Pseudo-Random Functions and Parallelizable Modes of Operations of a Block Cipher

Palash Sarkar

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

Abstract. This paper considers the construction and analysis of pseudo-random functions (PRFs) with specific reference to modes of operations of a block cipher. In the context of message authentication codes (MACs), earlier independent work by Bernstein and Vaudenay show how to reduce the analysis of relevant PRFs to some probability calculations. In the first part of the paper, we revisit this result and use it to prove a general result on constructions which use a PRF with a "small" domain to build a PRF with a "large" domain. This result is used to analyse two new parallelizable PRFs which are suitable for use as MAC schemes. The first scheme, called iPMAC, is based on a block cipher and improves upon the well-known PMAC algorithm. The improvements consist in faster masking operations and the removal of a design stage discrete logarithm computation. The second scheme, called VPMAC, uses a keyed compression function rather than a block cipher. The only previously known compression function based parallelizable PRF is called the protected counter sum (PCS) and is due to Bernstein. VPMAC improves upon PCS by requiring lesser number of calls to the compression function.
The second part of the paper takes a new look at the construction and analysis of modes of operations for authenticated encryption (AE) and for authenticated encryption with associated data (AEAD). Usually, the most complicated part in the security analysis of such modes is the analysis of authentication security. Previous work by Liskov, Rivest and Wagner and later Rogaway had suggested that this analysis is simplified by using a primitive called a tweakable block cipher (TBC). In contrast, we take a direct approach. We prove a general result which shows that the authentication security of an AE scheme can be proved from the privacy of the scheme and by showing a certain associated function to be a PRF. Two new AE schemes PAE and PAE-1 are described and analysed using this approach. In particular, it is shown that the authentication security of PAE follows easily from the security of iPMAC. As a result, no separate extensive analysis of the authentication security of PAE is required.
An AEAD scheme can be obtained by combining an AE scheme and an authentication scheme and it has been suggested earlier that a TBC based approach simplifies the analysis. Again, in contrast to the TBC based approach, we take a direct approach based on a simple masking strategy. Our idea uses double encryption of a fixed string and achieves the same effect of mask separation as in the TBC based approach. Using this idea, two new AEAD schemes PAEAD and PAEAD-1 are described. An important application of AEAD schemes is in the encryption of IP packets. The new schemes offer certain advantages over previously well known schemes such as the offset codebook (OCB) mode. These improvements include providing a wider variety of easily reconfigurable family of schemes, a small speed-up, a smaller size decryption algorithm for hardware implementation and uniform processing of only full-block messages.
Keywords: pseudo-random function, message authentication, protected counter sum, PMAC, encryption, authentication, authenticated encryption, authenticated encryption with associated data, OCB.

1 Introduction

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 $K \in \mathcal{K}$, $E_K : \mathcal{M} \rightarrow \mathcal{M}$, defined as

$E_K(\cdot) \stackrel{\Delta}{=} E(K, \cdot)$, is a bijection 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 and AES [10].

The security model of a block cipher is that of a pseudo-random permutation (PRP) family [22]. 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 run-time of $\mathcal{A}$. A stronger notion is that of strong pseudo-random permutation (SPRP), where $\mathcal{A}$ is also provided the inverse oracle.

A related general notion is that of a pseudo-random function (PRF). Let f be a random (but, not necessarily uniform random) function from a set $\mathcal{X}$ to a set $\mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are finite non-empty sets. The set $\mathcal{X}$ is usually the set of all binary strings of some maximum length. Considering $\mathcal{X}$ to be the set of *all* binary causes a technical difficulty. We will be interested in (uniform) random functions from $\mathcal{X}$ to $\mathcal{Y}$. If $\mathcal{X}$ is finite, then the set of all such functions is also finite and we are dealing with a discrete probability space. If, on the other hand, $\mathcal{X}$ is the set of all binary strings, then the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ is uncountable which leads us into non-discrete probability spaces. To avoid this issue and also because of the fact that in any conceivable situation, there will be a practical upper bound on the length of a string in $\mathcal{X}$ (such as 2^{64}), we consider $\mathcal{X}$ to be a finite set.

Let $\mathcal{A}$ be an adversary which has an oracle. The oracle takes as input an element of $\mathcal{X}$ and returns as output an element of $\mathcal{Y}$. As before, instantiate the oracle in two ways; either with f or with a uniform random function from $\mathcal{X}$ to $\mathcal{Y}$ 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 $|p_1 - p_0|$. The PRF-advantage is parametrized by the number of queries made and the run-time of the adversary.

A block cipher by itself can handle 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. In this work, we consider modes of operations of a block cipher for the tasks of authentication and authenticated encryption with the option of authenticating an additional associated data.

Authentication. 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.

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.

Intuitively, given a PRF f which outputs n-bit strings, it is easy to construct a basic authentication scheme: given a message x, the tag is $\mathsf{tag} = f(x)$ and verification is done by regenerating the tag. There are efficient known constructions of PRFs including one [15] which has been standardized by NIST [11]. This is a sequential algorithm which is based on the cipher block chaining (CBC) mode of operation.

Security of CBC-MAC has been analysed in several papers [2, 25, 39, 31]. An improved analysis by Bellare, Pietrzak and Rogaway [4] showed a bound of $\frac{mq^2}{2^n}(12 + \frac{8m^3}{2^n})$ 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 $mq^2/2^n$, which improves upon the previous bound of $m^2q^2/2^n$. In fact, the importance of the work in [4] is that it was the first paper to prove a bound of the type $mq^2/2^n$ for some PRF construction. Such bounds are usually called "beyond birthday bound" security. More recent work on CBC-type construction appears in [29].

A parallel MAC scheme called protected counter sum (PCS) was described by Bernstein [6]. This scheme uses a keyed compression function as its building block and cannot be replaced by a block cipher. Black and Rogaway [8] and later Rogaway [35] described block cipher based methods for parallel message authentication. The scheme in [8] was called PMAC and the one in [35] was called PMAC1; currently, the scheme in [35] itself is called PMAC. The construction in [35] is based on an efficient construction of tweakable block cipher (TBC) family [21]. Chakraborty and Sarkar [9] generalised the TBC construction in [35] 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 [4], this bound was improved by Minematsu and Matsushima [27] to a constant times $mq^2/2^n$, where m is the maximum of the lengths of all the queried messages. Nandi and Mandal [30] showed a bound of $(5q\sigma - 3.5q^2)/2^n$ for PMAC. Bounds of the type $q\sigma/2^n$ are also called beyond birthday bound.

Authenticated encryption with associated data. Packet switched networks communicate among various nodes using packets. Examples are the internet protocol version 4 (IPv4) which is widely deployed and the later IPv6. A typical packet structure consists of two sections – a header section and a data section. In IPv4, the header (without options) consists of 20 bytes, while the data section could be up to 65,535 bytes. The maximum transmission unit (MTU) could be smaller, for example, the MTU of ethernet is 1500 bytes. If 20 bytes are used for header, then the MTU for data comes to 1480 bytes, so that a 65,535-byte IPv4 packet will be broken up into $\lceil 65535/1480 \rceil = 45$ ethernet packets. More details can be found from [14].

The problem that concerns us is the need for secure communication of such packets. The data needs to be encrypted so that it becomes inaccessible to unauthorized persons. But, this is not enough. It is also required to ensure that any tampering of the encrypted data can be detected. This is called authentication and the security requirement on the data is both encryption and authentication. The header, however, cannot be encrypted as then the packet cannot be forwarded by intermediate routers. On the other hand, it is usually required to ensure that the header is authenticated, so that any tampering of the header can be detected by the receiver.

Abstracting to a more general setting, the requirement of both encrypting and authenticating a binary string is called authenticated encryption (AE). If additionally another string needs to be authenticated but not encrypted, then the problem is called authenticated encryption with associated data (AEAD). The basic idea of AE was known to designers earlier but, the formal security model of AE schemes was proposed in [3, 18]. The formal security security model for AEAD schemes was introduced in [34].

Intuitively, authenticated encryption can be achieved by making two passes over the data – the first pass encrypts the data, while the second pass authenticates it. Somewhat surprisingly, it can be shown that a single pass over the data is sufficient to achieve both encryption and authentication. This results in significant gain in efficiency compared to two-pass schemes. The first single-pass scheme was reported by Jutla [16] and independently by Gligor and Donescu [12]. Later work [36, 35, 32, 33] led to an efficient algorithm called OCB. (There are actually two versions of OCB and the version in [32, 33] is currently called OCB and is the one that we will actually refer to.) The work [35, 32] also shows how to efficiently handle associated data by combining with the PMAC algorithm. So, OCB currently refers to an AEAD scheme. This OCB version is based upon TBCs presented in [32]. This class of TBCs was generalised in [9] and provides a family of AEAD schemes of which OCB is a special case.

The currently known single-pass algorithms are covered by intellectual property claims. Somewhat unfortunately, this has resulted in the scientifically backward step of publishing two-pass algorithms [5, 23, 26] after one-pass algorithms had already become known. A two-pass algorithm has been adopted as a standard [1] by the NIST of USA.

1.1 Our Contributions

We make several contributions to the design and analysis of PRFs in the context of modes of operations of a block cipher for the tasks of authentication and authenticated encryption with associated data.

Analysis of fixed output length PRFs. A useful result for upper bounding PRF-advantage was proved by Bernstein [6] and Vaudenay [40]. Let $\mathcal{Y} = \{0,1\}^n$ and $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a random function and $\mathcal{U}$ is a "large" subset of $\mathcal{Y}^q$ for some positive integer q. For distinct $x_1, \ldots, x_q \in \mathcal{X}$ and $(y_1, \ldots, y_q) \in U$, $\Pr[f(x_1) = y_1, \ldots, f(x_q) = y_q]$ is called a q-interpolation probability [6]. In [40], it has been proved that if the "interpolation probabilities" of f can be lower bounded on a "large" subset of $\mathcal{Y}^q$, then the advantage of f as a PRF can be upper bounded. The special case where the subset $\mathcal{U}$ equals $\mathcal{Y}^q$ has been proved in [6]. We slightly modify this result so as to include a length function λ on $\mathcal{X}$. In applications, for $x \in \mathcal{X}$, $\lambda(x)$ 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. A general class of constructions considered in [17, 28] uses a directed acyclic graph (DAG) as the underlying combinatorial structure. The class of functions considered here cover the class of DAG-based constructions.

