A Simple and Unified Method of Proving Unpredictability

Mridul Nandi
School of Computer Science
University of Waterloo
Ontario, Canada N2L 3G1

August 3, 2006

Abstract

Recently Bernstein [4] has provided a simpler proof of unpredictability of CBC construction [3] which is giving insight of the construction. Unpredictability of any function intuitively means that the function behaves very closely to a uniform random function. In this paper we make a unifying and simple approach to prove unpredictability of many existing constructions. We first revisit Bernstein's proof. Using this idea we can show a simpler proof of unpredictability of a class of DAG based construction [7], XCBC [5], TMAC [8], OMAC [9] and PMAC [6]. We also provide a simpler proof for stronger bound of CBC [1] and a simpler proof of security of on-line Hash-CBC [2]. We note that there is a flaw in the security proof of Hash-CBC in [2]. This paper will help to understand security analysis of unpredictability of many constructions in a simpler way.

1 Introduction.

This paper deals how one can obtain a simple proof for a bound of distinguishing advantage of two classes of object, mainly two classes of functions. We consider several constructions and show how simply the distinguishing advantage can be obtained. Here we mostly consider distinguishing attack of existing constructions with popularly known *random function* (in this paper, we term it as ***uniform random function*** [4]). *Unpredictability* of a construction intuitively means that there is no efficient distinguisher which distinguishes this from the uniform random function. Bernstein has provided a simple proof of unpredictability of **CBC-MAC** (*Cipher Block Chaining-Message Authentication Code*) [4] which is the main motivation of this paper. We first revisit his proof [4] and show how simply one can extend the proof idea for a class of **DAG** (*Directed Acyclic Graph*) based general construction due to Jutla [7]. This class contains many constructions including CBC and a variant of PMAC [6]. We give a simpler proof of partial result of improved security analysis of CBC-MAC [1]. We also study distinguishing advantage with a different class known as *uniform random on-line function* introduced in Crypto 2001 [2]. We show that same idea of proof is also applicable in this scenario and we obtain a simpler proof of Hash-CBC construction [2]. The idea of all these proofs is based on statistical distribution of the *view of the distinguisher.* Thus, it gives information theoretic security and hence the security bound holds for computationally unbounded distinguishers also.

This simple idea can help to understand better about the insight of the construction and can help to come up with very nice constructions and results. For example, we modify slightly the the

DAG based class due to Jutla [7], so that it will include all known constructions like XCBC [5], TMAC [8], OMAC [9], PMAC [6] etc.

Organization of the paper. In this paper, we first build mathematics for the security bound of the distinguisher in Section 2 which would be used throughout this paper. Then we rewrite the simple proof of security of CBC given by D. J. Bernstein in Section 3. In Section 4, we generalize his idea of proof to have a proof for a general class proposed by Jutla. We see that security of arbitrary length MAC construction like XCBC, TMAC, OMAC, PMAC etc. can be derived from it. In Section 5, we give a simple proof of a part of result proving the improved bound of CBC given by Bellare et. al. [1]. In Section 6 we provide a simpler proof of security of Hash-CBC. We note that in the original paper there is a flaw in the proof. Finally we conclude.

2 Mathematics for security proof in Distinguishing Attack.

2.1 Different Notion of Distances and Its Cryptographic significance.

In this section, we define two notions of distances of two random variables. Then we state the relationship between them.

(1) Statistical Distance :

Let X and Y be two random variables taking values on a finite set S. We define *statistical distance* between two random variables by

$$\mathrm{d_{stat}}(X,Y) := \max_{T\subset S} \big|\Pr[X\in T] - \Pr[Y\in T]\big|.$$

Note that, $\Pr[X\in T]-\Pr[Y\in T] = \Pr[Y\notin T]-\Pr[X\notin T]$ and hence $\mathrm{d_{stat}}(X,Y) = \max_{T\subset S}\Pr[X\in T]-\Pr[Y\in T]$. It measures the distance between the distribution of the random variables. In fact, it is really a *metric* or *distance function* on the set of all distributions on S. It measures how close their distributions are. For identically distributed random variables X and Y, $\mathrm{d_{stat}}(X,Y)=0$ and if the random variables are disjoint[1] then the statistical distance is one. In all other cases it lies between zero and one. Now we prove an equivalent definition of statistical distance and study some standard examples.

Lemma 2.1. $\mathrm{d_{stat}}(X,Y) = \Pr[X\in T_0] - \Pr[Y\in T_0] = \frac{1}{2}\times\sum_{a\in S}\big|\Pr[X=a]-\Pr[Y=a]\big|$, *where* $T_0 = \{a\in S : \Pr[X=a]\geq\Pr[Y=a]\}$.

Proof. For T_0 as given in the Lemma 2.1, it is easy to see that

$$\sum_{a\in S}\big|\Pr[X=a]-\Pr[Y=a]\big| = 2\times\big(\Pr[X\in T_0]-\Pr[Y\in T_0]\big).$$

For any $T\subset S$, $\quad 2\times(\Pr[X\in T]-\Pr[Y\in T])$

$$= \sum_{a\in T}\big(\Pr[X=a]-\Pr[Y=a]\big) - \sum_{a\notin T}\big(\Pr[X=a]-\Pr[Y=a]\big)$$
$$\leq \sum_{a\in S}|\Pr[X=a]-\Pr[Y=a]|. \qquad \blacksquare$$

[1] X and Y are said to be disjoint if X occurs with some positive probability then Y does occur with probability zero and vice versa. More precisely, there exists a subset T such that $\Pr[X\in T]=1$ and $\Pr[Y\in T]=0$

Example 2.1. *Let X and Y be uniform distributions on S and $T \subset S$ respectively. Then by Lemma 2.1,* $\mathrm{d_{stat}}(X,Y) = \frac{1}{2} \times \big((\frac{1}{|T|} - \frac{1}{|S|}) \times |T| + \frac{|S|-|T|}{|S|}\big) = 1 - \frac{|T|}{|S|}$. *Thus, if size of T is very close to S then statistical distance is also very close to zero. On the other hand, if size of T is negligible compare to that of S then statistical distance is close to one.*

Example 2.2. *Let* $S = \mathrm{Func}(G,G)$ *where* $\mathrm{Func}(H,G)$ *denotes the set of all functions from H to G. Let* $T = \mathrm{Func}^{\mathrm{inj}}(G,G)$ *be the subset containing all injective functions (or permutation since domain and range are same). We say u (or v) is a* uniform random function *(or* uniform random injective function*) if it is a uniform distribution on S (or T respectively). Thus from Example 2.1 we know that* $\mathrm{d_{stat}}(u,v) = 1 - \frac{N!}{N^N}$ *which is very close to one for large N, where $|G| = N$.*

Example 2.3. *Given any distinct $x_1, \cdots, x_k \in G$, let the k-*sampling output *of u be* $(u(x_1),\cdots,u(x_k))$ *and denoted as $u[k](x_1, \cdots, x_k)$. Let $X = (u(x_1), \cdots, u(x_k))$ and $Y = (v(x_1),\cdots,v(x_k))$. Then we can see that X has a uniform distribution on $S = G^k$ and Y has a uniform distribution on $T = G[k] := \{(y_1, \cdots, y_k) \in G^k : y_i$'s are distinct$\}$ and hence (again by Example 2.1)* $\mathrm{d_{stat}}(X,Y) = 1 - \frac{N(N-1)\cdots(N-k+1)}{N^k} \approx 1 - \exp^{-k(k-1)/2N}$. *Here we note that if $k << \sqrt{N}$ then the statistical distance is very close to zero.*

Now, we prove two results which will help to give an upper bound of statistical distance of two distributions. If the probability of the event $\{X = a\}$ is not small compare to that of $\{Y = a\}$ for all choices of a (or on a set with high probability) then the statistical distance is also small. More precisely, we have the following two lemmas.

