Equivalence between MAC and PRF for Blockcipher based Constructions

Nilanjan Datta and Mridul Nandi

Cryptology Research Group
Applied Statistics Unit
Indian Statistical Institute
203, B.T. Road, Kolkata, India 700108
nilanjan_isi_jrf@yahoo.com, mridul.nandi@gmail.com

Abstract. In FSE 2010, Nandi proved a sufficient condition of pseudo random function (PRF) for affine domain extensions (ADE), wide class of block cipher based domain extensions. This sufficient condition is satisfied by all known blockcipher based ADE constructions, however, it is not a characterization of PRF. In this paper we completely characterize the ADE and show that *message authentication code (MAC) and weakly collision resistant (WCR) are indeed equivalent to PRF*. Note that a PRF is trivially a MAC and WCR, however, the converse need not be true in general. So our result suggests that it would be sufficient to ensure resisting against weakly collision attack or the forging attack to construct a pseudo random function ADE. Unlike FSE 2010 paper, here we consider the *forced collisions of inputs of underlying blockciphers by incorporating the final outputs of a domain extension queried by an adaptive adversary*. This is the main reason why we are able to obtain a characterization of PRF. Our approach is a more general and hence might have other theoretical interest.

Keywords: Affine Domain Extension, Blockcipher, MAC, PRF, WCR.

1 Introduction

Message Authentication Code. In Symmetric key setting where two parties, the sender and the receiver share a common key, say K, Message Authentication Code (MAC) is used to ensure the "integrity" of the message and the "authenticity" of the sender, during a message exchange protocol. When the sender wants to send a message M to the receiver, he or she also sends a tag $T = \mathcal{G}_K(M)$. The pair (M, T) is called a *valid pair*. The receiver verifies whether the obtained pair is valid or not. As far as security is concerned, a MAC needs to ensure that even if an adversary $\mathcal{F}$ possess some tagged messages (may be of adversary's own choice), it must not be able produce a valid tag corresponding to a new message, called *fresh* valid pair. More formally, we define the *forging advantage* or *mac advantage* of a forgery adversary $\mathcal{F}$ against a Message Authentication scheme $\mathcal{G}_K$ as follows.

$$\mathbf{Adv}^{\mathrm{mac}}_{\mathcal{G}_K}(\mathcal{F}) \stackrel{\Delta}{=} \mathbf{Pr}_{\mathrm{rand}(\mathcal{F}),K}[(M,T) \leftarrow \mathcal{F}^{\mathcal{G}_K},\ (M,T) \text{ is a fresh valid pair}]. \quad (1)$$

The algorithm $\mathcal{G}$ is called (t, Q, ϵ)-mac if for any forgery adversary $\mathcal{F}$ making at most Q queries with (time) complexity at most t has mac-advantage at most ϵ.

Weak Collision Resistant. Weak Collision Resistant (WCR) is the secret key version of collision security property of a keyed hash function $\mathcal{G}_K(\cdot)$. This notion is mainly adopted in [1] to prove other security notions such as MAC or PRF. The keyed function $\mathcal{G}_K$ is called (t, Q, ϵ)-wcr if for all collision adversaries $\mathcal{C}$ with complexity at most t making at most Q queries has *wcr-advantage*, as defined below, at most ϵ:

$$\mathbf{Adv}^{wcr}_{\mathcal{G}}(\mathcal{C}) \stackrel{\Delta}{=} \mathbf{Pr}_{\mathrm{rand}(\mathcal{C}),K}[\mathcal{C}^{\mathcal{G}_K} = (M, M'),\ \mathcal{G}_K(M) = \mathcal{G}_K(M'),\ M \neq M']. \quad (2)$$

Pseudo Random Function. Pseudo Random Function (PRF) [8] is a keyed function $\mathcal{G}_K$, whose behavior is indistinguishable from a random function $\mathcal{R}$ for any computational adversary. A random function (or permutation) is a function chosen uniformly at random from the set of all functions (permutation, respectively). The security of a cryptographic construction based on a random function, preserves it's security even when we replace the random function by a PRF. The formal definitions of a PRF and prf-advantage are given below :

Definition 1 (Pseudo Random Function). *A keyed function* $\mathcal{G} : \{0,1\}^k \times \mathcal{M} \to \{0,1\}^n$ *is called* (t, Q, ϵ)-secure pseudo random function *if for every distinguisher* $\mathcal{D}$ *with (time) complexity at most* t*, making at most* Q *queries, and key* K *chosen uniformly from* $\{0,1\}^k$*, the* **prf-advantage**[1] *of the distinguisher*

$$\mathbf{Adv}^{prf}_{\mathcal{G}}(\mathcal{D}) \stackrel{\Delta}{=} \mathbf{Pr}_{\mathcal{R}}[\mathcal{D}^{\mathcal{R}} = 1] - \mathbf{Pr}_{K \in_R \{0,1\}^k}[\mathcal{D}^{\mathcal{G}_K} = 1] \leq \epsilon.$$

We consider only those distinguishers which run in polynomial time. Without loss of generality, we simplify distinguisher which actually simplifies the analysis: We assume that the **distinguisher is deterministic** making at most, say Q **distinct queries** only. It is not difficult to see that for any arbitrary distinguisher there is a distinguisher satisfying above having advantage no less than the given one.

1.1 (Affine) Domain extension for PRF

Domain extension is a method by which functions of small domains are used to construct an extended function over an arbitrary domain for similar security notions, e.g. designing a hash function from a compression function. MACs are domain extensions extending small domain PRPs or PRFs to arbitrary domain PRPs [12] or PRFs [8] respectively. A domain extension based on a keyed blockcipher E_K (a keyed family of permutation usually modeled to be PRP) invokes

[1] Even though, in the original definition, absolute value is considered, it does not matter as we are interested in maximum advantage of all possible distinguisher and hence we change the sign of advantage by considering distinguisher $\overline{\mathcal{D}}$ (flipping the output bits of $\mathcal{D}$).

E_K several times sequentially. For a blockcipher E_K-based affine domain extension (or ADE), the inputs (called *intermediate inputs*) to E_K are determined by some *affine functions* of the previous outputs (called *intermediate outputs*). The output of the last invocation of the blockcipher is defined to be the final output of the ADE.

