On Stateless Schemes for Message Authentication Using Pseudorandom Functions

Palash Sarkar

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

November 11th, 2008

Abstract. We consider the construction and analysis of pseudorandom functions (PRF) for message authentication. Earlier work due to Bernstein and Vaudenay show how to reduce the analysis of PRFs to some probability calculations. We revisit this result and use it to prove some general results on constructions which use a PRF with "small" domain to build a PRF with "large" domain.
These results are then used to analyse several existing and new constructions. Important among them is a simplified proof of a bound on the PRF-property of the cipher block chaining (CBC) mode of operation of a block cipher for message authentication code (MAC). Several existing variants of CBC-MAC are analysed using our framework and new schemes are described. One of the new schemes improve upon the NIST standard CMAC scheme by reducing the number of block cipher invocations by one for messages which are longer than n bits.
Next, we consider parallelizable constructions. An improved version of the well known PMAC scheme is described; the improvement consists of removing the requirement of a discrete log computation in the design stage of PMAC. An earlier parallel construction called the protected counter sum (PCS) had been proposed by Bernstein. PCS uses a keyed compressing function rather than a block cipher. We describe a variant of PMAC which works with keyed compressing function and compared to PCS requires lesser number of invocations.
All our constructions are in the stateless setting, i.e., a setting where the sender and the receiver do not share any state (apart from the common secret key). One of the aspects of our work is the simple and direct approach to the analysis of PRFs. In particular, we avoid the extensive and heavy machinery of game-playing technique which is used in most papers on this topic.

Keywords: pseudorandom function, message authentication, CBC-MAC, CMAC, protected counter sum, PMAC.

1 Introduction

Authentication is one of the two basic tasks of cryptography with encryption being the other. In the symmetric key setting, the sender and the receiver share a common secret key K. Given a message x, the sender uses K to generate a tag, called a message authentication code (MAC), and sends (x, tag) to the receiver. The receiver uses K to verify that (x, tag) is a properly generated message-tag pair. In most cases, verification is simply to regenerate the tag on x and compare to the received value. The tag authenticates the message, or, in other words, it provides an assurance to the receiver that the message x was indeed sent by the sender. A method for tag generation and verification is called a MAC scheme.

An attack on a MAC scheme amounts to forging a message-tag pair, i.e., to find a valid pair which was not generated by the tag generation algorithm. An attacker (also called an adversary) is

said to be successful if he can indeed generate such a pair. It is usually assumed that the adversary can obtain some tags on messages of his choosing. In other words, the adversary is allowed to ask the sender to authenticate some messages (chosen by the adversary) and provide the corresponding tags to the adversary. This is modelled by considering the tag generation algorithm to be instantiated by a secret key (unknown to the adversary) and provided as an oracle to the adversary. The adversary interacts with this oracle by providing messages and obtains the corresponding tags. At the end of the interaction, the adversary outputs a "new" pair (x, tag), i.e., this (x, tag) does not equal any (x_i, tag_i), where tag_i was returned by the oracle on query x_i. The adversary is successful if (x, tag) passes the verification of the MAC scheme.

The description of MAC scheme given above does not require the sender and receiver to maintain state. Some constructions, on the other hand, are stateful. This means that for each message, apart from the secret key, the sender and receiver must have the same value of a variable called a nonce. This value itself need not be secret. The requirement on a nonce is that of freshness, i.e., the scheme should ensure that for a fixed key the nonce value is not repeated. In this work, we will not consider stateful MAC schemes.

A block cipher is a basic cryptographic primitive. Formally, it is a map $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{K}$ is the set of keys and $\mathcal{M}$ is the set of messages. For every $K \in \mathcal{K}$, $E_K : \mathcal{M} \to \mathcal{M}$ is bijective and hence a permutation of $\mathcal{M}$. In practical applications, $\mathcal{M} = \{0,1\}^n$ for some fixed positive integer n and similarly, $\mathcal{K}$ also consists of fixed length binary strings. Well known examples are DES, AES [9]. It is usually assumed that a block cipher provides both privacy and authentication. However, a block cipher by itself can authenticate only n-bit strings. Applications require the authentication of long and possibly variable length strings. Authentication of a long string requires several invocations of the block cipher. Proper methods of doing this are called modes of operations.

The security model of a block cipher is that of a pseudorandom permutation (PRP) [17]. Informally, this means that an adversary should not be able to distinguish the block cipher from a uniform random permutation of $\{0,1\}^n$. This is formalized in the following manner. The adversary $\mathcal{A}$ is given an oracle, which takes as input an n-bit string and also returns an n-bit string as output. $\mathcal{A}$ makes several queries to the oracle and finally outputs a bit b. Suppose a key K is chosen uniformly at random from $\mathcal{K}$ and the oracle is instantiated by $E_K()$ and let p_1 be the probability that $\mathcal{A}$ outputs 1 in this case. Similarly, let p_0 be the probability that $\mathcal{A}$ outputs 1 when the oracle is instantiated using a uniform random permutation. Then the advantage of $\mathcal{A}$ in attacking the PRP-property of the block cipher is given by $|p_1 - p_0|$. This advantage is parametrized by the number of queries that $\mathcal{A}$ makes and the runtime of $\mathcal{A}$. A stronger notion is that of strong pseudorandom permutation (SPRP), where $\mathcal{A}$ is also provided the inverse oracle. In this work we will not require SPRP.

A related notion is that of a pseudorandom function (PRF). Let F be a random (but, not necessarily uniform random) function from a set S to a set T, where T is a finite non-empty set. Let $\mathcal{A}$ be an adversary which has an oracle. The oracle takes as input an element of S and returns as output an element of T. As before, instantiate the oracle in two ways; either with F or with a uniform random function from S to T and let the corresponding probabilities of $\mathcal{A}$ outputting 1 be p_1 and p_0. Then the advantage of $\mathcal{A}$ in attacking the PRF-property of F is defined to be $|p_1 - p_0|$. As in the case of MAC, the PRF-advantage is also parametrized by the number of queries made and the runtime of the adversary.

PRFs have many applications in cryptography. In this work, we will be concerned with the use of a PRF as a MAC scheme. It is intuitively clear (and also not difficult to prove) that a PRF F

whose domain consists of arbitrary length binary strings and whose range is a short fixed length binary string, can serve as a MAC scheme. Basically, given a message x, the tag is $\mathsf{tag} = F(x)$; and verification is done by regenerating the tag. The PRF-property of F is a sufficient condition for it being used as a MAC scheme. There are very efficient known constructions of PRF including one which has been standardized by NIST [10]. In this work, we will be concerned with constructions of PRFs which are useful for MAC applications.

Many practical modes of operations for MAC schemes using block ciphers are analysed as a PRF. This analysis is done in two steps.

1. First analyse the scheme by replacing the block cipher with a uniform random permutation. This provides a bound on the PRF-advantage of an adversary. The bound on the advantage is information theoretic, i.e., it does not depend on the runtime of the adversary. In other words, the adversary is considered computationally unbounded and is only limited by the number of queries it can make. This forms the difficult part of the entire analysis.
2. Now, consider a block cipher instead of the uniform random permutation. Then, it is easy to show that the advantage obtained in Step 1 degrades by an additive term which is the advantage of the block cipher as a PRP.

Suppose that instead of a block cipher, a keyed compressing function is used to construct the PRF [4]. A similar analysis can be used; analysis is done using a uniform random function instead of the keyed function. In the second step, the advantage is adjusted by an additive term to reflect the strength of the keyed function as a PRF.

In view of this, in our analyses, we will only consider the first step. In other words, we will be analysing modes of operations which uses a uniform random permutation instead of a block cipher. Similarly, constructions using a keyed compressing function will be analysed with a uniform random function instead of the keyed function.

1.1 Our Contributions

A useful result for upper bounding PRF-advantage was proved by Bernstein [4] and Vaudenay [26]. Let $F : S \rightarrow T$ be a random function and U is a "large" subset of T^d for some positive integer d. For distinct $x_1, \ldots, x_d \in S$ and $(y_1, \ldots, y_d) \in U$, $\Pr[F(x_1) = y_1, \ldots, F(x_d) = y_d]$ is called a d-interpolation probability [4]. In [26], it has been proved that if F has large "interpolation probabilities" on a "large" subset of T^d, then the advantage of F as a PRF can be upper bounded for any adversary which makes at most d queries. The special case where the subset U equals T^d has been proved in [4]. We slightly modify this result so as to include a length function λ on S. In applications, for $x \in S$, $\lambda(x)$ would be the number of n-bit blocks into which x is formatted. This makes it easier to apply the result to concrete settings.

Suppose $F = \pi \circ F_1$, where π is a uniform random permutation; F_1 invokes π a finite number of times and the entire randomness of F_1 arises from the invocations of π. Such random functions F are typical of many well known constructions of MAC schemes. This class of functions covers the class of DAG based construction considered in [12, 20].

We consider this in the more general setting where $F = \rho \circ F_1$, with ρ being either a uniform random permutation or a uniform random function. Suppose x and x' are two inputs to F; $\mathcal{U}_1, \ldots, \mathcal{U}_m$ and $\mathcal{U}'_1, \ldots, \mathcal{U}'_{m'}$ are the inputs to the invocations of ρ during the computations of $\mathcal{Z} = F_1(x)$ and $\mathcal{Z}' = F_1(x')$ respectively. We define three events: collision (Coll), i.e., $\mathcal{Z} = \mathcal{Z}'$; self-disjoint (Self-Disjoint), i.e., $\wedge_{i=1}^{m}(\mathcal{Z} \neq \mathcal{U}_i)$; and pairwise-disjoint (Pairwise-Disjoint), i.e., $(\wedge_{i=1}^{m}(\mathcal{Z}' \neq$

$\mathcal{U}_i) \wedge (\wedge_{i=1}^{m'}(\mathcal{Z} \neq \mathcal{U}'_i))$. We show that if the probabilities of Coll, $\overline{\text{Self-Disjoint}}$ and $\overline{\text{Pairwise-Disjoint}}$ are all small, the the PRF-advantage of F is also small. Several variants of this result are also proved. These results are useful, since they reduce the task of bounding PRF-advantage of F to that of bounding the probabilites of certain events for F_1. Previous analysis of individual constructions have used this approach, but, to the best of our knowledge, the result has not been proved in this generality earlier.

Analysis of Constructions. Using the general results developed in this paper, we analyse several sequential and parallel constructions. These include both old and new constructions. In case of existing constructions, our analysis is simpler and provide neater and somewhat tighter bounds. The new constructions offer certain practical advantages over existing constructions. Details are provided later.

Of particular interest is our analysis of the MAC scheme using the cipher block chaining (CBC) mode of operation (CBC-MAC for short) as a PRF. CBC-MAC is a simple and well studied method which uses a block cipher modelled as a uniform random permutation of $\{0,1\}^n$. The previous best known bound on the PRF-advantage was obtained using an extensive game-based technique followed by a rather dense and difficult combinatorial analysis [2]. In contrast, our analysis is a much simpler probability calculation to bound collision probability and the probability of Self-Disjoint and Pairwise-Disjoint. This calculation directly reflects the structure of the CBC-MAC scheme. Several variants of CBC-MAC are also analysed including the following.

1. VCBC-MAC: a variant of CBC-MAC using a compressing function $\rho : \{0,1\}^\ell \to \{0,1\}^n$ instead of a block cipher.
2. CBC-MAC has two restrictions. First, it can only handle messages whose lengths are a multiple of n; second, the set of messages for which CBC-MAC can be proved to be secure has to be prefix-free, i.e., no message must be a prefix of any other message. We describe a simple and efficient technique to overcome both these problems. VCBC-MAC shares the same problems with CBC-MAC and our technique also works for VCBC-MAC. The technique is efficient in the sense that it increases the invocations of π (or ρ) by only a small number.
3. A variant of CBC-MAC has been standardized by NIST [10] under the name CMAC. We present a new analysis of CMAC obtaining a better bound in the process.
4. iCMAC: a variant of CBC-MAC is described and analysed. This variant improves upon CMAC by reducing one invocation of π for messages of length greater than n.
5. iVCBC-MAC: this is a variant of VCBC-MAC which removes the problems of VCBC-MAC mentioned above without increasing the number of invocations of ρ.

Parallel MAC schemes start with protected counter sum (PCS) [4] and PMAC [7, 23]. PMAC [23] uses a tweakable block cipher (TBC) and the design of the TBC requires the solution of a discrete log problem over $\mathbb{F}_{2^n}$. The designer in [23] provides solutions for $n = 64$ and $n = 128$; but, for $n = 256$ this will be difficult. Generalization of the TBC and PMAC has been done [8]. This can avoid the discrete log computation but is then (slightly) slower since it requires more masking operations.

We describe a new variant of PMAC called iPMAC which does not require discrete log computation but unlike [8] uses the same number of masking operations as in [23]. The PRF-bound that we obtain for iPMAC is similar to that obtained in [19, 21] for PMAC. Our technique, on the other hand, is cleaner and easier to understand.

The PCS scheme [4] uses a function ρ which maps ℓ bits to n bits, with $\ell > n$. It is of the type

$$\rho(\underline{0}, \rho(\underline{1}, P_1) + \rho(\underline{2}, P_2) + \cdots + \rho(\underline{m}, P_m))$$

where $\underline{i}$ is the $(\ell - n)$-bit binary representation of i. The inputs $\underline{i}$ to ρ "wastes" $(\ell - n)$ bits per invocation of ρ. We show that a variant of PMAC, which we call VPMAC, also works with such ρ and provides efficiency improvement. PCS requires $(1 + |x|/n)$ invocations of ρ whereas VPMAC requires $(1 + |x|/\ell)$ invocations of ρ. Consequently, VPMAC requires approximately a fraction n/ℓ of invocations of ρ compared to PCS.

The constructions and the bounds that we obtain are given in Table 1.

Table 1. Features of constructions analysed in this work. The column "new?" indicates whether the construction has been proposed earlier or whether it is proposed in this work. The column "perm?" indicates whether the construction (and the bound) works only with a permutation or not. In the column "bound", we provide the obtained upper bounds on the advantage of any adversary which makes at most d queries and provides at most $\sigma \geq d$ blocks in all the queries. The tag is considered to be an element of $T = \{0,1\}^n$.

scheme	new?	perm?	bound
CBC-MAC	no	yes	$\frac{d(d-1)+\sigma(1+2d)}{\#T-1}$
VCBC-MAC	yes	no	$\frac{\sigma(2d+5\sigma)}{2\#T}$
CMAC	no	yes	$\frac{3d(d-1)+4\sigma(1+2d)}{\#T-1}$
iCMAC	yes	yes	$\frac{d(d-1)+\sigma(1+2d)}{\#T-1}$
iVCBC-MAC	yes	no	$\frac{\sigma(2d+5\sigma)}{2\#T}$
iPMAC	yes	yes	$\frac{d(d-1)+4\sigma(1+3d)}{2\#T}$
VPMAC	yes	no	$\frac{d(d-1)+2\sigma(1+3d)}{\#T}$

1.2 Previous and Related Works

Bellare, Kilian and Rogaway [1] showed that for CBC-MAC working on equal length strings the advantage of an adversary making d queries each having m n-bit blocks is bounded above by $2m^2d^2/2^n$. Maurer [18] and Vaudenay [25] gave different proofs for essentially the same bound (upto a constant). Bernstein [5] also gave a different short proof. Petrank and Rackoff [22] proved the same bound (with a different constant) on CBC for messages with the prefix property, i.e., no message is a prefix of another.

The bound was improved by Bellare, Pietrzak and Rogaway [2] who proved a bound of $\frac{md^2}{2^n}(12 + \frac{8m^3}{2^n})$ for messages with the prefix property, where m is the maximum number of n-bit blocks in any query. The dominating term in this expression is $md^2/2^n$, which improves upon the previous bound of $m^2d^2/2^n$. In fact, the importance of the work in [2] is that it was the first paper to prove a bound of the type $md^2/2^n$ for some PRF construction. However, the techniques used in [2] is very dense and difficult. There is an extensive combinatorial analysis which consists of counting directed acyclic graphs (DAGs) with certain properties. In this work, we also describe bounds better than the birthday bound (which are comparable to and slightly better than that in [2]) using much simpler probability calculations.

Black and Rogaway [6] described a variant of CBC called XCBC which handles arbitrary length messages using one block cipher key and two n-bit keys (where n is the block size). A variant of

XCBC using a single additional n-bit key was developed by Kurosawa and Iwata [14] and was later modified by Iwata and Kurosawa [11] so that no additional n-bit keys are required. This last version was standardized by the NIST of the USA for message authentication under the name CMAC [10].

We have mentioned that we describe variants of CBC-MAC called VCBC-MAC and iVCBC-MAC which use a compressing function rather than a block cipher. Independent of our work, another variant of CBC-MAC which also works with compressing functions has been described in [27] and a bound of $cm^2d^2/2^n$ for some constant c has been proved, where m is the maximum number of n-bit blocks in any query. The bound is comparable to what we obtain, though our construction makes lesser number of invocations of the compressing function.

As mentioned earlier, a parallel MAC scheme called protected counter sum was described by Bernstein [4]. This scheme uses a compressing function as its building block and cannot be replaced by a block cipher. Black and Rogaway [7] and later Rogaway [23] described block cipher based methods for parallel message authentication. The scheme in [7] was called PMAC and the one in [23] was called PMAC1; currently, the scheme in [23] itself is called PMAC. The construction in [23] is based on an efficient construction of tweakable block cipher (TBC) family; the notion of tweakable block ciphers was introduced in [16]. Chakraborty and Sarkar [8] generalised the TBC construction in [23] and hence obtained several variants of PMAC.

The bound on the advantage of PMAC forgery was shown to be $c\sigma^2/2^n$, for some constant c, where σ is the total number of n-bit blocks provided by the adversary in all its queries. Following the work in [2], this bound was improved by Minematsu and Matsushima [19] to $md^2/2^n$, where m is the maximum of the lengths of all the queried messages. Nandi and Mandal [21] showed a bound of $(5d\sigma - 3.5d^2)/2^n$ for PMAC.

Jutla [12] introduced the notion of construction of pseudorandom functions based on directed acyclic graphs (DAG). This class was further analysed by Nandi [20].

Note. In this paper, "random" does *not* necessarily mean "uniform random". When required, we will explicitly mention the uniformity condition. In most papers, q is used to denote the number of queries made by an adversary. We use d to denote this quantity. This may be taken to denote the fact that we are counting only *distinct* queries.

2 Basic Definitions and Results

Let S be a finite non-empty set and define $\chi_d(S)$ to be

$$\chi_d(S) = \{(x_1, \ldots, x_d) \in S^d : x_i \neq x_j, 1 \leq i < j \leq d\}. \tag{1}$$

In other words, $\chi_d(S)$ consists of all $(x_1, \ldots, x_d)$ such that $x_1, \ldots, x_d$ are distinct elements of S.