Lemma 2.2. *Let X and Y be two random variable taking values on S and $\epsilon > 0$. If* $\Pr[X = a] \geq (1-\epsilon) \times \Pr[Y = a]$, $\forall a \in S$ *or* $\Pr[X = a] \leq (1+\epsilon) \times \Pr[Y = a]$, $\forall a \in S$ *then* $\mathrm{d_{stat}}(X,Y) \leq \epsilon$.

Proof. For any subset $T \subset S$, $\Pr[X \in T] \geq (1-\epsilon) \times \Pr[Y \in T]$ since $\Pr[X = a] \geq (1-\epsilon) \times \Pr[Y = a]$ $\forall$ a. So, $\Pr[Y \in T] - \Pr[X \in T] \leq \epsilon \times \Pr[Y \in T] \leq \epsilon$. Thus, $\mathrm{d_{stat}}(X,Y) \leq \epsilon$. Similarly one can prove for the other case. ■

Lemma 2.3. *Let X and Y be two random variables taking values on S. Let for a subset $T \subset S$,* $\Pr[X = a] \geq (1-\epsilon_1) \times \Pr[Y = a]$, $\forall a \in T$ *and* $\Pr[Y \notin T] \leq \epsilon_2$ *then* $\mathrm{d_{stat}}(X,Y) \leq 2\epsilon_1 + 2\epsilon_2$.

Proof. For any subset $T_1 \subset T$, $\Pr[Y \in T_1] - \Pr[X \in T_1] \leq \epsilon_1 \times \Pr[Y \in T_1] \leq \epsilon_1$. From the given relation we also note that $\Pr[X \in T] \geq (1-\epsilon_1) \times \Pr[Y \in T]$. Thus, $\Pr[X \notin T] \leq \epsilon_1 + \epsilon_2 - \epsilon_1\epsilon_2$. Thus, $\mathrm{d_{stat}}(X,Y) \leq \epsilon_1 + \Pr[X \in \neg T] + \Pr[Y \in \neg T] \leq 2(\epsilon_1 + \epsilon_2)$. ■

(2) Computational Distance :

The statistical distance is also popularly known as information theoretic distance. In cryptography, there is another notion of distance, known as *computational distance*. Let $\mathcal{A}(\cdot)$ be a *probabilistic algorithm* which runs with an input $a \in S$ and giving output 0 or 1. Define, $\mathcal{A}$-*distance* between X and Y as follows;

$$\mathrm{d}^{\mathcal{A}}(X,Y) = \big|\Pr[\mathcal{A}(X) = 1] - \Pr[\mathcal{A}(Y) = 1]\big|.$$

Here, $\mathcal{A}(X)$ means the distribution of output of $\mathcal{A}(z)$ where z follows the distribution of X. Similarly for $\mathcal{A}(Y)$. As $\mathcal{A}$ is a probabilistic algorithm it can use a string r chosen from some set $\mathcal{R}$ with a distribution which is *independent* with X and Y. So we consider that $\mathcal{A}$ is having two inputs $r \in \mathcal{R}$ and $z \in S$. We state a fact which shows a relationship between statistical and computational distances.

Lemma 2.4. *For any $\mathcal{A}$, $\mathrm{d}^{\mathcal{A}}(X,Y) \leq \mathrm{d}_{\mathrm{stat}}(X,Y)$. Conversely, there exists an algorithm $\mathcal{A}_0$ (may not be efficient) such that $\mathrm{d}^{\mathcal{A}_0}(X,Y) = \mathrm{d}_{\mathrm{stat}}(X,Y)$.*

Proof. Output of $\mathcal{A}$ is completely determined by a pair (r,z), where r is the random string chosen from $\mathcal{R}$ and z is the input. Let $S_{r_0} = \{a \in S : \mathcal{A}(r_0,a) = 1\}$. Thus, $\mathrm{d}^{\mathcal{A}}(X,Y)$

$= \big|\Pr[\mathcal{A}(r,X)=1] - \Pr[\mathcal{A}(r,Y)=1]\big|$

$= \big|\sum_{r_0 \in \mathcal{R}} \Pr[r=r_0]\big(\Pr[\mathcal{A}(r_0,X)=1 \mid r=r_0] - \Pr[\mathcal{A}(r_0,Y)=1 \mid r=r_0]\big)\big|$

$= \big|\sum_{r_0 \in \mathcal{R}} \Pr[r=r_0]\big(\Pr[\mathcal{A}(r_0,X)=1] - \Pr[\mathcal{A}(r_0,Y)=1]\big)\big|$

$= \big|\sum_{r_0 \in \mathcal{R}} \Pr[r=r_0]\big(\Pr[X \in S_{r_0}] - \Pr[Y \in S_{r_0}]\big)\big| \leq \mathrm{d}_{\mathrm{stat}}(X,Y)$.

The equality holds if $S_{r_0} = T_0$ as in Lemma 2.1. Thus, on input z, $\mathcal{A}_0$ computes the probability $\Pr[X=z]$, $\Pr[Y=z]$ and outputs 1 if $\Pr[X=z] \geq \Pr[Y=z]$, otherwise 0. Hence $\mathrm{d}^{\mathcal{A}_0}(X,Y) = \mathrm{d}_{\mathrm{stat}}(X,Y)$. ■

In the above proof note that $\mathcal{A}_0$ may not be efficient and does not use any random string. One can consider only deterministic algorithm when it has unbounded computational power. Intuitively, one can make computation for all random choices and choose the random string where it has the best performance. Later, we will show that we can ignore the random string while we distinguish two classes of functions by using unbounded computation.

2.2 Distinguisher of Families of Functions or Random Functions.

In this section we describe how a distinguisher can behave. We also show that how the advantage of the distinguisher can be obtained by computing the statistical distance of *view* of the distinguisher.

By random function we mean some distribution on the set $\mathrm{Func}(H,G)$, set of all functions from H to G. In Example 2.2, we have already defined two random functions, they are uniform random function and uniform random injective function. In cryptography, they are used as ideal candidates. In this paper we will also study another ideal function known as *uniform random on-line injective function*. We will define this in Section 6. Now we follow the notations used in Example 2.2 and 2.3. Let f be a random function. For each $\mathbf{x} = (x_1, \cdots, x_k) \in H[k]$, $f[k](\mathbf{x}) = (f(x_1), \cdots, f(x_k))$ follows the distribution induced by the distribution of f. More precisely, for any $\mathbf{y} = (y_1, \cdots, y_k) \in G^k$,

$$\Pr[f[k](\mathbf{x}) = \mathbf{y}] = \sum_{f_0 \in I} \Pr[f = f_0], \quad \text{where} \quad I := \{f \in \mathrm{Func}(H,G) : f[k](\mathbf{x}) = \mathbf{y}\}.$$

Let f and g be two random functions and a distinguisher $\mathcal{D}$ has a function oracle which can be either chosen from f or from g. Distinguisher is behaving as follows :

- First it chooses a random string r from $\mathcal{R}$.
- Based on r it makes oracle query say $x_1 := x_1(r) \in H$. It obtains a response $y_1 \in G$.
- Then it makes queries $x_2 = x_2(r, y_1) \in H$ and obtains response $y_2 \in G$ and so on.