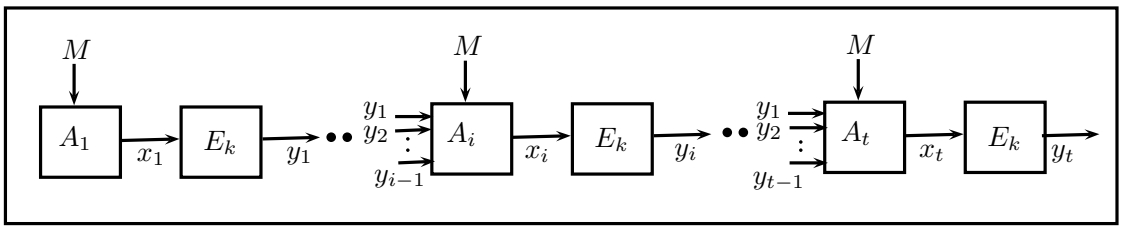

Fig. 1.1. Affine Domain Extension: Here A_i's are the affine functions, i.e. row vectors. The coefficient of the affine function is determined by the message M. The coefficient matrix A (see definition 2) is the combination of all these row vectors A_i's.

Definition 2. *A domain extension $\mathcal{G}$ (see Figure 1.1) is called* Affine Domain Extension *(ADE) over $\mathcal{M}$ if a lower triangular matrix $A_{l\times(l+1)}$, called* **coefficient matrix***, entries from the finite field $\mathbb{F}_{2^n}$, is associated with each message $M \in \mathcal{M}$ to compute $\mathcal{G}_K(M) \stackrel{\Delta}{=} y_l$ where y_i's are defined recursively as (1) $(x_1, \ldots, x_l)^{\mathrm{tr}} = A.(1\ y_1\ \ldots\ y_l)^{\mathrm{tr}}$ and (2) $E_K(x_i) = y_i$, $1 \le i \le l$.*

Throughout the paper we identify the underlying set of $\mathbb{F}_{2^n}$ as $\{0,1\}^n$. The integer $l := l(M)$ is the length of the message M. As E_K is a fixed permutation for a fixed key, the above definition can be similarly defined for a permutation π to define $\mathcal{G}^\pi(M)$. A class of popular constructions like CBC-MAC [6], GCBC* [17], OMAC [9], PMAC [7] etc. are some of such examples. The original PRF bounds for the above were about $\frac{\sigma^2}{2^n}$ or $\frac{l^2\cdot Q^2}{2^n}$ [4, 5, 10, 11, 18, 20] where ℓ and σ are the longest and total number of blocks present in at most Q queries, respectively. Bellare, Pietrzak and Rogaway in [3], showed first time an improved bound $\frac{lQ^2}{2^n}$ for CBC-MAC. Afterwards, similar improved bounds were given for PMAC [13, 14], OMAC [16] and EMAC [20, 21]. Nandi [15] showed an unified bound of PRF advantages of an ADE satisfying a sufficient condition mentioned below. It eventually gives an unified proof of all these existing analysis and bounds using well known as *Decorrelation* [23, 24] or Patarin's *coefficient H-technique* [19].

A sufficient condition for PRF of ADE. Informally, the sufficient condition is that the output of $\mathcal{G}^\pi(M)$ should not be in the force collision relation with any other specific intermediate output of π, while computing $\mathcal{G}^\pi(M')$ for some message M and M'. The *forced collision relation* is an equivalence relation for

which whenever i is related to j, the intermediate output of i^{th} and j^{th} invocation to the underlying blockcipher matches for all choices of π. Thus, the collisions are only due to some specific choices of the messages. For example, for CBC, messages with same prefixes have collisions in the computation of the blocks in the common prefix. Note that final outputs are not incorporated to define forced collision relation, only messages are used as if we are constructing the collision patterns for a non-adaptive adversary.

1.2 Known Implication among MAC, PRF and WCR

It is easily seen that any (t, Q, ϵ)-prf $\mathcal{G}$ is $(t', Q-1, \epsilon - \frac{1}{2^n})$-mac for some $t' \approx t$. Whenever a forgery adversary F forges a pair (M, T), a distinguisher can make the query M and if the response is T, it decides that it is interacting with $\mathcal{G}$, otherwise random function. The converse is not true for a secure MAC: $\mathcal{G}_K(M)$ $= f_K(M)||0$ where f_K is a PRF. Since it's last bit is always zero which can be easily used to distinguish from random function. If a keyed function is injective such as identity function, without using key, then clearly it is WCR as there is no collision present but one can easily forge. So WCR does not necessarily imply MAC.

1.3 Our Contribution

We know that a PRF implies a message authentication code and weakly collision resistant. However, the converse is not true in general. In this paper, we show that message authentication code (MAC) and weakly collision resistant (WCR) are indeed equivalent to PRF for ADEs. Thus we have a complete characterization of ADE. The previously known sufficient condition is not necessary as given an example below:

Example 1. Define the padding rule P on messages as:

$$P(M) = \begin{cases} 1||m_1 & \text{if } M = m_1 \\ 1||m_1||m_2 & \text{if } M = m_1||m_2 \\ 0||l||M & \text{else} \end{cases}$$

where l denotes the no. of blocks in message M. Clearly according to the definition of the padding above, for $M = m_1$ and $M' = m_1||m_2$ the sufficient condition is not satisfied. Hence the result can not be applied. But since the padding ensures any two message combination except M and M' are prefix-free condition, and for these two messages the output of M, say w_1 does not give a restriction unless $w_1 = m_2$ (which has low probability) hence it would not be difficult to show that the construction is a PRF. Note that, this construction doesn't have any practical importance, it is used just to theoretically show that the sufficient condition is not necessary always.

In this paper we prove the following theorem.

Theorem [Main theorem of the paper]. Let $\mathcal{G}$ be a ADE based on a random permutation π. Then for any distinguisher $\mathcal{D}$ there is a forgery and collision adversaries $\mathcal{F}$ and $\mathcal{C}$ respectively such that

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathcal{G}}(\mathcal{D}) \leq \frac{4\sigma^2}{2^n} + \frac{\mu}{2}$$

where $\mu = \min\{\mathbf{Adv}^{\mathrm{wcr}}_{\mathcal{G}}(\mathcal{C}),\ \mathbf{Adv}^{\mathrm{mac}}_{\mathcal{G}}(\mathcal{F})\}$.

In section 4 we demonstrate the reduction of $\mathcal{F}$ and $\mathcal{C}$ and provide the analysis. Difficulty in proving a MAC to be PRF is due to the lack of entropy in MAC which is must for a random function. As we consider ADE based on random permutation we have a potential source of randomness from the underlying random permutation. But it is not obvious why there is no other way to distinguish ADE from random function unless we forge or obtain a collision in final outputs.