Let ρ be a function from S to T and d be a positive integer. The natural extension of ρ to a function from S^d to T^d obtained by applying ρ to each component will be denoted by $\Gamma_{\rho,d}$, i.e., for any $\mathbf{x} = (x_1, \ldots, x_d) \in S^d$,

$$\Gamma_{\rho,d}(\mathbf{x}) = \Gamma_{\rho,d}(x_1, \ldots, x_d) = (\rho(x_1), \ldots, \rho(x_d)). \tag{2}$$

Note. The number of elements in a set S will be denoted by $\#S$ and the absolute value of a real number a will be denoted by $|a|$. The length of a binary string x will be denoted by $\mathsf{len}(x)$.

Definition 1. *A set U is said to be a δ-large subset of a set S, if U is a subset of S and $\#U \geq \delta \times \#S$.*

Let S be a non-empty set and λ be a function from S to non-negative integers, i.e., we associate a non-negative integer with each element of S. In our applications, the set S will consist of binary strings and for $x \in S$, $\lambda(x)$ will denote the number of n-bit blocks (counting partial blocks) into which x can be divided, for some fixed positive integer n. For the moment, however, we will not be requiring this interpretation. We will simply call λ to be a length function on S. Given $\mathbf{x} = (x_1, \ldots, x_d) \in S^d$, we define $\lambda(\mathbf{x}) = \sum_{i=1}^{d} \lambda(x_i)$.

Let $m \geq d \geq 1$. The following two functions will be useful later.

$$\left.\begin{array}{l} p(m,d) = m(m-1)(m-2)\cdots(m-(d-1)) \\ r(m,d) = \left(1-\frac{1}{m}\right)\left(1-\frac{2}{m}\right)\cdots\left(1-\frac{d-1}{m}\right). \end{array}\right\} \tag{3}$$

Proposition 1. *Let $m \geq d \geq 1$. Then $\dfrac{1}{p(m,d)} \geq \dfrac{1}{m^d}$ and $\dfrac{p(m,d)}{m^d} = r(m,d) \geq 1 - \dfrac{d(d-1)}{2m}$.*

Proof: The bound on $p(m,d)$ is obvious and the bound on $r(m,d)$ follows on noting that $(1 - a/m)(1 - b/m) \geq (1 - (a+b)/m)$. □

Proposition 2. *For a finite nonempty set S, $\#\chi_d(S) = p(\#S, d) \geq \left(1 - \dfrac{d(d-1)}{2\#S}\right)(\#S)^d$. Consequently, $\chi_d(S)$ is a $\left(1 - \frac{d(d-1)}{2\#S}\right)$-large subset of S.*

We will be studying functions from a set S to a finite non-empty set T. The set itself could be (countably) infinite, but, we will be interested in a finite number of elements of S. Our main object of study are random functions from S to T. Let T^S denote the set of all functions from S to T. By an uniform random function ρ from S to T we will mean an element T^S chosen uniformly at random. A more convenient way to view ρ is the following. For any $\mathbf{x} \in \chi_d(S)$, $\mathit{\Gamma}_{\rho,d}(\mathbf{x})$ is uniformly distributed over T^d, i.e., in other words, the outputs of ρ on distinct inputs are independent and uniformly distributed. If $S = T$, then we can talk about a permutation π of T, which is a bijection $\pi : T \to T$. By a uniform random permutation, we will mean a permutation chosen uniformly at random from the set of all permutations of T. Again, this means that for any $\mathbf{x} \in \chi_d(S)$, $\mathit{\Gamma}_{\rho,d}(\mathbf{x})$ is uniformly distributed over $\chi_d(T)$. Other examples of random (but not uniform random) functions can be obtained: let T be a finite field and $S = T^2$; choose a uniform random $\alpha \in T$ and define $\rho : S \to T$ as $\rho(a_0, a_1) = a_1\alpha + a_0$. Then ρ is also a random function but not a uniform random function.

2.1 Useful Inequalities

The following results will be useful later.

Lemma 1. *Let $m_1, \ldots, m_d$ be non-negative integers and $\sigma = \sum_{i=1}^{d} m_i$. Then*

1. $\displaystyle\sum_{1 \leq i < j \leq d} \min(m_i, m_j) \leq \sum_{1 \leq i < j \leq d} \max(m_i, m_j) \leq d\sigma.$
2. $\displaystyle\sum_{1 \leq i < j \leq d} (m_i + m_j) \leq 2d\sigma.$

Proof: Without loss of generality suppose that $m_1 \geq m_2 \geq \cdots \geq m_d$.

$$\begin{aligned}\sum_{1\leq i<j\leq d} \max(m_i, m_j) &= \sum_{i=1}^{d}\sum_{j=i+1}^{d} \max(m_i, m_j) \\ &= (d-1)m_1 + (d-2)m_2 + \cdots + m_{d-1} \\ &\leq d\sum_{i=1}^{d} m_i \\ &= d\sigma.\end{aligned}$$

Point (2) follows on noting that $m_i + m_j \leq 2\max(m_i, m_j)$. □

Lemma 2. *Let $m_1, \ldots, m_d$ be non-negative integers and $\sigma = \sum_{i=1}^{d} m_i$. Then*

$$\sum_{i=1}^{d}\sum_{j=i+1}^{d}\sum_{k=1}^{\max(m_i,m_j)} \min(k, \min(m_i, m_j)) \leq \sigma^2.$$

Proof: Since both max() and min() are symmetric in their arguments, without loss of generality we assume that $m_1 \geq m_2 \geq \cdots \geq m_d$. Then

$$\begin{aligned}\sum_{i=1}^{d}\sum_{j=i+1}^{d}\sum_{k=1}^{\max(m_i,m_j)} \min(k, \min(m_i, m_j)) &= \sum_{i=1}^{d}\sum_{j=i+1}^{d}\sum_{k=1}^{m_i} \min(k, m_j) \\ &= \sum_{i=1}^{d}\sum_{j=i+1}^{d} (1 + 2 + \cdots + m_j + (m_i - m_j)m_j) \\ &= \sum_{i=1}^{d}\sum_{j=i+1}^{d} \left(\frac{m_j(m_j+1)}{2} + m_i m_j - m_j^2\right) \\ &\leq \sum_{i=1}^{d}\sum_{j=i+1}^{d} m_i m_j \\ &= m_1(m_2 + m_3 + \cdots + m_d) + m_2(m_3 + \cdots + m_d) \\ &\quad + \cdots + m_{d-1}m_d \\ &\leq (m_1 + m_2 + \cdots + m_d)^2 \\ &= \sigma^2.\end{aligned}$$

□

Note. It can be shown that $\sum_{1\leq i<j\leq d} \max(m_i^2, m_j^2)$ is bounded above by $d\sigma^2$. The constants in the above bounds can be improved, but, then they will not be nice. The stated bounds are good enough for our purposes.

2.2 Interpolation and Collision Probabilities

Let S and T be sets and F be a random function from S to T. For $\mathbf{x} \in \chi_d(S)$ and $\mathbf{y} \in T^d$, the probability $\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] = \Pr[F(x_1) = y_1, \ldots, F(x_d) = y_d]$ has been called a d-interpolation probability in [4].

Definition 2. *Let $F : S \to T$ be a random function and λ be a length function on S. Let U be a subset of T^d. We will say that the function F is (d, σ, δ)-interpolating on U with respect to λ if for all $\mathbf{x} \in \chi_d(S)$ with $\lambda(\mathbf{x}) \leq \sigma$ and for all $y \in U$,*

$$\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] \geq \delta / \#U.$$

Here δ could possibly depend on d and σ.

A collision for a function F consists of two distinct elements x and x' in the domain of F such that $F(x) = F(x')$.

Definition 3. *Let F be a random function with domain S.*

1. *Let $x \neq x'$ be elements of S. The event $\mathsf{Coll}_F(x, x')$ is defined to be the event $F(x) = F(x')$. When F is clear from the context, then we will omit the subscript F.*
2. *For $\mathbf{x} \in \chi_d(S)$, we define the collision bound $\mathsf{CB}_F(\mathbf{x})$ to be*

$$\mathsf{CB}_F(\mathbf{x}) = \sum_{1 \leq i < j \leq d} \Pr[F(x_i) = F(x_j)].$$

An immediate consequence of this definition is the following result.

Lemma 3. *Let $F : S \to T$ be a random function and $\mathbf{x} \in \chi_d(S)$. Then*

$$\Pr[\Gamma_{F,d}(\mathbf{x}) \in \chi_d(T)] \geq 1 - \mathsf{CB}_F(\mathbf{x}). \tag{4}$$

Proof: Let $\mathcal{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_d) = \Gamma_{F,d}(\mathbf{x}) = (F(x_1), \ldots, F(x_d))$. Then

$$\Pr\left[\bigvee_{1 \leq i < j \leq d} (\mathcal{Y}_i = \mathcal{Y}_j)\right] \leq \sum_{1 \leq i < j \leq d} \Pr\left[(\mathcal{Y}_i = \mathcal{Y}_j)\right] = \mathsf{CB}_F(\mathbf{x})$$

and

$$\Pr\left[\mathcal{Y} \in \chi_d(T)\right] = \Pr\left[\bigwedge_{1 \leq i < j \leq d} (\mathcal{Y}_i \neq \mathcal{Y}_j)\right] = 1 - \Pr\left[\bigvee_{1 \leq i < j \leq d} (\mathcal{Y}_i = \mathcal{Y}_j)\right] \geq 1 - \mathsf{CB}_F(\mathbf{x}).$$

□

We define two kinds of collision resistance for F, depending on whether the collision probability depends on the length function or not.

Definition 4. *Let F be a random function with domain S and λ be a length function on S.*

1. *F is said to be ε-CR, if for any two distinct $x, x' \in S$, $\Pr[\mathsf{Coll}_F(x, x')] \leq \varepsilon$, for some constant ε.*
2. *F is said to be ε-CR with respect to λ, if for any two distinct $x, x' \in S$, $\Pr[\mathsf{Coll}_F(x, x')] \leq \varepsilon \times \max(\lambda(x_1), \lambda(x_2))$, for some constant ε.*

The following result shows the intuitively clear fact that if collisions are unlikely for a random function F, then it behaves like an injective function, i.e., with high probability distinct inputs are mapped to distinct outputs.

Lemma 4. *Let d and $\sigma \geq d$ be positive integers; and $F : S \to T$ be a random function and λ be a length function on S. Let $\mathbf{x} \in \chi_d(S)$ and $\sigma = \lambda(\mathbf{x})$.*

1. *If F is ε-CR, then* $\Pr[\Gamma_{F,d}(\mathbf{x}) \in \chi_d(T)] \geq \left(1 - \frac{d(d-1)\varepsilon}{2}\right)$.
2. *If F is ε-CR with respect to λ, then* $\Pr[\Gamma_{F,d}(\mathbf{x}) \in \chi_d(T)] \geq (1 - \varepsilon d\sigma)$.

Proof: We obtain bounds on $\mathsf{CB}_F(\mathbf{x})$ and then the results follows from Lemma 3. In the first case, it is easily seen that $\mathsf{CB}_F(\mathbf{x}) \leq (d(d-1)\varepsilon)/2$. For the second case, we have

$$\begin{aligned}\mathsf{CB}_F(\mathbf{x}) &\leq \sum_{1 \leq i < j \leq d} \Pr\left[F(x_i) = F(x_j)\right] \\ &\leq \sum_{1 \leq i < j \leq d} \varepsilon \max(\lambda(x_i), \lambda(x_j)) \\ &\leq \varepsilon d\sigma.\end{aligned}$$

The last inequality follows from Lemma 1. □

It may be noted that having low collision probabilities does not imply high interpolation probabilities. For example, let T be a finite field and F_α be a random function mapping T^2 to T by $(a_0, a_1) \mapsto a_0 + \alpha a_1$, where α is a uniform random element of T. Then it is easy to show that F_α has low collision probabilities whereas the value of F_α on two distinct inputs uniquely determines α and hence interpolation probabilities for $d > 2$ are low.

2.3 Linear Functions With Low Collision Probabilities

In some of the constructions to be described later, we will be making use of linear functions with certain properties. The purpose of this section is to define these properties and mention known functions which possesses these properties.

Definition 5. *Let $T = GF(2^n)$ be the finite field having 2^n elements. We say that a function $\psi : T \to T$ is a proper masking function if it satisfies the following properties.*

1. *For any fixed $\alpha \in T$; any non-negative integer k with $0 \leq k \leq 2^n - 2$; and a uniform random $\beta \in T$;* $\Pr[\psi^k(\beta) = \alpha] = 1/\#T$.
2. *For any fixed $\alpha \in T$; distinct integers k_1, k_2 with $0 \leq k_1 < k_2 \leq 2^n - 2$; and a uniform random $\beta \in T$;* $\Pr[\psi^{k_1}(\beta) \oplus \psi^{k_2}(\beta) = \alpha] = 1/\#T$.
3. *For any fixed $\alpha \in T$; distinct integers k_1, k_2 with $0 \leq k_1 < k_2 \leq 2^n - 2$; and uniform random $(\beta_1, \beta_2) \in \chi_2(T)$,* $\Pr[\psi^{k_1}(\beta_1) \oplus \psi^{k_2}(\beta_2) = \alpha] = 1/(\#T - 1)$.

There is a very general class of linear functions satisfying Definition 5.

Proposition 3. *Let $T = GF(2^n)$ and $\psi : T \to T$ be a linear function whose minimal polynomial $\tau(u)$ over $GF(2)$ is of degree n and is primitive over $GF(2)$. Then ψ satisfies Definition 5.*

Proof: Since $\tau(u)$ is primitive over $GF(2)$ and is of degree n, it follows that ψ is invertible and so for every non-negative integer k, ψ^k is also invertible. The first point follows from this observation.

Define $\phi_{i,j} : T \to T$ as $\phi_{i,j}(\gamma) = \psi^i(\gamma) \oplus \psi^j(\gamma)$. The second point will follow if we can show that $\phi_{i,j}$ is a bijection. For this, it is sufficient to show that $\phi_{i,j}$ is an injection. So, suppose that γ and γ' are distinct elements of T and let, if possible, $\phi_{i,j}(\gamma) = \phi_{i,j}(\gamma')$. Set $\delta = \gamma \oplus \gamma'$ and note that since $\gamma \neq \gamma'$, we have δ to be non-zero. Then

$$\begin{aligned}0 &= \phi_{i,j}(\gamma) \oplus \phi_{i,j}(\gamma') \\ &= \psi^i(\gamma) \oplus \psi^j(\gamma) \oplus \psi^i(\gamma') \oplus \psi^j(\gamma') \\ &= (\psi^i \oplus \psi^j)(\delta). \end{aligned} \tag{5}$$

For any non-zero element ν of $\mathbb{F}_{2^n}$, define $M_\nu(u)$ to be the minimal degree polynomial such that $(M_\nu(\psi))(\nu) = 0$. Since $\tau(u)$ is the minimal polynomial of ψ it follows that $\tau(\psi) = 0$, i.e., $\tau(\psi)$ maps all elements of T to 0. As a result, $(\tau(\psi))(\nu) = 0$. By the minimality of $M_\nu(u)$ it follows that $M_\nu(u)$ divides $\tau(u)$. But, $\tau(u)$ is irreducible and so $M_\nu(u) = \tau(u)$.

Consider the minimal polynomial $M_\delta(u)$ of δ. Since δ is non-zero, by the above argument, we have $M_\delta(u) = \tau(u)$. Also, from (5), it follows that $\tau(u) = M_\delta(u)$ divides $u^i \oplus u^j = u^i(1 \oplus u^{j-i})$ (assuming without loss of generality that $i < j$). Since $\tau(u)$ is primitive, it does not divide u^i and so $\tau(u)|(1 \oplus u^{j-i})$. It is well known that if $\tau(u)$ is a primitive polynomial of degree n, then it does not divide $1 \oplus u^i$ for any i with $0 < i < 2^n - 1$ (see for example [15]). Since $0 \leq i < j < 2^n - 1$, we have $0 < j - i < 2^{n-1}$ and hence, $\tau(u)|(1 \oplus u^{j-i})$ contradicts the primitivity property of $\tau(u)$. This shows that $\phi_{i,j}$ is a injection.

Consider the map $\zeta_{k_1,k_2} : T^2 \rightarrow T$ which takes (β_1, β_2) to $\psi^{k_1}(\beta_1) \oplus \psi^{k_2}(\beta_2)$. We count the number of pre-images of $\alpha \in T$ for ζ_{k_1,k_2}. For every value of β_1, $\beta_2 = \psi^{-k_2}(\psi^{k_1}(\beta_1) \oplus \alpha)$ is unique. Hence, there are $\#T$ pre-images for any α. Since (β_1, β_2) is uniformly distributed over $\chi_2(T)$, the result follows. □

There are known examples of ψ which satisfy Proposition 3.

1. In this case, $\psi : \beta \mapsto u\beta \bmod \tau(u)$ and $\psi^k : \beta \mapsto u^k\beta \bmod \tau(u)$.
2. Suitable word oriented linear feedback shift registers (LFSRs) can also be used. Using word oriented LFSRs provides a faster masking strategy.

The first strategy has been used in various constructions [7, 23, 11, 10], though the required properties has not really being brought out as clearly as given in Definition 5. The second strategy has been suggested in [8, 24]. The security of the constructions to be described later do not depend on the actual implementation of ψ. We will simply use the properties given by Definition 5.

2.4 Adversarial Model

We will consider computationally unbounded adversaries and consequently, without loss of generality, we consider an adversary $\mathcal{A}$ to be a deterministic algorithm. (This approach has been used earlier [26, 4].) This algorithm interacts with an oracle and outputs a bit. The oracle takes as input an element of a set S and produces as output an element of a finite non-empty set T. The adversary $\mathcal{A}$ makes d queries to the oracle and then produces its output. Without loss of generality, we will make the assumption that the adversary never repeats a query.

Since $\mathcal{A}$ is deterministic, the behaviour of $\mathcal{A}$ can be described by a sequence of functions $\phi_1, \phi_2, \ldots, \phi_d$ and another function ϕ. The function $\phi_1()$ does not take any input and produces $x_1 \in S$ as output. This is the first input provided by $\mathcal{A}$ to the oracle and gets back y_1 in return; $\mathcal{A}$ then computes $x_2 = \phi_2(y_1)$ as its second input and gets back y_2; in the general case, $\mathcal{A}$ computes $x_i = \phi_i(y_1, \ldots, y_{i-1})$ as its i-th oracle input and gets back y_i. Since no query is repeated, $\mathbf{x} = (x_1, \ldots, x_d) \in \chi_d(S)$.