Even if x_2 can depend on x_1, it is a function of r and y_1 since x_1 is a function of r only. Thus, x_i is a function of $(r, y_1, \cdots, y_{i-1})$. We say these functions $x_1, x_2, \cdots$ are *query functions* (or $\mathbf{x} = (x_1, \cdots, x_k)$ is k-query function) and the tuple $(y_1, \cdots, y_k) \in G^k$ is the *conditional view* of the distinguisher (condition on the random string r) where k is the number of queries. Note that the output of $\mathcal{D}$ is completely determined by the chosen random string r and the conditional view $(y_1, \cdots, y_k)$. We define the distinguishing advantage of $\mathcal{D}^O$ to distinguish between f and g

$$\mathrm{Adv}_{f,g}(\mathcal{D}) = |\Pr[\mathcal{D}^f = 1] - \Pr[\mathcal{D}^g = 1]|.$$

Define $\mathrm{d}_{f,g}(k) = \max_{\mathcal{D}}\mathrm{Adv}_{f,g}(\mathcal{D})$, where maximum is taken over all oracle algorithms $\mathcal{D}$ which make at most k queries. This denotes the maximum distinguishing advantage for two random functions f and g where the attacker is making at most k queries. Note that there is no restriction on the computational resources of $\mathcal{D}$. We can think $\mathcal{D}$ as a tuple of function $(x_1, \cdots, x_k, \mathcal{A})$ where x_i's are query functions and $\mathcal{A}$ is the final output function which takes input as $(r, y_1, \cdots, y_k)$. Denote this view without the random string $(y_1, \cdots, y_k)$ by $f[k]_{r,x_1,\cdots,x_k}$ or $g[k]_{r,x_1,\cdots,x_k}$ for the random function f and g respectively. Here, $\mathcal{A}$ is distinguishing two families of random variable $\{f[k]_{r,x_1,\cdots,x_k}\}_{r\in\mathcal{R}}$ and $\{g[k]_{r,x_1,\cdots,x_k}\}_{r\in\mathcal{R}}$. Thus,

$$\begin{aligned}\mathrm{Adv}_{f,g}(\mathcal{D}) &= \Big|\sum_{r\in\mathcal{R}}\Pr[\mathcal{A}(r, f[k]_{r,x_1,\cdots,x_k}) = 1] \times \Pr[r] - \sum_{r\in\mathcal{R}}\Pr[\mathcal{A}(r, g[k]_{r,x_1,\cdots,x_k}) = 1] \times \Pr[r]\Big| \\ &= \sum_{r\in\mathcal{R}}\Pr[r] \times \mathrm{d}^{\mathcal{A}}(f[k]_{r,x_1,\cdots,x_k}, g[k]_{r,x_1,\cdots,x_k}) \\ &\le \sum_{r\in\mathcal{R}}\Pr[r] \times \mathrm{d}_{\mathrm{stat}}(f[k]_{r,x_1,\cdots,x_k}, g[k]_{r,x_1,\cdots,x_k})\end{aligned}$$

So, given any probabilistic distinguisher $\mathcal{D} = (x_1, \cdots, x_k, \mathcal{A})$ one can define a deterministic distinguisher $\mathcal{D}_0 = (x_1, \cdots, x_k, \mathcal{A}_0)$ such that $\mathrm{Adv}_{f,g}(\mathcal{D}) \le \mathrm{Adv}_{f,g}(\mathcal{D}_0)$. Here, $\mathcal{D}_0$ chooses a random string r_0 with probability one (i.e., a deterministic algorithm) such that $\mathrm{d}_{\mathrm{stat}}(f[k]_{r_0,x_1,\cdots,x_k}, g[k]_{r_0,x_1,\cdots,x_k}) = \max_{r\in\mathcal{R}} \mathrm{d}_{\mathrm{stat}}(f[k]_{r,x_1,\cdots,x_k}, g[k]_{r,x_1,\cdots,x_k})$ and $\mathcal{A}_0$ behaves as in Lemma 2.4. Now we will make following assumptions in this paper.

Assumption 1 (Distinguishers are deterministic) : We assume that all distinguishing algorithms are deterministic. Thus, x_1 is a constat and x_i is a function of $(y_1, \cdots, y_{i-1})$.

Assumption 2 (Query functions are distinct) : To avoid complicity of notations we use the same notation x_i to denote the function as well as the output of the function. We will assume that all outputs of x_i's (or x_i as a functional value) are distinct (otherwise one can restrict on the set of distinct values of x_i).

Now we use the notation $f[k]_{x_1,\cdots,x_k}$ instead of $f[k]_{r,x_1,\cdots,x_k}$ to denote the view of the distinguisher. We can write that

$$d_{f,g}(k) = \max_{\mathbf{x}}\mathrm{d}_{\mathrm{stat}}(f[k]_{\mathbf{x}}, g[k]_{\mathbf{x}}),$$

where maximum is taken over all k-query functions $\mathbf{x} = (x_1, \cdots, x_k)$. Thus, to obtain an upper bound of $d_{f,g}(k)$, it would be enough to bound $\mathrm{d}_{\mathrm{stat}}(f[k]_{\mathbf{x}}, g[k]_{\mathbf{x}})$ for each k-query functions $\mathbf{x}$. The following theorem says how one can obtain this. This theorem has been stated and proved By D. J. Bernstein [4].

Theorem 2.5. *If $\Pr[f[k](\mathbf{a}) = \mathbf{y}] \ge (1-\epsilon) \times \Pr[g[k](\mathbf{a}) = \mathbf{y}]$ for each $\mathbf{a} \in H[k]$ and $\mathbf{y} \in G^k$, then for any k-query function $\mathbf{x} = (x_1, \cdots, x_k)$, $\mathrm{d}_{\mathrm{stat}}(f[k]_{\mathbf{x}},\ g[k]_{\mathbf{x}}) \le \epsilon$ and hence $\mathrm{d}_{f,g}(k) \le \epsilon$.*

Proof. $\Pr[f[k]_{x_1,\cdots,x_k} = (y_1,\cdots,y_k)]$

$= \Pr[f[k](a_1,\cdots,a_k) = (y_1,\cdots,y_k)]$ ($(a_1,\cdots,a_k)$ is uniquely determined by $(y_1,\cdots,y_k)$)

$\geq (1-\epsilon) \times \Pr[g[k](a_1,\cdots,a_k) = (y_1,\cdots,y_k)]$

$= (1-\epsilon) \times \Pr[g[k]_{x_1,\cdots,x_k} = (y_1,\cdots,y_k)]$, $\forall\ (y_1,\cdots,y_k) \in G^k$. The Theorem follows from Lemma 2.2. ■

3 A short proof of the unpredictability of CBC due to D. J. Bernstein [4].

Here, we rewrite the security proof of CBC based on uniform random function given by Bernstein [4]. We also show that the similar result can be obtained for uniform random injective function.

Let f be a function on a group $(G,+)$ (i.e, from $(G,+)$ to $(G,+)$) where $|G| = N$. For $m \geq 1$, define the iterated functions recursively as follow :

$$f^+(g_1,\cdots,g_m) := f^+_m(g_1,\cdots,g_m) = f(f^+_{m-1}(g_1,\cdots,g_{m-1}) + g_m),$$

where $g_i \in G$, $f^+_0() = f^+_0(\lambda) = 0$ and λ is the empty string. Let $\mathbf{x} = (x_1,\cdots,x_k) \in (G^m)^k$ and $(y_1,\cdots,y_k) \in G^k$ where $x_1,\cdots,x_k$ are distinct elements of G^m. We define $\mathbb{P} := \mathbb{P}(\mathbf{x}) \subset G\cup\cdots\cup G^m$, by the set of all non-empty prefixes of x_i's. Note that $\mathbb{P}(\mathbf{x}) \leq mk$ for any $\mathbf{x} \in (G^m)^k$. Let $\mathbb{P}_1 := \mathbb{P}_1(\mathbf{x}) = \mathbb{P}(\mathbf{x}) \setminus \{x_1,\cdots,x_k\}$.

