Analysis of Hash-CBC and a New Efficient One-Key Online Cipher

Mridul Nandi
School of Computer Science
University of Waterloo, Canada

May 7, 2007

Abstract

In Crypto 2001, Bellare *et al.* [1] introduced *online cipher* (or online permutation) and proposed two Hash-CBC mode constructions, namely **HCBC** and **HPCBC** along with security proofs. We observe that, the security proofs in [1] are *wrong* and it may not be fixed easily. In this paper, we provide a *simple* security analysis of these online ciphers. Moreover, we propose two variants of HPCBC, namely **MHCBC-1** and **MHCBC-2**. The first variant, MHCBC-1, is a slight modification of HPCBC so that it is more efficient in performance as well as in memory compare to HPCBC. The other one, MHCBC-2 requires only *one-key* (note that, HCBC and HPCBC require at least two and three keys respectively) and does not require any ε-ΔUniversal Hash Family (which is costly in general).

Keywords : random permutation, pseudo random permutation, Online function, Online permutation, ε- ΔUniversal Hash Family, Hash-CBC.

1 Introduction

In cryptography, a cipher over a domain G means a keyed function family $\{v_K\}_{K \in \mathcal{K}}$, where $\mathcal{K}$ is a key space and for each K in the key space we have a permutation $v_K : G \to G$. The primitive for a cipher is *pseudo random permutation* [14] (or PRP). The pseudo random permutation is a keyed function family which should be indistinguishable from the ideal cipher, the family of all permutation on G, $\mathbf{Perm}(G)$. One popular candidate (probably) for pseudo random permutation is **AES** [7] which is an example of block cipher. For AES, the domain is for example $G = \{0,1\}^{128}$. In most applications, we need a cipher with larger domain such as $G^{[1,\ell]} = \cup_{0 \le i \le \ell} G^i$ for a sufficiently large integer ℓ. Thus, we need to extend the domain of the cipher which is popularly known as *modes of operation*. Recently, there are many modes of operations. Most of these are variants of *Cipher-Block-Chaining* modes (or **CBC**) [2, 3, 5, 11, 13].

In this paper we consider ciphers defined on $G^{[1,\ell]}$ which are computable in the online manner. These ciphers are called *Online Cipher* or *Online Permutation*. That is, if the online cipher text is $(y_1, \cdots, y_\ell) = f(m_1, \cdots, m_\ell)$ for an online cipher f, then y_i can be computed from $m_1, \cdots, m_i$ only. It is easy to observed that online cipher can not be **PRP**. The appropriate security notion is *Pseudo Random Online Permutation* (or **PROP**) as introduced in [1]. In [1] authors have shown that the popular candidates such as **CBC** (with fixed IV, public or secret) [3], **ABC** [9] are not

Pseudo Random Online Permutation. In the same paper [1] two secure online ciphers **HCBC** or Hash-CBC for Chosen Plain text Attack (or **CPA**-secure) and **HPCBC** or Hash-PCBC for Chosen Cipher text Attack (or **CCA**-secure) have been introduced. These online ciphers are based on a pseudo random permutation such as AES and *Almost XOR Universal Hash family* or **AXU** hash family [18]. AXU hash family is a special case of ΔUniversal Hash Family [10, 17, 19].

Online cipher are *real time*, *length-preserving encryption* without having buffering. In fact, HCBC, HPCBC needs current and last message block and last cipher block to compute the current cipher block. Thus, Online cipher could be used where we need the encryption in an online manner with a very small amount of memory.

Our Work.

We first make a framework for analyzing indistinguishability similar to the work provided in [4, 16]. This framework would also help for some other similar scenario like pseudo random function [4, 12, 16], pseudo random permutation for general ciphers (pseudo random permutation defined over an arbitrary large domain) etcetera. We note that the security proofs for HCBC and HPCBC given in [1] are wrong and it may not be fixed easily. We have provided a simple security proof based on our framework. Finally, we introduce two secure variants of HPCBC. One of them is a single-key online cipher without using any universal hash families. These are more efficient in performance as well as in key size compare to HCBC and HPCBC.

Organization of the paper.

In Section 2, we first give a detail description of online function, online permutation and it's security notion. We also characterize a wide class of distinguishers. The main tool of the paper is given in the Theorem 3.2 in the Section 3. Next, we describe HCBC and HPCBC in Section 4 with a simple security proof. We also observe some flaws in the proofs given in [1]. Finally we design our new online cipher with security analysis in Section 5.

2 Online Permutation and Security Notion

In this section we describe online function, online cipher (or online permutation) and its security notion. We also characterize a wide class of distinguishers which includes adaptively chosen cipher text and chosen plain text attack algorithms.

2.1 Online Function and Online Permutation

To define an online permutation we first define an underlying function, which we term as **online function**. Let $(G, +)$ be a group with $|G| = N$ and denote $G^{[a,b]} = \cup_{a \le i \le b} G^i$ for nonnegative integers $a \le b$. We denote $G^0 = \{\lambda\}$, the set consisting of the empty string λ. Now the definition of an *online function* X on $G^{[1,\ell]}$ is given below for a sufficiently large integer ℓ.

Definition 2.1. (Online Function)
A function $X : G^{[1,\ell]} \to G$ *is said to be an* **online function** *if for every* $(m_1, \cdots, m_{i-1}) \in G^{i-1}$, $1 \le i \le \ell$, *the map* $x \mapsto y = X(m_1, \cdots, m_{i-1}, x)$ *is a permutation on* G *(as a function of* x*). We write* $\overline{X}(m_1, \cdots, m_{i-1}, x, 1) = y$ *and* $\overline{X}(m_1, \cdots, m_{i-1}, y, -1) = x$.

Now we denote some related sets.

1. $\mathbf{Perm}(G)$: the set of all permutations on G.
2. $\mathcal{F} \stackrel{\Delta}{=} \mathbf{Func}(G^{[0,\ell-1]}, \mathbf{Perm}(G))$, the set of all functions from $G^{[0,\ell-1]}$ to $\mathbf{Perm}(G)$.
3. $\mathcal{F}_1$: the set of all online functions on $G^{[1,\ell]}$.

An online function $X_0 \in \mathcal{F}_1$ is uniquely characterized by $\pi^{X_0} \in \mathcal{F}$ such that $\pi^{X_0}(q)(x) \stackrel{\Delta}{=} X(q,x)$, $x \in G, q \in G^{[0,\ell-1]}$. We denote q by $\mathbf{chop}(q,x)$ and x by $\mathbf{last}(q,x)$. Thus, $\overline{X_0}(p,1) = \pi^{X_0}(\mathbf{chop}(p))(\mathbf{last}(p))$ and $\overline{X_0}(p,-1) = \pi^{X_0}(\mathbf{chop}(p))^{-1}(\mathbf{last}(p))$.

- A random function from A to B is a probability distribution on $\mathbf{Func}(A,B)$, the set of all functions from A to B.
- A random function is said to be an uniform random function if it is an uniform distribution on $\mathbf{Func}(A,B)$. v is said to be a uniform random permutation on G if it is an uniform distribution on $\mathbf{Perm}(G)$.
- Similarly, a *random online function* X is a probability distribution on $\mathcal{F}$ (equivalently on $\mathcal{F}_1$). Thus, we denote X as a random variable taking values on $\mathcal{F}_1$.
- It is said to be *uniform random online function* (UROF) if the distribution is uniform on $\mathcal{F}$ and we denote it by U.

Note that, X, U are not any fixed online functions, instead they can be any fixed online function $X_0 \in \mathcal{F}_1$ with some probability. For example, for any $X_0 \in \mathcal{F}_1$, $\Pr[U = X_0] = \frac{1}{(N!)^{(1+N+\cdots+N^{\ell-1})}}$ since $|\mathcal{F}_1| = (N!)^{(1+N+\cdots+N^{\ell-1})}$. We use $X, U, X^{\mathrm{H}}, \cdots$ to denote random online functions and $X_0, X_1, \cdots$ to denote some fixed online functions from $\mathcal{F}_1$. Since U is an uniform distribution on $\mathcal{F}$, $\pi^U(p)$ is an uniform random permutation on G for all $p \in G^{[0,\ell-1]}$. Moreover, for a distinct set of elements $p_1, \cdots, p_s$, $\pi^U(p_1), \pi^U(p_2), \cdots, \pi^U(p_s)$ are independently distributed uniform random permutations.