Finally, the function ϕ takes as input $(y_1, \ldots, y_d)$ and produces as output a bit, which is taken to be the output of $\mathcal{A}$. Note that the functions $\phi_1, \ldots, \phi_d$ and ϕ do not depend on the oracle. We will use the notation $\phi_1^{\mathcal{A}}, \phi_2^{\mathcal{A}}, \ldots, \phi_d^{\mathcal{A}}$ and $\phi^{\mathcal{A}}$ when we wish to emphasize the association of the functions to the adversary $\mathcal{A}$. Denote by $\Pr[\mathcal{A}^F \rightarrow 1]$ the probability that $\mathcal{A}$ outputs 1, when the oracle is F. The probability is over the randomness of F since $\mathcal{A}$ itself is deterministic. Formally,

$$\Pr[\mathcal{A}^F \rightarrow 1] = \sum_{(y_1,\ldots,y_d) \in T^d} \Pr[(F(\phi_1^{\mathcal{A}}()) = y_1) \wedge (F(\phi_2^{\mathcal{A}}(y_1)) = y_2)$$

$$\wedge \cdots \wedge (F(\phi_d^{\mathcal{A}}(y_1, \ldots, y_{d-1})) = y_d) \wedge (\phi^{\mathcal{A}}(y_1, \ldots, y_d) = 1)]$$

$$= \sum_{(y_1,\ldots,y_d) \in \mathsf{Acc}(\mathcal{A})} \Pr[(F(\phi_1^{\mathcal{A}}()) = y_1) \wedge (F(\phi_2^{\mathcal{A}}(y_1)) = y_2)$$

$$\wedge \cdots \wedge (F(\phi_d^{\mathcal{A}}(y_1, \ldots, y_{d-1})) = y_d)]$$

where

$$\mathsf{Acc}(\mathcal{A}) = \{(y_1, \ldots, y_d) : \phi^{\mathcal{A}}(y_1, \ldots, y_d) = 1\}. \tag{6}$$

The set $\mathsf{Acc}(\mathcal{A})$ is the set of $(y_1, \ldots, y_d)$ which result in $\mathcal{A}$ producing 1 as output. This set does not depend on F and is determined entirely by $\mathcal{A}$.

Suppose that the oracle is instantiated twice by two random functions F and G both mapping S to T. Then the advantage of $\mathcal{A}$ in distinguishing between F and G is defined to be

$$\mathbf{Adv}_{\mathcal{A},(F,G)} = \Pr[\mathcal{A}^F \to 1] - \Pr[\mathcal{A}^G \to 1]. \tag{7}$$

If G is a uniform random function from S to T, then the advantage will be denoted by $\mathbf{Adv}_{\mathcal{A},F}$.

Let the domain of F be a non-empty set S and λ be a length function on S. For positive integers d and σ, we define $\mathbf{Adv}_F(d, \sigma)$ to be the maximum advantage of any adversary which makes at most d distinct queries $x_1, \ldots, x_d$ such that $\sum_{i=1}^{q} \lambda(x_i) \leq \sigma$. The quantity $\mathbf{Adv}_F(d, \sigma)$ is the PRF-advantage of F. Alternatively, if $\mathbf{Adv}_F(d, \sigma) \leq \varepsilon$, then we say that F is a (d, σ, ε)-PRF.

Vaudenay proved a useful result (Lemma 22 in [26]) which reduces the task of bounding the advantage of an adaptive adversary to that of a probability calculation. A special version of this result was given by Bernstein (Theorem 3.1 in [4]) with a different proof. Theorem 1 below is a restatement of Vaudenay's result in a form suitable for our requirement. The ideas given in the proof below are from [26, 4]; we provide more details.

Theorem 1. *Let d and $\sigma \geq d$ be positive integers; F be a random function from a set S to a set T; and λ be a length function on S. Suppose that U is a $(1 - \varepsilon_1)$-large subset of T^d and F is $(d, \sigma, 1 - \varepsilon_2)$-interpolating on U with respect to λ. Then,*

$$\mathbf{Adv}_F(d, \sigma) \leq \varepsilon_1 + \varepsilon_2.$$

Note. Here ε_2 could depend on d and σ and in our applications later, it indeed does.

Proof: For any adversary $\mathcal{A}$, let $V = \overline{\mathsf{Acc}(\mathcal{A})}$, where $\mathsf{Acc}(\mathcal{A})$ is as defined in (6). Then V is the subset of T^d such that if $\mathcal{A}$ receives any $\mathbf{y} \in V$ as reply to the oracle queries, then $\mathcal{A}$ outputs 0, i.e., $V = \{\mathbf{y} \in T^d : \phi^{\mathcal{A}}(\mathbf{y}) = 0\}$. As noted earlier, V is independent of the function F and depends only on the adversary $\mathcal{A}$. Then for any random function F,

$$\sum_{y \in V} \Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}] + \sum_{y \notin V} \Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}] = 1. \tag{8}$$

Also,

$$\Pr[\mathcal{A}^F \to 1] = \sum_{\mathbf{y} \notin V} \Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}] \text{ and similarly, } \Pr[\mathcal{A}^{F^*} \to 1] = \sum_{\mathbf{y} \notin V} \Pr[\mathit{\Gamma}_{F^*,d}(\mathbf{x}) = \mathbf{y}]. \tag{9}$$

Here F^* is a uniform random function from S to T. So,

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{A}} &= \Pr[\mathcal{A}^F \to 1] - \Pr[\mathcal{A}^{F^*} \to 1] \\
&\overset{(9)}{=} \sum_{\mathbf{y} \notin V} \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] - \sum_{\mathbf{y} \notin V} \Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] \\
&\overset{(8)}{=} \sum_{\mathbf{y} \in V} (\Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] - \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}]) \\
&= \sum_{\mathbf{y} \in V, \mathbf{y} \in U} (\Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] - \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}]) \\
&\quad + \sum_{\mathbf{y} \in V, \mathbf{y} \notin U} (\Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] - \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}]) . \qquad (10)
\end{aligned}$$

Since F is $(d, \sigma, 1-\varepsilon_2)$-interpolating on U with respect to λ, we have that for all $\mathbf{x} \in \chi_d(S)$ with $\lambda(\mathbf{x}) \leq \sigma$ and for all $\mathbf{y} \in U$,

$$\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] \geq (1-\varepsilon_2)/(\#U) \geq (1-\varepsilon_2)/(\#T)^d. \qquad (11)$$

F^* is a random function from S to T, and hence, for all $\mathbf{x} \in \chi_d(S)$ and for all $\mathbf{y} \in T^d$, $\Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] = 1/(\#T)^d$. Using this and (11) we have for all $\mathbf{x} \in \chi_d(S)$ and for all $\mathbf{y} \in U$,

$$\Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] - \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] \leq \varepsilon_2 \Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}].$$

Consequently,

$$\sum_{\mathbf{y} \in V, \mathbf{y} \in U} (\Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] - \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}]) \leq \varepsilon_2 \sum_{\mathbf{y} \in V, \mathbf{y} \in U} \Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] \leq \varepsilon_2. \qquad (12)$$

By the fact that U is a $(1-\varepsilon_1)$-large subset of T^d, $(\#T)^d - (\#U) \leq \varepsilon_1 (\#T)^d$, and so,

$$\begin{aligned}
\sum_{\mathbf{y} \in V, \mathbf{y} \notin U} (\Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] - \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}]) &\leq \sum_{\mathbf{y} \in V, \mathbf{y} \notin U} \Pr[\Gamma_{F^*,d}(\mathbf{x}) = \mathbf{y}] \\
&= \sum_{\mathbf{y} \in V, \mathbf{y} \notin U} \frac{1}{(\#T)^d} \\
&\leq \frac{(\#T)^d - (\#U)}{(\#T)^d} \\
&\leq \varepsilon_1. \qquad (13)
\end{aligned}$$

Substituting (12) and (13) in (10) gives the desired inequality. □

Informally, Theorem 1 states that if F has high interpolation probability on a large subset U of T^d, then F is a PRF. The case where U equals T^d was independently proved by Bernstein (Theorem 3.1 in [4]).

3 Domain Extenders

Many constructions use only a block cipher and the output of F_1 is obtained by invoking a block cipher several times. Such functions can be viewed as composition of the type $F = F_1 \circ F_2$, where

F_2 is a uniform random permutation and F_1 is built using F_2. When considered as keyed functions, F will have a single key which is the key for F_2.

More generally, suppose that we are given a random function ρ which maps from a set U to T. Using ρ, we wish to construct another random function F which maps from a set S to T, where S is larger than U. In other words, we wish to extend the domain from U to S. To capture such constructions, we have the following definition.

Definition 6. *Let $\rho : U \to T$ be a random function. A function $F : S \to T$ is said to be a domain extender for ρ if $F = \rho \circ F_1$, where $F_1 : S \to U$ and F_1 satisfies the following conditions.*

1. *On any input, F_1 invokes ρ a finite number of times.*
2. *The only randomness involved in computing F_1 comes from the invocations of ρ.*

We associate a canonical length function λ to S. For every x in S, $\lambda(x)$ denotes the total number of times ρ is invoked to compute the final output of F.

We wish to compute $\Pr[\Gamma_{F,d}(\mathbf{x}) = (\mathbf{y})]$, where $\mathbf{x} \in \chi_d(S)$ and $\mathbf{y} \in T^d$. F_1 and ρ "interact" and hence we need to account for such possibilities. To this end, we make the following definition.

Definition 7. *Let $\rho : U \to T$ be a random function and $F = \rho \circ F_1$ be a map from S to T satisfying Definition 6. For $x, x' \in S$ with $x \neq x'$, let $\mathcal{Z} = F_1(x)$, $\mathcal{Z}' = F_1(x')$; $\lambda(x) = m + 1$, $\lambda(x') = m' + 1$; and let $\mathcal{U}_1, \ldots, \mathcal{U}_m$ and $\mathcal{U}'_1, \ldots, \mathcal{U}'_m$ be the inputs to the different invocations of ρ in the computation of $F_1(x)$ and $F_1(x')$ respectively.*

1. *Define* $\mathsf{Self\text{-}Disjoint}(x)$ *to be the event* $\bigwedge_{i=1}^{m}(\mathcal{Z} \neq \mathcal{U}_i)$.
2. *Define* $\mathsf{Pairwise\text{-}Disjoint}(x, x')$ *to be the event* $\left(\bigwedge_{i=1}^{m}(\mathcal{Z}' \neq \mathcal{U}_i) \wedge \bigwedge_{j=1}^{m'}(\mathcal{Z} \neq \mathcal{U}'_i)\right)$.

Definition 8. *Continuing with Definition 7, we say that F_1 is $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ, if for all pairs of distinct $x, x' \in S$, there are constants $\varepsilon_1, \varepsilon_2$, such that*

$$\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \leq \varepsilon_1(\lambda(x)) \text{ and } \Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] \leq \varepsilon_2(\lambda(x) + \lambda(x')).$$

Note that the notion of disjointness is defined for F_1 rather than for F.

We now prove the main result on domain extenders. In the result below, we consider ρ to be either a uniform random function or a uniform random permutation. The more general case is when we have lower bound on the interpolation probabilities of ρ. A result of this type can be proved as in the result below; but, such a result is of less practical interest, since, in practice, ρ will mostly be a block cipher which is modelled as a uniform random permutation.

Theorem 2. *Let $\rho : U \to T$ be a random function and $F = \rho \circ F_1$ be a map from S to T satisfying Definition 6. Suppose that F_1 is ε-CR with respect to the length function λ and also $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ. Then for positive integers d and $\sigma \geq d$ the following holds.*

1. *If ρ is a uniform random function, then*

$$\mathbf{Adv}_F(d, \sigma) \leq \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2).$$

2. *If $U = T$ and ρ is a uniform random permutation, then*

$$\mathbf{Adv}_F(d, \sigma) \leq \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2) + \frac{d(d-1)}{2\#T}.$$

Proof: Let $\mathbf{x} = (x_1, \ldots, x_d) \in \chi_d(S)$ with $m_i + 1 = \lambda(x_i)$. Then $\sigma = d + \sum_{i=1}^{d} m_i$. Set $\mathcal{Z}_i = F_1(x_i)$ and let $U_{i,1}, \ldots, U_{i,m_i}$ be the inputs to ρ in the computation of $\mathcal{Z}_i$. Let $\mathsf{Distinct}(\mathbf{x})$ and $\mathsf{Disjoint}(\mathbf{x})$ be the events

$$\mathsf{Distinct}(\mathbf{x}) = \bigwedge_{1 \le i < j \le d} (\mathcal{Z}_i \neq \mathcal{Z}_j) \tag{14}$$

and

$$\mathsf{Disjoint}(\mathbf{x}) = \bigwedge_{i=1}^{d} \bigwedge_{j=1}^{d} \bigwedge_{k=1}^{m_j} (\mathcal{Z}_i \neq \mathcal{U}_{j,k}). \tag{15}$$

The event $\mathsf{Distinct}(\mathbf{x})$ is the event $\Gamma_{F_1,d}(\mathbf{x}) \in \chi_d(U)$. Using the fact that F_1 is ε-CR with respect to λ and Lemma 4, we have

$$\Pr\left[\overline{\mathsf{Distinct}(\mathbf{x})}\right] \le d\sigma\varepsilon. \tag{16}$$

We have

$$\begin{aligned}
\Pr\left[\overline{\mathsf{Disjoint}(\mathbf{x})}\right] &= \Pr\left[\bigvee_{i=1}^{d} \bigvee_{j=1}^{d} \bigvee_{k=1}^{m_j} (\mathcal{Z}_i = \mathcal{U}_{j,k})\right] \\
&= \Pr\left[\bigvee_{i=1}^{d} \bigvee_{k=1}^{m_i} (\mathcal{Z}_i = \mathcal{U}_{i,k})\right] + \Pr\left[\bigvee_{i=1}^{d} \bigvee_{\substack{j=1, \\ j \neq i}}^{d} \bigvee_{k=1}^{m_j} (\mathcal{Z}_i = \mathcal{U}_{j,k})\right] \\
&\le \sum_{i=1}^{d} \Pr\left[\bigvee_{k=1}^{m_i} (\mathcal{Z}_i = \mathcal{U}_{i,k})\right] + \Pr\left[\bigvee_{i=1}^{d} \bigvee_{j=i+1}^{d} \left(\bigvee_{k=1}^{m_j} (\mathcal{Z}_i = \mathcal{U}_{j,k}) \vee \bigvee_{k=1}^{m_i} (\mathcal{Z}_j = \mathcal{U}_{i,k})\right)\right] \\
&\le \sum_{i=1}^{d} \Pr\left[\overline{\mathsf{Self\text{-}Disjoint}(x_i)}\right] + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \Pr\left[\left(\bigvee_{k=1}^{m_j} (\mathcal{Z}_i = \mathcal{U}_{j,k}) \vee \bigvee_{k=1}^{m_i} (\mathcal{Z}_j = \mathcal{U}_{i,k})\right)\right] \\
&\le \sum_{i=1}^{d} \Pr\left[\overline{\mathsf{Self\text{-}Disjoint}(x_i)}\right] + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \Pr\left[\overline{\mathsf{Pairwise\text{-}Disjoint}(x_i, x_j)}\right].
\end{aligned} \tag{17}$$

Since F_1 is $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ, we have

$$\Pr\left[\overline{\mathsf{Self\text{-}Disjoint}(x_i)}\right] \le \varepsilon_1 \lambda(x_i) \text{ and } \Pr\left[\overline{\mathsf{Pairwise\text{-}Disjoint}(x_i, x_j)}\right] \le \varepsilon_2(\lambda(x_i) + \lambda(x_j)).$$

Using (17), we have

$$\begin{aligned}
\Pr\left[\overline{\mathsf{Disjoint}(\mathbf{x})}\right] &\le \sum_{i=1}^{d} \Pr\left[\overline{\mathsf{Self\text{-}Disjoint}(x_i)}\right] + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \Pr\left[\overline{\mathsf{Pairwise\text{-}Disjoint}(x_i, x_j)}\right] \\
&\le \varepsilon_1 \sum_{i=1}^{d} \lambda(x_i) + \varepsilon_2 \sum_{i=1}^{d} \sum_{j=i+1}^{d} (\lambda(x_i) + \lambda(x_j)) \\
&\le \varepsilon_1 \sigma + 2\varepsilon_2 d\sigma.
\end{aligned} \tag{18}$$

Lemma 1 is used in the last line. Combining (16) and (18) we have

$$\begin{aligned}\Pr[\mathsf{Distinct} \wedge \mathsf{Disjoint}] &= 1 - \Pr\left[\overline{\mathsf{Distinct}} \vee \overline{\mathsf{Disjoint}}\right] \\ &\geq 1 - \Pr\left[\overline{\mathsf{Distinct}}\right] - \Pr\left[\overline{\mathsf{Disjoint}}\right] \\ &\geq 1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2). \end{aligned} \tag{19}$$

Let $\mathbf{y} \in T^d$. Then,

$$\begin{aligned}\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] &\geq \Pr[(\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}) \wedge (\mathsf{Distinct} \wedge \mathsf{Disjoint})] \\ &= \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Distinct} \wedge \mathsf{Disjoint})] \times \Pr[\mathsf{Distinct} \wedge \mathsf{Disjoint}] \\ &\geq (1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2)) \times \Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Distinct} \wedge \mathsf{Disjoint})]. \end{aligned} \tag{20}$$

The event "$\mathsf{Distinct} \wedge \mathsf{Disjoint}$" means that the random variables $\mathcal{Z}_1, \ldots, \mathcal{Z}_d$ have distinct values and they are different from any previous inputs to ρ obtained during the computations of $\mathcal{Z}_i = F_1(x_i)$. In other words, the event "$\mathsf{Distinct} \wedge \mathsf{Disjoint}$" ensure that the set $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_d\}$ is a set of d "new" values in the domain of ρ. If ρ is a uniform random function, then $\Gamma_{\rho,d}(\mathcal{Z}_1, \ldots, \mathcal{Z}_d)$ is uniformly distributed over T^d, while, if ρ is a uniform random permutation, then $\Gamma_{\rho,d}(\mathcal{Z}_1, \ldots, \mathcal{Z}_d)$ is uniformly distributed over $\chi_d(T)$. We consider these two cases separately.

1. If ρ is a uniform random function, then for any $\mathbf{y} \in T^d$, $\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct})] = 1/(\#T)^d$ and so,
$$\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] \geq \frac{1}{\#T^d} \times (1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2)).$$
This lower bounds the interpolation probabilities of F. Now, applying Theorem 1, we have
$$\mathbf{Adv}_F(d, \sigma) \leq \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2).$$
2. If ρ is a uniform random permutation, then for any $\mathbf{y} \in \chi_d(T)$, $\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct})] = 1/p(\#T, d) \geq 1/(\#T)^d$ and so,
$$\Pr[\Gamma_{F,d}(\mathbf{x}) = \mathbf{y}] \geq \frac{1}{p(\#T, d)} \times (1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2)).$$
So, F is $(d, \sigma, (1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2)))$-interpolating on $\chi_d(T)$. Again, applying Theorem 1, we have
$$\mathbf{Adv}_F(d, \sigma) \leq \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2) + \frac{d(d-1)}{2\#T}.$$

This completes the proof of the result. □

A simpler variant of Theorem 2 is given by the following result. The difference to Theorem 2 is the condition on collision resistance. In this case, collision resistance does not depend on the length function λ.