Example 3.1. *Let $G = \mathbb{Z}_{100}$ and $\mathbf{x} = ((1,2,2),(1,2,3),(2,2,2))$ then $\mathbb{P}(\mathbf{x}) = \{1,2,(1,2),(2,2),$ $(1,2,2)$,$(1,2,3)$, $(2,2,2)\}$ and $\mathbb{P}_1(\mathbf{x}) = \{1,2,(1,2),(2,2)\}$.*

We fix any $\mathbf{x}$. Given any f, define the *intermediate induced output function* (or simply induced output function) $op_f : \mathbb{P}_1(\mathbf{x}) \to G$ as $op_f(p) = f^+(p)$. Any function from $\mathbb{P}_1(\mathbf{x})$ to G is called as output function. Note that all output functions may not be an induced output function. We characterize the output functions which are induced output functions. Given *op* define a *corresponding input function ip* : $\mathbb{P} \to G$ such that

$$\left.\begin{array}{rcll} ip(p) & = & op(\text{chop}(p)) + \text{last}(p) & \text{if } p \notin G \\ & = & p & \text{if } p \in G \end{array}\right\} \qquad (1)$$

where if $p = (q,g') \in G^i$, $\text{chop}(p) := q \in G^{i-1}$, $\text{last}(p) := g' \in G, i \geq 2$.

Lemma 3.1. *Let op be an output function and ip be its corresponding input function. An output function op is an induced output function if and only if $op(p_1) = op(p_2)$ whenever $ip(p_1) = ip(p_2)$. In particular, op is an induced output function if corresponding input function is injective (the above condition is vacuously true).*

Proof. Given any f, $op_f(p) = f^+(p) = f(ip(p))$ where ip is the corresponding input function of op_f. Thus, the converse of the statement is also true. Now we prove the forward implication of the Lemma. Given any op and its corresponding input function ip, we define

$$\left.\begin{array}{rcll} f(x) & = & op(p) & \text{if } ip(p) = x \\ & = & * & \text{otherwise} \end{array}\right\} \qquad (2)$$

Here, $*$ means that we can choose any arbitrary element from G. This is well defined as $ip(p_1) = ip(p_2) = x$ implies $op(p_1) = op(p_2)$. Recursively, one can check that $f^+(p) = op(p)$ and hence $op = op_f$. ■

Example 3.1. (contd.) Let $op(1) = op(1,2) = 99$, $op(2) = 1$ and $op(2,2) = 0$. Note that it satisfies the condition of above Lemma. For example, $op(1) = op(1,2)$ where, $ip((1,2)) = ip(1) = 1$. Thus for any f such that $f(1) = 99, f(2) = 1$ and $f(3) = 0$, $op_f = op$. Here note that $ip((1,2,2)) = 1$, $ip((1,2,2)) = 2$ and $ip((2,2,2)) = 2$. So, for this output function and for any f such that $op_f = op$, we have $f^+((1,2,2)) = 99$, $f^+((1,2,3)) = 1$ and $f^+((2,2,2)) = 1$.

Following lemma count the number of functions which induce a given induced output function.

Lemma 3.2. *Let op be an induced output function such that $|ip(\mathbb{P}_1)| = q$ where ip is the corresponding input function and $ip(\mathbb{P}_1) = \{ip(p) : p \in \mathbb{P}_1\}$ is the range of it. Then there are exactly N^{N-q} many f such that $op = op_f$.*

Proof. This is immediate from the construction of f in Equation 2. ■

Corollary 3.3. *If op is an output function such that corresponding input function ip is injective then there are $N^{N-|\mathbb{P}_1|}$ many f's such that $op_f = op$ and there are $N^{N-|\mathbb{P}_1|-k}$ many f's such that $op_f = op$ and $f^+[k](\mathbf{x}) = \mathbf{y}$.*

Example 3.1. (contd.) In this example, $ip(\mathbb{P}_1) = \{1,2,3\}$ and hence we have 100^{97} many f's such that $op_f = op$. More precisely, all functions f such that $f(1) = 99, f(2) = 1$ and $f(3) = 0$ hold.

Now we give a lower bound of the number of output functions such that corresponding input function is injective. For each $p_1 \neq p_2 \in \mathbb{P}$, let C_{p_1,p_2} be the set of all output functions such that the corresponding input function has same value on p_1 and p_2. Let C be the set of all output functions such that the induced input function is not injective. Thus, $C = \bigcup_{p_1 \neq p_2 \in \mathbb{P}} C_{p_1,p_2}$. Now for each $p_1 \neq p_2$ with $p_1 = (q_1, g_1)$ and (q_2, g_2) where $g_i \in G$,

$$\left.\begin{array}{rcll} C_{p_1,p_2} & = & \{op\ ;\ \ op(q_1) - op(q_2) = g_2 - g_1\} & \text{if } q_1 \neq q_2 \\ & = & \emptyset & \text{if } q_1 = q_2 \end{array}\right\} \quad (3)$$

Here, we define $op() = 0$. So we obtain that $|C_{p_1,p_2}| \leq N^{|\mathbb{P}_1|-1}$ and hence $|\neg C| \geq N^{|\mathbb{P}_1|}(1 - \frac{|\mathbb{P}|(|\mathbb{P}|-1)}{2N})$ (note that the total number of output functions is $N^{|\mathbb{P}_1|}$). Let $E = \{f \in \mathrm{Func}(G,G)\ ;\ \ f_m^+[k](\mathbf{x}) = \mathbf{y}\}$ then by Corollary 3.3

$$|E| \geq |\neg C| \times N^{N-|\mathbb{P}_1|-k} \geq N^{N-k}(1 - \frac{|\mathbb{P}|(|\mathbb{P}|-1)}{2N}).$$

Thus,

$$\Pr[u^+[k](x_1, \cdots, x_k) = (y_1, \cdots, y_k)] \geq \frac{(1-\epsilon)}{N^k},$$

where u is a uniform random function and $\epsilon = \frac{mk(mk-1)}{2N}$ since we have $|\mathbb{P}_1| \leq mk$. By Theorem 2.5 we have the following main Theorem of this section.

Theorem 3.4. *For any $\mathbf{x} = (x_1, \cdots, x_k) \in G[k]$ and $\mathbf{y} = (y_1, \cdots, y_k) \in G^k$ we have $\Pr[u^+[k](\mathbf{x}) = \mathbf{y}] \geq \frac{(1-\epsilon)}{N^k}$, where $\epsilon = \frac{mk(mk-1)}{2N}$. We also have, $\mathrm{d}_{\mathrm{stat}}(u_m^+[k]_\mathbf{x}, u^{(m)}[k]_\mathbf{x}) \leq \frac{mk(mk-1)}{2N}$ and hence $d_{u_m^+,u^{(m)}}(k) \leq \frac{mk(mk-1)}{2N}$ where $u^{(m)}$ is the uniform random function on $\mathrm{Func}(G^m,G)$ and $\mathbf{x}$ is any k-query function.*

3.1 CBC based on uniform random injective function.

Here we prove a similar result for uniform random injective function v. The proof is exactly same except in the place of counting the set $\{v : v^+[k](\mathbf{x}) = \mathbf{y}\}$, where y_i's are distinct. So we fix any $\mathbf{y} \in G[k]$. Let for each $p_1 \neq p_2 \in \mathbb{P}$, $C^1_{p_1,p_2}$ be the set of all output functions op such that $op(p_1) = op(p_2)$ and $C^1 = \bigcup_{p_1 \neq p_2} C^1_{p_1,p_2}$. We define $C^* = C \cup C^1$. Thus, $op \notin C^*$ means that both input and output functions are injective. It is easy to check that $|C^1| \leq N^{|\mathbb{P}_1|-1} \times \frac{(|\mathbb{P}_1|)(|\mathbb{P}_1|-1)}{2}$ and hence we have

- $|\neg C^*| \geq N^{|\mathbb{P}_1|} \times (1 - \frac{(}{mk-k)(mk-k-1)} 2N - \frac{mk(mk-1)}{2N}) \geq (1 - \frac{mk(mk-1)}{N})$.