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 (Self-Disjoint), i.e., $\wedge_{i=1}^{m}(Z \neq U_i)$; and pairwise-disjoint (Pairwise-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, $\overline{\text{Self-Disjoint}}$ and $\overline{\text{Pairwise-Disjoint}}$ are all small, then 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 probabilities of certain events for f_1.

Efficient masking functions. Many block cipher based modes of operations use masks. These masks are XORed to inputs before applying the block cipher. The number of masks required is approximately equal to the number of block cipher invocations. Masks are usually generated in sequence and require considerably less time compared to a block cipher invocation. Nonetheless, the generation of the masks can require about 3 to 5% of the total time to process a message.

Efficient mask generation technique uses finite field arithmetic. Let $\tau(x)$ be a primitive polynomial of degree n over $\mathbb{F}_2 = GF(2)$ and let $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/(\tau(x))$. For an element $\beta(x) \in \mathbb{F}_{2^n}$, the map $i \mapsto x^i\beta(x) \bmod \tau(x)$ is the so-called powering-up map used in [35] and other papers [8, 15, 11, 41].

We define a general notion of masking function and identify certain properties that are required to prove the correctness of the constructions given in this paper. (These same properties are also required in the other works, though, they have not been identified as such.) Suitable masking functions can be constructed from a linear map $\psi : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ whose minimal polynomial over $\mathbb{F}_2$ is primitive. The powering map can be seen as one particular example of this general formulation. Using a tower field representation of $\mathbb{F}_{2^n}$, it is possible to obtain more efficient masking functions. These correspond to what are called word oriented linear feedback shift registers (LFSRs).

We provide specific examples of word oriented LFSRs for different values of n. Software implementation of the powering up method and word oriented LFSRs show that the latter is about two times faster. The new modes of operations that we design use masking based on word oriented LFSRs. As a result, the constructions are faster than those that use the powering-up method. In particular, the new authentication scheme iPMAC is faster than PMAC and the new AEAD schemes PAEAD and PAEAD-1 are faster than OCB. We note, however, that with the currently reported best speeds of AES-128 [24, 7], the speed improvements are small (about 1 to 2%). But, these improvements are achieved at no additional trade-offs and since the algorithms are likely to be heavily used, even small speed improvements are worthwhile. Further, Intel plans to incorporate AES-128 instructions as processor instructions [13], which will significantly increase the speed of AES-128 encryption and decryption. In this eventuality, the proportional speed-up of using word-oriented LFSRs may be about 5% compared to using the powering-up map.

Avoiding the tweakable block cipher approach. Building on the work of Liskov, Rivest and Wagner [21], it has been argued Rogaway [35] that the notion of tweakable block ciphers simplify the construction and analysis of modes of operations of block ciphers. While this is true to a certain extent, it comes with a certain drawback. The construction of PMAC and OCB in [35] uses the notion of TBCs. This approach requires the computation of certain discrete logarithms in the design stage. For PMAC and OCB, it is required to compute the discrete logarithm of $(x \oplus 1)$ and $(x^2 \oplus x \oplus 1)$ in the field represented by the primitive polynomial $\tau(x)$ of degree n over $\mathbb{F}_2$. For $n = 64$ and 128, these values are given in [35] for a specific $\tau(x)$; for $n = 256$, computing these will take a few hours and is not convenient.

Apart from the inconvenience, this affects the reconfigurability of the design. If the specific $\tau(x)$ given in [35] is desired to be changed, then the discrete logarithms must be recomputed. A similar difficulty is faced in [41], where the tweak based approach is used for $n = 512$. Since computing discrete logarithms in $\mathbb{F}_{2^{512}}$ is very difficult, a bypass is adopted by using tweaks from different subgroups. It should be noted that it is possible to avoid discrete logarithm computations while keeping within the ambit of tweakable block cipher approach [9]. However, then the masking operations become slower. The task of avoiding the discrete logarithm computations *and* achieving faster masking compared to PMAC and OCB has not been achieved earlier.

We completely avoid the tweakable block cipher approach. Our construction of iPMAC uses a simple masking technique which does not require the notion of tweaks. A mask is used to distinguish whether the last block has been padded or not. PMAC also does the same, but, the masks are built using the tweak-based approach. In contrast, our approach is direct.

An AE scheme takes as input a pair (N, P) and produces as output (C, tag). Here N is a nonce and tag authenticates the message P. Further, given N and C, P is uniquely determined. Showing that an AE scheme achieves privacy is usually quite easy. In previous works on AE schemes [16, 12, 36], the main difficulty had been to show that the scheme achieves secure authentication. The TBC based approach in [35] simplified this task.

We prove a general result for AE schemes which states that if the scheme satisfies privacy and if the associated function taking (N, C) to tag is a PRF, then the scheme achieves authentication. This result greatly simplifies the analysis of authentication of AE schemes. In particular, this result is used to show the authentication security of two new AE schemes PAE and PAE-1. For PAE, we show that the authentication security follows from the PRF-property of a simple modification of iPMAC. Having already proved that iPMAC is a PRF, the authentication security of PAE is obtained as a simple corollary. In contrast, the TBC based approach in [35] requires separate proofs for the authentication security of OCB and the PRF-property of PMAC.

A consequence of the relation between the authentication security of PAE and (the modification of) iPMAC is to provide a simple answer to a question attributed to Rivest in [34] as to whether it is possible to obtain an authentication scheme from an AE scheme. The decryption algorithm for PAE can be used for this purpose. The essential idea is to consider the message to be authenticated to be a ciphertext and run the decryption algorithm of PAE on it, returning the tag that is generated.

Our approach to construction of AEAD schemes also avoids the tweakable approach. But, at a higher level of abstraction, it uses the same idea as that in [35]. The basic idea is to combine an AE and an authentication scheme: OCB and PMAC are combined in [35], whereas we combine the new constructions iPMAC and PAE or PAE-1. The combination is secure if it can be ensured that the set of inputs to the block cipher in the AE part is disjoint from the set of inputs to the block cipher in the authentication part. In [35], suitable disjoint tweak spaces are used to ensure this. In contrast, we use a simple mask separation strategy to ensure that with high probability any mask used in the PAE part is distinct from a mask used in the iPMAC part.

One consequence of not requiring the tweak based approach is to avoid the the discrete log computations required in [35]. As a consequence, we obtain a *family of easily reconfigurable modes of operations*. In our approach, the masks are generated using a tower field representation of $\mathbb{F}_{2^n}$. Simply changing this field representation provides a different construction with the same security and efficiency.

Constructions of parallelizable MAC schemes. We describe iPMAC which is a new parallelizable construction using a uniform random permutation of $\{0,1\}^n$. An extension called VPMAC

is also described. VPMAC uses a uniform random function which maps ℓ bits to n bits with $\ell \geq n$. Various features of the new and earlier parallelizable constructions are shown in Table 1.

Table 1. Features of the new and previous parallelizable constructions of PRFs suitable for MAC schemes.

scheme	perm?	bound	# invoc
PMAC [35]	yes	$\frac{5q\sigma - 3.5q^2}{2^n}$ [30]	$1 + \lceil \mathsf{len}(x)/n \rceil$
PCS [6]	no	$\frac{q(q-1)}{2^{n+1}}$ [6]	$1 + \lceil \mathsf{len}(x)/n \rceil$
iPMAC	yes	$\frac{(7q+2)\sigma}{2^n}$	$1 + \lceil \mathsf{len}(x)/n \rceil$
VPMAC	no	$\frac{(6q+2)\sigma}{2^n}$	$1 + \lceil \mathsf{len}(x)/\ell \rceil$

The PRF-bound that we obtain for iPMAC is similar to that obtained in [27, 30] for PMAC. As already mentioned, the advantages of iPMAC over PMAC are the faster masking operations and the removal of design stage discrete logarithm computations.

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 certain collisions do not happen at all, whereas in VPMAC these collisions are only probabilistically ruled out. Yasuda [41] describes a sequential construction of a PRF using a uniform random ρ with the restriction that $\ell \geq 2n$. The number of invocations required to process a message x is approximately $\mathsf{len}(x)/n$ which is the same as that of PCS and so is slower than VPMAC. The security bound on the other hand, is of the form $mq^2/2^{2n+1}$ which is better than that of PCS and VPMAC.

Used with a uniform random permutation $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$, bounds of the type $q\sigma/2^n$ or $mq^2/2^n$ are called beyond birthday bound security. The "beyond" refers to the numerator of the expression which is of the type $q\sigma$ or mq^2 instead of σ^2 or q^2m^2 which would be the so-called birthday bound security. If, however, the construction uses a uniform random function $\rho : \{0,1\}^\ell \rightarrow \{0,1\}^n$ with $\ell > n$, then there is scope for some ambiguity as to what may be called "beyond birthday bound". This specifically refers to what the denominator should be. Since the size of the tag is still n bits, one may say that bounds of the type $q\sigma/2^n$ or $mq^2/2^n$ are still beyond the birthday bound. Viewed in this manner, both PCS and VPMAC provide bounds of this type. On the other hand, a bound of the type $mq^2/2^{2n+1}$ obtained in [41] is even better. It must be noted though, that such a bound is obtained at the cost of efficiency, i.e., $\mathsf{len}(x)/n$ invocations of ρ compared to $\mathsf{len}(x)/\ell$ invocations of ρ as required in VPMAC. So [41] achieves better bound at the cost of efficiency. In absolute terms, the security bound for VPMAC is good enough for practical purposes and is comparable to the bounds obtained for PMAC. We believe that VPMAC strikes a good balance between security and efficiency. Further, VPMAC is a parallelizable scheme using a compression function, the construction of which was posed as an open problem in [41].