2 Affine Domain Extensions

Suppose we have q messages, $M_i \in \mathcal{M}$ of lengths l_i, $1 \leq i \leq q$ and their corresponding co-efficient matrix is given by $A_i = (m_i\ C_i)$. Then the joint co-efficient matrix A of the q messages is given by the following partition matrix

$$\begin{pmatrix} m_1 & C_1 & 0 & \cdots & 0 \\ m_2 & 0 & C_2 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ m_q & 0 & 0 & \cdots & C_q \end{pmatrix}_{t\times(t+1)} \qquad \text{where } t = t_q \text{ and } t_i = \textstyle\sum_{j=1}^{i} l_j.$$

To each permutation π we associate an intermediate input and output vectors are $x^\pi := x = (x_1, \ldots, x_t)$ and $y^\pi := y = (y_1, \ldots, y_t)$ respectively, where (I) $A \cdot \overline{y} = A.\binom{1}{y^{\mathrm{tr}}} = x$ and (II) $\pi(x_i) = y_i$, $i \in [1..t] := \{1, \ldots, t\}$ where $\overline{y} = (1, y_1, \ldots, y_t)^{\mathrm{tr}}$. The second condition justifies the terms intermediate input and output vectors as these are indeed inputs and outputs of the permutation π while computing $\mathcal{G}^\pi(M_i)$'s. The first condition says how the intermediate input is determined only from the intermediate outputs and it does not depend on the underlying permutation π. Thus, we write the input vector x by $A(y)$ or we write $y \to x$. Clearly, these conditions uniquely determine the input and output vector since A is a lower triangular matrix and hence x_i's and y_i can be defined recursively. We thus have a mapping $Y : \mathbb{P}_n \to (\{0,1\}^n)^t$ defined as $Y(\pi) = y^\pi$. Note that this function need not be surjective or even injective. We characterize all vectors which are in the image of this function. More precisely, we characterize all vectors y such that there is a permutation π such that $y = y^\pi$. We call these *output vectors.*

Lemma 1. *$y \in \{0,1\}^{nt}$ is an* **output vector** *if and only if $x_i = x_j \Leftrightarrow y_i = y_j$ where $y \to x$.*

Proof. "Only if" is obvious as $\pi(x_i) = y_i$ for all i, for some permutation π. To prove the "if" part, choose any permutation π such that $\pi(x_i) = y_i$ for all i. This is possible since equality pattern of both vectors x and y are same. For any such permutation π, $y = y^\pi$. □

Collision Relation. Let us define collision relation $\mathsf{coll}(y) :=\sim$ over $[1..t]$ of a vector y as $i \sim j$ iff $y_i = y_j$. It is an equivalence relation capturing the collisions of the elements of the vector y. We define $\mathsf{coll}_\pi \stackrel{\Delta}{=} \mathsf{coll}(y^\pi)$, the collision pattern of the output vector, which is the equivalence relation $\sim$ over $[1..t]$ such that $i \sim j$ if and only if $y_i \sim y_j$. Thus, the the characterization of an output vector can be restated as follows:

§ y is an output vector if and only if $\mathsf{coll}(y) = \mathsf{coll}(x)$ where $y \to x := A(y)$.

Now, an intermediate output function y can be associated with more than one permutations. We want to count the number of π's an output function y is associated with. Let $\mathbb{P}_n[y] := Y^{-1}(\{y\})$ denote the set of all permutations π with y as an output function, i.e. $y = y^\pi$. Clearly, all these permutations have to agree on the sets of all intermediate inputs as $\pi(x_i) = y_i$, $\forall i$, $1 \le i \le t$ (due to the second condition) as x is uniquely determined by y by the relation $x = A \cdot \overline{y}$. Now fix any permutation π such that $\pi(x_i) = y_i$ for all i. It is easy to see that $y^\pi = y$ and hence

$$\mathbb{P}_n[y] = \{\pi : \pi(x_i) = y_i, 1 \le i \le t\}, \quad |\mathbb{P}_n[y]| = (2^n - s)! \tag{3}$$

where s denotes the number of distinct values of the ouput vector y.

3 Estimation of Probability of a View

We fix a deterministic distinguisher D making only distinct queries, the number of queries is at most Q and the total length of all queries is at most σ. We identify the tuples of distinct elements $w = (w_1, \ldots, w_t)$ as set $\{w_1, \ldots, w_t\}$. From the context it must be clear. Given a subset $T = \{t_1, \ldots, t_q\} \subseteq [1..t] := \{1, 2, \ldots, t\}$ we define $w[T]$ by the sub-tuple $(w_{t_1}, \ldots, w_{t_q})$. For a matrix A, $A[i, \cdot]$ and $A[\cdot, j]$ denote the ith row and jth column respectively. Similarly we define the sub-matrices $A[1..i, \cdot]$ or $A[\cdot, 1..j]$ etc.

3.1 View of an Oracle Algorithm

Let $\mathcal{V}$ be the set of all tuples $w = (w_1, \ldots, w_q)$, $1 \le q \le Q$, such that D stops making queries on seeing w_q. Note that this is defined independent of the oracle. The view of $D^{\mathcal{O}}$, denoted $\mathsf{view}(D^{\mathcal{O}})$, by the tuple $(w_1, \ldots, w_q) \in \mathcal{V}$ where w_i denotes the response of the ith query, $1 \le i \le q$. The responses $w_1, \ldots, w_{i-1}$ uniquely determines ith query M_i if it queries or that D stops (as D is deterministic). The final response of D must be some function of its view. If $\mathcal{O}$ is a probabilistic oracle then the view as well as the number of queries q are random variables determined by the randomness of the oracle only. So given any

fixed view $w = (w_1, \ldots, w_q)$ the probability $\mathbf{Pr}_{\mathcal{O}}[\mathsf{view}(D^{\mathcal{O}}) = w]$ is computed over the randomness of $\mathcal{O}$. If the probability is positive then we say the view w is *realizable* or *$\mathcal{O}$-realizable.* and the set of all realizable views is denoted by $\mathcal{V}_{\mathcal{O}}$. Note that $\mathcal{V}_{\mathcal{O}} \subseteq \mathcal{V}_{\mathcal{R}} = \mathcal{V}$ where $\mathcal{R}$ is a random function. We denote the *truncated view* $\mathsf{view}(D^{\mathcal{O}})[i]$ by the i-tuple $(w_1, \ldots, w_i)$ where $\mathsf{view}(D^{\mathcal{O}}) = (w_1, \ldots, w_q)$, $i \leq q$. We can similarly define when a truncated view is $\mathcal{O}$-realizable. Note that for $w = (w_1, \ldots, w_i)$,

$$\mathbf{Pr}[\mathsf{view}(D^{\mathcal{O}})[i] = w] = \sum_{\substack{v \in \mathcal{V}: \\ v[1..i]=w}} \mathsf{Pr}[\mathsf{view}(D^{\mathcal{O}}) = v].$$

