New Parallelizable 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

February 20th, 2009.
Revised: July 27th, 2009.
Revised: August 11th, 2009.

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 a general result on constructions which use a PRF with "small" domain to build a PRF with "large" domain. This result is used to analyse new parallelizable schemes for PRF construction.

Two new schemes are described. 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 new construction which works with keyed compressing function and compared to PCS requires lesser number of invocations. The variant where the compression function is replaced by a block cipher leads to an improved version of the well known PMAC algorithm. The improvements consist in removing the requirement of a discrete log computation in the design stage of PMAC leading to an easily reconfigurable family of algorithms; and providing faster masking functions for software implementations.

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, 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 $E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$, where $\mathcal{K}$ is the set of keys and $\mathcal{M}$ is the set of messages. For every $K \in \mathcal{K}$, $E_K : \mathcal{M} \rightarrow \mathcal{M}$ 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 [8]. 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) [15]. 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 [9]. 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 [24]. Let $F : S \to T$ 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 [24], 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 $F = \pi \circ F_1^{(\pi)}$, where π is a uniform random permutation; $F_1^{(\pi)}$ 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 [11, 18].

We consider this in the more general setting where $F = \rho \circ F_1^{(\rho)}$, with ρ being either a uniform random permutation or a uniform random function. Suppose x and x' are two inputs to F; $U_1, \ldots, U_m$ and $U'_1, \ldots, U'_{m'}$ are the inputs to the invocations of ρ during the computations of $Z = F_1(x)$ and $Z' = F_1(x')$ respectively. We define three events: collision (Coll), i.e., $Z = Z'$; self-disjoint ($\mathsf{Self\text{-}Disjoint}$), i.e., $\wedge_{i=1}^{m}(Z \neq U_i)$; and pairwise-disjoint ($\mathsf{Pairwise\text{-}Disjoint}$), i.e., $(\wedge_{i=1}^{m}(Z' \neq U_i) \wedge (\wedge_{i=1}^{m'}(Z \neq U'_i))$. We show that if the probabilities of Coll, $\overline{\mathsf{Self\text{-}Disjoint}}$ and

$\overline{\mathsf{Pairwise\text{-}Disjoint}}$ are all small, the PRF-advantage of F is also small. The result is useful, since it reduces the task of bounding the 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.

New Constructions. Parallel MAC schemes start with protected counter sum (PCS) [4] and PMAC [6, 21]. PMAC [21] uses a tweakable block cipher (TBC) and the design of the TBC requires the solution of a discrete log problem over $GF(2^n)$. The designer in [21] 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 [7]. This can avoid the discrete log computation but is then (slightly) slower since it requires more masking operations.

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 describe VPMAC, which is a new parallel construction; also works with such ρ and provides an efficiency improvement. PCS requires $(1 + \mathsf{len}(x)/n)$ invocations of ρ whereas VPMAC requires $(1 + \mathsf{len}(x)/\ell)$ invocations of ρ. Consequently, VPMAC requires approximately a fraction n/ℓ of invocations of ρ compared to PCS.

In the case where $\ell = n$ and ρ is a uniform random permutation, we obtain a new variant of PMAC called iPMAC. This variant does not require discrete log computation but unlike [7] uses the same number of masking operations as in [21]. A consequence of avoiding the discrete log computation is to obtain an easily reconfigurable family of parallel MAC algorithms. A particular algorithm is fixed by a particular representation of the underlying finite field. Changing the field representing polynomial(s) is done by simply changing some constants in the program. The PRF-bound that we obtain for iPMAC is similar to that obtained in [17, 19] for PMAC.

The description of iPMAC is given using a general definition of the required masking functions. There are known instantiations [22] of this general definition which give rise to concrete masking functions which are faster than those used in PMAC. Such instantiations represent the underlying field as a tower field and the masking operations are then defined as clockings of word-oriented LFSRs. In software, these are faster than the so-called powering map used in PMAC.