Parallelizable schemes for authenticated encryption with associated data. The two new AE schemes show different ways of achieving authentication. Their combination with iPMAC provides the two new AEAD schemes. These new AE and AEAD schemes offer certain advantages over OCB and also over other previous (single-pass) modes [16, 12].

1. As mentioned earlier, faster masking operations and avoiding design stage discrete logarithm computations are two issues.

2. For PAE and PAEAD, encryption requires both E_K and E_K^{-1} while decryption requires only E_K^{-1}; for PAE-1 and PAEAD-1, encryption requires only E_K while decryption requires both E_K and E_K^{-1}. In this aspect, OCB is similar to the second strategy. Consider an application, where a central office encrypts a message containing a set of instructions and sends them to mobile sales-persons who only need to decrypt and follow the instructions. Such a salesperson does not need to encrypt any message. Further, suppose that the mobile clients are implemented in smart cards which have limited resources. It is then advantageous to have a scheme whose decryption algorithm requires lesser hardware space for implementation. For such an application, PAEAD is preferrable to OCB, since the decryption algorithm of PAEAD requires only E_K^{-1} to be implemented, while, the decryption algorithm of OCB requires both E_K and E_K^{-1}. If, on the other hand, some application requires the opposite, i.e., small hardware for encryption, then PAEAD-1 (and OCB) would be preferrable to PAEAD. Thus, compared to OCB, the new constructions provide a wider flexibility in designing applications.
3. For messages whose lengths are multiples of n, the new schemes do not distinguish between intermediate blocks and the last block. All blocks are processed in the same manner. The distinction arises only when the last block is less than n bits long. OCB, on the other hand, processes the last block in a different manner irrespective of its length. For most applications, the lengths of the messages are powers of 2 while the block length n is also a lower power of 2 and so the block length divides the message lengths. For such applications, since all blocks are processed in exactly the same manner, the implementation, especially in hardware, of the new schemes will be simpler compared to OCB.

Note. In this paper, "random" does *not* necessarily mean "uniform random". When required, we will explicitly mention the uniformity condition.

2 Basic Definitions and Results

We will be studying functions from a finite non-empty set $\mathcal{X}$ to a finite non-empty set $\mathcal{Y}$. For example, $\mathcal{X}$ could be the set of all binary strings of lengths between 0 and 2^{64} and $\mathcal{Y}$ could be the set of all binary strings of length 128. Given a function $f : \mathcal{X} \to \mathcal{Y}$ and binary strings $\mathsf{str}_1, \ldots, \mathsf{str}_k$, we will often write $f(\mathsf{str}_1, \ldots, \mathsf{str}_k)$ to denote $f(\mathsf{str}_1 || \ldots || \mathsf{str}_k)$. Define $\chi_q(\mathcal{X})$ to be

$$\chi_q(\mathcal{X}) = \{(x_1, \ldots, x_q) \in \mathcal{X}^q : x_i \neq x_j, 1 \leq i < j \leq q\}. \tag{1}$$

In other words, $\chi_q(\mathcal{X})$ consists of all $(x_1, \ldots, x_q)$ such that $x_1, \ldots, x_q$ are distinct elements of $\mathcal{X}$.

Let ρ be a function from $\mathcal{X}$ to $\mathcal{Y}$ and q be a positive integer. The natural extension of ρ to a function from $\mathcal{X}^q$ to $\mathcal{Y}^q$ obtained by applying ρ to each component will be denoted by ECB_ρ, i.e., for any $\mathbf{x} = (x_1, \ldots, x_q) \in \mathcal{X}^q$,

$$\mathrm{ECB}_\rho(\mathbf{x}) = \mathrm{ECB}_\rho(x_1, \ldots, x_q) = (\rho(x_1), \ldots, \rho(x_q)). \tag{2}$$

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

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

Let λ be a function from $\mathcal{X}$ to non-negative integers, i.e., we associate a non-negative integer with each element of $\mathcal{X}$. In our applications, the set $\mathcal{X}$ will consist of binary strings and for $x \in \mathcal{X}$, $\lambda(x)$ 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 $\mathcal{X}$. Given $\mathbf{x} = (x_1, \ldots, x_q) \in \mathcal{X}^q$, we define $\lambda(\mathbf{x}) = \sum_{i=1}^{q} \lambda(x_i)$.

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

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

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

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

Corollary 1. *For a finite nonempty set $\mathcal{X}$, $\#\chi_q(\mathcal{X}) = p(\#\mathcal{X}, q) \geq \left(1 - \dfrac{q(q-1)}{2\#\mathcal{X}}\right)(\#\mathcal{X})^q$. Consequently, $\chi_q(\mathcal{X})$ is a $\left(1 - \frac{q(q-1)}{2\#\mathcal{X}}\right)$-large subset of $\mathcal{X}^q$.*

Also, we will require the following result.

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

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

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

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

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

Our main object of study are random functions from $\mathcal{X}$ to $\mathcal{Y}$. Let $\mathcal{Y}^{\mathcal{X}}$ denote the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. By a uniform random function ρ from $\mathcal{X}$ to $\mathcal{Y}$ we will mean an element of $\mathcal{Y}^{\mathcal{X}}$ chosen uniformly at random. A more convenient way to view ρ is the following. For any $\mathbf{x} \in \chi_q(\mathcal{X})$, $\mathrm{ECB}_\rho(\mathbf{x})$ is uniformly distributed over $\mathcal{Y}^q$, i.e., in other words, the outputs of ρ on distinct inputs are independent and uniformly distributed. If $\mathcal{X} = \mathcal{Y}$, then we can talk about a permutation π of $\mathcal{Y}$, which is a bijection $\pi : \mathcal{Y} \to \mathcal{Y}$. By a uniform random permutation, we will mean a permutation chosen uniformly at random from the set of all permutations of $\mathcal{Y}$. Again, this means that for any $\mathbf{x} \in \chi_q(\mathcal{X})$, $\mathrm{ECB}_\pi(\mathbf{x})$ is uniformly distributed over $\chi_q(\mathcal{Y})$. Examples of random (but not uniform random) functions can be obtained: let $\mathcal{Y}$ be a finite field and $\mathcal{X} = \mathcal{Y}^2$; choose a uniform random $\alpha \in \mathcal{Y}$ and define $\rho : \mathcal{X} \to \mathcal{Y}$ 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 Information Theoretic Versus Computational Security

Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be an n-bit block cipher. Let π be a uniform random permutation of $\{0,1\}^n$. For practical applications, the encryption and authentication functions will be constructed using a block cipher E_K. On the other hand, following the information theoretic approach, we will analyse such constructions by replacing E_K with π. Consequently, the constructions below will be described in terms of a uniform random permutation π. These are easily translated into descriptions using E_K by simply replacing the occurrences of π by E_K (and the occurrences of π^{-1} by E_K^{-1}). The consequent effect on security will be an additive degradation of security by an amount which is equal to the computational advantage of an adversary in distinguishing E_K from π (or of distinguishing (E_K, E_K^{-1}) from (π, π^{-1})).

It is to be noted that the randomness of E_K comes from the uniform random choice of K while π is a uniform random permutation of $\{0,1\}^n$. This difference of randomness can be easily detected by a computationally unbounded adversary. Hence, to obtain a meaningful notion in this case, we need to bound the computational power of an adversary in distinguishing between E_K and π. A block cipher $E_K()$ is said to be a pseudo-random permutation (PRP) if a computationally bounded adversary has negligible advantage in distinguishing E_K from π; $E_K()$ is said to be a *strong* pseudo-random permutation (SPRP) if a computationally bounded adversary has negligible advantage in distinguishing (E_K, E_K^{-1}) from (π, π^{-1})). See [35] for details.

For the information theoretic analysis, we will consider computationally unbounded adversaries and consequently, without loss of generality, we consider such adversaries to be deterministic algorithms. (This approach has been used earlier [40, 6].)

Many practical modes of operations of block ciphers are analysed 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 run-time 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 or as an SPRP.

Suppose that instead of a block cipher, a keyed compressing function is used to construct the PRF [6]. A similar two-step approach 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 above. 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.

2.2 Notation

We provide below some of the notation that will be frequently used: n will denote the block size of the underlying uniform random permutation (or the block cipher) and q will denote the number of queries made by an adversary. Given a binary string X, the notation $\mathsf{First}_r(X)$ denotes the r bits of X from the left, i.e., the r most significant bits of X.

String parsing is done as shown in Table 2. Given a string (message) X of length $\mathsf{len}(X) \geq 0$ bits, formatting into l-bit blocks with $l \geq 1$, is done by calling $\mathsf{Format}(X, l)$. This defines the values of m and r and returns $X_1, \ldots, X_m$. If $r = l$, then the last block is a full block and if $r < l$, then the last block is a partial block which has been padded to obtain a full block.

Table 2. Padding and formatting of a string X. This also defines the values of m and r from $\mathsf{len}(X)$ and l.

$\mathsf{Format}(X, l)$
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 $X_1, \ldots, X_m$ each of length n;
return $(X_1, \ldots, X_m)$.

Note that the map $X \mapsto \mathsf{Format}(X, l)$ for $l > 1$ 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' may 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.