Note that, for $v \in \mathcal{V}$, we have, $\mathbf{Pr}_{\mathcal{R}}[\mathsf{view}(D^{\mathcal{R}})[i] = v[1..i]] = 2^{-ni}$. For an arbitrary probabilistic oracle the probability computation of views is not easy. In this section we provide an estimate of probability of realizing some views where the oracle is an affine domain extension $\mathcal{G}$ based on a random permutation Π on $\{0,1\}^n$.

Lemma 2. *Let $w = (w_1, \ldots, w_q) = v[1..q]$ for some $v \in \mathcal{V}$. Then either w is not realizable (i.e. the probability of realizing w is zero) or*

$$\mathbf{Pr}_{\Pi}[\mathit{view}(D^{\mathcal{G}^{\Pi}})[q] = w] = \sum_{s \geq 1} \frac{N_{w,s}}{P(2^n, s)} \tag{4}$$

where $P(2^n, s) = 2^n(2^n - 1)\ldots(2^n - s + 1)$ and $N_{w,s}$ denotes the number of output vectors y with s many distinct elements and $y_{t_i} = w_i$, $1 \leq i \leq q$.

Proof. Let $w = (w_1, \ldots, w_q)$ and $M_1, \ldots, M_q$ be the corresponding queries by $\mathcal{D}$. As $\mathcal{G}$ is an ADE there is a lower triangular matrix (joint coefficient matrix) A with a tuple of final indices $T = (t_1, \ldots, t_q)$ and for each permutation π we associate an intermediate output vector $y := y^{\pi}$ such that $A \cdot \overline{y} = x$ and $\pi(x_i) = y_i$, $1 \leq i \leq t := t_q$. Now,

$$\begin{aligned}
\mathbf{Pr}_{\Pi}[\mathsf{view}(D^{\mathcal{G}^{\Pi}}) = w] &= \frac{1}{2^n!} \times |\{\pi : \mathcal{G}^{\pi}(M_i) = w_i, i = 1, \ldots, q\}| \\
&= \frac{1}{2^n!} \times |\{\pi : y^{\pi}_{t_i} = w_i, i = 1, \ldots, q\}| \\
&= \frac{1}{2^n!} \times \sum_{s \geq 1} \sum_{y} |\{\pi : y^{\pi} = y, i = 1, \ldots, q\}| \times \frac{1}{2^n!}
\end{aligned}$$

where the second sum is taken over all *output vectors y* such that the number of distinct elements of y is s and $y_{t_i} = w_i$, $1 \leq i \leq q$. Using the counting of the number of permutations given in equation 4,the result follows. □

To use the above lemma we need to provide an estimate of $N_{v,s}$ which can be done by identifying a special equivalence relation $\sim^*$, called *forced relation*, such that there are sufficient number of output vectors y inducing the forced collision relation, i.e., $\mathsf{coll}(y) = \sim^*$. Since for all these output vectors the s value

is same with the number of equivalence classes of $\sim^*$, we will immediately have a lower bound of the probability of the view. More precisely, if we can show the following:

§ **existence of forced relation:** there is a relation with $s+q$ many classes such that the number of output vectors y with $y_{t_i} = w_i$ for all i is at least $2^{ns}(1-\epsilon)$,

then

$$\mathbf{Pr}_{\Pi}[\mathsf{view}(D^{\mathcal{G}^{\Pi}})[q] = w] = \sum_{s\geq 1} \frac{N_{w,s}}{P(2^n,s)} \geq \frac{2^{ns}(1-\epsilon)}{P(2^n,s+q)} \geq \frac{1-\epsilon}{2^{nq}}.$$

3.2 Forced Relation

Let $\mathcal{V}_{\text{dist}} = \{(w_1,\ldots,w_q) \in \mathcal{V} : w_i\text{'s are distinct}\}$, $\mathcal{V}_{\text{coll}} = \mathcal{V} \setminus \mathcal{V}_{\text{dist}}$. We study the following problem motivated from the probability computation of realizing a view $w = (w_1,\ldots,w_q) \in \mathcal{V}_{\text{dist}}$ as discussed above. Let $A = (m\ C)$ be a coefficient matrix with a strictly lower triangular matrix $C_{t\times t}$ and a vector $m_{t\times 1}$ whose elements are from $\mathbb{F}_{2^n}$. Let $\sim$ be an equivalence relation over $[t]$.

Problem 1. Reduce the affine function $A : y \mapsto A(y) := C\cdot y + m$, given that
(i) $\mathsf{coll}(y) = \sim$ and
(ii) $y[T] = w$ where $T = (t_1,\ldots,t_q)$, t_i's are distinct element from $[t]$.

There may be different ways to reduce a system of affine equations. We reduce the affine function by incorporating the given constraints as much as possible. The equivalence relation is considered not to have any collision on T, i.e. for all $i \neq j \in T$, $i \nsim j$, as we fix distinct final outputs w_i's. Let the leader set (consists of one element from each equivalence class) of $\sim$ be $L \sqcup T$. We choose elements of $L := \{i_1,\ldots,i_s\}$ to be the minimum elements of the equivalence classes.