The constructions and the bounds that we obtain are given in Table 1. From the table, it is clear that VPMAC requires lesser number of invocations compared to PCS. The bound for PCS, on the other hand, is better. This is due to the fact that in PCS, concatenating the counter values ensures that certain collisions do not happen at all, whereas in VPMAC these collisions are only probabilistically ruled out. Having said this, we note that the bound for VPMAC is also good enough and comparable to the bound that have been obtained earlier for PMAC.

Comparing PMAC and iPMAC, we see that the number of invocations are same and the bounds are similar. The advantages of iPMAC over PMAC has been discussed above.

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

Table 1. Features of constructions analysed in this work. For PMAC and iPMAC a uniform random permutation of $\{0,1\}^n$ is to be used, whereas for PCS and VPMAC a uniform random function from $\{0,1\}^\ell$ to $\{0,1\}^n$ is to be used. The column "bound", provides the upper bound 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 column "# invoc" provides the number of invocations of the permutation/function required to process a message x.

scheme	perm?	bound	# invoc
PMAC [21]	yes	$\frac{5d\sigma-3.5d^2}{2^n}$ [19]	$1+\lceil \mathsf{len}(x)/n \rceil$
PCS [4]	no	$\frac{d(d-1)}{2^{n+1}}$ [4]	$1+\lceil \mathsf{len}(x)/n \rceil$
iPMAC	yes	$\frac{(3.5d+1)\sigma}{2^{n-1}}$	$1+\lceil \mathsf{len}(x)/n \rceil$
VPMAC	no	$\frac{(3d+1)\sigma}{2^{n-1}}$	$1+\lceil \mathsf{len}(x)/\ell \rceil$

$2m^2d^2/2^n$. Maurer [16] and Vaudenay [23] gave different proofs for essentially the same bound (upto a constant). Bernstein [5] also gave a different short proof. Petrank and Rackoff [20] 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.

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 [6] and later Rogaway [21] described block cipher based methods for parallel message authentication. The scheme in [6] was called PMAC and the one in [21] was called PMAC1; currently, the scheme in [21] itself is called PMAC. The construction in [21] is based on an efficient construction of tweakable block cipher (TBC) family; the notion of tweakable block ciphers was introduced in [14]. Chakraborty and Sarkar [7] generalised the TBC construction in [21] 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 [17] to a constant times $md^2/2^n$, where m is the maximum of the lengths of all the queried messages. Nandi and Mandal [19] showed a bound of $(5d\sigma - 3.5d^2)/2^n$ for PMAC.

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

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

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 $\mathit{\Gamma}_{\rho,d}$, i.e., for any $\mathbf{x} = (x_1, \ldots, x_d) \in S^d$,

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

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^d.*

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 of 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 $\mathbf{x} \in \chi_d(S)$, $\mathit{\Gamma}_{\pi,d}(\mathbf{x})$ 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 A Useful Inequality

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

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

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

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

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 $\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}] = \Pr[F(x_1) = y_1, \ldots, F(x_d) = y_d]$ 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 $\lambda(\mathbf{x}) \le \sigma$ and for all $\mathbf{y} \in U$,*

$$\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}] \ge \delta/\#U.$$

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*

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

An immediate consequence of this definition is the following result.

Lemma 2. *Let $F : S \to T$ be a random function and $\mathbf{x} \in \chi_d(S)$. Then*

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

Proof: Let $\mathbf{Y} = (Y_1, \ldots, Y_d) = \mathit{\Gamma}_{F,d}(\mathbf{x}) = (F(x_1), \ldots, F(x_d))$. Then

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

and

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

□

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

Proof: We obtain bounds on $\mathsf{CB}_F(\mathbf{x})$ and then the results follow from Lemma 2. 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 $0 \le k \le 2^n - 2$; 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 $0 \le k_1 < k_2 \le 2^n - 2$; 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 $0 \le k_1 < k_2 \le 2^n - 2$; 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 [13]). Since $0 \le i < j < 2^n - 1$, 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 $\zeta_{k_1,k_2} : T^2 \to T$ 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 [6, 21, 10, 9], though the required properties have not really being brought out as clearly as given in Definition 5. The second strategy has been suggested in [7, 22]. 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 [24, 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 $\Pr[\mathcal{A}^F \to 1]$ 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,