Lemma 2.1. *If $p_1, \cdots, p_s$ are distinct elements from $G^{[0,\ell-1]}$ then $\pi^U(p_1), \pi^U(p_2), \cdots, \pi^U(p_s)$ are independently distributed uniform random permutations. Thus, $U(p)$ is independent with $(U(p_1), \cdots, U(p_s))$ if $\mathbf{chop}(p) \neq p_i$, $1 \le i \le \sigma$.*

Proof. Since U has uniform distribution on $\mathcal{F}_1$, the corresponding random function π^U also has uniform distribution on $\mathcal{F} = \mathbf{Func}(G^{[0,\ell-1]}, \mathbf{Perm}(G))$. Thus, for distinct $p_1, \cdots, p_s$, the random functions $\pi^U(p_1), \cdots, \pi^U(p_s)$ have independent and uniform distribution on $\mathbf{Perm}(G)$. ■

Lemma 2.2. (**UROF is close to URF**) *Let $\sigma \ge 0$, $y_i \in G$ and $p \neq p_i \in G^{[1,\ell]}$ for $1 \le i \le \sigma$ such that $\Pr[U(p_1) = y_1, \cdots, U(p_\sigma) = y_\sigma] > 0$. Then the conditional probability*

$$\Pr[U(p) = y \mid U(p_1) = y_1, \cdots, U(p_\sigma) = y_\sigma] \le \frac{1}{N-\sigma}.$$

For $\sigma = 0$, the lemma says that $\Pr[U(p) = y] \le 1/N$ (in fact equal).

Proof. Let $p = (q, x)$ and $p_i = (q_i, x_i)$ where $q = \mathbf{chop}(p), q_i = \mathbf{chop}(p_i), x = \mathbf{last}(p)$ and $x_i = \mathbf{last}(p_i)$, $1 \leq i \leq \sigma$. Without loss of generality we assume that $q = q_1 = \cdots = q_j$ and $q_i \neq q$ for some $j \geq 0$ and for all $\sigma \geq i > j$. Thus,

$$\Pr[U(p) = y \mid U(p_1) = y_1, \cdots, U(p_\sigma) = y_\sigma]$$

$$= \Pr[U(p) = y \mid U(p_1) = y_1, \cdots, U(p_j) = y_j] \leq \frac{1}{N-j} \leq \frac{1}{N-\sigma}.$$

The first equality holds since $U(p_{j+1}), \cdots, U(p_\sigma)$ are independent with $U(p)$. Now the conditional probability for $U(p) = y$ given that $U(p_1) = y_1, \cdots, U(p_j) = y_j$ is either zero (if for some $i \leq j$, $y = y_i$) or $\frac{1}{N-j_1}$ for some $j_1 \leq j$. More precisely, j_1 is the number of distinct p_i's among $p_1, \cdots, p_j$. In case of $\sigma = 0$, $U(p)$ is uniformly distributed on G. ■

For an uniform random function u from $G^{[1,\ell]}$ to G, $\Pr[u(p) = y \mid u(p_1) = y_1, \cdots, u(p_\sigma) = y_\sigma] = \frac{1}{N}$ whereas the above Lemma says that UROF has conditional probability at most $\frac{1}{N-\sigma}$. These two conditional probabilities are very close. In other words, the difference between these two probabilities are negligible provided σ is small compare to N. Next we compute the interpolation probability for UROF. Note that, for an uniform random function u, we have $\Pr[u(p_1) = y_1, \cdots, u(p_\sigma) = y_\sigma] = \frac{1}{N^\sigma}$.

Lemma 2.3. (**Interpolation probability for** UROF) *For any distinct $p_i \in G^{[1,\ell]}$ and any $y_i \in G$, $1 \leq i \leq \sigma$ we have,*

$$\Pr[U(p_1) = y_1, \cdots, U(p_\sigma) = y_\sigma] \leq \frac{1}{N(N-1)\cdots(N-\sigma+1)}. \tag{1}$$

Proof. If for some i, $\Pr[U(p_1) = y_1, \cdots, U(P_i) = y_i] = 0$ then $\Pr[U(p_1) = y_1, \cdots, U(p_\sigma) = y_\sigma] = 0$ and we are done. So we assume that for all i, $\Pr[U(p_1) = y_1, \cdots, U(P_i) = y_i] > 0$. Now, $\Pr[U(p_1) = y_1, \cdots, U(p_\sigma) = y_\sigma] = \Pr[U(p_1) = y_1] \times \Pr[U(p_2) = y_2 \mid U(p_1) = y_1] \times \cdots \times \Pr[U(p_\sigma) = y_\sigma \mid U(p_1) = y_1, \cdots, U(p_{\sigma-1}) = y_{\sigma-1}]$. By applying Lemma 2.2 for each factor we obtain the bound. ■

Definition 2.2. Online Permutation (Cipher) : *Given an online function X_0, we define an* online permutation *or* an online cipher f_{X_0} *as follows :* $f_{X_0} : G^{[1,\ell]} \to G^{[1,\ell]}$, *such that*

$$f_{X_0}(m_1, \cdots, m_i) = (X_0(m_1),\ X_0(m_1, m_2),\ \cdots,\ X_0(m_1, \cdots, m_i))$$

where $m_1, \cdots, m_i \in G,\ 1 \leq i \leq \ell$.

Note that, if $X_1 \neq X_2$ then $f_{X_1} \neq f_{X_2}$. If X is a random online function then f_X is called a *random online permutation*. We denote the set of all online permutations by $\mathcal{F}_2$. Thus a random online permutation f_X has a distribution on $\mathcal{F}_2$ such that $\Pr[f_X = f_{X_0}] = \Pr[X = X_0]$. We call f_U by an *uniform random online permutation* where U is an uniform random online function. For $a_1, \cdots, a_n \in G, 1 \leq i \leq j \leq n$, we denote $(a_1, \cdots, a_n)[i, j] = (a_i, \cdots, a_j)$. We write $(a_1, \cdots, a_n)[i]$ instead of $(a_1, \cdots, a_n)[i, i]$. Now we have some properties of online cipher.

Proposition 2.4. *Every online cipher f_{X_0} is a* length preserving permutation (*i.e., the restricted function $f_{X_0}|_{G^i}$ is a permutation on G^i*) *which can be computed in an online manner* (*that is, for $j \leq i, f_{X_0}(m_1, \cdots, m_i)[j]$ depends only on $(m_1, \cdots, m_j)$ and not on $m_{j+1}, \cdots, m_i$). We call the last property by* online computation property).

Proof. Length preserving and online computation property follow directly from the definition of f_{X_0}. Suppose, $f_{X_0}(m_1^1, \cdots, m_i^1) = f_{X_0}(m_1^2, \cdots, m_j^2)$, then we have $i = j$ (since it is length preserving) and

$$X_0(m_1^1) = X_0(m_1^2) \;\Rightarrow\; m_1^1 = m_1^2 = m_1 \text{ (say)}$$
$$X_0(m_1, m_2^1) = X_0(m_1, m_2^2) \;\Rightarrow\; m_2^1 = m_2^2 = m_2 \text{ (say)}$$
$$\vdots$$
$$X_0(m_1, \cdots, m_{i-1}, m_i^1) = X_0(m_1, \cdots, m_{i-1}, m_i^2) \;\Rightarrow\; m_i^1 = m_i^2 = m_i \text{ (say)}$$

Thus, $(m_1^1, \cdots, m_i^1) = (m_1^2, \cdots, m_i^2)$ and hence f_{X_0} is a permutation on $G^{[1,\ell]}$. ■

2.2 Characterization of a Class of Distinguisher

Now we study a wide class of distinguishers and characterize them. Intuitively, we consider all oracle algorithms $\mathcal{D}$, (1) which may be probabilistic, (2) which has access of two oracles f (online cipher on $G^{[1,\ell]}$) and f^{-1} (inverse of f), (3) which can make at most k queries and (4) the queries and final outputs are adaptive, i.e., it depends on the previous query-responses. Now we have some notations to define a distinguisher more precisely.

- $V \stackrel{\Delta}{=} G^{[1,\ell]} \times G^{[1,\ell]} \times \{-1, 0, 1\}$.
- Let R be a random variable taking values on $\mathcal{R}$.
- For each $r \in \mathcal{R}$ we have a tuple $(Q_1^r, \cdots, Q_k^r, \mathcal{D}_{\text{out}}^r)$. For $1 \leq i \leq k$

$$Q_i^r : V^{i-1} \to G^{[1,\ell]} \times \{-1, 0, 1\} \text{ and } \mathcal{D}_{\text{out}}^r : V^{[1,k]} \to \{0, 1\}.$$

Q_i^r denotes the i^{th} query when r is chosen as a random string for $\mathcal{D}$. There are three possible behaviors of the distinguisher. '1' denotes f-query, '-1' denotes f^{-1}-query and '0' denotes that no more query. $\mathcal{D}_{\text{out}}^r$ returns final output based on all query-responses. Now we have the detailed algorithm for the distinguisher $\mathcal{D}$.

|Algorithm $\mathcal{D} = (\mathbf{R}, (\mathbf{Q_1^r}, \cdots, \mathbf{Q_k^r}, \mathcal{D}_{\text{out}}^{\mathbf{r}})_{\mathbf{r} \in \mathcal{R}})$|

1. It chooses a string r from the distribution R on $\mathcal{R}$.
2. For $i = 1$ to k
   - $(p, \delta_i) = Q_i^r((M_1, C_1, \delta_1), \cdots, (M_{i-1}, C_{i-1}, \delta_{i-1}))$.
   - If $\delta_i = 1$ then $M_i = p$ and it makes f-query with input M_i and obtains response C_i.
   - If $\delta_i = -1$ then $C_i = p$ and it makes f^{-1}-query with input C_i and obtains response M_i.
   - If $\delta_i = 0$ then it set $k_1 = i - 1$ and returns $\mathcal{D}_{\text{out}}^r((M_1, C_1, \delta_1), \cdots, (M_{i-1}, C_{i-1}, \delta_{i-1}))$.
3. It returns $\mathcal{D}_{\text{out}}^r((M_1, C_1, \delta_1), \cdots, (M_k, C_k, \delta_k))$ and set $k_1 = k$.