We have a similar result like Corollary 3.3. For each $op \notin C^*$, there are exactly $\frac{N!}{(N-|\mathbb{P}|)!}$ many injective f's which induces op and $f^+[k](\mathbf{x}) = \mathbf{y}$ (see the constructions of all f in Equation 2 in the proof of Lemma 3.1). Thus,

$$|\{f \in \mathrm{Func}^{\mathrm{inj}}(G,G) : f^+(\mathbf{x}) = \mathbf{y}\}| \ \geq \ N^{|\mathbb{P}_1|} \times (1 - \epsilon_1) \times \frac{N!}{(N - |\mathbb{P}_1| - k)!}$$

where $\epsilon_1 = \frac{mk(mk-1)}{N}$. Hence, $\Pr[v^+[k](\mathbf{x}) = \mathbf{y}] \ \geq N^{-k} \times (1 - \epsilon_1)$ for all $\mathbf{y} \in T := G[k] = \{\mathbf{y} \in G^k : y_1, \cdots, y_k$ are distinct$\}$ and $\mathbf{x} \in G^m[k]$. Now we have, $\Pr[u^{(m)}[k](\mathbf{x}) \notin T] \leq \frac{k(k-1)}{2N}$. Thus by Lemma 2.3 we have,

$$\mathrm{d}_{\mathrm{stat}}(v^+_m[k]_\mathbf{x}, u^{(m)}[k]_\mathbf{x}) \leq \frac{k(k-1)}{N} + \frac{2mk(mk-1)}{N}$$

for any k-query functions $\mathbf{x}$ and hence

$$\mathrm{d}_{v^+_m, u^{(m)}}(k) \leq \frac{k(k-1)}{N} + \frac{2mk(mk-1)}{N}.$$

Theorem 3.5. $\mathrm{d}_{\mathrm{stat}}(v^+_m[k]_\mathbf{x}, u^{(m)}[k]_\mathbf{x}) \leq \frac{k(k-1)}{N} + \frac{2mk(mk-1)}{N}$ *for any* k*-query function* $\mathbf{x} = (x_1, \cdots, x_k)$. *and hence* $\mathrm{d}_{v^+_m, u^{(m)}}(k) \leq \frac{k(k-1)}{N} + \frac{2mk(mk-1)}{N}$.

4 DAG (Directed Acyclic Graph) based PRF [7]

In this section, we state a class of PRF based on DAG proposed by Jutla [7]. We modify the class slightly so that it contains many known constructions like PMAC, OMAC, TMAC, XCBC etc. The security analysis would be immediate from that of the general class. We first give some terminologies related to DAG.

Terminologies on DAG :

Let $D = (V, E)$ be a directed acyclic graph with finite vertex set V and edges E. We say that $u \prec v$ if there is a directed path from u to v. Note that it is a partial order on V. Let D have exactly one sink node v_f (the maximum element with respect to $\prec$) and at most two source nodes (the minimum element with respect to $\prec$). If there are two such we call them as v_s and v_{iv}. In the original paper, Jutla considered only one source node. Here we extend it to two so that it can contain one more source node for initial value.

- For each node $v \in V$, define D_v by the subgraph induced by the vertex set $V_v = \{u : u \prec v\}$. We define, $N(v) = \{u \in V : (u, v) \in E\}$, the *neighborhood* of v.

- Any map $c : E \to \mathcal{M}$ is said to be color map on D where $\mathcal{M}$ is a field. A colored DAG is pair (D, c) where c is a color map on D.

- Two colored DAG (D_1, c_1) and (D_2, c_2) are said to be *isomorphic* if there is a graph isomorphism between D_1 and D_2 which preserves the color map. More precisely, a graph isomorphism $\rho : D_1 \to D_2$ satisfies $c_2(\rho(e)) = c_1(e)\ \forall\ e \in E_1$. In this case we write $(D_1, c_1) \cong (D_2, c_2)$.

Definition 4.1. *We say a colored graph $G = (D, c)$ is* non-singular *if for all $u, v \in V$, $G_u := (D_u, c[u]) \cong (D_v, c[v]) := D_v$ implies either $u = v$ or $\{u, v\} = \{v_s, v_{iv}\}$ with $c(v_s, w) \neq c(v_{iv}, w)$ whenever (v_s, w) and $(v_{iv}, w) \in E$. Here the color map $c[u]$ is the restriction of c on D_u.*

Definition 4.2. *We say a sequence of colored graph $\mathcal{S} = \langle G^l = (D^l, c^l) = ((V^l, E^l), c^l)\rangle_{l\geq 1}$ is* PRF-preserving *if each D^l is non-singular and $G^l \ncong G^{l'}_u = (D^{l'}_u, c^{l'}[u])$ for $u \in V^{l'}$ and $l' \neq l$.*

Functional Representation of Message

Given a sequence of colored graph $\langle G^l = (D^l, c^l)\rangle_{l\geq 1}$, let $U^l = V^l \setminus \{v^l_{iv}\}$. We fix a sequence of initial values $\mathrm{iv}_l \in \mathcal{M}$, $l \geq 1$. Let $X : U^l \to \mathcal{M}$ be a function, called as a message function. We define its corresponding message-initial value function $\overline{X}$ on G^l as follows :

$$\left.\begin{array}{rcll} \overline{X}(v) & = & X(v) & \text{if } v \in U^l \\ & = & \mathrm{iv}_l & \text{if } v = v^l_{iv} \end{array}\right\} \tag{4}$$

In the definition of $\overline{X}$ we include the graph G^l as a domain even if it is defined only on the set of vertices. Here, we look message in $\mathcal{M}^l$ as a message function on G^l. For any well order $<$ on U^l we can correspond $\mathcal{M}^l$ with a message function on U^l where $|U^l| = l$. Namely, $X(u_1) \parallel \cdots \parallel X(u_l) \in \mathcal{M}^l$ where $u_1 < \cdots < u_l$ and $U^l = \{u_1, \cdots, u_l\}$. Later we will see that each node of the DAG has the underlying function f. The input for the invocation of f at any node is the sum of previous output (outputs of neighborhood nodes) and the value of message-initial value function $\overline{X}$ at that node.

PRF (pseudo random function) Domain Extension Algorithm :

Let $f : \mathcal{M} \to \mathcal{M}$ be a function, $(\mathcal{M}, +, \cdot)$ be a field with $|\mathcal{M}| = N$. Let $\mathcal{S} = \langle G^l\rangle_{l\geq 1}$ be a PRF-preserving sequence of DAG. Given any $X : U^l \to \mathcal{M}$ we have message-initial value function, $\overline{X} : V^l \to \mathcal{M}$. We define two functions, $a_f, b_f : V^l \to \mathcal{M}$ recursively as follows :

$$a_f(v) = \overline{X}(v) + \sum_{w \in N(v)} c^l((w, v)) \cdot b_f(w) \qquad \text{and} \qquad b_f(v) = f(a_f(v)), v \in V^l. \tag{5}$$

The output of $f^{\mathcal{S}}(X)$ is $b_f(v^l_f)$ where v^l_f is the unique sink node. When v is a source node, $N(v) = \emptyset$ and hence $a_f(v) = X(v)$.

Security Analysis

Two message-input functions on colored DAG, $X_1 : G_1 \to \mathcal{M}$ and $X_2 : G_2 \to \mathcal{M}$ are said to be *identical* if $G_1 \cong G_2$ and $X_1(u) = X_2(v)$ where v is the image of u under a graph isomorphism. If not then we say that they are *non-identical.* We identify all identical message-functions. Given $v \in V$ and a message-initial value function on $G = (D, c)$ we define $X[v]$ by the function X restricted on G_v.

Let $X_1, \cdots, X_k$ be k distinct functions, $X_i : U^{l_i} \to \mathcal{M}$ and let $\overline{X_i}$ be its corresponding message-initial value function. Let $\mathbb{P} := \mathbb{P}(\mathbf{X}) = \{X : X = X_i[v], v \in V^{l_i}\}$ where $\mathbf{X} = (X_1, \cdots, X_k)$. We call this also prefix set for $\mathbf{X}$. This is a generalized notion for prefixes of messages in CBC case (see Section 3). Here we similarly have $|\mathbb{P}| \leq Q$, where Q is the total number of message blocks from $\mathcal{M}$. Now we make similar analysis like CBC.