$$\begin{aligned}
\Pr[\mathcal{A}^F \to 1] = & \sum_{(y_1,\ldots,y_d)\in T^d} \Pr[(F(\phi_1^{\mathcal{A}}()) = x_1) \wedge (F(\phi_2^{\mathcal{A}}(y_1)) = x_2) \\
& \qquad \wedge \cdots \wedge (F(\phi_d^{\mathcal{A}}(y_1, \ldots, y_{d-1})) = x_d) \wedge (\phi^{\mathcal{A}}(y_1, \ldots, y_d) = 1)] \\
= & \sum_{(y_1,\ldots,y_d)\in \mathsf{Acc}(\mathcal{A})} \Pr[(F(\phi_1^{\mathcal{A}}()) = x_1) \wedge (F(\phi_2^{\mathcal{A}}(y_1)) = x_2) \\
& \qquad \wedge \cdots \wedge (F(\phi_d^{\mathcal{A}}(y_1, \ldots, y_{d-1})) = x_d)]
\end{aligned}$$

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 [24]) 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 [24, 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,

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

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 $\mathbf{y} \in U$,

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

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

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

Consequently,

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

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,

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

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.

Note. In many situations, it is difficult to directly lower bound an interpolation probability of the form $\Pr[F(x_1) = y_1, \ldots, F(x_d) = y_d]$. Instead, it turns out to be easier to lower bound $\Pr[\mathcal{E}]$ and $\Pr[F(x_1) = y_1, \ldots, F(x_d) = y_d | \mathcal{E}]$, where $\mathcal{E}$ is a suitably chosen event. Usually, $\mathcal{E}$ stands for the event that there are no internal collisions. Suppose that $\Pr[\mathcal{E}] \geq 1 - \varepsilon_{2,1}$ and $\Pr[F(x_1) = y_1, \ldots, F(x_d) = y_d | \mathcal{E}] \geq 1 - \varepsilon_{2,2}$, then $\Pr[F(x_1) = y_1, \ldots, F(x_d) = y_d] \geq 1 - \varepsilon_2$ where $\varepsilon_2 = \varepsilon_{2,1} + \varepsilon_{2,2}$.

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 $F = F_2 \circ F_1$, 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 $F = \rho \circ F_1^{(\rho)}$, 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.*

When ρ is clear from the context, we will write F_1 instead of $F_1^{(\rho)}$. 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 $\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}]$, 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 $Z = F_1(x)$, $Z' = F_1(x')$; $\lambda(x) = m + 1$, $\lambda(x') = m' + 1$; and let $U_1, \ldots, U_m$ and $U'_1, \ldots, U'_m$ 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* $\bigwedge_{i=1}^{m}(Z \neq U_i)$.
2. *Define* $\mathsf{Pairwise\text{-}Disjoint}(x, x')$ *to be the event* $\left(\bigwedge_{i=1}^{m}(Z' \neq U_i) \wedge \bigwedge_{j=1}^{m'}(Z \neq U'_i)\right)$.

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 $x, x' \in S$,*

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

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

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 $Z_i = F_1(x_i)$ and let $U_{i,1}, \ldots, U_{i,m_i}$ be the inputs to ρ in the computation of Z_i. Let $\mathsf{Distinct}(\mathbf{x})$ and $\mathsf{Disjoint}(\mathbf{x})$ be the events

$$\mathsf{Distinct}(\mathbf{x}) = \bigwedge_{1 \leq i < j \leq d} (Z_i \neq Z_j) \tag{14}$$

and

$$\mathsf{Disjoint}(\mathbf{x}) = \bigwedge_{i=1}^{d} \bigwedge_{j=1}^{d} \bigwedge_{k=1}^{m_j} (Z_i \neq U_{j,k}). \tag{15}$$

The event $\mathsf{Distinct}(\mathbf{x})$ is the event $\varGamma_{F_1,d}(\mathbf{x}) \in \chi_d(U)$. Using the fact that F_1 is ε-CR with respect to λ and Lemma 3, we have

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

We have

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

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

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

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

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

The event "$\mathsf{Distinct} \wedge \mathsf{Disjoint}$" means that the random variables $Z_1, \ldots, Z_d$ have distinct values and they are different from any previous inputs to ρ obtained during the computations of $Z_i = F_1(x_i)$. In other words, the event "$\mathsf{Distinct} \wedge \mathsf{Disjoint}$" ensures that the set $\{Z_1, \ldots, Z_d\}$ is a set of d "new" values in the domain of ρ.