We call the tuple $((M_1, C_1, \delta_1), \cdots, (M_{k_1}, C_{k_1}, \delta_{k_1}))$ as the *transcript* of the distinguisher. The output of $\mathcal{D}^r_{\text{out}}$ completely depends on the transcript and the probability distribution of the transcript is induced by the probability distribution of the oracle f.

If $M \in G^i$, we write $|M| = i$. Let $k_1 \leq k$ denotes the total number of queries made by the above distinguisher $\mathcal{D}$ and $\sigma = \sum_{i=1}^{k_1} |M_i|$, the total number of message blocks. We are interested in all distinguisher with runtime at most t (later we see that the security bound is independent of t), making at most k queries having at most σ many blocks.

2.3 Security Notion for Online Cipher

Let $\mathcal{D}$ be a distinguisher (as defined in the previous Subsection 2.2) which wants to distinguish (f_X, f_X^{-1}) from (f_U, f_U^{-1}). Define advantage of $\mathcal{D}$ by

$$\mathbf{Adv}_{X,U}(\mathcal{D}) = \big|\Pr[\mathcal{D}^{f_X, f_X^{-1}} = 1] - \Pr[\mathcal{D}^{f_U, f_U^{-1}} = 1]\big|.$$

Note that the probability is taken under the probability distribution due to R the random choice of the distinguisher and the distribution induced from the underlying random online function X or X^{U}. We define,

$$\mathbf{Adv}_{X,U}(t, k, \sigma) \stackrel{\Delta}{=} \mathbf{max}_{\mathcal{D}}\ \mathbf{Adv}_{X,U}(\mathcal{D})$$

where maximum is taken over all choices of distinguishers which runs in time t making at most k queries having at most σ many blocks. Now note that, $\Pr[\mathcal{D}^{f_X, f_X^{-1}} = 1] = \sum_{r \in \mathcal{R}} \Pr[\mathcal{D}_r^{f_X, f_X^{-1}} = 1] \times \Pr[R = r]$ where $\mathcal{D}_r$ is the deterministic algorithm which behaves exactly same as $\mathcal{D}$ after choosing the random string r in the first step of the algorithm $\mathcal{D}$ (given in previous Subsection 2.2). Thus,

$$\mathbf{Adv}_{X,U}(\mathcal{D}) = \big|\sum_{r \in \mathcal{R}} (\Pr[\mathcal{D}_r^{f_X, f_X^{-1}} = 1] - \Pr[\mathcal{D}_r^{f_U, f_U^{-1}} = 1]) \times \Pr[R = r]\big|$$

$$\leq \mathbf{max}_{r \in \mathcal{R}}\ \big|\Pr[\mathcal{D}_r^{f_X, f_X^{-1}} = 1] - \Pr[\mathcal{D}_r^{f_U, f_U^{-1}} = 1]\big| = \mathbf{Adv}_{X,U}(\mathcal{D}_{r^*}),$$

where the maximum takes place at $r = r^*$. Thus we have,

$$\mathbf{Adv}_{X,U}(t, k, \sigma) = \mathbf{max}_{\mathcal{D}}\ \mathbf{Adv}_{X,U}(\mathcal{D})$$

where maximum is taken over all choices of **deterministic** distinguishers which runs in time t making at most k queries having at most σ many blocks. So, now onwards we can assume that the distinguisher $\mathcal{D}$ is deterministic which can be characterized by a tuple $(Q_1, \cdots, Q_k, \mathcal{D}_{\text{out}})$ as defined in the previous subsection. Also note that

$$\mathbf{Adv}_{X,U}(t, k, \sigma) \stackrel{\Delta}{=} \mathbf{max}_{\mathcal{D}}\big(\Pr[\mathcal{D}^{f_U, f_U^{-1}} = 1] - \Pr[\mathcal{D}^{f_X, f_X^{-1}} = 1]\big)$$

This is true since we can always consider $\overline{\mathcal{D}} = (Q_1, \cdots, Q_k, \overline{\mathcal{D}}_{\text{out}})$ where $\mathcal{D} = (Q_1, \cdots, Q_k, \mathcal{D}_{\text{out}})$ and $\overline{\mathcal{D}}_{\text{out}}$ returns the complement of what $\mathcal{D}_{\text{out}}$ returns. Thus, if the maximum is attained for the distinguisher $\mathcal{D}$ with $\Pr[\mathcal{D}^{f_U, f_U^{-1}} = 1] < \Pr[\mathcal{D}^{f_X, f_X^{-1}} = 1]$ then $\mathbf{Adv}_{X,U}(t, k, \sigma)$

$$= \big|\Pr[\mathcal{D}^{f_U, f_U^{-1}} = 1] - \Pr[\mathcal{D}^{f_X, f_X^{-1}} = 1]\big| = \big(\Pr[\overline{\mathcal{D}}^{f_U, f_U^{-1}} = 1] - \Pr[\overline{\mathcal{D}}^{f_X, f_X^{-1}} = 1]\big).$$

So we will consider only those distinguisher for which $\Pr[\mathcal{D}^{f_U, f_U^{-1}} = 1] \geq \Pr[\mathcal{D}^{f_X, f_X^{-1}} = 1]$.

3 Main Tool of the Paper

3.1 An Equivalent Distinguisher $\widehat{\mathcal{D}}$

Lemma 3.1. *For any given random online function X, $f_X(m_1, \cdots, m_i) = (\overline{X}(m_1, 1), \cdots, \overline{X}(m_1, \cdots, m_i, 1))$ and $f_X^{-1}(y_1, \cdots, y_i) = (\overline{X}(y_1, -1), \overline{X}(m_1, y_2, -1), \cdots, \overline{X}(m_1, \cdots, m_{i-1}, y_i, -1))$ where $m_1 = \overline{X}(y_1, -1)$, $m_2 = \overline{X}(m_1, y_2, -1), \cdots$, $m_{i-1} = \overline{X}(m_1, \cdots, m_{i-2}, y_{i-2}, -1)$.*

Proof. The first equality follows directly from the definition of f_X (see definition of online permutation) and the definition of $\overline{X}$ (see Definition 2.1). Let $f_X^{-1}(y_1, \cdots, y_i) = (m_1, \cdots, m_i)$, that is $f_X(m_1, \cdots, m_i) = (y_1, \cdots, y_i)$. Thus, $X(m_1) = y_1$ and hence $m_1 = \overline{X}(y_1, -1)$. Similarly, $X(m_1, m_2) = y_2$ and hence $m_2 = \overline{X}(m_1, y_2, -1)$ and so on. ∎

By above lemma, a query $f_X(M)$ can be replaced by several queries $\overline{X}(m_1, 1), \cdots, \overline{X}(m_1, \cdots, m_i, 1)$ where $M = (m_1, \cdots, m_i)$. Similarly, a query $f_X^{-1}(C)$ is replaced by queries $\overline{X}(y_1, 1) = m_1$, $\overline{X}(m_1, y_2, -1) = m_2, \cdots, \overline{X}(m_1, \cdots, m_{i-1}, y_i, -1) = m_i$ where $C = (y_1, \cdots, y_i)$.

Now we describe an oracle algorithm which interacts with random online function instead of random online cipher. Now we define an oracle algorithm $\widehat{\mathcal{D}}^{\overline{X}}$ which behaves exactly same as $\mathcal{D}^{f_X, f_X^{-1}}$. Note that we can split the f_X query into several $\overline{X}(\cdot, 1)$ queries and similarly split f_X^{-1} queries into several $\overline{X}(\cdot, -1)$ queries. It skips all identical queries. Also, if $\overline{X}(m_1, \cdots, m_{i-1}, m_i, 1) = c_i$ has been queried before then it skips the query $\overline{X}(m_1, \cdots, m_{i-1}, c_i, -1)$ since the response is m_i always. Similarly, for the converse statement. This modified distinguishing algorithm is denote as $\widehat{\mathcal{D}}$. The final output of $\widehat{\mathcal{D}}$ is exactly same as $\mathcal{D}$ and it can be computed as all the query-responses of $\mathcal{D}$ is determined by $\widehat{\mathcal{D}}$.