Later we will consider an adversary making several multi-block queries. Quantities corresponding to the s-th query are denoted by the superscript $^{(s)}$; for example, length of the s-th query is $\ell^{(s)}$; number of blocks is $m^{(s)}$; length of the last block is $r^{(s)}$; nonce is $N^{(s)}$; message blocks are $P_1^{(s)}, \ldots, P_{m^{(s)}-1}, P_{m^{(s)}}$; ciphertext blocks are $C_1^{(s)}, \ldots, C_{m^{(s)}-1}, C_{m^{(s)}}$; tag is $\mathsf{tag}^{(s)}$. Similarly for the internal variables.

2.3 Interpolation and Collision Probabilities

Let $\mathcal{X}$ and $\mathcal{Y}$ be sets and f be a random function from $\mathcal{X}$ to $\mathcal{Y}$. For $\mathbf{x} \in \chi_q(\mathcal{X})$ and $\mathbf{y} \in \mathcal{Y}^q$, the probability $\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] = \Pr[f(x_1) = y_1, \ldots, f(x_q) = y_q]$ has been called a q-interpolation probability in [6].

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

$$\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] \geq \delta / \#U.$$

Here δ could possibly depend on q 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 $\mathcal{X}$.*

1. *Let $x \neq x'$ be elements of $\mathcal{X}$. 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 $\mathbf{x} \in \chi_q(\mathcal{X})$, we define the collision bound $\mathsf{CB}_f(\mathbf{x})$ to be*

$$\mathsf{CB}_f(\mathbf{x}) = \sum_{1 \leq i < j \leq q} \Pr[f(x_i) = f(x_j)].$$

An immediate consequence of this definition is the following result.

Lemma 2. *Let $f : \mathcal{X} \to \mathcal{Y}$ be a random function and $\mathbf{x} \in \chi_q(\mathcal{X})$. Then*

$$\Pr[\mathrm{ECB}_f(\mathbf{x}) \in \chi_q(\mathcal{Y})] \geq 1 - \mathsf{CB}_f(\mathbf{x}). \tag{4}$$

Proof. Let $\mathbf{y} = (y_1, \ldots, y_q) = \mathrm{ECB}_f(\mathbf{x}) = (f(x_1), \ldots, f(x_q))$. Then

$$\Pr\left[\bigvee_{1 \leq i < j \leq q} (y_i = y_j)\right] \leq \sum_{1 \leq i < j \leq q} \Pr\left[(y_i = y_j)\right] = \mathsf{CB}_f(\mathbf{x})$$

and

$$\Pr\left[\mathbf{y} \in \chi_q(\mathcal{Y})\right] = \Pr\left[\bigwedge_{1 \leq i < j \leq q} (y_i \neq y_j)\right] = 1 - \Pr\left[\bigvee_{1 \leq i < j \leq q} (y_i = y_j)\right] \geq 1 - \mathsf{CB}_f(\mathbf{x}).$$

□

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 $\mathcal{X}$ and λ be a length function on $\mathcal{X}$.*

1. *f is said to be ε-CR, if for any two distinct $x, x' \in \mathcal{X}$, $\Pr[\mathsf{Coll}_f(x, x')] \leq \varepsilon$, for some constant ε.*
2. *f is said to be ε-CR with respect to λ, if for any two distinct $x, x' \in \mathcal{X}$, $\Pr[\mathsf{Coll}_f(x, x')] \leq \varepsilon \times \max(\lambda(x_1), \lambda(x_2))$, 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 q and $\sigma \geq q$ be positive integers; and $f : \mathcal{X} \to \mathcal{Y}$ be a random function and λ be a length function on $\mathcal{X}$. Let $\mathbf{x} \in \chi_q(\mathcal{X})$ and $\sigma = \lambda(\mathbf{x})$.*

1. *If f is ε-CR, then $\Pr[\mathrm{ECB}_f(\mathbf{x}) \in \chi_q(\mathcal{Y})] \geq \left(1 - \frac{q(q-1)\varepsilon}{2}\right)$.*
2. *If f is ε-CR with respect to λ, then $\Pr[\mathrm{ECB}_f(\mathbf{x}) \in \chi_q(\mathcal{Y})] \geq (1 - \varepsilon q \sigma)$.*

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 $\mathsf{CB}_f(\mathbf{x}) \leq (q(q-1)\varepsilon)/2$. For the second case, we have

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

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 $\mathbb{F}$ be a finite field and f_α be a random function mapping $\mathbb{F}^2$ to $\mathbb{F}$ by $(a_0, a_1) \mapsto a_0 + \alpha a_1$, where α is a uniform random element of $\mathbb{F}$. 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 $q > 2$ cannot be lower bounded.

2.4 Adversarial Model

In the information theoretic approach, there is no bound on the computation time of an adversary. So, as mentioned earlier, we consider the adversary to be a deterministic algorithm. The adversary interacts with an oracle and outputs a bit. The oracle takes as input an element of a set $\mathcal{X}$ and produces as output an element of a finite non-empty set $\mathcal{Y}$. The adversary $\mathcal{A}$ makes q 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 $\Phi_1, \Phi_2, \ldots, \Phi_q$ and another function Φ. The function $\Phi_1()$ does not take any input and produces $x_1 \in \mathcal{X}$ 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, $\mathbf{x} = (x_1, \ldots, x_q) \in \chi_q(\mathcal{X})$.

Finally, the function Φ takes as input $(y_1, \ldots, y_q)$ and produces as output a bit, which is taken to be the output of $\mathcal{A}$. Note that the functions $\Phi_1, \ldots, \Phi_q$ and Φ do not depend on the oracle. We will use the notation $\Phi_1^{\mathcal{A}}, \Phi_2^{\mathcal{A}}, \ldots, \Phi_q^{\mathcal{A}}$ 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,

$$\begin{aligned}\Pr[\mathcal{A}^f \to 1] = \sum_{(y_1,\ldots,y_q)\in\mathcal{Y}^q} & \Pr[(f(\Phi_1^{\mathcal{A}}()) = y_1) \wedge (f(\Phi_2^{\mathcal{A}}(y_1)) = y_2) \\ & \qquad \wedge \cdots \wedge (f(\Phi_q^{\mathcal{A}}(y_1, \ldots, y_{q-1})) = y_q) \wedge (\Phi^{\mathcal{A}}(y_1, \ldots, y_q) = 1)] \\ = \sum_{(y_1,\ldots,y_q)\in\mathsf{Acc}(\mathcal{A})} & \Pr[(f(\Phi_1^{\mathcal{A}}()) = y_1) \wedge (f(\Phi_2^{\mathcal{A}}(y_1)) = y_2) \\ & \qquad \wedge \cdots \wedge (f(\Phi_q^{\mathcal{A}}(y_1, \ldots, y_{q-1})) = y_q)]\end{aligned}$$

where

$$\mathsf{Acc}(\mathcal{A}) = \{(y_1, \ldots, y_q) : \Phi^{\mathcal{A}}(y_1, \ldots, y_q) = 1\}. \tag{5}$$

The set $\mathsf{Acc}(\mathcal{A})$ is the set of $(y_1, \ldots, y_q)$ 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 $\mathcal{X}$ to $\mathcal{Y}$. Then the advantage of $\mathcal{A}$ in distinguishing between f and g is defined to be

$$\mathbf{Adv}_{\mathcal{A},(f,g)} = \Pr[\mathcal{A}^f \to 1] - \Pr[\mathcal{A}^g \to 1]. \tag{6}$$

If g is a uniform random function from $\mathcal{X}$ to $\mathcal{Y}$, then the advantage will be denoted by $\mathbf{Adv}_f^{\mathrm{prf}}(\mathcal{A})$. For positive integers q and σ, we define $\mathbf{Adv}_f^{\mathrm{prf}}(q,\sigma)$ to be the maximum advantage of any adversary which makes at most q distinct queries $x_1,\ldots,x_q$ such that $\sum_{i=1}^{q}\lambda(x_i)\leq\sigma$. The quantity $\mathbf{Adv}_f^{\mathrm{prf}}(q,\sigma)$ is the PRF-advantage of f against any (q,σ)-bounded adversary.

Note that $\mathbf{Adv}^{\mathrm{prf}}(q,\sigma)$ is always non-negative even though $\mathbf{Adv}_f^{\mathrm{prf}}(\mathcal{A})$ can be negative for some adversaries. This is easily seen as follows. Suppose $\mathcal{A}$ is an adversary for which $\Pr[\mathcal{A}^f\Rightarrow 1]\leq\Pr[\mathcal{A}^{f^*}\Rightarrow 1]$, and let $\mathcal{A}'$ be the adversary which returns the complement of the bit produced by $\mathcal{A}$. Then

$$\begin{aligned}\mathbf{Adv}_f^{\mathrm{prf}}(\mathcal{A}') &= \Pr[\mathcal{A}'^f\Rightarrow 1]-\Pr[\mathcal{A}'^{f^*}\Rightarrow 1]\\ &= (1-\Pr[\mathcal{A}^f\Rightarrow 1])-(1-\Pr[\mathcal{A}^{f^*}\Rightarrow 1])\\ &= \Pr[\mathcal{A}^{f^*}\Rightarrow 1]-\Pr[\mathcal{A}^f\Rightarrow 1]\\ &\geq 0.\end{aligned}$$

Proposition 2. *Let f be a random function from $\mathcal{X}$ to $\mathcal{Y}$. Let $g:\mathcal{Y}\rightarrow\mathcal{Z}$ be a regular function, i.e., for any two $z_1,z_2\in\mathcal{Z}$, $\#g^{-1}(z_1)=\#g^{-1}(z_2)$. Define $h\stackrel{\Delta}{=}g\circ f$, so that $h(x)=g(f(x))$. Then*

$$\mathbf{Adv}_f^{\mathrm{prf}}(q,\sigma)=\mathbf{Adv}_h^{\mathrm{prf}}(q,\sigma).$$

Proof. Let f^* be chosen uniformly at random from $\mathcal{X}$ to $\mathcal{Y}$ and $h^*=g\circ f^*$. Then h^* is a uniform random function from $\mathcal{X}$ to $\mathcal{Z}$. Let $\mathcal{A}$ be an algorithm which distinguishes h from h^*. Using $\mathcal{A}$ we can build an algorithm $\mathcal{B}$ which distinguishes f from f^*. Any query made by $\mathcal{A}$ is processed by $\mathcal{B}$ by first asking its own oracle and then applying g on the output. Finally, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. If $\mathcal{B}$'s oracle is f, then $\mathcal{A}$ interacts with h, while if $\mathcal{B}$'s oracle is f^*, then $\mathcal{A}$ interacts with h^*. From this it easily follows that $\mathbf{Adv}_h^{\mathrm{prf}}(\mathcal{A})=\mathbf{Adv}_f^{\mathrm{prf}}(\mathcal{B})$. Further, the query complexities of both $\mathcal{A}$ and $\mathcal{B}$ are the same. Thus, maximising both sides on the query complexity gives us the desired result. □

Of special interest is the situation when $\mathcal{Y}=\{0,1\}^n$ and g truncates its input to produce a t-bit output. This corresponds to the situation when an n-bit message authentication tag is truncated to produce a t-bit tag. Proposition 2 shows that the PRF-advantage of the function with the truncated output does not change from that of the original function.