If ρ is a uniform random function, then $\mathit{\Gamma}_{\rho,d}(Z_1, \ldots, Z_d)$ is uniformly distributed over T^d. If ρ is a uniform random permutation, then the situation is more complicated. We consider these two cases separately.

1. If ρ is a uniform random function, then for any $\mathbf{y} \in T^d$, $\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct})] = 1/(\#T)^d$ and so,
$$\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}] \geq \frac{1}{\#T^d} \times (1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2)).$$
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. Suppose that ρ is a uniform random permutation of T. Let $V_{i,j} = \rho(U_{i,j})$ and define $\mathcal{V} = \cup_{i=1}^{d} \{V_{i,1}, \ldots, V_{i,m_i}\}$. Fix $\mathbf{y} = (y_1, \ldots, y_d) \in \chi_d(T)$. Let $\mathsf{Allowed}(\mathbf{y})$ be the event $\wedge_{k=1}^{d} (y_k \notin \mathcal{V})$. Since ρ is a uniform random permutation, for any y_k, $\Pr[y_k = V_{i,j}] = 1/\#T$. Note that $\sigma = d + \sum_{i=1}^{d} m_i$.
$$\begin{aligned}\Pr[\mathsf{Allowed}(\mathbf{y})] = \Pr[\bigwedge_{k=1}^{d} (y_k \notin \mathcal{V})] &= 1 - \Pr\left[\bigvee_{k=1}^{d} (y_k \in \mathcal{V})\right] \\ &\geq 1 - \sum_{k=1}^{d} \Pr[y_k \in \mathcal{V}] \\ &= 1 - \sum_{k=1}^{d} \Pr[\bigvee_{i=1}^{d} \bigvee_{j=1}^{m_i} (y_k = V_{i,j})] \\ &\geq 1 - \sum_{k=1}^{d} \sum_{i=1}^{d} \sum_{j=1}^{m_i} \Pr[y_k = V_{i,j}] \\ &= 1 - \frac{d(\sigma - d)}{\#T}.\end{aligned}$$
Let us consider when $\rho(Z_k)$ can be equal to y_k. Suppose that $\mathsf{Allowed}(\mathbf{y})$ does not occur, i.e., there is a y_k which is equal to some $V_{i,j}$. Then $\rho(Z_k) = y_k$ implies that $Z_k = U_{i,j}$. The last event cannot happen if $\mathsf{Disjoint}$ occurs. So,
$$\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct} \wedge \overline{\mathsf{Allowed}(\mathbf{y})})] = 0.$$
On the other hand, if both $(\mathsf{Disjoint} \wedge \mathsf{Distinct})$ and $\mathsf{Allowed}(\mathbf{y})$ occur, then
$$\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct} \wedge \mathsf{Allowed}(\mathbf{y}))] = \frac{1}{p(\#T - (\sigma - d), d)} \geq \frac{1}{(\#T)^d}.$$
Now, for any $\mathbf{y} \in \chi_d(T)$,
$$\begin{aligned}\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct})] &\geq \frac{1}{(\#T)^d} \times \Pr[\mathsf{Allowed}(\mathbf{y})] \\ &\geq \frac{1}{(\#T)^d} \left(1 - \frac{d(\sigma - d)}{\#T}\right).\end{aligned}$$

This gives

$$\begin{aligned}\Pr[\mathit{\Gamma}_{F,d}(\mathbf{x}) = \mathbf{y}] &\geq \frac{1}{(\#T)^d}\left(1 - \frac{d(\sigma - d)}{\#T}\right) \times (1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2)) \\ &\geq \frac{1}{(\#T)^d}\left(1 - \sigma(d\varepsilon + \varepsilon_1 + 2d\varepsilon_2) - \frac{d(\sigma - d)}{\#T}\right).\end{aligned}$$

Again, applying Theorem 1, we have

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

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 $\sigma \geq d$*

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

The advantage of Theorems 2 and 3 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 New 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 [6, 21] 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].