Let $\overline{V} = G^{[1,\ell]} \times \{1, -1\} \times G$, $\widehat{Q}_i : \overline{V}^{i-1} \to G^{[1,\ell]} \times \{1, -1\}$, $1 \leq i \leq \sigma$ and $\widehat{\mathcal{D}}_{\mathrm{out}} : \overline{V}^{\sigma} \to \{0, 1\}$. Note that, $\overline{X} : G^{[1,\ell]} \times \{1, -1\} \to G$. We define $\widehat{D}$ by the tuple $(\widehat{Q}_1, \cdots, \widehat{Q}_\sigma, \widehat{\mathcal{D}}_{\mathrm{out}})$ and works as given below. For simplicity, we assume that $\widehat{D}$ makes exactly σ many queries.

1. For $i = 1$ to σ
   - It computes $(p_i, \delta_i) = \widehat{Q_i}(p_1, \delta_i, z_1, \cdots, p_{i-1}, \delta_{i-1}, z_{i-1})$ and makes query $\overline{X}(p_i, \delta_i)$ and obtains response say z_i, where $p_1, \cdots, p_i \in G^{[1,\ell]}$, $z_1, \cdots, z_i \in G$ and $\delta_i = \pm 1$.
2. It returns $\widehat{\mathcal{D}}_{\mathrm{out}}(p_1, \delta_i, z_1, \cdots, p_\sigma, \delta_\sigma, z_\sigma)$.

We consider those distinguisher which make no identical and trivial queries. If $\delta_i = 1$ then we write $x_i = \mathbf{last}(p_i)$, $y_i = z_i$ and if $\delta_i = -1$ then we write $y_i = \mathbf{last}(p_i)$, $x_i = z_i$. Denote $p'_i = (\mathbf{chop}(p_i), x_i)$. Thus, p'_i's are distinct. From the above query-responses we can say that, $X(p'_i) = y_i$. We consider all distinguisher $\widehat{\mathcal{D}}$ where the set $\mathbb{P} = \{p'_1, \cdots, p'_\sigma\}$ is *prefix closed.* A set $\mathbb{P}$ is said to be prefix closed if either $\mathbf{chop}(p) = \lambda$ or $\mathbf{chop}(p) \in \mathbb{P}$. Thus, there exists a set of messages $M_1, \cdots, M_k$ such that $\mathbb{P} = \{p : p \text{ is a prefix for some } M_i\}$. We denote $\mathbb{P} = \mathbb{P}[M_1, \cdots, M_k]$. Note that the distinguisher $\widehat{\mathcal{D}}$ (derived from $\mathcal{D}$) also makes a set of queries with a prefix closed set $\mathbb{P}$ (since it makes $\overline{X}$ queries for all prefixes of M_i or C_i). From now onwards we always assume that $\mathbb{P}$ is prefix closed, where $\mathbb{P}$ is obtained as above by the distinguisher $\widehat{\mathcal{D}}$. It is easy to see that

$$\mathbf{Adv}_{X,U}(t, k, \sigma) = \mathbf{max}_{\widehat{\mathcal{D}}} \Pr[\widehat{\mathcal{D}}^{\overline{U}} = 1] - \Pr[\widehat{\mathcal{D}}^{\overline{X}} = 1].$$

In the next subsection, we study how one can bound the advantage by studying the distinguisher $\widehat{\mathcal{D}}$. Like in previous Section we consider only those distinguisher for which $\Pr[\widehat{\mathcal{D}}^{\overline{U}} = 1] \geq \Pr[\widehat{\mathcal{D}}^{\overline{X}} = 1]$.

3.2 Main Result

A tuple $v = (p_1, \sigma_1, z_1, \cdots, p_\sigma, \delta_\sigma, z_\sigma)$ is called $\widehat{\mathcal{D}}$-*compatible view* or (only *view*) if $Q_i(p_1, \sigma_1, z_1, \cdots, p_{i-1}, \delta_{i-1}, z_{i-1}) = (p_i, \delta_i)$, for $1 \le i \le \sigma$. Denote the set of all views by $\mathcal{V}$ and denote the set of all views v such that $\widehat{\mathcal{D}}_{\text{out}}(v) = 1$ by $\mathcal{V}_1$. Given an online function X_0, we denote the view obtained by the distinguisher when interact with $\overline{X}_0$ as v_{X_0}. Similarly, v_X denotes the random variable whose probability distribution is induced from the random online function X.

Theorem 3.2. *Suppose for all* $v \in \mathcal{V} \setminus \text{Bad}$, $\Pr[v_X = v] \ge (1 - \varepsilon_1) \times \Pr[v_U = v]$ *where* Bad *is some set of views and* $\Pr[v_U \in \text{Bad}] \le \varepsilon_2$, *then we have,* $\mathbf{Adv}_{X,U}(\widehat{\mathcal{D}}) \le \varepsilon_1 + \varepsilon_2$.

Proof.

$$\begin{aligned}\mathbf{Adv}_{X,U}(\widehat{\mathcal{D}}) &= \Pr[v_U \in \text{Bad} \cap \mathcal{V}_1] - \Pr[v_X \in \text{Bad} \cap \mathcal{V}_1] \\ &\quad + \sum_{v \in \mathcal{V}_1 \setminus \text{Bad}} \big(\Pr[v_U = v] - \Pr[v_X = v]\big) \\ &\le \varepsilon_2 + \varepsilon_1 \times \Pr[v_U \in \mathcal{V}_1 \setminus \text{Bad}] \le \varepsilon_1 + \varepsilon_2. \qquad \blacksquare\end{aligned}$$

In this paper we will be interested in four types of bad sets Bad denoted as B_1, B_2, B_3 and B_4. Let $v = (p_1, \delta_1, z_1, \cdots, p_\sigma, \delta_\sigma, z_\sigma) \in \mathcal{V}$ and $p'_i, x_i = \mathbf{last}(p'_i)$ and y_i are defined as in Section 3.1. Define,

- $B_1 = \{v : y_i = y_j \text{ for some } i \ne j \text{ or } y_i = 0 \text{ for some } i\}$
- $B_2 = \{v : (y_i, x_i) = (y_j, x_j) \text{ for some } i \ne j \text{ or } (y_i, x_i) = (0, 0) \text{ for some } i\}$.
- $B_3 = \{v : y_i + x_i = y_j + x_j \text{ for some } i \ne j \text{ or } y_i + x_i = 0 \text{ for some } i\}$.
- $B_4 = \{v : y_i + x_i = y_j + x_j \text{ for some } i \ne j \text{ or } y_i + x_i = 0 \text{ or } y_i + x_i = 1 \text{ for some } i\}$.

Now we show that the above bad sets can arise with very small probabilities when an adversary interacts with an UROF U.

Lemma 3.3. *For a chosen plain text distinguisher (that is,* $\delta_i = 1$ *for all* i*),* $\Pr[v_U \in B_1] \le \frac{\sigma(\sigma+1)}{N}$ *where* $\sigma < N/2$.

Proof. Note that, given $(p_1, \sigma_1, y_1, \cdots, p_{i-1}, \sigma_{i-1}, y_{i-1})$ (the view up to $(i-1)$th query), the probability that $X(p_i) = y_j$ or 0 for some $j < i$ is at most $i/(N - i + 1)$ (see Lemma 2.2). Thus, $\Pr[v_U \in B_1] \le \sum_{i=1}^{\sigma} \frac{i}{N-i+1} \le \frac{\sigma(\sigma+1)}{N}$ if $\sigma < N/2$. $\blacksquare$

Lemma 3.4. $\Pr[v_U \in B_2] \le \frac{\sigma(\sigma+1)}{N}$ *where* $\sigma < N/2$.

Proof. Given $(p_1, \sigma_1, y_1, \cdots, p_{i-1}, \sigma_{i-1}, y_{i-1})$ (the view up to $(i-1)$th query) and $\delta_i = 1$, the probability that $\overline{X}(p_i, \delta_i) = y_j$ or 0 for some $j < i$ is at most $i/(N-i+1)$ (see Lemma 2.2). Similarly, for given $(p_1, \sigma_1, y_1, \cdots, p_{i-1}, \sigma_{i-1}, y_{i-1})$ and $\delta_i = -1$, the probability that $\overline{X}(\mathbf{chop}(p_i), y_i, -1) = x_j$ or 0 for some $j < i$ is at most $i/(N-i+1)$ (by similar reason). Thus, $\Pr[v_U \in B_2] \le \sum_{i=1}^{\sigma} \frac{i}{N-i+1} \le \frac{\sigma(\sigma+1)}{N}$ if $\sigma < N/2$. $\blacksquare$

Lemma 3.5. $\Pr[v_U \in B_3] \le \frac{\sigma(\sigma+1)}{N}$ *where* $\sigma < N/2$.