Vaudenay proved a useful result (Lemma 22 in [40]) 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 [6]) 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 [40, 6]; we provide more details.

Theorem 1. *Let q and $\sigma\geq q$ be positive integers; f be a random function from a set $\mathcal{X}$ to a set $\mathcal{Y}$; and λ be a length function on $\mathcal{X}$. Suppose that $\mathcal{U}$ is a $(1-\varepsilon_1)$-large subset of $\mathcal{Y}^q$ and f is $(q,\sigma,1-\varepsilon_2)$-interpolating on $\mathcal{U}$ with respect to λ. Then,*

$$\mathbf{Adv}_f^{\mathrm{prf}}(q,\sigma)\leq\varepsilon_1+\varepsilon_2.$$

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

Proof. For any adversary $\mathcal{A}$, let $\mathcal{V} = \overline{\mathsf{Acc}(\mathcal{A})}$, where $\mathsf{Acc}(\mathcal{A})$ is as defined in (5). Then $\mathcal{V}$ is the subset of $\mathcal{Y}^q$ such that if $\mathcal{A}$ receives any $\mathbf{y} \in \mathcal{V}$ as reply to the oracle queries, then $\mathcal{A}$ outputs 0, i.e., $\mathcal{V} = \{\mathbf{y} \in \mathcal{Y}^q : \Phi^{\mathcal{A}}(\mathbf{y}) = 0\}$. As noted earlier, $\mathcal{V}$ is independent of the function f and depends only on the adversary $\mathcal{A}$. Then for any random function f,

$$\sum_{y \in \mathcal{V}} \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] + \sum_{y \notin \mathcal{V}} \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] = 1. \tag{7}$$

Also,

$$\Pr[\mathcal{A}^f \to 1] = \sum_{\mathbf{y} \notin \mathcal{V}} \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] \text{ and similarly, } \Pr[\mathcal{A}^{f^*} \to 1] = \sum_{\mathbf{y} \notin \mathcal{V}} \Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}]. \tag{8}$$

Here f^* is a uniform random function from $\mathcal{X}$ to $\mathcal{Y}$. So,

$$\begin{aligned}
\mathbf{Adv}(\mathcal{A}) &= \Pr[\mathcal{A}^f \to 1] - \Pr[\mathcal{A}^{f^*} \to 1] \\
&\overset{(8)}{=} \sum_{\mathbf{y} \notin \mathcal{V}} \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] - \sum_{\mathbf{y} \notin \mathcal{V}} \Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] \\
&\overset{(7)}{=} \sum_{\mathbf{y} \in \mathcal{V}} (\Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] - \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}]) \\
&= \sum_{\mathbf{y} \in \mathcal{V}, \mathbf{y} \in \mathcal{U}} (\Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] - \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}]) \\
&\quad + \sum_{\mathbf{y} \in \mathcal{V}, \mathbf{y} \notin \mathcal{U}} (\Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] - \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}]) \,.
\end{aligned} \tag{9}$$

Since f is $(q, \sigma, 1 - \varepsilon_2)$-interpolating on $\mathcal{U}$ with respect to λ, we have that for all $\mathbf{x} \in \chi_q(\mathcal{X})$ with $\lambda(\mathbf{x}) \le \sigma$ and for all $\mathbf{y} \in \mathcal{U}$,

$$\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] \ge (1 - \varepsilon_2)/(\#\mathcal{U}) \ge (1 - \varepsilon_2)/(\#\mathcal{Y})^q. \tag{10}$$

The function f^* is a random function from $\mathcal{X}$ to $\mathcal{Y}$, and hence, for all $\mathbf{x} \in \chi_q(\mathcal{X})$ and for all $\mathbf{y} \in \mathcal{Y}^q$, $\Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] = 1/(\#\mathcal{Y})^q$. Using this and (10) we have for all $\mathbf{x} \in \chi_q(\mathcal{X})$ and for all $\mathbf{y} \in \mathcal{U}$,

$$\Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] - \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] \le \varepsilon_2 \Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}].$$

Consequently,

$$\sum_{\mathbf{y} \in \mathcal{V}, \mathbf{y} \in \mathcal{U}} (\Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] - \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}]) \le \varepsilon_2 \sum_{\mathbf{y} \in \mathcal{V}, \mathbf{y} \in \mathcal{U}} \Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] \le \varepsilon_2. \tag{11}$$

By the fact that $\mathcal{U}$ is a $(1 - \varepsilon_1)$-large subset of $\mathcal{Y}^q$, $(\#\mathcal{Y})^q - (\#\mathcal{U}) \le \varepsilon_1 (\#\mathcal{Y})^q$, and so,

$$\begin{aligned}
\sum_{\mathbf{y} \in \mathcal{V}, \mathbf{y} \notin \mathcal{U}} (\Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] - \Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}]) &\le \sum_{\mathbf{y} \in \mathcal{V}, \mathbf{y} \notin \mathcal{U}} \Pr[\mathrm{ECB}_{f^*}(\mathbf{x}) = \mathbf{y}] \\
&= \sum_{\mathbf{y} \in \mathcal{V}, \mathbf{y} \notin \mathcal{U}} \frac{1}{(\#\mathcal{Y})^q} \\
&\le \frac{(\#\mathcal{Y})^q - (\#\mathcal{U})}{(\#\mathcal{Y})^q} \\
&\le \varepsilon_1.
\end{aligned} \tag{12}$$

Substituting (11) and (12) in (9) gives the desired inequality. □

Informally, Theorem 1 states that if f has high interpolation probability on a large subset $\mathcal{U}$ of $\mathcal{Y}^q$, then f is a PRF.

Note. In many situations, it is difficult to directly lower bound an interpolation probability of the form $\Pr[f(x_1) = y_1, \ldots, f(x_q) = y_q]$. Instead, it turns out to be easier to lower bound $\Pr[\mathcal{E}]$ and $\Pr[f(x_1) = y_1, \ldots, f(x_q) = y_q | \mathcal{E}]$, 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 $\Pr[f(x_1) = y_1, \ldots, f(x_q) = y_q | \mathcal{E}] \geq 1 - \varepsilon_{2,2}$, then $\Pr[f(x_1) = y_1, \ldots, f(x_q) = y_q] \geq 1 - \varepsilon_2$ where $\varepsilon_2 = \varepsilon_{2,1} + \varepsilon_{2,2}$.

2.5 Pseudo-Random Functions and Message Authentication

Suppose that the output of f on any input is a t-bit string. Then the function f can be used to authenticate a message x using a t-bit tag. The authenticity of f is defined as follows. The adversary has access to f as an oracle and can submit queries in an adaptive manner. Finally, $\mathcal{A}$ outputs a "forged" pair (x, y) and is said to be successful if $f(x) = y$. The pair (x, y) must not be equal to any previous pair (x_i, y_i), where x_i was the i-th query and y_i was the corresponding response.

By $(x, y) \leftarrow \mathcal{A}^f$ we denote the event that $\mathcal{A}$ produces (x, y) as output after interacting with f. The event $\mathsf{succ}(\mathcal{A})$ denotes the event $(((x, y) \leftarrow \mathcal{A}^f) \wedge (f(x) = y))$. Since f is a function, $x = x_i$ implies that $f(x) = f(x_i)$. So, if $x = x_i$ but $y \neq y_i$, then the forgery (x, y) is clearly invalid. Therefore, for a valid forgery saying that (x, y) is not equal to (x_i, y_i) is equivalent to saying that $x \neq x_i$. The advantage of $\mathcal{A}$ in breaking the authenticity of f is defined to be

$$\mathbf{Adv}_f^{\mathrm{auth}}(\mathcal{A}) = \Pr[\mathsf{succ}(\mathcal{A})] = \Pr[((x, y) \leftarrow \mathcal{A}^f) \wedge (f(x) = y)]. \tag{13}$$

As in the case of PRF, we define $\mathbf{Adv}_f^{\mathrm{auth}}(q, \sigma)$ to be the maximum of $\mathbf{Adv}_f^{\mathrm{auth}}(\mathcal{A})$ taken over all adversaries making at most q queries and having query complexity at most σ. In this case, the query complexity also counts the number of blocks in the forgery attempt.

Proposition 3. *Let f be a random function chosen from $\mathcal{X}$ to $\mathcal{Y}$ where $\mathcal{Y} = \{0,1\}^t$, $t > 0$. Then*

$$\mathbf{Adv}_f^{\mathrm{auth}}(q, \sigma) = \frac{1}{2^t} + \mathbf{Adv}_f^{\mathrm{prf}}(q, \sigma).$$

Proof. If $\mathcal{A}$ is an adversary attacking the authenticity property of f, then we can use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ attacking the PRF-property of f. $\mathcal{B}$ has access to either f or f^*, where f^* is a uniform random function from $\mathcal{X}$ to $\mathcal{Y}$.