In Figure 1, 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.

We note that the map $x \mapsto \mathsf{Format}(x, l)$ is not an injective map. Non-injectivity arises due to strings of the following type: x is a string of length il (for some $i \geq 1$) ending with 10^j (for some $0 \leq j \leq l-2$) and x' is the prefix of x of length $il - j - 1$. Then $\mathsf{Format}(x, l) = \mathsf{Format}(x', l)$. On the other hand, the function

$$\left(\bigcup_{i\geq 1}\{0,1\}^i \setminus \bigcup_{i\geq 1}\{0,1\}^{il}\right) \overset{\mathsf{Format}}{\longmapsto} \bigcup_{i\geq 1}\{0,1\}^{il}$$

is an injective function. So, the only way in which x and x' map to the same string under Format is when l divides the length of one but not the length of the other. In our construction, we take care of this difference by using suitable masks.

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

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 $x \mathbin{\oplus\!\!\rightarrow} y$ 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.

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 1 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 $1 \leq i \leq m$ and

$$D_i = \begin{cases} \psi^i(R) \mathbin{\oplus\!\!\rightarrow} P_i & 1 \leq i \leq m-1; \\ (C_1 \oplus \cdots \oplus C_{m-1}) \mathbin{\oplus\!\!\rightarrow} P_m & i = m, r = \ell; \\ (C_1 \oplus \cdots \oplus C_{m-1} \oplus \psi^m(R)) \mathbin{\oplus\!\!\rightarrow} P_m & i = m, r < \ell. \end{cases} \qquad (21)$$

Figure 2 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.

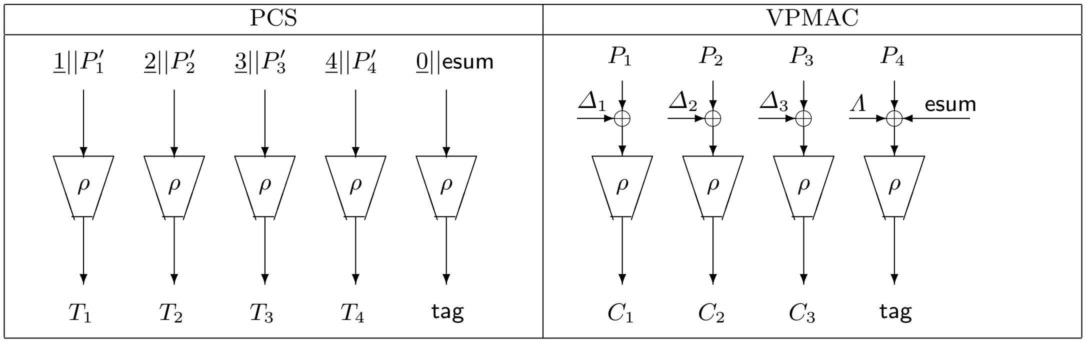

Fig. 2. Tag generation using PCS and VPMAC. In PCS, $\mathsf{esum} = T_1 \oplus T_2 \oplus T_3 \oplus T_4$. In VPMAC, $\Delta_i = \psi^i(R)$; $\Lambda = 0^n$ if the last block is full and $\Lambda = \psi^4(R)$ if the last block has been padded; and $\mathsf{esum} = C_1 \oplus C_2 \oplus C_3$.

Lemma 4. *Let x and x' be distinct messages which are mapped to D_m and $D'_{m'}$ under* VPHASH. *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 ℓ or equal to ℓ.

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,

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

C_{m-1} is the output of the uniform random function ρ on $D_{m-1} = \psi^{m-1}(R) \oplus\!\!\rightarrow P_{m-1}$, where $R = \rho(\mathsf{fStr})$. The probability that $\psi^{m-1}(R) \oplus\!\!\rightarrow P_{m-1}$ equals fStr is $1/\#T$. Using this a simple calculation shows the probability that C_{m-1} equals $\mathsf{bot}(P_m \oplus P'_{m'})$ is at most $2/\#T$.