Proof. Given $(p_1, \sigma_1, y_1, \cdots, p_{i-1}, \sigma_{i-1}, y_{i-1})$ (the view up to $(i-1)$th query) and $\delta_i = 1$, the probability that $\overline{X}(p_i, \delta_i) = x_j + y_j - x_i$ or $-x_i$ for some $j < i$ is at most $i/(N-i+1)$ (see Lemma 2.2). Similarly, for given $(p_1, \sigma_1, y_1, \cdots, p_{i-1}, \sigma_{i-1}, y_{i-1})$ and $\delta_i = -1$, the probability that $\overline{X}(\mathbf{chop}(p_i), y_i, -1) = x_j + y_j - y_i$ or $-y_i$ for some $j < i$ is at most $i/(N-i+1)$ (by similar reason). Thus, $\Pr[v_U \in B_3] \leq \sum_{i=1}^{\sigma} \frac{i}{N-i+1} \leq \frac{\sigma(\sigma+1)}{N}$ if $\sigma < N/2$. ∎

Lemma 3.6. $\Pr[v_U \in B_4] \leq \frac{(\sigma+1)(\sigma+2)}{N}$ *where* $\sigma < N/2$.

Proof. A similar proof can be applied as in the proof of the above Lemma 3.5. Here we consider the probability that $\overline{X}(p_i, \delta_i) = x_j + y_j - x_i$ or $-x_i$ or $1 - x_i$ for some $j < i$. This conditional probability is at most $\frac{i+1}{N-i+1}$. So, $\Pr[v_U \in B_4] \leq \sum_{i=1}^{\sigma} \frac{i+1}{N-i+1} \leq \frac{(\sigma+1)(\sigma+2)}{N}$ if $\sigma < N/2$. ∎

To apply the Theorem 3.2, we also have to prove that $\Pr[v_X = v] \geq (1-\varepsilon_1) \times \Pr[v_U = v]$ for all $v \in \mathcal{V} \setminus \mathrm{Bad}$. Note that, $\Pr[v_X = v] = Pr[X(p'_1) = y_1, \cdots, X(p'_\sigma) = y_\sigma]$. In the next Section 4 we will prove the following.

1. $Pr[X^H(p'_1) = y_1, \cdots, X^H(p'_\sigma) = y_\sigma] \geq (1-\varepsilon_1) \times \Pr[U(p'_1) = y_1, \cdots, U(p'_\sigma) = y_\sigma]$ where y_i's are nonzero distinct elements, X^H is the HCBC random online function and ε_1 is some negligible constant which will be determined later. Lemma 2.3 says that $\Pr[U(p'_1) = y_1, \cdots, U(p'_\sigma) = y_\sigma] \leq \frac{1}{N(N-1)\cdots(N-\sigma+1)}$. Thus it would be enough to show that $Pr[X^H(p'_1) = y_1, \cdots, X^H(p'_\sigma) = y_\sigma] \geq \frac{(1-\varepsilon_1)}{N(N-1)\cdots(N-\sigma+1)}$.

2. $\Pr[X^P(p'_1) = y_1, \cdots, X^P(p'_\sigma) = y_\sigma] \geq \frac{(1-\varepsilon_1)}{N(N-1)\cdots(N-\sigma+1)} \geq (1-\varepsilon_1) \times \Pr[U(p'_1) = y_1, \cdots, U(p'_\sigma) = y_\sigma]$ where $(\mathbf{last}(p'_i) + y_i)$'s are nonzero distinct, X^P is the HPCBC random online function and ε_1 is some negligible constant which will be determined later.

4 Hash-CBC (or HCBC) and Hash-PCBC (or HPCBC) [1]

4.1 ε-ΔUniversal Hash Family [10, 19] and Uniform Random Permutation

Definition 4.1. (ε-Δ**Universal Random Function**)
Let H be a random function from G' to G such that $\Pr[H(x_1) - H(x_2) = y] \leq \varepsilon$ for all $x_1 \neq x_2 \in G'$ and for all $y \in G$. We say this random function by ε-ΔUniversal Random Function *or* ε-ΔUniversal Hash Family *from G' to G.*

Lemma 4.1. *Let $a_p \in G'$ and $b_p \in G$ (a group with addition $+$), for all $p \in \mathbb{P}$ such that (a_p, b_p)'s are distinct. Then,* $\Pr[(H(a_p) + b_p)$*'s are distinct for all $p \in \mathbb{P}] \geq (1 - \binom{\sigma}{2}\varepsilon)$ where H is an ε-Δ Universal Hash Family from G' to G and $\sigma = |\mathbb{P}|$.*

Proof. For all $p_1 \neq p_2 \in \mathbb{P}$, $\Pr[H(a_{p_1}) + b_{p_1} = H(a_{p_2}) + b_{p_2}] \leq \varepsilon$. This is true since if $a_{p_1} = a_{p_2}$ the probability is zero and if $a_{p_1} \neq a_{p_2}$ the probability is at most ε from the definition of ε-ΔUniversal Hash Family H. Thus, $\Pr[H(a_{p_1}) + b_{p_1} = H(a_{p_2}) + b_{p_2}$ for some $p_1 \neq p_2 \in \mathbb{P}] \leq \binom{\sigma}{2}\varepsilon$. Hence $\Pr[(H(a_p) + b_p)$'s are distinct for all $p \in \mathbb{P}] \geq (1 - \binom{\sigma}{2}\varepsilon)$. ∎

A random function v is said to be an *uniform random permutation* if it is uniformly distributed on $\mathbf{Perm}(G)$. It is easy to check that for any distinct $x_1, \cdots, x_i$ and distinct $y_1, \cdots, y_i$, $\Pr[v(x_1) = y_1, \cdots, v(x_i) = y_i] = \frac{1}{N(N-1)\cdots(N-i+1)}$.

4.2 Hash-CBC or HCBC and a Simple Security Analysis

Given an ε-ΔUniversal Random Function H from G' to G and an independently distributed uniform random permutation v on G we define a random on-line function $X^{\mathbf{HCBC}}$ (or simply we write X^H), known as **HCBC** (or *Hash-CBC*). For $(m_1, \cdots, m_i) \in G^i$, define y_j's recursively as follows

$$y_j = v(H(y_{j-1}) + m_j),\ 1 \le j \le i \text{ and } y_0 = 0.$$

Now define $X^{\mathbf{H}}(m_1 \cdots m_i) = y_i$. Thus, the corresponding online cipher is $f_{\mathbf{H}}(m_1, \cdots, m_i) = (y_1, \cdots, y_i)$. An illustration of **HCBC** is given in Figure 1. It is easy to check that $X^{\mathbf{H}}$ is a random on-line function.

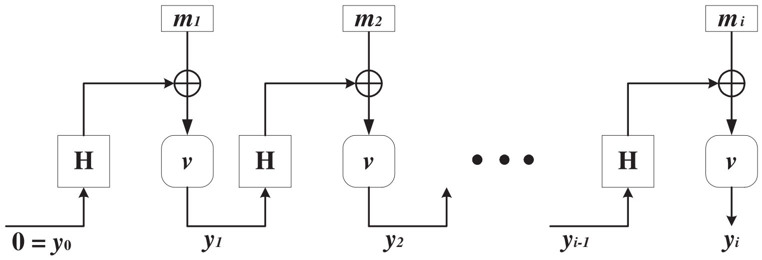

Figure 1: Hash-CBC online function

Theorem 4.2. *Suppose y_p's are non zero distinct elements from G for all $p \in \mathbb{P} = \mathbb{P}[M_1, \cdots, M_k]$. Let $\sigma = |\mathbb{P}|$, then*

$$\Pr[\ X^{\mathbf{H}}(p) = y_p, \forall p \in \mathbb{P}\ ] \ge \frac{1-\binom{\sigma}{2}\varepsilon}{N(N-1)\cdots(N-\sigma+1)}\ .$$