$\mathcal{B}$ runs $\mathcal{A}$ and responds to all queries from $\mathcal{A}$ using its own oracle. At the end, $\mathcal{A}$ outputs the forgery (x, y); $\mathcal{B}$ now makes one more call to its oracle and receives y'; if $y = y'$, then $\mathcal{B}$ outputs 1, else $\mathcal{B}$ outputs 0. Clearly, the query complexities of both $\mathcal{A}$ and $\mathcal{B}$ are equal. Also, $\Pr[\mathcal{B}^f \Rightarrow 1] = \Pr[(x, y) \leftarrow \mathcal{A}^f : f(x) = y]$. Since x is a "new" value and f^* is a uniform random function $\Pr[\mathcal{B}^{f^*} \Rightarrow 1] = 1/2^t$. Now

$$\begin{aligned}
\mathbf{Adv}_f^{\mathrm{prf}}(\mathcal{B}) &= \Pr[\mathcal{B}^f \Rightarrow 1] - \Pr[\mathcal{B}^{f^*} \Rightarrow 1] \\
&= \Pr[(x, y) \leftarrow \mathcal{A}^f : f(x) = y] - \frac{1}{2^t} \\
&= \mathbf{Adv}_f^{\mathrm{auth}}(\mathcal{A}) - \frac{1}{2^t}.
\end{aligned}$$

So, $\mathbf{Adv}_f^{\mathrm{auth}}(\mathcal{A}) = \mathbf{Adv}_f^{\mathrm{prf}}(\mathcal{B}) + 1/2^t$ and maximising both sides on the query complexity gives us the desired result. □

The advantage of Proposition 3 is that it allows us to concentrate on proving the PRF-property of a random function, from which the authenticity property automatically follows.

A combination of Propositions 2 and 3 simplifies reasoning about certain situations. Suppose that f is a function which produces an n-bit output. This output is truncated to t bits and let the resulting function be h; further, suppose h is used for message authentication. Then we have $\mathbf{Adv}_h^{\mathrm{auth}}(q, \sigma) = 1/2^t + \mathbf{Adv}_f^{\mathrm{prf}}(q, \sigma)$. Thus, to show that h provides good authentication, it is sufficient to show that f is a good PRF.

For message authentication applications, we will be interested in $\mathcal{X}$ to be the set of all binary strings of lengths between 0 and some maximum value (say L), i.e., $\mathcal{X} = \cup_{i=0}^{L}\{0,1\}^i$. (Here L will be large enough to cover practical sized messages; for example $L = 2^{64}$ is a possible value.) A PRF whose domain is $\mathcal{X}$ can handle the empty string as well as small strings. One can also put a restriction on the minimum length (say n) of a string. The following simple result shows that given a PRF over such a restricted domain, it is easy to obtain a PRF without such a restriction.

Proposition 4. *Let $\mathcal{Y}$ be a finite non-empty set of binary strings and $f : \cup_{i=n}^{L+n}\{0,1\}^i \to \mathcal{Y}$, where the length function λ formats a binary string into n-bit blocks with possibly one partial block at the end. For any $\mathsf{fStr} \in \{0,1\}^n$, define the function $g : \cup_{i=0}^{L}\{0,1\}^i \to \mathcal{Y}$ as*

$$g(x) = f(\mathsf{fStr}, x).$$

Then $\mathbf{Adv}_g(q, \sigma) \leq \mathbf{Adv}_f(q, \sigma + q)$.

Proof. If $x_1, \ldots, x_q$ are q distinct binary strings with $0 \leq \mathsf{len}(x_i) \leq L$, then $(\mathsf{fStr}, x_1), \ldots, (\mathsf{fStr}, x_q)$ are also q distinct binary strings with $n \leq \mathsf{len}(\mathsf{fStr}, x_i) \leq L + n$. So, given an adversary for f, one easily constructs an adversary for g. If the adversary for g has query complexity σ, then clearly the adversary for f has query complexity $q + \sigma$, corresponding to the fStr that has to be given as initial n bits of each of the q queries. □

For constructing PRFs, Proposition 4 is helpful in the following manner. First construct a PRF f which can handle strings of lengths n or more (up to $L + n$). Then use Proposition 4 to convert it into a PRF g which handles strings of lengths between 0 and L. This may seem simple at this point. But, later we will find this to be quite convenient in analysing the relation between PRFs and authentication of AE schemes.

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 a 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 $\mathcal{U}$ to $\mathcal{Y}$. Using ρ, we wish to construct another random function f which maps from a set $\mathcal{X}$ to $\mathcal{Y}$, where $\mathcal{X}$ is larger than $\mathcal{U}$. In other words, we wish to extend the domain from $\mathcal{U}$ to $\mathcal{X}$. To capture such constructions, we have the following definition.

Definition 5. *Let $\rho : \mathcal{U} \to \mathcal{Y}$ be a random function. A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be a domain extender for ρ if $f = \rho \circ f_1^{(\rho)}$, where $f_1 : \mathcal{X} \to \mathcal{U}$ 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 $\mathcal{X}$. For every x in $\mathcal{X}$, $\lambda(x)$ denotes the total number of times ρ is invoked to compute the final output of f.

We wish to compute $\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}]$, where $\mathbf{x} \in \chi_q(\mathcal{X})$ and $\mathbf{y} \in \mathcal{Y}^q$. Here f_1 and ρ "interact" and hence we need to account for such possibilities. To this end, we make the following definition.

Definition 6. *Let $\rho : \mathcal{U} \to \mathcal{Y}$ be a random function and $f = \rho \circ f_1$ be a map from $\mathcal{X}$ to $\mathcal{Y}$ satisfying Definition 5. For $x, x' \in \mathcal{X}$ 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 7. *Continuing with Definition 6, we say that f_1 is $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ, if for all pairs of distinct $x, x' \in \mathcal{X}$,*

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

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.

Theorem 2. *Let $\rho : \mathcal{U} \to \mathcal{Y}$ be a random function and $f = \rho \circ f_1$ be a map from $\mathcal{X}$ to $\mathcal{Y}$ satisfying Definition 5. 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 q and $\sigma \geq q$ the following holds.*

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

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

2. *If $\mathcal{U} = \mathcal{Y}$ and ρ is a uniform random permutation, then*

$$\mathbf{Adv}_f(q, \sigma) \leq \sigma(q\varepsilon + \varepsilon_1 + 2q\varepsilon_2) + \frac{q\sigma}{\#\mathcal{Y}}.$$

Proof. Let $\mathbf{x} = (x_1, \ldots, x_q) \in \chi_q(\mathcal{X})$ with $m_i + 1 = \lambda(x_i)$. Then $\sigma = q + \sum_{i=1}^{q} m_i$. 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

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

and

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

The event $\mathsf{Distinct}(\mathbf{x})$ is the event $\mathrm{ECB}_{f_1,q}(\mathbf{x}) \in \chi_q(\mathcal{U})$. Using the fact that f_1 is ε-CR with respect to λ and Lemma 3,

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

We compute

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

Since f_1 is $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ, we have

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

Using (17),

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

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

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

Let $\mathbf{y} \in \mathcal{Y}^q$. Then,

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

The event "$\mathsf{Distinct} \wedge \mathsf{Disjoint}$" means that the random variables $Z_1, \ldots, Z_q$ 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 $\{Z_1, \ldots, Z_q\}$ is a set of q "new" values in the domain of ρ.

If ρ is a uniform random function, then $\mathrm{ECB}_\rho(Z_1, \ldots, Z_q)$ is uniformly distributed over $\mathcal{Y}^q$. 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 $\mathbf{y} \in \mathcal{Y}^q$, $\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct})] = 1/(\#\mathcal{Y})^q$ and so,
$$\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] \geq \frac{1}{\#\mathcal{Y}^q} \times (1 - \sigma(q\varepsilon + \varepsilon_1 + 2q\varepsilon_2)).$$
This lower bounds the interpolation probabilities of f. Now, applying Theorem 1,
$$\mathbf{Adv}_f(q, \sigma) \leq \sigma(q\varepsilon + \varepsilon_1 + 2q\varepsilon_2).$$

2. Suppose that ρ is a uniform random permutation of $\mathcal{Y}$. Let $V_{i,j} = \rho(U_{i,j})$ and define $\mathcal{V} = \cup_{i=1}^{q} \{V_{i,1}, \ldots, V_{i,m_i}\}$. Fix $\mathbf{y} = (y_1, \ldots, y_q) \in \chi_q(\mathcal{Y})$. Let $\mathsf{Allowed}(\mathbf{y})$ be the event $\wedge_{k=1}^{q} (y_k \notin \mathcal{V})$. Since ρ is a uniform random permutation, for any y_k, $\Pr[y_k = V_{i,j}] = 1/\#\mathcal{Y}$. Note that $\sigma = q + \sum_{i=1}^{q} m_i$.
$$\begin{aligned}
\Pr[\mathsf{Allowed}(\mathbf{y})] = \Pr\left[\bigwedge_{k=1}^{q} (y_k \notin \mathcal{V})\right] &= 1 - \Pr\left[\bigvee_{k=1}^{q} (y_k \in \mathcal{V})\right] \\
&\geq 1 - \sum_{k=1}^{q} \Pr[y_k \in \mathcal{V}] \\
&= 1 - \sum_{k=1}^{q} \Pr\left[\bigvee_{i=1}^{q} \bigvee_{j=1}^{m_i} (y_k = V_{i,j})\right] \\
&\geq 1 - \sum_{k=1}^{q} \sum_{i=1}^{q} \sum_{j=1}^{m_i} \Pr[y_k = V_{i,j}] \\
&= 1 - \frac{q(\sigma - q)}{\#\mathcal{Y}}.
\end{aligned}$$
Let us consider when can $\rho(Z_k)$ 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,
$$\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct} \wedge \overline{\mathsf{Allowed}(\mathbf{y})})] = 0.$$
On the other hand, if both $(\mathsf{Disjoint} \wedge \mathsf{Distinct})$ and $\mathsf{Allowed}(\mathbf{y})$ occur, then
$$\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct} \wedge \mathsf{Allowed}(\mathbf{y}))] = \frac{1}{p(\#\mathcal{Y} - (\sigma - q), q)} \geq \frac{1}{(\#\mathcal{Y})^q}.$$
Now, for any $\mathbf{y} \in \chi_q(\mathcal{Y})$,
$$\begin{aligned}
\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y} | (\mathsf{Disjoint} \wedge \mathsf{Distinct})] &\geq \frac{1}{(\#\mathcal{Y})^q} \times \Pr[\mathsf{Allowed}(\mathbf{y})] \\
&\geq \frac{1}{(\#\mathcal{Y})^q} \left(1 - \frac{q(\sigma - q)}{\#\mathcal{Y}}\right).
\end{aligned}$$