We fix any $\mathbf{X}$. Given any f, define the *intermediate induced output function* (or simply induced output function) $op_f : \mathbb{P}_1(\mathbf{X}) \to \mathcal{M}$ as $op_f(p) = b_f(v)$ where $p = X_i[v]$ and b_f is given as in Equation 5 while we compute $f^{\mathcal{S}}(\overline{X})$ using the colored graph G^{l_i}. Any function from $\mathbb{P}_1(\mathbf{X})$ to $\mathcal{M}$ is called as output function. Let $p = X_i[v] \in \mathbb{P}$, define $\text{last}(p) = X_i(v)$ and $\text{chop}(p) = \{X_i[u] : u \in N(v)\}$. It is an empty set for source node v. Let $X_i[u] = q \in \text{chop}(p)$, then we denote the edge (u, v) by $e_{q,p}$. Given op, define a corresponding *input function* $ip : \mathbb{P} \to \mathcal{M}$ as

$$ip(p) = \text{last}(p) + \sum_{q \in \text{chop}(p)} c^{l_i}(e_{q,p}) \cdot op(q).$$

Now we state a analogous statement of Lemma 3.2 and Corollary 3.3.

Lemma 4.1. *Let op be an induced output function such that $|ip(\mathbb{P}_1)| = q$ where ip is the corresponding input function and $ip(\mathbb{P}_1)$ is the range of it. Then for any $\mathbf{y} = (y_1, \cdots, y_k) \in G^k$ there are exactly N^{N-q} many f such that $op = op_f$ and $f^{\mathcal{S}}[k](\mathbf{x}) = \mathbf{y}$.*

Corollary 4.2. *If op is an induced output function such that corresponding input function ip is injective then there are $N^{N-|\mathbb{P}_1|}$ many f's such that $op_f = op$ and there are $N^{N-|\mathbb{P}_1|-k}$ many f's such that $op_f = op$ and $f^+[k](\mathbf{x}) = \mathbf{y}$.*

Now we give a lower bound of the number of output functions such that corresponding input function is injective. For each $p_1 \neq p_2 \in \mathbb{P}$, let C_{p_1,p_2} be the set of all output functions such that the induced input function has same value on p_1 and p_2. Let C be the set of all output functions such that the induced input function is not injective. Thus, $C = \bigcup_{p_1 \neq p_2 \in \mathbb{P}} C_{p_1,p_2}$. Let $X_{i_1}[v_1] = p_1 \neq p_2 = X_{i_2}[v_2]$, $\text{chop}(p_1) = \{q_i = X_{i_1}[u_i] : 1 \leq i \leq l\}$ and $\text{chop}(p_2) = \{q'_i = X_{i_2}[w_i] : 1 \leq i \leq l'\}$. Now we have three possible cases as given below :

Case-1: $\text{chop}(p_1) = \text{chop}(p_2) = \{q_1, \cdots, q_l\}$ and $c_1(e_{q_i,p_1}) = c_2(e_{q_i,p_2})$, $\forall\ i$ where c_i is the color function corresponding to p_i. Then the underlying graphs for p_1 and p_2 are identical. Since $p_1 \neq p_2$, $X(v_1) \neq X(v_2)$ and hence $C_{p_1,p_2} = \emptyset$.

Case-2: Let $\text{chop}(p_1) = \text{chop}(p_2) = Q$ but there exists $q \in Q$ such that $c_1(e_{q,p_1}) \neq c_2(e_{q_i,p_2})$. Then $ip(p_1) = ip(p_2)$ implies $X_{i_1}(v_1) + \sum_{q \in \text{chop}(p_1)} e_{q,p} \cdot op(q) = X_{i_2}(v_2) + \sum_{q \in \text{chop}(p_2)} e_{q,p} \cdot op(q)$. Hence, $\sum_{q \in Q} a_q \cdot op(q) = a$ for some constants a_q and a where all a_q's are not zero (since color functions are different on Q). Thus, $|C_{p_1,p_2}| = N^{|\mathbb{P}_1|-1}$.

Case-3: Let $\text{chop}(p_1) \neq \text{chop}(p_2)$. In this case $ip(p_1) = ip(p_2)$ implies $\sum_{q \in Q} a_q \cdot op(q) = a$ where Q is not empty and all a_q's are not zero. Thus, $|C_{p_1,p_2}| = N^{|\mathbb{P}_1|-1}$.

So we obtain that $|C_{p_1,p_2}| \leq N^{|\mathbb{P}_1|-1}$ and hence $|\neg C| \geq N^{|\mathbb{P}_1|}(1 - \frac{|\mathbb{P}|(|\mathbb{P}|-1)}{2N})$. By Corollary 3.3

$$|E| \geq |\neg C| \times N^{N-|\mathbb{P}_1|-k} \geq N^{N-k}(1 - \frac{|\mathbb{P}|(|\mathbb{P}|-1)}{2N}),$$

where $E = \{f \in \text{Func}(G,G) \ ; \ \ f^{\mathcal{S}}[k](\mathbf{X}) = \mathbf{y}\}$. Thus,

$$\Pr[u^{\mathcal{S}}[k](\mathbf{X}) = \mathbf{y}] \geq \frac{(1-\epsilon)}{N^k},$$

where $\epsilon = \frac{Q(Q-1)}{2N}$ and u is a uniform random function. By Theorem 2.5 we have the following main Theorem of this section.

Theorem 4.3. *For any* $\mathbf{X} = (X_1, \cdots, X_k)$ *and* $\mathbf{y} = (y_1, \cdots, y_k) \in G^k$ *where* X_i*'s are distinct message function, we have* $\Pr[u^{\mathcal{S}}[k](\mathbf{X}) = \mathbf{y}] \geq \frac{(1-\epsilon)}{N^k}$*, where* $\epsilon = \frac{Q(Q-1)}{2N}$ *and* Q *is the total number of message blocks in queries. We also have,* $\text{d}_{\text{stat}}(u^{\mathcal{S}}[k]_{\mathbf{X}}, U[k]_{\mathbf{X}}) \leq \frac{mk(mk-1)}{2N}$ *and hence* $d_{u^{\mathcal{S}},U}(k) \leq \frac{mk(mk-1)}{2N}$ *where* U *is the uniform random function from the set of all message functions to* $\mathcal{M}$.

Remark 4.4. *The same security analysis can be made for the PRF based on a uniform random injective function like CBC case. We leave the details to the reader as it is very much similar to the CBC case.*

Remark 4.5. *Let* $\mathcal{M} = \{0,1\}^n := \text{GF}(2^n)$*. To define a pseudo random function on* $\{0,1\}^*$ *one can pad* 10^i *(for minimum* $i \geq 0$*) so that the length is the multiple of* n *and then can apply the PRF algorithm as above. So for any distinct messages, the padded messages are also distinct and hence it would be a pseudo random function on the input set* $\{0,1\}^*$*. There is another way to pad it. We pad* 10^i *to a message* X *if it is not a multiple of* n*, otherwise we would not pad anything (this is the case for OMAC,TMAC, XCBC etc.). In this case we have two sequences of colored graph* G_1^l *and* G_2^l *(for all messages with size multiple of* n *and all messages with size not multiple of* n *respectively). Here, we require the combined sequence* $\langle G_1^l, G_2^l \rangle_{l \geq 1}$ *is* PRF-preserving *(thus, even if after padding the messages are equal the corresponding message functions are not* identical*). The similar analysis also can be made in this scenario.*

4.1 Some Known PRFs for Variable length Input.

There are three popularly known constructions which deals with variable size input and uses CBC mode. These are XCBC [5], TMAC [8], OMAC [9] and PMAC [6]. Let K_1 and K_2 be two secret constants from $\{0,1\}^n$. Given $M = M_1 \parallel \cdots M_{l-1} \parallel M_l$ with $|M_1| = \cdots = |M_{l-1}| = n$, $|M_l| = n_1$, $1 \leq n_1 \leq n$ and a random function f on $\{0,1\}^n$, define f^* as follows :

$$\left.\begin{array}{rcll} f^*(M) & = & f_l^+(M_1 \parallel \cdots \parallel M_{l-1} \parallel (M_l \oplus K_1)) & \text{if } n_1 = n \\ & = & f_l^+(M_1 \parallel \cdots \parallel M_{l-1} \parallel (M_l 10^i \oplus K_2)) & \text{if } n_1 < n, i = n - n_1 - 1 \end{array}\right\} \quad (6)$$

XCBC, **TMAC** and **OMAC** are defined on the basis of choices of K_1 and K_2.