$$\begin{aligned} C\cdot y + m &= m + (C[\cdot,1]\cdot y_1 + \ldots + C[\cdot,t]y_t) \\ &= (m + \sum_{t_i\in L_f} w_i \sum_{j\sim t_i} C[\cdot,j]) + \sum_{i\in L}(\sum_{j\sim i} C[\cdot,j])y_i \\ &= A^{\mathsf{rd}}[.,0] + \sum_{i\in L} A^{\mathsf{rd}}[.,i]y_i \end{aligned}$$

where $\mathsf{rd} = (\sim, T, w)$ to denote that we reduce the matrix A using the triple rd. We can complete the matrix $A^{\mathsf{rd}}_{t\times(t+1)}$ by defining $A^{rd}[.,i] = \mathbf{0}$ for all $i \notin \{0\}\cup L$. Thus, we have

$$A(y) = x,\ \mathsf{coll}(y) = \sim,\ y[T] = w$$

$$\Leftrightarrow\ A^{\mathsf{rd}}[.,0] + \sum_{i_j\in L} A^{\mathsf{rd}}[.,i_j]z_j = x,\ z_j\text{'s are distinct and different from } w_i's$$

where $z_j = y_{i_j}$, $1 \leq j \leq s$. In fact, given a solution z, we construct an unique solution y as $y[L] = z$, $y[T] = w$ and the other y_i's are defined through the

relation $\sim$, i.e. $y_i = w_j$ if $i \sim t_j$ or $y_i = z_j$ if $i \sim i_j$. This reduction helps to solve y for the following equations:

$$\mathsf{coll}(y) \;=\; \mathsf{coll}(m + C \cdot y) \;= \sim,\; y[T] = w. \tag{5}$$

If we denote $y[L] = z$ then the above equation is equivalently written as (i) $\mathsf{coll}(A^{\mathsf{rd}}(z)) = \sim$, (ii) z_i's are distinct and different from w_j's. Note that $\sim$ is fixed for which no collision on T. To have a solution we have the following immediate necessary condition:

$$A^{rd}[i,.] = A^{rd}[j,.] \;\Rightarrow\; i \sim j.$$

In fact, there are other differnet necessary conditions. However, we consider a special equivalence relation which would satisfy all necessary conditions and also gives several solutions of z and hence y.

Definition 3. *We say that an equivalence relation $\sim$ over $[t]$ is* **forced relation** *w.r.t A, T and w if*

$$A^{\mathsf{rd}}[i,.] = A^{\mathsf{rd}}[j,.] \;\Leftrightarrow\; i \sim j, \quad \textit{where } \mathsf{rd} = (\sim, T, w). \tag{6}$$

Note that there may not exist forced relation with no collision in T. Clearly, if $\sim$ is a forced relation with no collision in T then the Eq. **??** is equivalently rewritten as $(A^{rd}[i,.] - A^{rd}[j,.])\overline{z} \neq 0$ for all $i \nsim j$ and (ii) z_i's are distinct and different from w_j's. The number of such z, equivalently y, is at least

$$2^{ns} \times (1 - \frac{\binom{s}{2} + \binom{t}{2} + st}{2^n}).$$

This can be easily seen as total possible choices without any constraint is 2^{ns} and number of z which does not satisfy a given constraint is $2^{n(s-1)}$. The number of constraint is at most $\binom{s}{2} + \binom{t}{2} + st$ which includes the distinct choices of z, the number of pairs (i,j) for which $i \nsim j$ and different from w_i's. Now we prove the existence of forced collision which may or may not have collisions in T. In fact, we prove a more general statement which says the existence of extending a given relation to a forced relation.

Lemma 3 (Extension Lemma). *Given any relation $\sim$ satisfying the property $i \sim j \Rightarrow A^{\mathsf{rd}}[i,\cdot] = A^{\mathsf{rd}}[j,\cdot]$ where $\mathsf{rd} = (\sim, T, w)$ then there is a forced relation $\sim'$, denoted $\mathsf{Ext}_A(\sim)$, containing $\sim$.*

Moreover, Ext can be defined in a way such that whenever $\sim$ is a forced collision w.r.t. $A[1..t',\cdot]$, T and w for some $t' \leq t_q$ then $\sim' = \sim$ on $[1..t']$.

Proof. We provide an existence proof. Given the relation $\sim$ and the property $i \sim j \Rightarrow A^{\mathsf{rd}}[i,\cdot] = A^{\mathsf{rd}}[j,\cdot]$, we need to construct an algorithm to obtain $\sim'$ such that $A^{\mathsf{rd}'}[i,\cdot] = A^{\mathsf{rd}'}[j,\cdot] \Rightarrow i \sim' j$ where $\mathsf{rd} = (\sim', T, w)$. Our algorithm $\mathsf{Ext}_A(\sim)$ works as follows :

- Step 1. Find a (i,j) pair such that such that $i \nsim j$ but $A^{\mathsf{rd}}[i,\cdot] = A^{\mathsf{rd}}[j,\cdot]$. If no such pair exist, then return $\sim$ and call it $\sim'$. Else do the following :

- Step 2. Add (i, j) pair in $\sim$ and define $\sim$ to be the minimum equivalence relation containing the previous $\sim$ and (i, j). Reduce the A^{rd} matrix with respect to the modified $\sim$. Go to Step 1.

Look that at each step we are adding a new pair to the collision relation which satisfies the initial given condition. As at most $\binom{t}{2}$ pair can be present in a collision relation over $[1..t]$, the algorithm terminates with at most $\binom{t}{2}$ steps executed. When the algorithm terminates, we have $i \sim' j$ iff $A^{\mathsf{rd}'}[i, \cdot] = A^{\mathsf{rd}'}[j, \cdot]$. Hence, $\sim'$ is a forced collision relation.

For the 2nd part of the lemma, look that $\sim$ is a forced collision w.r.t. $A[1..t', \cdot]$, T and w for some $t' \leq t_q$. Hence if the algorithm find a (i, j)-pair, one of i and j must have index $> t'$. This property and the lower triangular property of A ensures that, even the next reduction may change the values of a column whose index is $< t'$ but it changes uniformly over each row, hence will not affect the collision relation over $[1..t']$. Hence the result follows. □

Corollary 1. *If we choose $\sim$ to be an empty relation then from the above lemma: there is always a forced collision relation.*

The existence of the forced relation is guranteed but it may have collision in T. For a given $w \in \mathcal{V}_{\text{dist}}$ we can arise into two possible cases.

Case-1 : There is a forced relation $\sim^*$ with no collision in T. In this case we have high interpolation probability as we have seen already. We call such a view w random and we use Decorrelation technique to prove that distinguishing ADE from a random function for these views is difficult.

Case-2 : The forced relation has collision. If we detect the collision in right time then we would be able to forge ADE. We call those views forge. We can show that there is a set of small size, called forbidden set, such that if the output is not from the forbidden set the collision would be detected in right time.