This gives

$$\begin{aligned}\Pr[\mathrm{ECB}_f(\mathbf{x}) = \mathbf{y}] &\geq \frac{1}{(\#\mathcal{Y})^q}\left(1 - \frac{q(\sigma-q)}{\#\mathcal{Y}}\right) \times (1 - \sigma(q\varepsilon + \varepsilon_1 + 2q\varepsilon_2)) \\ &\geq \frac{1}{(\#\mathcal{Y})^q}\left(1 - \sigma(q\varepsilon + \varepsilon_1 + 2q\varepsilon_2) - \frac{q(\sigma-q)}{\#\mathcal{Y}}\right).\end{aligned}$$

Again, applying Theorem 1,

$$\begin{aligned}\mathbf{Adv}_f(q,\sigma) &\leq \sigma(q\varepsilon + \varepsilon_1 + 2q\varepsilon_2) + \frac{q(\sigma-q)}{\#\mathcal{Y}} + \frac{q(q-1)}{2\#\mathcal{Y}} \\ &\leq \sigma(q\varepsilon + \varepsilon_1 + 2q\varepsilon_2) + \frac{q\sigma}{\#\mathcal{Y}}.\end{aligned}$$

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 $\pi : T \to \mathcal{Y}$ be a uniform random permutation and $f = \pi \circ f_1$ be a map from $\mathcal{X}$ to $\mathcal{Y}$ satisfying Definition 5. Suppose that f_1 is ε-CR and it is $(\varepsilon_1, \varepsilon_2)$-disjoint with respect to λ. Then for positive integers q and $\sigma \geq q$*

$$\mathbf{Adv}_f(q,\sigma) \leq \frac{q(q-1)\varepsilon}{2} + \sigma(\varepsilon_1 + 2q\varepsilon_2) + \frac{q\sigma}{\#\mathcal{Y}}.$$

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.

4 Masking Functions

In the constructions to be described, we will be making use of linear functions with certain properties. Let $\mathbb{F}_{2^n} = GF(2^n)$ be the Galois field of 2^n elements. For the moment, we do not consider any particular representation of this field. The issue of field representation will be discussed later.

Let $\psi : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ be a linear function. The iterates ψ^k, $k \geq 0$ are defined in the usual manner. Let n_1 divide n, so that $\mathbb{F}_{2^{n_1}}$ is a subfield of $\mathbb{F}_{2^n}$. The minimal polynomial of ψ over $\mathbb{F}_{2^{n_1}}$ is defined to be the minimum degree monic polynomial $\tau(x) \in \mathbb{F}_{2^{n_1}}[x]$ such that $\tau(\psi) = 0$, i.e., τ annihilates ψ. Note that the coefficients of $\tau(x)$ are in the field $\mathbb{F}_{2^{n_1}}$. Given ψ, we define another function $\phi : \mathbb{F}_{2^n} \times \{0, 1, \ldots, 2^n - 2\} \to \mathbb{F}_{2^n}$ as

$$\phi(\beta, i) = \psi^i(\beta) \tag{21}$$

and we use the notation $\phi_\beta(i) \stackrel{\Delta}{=} \phi(\beta, i)$.

Definition 8. *We say that the function ϕ defined in (21) is a proper masking function if it satisfies the following properties.*

1. *For any $\alpha \in \mathbb{F}$; any non-negative integer k with $0 \leq k \leq 2^n - 2$; and a uniform random $\beta \in \mathcal{Y}$;* $\Pr[\phi_\beta(k) = \alpha] = 1/2^n$.

2. *For any* $\alpha \in \mathbb{F}$*; integers* k_1, k_2 *with* $0 \leq k_1 < k_2 \leq 2^n - 2$*; and a uniform random* $\beta \in \mathbb{F}$*;* $\Pr[\phi_\beta(k_1) \oplus \phi_\beta(k_2) = \alpha] = 1/2^n$.
3. *For any* $\alpha \in \mathbb{F}$*; integers* k_1, k_2 *with* $0 \leq k_1, k_2 \leq 2^n - 2$*; and uniform random* $(\beta_1, \beta_2) \in \chi_2(\mathbb{F})$*,* $\Pr[\phi_{\beta_1}(k_1) \oplus \phi_{\beta_2}(k_2) = \alpha] = 1/(2^n - 1)$.

There is a very general class of functions satisfying Definition 8.

Proposition 5. *Let* $\psi : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ *be a linear function whose minimal polynomial* $\tau(u)$ *over* $\mathbb{F}_2$ *is of degree* n *and is primitive over* $\mathbb{F}_2$*. Then* ϕ *defined in (21) satisfies Definition 8.*

Note. The lemma does not insist on any particular representation of $\mathbb{F}_{2^n}$. In particular, $\mathbb{F}_{2^n}$ can have a tower field representation. The condition is that whatever be the representation of $\mathbb{F}_{2^n}$, its minimal polynomial over $\mathbb{F}_2$ should be of degree n and primitive.

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.

For $0 \leq i < j \leq 2^n - 2$, define $\eta_{i,j} : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ as

$$\eta_{i,j}(\gamma) = \psi^i(\gamma) \oplus \psi^j(\gamma) = \phi_\gamma(i) \oplus \phi_\gamma(j).$$

The second point will follow if we can show that $\eta_{i,j}$ is a bijection. For this, it is sufficient to show that $\eta_{i,j}$ is an injection. So, suppose that γ and γ' are distinct elements of $\mathbb{F}_{2^n}$ and let, if possible, $\eta_{i,j}(\gamma) = \eta_{i,j}(\gamma')$. Set $\delta = \gamma \oplus \gamma'$ and note that since $\gamma \neq \gamma'$, we have δ to be non-zero. Then

$$\begin{aligned} 0 &= \eta_{i,j}(\gamma) \oplus \eta_{i,j}(\gamma') \\ &= \psi^i(\gamma) \oplus \psi^j(\gamma) \oplus \psi^i(\gamma') \oplus \psi^j(\gamma') \\ &= (\psi^i \oplus \psi^j)(\delta). \end{aligned} \tag{22}$$

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 $\mathbb{F}_{2^n}$ 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 (22), 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 [20]). 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 $\eta_{i,j}$ is a injection.

For the third point, consider the map $\zeta_{k_1,k_2} : \mathbb{F}_{2^n}^2 \to \mathbb{F}_{2^n}$ which takes (β_1, β_2) to $\psi^{k_1}(\beta_1) \oplus \psi^{k_2}(\beta_2) = \phi_{\beta_1}(k_1) \oplus \phi_{\beta_2}(k_2)$. We count the number of pre-images of $\alpha \in \mathbb{F}_{2^n}$ 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 2^n pre-images for any α. Since (β_1, β_2) is uniformly distributed over $\chi_2(\mathbb{F}_{2^n})$, the result follows. □

There are known examples of ψ which satisfy Proposition 5.

Powering up method. Let $\tau(u)$ be a primitive polynomial of degree n over $GF(2)$ and define $\psi : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ as $\psi : \beta \mapsto u\beta \bmod \tau(u)$ and so $\psi^k : \beta \mapsto u^k\beta \bmod \tau(u)$. The minimal polynomial of ψ is clearly $\tau(u)$ which by definition is primitive over $\mathbb{F}_{2^n}$. This strategy is quite common and has been used in many papers [8, 35, 15, 11, 41].

Word oriented LFSR. Faster instantiations can be obtained using a tower field representation of $\mathbb{F}$ [37]. Let $n = n_1 \times n_2$ and let $\rho(\alpha)$ be an irreducible polynomial of degree n_1 which is used to define $\mathbb{F}_{2^{n_1}}$ over $\mathbb{F}_2$, i.e., $\mathbb{F}_{2^{n_1}} = \mathbb{F}_2[\alpha]/(\rho(\alpha))$. Let $\mu(x) = x^{n_2} \oplus t_{n_2-1}x^{n_2-1} \oplus \cdots \oplus t_1 x \oplus t_0$, with $t_{n_2-1}, \ldots, t_0 \in \mathbb{F}_{2^{n_1}}$ be a primitive polynomial over $\mathbb{F}_{2^{n_1}}$ which is used to define $\mathbb{F}_{2^n}$ over $\mathbb{F}_{2^{n_1}}$. The field $\mathbb{F}_{2^n}$ can be represented by the polynomial basis $\{1, x, x^2, \ldots, x^{n_2-1}\}$ with multiplication done modulo $\mu(x)$.