- If K_1 and K_2 are chosen independently from f then it is known as XCBC.
- If $K_2 = c \cdot K_1$ and K_1 is chosen independently from f then it is known as TMAC where $c \in \{0,1\}^n$ is some fixed known constant not equal to 1 and 0, and $\cdot$ is a field multiplication on $\{0,1\}^n = \mathrm{GF}(2^n)$.
- If $K_1 = c \cdot L$ and $K_2 = c^2 \cdot L$ where $L = f(0)$ then it is known as OMAC.

Security of OMAC

Here we only consider security for OMAC. For the other constructions, one can make a similar treatment as in CBC. For OMAC as in the previous Remark 4.5 we have two sequences of colored DAGs G_1^l and G_2^l. Each graph is a sequential graph with one more edge at the end. More precisely, $V^l = \{v_s = 1, \cdots, l = v_f, v_{iv}\}$ and $E^l = \{(1,2), \cdots, (l-1,l), (v_{iv}, l)\}$. The color function for G_1^l is as follows : $c^l((i,i+1)) = 1$ and $c^l((v_{iv}, l)) = c$, where $c \neq 0, 1$. Similarly, the color function for G_2^l is as follows : $c^l((i,i+1)) = 1$ and $c^l((v_{iv}, l)) = c^2$. We choose $\mathrm{iv}_l = \mathbf{0} \in \{0,1\}^n$. It is easy to check that each colored DAG is non-singular. Any colored DAG can not be isomorphic to a colored subgraph as the sink node has inward degree 2 where as the other nodes have inward degree 1. Thus, the sequence is admissible. The pseudo randomness property follows from the Theorem 4.3.

Security of PMAC

One can similarly observe that PMAC also belong to this class. The underlying graph $D^l = (V^l, E^l)$, where $V^l = \{v_{iv}, 1, \cdots, l-1, v_f\}$ and $E^l = \{(v_{iv}, i), (i, v_f), 1 \leq i \leq l-1\} \cup \{(v_{iv}, v_f)\}$. There is two color functions depending on the message size. When message size is multiple of n, $c_1(v_{iv}, i) = c^i$ and $c_1(v_{iv}, v_f) = 0$, otherwise it takes constant 1. The other color function is same except that $c_2(v_{iv}, v_f) = a$. Here, c and a are some constants not equal to 0 and 1, and $\mathrm{iv}_l = \mathbf{0} \in \{0,1\}^n$.

5 Improved Security bound of CBC [1].

In this section we will give a simple partial proof of improved security analysis given by Bellare *et. al.* [1]. We will follow same notation as in Section 3. We say an output function *op* is *induced* if there exists an u such that $op_u = op$. We define an event $D^*[k]$ where the corresponding input function of induced output function $ip : \mathbb{P} \to G$ satisfies the following property :

$$\forall\ p_1 \in \{x_1, \cdots, x_k\}, p_2 \in \mathbb{P} \text{ and } p_1 \neq p_2, ip(p_1) \neq ip(p_2). \tag{7}$$

In [1] for $k = 2$, it has been proved that $\Pr[\neg D^*[2]] \leq (8m/N + 64m^4/N^2)$. For $k \geq 2$ it is easy to check that $\Pr[\neg D^*[k]] \leq k(k-1)/2 \times (8m/N + 64m^4/N^2)$. Here we will assume this result as we have not found any simple proof of this. Secondly, one can translate this into a purely combinatorial problem which was solved rigorously by Bellare *et. al.* (Lemma 2 of [1]). Now $\Pr[u^+_{x_1,\cdots,x_k} = (y_1, \cdots, y_k) \mid D^*[k]] = 1/N^k$. This is true that for any *induced* output function *op* with above property there exists N^{N-q_1} many u's which induces *op* and there are N^{N-q_1-k} many u's which induces *op* and $u^+(x_i) = y_i\ \forall\ i$, where q_1 denotes the size of range of induced input function of *op* (see Corollary 3.3). Here, we do not need that the corresponding input function is injective. We can still have a similar statement like in Corollary 3.3 as the input function is taking completely different values on $\{x_1, \cdots, x_k\}$ from the values on $\mathbb{P}_1$ (see Equation 7). Thus, $\Pr[u^+_{x_1,\cdots,x_k} = (y_1, \cdots, y_k)] \geq \big(1 - \frac{k(k-1) \times (8m/N + 64m^4/N^2)}{2}\big) \times \frac{1}{N^k}$. Thus we have,

Theorem 5.1. $\mathrm{Adv}_{u_m^+, u^{(m)}}(k) \leq k(k-1)/2 \times (8m/N + 64m^4/N^2)$.

6 A simple proof for On-line Cipher Hash-CBC [2].

In this section we define what is meant by on-line cipher and what is the ideal candidate for that. Then we give a simpler security proof of Hash-CBC [2] and note that in the original proof there is a flaw which could not not be easily taken care unless we make further assumptions.

An online cipher, Hash-CBC construction is given by Bellare *et. al.* [2]. In Crypto 2001 [2], the notion of On-Line cipher has been introduced and a secure Hash-CBC construction has been proposed. First we define what is meant by On-Line cipher and the definition of Hash-CBC construction.

1. Let G be a group and $G^{[1,m]} = \cup_{1 \leq i \leq n} G^i$ and $|G| = N$. A function $f : G^{[1,m]} \to G^{[1,m]}$ is called a *length preserving injective function* if f restricted to G^i is an injective map from G^i to G^i.

2. Let f be a length-preserving injective function and $M = M_1 \parallel \cdots \parallel M_m$, then we write $f(M) = (f^{(1)}(M), \cdots, f^{(m)}(M))$, where $f^{(i)}(M) \in G$. f is said to be *on-line* if there exists a function $X : G^{[1,m]} \to G$ such that for every $M = M[1] \parallel \cdots \parallel M[m]$, $f^{(i)}(M) = X(M[1] \parallel \cdots \parallel M[i])$. It says that first i blocks of cipher only depends on the first i blocks of message. Note that for each $i \geq 1$, and $(M[1] \parallel \cdots \parallel M[i-1]) \in G^{i-1}$, $X(M[1] \parallel \cdots \parallel M[i-1] \parallel x)$ is an injective function from G to G as a function of x since f is length-preserving injective function. We also say that X is an on-line function.

3. X^U is said to be uniform random on-line function if X is chosen uniformly from the set of all on-line functions from $G^{[1,m]}$ to G.