Remark 1. The reason we may not able to detect collision in right time that when we update the forced relation $\sim_i$ on ith query we find a collision in previous final inputs i.e. $t_j \sim_i t_l$ where $j, l < i$.

4 Reducing Distinguishing to Forgery

A distinguisher D whose job is to distinguish between a random function chosen uniformly and an ADE $\mathcal{G}^{\Pi}$ based on a random permutation Π. We define a forgery $\mathcal{F}$ which has access of $\mathcal{G}^{\Pi}$ and aims to forge, i.e. to generate a fresh valid pair. The way $\mathcal{F}$ runs as follows:

§ **Initial step**: It runs a distinguisher D. So $\mathcal{F}$ has to reply the responses of the queries, say M, of D to get the next queries.

§ **On query** M **from** D: It updates the "*forced internal collision patterns*" (sure collisions of intermediate inputs of the random permutation) given the

view obtained so far. It has been computed before observing the final output $\mathcal{G}^{\Pi}(M)$.

Case 1 (forge event): If it finds that the final output of the current query collides with the previous query, say M' having the response w', then $\mathcal{F}$ forges (M, w'). It is a valid pair which is guaranteed by the forced collision pattern.

Case 2 (bad event): Otherwise it forwards the query to $\mathcal{G}^{\Pi}$ and obtains response w. If w is not in a bad set, called "*forbidden set*", it forwards the response to D, otherwise abort. The reason of considering forbidden set is to have consistence update of forced collision pattern.

§ **Finalization**: If it neither aborts nor forges then it aborts and we would be able to prove that, in this case, D can not distinguish $\mathcal{G}^{\Pi}$ from random function. The more details of the above description is given below.

4.1 Formal Description of Distinguish-Forge Game

Game $D \leftrightarrow \mathcal{F} \leftrightarrow \mathcal{G}^{\Pi}$ **:**

1. $\mathcal{F}$ runs D and hence to obtain next query it has to reply a query of D.
2. On ith query M_i, it computes $\mathsf{Ext}_{A_i}(\sim_{i-1})$ for $w = (w_1, \ldots, w_{i-1})$ and $T = (t_1, \ldots, t_{i-1})$.
3. If $t_i \sim_i t_j$ for some $j < i$ then forge event sets true and forge by the pair (M_i, w_j) and `stop`.
4. Otherwise, it obtains a response w_i. Define F_i, the forbidden set, to be the set of all values $f \notin F_z, z < i$ such that $\exists a, b < t_i$ with, $B[a,k] \neq B[b,k]$ and $B[a,z] = B[b,z]$ $\forall z \neq k$ and $f = \frac{B[a,0]-B[b,0]}{B[a,k]-B[b,k]}$, where B is the reduced co-efficient matrix upto M_i.
5. If $w_i \in F_i$ then `abort`.
6. Otherwise it forwards the response w_i to $\mathcal{D}$.
7. When D sends his guess bits to $\mathcal{F}$, it `stop`.

Lemma 4. *If $\sim_i$ is force collision relation with respect to A, $w = (w_1, \ldots, w_{i-1})$ and $T = (t_1, \ldots, t_{i-1})$. Then if $w_i \notin F_i$, then force collision relation doesn't change.*

Proof. If $\sim_i$ is force collision relation with respect to A, $w = (w_1, \ldots, w_{i-1})$ and $T = (t_1, \ldots, t_{i-1})$. Then if $w_i \notin F_i$, then following Reduction module 1 and 2, it is clear that even if some changes occur in columns $\leq t_{i-1}$, it will be uniform over the rows and hence the force collision relation won't get changed. □

We make another reasonable assumption that whenever forge event occurs (which can be computed by $\mathcal{D}$ also) it checks the response w_i is same as w_j or not. If not then it returns 1, otherwise 0. It is not difficult to see that with this transformation from $\mathcal{D}'$ to $\mathcal{D}$ the prf-advantage is not differ by more than $\frac{1}{2^n}$. More precisely, $\mathbf{Adv}^{\mathrm{prf}}(\mathcal{D}') \leq \mathbf{Adv}^{\mathrm{prf}}(\mathcal{D}) - \frac{1}{2^n}$. Now, we categorize the possible views of $\mathcal{D}$ into the following four classes - (i) collision view $\mathcal{V}_{\mathrm{coll}}$ (collisions in w_i values), (ii) random view (denoted by $\mathcal{V}_{rand}$), (iii) forbidden view (denoted by $\mathcal{V}_{forb}$) and (iv) forge view (denoted by $\mathcal{V}_{forge}$). The definitions of these views are given below :

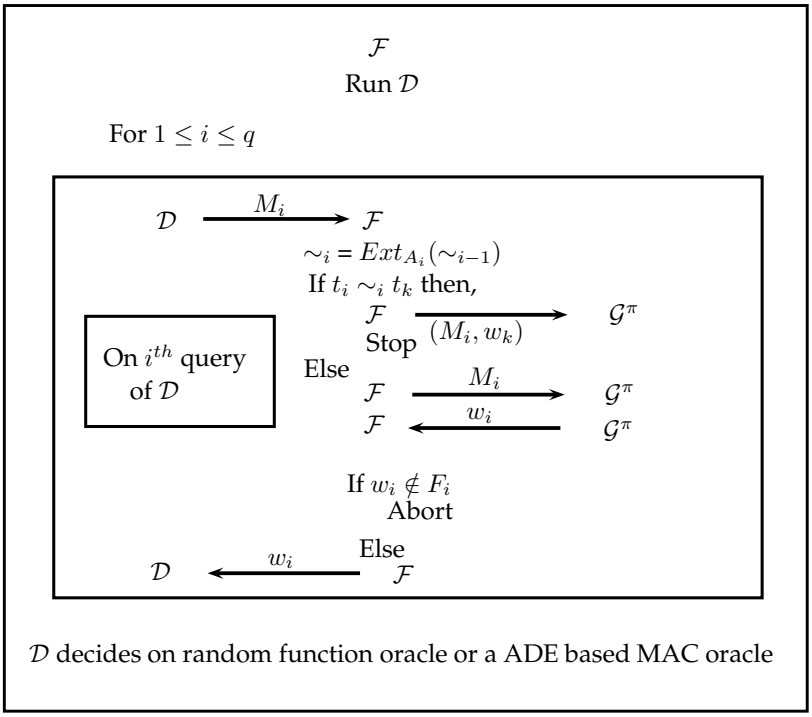

Fig. 4.1. Pictorial representation of the definition of $\mathcal{F}$. Here A_i denotes the joint coefficient matrix of $M_1 \cdots M_i$

.