Proof. Let $a_p = y_{\mathbf{chop}(p)}$ and $b_p = \mathbf{last}(p)$ where $y_\lambda = 0$. Now check that for any $p_1 \ne p_2 \in \mathbb{P}$, $(a_{p_1}, b_{p_1}) \ne (a_{p_2}, b_{p_2})$. Suppose not, $a_{p_1} = a_{p_2}$ and $b_{p_1} = b_{p_2}$ then $y_{\mathbf{chop}(p_1)} = y_{\mathbf{chop}(p_2)}$ and $\mathbf{last}(p_1) = \mathbf{last}(p_2)$. Since y_p's are nonzero distinct $\mathbf{chop}(p_1) = \mathbf{chop}(p_2)$ and hence $p_1 = p_2$. Let D be the event that for all $p \in \mathbb{P}$, $(H(y_{\mathbf{chop}(p)}) + \mathbf{last}(p))$'s are distinct. Now by Lemma 4.1, $\Pr[D] \ge 1 - \frac{\sigma(\sigma-1)\varepsilon}{2}$. We also denote the set of all functions H_0 from G to G such that $(H_0(y_{\mathbf{chop}(p)}) + \mathbf{last}(p))$'s are distinct. Now,

$$\begin{aligned}
&\Pr[X^{\mathbf{H}}(p) = y_p, \forall p \in \mathbb{P}]\\
&\ge \Pr[X^{\mathbf{H}}(p) = y_p, \forall p \in \mathbb{P}\ \wedge\ D]\\
&= \Pr[v(H(y_{\mathbf{chop}(p)}) + \mathbf{last}(p)) = y_p \wedge D]\\
&= \textstyle\sum_{H_0 \in D} \Pr[v(H_0(y_{\mathbf{chop}(p)}) + \mathbf{last}(p)) = y_p \forall p \in \mathbb{P}] \Pr[H = H_0] \text{ (since } H \text{ and } v \text{ are independent)}\\
&= \frac{\Pr[D]}{N(N-1)\cdots(N-\sigma+1)}
\end{aligned}$$

(since for any $H_0 \in D$, $(H_0(y_{\mathbf{chop}(p)}) + \mathbf{last}(p))$'s are distinct and y_p's are distinct and hence $\Pr[v(H_0(y_{\mathbf{chop}(p)}) + \mathbf{last}(p)) = y_p \forall p \in \mathbb{P}] = \frac{1}{N(N-1)\cdots(N-\sigma+1)}$)

$$\ge \frac{(1-\binom{\sigma}{2}\varepsilon)}{N(N-1)\cdots(N-\sigma+1)}.$$

■

Corollary 4.3. $\mathbf{Adv}_{X^H, U}(\mathcal{D}) \le \binom{\sigma}{2}\varepsilon + \frac{\sigma(\sigma+1)}{N}$ *for any chosen plain text distinguisher $\mathcal{D}$ whose runtime is at most t and makes at most k queries having at most σ many blocks.*

Proof. The result follows from Theorem 3.2, 4.2 and Lemma 3.3. ■

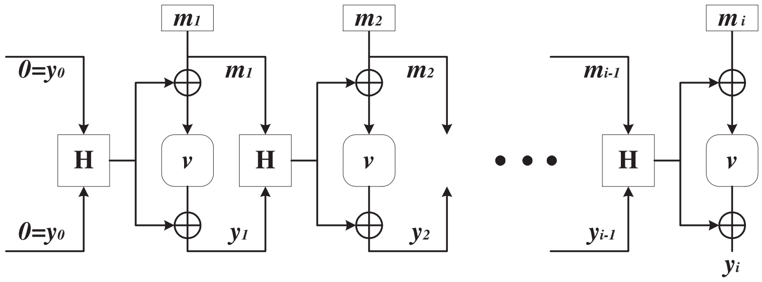

Figure 2: HPCBC online function

4.3 HPCBC and a Simple Security Analysis

Let $H : G^2 \to G$ be an ε-ΔUniversal Hash Family and v be an independently distributed uniform random permutation G. We define a random online function $X^{\mathbf{HPCBC}}$ (or simply we write X^P), known as **HPCBC** (or *Hash-PCBC*). For $(m_1, \cdots, m_i) \in G^i$, define y_j's recursively as follows

$$y_j = v(H(m_{j-1}, y_{j-1}) + m_j) + H(m_{j-1}, y_{j-1}),$$

$1 \le j \le i$ and $m_0 = y_0 = 0$. Now define $X^P(m_1 \cdots m_i) = y_i$. Thus, the corresponding online cipher is $f_{\mathbf{HPCBC}}(m_1, \cdots, m_i) = (y_1, \cdots, y_i)$. An illustration of **HPCBC** is given in Figure 2. It is easy to check that X^P is a random online function. Let $\mathbb{P}$ be a prefix closed set with $|\mathbb{P}| = \sigma$ and $x_p = \mathbf{last}(p)$, $x_\lambda = 0$.

Theorem 4.4. *For each $p \in \mathbb{P}$, let $y_p \in G$ such that (x_p, y_p)'s are distinct and not equal to $(0,0)$.*

$$\mathbf{Pr}[X^P(p) = y_p, p \in \mathbb{P}] \ge \frac{1 - \sigma(\sigma-1)\varepsilon}{N(N-1)\cdots(N-\sigma+1)}.$$

Proof. The proof of the theorem is same as that of Theorem 4.2 except in the definition of the event D. The event D denotes that both inputs and outputs of v are distinct. More precisely, $D = D_1 \cap D_2$ where $D_1 = \{H_0 : (H_0(x_{\mathbf{chop}(p)}, y_{\mathbf{chop}(p)}) + x_p)\text{'s are distinct}\}$ and $D_2 = \{H_0 : (H_0(x_{\mathbf{chop}(p)}, y_{\mathbf{chop}(p)}) + y_p)\text{'s are distinct}\}$.

Define, $a_p = (x_{\mathbf{chop}(p)}, y_{\mathbf{chop}(p)})$ and $b_p = x_p$. Now note that the pairs (a_p, b_p)'s are distinct. If not, then for some p_1, p_2, $\mathbf{last}(p_1) = x_{p_1} = x_{p_2} = \mathbf{last}(p_2)$ and $(x_{\mathbf{chop}(p_1)}, y_{\mathbf{chop}(p_1)}) = (x_{\mathbf{chop}(p_2)}, y_{\mathbf{chop}(p_2)})$ and hence $\mathbf{chop}(p_1) = \mathbf{chop}(p_2)$. Thus, $p_1 = p_2$. Now, by Lemma 4.1, $\Pr[\overline{D_1}] \le \varepsilon\binom{\sigma}{2}$.

Similarly, $\Pr[\overline{D_2}] \le \varepsilon\binom{\sigma}{2}$ and hence $\Pr[D_1 \cap D_2] \ge 1 - \varepsilon\sigma(\sigma-1)$. Given D is true, note that all inputs and outputs are distinct and hence $\Pr[X^P(p) = y_p, \forall p \in \mathbb{P}] \ge \frac{(1-\varepsilon\sigma(\sigma-1)}{N(N-1)\cdots(N-\sigma+1)}$. ■

Corollary 4.5. $\mathbf{Adv}_{X^H, U}(t, k, \sigma) \le \sigma(\sigma-1)\varepsilon + \sigma(\sigma+1)/N$.

Proof. The result follows from Theorem 3.2, 4.4 and Lemma 3.4. ■

4.4 A flaw in original proof of security bound [1]

Let D be the event such that all inputs to v are distinct while computing HCBC for different queries. $\overline{C}_2$ denotes that all responses are non zero and $\overline{C}_1$ denotes the event that all responses are nonzero distinct. In [1], they have used the following claim to prove the security of HCBC and HPCBC.

Claim 6.5 [1] For any y_p's, $\mathbf{Pr}[X(p) = y_p | D \wedge \overline{C}_2] = \mathbf{Pr}[X^U(p) = y_p \ \forall\ p | \overline{C}_1]$.

Intuitively, it says that given that all outputs are nonzero distinct the distribution of outputs is uniform on the set of all nonzero distinct elements when adversary interacts with uniform random online function. This is absolutely true.

According to their clam [1], given that "all inputs to v are distinct (hence all outputs are distinct with probability one) and the outputs are nonzero", the probability distribution of output is uniformly distributed on the set of all nonzero distinct elements. We show that this is not true in general and in fact it depends on the distribution due to the hash family H. Note that, the fact that H is an ε-ΔUniversal Random Hash Family is used on the computation of probabilities of events D and C_2 and not on the above statement. Suppose $\mathbb{P} = \{x_1, (x_1, x_2)\}$, and $Y_1 = X^H(x_1)$ and $Y_2 = X^H(x_1, x_2)$. If H is some random function such that $H(0) + x_1 = H(1) + x_1$ with probability one then Y_1 can not be one given D (i.e., $H(0) + x_1$ and $H(Y_1) + x_2$ are distinct). Thus for any pairs of non zero distinct values (y_1, y_2) such that $y_1 = 1$, $\Pr[Y_1 = y_1, Y_2 = y_2 | D \wedge \overline{C}_2] = 0$. Thus, the claim is false.