Theorem 3. *Let $\pi : T \to T$ be a uniform random permutation and $F = \pi \circ F_1$ be a map from S to T satisfying Definition 6. Suppose that F_1 is ε-CR and it is $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ. Then for positive integers d and $\sigma \geq d$*

$$\mathbf{Adv}_F(d, \sigma) \leq \frac{d(d-1)\varepsilon}{2} + \sigma(\varepsilon_1 + 2d\varepsilon_2) + \frac{d(d-1)}{2\#T}.$$

In certain cases it is not possible to bound Self-Disjoint and Pairwise-Disjoint as required by Definition 8. The following result states another useful case when Self-Disjoint and Pairwise-Disjoint are differently bound.

Theorem 4. *Let $\rho : U \to T$ be a uniform random function and $F = \rho \circ F_1$ be a map from S to T satisfying Definition 6 and the following conditions hold.*

1. *F_1 is $\frac{1}{\#T}$-CR with respect to λ.*
2. *For any $x \in S$, $\Pr\left[\overline{\mathsf{Self\text{-}Disjoint}(x)}\right] \leq \dfrac{\lambda(x)(\lambda(x)-1)}{2\#T}$.*
3. *For any two distinct $x, x' \in S$, with $\lambda(x) = m$ and $\lambda(x') = m'$,*

$$\Pr\left[\overline{\mathsf{Pairwise\text{-}Disjoint}(x,x')}\right] \leq \frac{1}{\#T} \times \left(\sum_{k=1}^{m} \min(k,m') + \sum_{k=1}^{m'} \min(k,m)\right).$$

Then for positive integers d and $\sigma \geq d$

$$\mathbf{Adv}_F(d,\sigma) \leq \left(1 - \frac{\sigma(2d+5\sigma)}{2\#T}\right).$$

Proof: First note that

$$\sum_{k=1}^{m} \min(k,m') + \sum_{k=1}^{m'} \min(k,m) \leq 2 \sum_{k=1}^{\max(m,m')} \min(k, \min(m,m')). \tag{21}$$

The basic structure of the proof is the same as that of Theorem 2. As before, let $\mathbf{x} \in S^d$ with $\lambda(\mathbf{x}) \leq \sigma$. The events $\mathsf{Distinct}(\mathbf{x})$ and $\mathsf{Disjoint}(\mathbf{x})$ are defined as in the proof of Theorem 2. Since F_1 is $1/\#T$-CR with respect to λ, we have as before $\Pr\left[\overline{\mathsf{Distinct}(\mathbf{x})}\right] \leq d\sigma/\#T$. The relation between Disjoint and Self-Disjoint and Pairwise-Disjoint given by (17) holds in general for any domain extender and hence also holds in the present case. This gives

$$\begin{aligned}
\Pr\left[\overline{\mathsf{Disjoint}(\mathbf{x})}\right] &\leq \sum_{i=1}^{d} \Pr\left[\overline{\mathsf{Self\text{-}Disjoint}(x_i)}\right] + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \Pr\left[\overline{\mathsf{Pairwise\text{-}Disjoint}(x_i,x_j)}\right] && (22)\\
&\leq \frac{1}{\#T}\left(\sum_{i=1}^{d} \frac{\lambda(x_i)(\lambda(x_i)-1)}{2} + 2 \times \sum_{i=1}^{d} \sum_{j=i+1}^{d} \sum_{k=1}^{\max(m,m')} \min(k, \min(m,m'))\right) && (23)\\
&\leq \frac{1}{\#T}\left(\frac{1}{2}\left(\sum_{i=1}^{d} \lambda(x_i)\right)^2 + 2\sigma^2\right) && (24)\\
&\leq \frac{5\sigma^2}{2\#T}. && (25)
\end{aligned}$$

The last but one line follows from Lemma 2. This lower bounds the probability of $\mathsf{Distinct}(\mathbf{x}) \wedge \mathsf{Disjoint}(\mathbf{x})$ to be $\left(1 - \frac{\sigma(2d+5\sigma)}{2\#T}\right)$. Applying Theorem 1 now gives the required result. □

The advantage of Theorems 2, 3 and 4 is that they reduce the problem of upper bounding the PRF-advantage for F to computing certain probabilities. These can be done using purely combinatorial/probabilistic methods. Previous works have identified similar tasks for specific functions, e.g., CBC-MAC. To the best of our knowledge, the generality with which we have worked has not been done earlier.

4 Sequential Constructions

In this section, we describe CBC-MAC and several variants of it. The proof for CBC-MAC is new and considerably simpler than the earlier proof [2]. NIST has standardized a variant of CBC-MAC under the name CMAC. We provide a new security proof and an improved security bound for CMAC. Another variant of CBC-MAC is described which improves upon CMAC by reducing one invocation of the underlying permutation.

CBC-MAC uses a permutation. We describe schemes which can use a compressing function from ℓ bits to n bits (with $\ell > n$). These bounds have a weaker security bound compared to CBC-MAC, although the bound itself is good enough for practical purposes. The advantage of using a compressing function is that a lesser number of invocations of the function is required which may lead to efficiency improvements. A suitable compressing function which has been suggested in [4] is surf [3].

4.1 CBC-MAC

Let $T = \{0,1\}^n$ and π be a uniform random permutation of T. Let $\mathcal{D}$ be the maximal subset of $\cup_{i\geq 1} T^i$ with the prefix property, i.e., for any two strings in $\mathcal{D}$, one is not a prefix of the other and $\mathcal{D}$ is the largest subset of $\cup_{i\geq 1} T^i$ with this property. We define a function $\mathsf{CBC\text{-}MAC}_\pi : \mathcal{D} \to T$ as follows:

$$\mathsf{CBC\text{-}MAC}_\pi : (P_1, \ldots, P_m) \mapsto C_m$$

where $C_i = \pi(D_i)$ for $1 \leq i \leq m$ and

$$\left.\begin{array}{l} D_1 = P_1; \\ D_i = C_{i-1} \oplus P_i = \pi(D_{i-1}) \oplus P_i \ \text{ for } 2 \leq i \leq m. \end{array}\right\} \tag{26}$$

When π is clear from the context, we will simply write $\mathsf{CBC\text{-}MAC}$ instead of $\mathsf{CBC\text{-}MAC}_\pi$. With the above notation, we define another function $\mathsf{CBC\text{-}HASH} : \mathcal{D} \to T$ to be

$$\mathsf{CBC\text{-}HASH} : (P_1, \ldots, P_m) \mapsto D_m.$$

Then $\mathsf{CBC\text{-}MAC}(P_1, \ldots, P_{m+1}) = \pi(\mathsf{CBC\text{-}HASH}(P_1, \ldots, P_{m+1}))$ and consequently, $\mathsf{CBC\text{-}MAC}$ is domain extender for π in the sense of Definition 6, where $\mathsf{CBC\text{-}MAC}$ is F and $\mathsf{CBC\text{-}HASH}$ is F_1. The canonical length function λ on T^{m+1} is $\lambda(x) = m+1$ for all $x \in T^{m+1}$. This represents the fact that π is applied $(m+1)$ (resp. m) times in the computation of $\mathsf{CBC\text{-}MAC}$ (resp. $\mathsf{CBC\text{-}HASH}$). Figure 1 shows an example of CBC-MAC computation on a 5-block message.

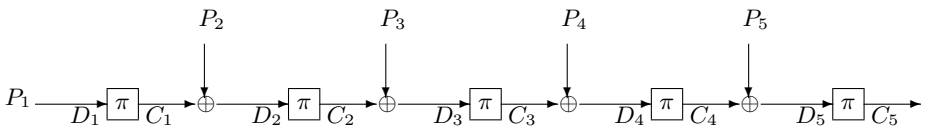

Fig. 1. CBC-MAC.

A Notation. For a proposition ϕ, we define $[[\phi]]$ to be 1 if ϕ is true and 0 if ϕ is false. For example, $[[P_1 = P_2]]$ is 1 if P_1 equals P_2, otherwise it is 0. In computations, we will use $[[\phi]]$ as an integer. It is clear from the definition that $[[\phi_1]] \times [[\phi_2]] = [[\phi_1 \wedge \phi_2]]$.

We wish to upper bound the advantage of CBC-MAC. For this, it is sufficient to bound the probabilities of Coll, Self-Disjoint and Pairwise-Disjoint. We need to consider two unequal messages x and x'. In the following, quantities related to x and x' will be denoted by unprimed and primed variables. The results below are stated in terms of the notation used for definining CBC-MAC.

Lemma 5. *Let $i, j > 1$. Then $\Pr[D_i = D'_1] = 1/\#T$ and $\Pr[D'_j = D_1] = 1/\#T$.*

Proof: By construction $D'_1 = P'_1$ and $D_i = \pi(D_{i-1}) \oplus P_i$ and so $D'_1 = D_i$ holds if and only if $\pi(D_{i-1}) = P'_1 \oplus P_i$. Since π is a uniform random permutation, the output of π on any point is uniformly distributed over T. Consequently, the probability that $\pi(D_{i-1})$ equals the fixed value $P'_1 \oplus P_i$ is $1/\#T$. The proof of the second equality is similar. □

Lemma 6. *Let $i, j > 1$. Then*

1. $\Pr[D'_j = D_i | D'_{j-1} = D_{i-1}] = [[P'_j = P_i]]$.
2. $\Pr[D'_j = D_i | D'_{j-1} \neq D_{i-1}] \leq [[P'_j \neq P_i]]/(\#T - 1)$.

Proof: By definition, $D'_j = \pi(D'_{j-1}) \oplus P'_j$ and $D_i = \pi(D_{i-1}) \oplus P_i$, where $C'_{j-1} = \pi(D'_{j-1})$ and $C_{i-1} = \pi(D_{i-1})$. This gives

$$D'_j \oplus D_i = (\pi(D'_{j-1}) \oplus \pi(D_{i-1})) \oplus (P'_j \oplus P_i) = (C'_{j-1} \oplus C_{i-1}) \oplus (P'_j \oplus P_i).$$

The first statement is easy to see. Given that $D'_{j-1} = D_{i-1}$, we have $D'_j = D_i$ if and only if $P'_j = P_i$.

For the second statement we are given that $D'_{j-1} \neq D_{i-1}$. Using the fact that π is a permutation, we have $\pi(D'_{j-1}) \neq \pi(D_{i-1})$. So, if $P'_j = P_i$, then $D'_j \neq D_i$. Hence, $P'_j \neq P_i$ is a necessary condition for $D'_j = D_i$ (conditioned on the event $D'_{j-1} \neq D_{i-1}$).

Suppose $P'_j \neq P_i$. Since D'_{j-1} and D_{i-1} are distinct values and π is a uniform random permutation, the pair (C'_{j-1}, C_{i-1}) is uniformly distributed over $\chi_2(T)$ and takes any value of $\chi_2(T)$ with probability $1/(\#T(\#T-1))$. Consequently, for any fixed P'_j and P_i (with $P'_j \neq P_i$), there are exactly $\#T$ values that the pair (C'_{j-1}, C_{i-1}) can take satisfying the condition $C'_{j-1} \oplus C_{i-1} = P_i \oplus P'_j$. From this the result follows. □

Lemma 7. *Let x and x' be elements of $\mathcal{D}$ with $\lambda(x) = m$ and $\lambda(x') = m'$. Then for $1 \leq i \leq m$,*

$$\Pr[D'_{m'} = D_i] \leq 1/(\#T - 1). \tag{27}$$

Consequently, for $\varepsilon = \varepsilon_2 = 1/(\#T - 1)$

1. CBC-HASH *is ε-CR;*
2. CBC-HASH *satisfies* $\Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] \leq \varepsilon_2(\lambda(x) + \lambda(x'))$.

Proof: Assume without loss of generality that $m \geq m'$. If $m = m' = 1$, then x and x' are distinct single block messages, so that $D_m = P_1 \neq P'_1 = D'_{m'}$. Then $\Pr[D'_{m'} = D_m] = 0$. If $m' = 1$ and $m > 1$, then using Lemma 5, for $1 < i \leq m$, $\Pr[D'_{m'} = D_i] = 1/\#T < 1/(\#T - 1)$. So, we can assume that both $m, m' > 1$.

Case: $1 < m' \leq i \leq m$**.**

$$\begin{aligned}
\Pr[D'_{m'} = D_i] &= \Pr[(D'_{m'} = D_i) \wedge ((D'_{m'-1} = D_{i-1}) \vee (D'_{m'-1} \neq D_{i-1}))] \\
&= \Pr[(D'_{m'} = D_i) | (D'_{m'-1} = D_{i-1})] \Pr[D'_{m'-1} = D_{i-1}] \\
&\quad + \Pr[(D'_{m'} = D_i) | (D'_{m'-1} \neq D_{i-1})] \Pr[D'_{m'-1} \neq D_{i-1}] \\
&\leq [[P'_{m'} = P_i]] \Pr[D'_{m'-1} = D_{i-1}] + \Pr[(D'_{m'} = D_i) | (D'_{m'-1} \neq D_{i-1})] \\
&\leq \frac{[[P'_{m'} \neq P_i]]}{\#T - 1} + [[P'_{m'} = P_i]] \Pr[D'_{m'-1} = D_{i-1}]. \qquad (28)
\end{aligned}$$

We have used Lemma 6 in the last two steps of the derivation. Let ϕ_j be true if $P'_{m'-j} = P_{i-j}$ and false otherwise. Applying (28) inductively, we obtain

$$\begin{aligned}
\Pr[D'_{m'} = D_i] &\leq \frac{[[\overline{\phi_0}]]}{\#T - 1} + [[\phi_0]] \times \left(\frac{[[\overline{\phi_1}]]}{\#T - 1} + [[\phi_1]] \Pr[D'_{m'-2} = D_{i-2}] \right) \\
&= \frac{1}{\#T - 1} \times \left([[\overline{\phi_0}]] + [[\phi_0 \wedge \overline{\phi_1}]] \right) + [[\phi_0 \wedge \phi_1]] \Pr[D'_{m'-2} = D_{i-2}] \\
&\quad \cdot \ \cdots \\
&\leq \frac{1}{\#T - 1} \times A + B \times \Pr[D'_1 = D_{i-m'+1}]
\end{aligned}$$

where

$$\begin{aligned}
A &= [[\overline{\phi_0}]] + [[\phi_0 \wedge \overline{\phi_1}]] + [[\phi_0 \wedge \phi_1 \wedge \overline{\phi_2}]] + \cdots + [[\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \overline{\phi_{m'-2}}]]; \\
B &= [[\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \phi_{m'-2}]].
\end{aligned}$$

Note that $A + B = 1$, since exactly one of the terms of $A + B$ is one and all others are zero.

Subcase $i > m'$**:** From Lemma 5, $\Pr[D'_1 = D_{i-m'+1}] = 1/\#T \leq 1/(\#T - 1)$. So $\Pr[D'_{m'} = D_i] \leq (A + B)/(\#T - 1) = 1/(\#T - 1)$.

Subcase $i = m'$**:** So,

$$B \times \Pr[D'_1 = D_{i-m'+1}] = [[\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_{m'-2} \wedge \phi_{m'-1}]]. \qquad (29)$$

The right hand side of (29) is 1 if and only if all the ϕ_js are true, i.e., if and only if $P'_j = P_j$ for $1 \leq j \leq m'$, which means x' is a prefix of x. But, since $x, x' \in \mathcal{D}$, x' is not a prefix of x and so the right hand side is 0. Consequently, $\Pr[D'_{m'} = D_i] \leq A/(\#T - 1) \leq 1/(\#T - 1)$.

This completes the proof when $m \geq i \geq m' > 1$.

Case $1 < i < m'$**:** This case can be treated by a similar analysis to show that in this case also $\Pr[D'_{m'} = D_i] \leq 1/(\#T - 1)$ holds.

The case $i = m$ shows the statement on ε-CR of CBC-HASH. In the definition of Pairwise-Disjoint given in Definition 7, the quantities $\mathcal{Z}$ and $\mathcal{Z}'$ correspond to D_m and $D'_{m'}$ respectively; while the quantities $\mathcal{U}_1, \ldots, \mathcal{U}_{m-1}$ and $\mathcal{U}'_1, \ldots, \mathcal{U}'_{m-1}$ correspond to the quantities $D'_1, \ldots, D'_{m-1}$ and $D_1, \ldots, D_{m-1}$ respectively. With this correspondence, we have

$$\Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] = \Pr\left[\bigvee_{i=1}^{m-1} (D'_{m'} = D_i) \vee \bigvee_{i=1}^{m'-1} (D_m = D'_i) \right]$$

$$\leq \sum_{i=1}^{m-1} \Pr[D'_{m'} = D_i] + \sum_{i=1}^{m'-1} \Pr[D_m = D'_i].$$
$$\leq (m + m')/(\#T - 1)$$
$$< \frac{\lambda(x) + \lambda(x')}{\#T - 1}.$$

□

The following two results can be proved in manner similar to that of Lemmas 5 and 6.

Lemma 8. *Let $i > 1$. Then $\Pr[D_i = D_1] = 1/\#T$.*

Lemma 9. *Let $j > i > 1$. Then*

1. $\Pr[D_j = D_i | D_{j-1} = D_{i-1}] = [[P_j = P_i]]$.
2. $\Pr[D_j = D_i | D_{j-1} \neq D_{i-1}] \leq [[P_j \neq P_i]]/(\#T - 1)$.

The following result can be proved in a manner similar to that of Lemma 7.

Lemma 10. *Let $x \in \mathcal{D}$ with $\lambda(x) = m$. Then for $1 \leq i < m$, $\Pr[D_m = D_i] \leq 1/(\#T - 1)$. Consequently, for $\varepsilon_1 = 1/(\#T - 1)$ $\mathsf{CBC\text{-}HASH}$ satisfies $\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \leq \varepsilon_1 \lambda(x)$.*

Proof: We proceed with the first few steps as in the proof of Lemma 7.

$$\begin{aligned}\Pr[D_m = D_i] &= \Pr[(D_m = D_i) | (D_{m-1} = D_{i-1})] \Pr[D_{m-1} = D_{i-1}] \\ &\quad + \Pr[(D_m = D_i) | (D_{m-1} \neq D_{i-1})] \Pr[D_{m-1} \neq D_{i-1}] \\ &\leq [[P_m = P_i]] \times \Pr[D_{m-1} = D_{i-1}] + \frac{[[P_m \neq P_i]]}{\#T - 1}.\end{aligned}$$

Now an analysis similar to that for Lemma 7 proves the result. □

The advantage of $\mathsf{CBC\text{-}MAC}$ is given by the following result.

Theorem 5. *Let d and $\sigma \geq d$ be positive integers. Then*

$$\mathbf{Adv}_{\mathrm{CBC\text{-}MAC}}(d, \sigma) \leq \frac{d(d-1) + \sigma(2d+1)}{\#T - 1}.$$

Proof: Using Lemmas 7 and 10, for the function $\mathsf{CBC\text{-}HASH}$ we have the following.