Input: A, T, W, $\sim$

Extension Algorithm $Ext_A(\sim)$

let T be the set of final output indexs, L is the set of smallest indexes corresponding to an equivalence class which are not $\sim$-related to any element of T.
If $k \in T$ (Case : 1)
    Add $A^{\sim}[*, j].w_k$ to $A^{\sim}[*, 0]$
    Make $A^{\sim}[*, j] = 0$
    Add the pair (t_k, j) to $\sim$
If $k \in L$ (Case : 2)
    Add $A^{\sim}[*, j]$ to $A^{\sim}[*, k]$
    make $A^{\sim}[*, j] = 0$
    Add the pair (k, j) to $\sim$

Algorithm 1: Extension Algorithm

- $\mathcal{V}_{rand} = \{(w_1, w_2, \cdots, w_q) : \forall i,\, j{\neq}i,\, w_i \notin F_i \text{ and } t_i \nsim^* t_j \}$
- $\mathcal{V}_{forb} = \{(w_1, w_2, \cdots, w_i) : w_i \in F_i \text{ and } \forall j \leq i,\, k < j,\, t_k \nsim^* t_j \}$
- $\mathcal{V}_{forge} = \{(w_1, w_2, \cdots, w_i) : \forall k < i,\, w_k \notin F_k \text{ and } \exists j < i,\, t_i \sim^* t_j \}$

It is easy to see that that $\mathcal{F}$ forges whenever the view of $D^{\mathcal{G}^{\Pi}}$ is a forge view. (We skip the proof)

Lemma 5. $\mathbf{Pr}[\mathit{view}(D^{\mathcal{G}^{\Pi}})$ *sets* forge *true*$] = Pr[\mathcal{F}$ *forges*$]$.

Lemma 6. $Pr[\mathit{view}(\mathcal{D}^{\mathcal{R}}) \in \mathcal{V}_{\text{forb}}] \leq \varepsilon_1$ *where* $\varepsilon_1 = \frac{\binom{t}{2}}{2^n}$

Proof. Look that if (a, b) is pair used to give a forbidden value f for F_i. then the way we have extended our collision relation, it ensures that (a, b) no longer can be used to give another forbidden value later as the a^{th} and b^{th} row will be identical after i^{th} message. Hence each pair can be at most in 1 forbidden set F_i. As maximum $\binom{t}{2}$ pairs can be chosen hence $|F_i| \leq \binom{t}{2}\ \forall i \leq q$.
Hence, $Pr[View(D^R) \in V_{forb}] = \sum_{i=1}^{q} Pr[w_i \in F_i] \leq \frac{\binom{t}{2}}{2^n}$ □

The definition of $\mathcal{C}$ is exactly same as $\mathcal{F}$ except that when $\mathcal{F}$ forges by the pair (M_i, w_j) it returns the collision pair (M_i, M_j).

Theorem 1 (Main theorem of the paper). *Let $\mathcal{G}$ be a ADE based on a random permutation Π. Then for any distinguisher $\mathcal{D}$ there is a forgery and collision adversaries $\mathcal{F}$ and $\mathcal{C}$ respectively such that*

$$\mathbf{Adv}^{\text{prf}}_{\mathcal{G}}(\mathcal{D}) \leq \frac{4\sigma^2}{2^n} + 2 \cdot \mu$$

where $\mu = \min\{\mathbf{Adv}^{\text{wcr}}_{\mathcal{G}}(\mathcal{C}),\ \mathbf{Adv}^{\text{mac}}_{\mathcal{G}}(\mathcal{F})\}$.

Proof. Note that $t \leq \sigma$ the maximum number of blocks in all queries. Recall that we have four types of disjoint views $\mathcal{V}_{\text{coll}}$, $\mathcal{V}_{\text{forb}}, \mathcal{V}_{forge}$ and $\mathcal{V}_{\text{rand}}$. Since for all random views $v \in \mathcal{V}_{\text{rand}}$, we have

$$\mathsf{Pr}[\mathsf{view}(\mathcal{D}^{\mathcal{G}}) = v] \geq (1 - \epsilon) \times \mathsf{Pr}[\mathsf{view}(\mathcal{D}^{\mathcal{R}}) = v]$$

where $\epsilon \leq 2\sigma^2/2^n$ (as shown before). By using coefficien H-technique we have $\mathbf{Adv}^{\text{prf}}_{\mathcal{G}}(\mathcal{D}) \leq \epsilon + \mathsf{Pr}[\mathsf{view}(\mathcal{D}^{\mathcal{G}}) \in \mathcal{V} \setminus \mathcal{V}_{\text{rand}}]$. Now from counting of $\mathcal{V}_{\text{coll}}$ and lemma 6 we know that $\mathsf{Pr}[\mathsf{view}(\mathcal{D}^{\mathcal{R}}) \in \mathcal{V}_{\text{forb}} \cup \mathcal{V}_{\text{coll}}] \leq \frac{\binom{q}{2}+\binom{\sigma}{2}}{2^n}$. Now we need to bound $\mathsf{Pr}[\mathsf{view}(\mathcal{D}^{\mathcal{R}}) \in \mathcal{V}_{\text{forge}}]$. Since the oracle of the distinguisher is random function, not the ADE, we use the following relationship for all forge views $v = (w_1, \ldots, w_i)$ (note that the first $(i-1)$-tuple determines the forge event and w_i can be chosen freely) :

$$\mathsf{Pr}[\mathsf{view}(\mathcal{D}^{\mathcal{G}})[i-1] = v[1..i-1]] \geq (1 - \epsilon) \times \mathsf{Pr}[\mathsf{view}(\mathcal{D}^{\mathcal{R}})[i] = v[1..i-1]].$$

Since the view $(w_1, \ldots, w_{i-1})$ is actually a random view (as both forge and forbidden did not occur before) we have the above inequality. So combining this, we have

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathcal{G}}(\mathcal{D}) \leq \frac{4\sigma^2}{2^n} + \frac{2^n}{1-2\sigma^2} \times \mathsf{Pr}_{\Pi}[\mathsf{view}(\mathcal{D}^{\mathcal{G}}) \in \mathcal{V}_{\mathrm{forge}}] \leq \frac{4\sigma^2}{2^n} + 2 \cdot \mathbf{Adv}^{\mathrm{mac}}_{\mathcal{G}}(\mathcal{F})$$

since we may assume that $2\sigma^2/2^n \leq 1/2$ hence otherwise the bound is obviuosly true. This proves our main theorem. Similarly we have the result for weak collision resistant. □