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 E be the event

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

Then

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

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,

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

We have

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

Consider the event $((D_m \neq D'_{m'})|E)$. 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 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 $r = r' = \ell$, 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 $C_j \oplus Q = \mathsf{bot}(P_m \oplus P'_{m'})$. If $D_j \neq \mathsf{fStr}$, then C_j is uniformly distributed over T. Using this, we have $\Pr[D_m \neq D'_{m'}|E] \geq 1 - 2/\#T$.

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 $r < \ell$, $r' < \ell$: 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 $r = r' = n$. (If $r \neq r'$, then Format ensures that the last blocks are distinct, i.e., $P_m \neq P'_{m'}$. If $P_m = P'_{m'}$ (and so necessarily $r = r'$), then there is an i with $1 \leq i \leq m' - 1$, such that $P_i = P'_i$.)

So suppose $m > m'$. Let 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 E is at least $1 - (m + m' - 2)/\#T$. In a manner similar to the previous case, it can be shown $\Pr[D_m \neq D'_{m'}|E] \geq 1 - 1/\#T$ so that we again have, $\Pr[D_m \neq D'_{m'}] \leq (m + m')/\#T$.

Cases ($r = \ell$, $r' < \ell$) and ($r < \ell$, $r' = \ell$): 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 $1 \le i \le m$ 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 $1 \le i \le m'$, then an analysis similar to the previous case shows the desired result. □

Lemma 5. *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* $\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \le 2m/\#T$.

Combining Lemmas 4, 5 and Theorem 2 with $\varepsilon = \varepsilon_1 = \varepsilon_2 = 2/\#T$, we have the following result.

Theorem 4. *Let d and $\sigma \ge d$ be positive integers. Then*

$$\mathbf{Adv}_{\mathrm{VPMAC}}(d,\sigma) \le \frac{(6d+2)\sigma}{\#T}.$$

4.1 iPMAC

Suppose that in the construction for VPMAC we have $\ell = n$ and $\rho = \pi$ is a uniform random permutation from T to T. Then the construction that we get works with block ciphers. In more details, we 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

$$D_i = \begin{cases} P_i \oplus \psi^i(R) & 1 \le i \le m-1; \\ C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m & i = m, r = n; \\ C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \psi^m(R) & i = m, r < n. \end{cases} \tag{22}$$

Processing of a 4-block message using iPMAC is shown in Figure 3. The same figure also describes the processing of a 4-block message using PMAC. The difference is in the interpretation of Λ.

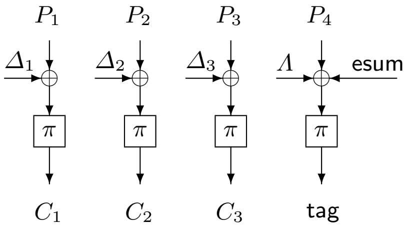

Fig. 3. Tag generation using iPMAC Here $\Delta_i = \psi^i(R)$; $\Lambda = 0^n$ if the last block is full and $\Lambda = \psi^4(R)$ if the last block has been padded; and $\mathsf{esum} = C_1 \oplus C_2 \oplus C_3$.

- 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. For $n = 128$, if the last block is full, then $\Lambda = u^m(u \oplus 1)R \bmod \tau(u)$ and if the last block is partial (and padded), then $\Lambda = (u^2 \oplus u \oplus 1)u^m R \bmod \tau(u)$. The map $m \mapsto u^m \bmod \tau(u)$ is called the powering-up map [21]. 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 $n = 128$ and the choice of $\tau(x)$ to be the first lexicographically first primitive polynomial the value is given in [21]. 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.

There are several advantages to using iPMAC instead of PMAC. For one thing, iPMAC entirely avoids the discrete log computation required during the design stage. As a result, it is possible to simply change $\tau(x)$ to another primitive polynomial and obtain a distinct algorithm. So, iPMAC can be seen as providing an easily reconfigurable family of algorithms.