1. $\mathsf{CBC\text{-}HASH}$ is ε-CR where $\varepsilon = 1/(\#T - 1)$.
2. $\mathsf{CBC\text{-}HASH}$ is $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ, where $\varepsilon_1 = \varepsilon_2 = 1/(\#T - 1)$.

Using Theorem 3, we obtain

$$\begin{aligned}\mathbf{Adv}_{\mathrm{CBC\text{-}MAC}}(d, \sigma) &\leq \frac{d(d-1)}{2(\#T - 1)} + \frac{\sigma(1 + 2d)}{\#T - 1} + \frac{d(d-1)}{2\#T} \\ &\leq \frac{1}{(\#T - 1)} \left(d(d-1) + \sigma(2d+1)\right).\end{aligned}$$

□

4.2 A Variant of CBC-MAC

Fix positive integers ℓ and n with $\ell \geq n$ and let $U = \{0,1\}^\ell$ and $T = \{0,1\}^n$. Let $\rho : U \to T$ be a uniform random function. The natural additive operation on equal length binary strings is $\oplus$. If x and y are unequal length binary strings, we define $x \oplus\!\!\!\rightarrow y$ to be the binary string obtained by XORing the shorter string into the least significant bits of the longer string. By $\mathsf{bot}(P)$ we will mean the n least significant bits of P; by $\mathsf{top}(P)$ we will mean the $(\ell - n)$ most significant bits of P.

We describe a method similar to CBC-MAC. Everything is similar except that in this case ρ is used instead of π. In fact, if $\ell = n$, then the description of VCBC-MAC can be obtained from that of CBC-MAC by replacing π with ρ. If $\ell > n$, then we also need to replace $\oplus$ with $\oplus\!\!\!\rightarrow$. For completeness we briefly describe the method. Let $\mathcal{D}$ be the maximal subset of $\cup_{i\geq 1} U^i$ with the prefix property, i.e., for any two strings in $\mathcal{D}$, one is not a prefix of the other and $\mathcal{D}$ is the largest subset of $\cup_{i\geq 1} U^i$ with this property. We define a function $\mathsf{VCBC\text{-}MAC}_\rho : \mathcal{D} \to T$ as follows:

$$\mathsf{VCBC\text{-}MAC}_\rho : (P_1, \ldots, P_m) \mapsto C_m$$

where $C_i = \rho(D_i)$ for $1 \leq i \leq m$ and

$$\left.\begin{array}{l} D_1 = P_1; \\ D_i = \rho(D_{i-1}) \oplus\!\!\!\rightarrow P_i \ \text{ for } 2 \leq i \leq m. \end{array}\right\} \tag{30}$$

The function VCBC-HASH is defined in a manner similar to CBC-HASH and has output $D_m \in U$. Let as before x and x' be unequal messages from $\mathcal{D}$. In the following, quantities related to x and x' will be denoted by unprimed and primed variables.

Lemma 11. *Let $i, j > 1$. Then* $\Pr[D_i = D'_1] = [[\mathsf{top}(P_i) = \mathsf{top}(P'_1)]]/\#T$ *and* $\Pr[D'_j = D_1] = [[\mathsf{top}(P'_i) = \mathsf{top}(P_1)]]/\#T$.

Proof: By construction $D'_1 = P'_1$ and $D_i = \rho(D_{i-1}) \oplus\!\!\!\rightarrow P_i$ and so $D'_1 = D_i$ holds if and only if $\mathsf{top}(P_i) = \mathsf{top}(P'_1)$ and $\rho(D_{i-1}) = \mathsf{bot}(P'_1 \oplus P_i)$. Since ρ is a uniform random function, the output of ρ on any input is uniformly distributed over T. Consequently, the probability that $\rho(D_{i-1})$ equals the fixed value $\mathsf{bot}(P'_1 \oplus P_i)$ is $1/\#T$. The proof of the second equality is similar. □

Lemma 12. *Let $i, j > 1$. Then*

1. $\Pr[D'_j = D_i | D'_{j-1} = D_{i-1}] = [[P'_j = P_i]]$.
2. $\Pr[D'_j = D_i | D'_{j-1} \neq D_{i-1}] \leq [[\mathsf{top}(P'_j) = \mathsf{top}(P_i)]]/\#T$.

Proof: By definition, $D'_j = \rho(D'_{j-1}) \oplus\!\!\!\rightarrow P'_j$ and $D_i = \rho(D_{i-1}) \oplus\!\!\!\rightarrow P_i$, where $C'_{j-1} = \rho(D'_{j-1})$ and $C_{i-1} = \rho(D_{i-1})$. This gives

$$D'_j \oplus D_i = (\rho(D'_{j-1}) \oplus \rho(D_{i-1})) \oplus\!\!\!\rightarrow (P'_j \oplus P_i) = (C'_{j-1} \oplus C_{i-1}) \oplus\!\!\!\rightarrow (P'_j \oplus P_i).$$

The first statement is easy. Given that $D'_{j-1} = D_{i-1}$, we have $D'_j = D_i$ if and only if $P'_j = P_i$.

For the second statement, it is given that $D'_{j-1} \neq D_{i-1}$. Since $D'_{j-1} \neq D_{i-1}$ and ρ is a uniform random function, $C'_{j-1} \oplus C_{i-1}$ is uniformly distributed over T and so with probability $1/\#T$ it equals $\mathsf{bot}(P'_j \oplus P_i)$. Further, for D'_j to be equal to D_i we must also have $\mathsf{top}(P'_j) = \mathsf{top}(P_i)$. This gives the result. □

Lemma 13. *Let x and x' be distinct elements of $\mathcal{D}$ with $\lambda(x) = m$ and $\lambda(x') = m'$. Then for $1 \leq i \leq m$, $\Pr[D'_{m'} = D_i] \leq \min(i, m')/\#T$. Consequently*

1. VCBC-HASH *is $\frac{1}{\#T}$-CR with respect to λ.*
2. VCBC-HASH *satisfies* $\Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] \leq \sum_{i=1}^{m} \min(i, m') + \sum_{i=1}^{m'} \min(i, m)$.

Proof: The cases when either m or m' equals 1 is tackled as in the proof of Lemma 7. So, we assume that both $m, m' > 1$ and without loss of generality $m \geq m'$. Again, as in the proof of Lemma 7, we consider two cases: $1 < m' \leq i \leq m$ and $1 < i < m' \leq m$. We provide the proof for the first case, the proof for the second case being similar.

$$\begin{aligned}
\Pr[D'_{m'} = D_i] &= \Pr[(D'_{m'} = D_i) \wedge ((D'_{m'-1} = D_{i-1}) \vee (D'_{m'-1} \neq D_{i-1}))] \\
&= \Pr[(D'_{m'} = D_i) | (D'_{m'-1} = D_{i-1})] \Pr[D'_{m'-1} = D_{i-1}] \\
&\quad + \Pr[(D'_{m'} = D_i) | (D'_{m'-1} \neq D_{i-1})] \Pr[D'_{m'-1} \neq D_{i-1}] \\
&\leq [[P'_{m'} = P_i]] \Pr[D'_{m'-1} = D_{i-1}] + \Pr[(D'_{m'} = D_i) | (D'_{m'-1} \neq D_{i-1})] \\
&\leq \frac{[[\mathsf{top}(P'_{m'}) = \mathsf{top}(P_i)]]}{\#T} + [[P'_{m'} = P_i]] \Pr[D'_{m'-1} = D_{i-1}]. \qquad (31)
\end{aligned}$$

Lemma 12 is used in the last two steps of this derivation. Let ϕ_j be true if $P'_{m'-j} = P_{i-j}$ and false otherwise. Also, let μ_j be true if $\mathsf{top}(P'_{m'-j}) = \mathsf{top}(P_{i-j})$ and false otherwise. Thus, ϕ_j being true implies μ_j being true, but, not conversely. Applying (31) inductively, we obtain

$$\begin{aligned}
\Pr[D'_{m'} = D_i] &\leq \frac{[[\mu_0]]}{\#T} + [[\phi_0]] \times \left(\frac{[[\mu_1]]}{\#T} + [[\phi_1]] \Pr[D'_{m'-2} = D_{i-2}] \right) \\
&= \frac{1}{\#T} \times ([[\mu_0]] + [[\phi_0 \wedge \mu_1]]) + [[\phi_0 \wedge \phi_1]] \Pr[D'_{m'-2} = D_{i-2}] \\
&\quad \cdot \ \cdots \\
&\leq \frac{1}{\#T} \times A + B \times \Pr[D'_1 = D_{i-m'+1}]
\end{aligned}$$

where

$$\begin{aligned}
A &= [[\mu_0]] + [[\phi_0 \wedge \mu_1]] + [[\phi_0 \wedge \phi_1 \wedge \mu_2]] + \cdots + [[\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \mu_{m'-2}]]; \\
B &= [[\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \phi_{m'-2}]].
\end{aligned}$$

In contrast to the proof of Lemma 7, in this case we have $A + B \leq m'$.

Subcase $i > m'$**:** From Lemma 11, $\Pr[D'_1 = D_{i-m'+1}] = [[\mu_{m'-1}]]/\#T$. So, $\Pr[D'_{m'} = D_i] \leq (A + B)/\#T = m'/\#T$.

Subcase $i = m'$**:** So,

$$B \times \Pr[D'_1 = D_{i-m'+1}] = [[\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_{m'-2} \wedge \phi_{m'-1}]].$$

The right hand side is 1 if all the ϕ_js are true, i.e., if and only if $P'_j = P_j$ for $1 \leq j \leq m'$, which means x' is a prefix of x. But, since $x, x' \in \mathcal{D}$, x' is not a prefix of x and so the right hand side is 0. Consequently, $\Pr[D'_{m'} = D_i] \leq A/\#T \leq m'/\#T$.

□

The following two results can be proved in manner similar to that of Lemma 11 and 12.

Lemma 14. *Let $i > 1$. Then* $\Pr[D_i = D_1] = [[\mathsf{top}(P_i) = \mathsf{top}(P_1)]]/\#T$.

Lemma 15. *Let $j > i > 1$. Then*

1. $\Pr[D_j = D_i | D_{j-1} = D_{i-1}] = [[P_j = P_i]]$.
2. $\Pr[D_j = D_i | D_{j-1} \neq D_{i-1}] \leq [[\mathsf{top}(P_j) = \mathsf{top}(P_i)]]/\#T$.

The following result can be proved in a manner similar to that of Lemma 13.

Lemma 16. *Let $x \in \mathcal{D}$ with $\lambda(x) = m$. Then for $1 \leq i < m$,* $\Pr[D_m = D_i] \leq i/\#T$. *Consequently,* VCBC-HASH *satisfies* $\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \leq m(m-1)/2$.

The advantage of VCBC-MAC can now be obtained as in the case of CBC-MAC. The only difference is that we will use Theorem 4 instead of Theorem 3.

Theorem 6. *Let d and $\sigma \geq d$ be positive integers. Then*

$$\mathbf{Adv}_{\text{VCBC-MAC}}(d, \sigma) \leq \frac{\sigma(2d + 5\sigma)}{2\#T}.$$

This bound is weaker than the bound for Theorem 5. The reason is that the bounds for collision probability and disjointness for VCBC-MAC given by Lemmas 13 and 16 are weaker than those for CBC-MAC given by Lemmas 7 and 10.

4.3 Handling Arbitrary Messages

There are two basic requirements to be achieved using any variant of CBC-MAC which handles arbitrary and variable length messages.

1. Ensure that prefixes do not cause problems.
2. Distinguish between the cases where the last block is full or partial.

Both CBC-MAC and VCBC-MAC handles messages which consists of "full" blocks from a domain satisfying the prefix property. There is an easy extension to get rid of the prefix property restriction and also properly tackle the last block. This is done by a unique encoding of a message into a prefix-free set. For CBC-MAC we use a permutation $\pi : T \to T$ and for VCBC-MAC we use a function from $\rho : U \to T$, where $U = \{0,1\}^\ell$ and $T = \{0,1\}^n$ with $\ell \geq n$. The encoding does not differentiate between the permutation π and the function ρ.

In Figure 2, we define a formatting function which takes as input a binary string x of length $\mathsf{len}(x) \geq 0$; a positive integer l; and returns as output $P_1, \ldots, P_m$ for some $m \geq 1$ and where each P_i is an l-bit string. Further, the function $\mathsf{Format}(x, l)$ also defines an integer r with $1 \leq r \leq l$. If $r = l$, then this denotes that l divides $\mathsf{len}(x)$, and if $1 \leq r < l$, then this denotes that l does not divide $\mathsf{len}(x)$ and that the last block P_m has been padded to length l.

Let w $(2 \leq w < \ell)$ be a parameter to be specified later. Let σ_0, σ_1 and σ_2 be three distinct binary strings of length w each. Given x, let $(P_1, \ldots, P_m)$ be the output of $\mathsf{Format}(x, \ell - w)$. We define $\mathsf{encode}(x)$ to be the following map.

$$\mathsf{encode} : x \mapsto (P_1 || \sigma_0, \ldots, P_{m-1} || \sigma_0, P_m || \sigma_b) \tag{32}$$

$\mathsf{Format}(x, l)$.
1. Write $\mathsf{len}(x) = (m-1)l + r$, where $1 \leq r \leq l$.
2. If $r < l$, then set $\mathsf{pad}(x) = x \| 10^{l-r-1}$.
3. Else set $\mathsf{pad}(x) = x$.
4. Format $\mathsf{pad}(x)$ into m blocks $P_1, \ldots, P_m$ each of length l.
Return $(P_1, \ldots, P_m)$.

Fig. 2. Padding and formatting of arbitrary length strings. This also defines the values of m and r from $\mathsf{len}(x)$ and l.

where $b = 1$ if the last block is a full block and $b = 2$ if the last block is a partial block. Let $\mathcal{S}$ be the set of all binary strings and $\mathcal{D} = \mathsf{encode}(\mathcal{S}) = \{\mathsf{encode}(x) : x \in \mathcal{S}\}$. We have the following result which is easy to see from the definition of encode.

Proposition 4. *The map* encode *is an injective function from* $\mathcal{S}$ *to* $\mathcal{D}$*. Further,* $\mathcal{D}$ *satisfies the prefix property.*

Given this encoding function, we define the extensions of CBC-MAC and VCBC-MAC to handle strings from the set $\mathcal{S}$ in the following manner.

$$\mathsf{CBC\text{-}MAC}^*(x) = \mathsf{CBC\text{-}MAC}(\mathsf{encode}(x)); \tag{33}$$

$$\mathsf{VCBC\text{-}MAC}^*(x) = \mathsf{VCBC\text{-}MAC}(\mathsf{encode}(x)). \tag{34}$$

Since CBC-MAC and VCBC-MAC are applied to a set of strings satisfying the prefix property, the security of CBC-MAC* and VCBC-MAC* is immediate from the security of CBC-MAC and VCBC-MAC.

For $w = 2$ we can use $\sigma_0 = 00$, $\sigma_1 = 01$ and $\sigma_2 = 10$. Due to the reduction in the block length from ℓ to $\ell - w$, there is a loss in efficiency. We form an approximate estimate of this efficiency loss. With ℓ-bit blocks the number of invocations of ρ would be around $\mathsf{len}(x)/\ell$, whereas with $(\ell - w)$-bit blocks the number of invocations is around $\mathsf{len}(x)/(\ell - w)$. The ratio of increase is approximately $\ell/(\ell - w)$. If we choose w to be 2, then the proportionate increase is not significant. But, with $w = 2$, formatting the message can cause problems due to non-alignment with byte boundaries. With $w = 8$ this problem will go away but the proportionate increase in the number of invocations will be $\ell/(\ell - 8)$. A typical value of ℓ is 128. For example, if $\mathsf{len}(x) = 2^{13}$ (1024 bytes), then with $w = 2$, the number of invocations required will be 66 and for $w = 2$, the number of invocations required will be 69. In comparison, CMAC (see Section 4.4 for a description) will require 64 invocations if the encryption of 0^ℓ is precomputed and will require 65 invocations otherwise. Given the simplicity of the construction, this may be a tolerable loss of efficiency.

4.4 CMAC

The NIST of USA has standardized a variant of CBC-MAC which can tackle arbitrary messages [10]. Let $T = \{0,1\}^n$ and π be a uniform random permutation of T. As in Section 2.3, assume that $T = GF(2^n)$ under the usual identification of n-bit strings and polynomials over $GF(2)$ of degree less than n. Let $\psi : T \to T$ be the map $\psi(\alpha) = u\alpha \bmod \tau(u)$; as mentioned in Section 2.3 this map satisfies Definition 5. Let $L = \pi(0^n)$ and let $L_1 = uL \bmod \tau(u)$ and $L_2 = u^{-1}L \bmod \tau(u)$. (Note that even though CMAC uses a specific form for ψ, it would work with any ψ satisfying Definition 5.)

The construction handles arbitrary length strings. Let x be a string of length $\mathsf{len}(x) \geq 0$ and let $(P_1, \ldots, P_m)$ be the output of $\mathsf{Format}(x, n)$ given in Figure 2 which also defines the values of m and r. Then CMAC is defined to be

$$\mathsf{CMAC} : x \mapsto \mathsf{CBC\text{-}MAC}(P_1, \ldots, P_{m-1}, P_m \oplus L_b) \tag{35}$$

where $b = 1$ if the last block is full, i.e., if $r = n$ and $b = 2$ if the last block is partial, i.e., if $1 \leq r < n$. Similarly, define $\mathsf{CMAC\text{-}HASH}$ to be

$$\mathsf{CMAC\text{-}HASH} : x \mapsto \mathsf{CBC\text{-}HASH}(P_1, \ldots, P_{m-1}, P_m \oplus L_b) \tag{36}$$

Figure 3 shows an example of processing a 5-block message with CMAC.

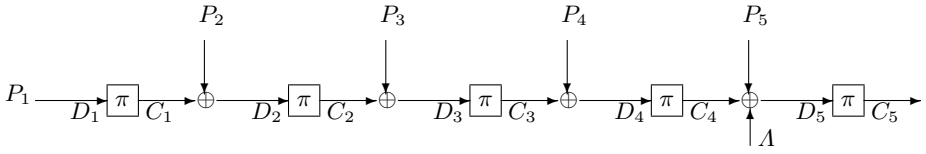

Fig. 3. CMAC. Here $\Lambda = L_1$ if the last block is full and $\Lambda = L_2$ if the last block is partial.

Note that to process m blocks, CMAC requires $m + 1$ invocations of π, with m invocations to process the m blocks and one initial invocation to obtain L. As a result, if x is formatted into m n-bit blocks, from the definition of λ in Definition 6, we have $\lambda(x) = m + 1$.

To prove the security of CMAC, we need to obtain bounds on collision probabilities and also the probabilities of disjointness on $\mathsf{CMAC\text{-}HASH}$.