Hash-CBC :

Let H be a random function from G to G which satisfies the following property. $\Pr[H(x_1) - H(x_2) = y] \leq \epsilon$ for all $x_1 \neq x_2 \in G$ and $y \in G$. We say this random function by ϵ-almost universal random function. Thus for any (x_i, y_i), $1 \leq i \leq k$, with distinct x_i's we have,

$$\Pr[H(x_i) + y_i = H(x_i) + y_j \text{ for some } i \neq j] \leq \frac{k(k-1)\epsilon}{2}. \qquad (8)$$

Given an ϵ-almost universal random function and a uniform random injective function v on G we define a random on-line function F, known as **HCBC** (or **Hash-CBC**), as follows:

$$X(M[1] \cdots M[j]) = C[j], \text{ where } C[i] = v(H(C[i-1]) + M[i]),\ 1 \leq i \leq j \text{ and } C[0] = 0.$$

Note that X is a random on-line function. Let $x_1, \cdots, x_k \in G^{[1,m]}$ and $\mathbb{P}$ be the set of all non-empty prefixes of these messages. Let $y_p \in G$, where y_p's are distinct and not equal to 0 and $|\mathbb{P}| = q$. Now we want to compute $\Pr[X(p) = y_p, \forall p \in \mathbb{P}]$ where the probability is based on uniform random injective function v and ϵ-almost universal random function H. Let D be the event that for all p,

$(H(y_{\mathrm{chop}(p)}) + \mathrm{last}(p))$'s are distinct where $y_\lambda := 0$ and λ is the empty string. Since y_p's are distinct and not equal to 0, $\Pr[D] \geq 1 - \frac{q(q-1)\epsilon}{2}$. Condition on D all inputs of v are distinct. Thus,

$$\Pr[X(p) = y_p, \forall p \in \mathbb{P} \mid D] = \frac{1}{N(N-1)\cdots(N-q+1)} \text{ and hence}$$

$$\Pr[X(p) = y_p, \forall p \in \mathbb{P}] \geq \frac{(1 - \frac{q(q-1)\epsilon}{2})}{N(N-1)\cdots(N-q+1)} \text{ by Equation 8}$$

$$\geq (1 - \frac{q(q-1)\epsilon}{2}) \times \Pr[X^U(p) = y_p, \forall p \in \mathbb{P}]$$

since $\Pr[X^U(p) = y_p, \forall p \in \mathbb{P}] \leq 1/N(N-1)\cdots(N-q+1)$. Given any query functions, let $X^U[q]$ and $X[q]$ denote the joint distribution of X^U and X on $\mathbb{P}$ respectively. Let

$$T = \{(y_p)_{p \in \mathbb{P}} : y_{\mathrm{chop}(p)} \neq 0 \ \forall p, \text{ and } y_p{}'\text{s are distinct } \}$$

It is easy to check that $\Pr[X^U \notin T] \leq \frac{q(q-1)}{2N}$. Now by Lemma 2.3 we obtain the following main Theorem of this section.

Theorem 6.1. *For any query function, the statistical distance* $\mathrm{d_{stat}}(X^U[q], X[q]) \leq q(q-1)\epsilon + \frac{q(q-1)}{N}$ *and hence* $\mathrm{Adv}_{X^U,X}(q) \leq q(q-1)\epsilon + \frac{q(q-1)}{N}$.

Remark 6.2. *In [2], authors also consider chosen-cipher text security for a variant of the above construction. In this scenario, there are two different types of queries. Let $\mathbb{P}$ denotes the set of all prefixes of the queries of on-line function X and $\mathbb{P}^*$ denotes the set of all prefixes of queries of corresponding inverse on-line function Y (say). Now one can similarly prove that*

$$\Pr[X(p) = y_p, \forall p \in \mathbb{P} \text{ and } Y(p) = w_p \forall p \in \mathbb{P}^*]$$

$$\leq (1 - \epsilon) \times \Pr[X^U(p) = y_p, \forall p \in \mathbb{P} \text{ and } Y^U(p) = w_p \forall p \in \mathbb{P}^*],$$

where X^U and Y^U denote the uniform random on-line function and it's corresponding inverse function respectively. So we have same security analysis. We leave reader to verify all the details of the chosen cipher text security.

6.1 A flaw in the proof of the original paper [2]

In the original paper due to Bellare *et. al.* [2], the security proof has some flaw. The Claim 6.5 of [2] says that if some bad event does not occur then the the distribution of the view is identical for both classes of functions. More precisely, $X(p)$'s and $X^U(p)$'s are identically distributed condition on some bad event does not occur (i.e., the inputs of uniform random injective function v are distinct). In case of X^U, all conditional random variables $X^U(p)$'s are uniformly and identically distributed on the set T. But, conditional distribution of $X(p)$'s is not so as the condition is itself involved with $X(p)$ and an unknown distribution due to H. For example, when $p_1 = x_1$ and $p_2 = x_1 \parallel x_2$ then $X(p_1) = v(H(0) \oplus x_1)$ and $X(p_2) = v(H(X(p_1)) \oplus x_2)$. The conditional event E (the complement of bad event) is $H(0) \oplus x_1 \neq H(X(p_1)) \oplus x_2$ and $v(H(0) \oplus x_1) \neq 0$. According to their claim for any $0 \neq y_1 \neq y_2$, $p = \Pr[v(H(0) \oplus x_1) = y_1, v(H(X(p_1)) \oplus x_2) = y_2 \mid E] = \frac{1}{(N-1)(N-1)}$ (note that y_2 can be zero). Let $a := x_1 \oplus x_2$, $C = H(0) \oplus x_1$ and $\epsilon_{y,z,c} = \Pr[H(y) \oplus H(z) = c]$.

Now, $p_1 := \Pr[v(H(0) \oplus x_1) = y_1 \ \wedge \ v(H(y_1) \oplus x_2) = y_2 \ \wedge \ E]$

$$= \sum_{h_1,h_2 \,:\, h_1 \oplus h_2 \neq a} \Pr[v(h_1 \oplus x_1) = y_1, v(h_2 \oplus x_2) = y_2, H(0) = h_1, H(y_1) = h_2] = \frac{\epsilon_{0,y_1,a}}{N(N-1)}$$

and $p_2 := \Pr[E] = \Pr[v(C) \neq 0, H(v(C) \oplus x_2) \neq C]$

$$= \sum_{z,h \,:\, z \neq 0} \Pr[v(h \oplus x_1) = z, H(0) = h, H(z) \neq h \oplus a] = \frac{1}{N} \times \sum_{z \,:\, z \neq 0} \Pr[H(0) \oplus H(z) \neq a].$$

Thus, $p = \frac{\epsilon_{0,y_1,a}}{(N-1) \times \sum_{z \neq 0} \epsilon_{0,z,a}} \neq \frac{1}{(N-1)(N-1)}$ in general. This can occur if $\epsilon_{0,z,a} = \epsilon_{0,y_1,a}$ for all $z \neq 0$, but there is no such assumption for H in [2]. A similar flaw can be observed in the Claim 8.6 of [2] where the chosen cipher text security is considered.

7 Conclusion and Future Work.

In this paper we make a unifying approach to prove the unpredictability of many existing constructions. This paper attempts to clean up several results regarding unpredictability so that the researchers can feel and understand the subject in a better and simpler way. As a concluding remark we would like to say that one can view the security analysis in the way we have observed in this paper and can have better and simpler proof for it. Some cases people have wrong proofs due to length and complicity of it. Thus, a more concrete as well as simple proof is always welcome.

In future, this unifying idea may help us to make good constructions. It seems that one may find constructions where the security bound is more than the birth day attack bound. Till now, there is no known construction based on ideal function (having output n-bit) which has security close to 2^n. One may obtain a better bound for CBC as we have used only those output functions which induces an injective input functions. One can try to estimate the other output functions also.

References

[1] 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.

[2] M. Bellare, A. Boldyreva, L. Knudsen and C. Namprempre. On-Line Ciphers and the Hash-CBC constructions. Advances in Cryptology - CRYPTO 2000. Lecture Notes in Computer Science, Volume **2139**, pp 292-309.

[3] M. Bellare, J. Killan and P. Rogaway. The security of the 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. ID 24120a1f8b92722b5e1 5fbb6a86521a0.

[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] 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.

[8] 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.

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