The second advantage is the possibility of faster masking operations. The description of iPMAC does not depend on the actual choice of ψ. As a result, one can use a tower representation of the underlying field and a word-oriented LFSR to instantiate ψ. In software, this leads to a faster algorithm compared to the powering-up map. See [22] for more details on the comparison between word-oriented LFSR and the powering-up map.

The security of iPMAC is obtained in a manner similar to that of VPMAC. In this case, we have to be careful that π is a permutation rather than a function. The analysis, however, is very similar.

Lemma 6. *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. *Assume that $m + m' - 2 \leq \#T$. Then* $\Pr[D_m = D'_{m'}] \leq (m + m')/\#T \leq 2\max(m, m')/\#T$.

Note. Since $\#T = 2^n$, the assumption $m + m' - 2 \leq \#T$ is mild one.
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

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

Consider the set of random variables.

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

Some of the random variables in these set can be equal. We are interested in a subset of random variables taking equal values only if the number of elements in this subset is odd. Let there be $t \geq 0$ such subsets and $Q_1, \ldots, Q_t$ be random variables where each Q_i is the XOR of the random variables in each subset. 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

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

If $m = m'$, then the terms involving ψ cancel out and the analysis is exactly the same as that for the case $r = r' = n$. (If $r \neq r'$, then Format ensures that the last blocks are distinct, i.e., $P_m \neq P'_{m'}$. If $P_m = P'_{m'}$ (and so necessarily $r = r'$), then there is an i with $1 \leq i \leq m' - 1$, such that $P_i = P'_i$.)

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

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

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, $D_m \oplus D'_{m'} = \psi^m(R)$ 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 7. *Let x and x' be two distinct messages having m and m' blocks respectively. Then*

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

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

Using Theorem 2 with $\varepsilon = \varepsilon_1 = \varepsilon_2 = 2/\#T$ and noting that in this case π is a uniform random permutation gives the following result.

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

$$\mathbf{Adv}_{\mathrm{iPMAC}}(d, \sigma) \leq \frac{(7d + 2)\sigma}{\#T}.$$

Note. A proof for PMAC can be obtained from the above proof with small changes and leads to the same bound.

5 Conclusion

We have analysed pseudorandom functions for use in symmetric key message authentication. Starting from a useful result by Vaudenay [24] and Bernstein [4], we prove a general result on bounding the advantage of a PRF built using a uniform random permutation or a uniform random function.

A new parallelizable construction VPMAC has been described and analysed using this general result. VPMAC uses a keyed compressing function and reduces the number of invocations of the compressing function compared to PCS [4]. The variant where the compressing function is replaced by a block cipher is called iPMAC. This improves upon PMAC [21] by removing the requirement of a design stage discrete log computation.

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.

Acknowledgement

We would like to thank Mridul Nandi for pointing out Lemma 22 of [24] 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. A block-cipher mode of operation for parallelizable message authentication. In Knudsen [12], pages 384–397.
7. 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.
8. Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES – The Advanced Encryption Standard (Information Security and Cryptography).* Springer, Heidelberg, 2002.
9. 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.
10. 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.
11. 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.
12. 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.
13. R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications, revised edition.* Cambridge University Press, 1994.
14. 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.
15. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
16. Ueli M. Maurer. Indistinguishability of random systems. In Knudsen [12], pages 110–132.
17. 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.
18. 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.

19. Mridul Nandi and Avradip Mandal. Improved Security Analysis of PMAC. Cryptology ePrint Archive, Report 2007/031, 2007. `http://eprint.iacr.org/`.
20. Erez Petrank and Charles Rackoff. CBC MAC for Real-Time Data Sources. *J. Cryptology*, 13(3):315–338, 2000.
21. 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.
22. Palash Sarkar. A general mixing strategy for the ECB-Mix-ECB mode of operation. *Inf. Process. Lett.*, 109(2):121–123, 2008.
23. 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.
24. Serge Vaudenay. Decorrelation: A theory for block cipher security. *J. Cryptology*, 16(4):249–286, 2003.