Lemma 17. *Let $x \neq x'$ and $\lambda(x) = m + 1$, $\lambda(x') = m' + 1$. Then for $1 \leq i \leq m$, $\Pr[D'_{m'} = D_i] \leq 2/(\#T - 1)$. Consequently, for $\varepsilon = 2/(\#T - 1)$ and $\varepsilon_2 = 4/(\#T - 1)$*

1. $\mathsf{CMAC\text{-}HASH}$ *is ε-CR;*
2. $\mathsf{CMAC\text{-}HASH}$ *satisfies* $\Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] \leq \varepsilon_2(\lambda(x) + \lambda(x'))$.

Proof: Assume without loss of generality that $m \geq m'$. If $i = m$, then there are several cases to consider depending on whether n divides $\mathsf{len}(x)$ and $\mathsf{len}(x')$ or not. Suppose n divides both $\mathsf{len}(x)$ and $\mathsf{len}(x')$. Then $D'_{m'} = D_m$ if and only if $\pi(D'_{m'-1}) \oplus P'_m = \pi(D_{m-1}) \oplus P_m$ and hence, this reduces to the analysis for $\mathsf{CBC\text{-}MAC}$, so that the probability for $D'_{m'} = D_m$ is at most $1/(\#T - 1)$. Similarly, if n does not divide any of $\mathsf{len}(x)$ or $\mathsf{len}(x')$, it is possible to show that the probability for $D'_{m'} = D_m$ is at most $1/(\#T - 1)$. Suppose that n divides $\mathsf{len}(x)$ but not $\mathsf{len}(x')$ (the other case is similar). In this case, $D'_{m'} = D_m$ if and only if $\pi(D'_{m'-1}) \oplus P'_m \oplus L_2 = \pi(D_{m-1}) \oplus P_m \oplus L_1$. If $D'_{m'-1} = D_{m-1}$, then $D'_{m'} = D_m$ if and only if $P'_m \oplus L_2 = P_m \oplus L_1$. From Definition 5, the last event holds with probability $1/\#T$. If $D'_{m'-1} \neq D_{m-1}$, then $D'_{m'} = D_m$ holds with probability $1/(\#T - 1)$. So, we have

$$\begin{aligned}
\Pr[D'_{m'} = D_m] &= \Pr[D'_{m'} = D_m | (D'_{m'-1} = D_{m-1})] \Pr[D'_{m'-1} = D_{m-1}] \\
&\quad + \Pr[D'_{m'} = D_m | (D'_{m'-1} \neq D_{m-1})] \Pr[D'_{m'-1} \neq D_{m-1}] \\
&\leq \frac{1}{\#T - 1} (\Pr[D'_{m'-1} = D_{m-1}] + \Pr[D'_{m'-1} \neq D_{m-1}]) \\
&= \frac{1}{\#T - 1}.
\end{aligned}$$

Now suppose $i < m$. Again there are several cases depending on whether $m' \leq i$ and/or $n|\mathsf{len}(x')$ or not. We consider the case $m' \leq i$ and $n|\mathsf{len}(x')$, the other cases being similar. We have

$$D'_{m'} = \pi(D'_{m'-1}) \oplus P'_{m'} \oplus L_1;\ D_1 = P_1 \text{ and } D_i = \pi(D_{i-1}) \oplus P_i \text{ for } i > 1.$$

If $i = 1$, then the result easily holds from the fact that $\pi(D'_{m'-1})$ is uniformly distributed over T; so we consider $i > 1$. Note that if $P'_{m'} \oplus L_1 = P_i$ and $D'_{m'-1} \neq D_{i-1}$, then $D'_{m'} \neq D_i$. On the other hand, if $P'_{m'} \oplus L_1 \neq P_i$ and $D'_{m'-1} \neq D_{i-1}$, then $D'_{m'} \neq D_i$ holds with probability at most $1/(\#T - 1)$. We compute

$$\begin{aligned}
\Pr[D'_{m'} = D_i] &= \Pr[D'_{m'} = D_i | (D'_{m'-1} = D_{i-1})] \Pr[D'_{m'-1} = D_{i-1}] \\
&\quad + \Pr[D'_{m'} = D_i | (D'_{m'-1} \neq D_{i-1})] \Pr[D'_{m'-1} \neq D_{i-1}] \\
&\leq \Pr[P'_{m'} \oplus L_1 = P_i] \Pr[D'_{m'-1} = D_{i-1}] \\
&\quad + \Pr[D'_{m'} = D_i | (D'_{m'-1} \neq D_{i-1}) \wedge (P'_{m'} \oplus L_1 = P_i)] \Pr[P'_{m'} \oplus L_1 = P_i] \\
&\quad + \Pr[D'_{m'} = D_i | (D'_{m'-1} \neq D_{i-1}) \wedge (P'_{m'} \oplus L_1 \neq P_i)] \Pr[P'_{m'} \oplus L_1 \neq P_i] \\
&\leq \Pr[P'_{m'} \oplus L_1 = P_i] \Pr[D'_{m'-1} = D_{i-1}] + \frac{\Pr[P'_{m'} \oplus L_1 \neq P_i]}{\#T - 1}.
\end{aligned}$$

The analysis of the event $D'_{m'-1} = D_{i-1}$ does not involve L_1 (or L_2) and we can proceed exactly as in the proof of Lemma 7. As before, let ϕ_j be a Boolean predicate which is true if $P'_{m-j} = P_{i-j}$ and false otherwise. We obtain

$$\Pr[D'_{m'} = D_i] \leq \frac{A}{\#T - 1} + B \times \Pr[D'_1 = D_{i-m'+1}]$$

where

$$\begin{aligned}
A &= \Pr[P'_{m'} \oplus L_1 \neq P_i] + \Pr[P'_{m'} \oplus L_1 = P_i] \times \\
&\qquad \left([[\overline{\phi_1}]] + [[\phi_1 \wedge \overline{\phi_2}]] + \cdots + [[\phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \overline{\phi_{m'-2}}]] \right) \\
B &= \Pr[P'_{m'} \oplus L_1 = P_i] \times [[\phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \phi_{m'-2}]].
\end{aligned}$$

Define

$$C = \left([[\overline{\phi_1}]] + [[\phi_1 \wedge \overline{\phi_2}]] + \cdots + [[\phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \overline{\phi_{m'-2}}]] \right) + [[\phi_1 \wedge \cdots \wedge \phi_{m'-3} \wedge \phi_{m'-2}]]$$

and note that $C = 1$. If $m' < i$, then $\Pr[D'_1 = D_{i-m'+1}] = 1/\#T < 1/(\#T - 1)$. Then

$$\begin{aligned}
\Pr[D'_{m'} = D_i] &\leq \frac{A + B}{\#T - 1} \\
&\leq \frac{1}{(\#T - 1)} \left(\Pr[P'_{m'} \oplus L_1 \neq P_i] + \Pr[P'_{m'} \oplus L_1 = P_i] \times C \right) \\
&\leq \frac{1}{\#T - 1}.
\end{aligned}$$

If $m' = i$, then $\Pr[D'_1 = D_1] = [[P'_1 = P_1]]$ and in this case it is possible to show that

$$\Pr[D'_{m'} = D_i] \leq \frac{1}{(\#T - 1)} + \frac{1}{\#T} \leq \frac{2}{\#T - 1}.$$

□

In a similar manner we obtain the following result.

Lemma 18. *Let x be a message such that $\lambda(x) = m + 1$. Then for $1 \le i < m$, $\Pr[D_m = D_i] \le 1/(\#T - 1)$.*
Consequently, for $\varepsilon_1 = 4/(\#T - 1)$, CMAC-HASH satisfies $\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \le \varepsilon_1 \lambda(x)$.

Combining Lemmas 17 and 18 with Theorem 3, we obtain the following result.

Theorem 7. *Let d and $\sigma \ge d$ be positive integers. Then*

$$\mathbf{Adv}_{\mathrm{CMAC}}(d, \sigma) \le \frac{3d(d-1) + 4\sigma(1+2d)}{\#T - 1}.$$

4.5 iCMAC: A New CBC-MAC Based Construction

Let $T = \{0,1\}^n$ and π be a uniform random permutation of T. As in Section 2.3, assume that $T = GF(2^n)$ under the usual identification of n-bit strings and polynomials over $GF(2)$ of degree less than n. Let $\psi : T \to T$ be a map satisfying Definition 5. Let fStr be a fixed element of T (CMAC uses $\mathsf{fStr} = 0^n$); and k_1, k_2 be two distinct integers in the range $1 \le k_1, k_2 \le 2^n - 2$. Typical values of k_1 and k_2 are $k_1 = 1$ and $k_2 = 2^n - 2 = -1 \bmod (2^n - 1)$. Define $L = \pi(\mathsf{fStr})$, $L_1 = \psi^{k_1}(L)$ and $L_2 = \psi^{k_2}(L_2)$.

The construction handles arbitrary length strings. Let x be a string of length $\mathsf{len}(x) \ge 0$ and let $(P_1, \ldots, P_m)$ be the output of $\mathsf{Format}(x, n)$ given in Figure 2. The values of m and r are obtained from n and $\mathsf{len}(x)$ as in Figure 2.

$$\mathsf{iCMAC} : x \mapsto C_m \tag{37}$$

where $C_i = \pi(D_i)$ for $1 \le i \le m$ and

$$D_i = \begin{cases} P_1 \oplus L_1 & \text{if } i = m = 1,\ r < n; \\ P_1 \oplus L_2 & \text{if } i = m = 1,\ r = n; \\ P_1 & \text{if } i = 1,\ m > 1; \\ \pi(D_{i-1}) \oplus P_i & \text{if } 1 < i < m,\ m > 1; \\ \psi^{k_1}(\pi(D_{m-1})) \oplus P_m & \text{if } i = m,\ m > 1, r < n; \\ \psi^{k_2}(\pi(D_{m-1})) \oplus P_m & \text{if } i = m,\ m > 1, r = n; \end{cases} \tag{38}$$

The output of iCMAC-HASH on input x is defined to be D_m. For notational convenience, we put $D_0 = \mathsf{fStr}$ so that $L_1 = \psi^{k_1}(\pi(D_0))$ and $L_2 = \psi^{k_2}(\pi(D_0))$. Figure 4 shows an example of processing a 5-block message using iCMAC.

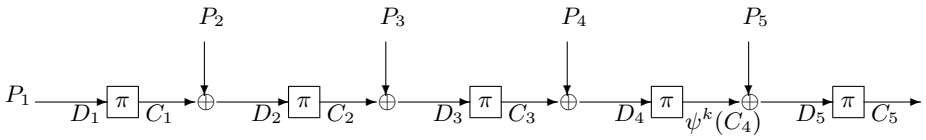

Fig. 4. iCMAC. Here k takes the value k_1 if the last block is a full block and takes the value k_2 if the last block is partial (and padded).

We call the new construction iCMAC for improved CMAC. The construction is essentially the same as CBC-MAC and CMAC. In CMAC, the last block is XOR-ed with either L_1 or L_2 according

as whether $r < n$ or $r = n$. The L_is are obtained by applying π to 0^n. This requires an extra invocation of π over and above that of the m invocations required to process the m blocks. To avoid this extra invocation, we handle the processing of the last block differently. Suppose there are more than one blocks, i.e., $m > 1$. Then $\pi(D_{m-1})$ is "tweaked" by either ψ^{k_1} or ψ^{k_2} according as $r < n$ or $r = n$. This tweaking is sufficient to both distinguish between padded and unpadded last blocks as also to ensure that the requirement of prefix condition by CBC-MAC is removed. But, this tweaking cannot be applied when $m = 1$ and so in this case, we fall back upon the CMAC strategy.

From an efficiency point of view, the main feature of iCMAC is that if $\mathsf{len}(x) > n$, then exactly m invocations of π are required *and* only a single key is required. For CMAC, *either* $(m + 1)$ invocations of π are required *or* an extra n-bit key material is required. Most applications will authenticate messages of length greater than n and hence, iCMAC is an improvement over CMAC for most applications. Admittedly, the improvement is small, but, any improvement is desirable when a scheme is likely to be heavily used and especially if it comes at no extra cost. There is, on the other hand, the issue of complexity of proof. The proof requires more cases to be handled. Most of these cases, however, are of routine nature and the central part is that of CBC-MAC.

Lemma 19. *Let P, Q and $Q_1 \neq Q_2$ be elements of T; l, $l_1 \neq l_2$ be from $\{0, \ldots, 2^n - 2\}$. Then we have the following.*

1. $\Pr[\psi^l(\pi(Q)) = P] = 1/\#T$.
2. $\Pr[\psi^{l_1}(\pi(Q)) \oplus \psi^{l_2}(\pi(Q)) = P] = 1/\#T$.

Proof: $R = \pi(Q)$ is uniformly distributed over T and so (1) and (2) follow from Definition 5. □

Lemma 20. *Let P be an element of T; l, $l_1 \neq l_2$ be from $\{0, \ldots, 2^n - 2\}$; and i, j be such that $0 \leq i < m$, $0 \leq j < m'$. Then*

1. $\Pr[\psi^l(\pi(D_i)) \oplus \psi^l(\pi(D'_j)) = P_{i+1} \oplus P'_{j+1}] = 1/(\#T - 1)$.
2. $\Pr[\psi^{l_1}(\pi(D_i)) \oplus \psi^{l_2}(\pi(D'_j)) = P_{i+1} \oplus P'_{j+1}] = 1/(\#T - 1)$.

Proof: Consider (1). By linearity of ψ, $\psi^l(\pi(D_i)) \oplus \psi^l(\pi(D'_j)) = \psi^l(\pi(D_i) \oplus \pi(D'_j))$. Let $Q = \psi^{-l}(P_{i+1} \oplus P'_{j+1})$. We have

$$
\begin{aligned}
&\Pr[\psi^l(\pi(D_i)) \oplus \psi^l(\pi(D'_j)) = P_{i+1} \oplus P'_{j+1}] \\
&\quad = \Pr[\pi(D_i) \oplus \pi(D'_j) = Q] \\
&\quad = \Pr[\pi(D_i) \oplus \pi(D'_j) = Q | (D_i = D'_j)] \times \Pr[D_i = D'_j] \\
&\qquad + \Pr[\pi(D_i) \oplus \pi(D'_j) = Q | (D_i \neq D'_j)] \times \Pr[D_i \neq D'_j] \\
&\quad \leq [[P_{i+1} = P'_{j+1}]] \Pr[D_i = D'_j] + \tfrac{[[P_{i+1} \neq P'_{j+1}]]}{\#T-1} \times \Pr[D_i = D'_j] \\
&\quad \leq \tfrac{[[P_{i+1} \neq P'_{j+1}]]}{\#T-1} + [[P_{i+1} = P'_{j+1}]] \Pr[D_i = D'_j] \\
&\quad \leq \tfrac{1}{\#T-1}.
\end{aligned}
$$

The last inequality follows from an analysis similar to that done for CBC-MAC in Lemma 7.

Now consider (2). Let $P = P_{i+1} \oplus P'_{j+1}$ and $Q = \pi(D_i) = \pi(D'_j)$, if $D_i = D'_j$; and $Q_1 = \pi(D_i)$, $Q_2 = \pi(D'_j)$, if $D_i \neq D'_j$.

$$
\begin{aligned}
&\Pr[\psi^{l_1}(\pi(D_i)) \oplus \psi^{l_2}(\pi(D'_j)) = P] \\
&\quad = \Pr[\psi^{l_1}(\pi(D_i)) \oplus \psi^{l_2}(\pi(D'_j)) = P | (D_i = D'_j)] \times \Pr[D_i = D'_j]
\end{aligned}
$$

$$
\begin{aligned}
&+ \Pr[\psi^{l_1}(\pi(D_i)) \oplus \psi^{l_2}(\pi(D'_j)) = P | (D_i \neq D'_j)] \times \Pr[D_i \neq D'_j] \\
\leq\ & \Pr[\psi^{l_1}(Q) \oplus \psi^{l_2}(Q) = P|] \times \Pr[D_i = D'_j] + \Pr[\psi^{l_1}(Q_1) \oplus \psi^{l_2}(\pi(Q_2)) = P|] \times \Pr[D_i \neq D'_j] \\
\leq\ & \tfrac{1}{\#T-1} \Pr[D_i = D'_j] + \tfrac{1}{\#T-1} \Pr[D_i \neq D'_j] \\
=\ & \tfrac{1}{\#T-1}.
\end{aligned}
$$

The last but one row follows from the properties of ψ given in Definition 5. □

The following results follows from Lemmas 19 and 20 using essentially a case analysis.

Lemma 21. *If $m = 1$ or $m' = 1$, then for $1 \leq i \leq m$, $\Pr[D'_{m'} = D_i] \leq 1/\#T$.*

Proof: The proof consists of several cases.

Case $m' = m = 1$:

Subcase ($r' < n$, $r < n$) or ($r' = n$, $r = n$):

$D'_1 = P'_1 \oplus L_b$; $D_1 = P_1 \oplus L_b$; with $b = 1$ or 2.

$\Pr[D'_1 = D_1] = \Pr[P'_1 \oplus L_b = P_1 \oplus L_b] = [[P'_1 = P_1]] = 0$.

Subcase ($r' < n$, $r = n$) or ($r' = n$, $r < n$):

Both conditions are similar and we consider the first one.

$D'_1 = P'_1 \oplus L_1$; $D_1 = P_1 \oplus L_2$;

$\Pr[D'_1 = D_1] = \Pr[P'_1 \oplus \psi^{k_1}(\pi(D_0)) = P_1 \oplus \psi^{k_2}(\pi(D_0))] \leq 1/\#T$ (Lemma 19(2)).

Case $m' = 1$, $m > 1$:

Subcase $r' < n$, $i = 1$:

$\Pr[D'_1 = D_1] = \Pr[P'_1 \oplus \psi^{k_1}(\pi(D_0)) = P_1] \leq 1/\#T$ (Lemma 19(1)).

Subcase $r' < n$, $1 < i < m$:

$\Pr[D'_1 = D_i] = \Pr[P'_1 \oplus \psi^{k_1}(\pi(D_0)) = \pi(D_{i-1}) \oplus P_i] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' < n$, $i = m$, $r < n$:

$\Pr[D'_1 = D_i] = \Pr[P'_1 \oplus \psi^{k_1}(\pi(D_0)) = \psi^{k_1}(\pi(D_{m-1})) \oplus P_m] \leq 1/\#T$ (Lemma 20(1)).

Subcase $r' < n$, $i = m$, $r = n$:

$\Pr[D'_1 = D_i] = \Pr[P'_1 \oplus \psi^{k_1}(\pi(D_0)) = \psi^{k_2}(\pi(D_{m-1})) \oplus P_m] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $i = 1$:

$\Pr[D'_1 = D_1] = \Pr[P'_1 \oplus \psi^{k_2}(\pi(D_0)) = P_1] \leq 1/\#T$ (Lemma 19(1)).