One can say that the random function H is not ε-ΔUniversal Random Function for any $\varepsilon < 1$. But it is clear that the $\Pr[Y_1 = y_1, Y_2 = y_2 | D \wedge \overline{C}_2]$ depends on the probability that $\Pr[H(0) + x_1 = H(y_1) + x_2]$ which may not be equal for all choices of y_1 and y_2. Thus the claim need not be true. But it is intuitively clear that these two distributions are very close. In this paper, we have proved a sort of closeness of two distributions and use this closeness to bound the advantage. We provide exact computation of the probabilities in the claim and see what happened to the quantities.

1. $\overline{C}_1 : Y_1, Y_2 \neq 0 \wedge Y_1 \neq Y_2$. Thus, $\mathbf{Pr}[X^U(x_1) = y_1, X^U(x_1, x_2) = y_2 | \overline{C}_1] = 1/N(N-1)$ whenever y_1, y_2 are nonzero distinct, otherwise the probability is zero.
2. $D : Z_1 = H(0) \oplus x_1 \neq H(Y_1) \oplus x_2 = Z_2$.
3. $\overline{C}_2 : Y_1 := v(H(0) \oplus x_1) = v(Z_1) \neq 0, v(H(Y_1) \oplus x_2) = v(Z_2) \neq 0$.

Let y_1, y_2 be nonzero distinct elements from G. Now, $\mathbf{Pr}[v(H(0) \oplus x_1) = y_1, v(H(Y_1) \oplus x_2) = y_2 | D \wedge \overline{C}_2]$

$$= \frac{\mathbf{Pr}[v(H(0) \oplus x_1) = y_1, v(H(Y_1) \oplus x_2) = y_2, D \wedge \overline{C}_2]}{\mathbf{Pr}[D \wedge \overline{C}_2]}.$$

Now, $\mathbf{Pr}[v(H(0) \oplus x_1) = y_1, v(H(y_1) \oplus x_2) = y_2, H(0) \oplus x_1 \neq H(Y_1) \oplus x_2]$

$$= \sum_{h_0, h_1 : h_0 \oplus x_1 \neq h_1 \oplus x_2} \mathbf{Pr}[v(h_0 \oplus x_1) = y_1, v(h_1 \oplus x_2) = y_2] \mathbf{Pr}[H(0) = h_0, H(y_1) = h_1]$$

$$= \mathbf{Pr}[H(0) \oplus x_1 \neq H(y_1) \oplus x_2]/N(N-1) = \epsilon(0, y_1, x_2 \oplus x_1)/N(N-1)$$

where $\epsilon(0, y, a)$ denotes the probability $\Pr[H(0) \oplus H(y) \neq a]$. Now, $\Pr[D \wedge \overline{C}_2] = \Pr[Y_1 := v(H(0) \oplus x_1) \neq 0,\ v(H(Y_1) \oplus x_2) \neq 0,\ H(Y_1) \oplus x_2 \neq H(0) \oplus x_1]$

$$= \sum_{y, h_0, h_1 \ :\ y \neq 0, h_1 \neq h_0 \oplus a} \Pr[v(h_0 \oplus x_1) = y, H(0) = h_0, H(y) = h_1, v(h_1 \oplus x_2) \neq 0]$$

$$= \frac{(N-2)}{N(N-1)} \sum_{y \neq 0, h_0, h_1 \ : h_1 \neq h_0 \oplus a} \Pr[H(0) = h_0, H(y) = h_1]$$

$$= \frac{N-2}{N(N-1)} \sum_{y \ : \ y \neq 0} \epsilon(0, y, a).$$

Thus, $\mathbf{Pr}[v(H(0) \oplus x_1) = y_1, v(H(y_1) \oplus x_2) = y_2 | D \wedge \overline{C}_2]$

$$= \frac{\epsilon(0, y_1, a)}{(N-2) \sum_{y \ : \ y \neq 0} \epsilon(0, y, a)}$$

$$\neq \frac{1}{N(N-1)} \text{ (in general).}$$

A similar flaw can be observed in the Claim 8.6 of [1] where the chosen cipher text security is considered.

5 Efficient and Secure Variants of HPCBC Online Cipher

5.1 First Modification of HPCBC or MHPCBC-1

In this section we propose an online cipher which is a variant of PHCBC. Suppose $H_1 : G \to G$ is an ε-ΔUniversal Hash Family. Define, $H : G^2 \to G$ such that $H(x, y) = H_1(x + y)$. Clearly, it is not an universal hash family. Still, the online function HPCBC based on H is secure. Call this by MHPCBC-1, Modified Hash-PCBC-1 (denoted by X^{M_1}) (see Figure 3).

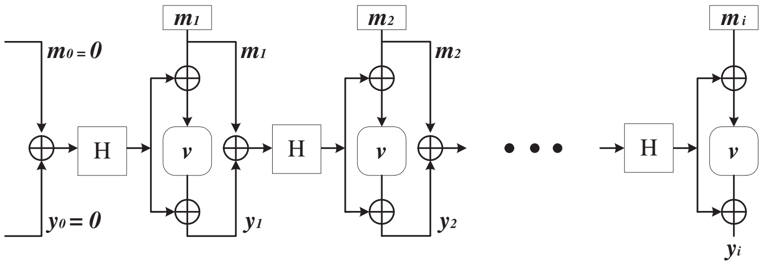

Figure 3: HPCBC online function

Theorem 5.1. *For all $p \in \mathbb{P}, y_p \in G$ such that $(x_p + y_p)$'s are nonzero distinct we have,*

$$\Pr[X^{M_1}(p) = y_p, \forall p \in \mathbb{P}] \geq \frac{1 - \varepsilon\sigma(\sigma - 1)}{N(N-1)\cdots(N-\sigma+1)}.$$

Proof. We denote $\mathbf{last}(p)$ by x_p. Let $D = D_1 \cap D_2$ where $D_1 = \{H_0 : (H_0(x_{\mathbf{chop}(p)} + y_{\mathbf{chop}(p)}) + x_p)\text{'s are distinct}\}$ and $D_2 = \{H_0 : (H_0(x_{\mathbf{chop}(p)} + y_{\mathbf{chop}(p)}) + y_p)\text{'s are distinct}\}$. Define, $a_p = x_{\mathbf{chop}(p)} + y_{\mathbf{chop}(p)}$ and $b_p = x_p$. Now note that the pairs (a_p, b_p)'s are distinct. If not, then for some p_1, p_2, $\mathbf{last}(p_1) = x_{p_1} = x_{p_2} = \mathbf{last}(p_2)$ and $x_{\mathbf{chop}(p_1)} + y_{\mathbf{chop}(p_1)} = x_{\mathbf{chop}(p_2)} + y_{\mathbf{chop}(p_2)}$ and hence $\mathbf{chop}(p_1) = \mathbf{chop}(p_2)$. Thus, $p_1 = p_2$. Now, by Lemma 4.1, $\Pr[\overline{D_1}] \leq \varepsilon\binom{\sigma}{2}$. Similarly, $\Pr[\overline{D_2}] \leq \varepsilon\binom{\sigma}{2}$ and hence $\Pr[D] \geq 1 - \varepsilon\sigma(\sigma - 1)$. Like in the proof of the Theorem 4.2 and Theorem 4.4, we have $\Pr[X^{M_1}(p) = y_p, \forall p \in \mathbb{P}] \geq \frac{1-\varepsilon\sigma(\sigma-1)}{N(N-1)\cdots(N-\sigma+1)}$. ∎

Corollary 5.2. $\mathbf{Adv}_{X^{M_1},U}(t,k,\sigma) \le \varepsilon\sigma(\sigma-1) + \sigma(\sigma+1)/N$.

Proof. The result follows from Theorem 3.2, 5.1 and Lemma 3.5. ■

5.2 Second Variant MHPCBC-2, without Universal Hash Family

Note that, to design an $\frac{1}{N}$-ΔUniversal hash family from G to G we need key space of size at least N [19]. The key space has size at least N^2 for $1/N$-ΔUniversal hash family from G^2 to G. So, besides keys for pseudo random permutation v, we need log$N$ (or 2log$N$) key size for HCBC, first variant of HPCBC i.e., MHPCBC-1 (or HPCBC respectively). Now we propose a construction where we can avoid the universal hash families at the same time the extra keys. In this definition, we use $H : G^2 \to G$ such that $H(x,y) = v(x+y)$. An illustration is given in Figure 4.

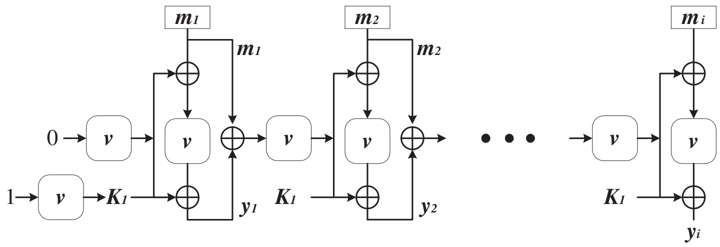

Figure 4: MHPCBC-2 online function, $K_1 = v(1)$.