5 Conclusion and Future Works

In this paper we showed that message authentication code (MAC) and weakly collision resistant (WCR) are indeed equivalent to PRF. We know that a PRF implies a MAC and WCR, but the converse is not true in general. Our result shows that, the sufficient condition for an ADE to be Pseudorandom function, is to resist the weakly collision attack or message forgery attack. Unlike FSE 2010 paper where the author considered collision pattern of inputs of the underlying blockcipher for a *non-adaptive adversary*, here we considered the "dynamic" collision pattern of inputs for an *adaptive adversary*. Moreover we incorporate collisions among final outputs with other non-final outputs while bounding the PRF advantages of ADE. We introduce the notion of force collision and checked after each message query, whether the current final output is forced related with a previous outputs, in that case, we *forge* the ADE, as it knows the output. The way we have characterizes ADE, makes our approach more general and it might have other theoretical interest. We havn't provided any practical application of the result in this paper as it is beyond our scope and it is itself a strong theoretical result to be self-motivated. However, it would be nice to construct an efficient ADE based MAC (not as example 1 given in section 1) that doesn't satisfy the sufficient condition for an ADE to be a PRF according to FSE 2010 paper but proved out to be a PRF because of it's resistance of MAC forging attack or Weak collision attack.

References

[1] Mihir Bellare, *New Proofs for NMAC and HMAC: Security without Collision-Resistance* **4117** (2006), *Advances in Cryptology - Crypto 2006*, Lecture Notes in Computer Science, 2006. Citations in this document: §1.

[2] Mihir Bellare, Roch Guerin, Phillip Rogaway, *XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions* (1995), 15–28, *CRYPTO 1995*, 963, 1995.

[3] M. Bellare, K. Pietrzak, P. Rogaway, *Improved Security Analysis for CBC MACs* **3621** (2005), 527–545, *Advances in Cryptology - CRYPTO 2005*, Lecture Notes in Computer Science, 2005. Citations in this document: §1.1.

[4] M. Bellare, J. Killan, P. Rogaway, *The security of the cipher block chanining Message Authentication Code* (1994), 341–358, *Advances in Cryptology - CRYPTO 1994*, Lecture Notes in Computer Science, 839, 1994.
[5] Daniel J. Bernstein, *A short proof of the unpredictability of cipher block chaining* (2005). URL: http://cr.yp.to/papers.html#easycbc.
[6] J. Black, P. Rogaway, *CBC MACs for arbitrary length messages* (2000), 197–215, *Advances in Cryptology - CRYPTO 2000*, Lecture Notes in Computer Science, 1880, 2000. Citations in this document: §1.1.
[7] J. Black, P. Rogaway, *A Block-Cipher Mode of Operations for Parallelizable Message Authentication* (2002), 384–397, *Advances in Cryptology - Eurocrypt 2002*, Lecture Notes in Computer Science, 2332, 2002. Citations in this document: §1.1.
[8] Oded Goldreich, Shafi Goldwasser, Silvio Micali, *How to construct random functions, JACM 1986* (1986), 792–807. Citations in this document: §1, §1.1.
[9] T. Iwata, K. Kurosawa, *One-Key CBC MAC* (2003), 129–153, *Fast Software Encryption, 10th International Workshop, FSE 2003*, Lecture Notes in Computer Science, 2887, 2003. Citations in this document: §1.1.
[10] T. Iwata, K. Kurosawa, *Stronger Security Bounds for OMAC, TMAC, and XCBC* (2003), 402–415, *Progress in Cryptology - INDOCRYPT 2003*, Lecture Notes in Computer Science, 2904, 2003.
[11] C. S. Jutla, *PRF Domain Extension using DAG.* (2006), 561–580, *Theory of Cryptography: Third Theory of Cryptography Conference, TCC*, Lecture Notes in Computer Science, 3876, 2006.
[12] Michael Luby, Charles Rackoff, *How to construct pseudorandom permutations from pseudorandom functions*, SIAM Journal of Computing (1988), 373–386. Citations in this document: §1.1.
[13] K. Minematsu, T. Matsushima, *Improved Security Bounds for PMAC, TMAC, and XCBC* (2007), 434–451, *Fast Software Encryption 2007*, Lecture Notes in Computer sciences, 4593, 2007.
[14] A. Mandal, M. Nandi, *Improved Security Analysis of PMAC*, Journal of Mathematical Cryptology, July 2008 (2008), 149–162.
[15] Mridul Nandi, *A Unified Method for Improving PRF Bounds for a Class of Blockcipher based MACs* (2010), 212–219, *FSE 2010*, Lecture Notes in Computer Science, 6147, 2010. Citations in this document: §1.1.
[16] Mridul Nandi, *Improved security analysis for OMAC as a pseudorandom function*, Journal of Mathematical Cryptology (2009), 133–148. Citations in this document: §1.1.
[17] Mridul Nandi, *Fast and Secure CBC-Type MAC Algorithms* (2009), 375–393, *FSE 2009*, Lecture Notes in Computer Science, 5665, 2009. Citations in this document: §1.1.
[18] M. Nandi, *A Simple and Unified Method of Proving Indistinguishability* (2006), 317–334, *Progress in Cryptology - INDOCRYPT 2006*, Lecture Notes in Computer Science, 4329, 2006.
[19] J. Patarin, *Etude des Générateurs de Permutations Basés sur le Schéma du D.E.S.,* Phd Thèsis de Doctorat de l'Université de Paris 6 (1991). Citations in this document: §1.1.
[20] E. Petrank, C. Rackoff, *CBC MAC for real-time data sources*, Journal of Cryptology **13** (2000), 315-338.
[21] Krzysztof Pietrzak, *A Tight Bound for EMAC* (2006), 168–179, *ICALP (2)*, 2006.
[22] Palash Sarkar, *Pseudo-Random Functions and Parallelizable Modes of Operations of a Block Cipher* (2009). URL: http://eprint.iacr.org/2009/217.

[23] S. Vaudenay, *Decorrelation over infinite domains: the encrypted CBC-MAC case* (2001), 75–85.
[24] Serge Vaudenay, *Decorrelation: A Theory for Block Cipher Security*, Journal of Cryptology (2003), 249–286.