Subcase $r' = n$, $1 < i < m$:

$\Pr[D'_1 = D_i] = \Pr[P'_1 \oplus \psi^{k_2}(\pi(D_0)) = \pi(D_{i-1}) \oplus P_i] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $i = m$, $r < n$:

$\Pr[D'_1 = D_i] = \Pr[P'_1 \oplus \psi^{k_2}(\pi(D_0)) = \psi^{k_1}(\pi(D_{m-1})) \oplus P_m] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $i = m$, $r = n$:

$\Pr[D'_1 = D_i] = \Pr[P'_1 \oplus \psi^{k_2}(\pi(D_0)) = \psi^{k_2}(\pi(D_{m-1})) \oplus P_m] \leq 1/\#T$ (Lemma 20(1)).

Case $m' > 1$, $m = 1$:

Subcase $r' < n$, $r < n$:

$\Pr[D'_{m'} = D_1] = \Pr[\psi^{k_1}(\pi(D'_{m'-1})) \oplus P'_{m'} = P_1 \oplus \psi^{k_1}(\pi(D_0))] \leq 1/\#T$ (Lemma 20(1)).

Subcase $r' < n$, $r = n$:

$\Pr[D'_{m'} = D_1] = \Pr[\psi^{k_1}(\pi(D'_{m'-1})) \oplus P'_{m'} = P_1 \oplus \psi^{k_2}(\pi(D_0))] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $r < n$:

$\Pr[D'_{m'} = D_1] = \Pr[\psi^{k_2}(\pi(D'_{m'-1})) \oplus P'_{m'} = P_1 \oplus \psi^{k_1}(\pi(D_0))] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $r = n$:

$\Pr[D'_{m'} = D_1] = \Pr[\psi^{k_2}(\pi(D'_{m'-1})) \oplus P'_{m'} = P_1 \oplus \psi^{k_2}(\pi(D_0))] \leq 1/\#T$ (Lemma 20(1)).

Lemma 22. *For $1 \leq i \leq m$, $\Pr[D'_{m'} = D_i] \leq 1/(\#T - 1)$.*

Proof: For $m = 1$ or $m' = 1$, the result is given by Lemma 21. So, we consider both m and m' to be greater than 1.

Subcase $r' < n$, $i = 1$:
$\Pr[D'_{m'} = D_1] = \Pr[\psi^{k_1}(\pi(D'_{m'-1})) \oplus P'_{m'} = P_1] \leq 1/\#T$ (Lemma 19(1)).

Subcase $r' < n$, $1 < i < m$:
$\Pr[D'_{m'} = D_i] = \Pr[\psi^{k_1}(\pi(D'_{m'-1})) \oplus P'_{m'} = \pi(D_{i-1}) \oplus P_i] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' < n$, $i = m$, $r < n$:
$\Pr[D'_{m'} = D_m] = \Pr[\psi^{k_1}(\pi(D'_{m'-1})) \oplus P'_{m'} = \psi^{k_1}(\pi(D_{m-1})) \oplus P_m] \leq 1/(\#T - 1)$ (Lemma 20(1)).

Subcase $r' < n$, $i = m$, $r = n$:
$\Pr[D'_{m'} = D_m] = \Pr[\psi^{k_1}(\pi(D'_{m'-1})) \oplus P'_{m'} = \psi^{k_2}(\pi(D_{m-1})) \oplus P_m] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $i = 1$:
$\Pr[D'_{m'} = D_1] = \Pr[\psi^{k_2}(\pi(D'_{m'-1})) \oplus P'_{m'} = P_1] \leq 1/\#T$ (Lemma 19(1)).

Subcase $r' = n$, $1 < i < m$:
$\Pr[D'_{m'} = D_i] = \Pr[\psi^{k_2}(\pi(D'_{m'-1})) \oplus P'_{m'} = \pi(D_{i-1}) \oplus P_i] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $i = m$, $r < n$:
$\Pr[D'_{m'} = D_m] = \Pr[\psi^{k_2}(\pi(D'_{m'-1})) \oplus P'_{m'} = \psi^{k_1}(\pi(D_{m-1})) \oplus P_m] \leq 1/\#T$ (Lemma 20(2)).

Subcase $r' = n$, $i = m$, $r = n$:
$\Pr[D'_{m'} = D_m] = \Pr[\psi^{k_2}(\pi(D'_{m'-1})) \oplus P'_{m'} = \psi^{k_2}(\pi(D_{m-1})) \oplus P_m] \leq 1/(\#T - 1)$ (Lemma 20(1)).

In each case, the probability is at most $1/(\#T - 1)$ which proves the result. □

The following result is proved by a similar case analysis.

Lemma 23. *For $1 \leq i < m$, $\Pr[D_m = D_i] \leq 1/(\#T - 1)$.*

Using Lemmas 22 and 23 show that iCMAC-HASH is $1/(\#T-1)$-CR and $(1/(\#T-1), 1/(\#T-1))$-disjoint with respect to λ. Using this and Theorem 3, we obtain a bound on the advantage of iCMAC in a manner similar to that of CBC-MAC.

Theorem 8. *Let d and $\sigma \geq d$ be positive integers. Then*

$$\mathbf{Adv}_{\mathrm{iCMAC}}(d, \sigma) \leq \frac{d(d-1) + \sigma(1+2d)}{\#T - 1}.$$

4.6 iVCBC-MAC

CBC-MAC has been modified to obtain iCMAC. In a similar manner, we can modify VCBC-MAC to obtain a variant iVCBC-MAC which can handle arbitrary length strings. In this case, a uniform random function $\rho : \{0,1\}^{\ell} \rightarrow \{0,1\}^n$ will be used, where $\ell \geq n$. The message x is an arbitrary length string which is formatted into $(P_1, \ldots, P_m)$ by $\mathsf{Format}(x, \ell)$ given in Figure 2 which also defines the values of m and r.

Again, assume that $T = GF(2^n)$ under the usual identification of n-bit strings and polynomials over $GF(2)$ of degree less than n. Let $\psi : T \rightarrow T$ be a map satisfying Definition 5. Let fStr be a

fixed string of length ℓ; and k_1, k_2 be two distinct integers in the range $1 \le k_1, k_2 \le 2^n - 2$. Define $L = \rho(\mathsf{fStr})$, $L_1 = \psi^{k_1}(L)$ and $L_2 = \psi^{k_2}(L_2)$.

$$\mathsf{iVCBC\text{-}MAC} : x \mapsto C_m \tag{39}$$

where $C_i = \rho(D_i)$ for $1 \le i \le m$ and

$$D_i = \begin{cases} L_1 \oplus\!\!\!\rightarrow P_1 & \text{if } i = m = 1,\ r < n; \\ L_2 \oplus\!\!\!\rightarrow P_1 & \text{if } i = m = 1,\ r = n; \\ P_1 & \text{if } i = 1,\ m > 1; \\ \rho(D_{i-1}) \oplus\!\!\!\rightarrow P_i & \text{if } 1 < i < m,\ m > 1; \\ \psi^{k_1}(\rho(D_{m-1})) \oplus\!\!\!\rightarrow P_m & \text{if } i = m,\ m > 1, r < n; \\ \psi^{k_2}(\rho(D_{m-1})) \oplus\!\!\!\rightarrow P_m & \text{if } i = m,\ m > 1, r = n; \end{cases} \tag{40}$$

Figure 5 shows an example of processing a 3-block message with iVCBC-MAC. A scheme using a compressing function is also described in [27]. (This scheme has been called iCBC-MAC in [27].) For ease of comparison to our scheme, in Figure 6, we show how a 3-block message is processed using the method in [27]. Compared to iCBC-MAC, iVCBC-MAC will require two less invocations of ρ.

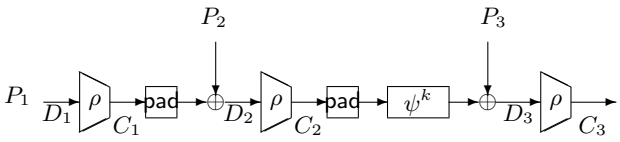

Fig. 5. iVCBC-MAC. Here k takes the value k_1 if the last block is full and the value k_2 if the last block is partial (and padded).

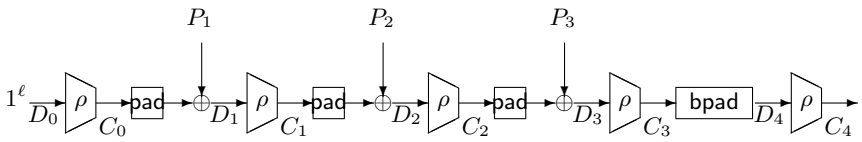

Fig. 6. The scheme from [27]. Here bpad refers to padding with with $0^{\ell-n}$ in the front and with 10^* at the end.

The security of the scheme can be shown using analysis similar to that of the other schemes. Mainly, we would have to combine the analysis of VCBC-MAC with that of iCMAC. Given these two analysis, the details for iVCBC-MAC are routine and we simply present the final result.

Theorem 9. *Let d and $\sigma \ge d$ be positive integers. Then*

$$\mathbf{Adv}_{\mathrm{iVCBC-MAC}}(d, \sigma) \le \frac{\sigma(2d + 5\sigma)}{2\#T}.$$

5 Parallelizable Constructions

The basic idea of parallelizing is to apply the permutation π separately on (masked) blocks and then XOR the outputs together and apply π on this XOR. Though simple in principle, this idea

needs to be worked out carefully. PCS [4] and PMAC [7, 23] are based on this principle. PCS uses a compressing function, whereas PMAC uses a permutation. We describe two constructions: iPMAC and VPMAC using a permutation and a compressing function respectively. The construction iPMAC improves upon PMAC by removing a design stage discrete log computation requirement while VPMAC improves upon PCS by reducing the number of invocations of the compressing function. As mentioned earlier, a suitable compressing function which can be used in practical constructions is surf [3] and has been suggested in [4].

5.1 iPMAC

Let $T = \{0,1\}^n$ and π be a uniform random permutation of T. Let fStr be a fixed element of T, i.e., a fixed n-bit string and $R = \pi(\mathsf{fStr})$. Let ψ be a function satisfying Definition 5. Given any binary string x, let $(P_1, \ldots, P_m)$ be the output of $\mathsf{Format}(x, n)$ given in Figure 2 which also defines the values of m and r. Define iPMAC to be a function

$$\mathsf{iPMAC} : x \mapsto C_m$$

where $C_i = \pi(D_i)$ for $1 \le i \le m$ and

$$D_i = \begin{cases} P_i \oplus \psi^i(R) & 1 \le i \le m-1; \\ C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m & i = m, r = n; \\ C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \psi^m(R) & i = m, r < n. \end{cases} \tag{41}$$

Processing of a 4-block message using iPMAC is shown in Figure 7. The same figure also describes the processing of a 4-block message using PMAC. The difference is in the interpretation of Λ.

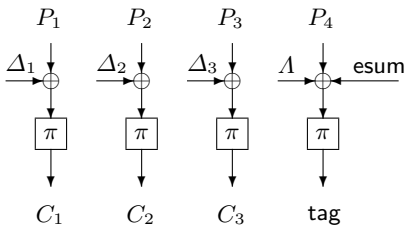

Fig. 7. Tag generation using iPMAC Here $\Delta_i = \psi^i(R)$; $\Lambda = 0^n$ if the last block is full and $\Lambda = \psi^4(R)$ if the last block has been padded; and $\mathsf{esum} = C_1 \oplus C_2 \oplus C_3$.

iPMAC. If the last block is full, then $\Lambda = 0^n$ and if the last block is partial (and padded), then $\Lambda = \psi^m(R)$. (See Section 2.3 for more details about this map.)

PMAC. If the last block is full, then $\Lambda = u^m R \bmod \tau(u)$ and if the last block is partial (and padded), then $\Lambda = (u \oplus 1)u^m R \bmod \tau(u)$. For this scheme to be secure, the discrete log of $(u \oplus 1)$ to the base u has to be "large". The actual value depends on $\tau(x)$ and for specific values of $\tau(x)$ with $n = 64$ and $n = 128$, the values are given in [23]. Changing $\tau(x)$ will require a re-computation of the discrete log to ensure that it is "large"; also for $n = 256$, it will be difficult to compute the discrete log. In contrast, iPMAC entirely avoids this discrete log computation.

Define iPHASH to be the function which maps x to D_m. Then iPMAC $= \pi \circ$ iPHASH and iPMAC is a domain extender for π in the sense of Definition 6. We compute the collision and disjointness probabilities for iPHASH.

Lemma 24. *Let x and x' be two distinct messages with $m = \lambda(x)$ and $m' = \lambda(x')$, which are mapped to D_m and $D'_{m'}$ under* iPHASH. *Then* $\Pr[D_m = D'_{m'}] \leq (m+m')/\#T \leq 2\max(m, m')/\#T$.

Proof: Assume without loss of generality that $m \geq m'$. There are four cases depending on whether r and r' are less than n or equal to n.

Case $r = n$, $r' = n$: Since $x \neq x'$, let j be the first index such that either ($1 \leq j \leq m'$ and $P_j \neq P'_j$) or ($j = m' + 1$ and $P_i = P'_i$ for $1 \leq i \leq m'$).

If $j = m = m'$, then $P_i = P'_i$ for $1 \leq i \leq m' - 1$ and so $C_i = C'_i$ for $1 \leq i \leq m - 1$. So, $D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \neq C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus P'_m = D'_{m'}$ and $\Pr[D_m = D'_{m'}] = 0$.

If $j = m = m' + 1$, then $P_i = P'_i$ for $1 \leq i \leq m' - 1$ and so $C_i = C'_i$ for $1 \leq i \leq m' - 1$. So,

$$\begin{aligned} D_m \oplus D'_{m'} &= C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus P'_{m'} \\ &= C_{m-1} \oplus P_m \oplus P'_{m'} \end{aligned}$$

Since C_{m-1} is the output of π, it is uniformly distributed over T and hence, the last expression is zero with probability $1/\#T$.

So, we can assume that either ($m > m' + 1$, $j = m' + 1$) or ($1 \leq j \leq m'$ and $m > m'$). In either case, $D_j = \psi^j(R) + P_m$. We claim that with high probability D_j is different from $D_1, \ldots, D_{j-1}, D_{j+1}, \ldots, D_{m-1}$ and $D'_1, \ldots, D'_{m-1}$. To see this, first note that $D_i = P_i \oplus \psi^i(R)$, $1 \leq i \leq m - 1$; and $D'_k = P'_k \oplus \psi^k(R)$, $1 \leq k \leq m' - 1$. Let $\mathcal{E}$ be the event

$$\mathcal{E} : \left(\bigwedge_{\substack{i=1, \\ i \neq j}}^{m-1} (D_j \neq D_i) \right) \wedge \left(\bigwedge_{i=1}^{m'-1} (D_j \neq D'_i) \right).$$

In other words, the event $\mathcal{E}$ happens when D_j is distinct from all other D_i's and is also distinct from $D'_1, \ldots, D'_{m'-1}$. We first show that $\mathcal{E}$ occurs with high probability.

$$\begin{aligned} \Pr[\mathcal{E}] &= 1 - \Pr[\overline{\mathcal{E}}] \\ &\geq 1 - \sum_{\substack{i=1, \\ i \neq j}}^{m-1} \Pr[D_j = D_i] - \sum_{i=1}^{m'-1} \Pr[D_j = D'_i]. \end{aligned}$$

If $j < m'$, then since $P_j \neq P'_j$, $D_j = P_j \oplus \psi^j(R) \neq P'_j \oplus \psi^j(R) = D'_j$ so that $\Pr[D_j = D'_j] = 0$. In all other cases, the individual probabilities of either $D_j = D'_i$ or $D_j = D_i$ for $i \neq j$ are $1/\#T$ by the properties of ψ given in Definition 5. So,

$$\Pr[\mathcal{E}] \geq \left(1 - \frac{m + m' - 3}{\#T}\right).$$

We have

$$\begin{aligned} \Pr[D_m \neq D'_{m'}] &\geq \Pr[(D_m \neq D'_{m'}) \wedge \mathcal{E}] \\ &= \Pr[(D_m \neq D'_{m'}) | \mathcal{E}] \Pr[\mathcal{E}] \\ &\geq \Pr[(D_m \neq D'_{m'}) | \mathcal{E}] \times \left(1 - \frac{m + m' - 3}{\#T}\right) \end{aligned}$$

Consider the event $((D_m \neq D'_{m'})|\mathcal{E})$. Since π is a permutation and D_j is distinct from all other D_is and $D'_1, \ldots, D'_{m'-1}$, we have that C_j is distinct from all other C_is and $C'_1, \ldots, C'_{m'-1}$.

Since $r = r' = n$, we have

$$D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m$$
$$D'_{m'} = C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus P'_{m'}.$$

Consider the possible multiset

$$\{C_1, \ldots, C_{j-1}, C_{j+1}, \ldots, C_{m-1}, C'_1, \ldots, C'_{m'-1}\}.$$

Let $Q_1, \ldots, Q_t$, for $t \geq 0$ be the elements of this multiset which occur with odd frequencies. So, $D_m \oplus D'_{m'} = 0$ implies that $C_j \oplus Q_1 \oplus \ldots \oplus Q_t = P_m \oplus P'_{m'}$ for some $t \geq 0$ and $(C_j, Q_1, \ldots, Q_t)$ is distributed uniformly over $\chi_{t+1}(T)$.

1. If $t = 0$, then $\Pr[D_m \neq D'_{m'}|\mathcal{E}] = \Pr[C_j \neq P_m \oplus P'_{m'}] = (1 - 1/\#T)$.
2. If $t = 1$ and $P_m = P'_{m'}$, then $\Pr[D_m \neq D'_{m'}|\mathcal{E}] = \Pr[C_j \neq Q_t] = 1$.
3. In all other cases, $\Pr[D_m \neq D'_{m'}|\mathcal{E}] = \Pr[C_j \oplus Q_1 \oplus \cdots \oplus Q_t \neq P_m \oplus P'_{m'}] \geq 1 - 1/(\#T - t) \geq 1 - 1/(\#T - (m + m' - 2)) \geq 1 - 2/\#T$ (assuming $m + m' - 2 \leq \#T$).

Thus, the inequality, $\Pr[D_m \neq D'_{m'}|\mathcal{E}] \geq 1 - 2/\#T$ holds for all t.

From this we have $\Pr[D_m \neq D'_{m'}] \geq (1-(m+m'-1)/\#T)$ and so $\Pr[D_m = D'_{m'}] \leq (m+m')/\#T$.

Case $r < n$, $r' < n$: In this case, we have

$$D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \psi^m(R)$$
$$D'_{m'} = C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus P'_{m'} \oplus \psi^{m'}(R).$$

If $m = m'$, then the terms involving ψ cancel out and the analysis is exactly the same as that for the case $r = r' = n$. So suppose $m > m'$. Let $\mathcal{E}$ be the event that fStr is not equal to any of $D_1, \ldots, D_{m-1}$ or $D'_1, \ldots, D'_{m'-1}$. The probability of $\mathcal{E}$ is at least $1-(m+m'-2)/\#T$. In a manner similar to the previous case, it can be shown $\Pr[D_m \neq D'_{m'}|\mathcal{E}] \geq 1 - 2/\#T$ so that we again have $\Pr[D_m \neq D'_{m'}] \leq (m + m')/\#T$.