Given $(m_1, \cdots, m_\ell)$ we first compute an inner secret key $K_1 = v(1)$ and then we compute recursively y_i as follows :

$$y_i = v\big(v(y_{i-1} + m_{i-1}) + m_i\big) + K_1 + v(y_{i-1} + m_{i-1}),\ y_0 = m_0 = 0.$$

We write $X^{M_2}(m_1, \cdots, m_\ell) = y_\ell$ (the output of online function MHPCBC-2). The random online cipher is defined in a natural way. Let $\mathbb{P}$ be the set of inputs to X^{M_2} for all queries of $\widehat{\mathcal{D}}$ and $X^{M_2} = y_p$'s are the outputs. Note that the original queries can be either $\overline{X}(\cdot, 1)$ or $\overline{X}(\cdot, -1)$.

- Now the set of all inputs of v in all computations of $X^{M_2}(p), p \in \mathbb{P}$ are $0, 1, y_{\mathbf{chop}(p)} + x_{\mathbf{chop}(p)}, z_p + x_p$ where $x_p = \mathbf{last}(p)$ and $z_{\mathbf{chop}(p)} = v(y_{\mathbf{chop}(p)} + x_{\mathbf{chop}(p)})$ and $x_\lambda = y_\lambda = 0$.
- Similarly all outputs of v are $K_1 = v(1), z_{\mathbf{chop}(p)}, z_{\mathbf{chop}(p)} + y_p + K_1$. Let $x_p, y_p \in G$ such that $(x_p + y_p)$'s are distinct elements of G and not equal to 0 and 1.

We say a tuple of elements $(z_p)_{p \in \mathbb{P}_1}$ is good if all inputs are distinct and all outputs are distinct where $\mathbb{P}_1 = \{\mathbf{chop}(p) : p \in \mathbb{P}\} \cup \{K_1\}$ where z_{K_1} denotes the values of K_1 and $\sigma_1 = |\mathbb{P}_1|$. The total number of tuples are N^{σ_1}. We give an upper bound of the number of tuples which are not good.

- $z_p = x_p$ or $z_p = x_p + 1$ or $z_p = x_p + z_{p_1} + x_{p_1}$ or $z_p = x_p + x_{\mathbf{chop}(p_1)} + y_{\mathbf{chop}(p_1)}$ implies that the inputs are not distinct. There are at most $N^{\sigma_1 - 1}(2\sigma + 2\sigma(\sigma-1))$.
- Similarly, for outputs we have $N^{\sigma_1 - 1}2\sigma(2\sigma+1)$ tuples. Thus, the number of good tuples is at least $N^{\sigma_1+1}(1 - \frac{4\sigma^2+2\sigma}{N})$.

For each such good tuple, the probability that $v(x_p + y_p) = z_p, v(1) = z_{K_1}$ and $v(z_{\mathbf{chop}(p)+x_p}) = z_{\mathbf{chop}(p)} + y_p + z_{K_1}$ where $p \in \mathbb{P}_1$ is $\frac{1}{N(N-1)\cdots(N-\sigma-\sigma_1+1)}$. Thus the probability

$$\Pr[X^{M_2}(p) = y_p \forall p \in \mathbb{P}] \geq \frac{(1 - \frac{4\sigma^2+2\sigma}{N})}{N(N-1)\cdots(N-\sigma)} \geq (1-\varepsilon_1)\Pr[U(p) = y_p \forall p \in \mathbb{P}]$$

where $\varepsilon_1 = \frac{4\sigma^2+2\sigma}{N}$. This is true for all choices of y_p and p such that $(x_p + y_p)$'s are distinct and not equal to 0 and 1. Now note that $\Pr[v_U \in B_4] \leq (\sigma+1)(\sigma+2)/N$ where B_4 denotes the set of all views such that $(x_p + y_p)$'s are not distinct or equal to either 0 or 1. The upper bound of the probability is proved in Lemma 3.6. Thus, we have the following bound of advantage for MHPCBC-2 by applying Theorem 3.2.

Theorem 5.3. $\mathbf{Adv}_{X^{M_2},U}(t,k,\sigma) \leq \frac{4\sigma^2+2\sigma+(\sigma+1)(\sigma+2)}{N} \leq \frac{6\sigma^2}{N}$.

6 Conclusion

This paper provides a framework to prove the indistinguishability between two classes of functions called random functions. We consider the object known as online cipher and showed a simple security proof for for HCBC and HPCBC which were proposed by Bellare *et al.* [1]. Unfortunately, their proof contains some serious mistakes. We explain those mistakes in this paper. At the end we come up with two new proposal of secure online cipher which are much more efficient and needs less keys than HPCBC. We hope the approach taken in this paper will be helpful for making other similar security analysis.

Acknowledgement

I would like to acknowledge Professor Palash Sarkar who has inspired me to write this paper.

References

[1] M. Bellare, A. Boldyreva, L. Knudsen and C. Namprempre. On-Line Ciphers and the Hash-CBC constructions. Advances in Cryptology - CRYPTO 2001. Lecture Notes in Computer Science, Volume **2139**, pp 292-309.

[2] M. Bellare, K. Pietrzak and P. Rogaway. Improved Security Analysis for CBC MACs. Advances in Cryptology - CRYPTO 2005. Lecture Notes in Computer Science, Volume **3621**, pp 527-545.

[3] M. Bellare, J. Killan and P. Rogaway. The security of the cipher block chanining Message Authentication Code. Advances in Cryptology - CRYPTO 1994. Lecture Notes in Computer Science, Volume **839**, pp 341-358.

[4] Daniel J. Bernstein. A short proof of the unpredictability of cipher block chaining (2005). URL: http://cr.yp.to/papers.html#easycbc.

[5] J. Black and P. Rogaway. CBC MACs for arbitrary length messages. Advances in Cryptology - CRYPTO 2000. Lecture Notes in Computer Science, Volume **1880**, pp 197-215.

[6] J. Black and P. Rogaway. A Block-Cipher Mode of Operations for Parallelizable Message Authentication. Advances in Cryptology - Eurocrypt 2002. Lecture Notes in Computer Science, Volume **2332**, pp 384-397.

[7] J. Daemen and V. Rijmen. Resistance Against Implementation Attacks. A Comparative Study of the AES Proposals. In Proceedings of the Second AES Candidate Conference (AES2), Rome, Italy, March 1999. Available at http://csrc.nist.gov/encryption/aes/aes_ home.htm.

[8] Alison L. Gibbs and Francis Edward Su. On Choosing and Bounding Probability Metrics, Jan 2002.

[9] L. Knudsen. Block chaining modes of operation. Symmetric Key Block Cipher Modes of Operation Workshop, `http://csrc.nist.gov/encryption/modes/workshop1/`, Oct. 2000.

[10] H. Krawczyk. LFSR-based hashing and authenticating. Advances in Cryptology, CRYPTO 1994, Lecture Notes in Computer Science, Volume **839**, pp 129-139, Springer-Verlag 1994.

[11] T. Iwata and K. Kurosawa. OMAC : One-Key CBC MAC. Fast Software Encryption, 10th International Workshop, FSE 2003. Lecture Notes in Computer Science, Volume **2887**, pp 129-153.

[12] C. S. Jutla. PRF Domain Extension using DAG. Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006. Lecture Notes in Computer Science, Volume **3876** pp 561-580.

[13] K. Kurosawa and T. Iwata. TMAC : Two-Key CBC MAC. Topics in Cryptology - CT-RSA 2003: The Cryptographers' Track at the RSA Conference 2003. Lecture Notes in Computer Science, Volume **2612**, pp 33-49.

[14] M. Luby and C. Rackoff. How to construct pseudo-random permutations from pseudo-random functions. Advances in Cryptology, CRYPTO' 85, Lecture Notes in Computer Science, Volume **218**, pp 447, Springer-Verlag 1985.

[15] C. Meyer and Matyas. A new direction in Computer Data Security. John Wiley & Sons, 1982.

[16] M. Nandi. A Simple and Unified Method of Proving Indistinguishability. Indocrypt 2006, Lecture Notes in Computer Science, Volume **4329**, pp 317-334.

[17] W. Nevelsteen and B. Preneel. Software performance of universal hash functions. Advances in Cryptology, EUROCRYPT '99, Lecture Notes in Computer Science, Volume **1592**, pp 24-41, Springer-Verlag 1999.

[18] P. Rogaway. Bucket Hashing and Its Application to Fast Message Authentication. Advances in Cryptology, CRYPTO 1995, Lecture Notes in Computer Science, Volume **963**, pp 29-42, Spronger-Verlag, 1995.

[19] D. R. Stinson. On the connections between universal hashing, combinatorial designs and error-correcting codes. Congressus Numerantium **114**, 1996, pp 7-27.