Cases ($r = n$, $r' < n$) and ($r < n$, $r' = n$): Both the cases are similar and we consider only $r = n$ and $r' < n$. In this case, we have

$$D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \psi^m(R)$$
$$D'_{m'} = C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus P'_{m'}.$$

It is possible that $m = m'$ and $P_i = P'_i$ for $1 \leq i \leq m$ even though $x \neq x'$. This happens when $x = \mathsf{pad}(x') \neq x'$. Then, $D_m \oplus D'_{m'} = \psi^m(R)$ which is equal to 0 with probability $1/\#T$. If $m > m'$ or $P_i \neq P'_i$ for some $1 \leq i \leq m'$, then an analysis similar to the previous case shows the desired result. □

The disjointness probabilities can be bound in a similar manner and is given by the following result.

Lemma 25. *Let x and x' be two distinct messages having m and m' blocks respectively. Then*

1. $\Pr[D'_{m'} = D_i] \le 2/\#T$ *for* $1 \le i \le m-1$;
2. $\Pr[D_m = D_i] \le 2/\#T$ *for* $1 \le i \le m-1$.

Consequently, $\Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] \le (m+m')/\#T$ *and* $\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \le 2m/\#T$.

Using Theorem 2 gives the following result.

Theorem 10. *Let* d *and* $\sigma \ge d$ *be positive integers. Then*

$$\mathbf{Adv}_{\mathrm{iPMAC}}(d, \sigma) \le \frac{d(d-1) + 4\sigma(1+3d)}{2\#T}.$$

Note. A proof for PMAC can be obtained from the above proof with small changes. Basically, when $r = n$, we will be masking with $\psi^m(R)$ and when $r < n$, we will be masking with $((1 \oplus \psi)\psi^m)(R)$. This does not change the nature of the arguement too much and we obtain the same bound as for iPMAC.

5.2 VPMAC: A Variant of iPMAC

Let $T = \{0,1\}^n$ and $\rho : \{0,1\}^\ell \to \{0,1\}^n$ with $\ell > n$ be a uniform random function. Let fStr be a fixed element of U, i.e., a fixed ℓ-bit string and $R = \rho(\mathsf{fStr}) \in T$. Let ψ be a function satisfying Definition 5. Given any binary string x, let $(P_1, \ldots, P_m)$ be the output of $\mathsf{Format}(x, \ell)$ given in Figure 2 which also defines the values of m and r. Define VPMAC to be a function

$$\mathsf{VPMAC} : x \mapsto C_m$$

where $C_i = \rho(D_i)$ for $1 \le i \le m$ and

$$D_i = \begin{cases} \psi^i(R) \oplus\!\!\rightarrow P_i & 1 \le i \le m-1; \\ (C_1 \oplus \cdots \oplus C_{m-1}) \oplus\!\!\rightarrow P_m & i = m, r = n; \\ (C_1 \oplus \cdots \oplus C_{m-1} \oplus \psi^m(R)) \oplus\!\!\rightarrow P_m & i = m, r < n. \end{cases} \tag{42}$$

Figure 8 shows how a 4-block message is processed using PCS and VPMAC. The message lengths, however, are different. PCS processes $4n$ bits, while VPMAC processes 4ℓ bits. In general, PCS requires $1 + \lceil \mathsf{len}(x)/n \rceil$ invocations of ρ to process a message x, while VPMAC requires $1 + \lceil \mathsf{len}(x)/\ell \rceil$ invocations of ρ. Thus, VPMAC requires approximately a fraction n/ℓ of the invocations of PCS.

Define VPHASH to be the function which maps x to D_m. Then $\mathsf{VPMAC} = \rho \circ \mathsf{VPHASH}$ and VPMAC is a domain extender for ρ in the sense of Definition 6. The following results can be proved in a manner similar to that for iPMAC.

Lemma 26. *Let* x *and* x' *be two distinct messages which are mapped to* D_m *and* $D'_{m'}$ *under* VPHASH. *Then* $\Pr[D_m = D'_{m'}] \le (m+m')/\#T$.

Note. The proof is very similar to that of Lemma 24. An interesting aspect of the result is that the expression for the collision probability is quite simple even though we use a function ρ rather than a permutation π. In the case of sequential construction, using ρ led to a more complicated expression (see Lemma 11). This does not happen for the parallel construction and to emphasize this, we give a proof of Lemma 26.

Proof: Assume without loss of generality that $m \ge m'$. There are four cases depending on whether r and r' are less than ℓ or equal to ℓ.

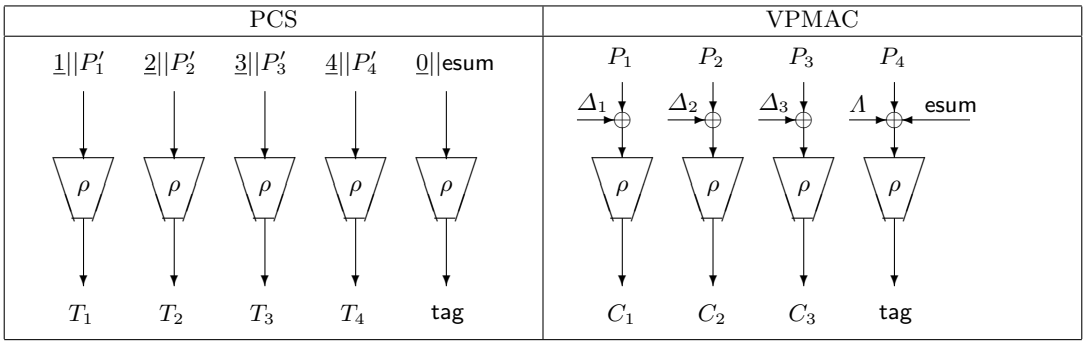

Fig. 8. Tag generation using PCS and VPMAC. In PCS, $\mathsf{esum} = T_1 \oplus T_2 \oplus T_3 \oplus T_4$. In VPMAC, $\Delta_i = \psi^i(R)$; $\Lambda = 0^n$ if the last block is full and $\Lambda = \psi^4(R)$ if the last block has been padded; and $\mathsf{esum} = C_1 \oplus C_2 \oplus C_3$.

Case $r = \ell$, $r' = \ell$: Since $x \neq x'$, let j be the first index such that either ($1 \leq j \leq m'$ and $P_j \neq P'_j$) or ($j = m' + 1$ and $P_i = P'_i$ for $1 \leq i \leq m'$).

If $j = m = m'$, then $P_i = P'_i$ for $1 \leq i \leq m' - 1$ and so $C_i = C'_i$ for $1 \leq i \leq m - 1$. Then $D_m \oplus D'_{m'} = ((C_1 \oplus \cdots \oplus C_m) \oplus\!\!\!\rightarrow P_m) \oplus ((C'_1 \oplus \cdots \oplus C'_m) \oplus\!\!\!\rightarrow P'_m) = P_m \oplus P'_m)$ and so $\Pr[D_m = D'_{m'}] = 0$.

If $j = m = m' + 1$, then $P_i = P'_i$ for $1 \leq i \leq m' - 1$ and so $C_i = C'_i$ for $1 \leq i \leq m - 1$. So,

$$\begin{aligned} D_m \oplus D'_{m'} &= ((C_1 \oplus \cdots \oplus C_{m-1}) \oplus\!\!\!\rightarrow P_m) \oplus ((C'_1 \oplus \cdots \oplus C'_{m'-1}) \oplus\!\!\!\rightarrow P'_{m'}) \\ &= C_{m-1} \oplus\!\!\!\rightarrow (P_m \oplus P'_{m'}) \end{aligned}$$

Since C_{m-1} is the output of π, it is uniformly distributed over T and hence it is equal to $\mathsf{bot}(P_m \oplus P'_{m'})$ with probability $1/\#T$.

So, we can assume that either ($m > m' + 1$, $j = m' + 1$) or ($1 \leq j \leq m'$ and $m > m'$). In both cases, we have $D_j = \psi^j(R) \oplus\!\!\!\rightarrow P_j$. We claim that with high probability D_j is different from $D_1, \ldots, D_{j-1}, D_{j+1}, \ldots, D_{m-1}$ and $D'_1, \ldots, D'_{m-1}$. To see this first note that $D_i = \psi^i(R) \oplus\!\!\!\rightarrow P_i$, $1 \leq i \leq m - 1$; and $D'_k = \psi^k(R) \oplus\!\!\!\rightarrow P'_k$, $1 \leq k \leq m' - 1$. Let $\mathcal{E}$ be the event

$$\mathcal{E} : \left(\bigwedge_{\substack{i=1, \\ i \neq j}}^{m-1} (D_j \neq D_i) \right) \wedge \left(\bigwedge_{i=1}^{m'-1} (D_j \neq D'_i) \right).$$

Then

$$\begin{aligned} \Pr[\mathcal{E}] &= 1 - \Pr[\overline{\mathcal{E}}] \\ &\geq 1 - \sum_{\substack{i=1, \\ i \neq j}}^{m-1} \Pr[D_j = D_i] - \sum_{i=1}^{m'-1} \Pr[D_j = D'_i]. \end{aligned}$$

If $j < m'$, then since $P_j \neq P'_j$, $D_j = \psi^j(R) \oplus\!\!\!\rightarrow P_j \neq \psi^j(R) \oplus\!\!\!\rightarrow P'_j = D'_j$. In all other cases, the individual probabilities are $1/\#T$ by properties of ψ given in Definition 5. So,

$$\Pr[\mathcal{E}] \geq \left(1 - \frac{m + m' - 3}{\#T}\right).$$

We have

$$\begin{aligned}\Pr[D_m \neq D'_{m'}] &\geq \Pr[(D_m \neq D'_{m'}) \wedge \mathcal{E}] \\ &= \Pr[(D_m \neq D'_{m'})|\mathcal{E}] \Pr[\mathcal{E}] \\ &\geq \Pr[(D_m \neq D'_{m'})|\mathcal{E}] \times \left(1 - \frac{m+m'-3}{\#T}\right)\end{aligned}$$

Consider the event $((D_m \neq D'_{m'})|\mathcal{E})$. Since ρ is a uniform random function from U to T, and D_j is distinct from all other D_is and $D'_1, \ldots, D'_{m'-1}$, we have that C_j is uniformly distributed over T and is independent of all other C_is and $C'_1, \ldots, C'_{m'-1}$. Let

$$Q = (C_1 \oplus \cdots \oplus C_{j-1} \oplus C_{j+1} \oplus \cdots \oplus C_{m-1}) \oplus (C'_1 \oplus \cdots \oplus C'_{m'-1}).$$

Then C_j is independent of Q.

Since $r = r' = n$, we have

$$\begin{aligned}D_m &= (C_1 \oplus \cdots \oplus C_{m-1}) \oplus\!\!\rightarrow P_m \\ D'_{m'} &= (C'_1 \oplus \cdots \oplus C'_{m'-1}) \oplus\!\!\rightarrow P'_{m'}.\end{aligned}$$

So, $D_m \oplus D'_{m'} = 0$ implies that $C_j \oplus Q = \mathsf{bot}(P_m \oplus P'_{m'})$. Since C_j is uniformly distributed over T, this holds with probability $1/\#T$, i.e., $\Pr[D_m \neq D'_{m'}|\mathcal{E}] \geq 1 - 1/\#T$.

From this we have, $\Pr[D_m \neq D'_{m'}] \geq (1 - (m + m' - 2)/\#T)$ and so $\Pr[D_m = D'_{m'}] \leq (m + m')/\#T$.

Case $r < n$, $r' < n$: In this case, we have

$$\begin{aligned}D_m &= (C_1 \oplus \cdots \oplus C_{m-1} \oplus \psi^m(R)) \oplus\!\!\rightarrow P_m \\ D'_{m'} &= (C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus \psi^{m'}(R)) \oplus\!\!\rightarrow P'_{m'}.\end{aligned}$$

If $m = m'$, then the terms involving ψ cancel out and the analysis is exactly the same as that for the case $r = r' = n$. So suppose $m > m'$. Let $\mathcal{E}$ be the event that fStr is not equal to any of $D_1, \ldots, D_{m-1}$ or $D'_1, \ldots, D'_{m'-1}$. The probability of $\mathcal{E}$ is at least $1-(m+m'-2)/\#T$. In a manner similar to the previous case, it can be shown $\Pr[D_m \neq D'_{m'}|\mathcal{E}] \geq 1 - 1/\#T$ so that we again have, $\Pr[D_m \neq D'_{m'}] \leq (m + m')/\#T$.

Cases ($r = n$, $r' < n$) and ($r < n$, $r' = n$): Both the cases are similar and we consider only $r = n$ and $r' < n$. In this case, we have

$$\begin{aligned}D_m &= (C_1 \oplus \cdots \oplus C_{m-1} \oplus \psi^m(R)) \oplus\!\!\rightarrow P_m \\ D'_{m'} &= (C'_1 \oplus \cdots \oplus C'_{m'-1}) \oplus\!\!\rightarrow P'_{m'}.\end{aligned}$$

It is possible that $m = m'$ and $P_i = P'_i$ for $1 \leq i \leq m$ even though $x \neq x'$. This happens when $x = \mathsf{pad}(x') \neq x'$. Then, $\mathsf{bot}(D_m \oplus D'_{m'}) = \psi^m(R)$ which is equal to 0 with probability $1/\#T$. If $m > m'$ or $P_i \neq P'_i$ for some $1 \leq i \leq m'$, then an analysis similar to the previous case shows the desired result. □

Lemma 27. *Let x and x' be two distinct messages having m and m' blocks respectively. Then*

1. $\Pr[D'_{m'} = D_i] \leq 1/\#T$ *for* $1 \leq i \leq m-1$;
2. $\Pr[D_m = D_i] \leq 1/\#T$ *for* $1 \leq i \leq m-1$.

Consequently, $\Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] \leq (m+m')/\#T$ *and* $\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \leq m/\#T$.

Combining Lemmas 26 and 27 with Theorem 2, we have the following result.

Theorem 11. *Let* d *and* $\sigma \geq d$ *be positive integers. Then*

$$\mathbf{Adv}_{\mathrm{VPMAC}}(d, \sigma) \leq \frac{d(d-1) + 2\sigma(1+3d)}{\#T}.$$

6 Conclusion

We have analysed pseudorandom functions for use in symmetric key message authentication. Starting from a useful result by Vaudenay [26] and Bernstein [4], we prove several general results on PRF built using a uniform random permutation or a uniform random function.

These results are used to analyse several known and new constructions. Among the know constructions we analyse CBC-MAC and CMAC. New sequential constructions include a variant which improves upon CMAC and a variant of CBC-MAC which uses a compressing function rather than a permutation. New parallel constructions include iPMAC which improves upon PMAC [23] by removing the requirement of a design stage discrete log computation and VPMAC which improves upon PCS [4] by requiring lesser invocations of the compressing functions.

An important feature of our work is the avoidance of the heavy machinery of game playing technique. The entire analysis is reduced to simple probability calculations which are done using elementary techniques. For CBC-MAC, the earlier best known bound was obtained using game-playing strategy in conjunction with rather dense combinatorial techniques. We obtain similar (actually better) bounds using much simpler analysis.

Acknowledgement

We would like to thank Mridul Nandi for pointing out Lemma 22 of [26] to us.

References

1. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.
2. Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved Security Analyses for CBC MACs. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 527–545. Springer, 2005.
3. Daniel J. Bernstein. SURF: Simple Unpredictable Random Functions. `http://pobox.com/~djb/papers/surf.dvi`.
4. Daniel J. Bernstein. How to stretch random functions: The security of protected counter sums. *J. Cryptology*, 12(3):185–192, 1999.
5. Daniel J. Bernstein. A short proof of the unpredictability of the cipher block chaining, 2005. `http://cr.yp.to/papers.html#easycbc`.
6. John Black and Phillip Rogaway. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. In Mihir Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 197–215. Springer, 2000.
7. John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Knudsen [13], pages 384–397.

8. Debrup Chakraborty and Palash Sarkar. A general construction of tweakable block ciphers and different modes of operations. *IEEE Transactions on Information Theory*, 54(5):1991–2006, 2008.
9. Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES – The Advanced Encryption Standard (Information Security and Cryptography)*. Springer, Heidelberg, 2002.
10. M. Dworkin. Recommendation for block cipher modes of operations: the CMAC mode for authentication, May 2005. National Institute of Standards and Technology, U.S. Department of Commerce. NIST Special Publication 800-38B.
11. Tetsu Iwata and Kaoru Kurosawa. OMAC: One-Key CBC MAC. In Thomas Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer, 2003.
12. Charanjit S. Jutla. PRF Domain Extension Using DAGs. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 561–580. Springer, 2006.
13. Lars R. Knudsen, editor. *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*. Springer, 2002.
14. Kaoru Kurosawa and Tetsu Iwata. TMAC: Two-Key CBC MAC. In Marc Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 33–49. Springer, 2003.
15. R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications, revised edition.* Cambridge University Press, 1994.
16. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.
17. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
18. Ueli M. Maurer. Indistinguishability of random systems. In Knudsen [13], pages 110–132.
19. Kazuhiko Minematsu and Toshiyasu Matsushima. New Bounds for PMAC, TMAC, and XCBC. In Alex Biryukov, editor, *FSE*, volume 4593 of *Lecture Notes in Computer Science*, pages 434–451. Springer, 2007.
20. Mridul Nandi. A simple and unified method of proving indistinguishability. In Rana Barua and Tanja Lange, editors, *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 317–334. Springer, 2006.
21. Mridul Nandi and Avradip Mandal. Improved Security Analysis of PMAC. Cryptology ePrint Archive, Report 2007/031, 2007. `http://eprint.iacr.org/`.
22. Erez Petrank and Charles Rackoff. CBC MAC for Real-Time Data Sources. *J. Cryptology*, 13(3):315–338, 2000.
23. Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.
24. Palash Sarkar. A general mixing strategy for the ECB-Mix-ECB mode of operation. *Information Processing Letters*. To appear.
25. Serge Vaudenay. Decorrelation over Infinite Domains: The Encrypted CBC-MAC Case. In Douglas R. Stinson and Stafford E. Tavares, editors, *Selected Areas in Cryptography*, volume 2012 of *Lecture Notes in Computer Science*, pages 189–201. Springer, 2000.
26. Serge Vaudenay. Decorrelation: A theory for block cipher security. *J. Cryptology*, 16(4):249–286, 2003.
27. Kan Yasuda. A Single-Key Domain Extender for Privacy-Preserving MACs and PRFs. In *ICISC*, Lecture Notes in Computer Science. Springer, 2008. To appear.