Define a map $\psi : \mathbb{F}_{2^{n_1}}^{n_2} \to \mathbb{F}_{2^{n_1}}^{n_2}$, with $(b_{n_2-1}, \ldots, b_0) = \psi(a_{n_2-1}, \ldots, a_0)$ as follows.

$$\left.\begin{array}{rl} b_i = a_{i+1} & \text{if } 0 \le i \le n_2 - 2; \\ b_{n_2-1} = t_0 a_{n_2-1} \oplus \cdots \oplus t_{n_2-1} a_0. & \end{array}\right\} \tag{23}$$

This defines an LFSR over $\mathbb{F}_{2^{n_1}}$. The minimal polynomial of ψ is $\mu(x)$ which is in $\mathbb{F}_{2^{n_1}}[x]$ and is primitive. To show that Proposition 5 is satisfied, we need to show that the minimal polynomial of ψ over $\mathbb{F}_2$ is also primitive and of degree n. The result below is actually more general than what is required. From this result, it easily follows that ψ satisfies Proposition 5.

Lemma 4. *Let $n = n_1 \times n_2$, $\rho(\alpha)$ be an irreducible polynomial of degree n_1 over $\mathbb{F}_2$ and $\mathbb{F}_{2^{n_1}} = \mathbb{F}_2[\alpha]/(\rho(\alpha))$. Further, let $\psi : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ be a linear map whose minimal polynomial, $\mu(x)$, over $\mathbb{F}_{2^{n_1}}$ is primitive and of degree n_2. Then the minimal polynomial of ψ over $\mathbb{F}_{2^n}$ is of degree n and is primitive over $\mathbb{F}_2$.*

Note. The lemma does not assume ψ to be an LFSR map. The condition states that ψ can be any linear map defined using the tower field representation of $\mathbb{F}_{2^n}$ and whose minimal polynomial over the intermediate field is primitive.

Proof. Suppose that $\mathbb{F}_{2^{n_1}}$ is represented by the polynomial basis $\{1, \alpha, \alpha^2, \ldots, \alpha^{n_1-1}\}$ over $\mathbb{F}_2$. Then any element γ in $\mathbb{F}_{2^{n_1}}$ can also be considered to be a polynomial $\gamma(\alpha)$ over $\mathbb{F}_2$ whose degree is less than n_1. Further, for any $\beta \in \mathbb{F}_{2^{n_1}}$, the "multiply-by-$\beta$" map from $\mathbb{F}_{2^{n_1}}$ to $\mathbb{F}_{2^{n_1}}$ given by $\gamma(\alpha) \mapsto \beta(\alpha)\gamma(\alpha) \bmod \rho(\alpha)$ can be represented by an $(n_1 \times n_1)$ matrix M_β over $\mathbb{F}_2$ which maps $\mathbb{F}_2^{n_1}$ to $\mathbb{F}_2^{n_1}$.

Fix the polynomial basis $\{1, x, x^2, \ldots, x^{n_2-1}\}$ of $\mathbb{F}_{2^n}$ over $\mathbb{F}_{2^{n_1}}$. Then the map ψ is given by an $n_2 \times n_2$ matrix $B = ((\beta_{i,j}))$ over $\mathbb{F}_{2^{n_1}}$, so that each $\beta_{i,j}$ is an element of $\mathbb{F}_{2^{n_1}}$. Since the minimal polynomial $\mu(x)$ of ψ is primitive and of degree n_2, it follows that for any non-zero vector $\delta = (\delta_0, \ldots, \delta_{n_2-1}) \in \mathbb{F}_{2^{n_1}}^{n_2}$, the sequence $\delta, \delta B, \delta B^2, \ldots$ has period $2^{n_1 n_2} - 1 = 2^n - 1$.

From B construct an $n \times n$ matrix C by replacing each $\beta_{i,j}$ by the matrix $M_{\beta_{i,j}}$ representing the "multiply-by-$\beta_{i,j}$" map. It then follows that for any $k \ge 0$, C^k is obtained from B^k using the same procedure. Each δ_i can be considered to be an element of $\mathbb{F}_2^{n_1}$, so that δ can be considered to be an element of $\mathbb{F}_2^n$. So we can talk of the product δC^k for $k \ge 0$. Then the sequence $\delta, \delta C, \delta C^2, \ldots$ is exactly the sequence $\delta, \delta B, \delta B^2, \ldots$ and hence is of period $2^n - 1$. This can only happen when the minimal polynomial of C over $\mathbb{F}_2$ is primitive and of degree n. But, the minimal polynomial of C over $\mathbb{F}_2$ is also the minimal polynomial of ψ over $\mathbb{F}_2$, which proves the result. □

We implemented extension field arithmetic to generate suitable pairs of $(\rho(\alpha), \mu(x))$. For different values of n, Table 3 provides examples of pairs of polynomials which can be used to define ψ as word oriented LFSRs.

Efficiency improvement over powering method. We implemented both the powering method and the word oriented LFSR method on Intel Core 2 DuoProcessor P8700 (2.53 GHz) running Fedora Release 11. For word oriented LFSRs, we took the intermediate field to be $\mathbb{F}_{2^{32}}$. The results

for different values of n are shown in Table 4. This shows that implementation using word-oriented LFSRs is about twice as fast as that using the powering method.

Effect on modes of operations. A mode of operation will typically have one block cipher call and one application of ϕ per n-bit block. The dominant time will be the time for the block cipher call. Faster masking, though, will provide some overall speed-up.

To get an idea of this speed-up for $n = 128$, let us consider the values of 0.19 cycles/byte and 0.32 cycles/byte respectively for word oriented LFSR and powering up, as given in Table 4. The reported speeds of AES-128 on different platforms are quite varied. The best reported speed for Intel Core 2 Duo is 9.2 cycles/byte for a bit-slice implementation [24] to about 10.6 [7]. Assuming 10 cycles/byte for AES-128, the speed-up obtained by using word-oriented LFSR will be about 1 to 2%. Admittedly, this is not much. But, it is to be noted that there are no trade-offs involved and for algorithms which are likely to be heavily used it might be worthwhile to go for even small improvements in efficiency.

Related to this is the issue of Intel proposing to implement AES-128 instructions as processor instructions starting from a processor called Westmere [13]. This will significantly increase the speed of AES-128 and possibly bring down the time to 2 or 3 cycles/byte. In such a scenario, the speed improvement of 0.17 cycles/byte can be expected to lead to about 5% speed-up.

Table 3. Examples of $\mathbb{F}_{2^n}$ represented as a tower field.

n_1	n_2	$n = n_1 \times n_2$	$\rho(\alpha)$	$\mu(x)$
32	2	64	$\alpha^{32} + \alpha^{31} + \alpha^{16} + \alpha^2 + 1$	$x^2 + x + \alpha$
16	5	80	$\alpha^{16} + \alpha^{14} + \alpha^{11} + \alpha^7 + 1$	$x^5 + x^2 + \alpha$
96	3	96	$\alpha^{32} + \alpha^{26} + \alpha^{22} + \alpha^7 + 1$	$x^3 + x + \alpha$
32	4	128	$\alpha^{32} + \alpha^{27} + \alpha^{25} + \alpha^5 + 1$	$x^4 + x^3 + x + \alpha$
16	8	128	$\alpha^{16} + \alpha^{10} + \alpha^9 + \alpha^6 + 1$	$x^8 + x^3 + x + \alpha$
8	16	128	$\alpha^8 + \alpha^7 + \alpha^3 + \alpha^2 + 1$	$x^{16} + x^7 + x + \alpha$
32	8	256	$\alpha^{32} + \alpha^{25} + \alpha^{14} + \alpha^{13} + 1$	$x^8 + x^7 + x^5 + x^1 + \alpha$
32	12	384	$\alpha^{32} + \alpha^{26} + \alpha^{20} + \alpha^{11} + 1$	$x^{12} + x^3 + x^1 + \alpha$
32	16	512	$\alpha^{32} + \alpha^{15} + \alpha^{10} + \alpha^1 + 1$	$x^{16} + x^3 + x^2 + \alpha$

Table 4. Speed comparison between powering method and word oriented LFSR. The figures in the first row are the different values of n and the values in the other two rows are given in cycles per byte.

method	128	192	256	384	512
powering	0.32	0.29	0.28	0.28	0.28
WLFSR	0.19	0.17	0.10	0.10	0.10

Notes.

1. The security of the constructions to be described later do not depend on the actual implementation of ψ. Only the properties given by Definition 8 will be used.
2. In both the representations of $\mathbb{F}_{2^n}$ discussed above (i.e., either as $\mathbb{F}_2[x]/(\tau(x))$ or as a tower field), an element of $\mathbb{F}_{2^n}$ can be naturally considered to be an n-bit string. This will be assumed without further mention.

5 New Parallelizable Constructions of Pseudo-Random Functions

The basic idea of parallelizing is to apply a 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 [6] and PMAC [8, 35] are based on this principle. PCS uses a compressing function, whereas PMAC uses a permutation.

5.1 iPMAC

Let π be a uniform random permutation of $\{0,1\}^n$ and ϕ be a function satisfying Definition 8. For any binary string x with $\mathsf{len}(x) \geq n$, let $(\mathsf{str}, P_1, \ldots, P_m)$ be the output of $\mathsf{Format}(x, n)$ and $\lambda(x) = m$. Here str is an n-bit string and $(P_1, \ldots, P_m)$ is of length greater than or equal to zero. Let $\gamma = \pi(\mathsf{str})$ and $\delta = \pi(\gamma)$. The function iPMAC_π is a map

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

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

$$D_i = \begin{cases} P_i \oplus \phi_\gamma(i) & 1 \leq i \leq m-1; \\ C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m & i = m, r = n; \\ C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \phi_\gamma(m) & i = m, r < n; \\ \delta \oplus P_1 & i = m = 1, r = n; \\ \delta \oplus P_1 \oplus \phi_\gamma(1) & i = m = 1, r < n. \end{cases} \tag{24}$$

Notes.

1. The range of iPMAC is $\mathcal{Y} = \{0,1\}^n$. From the manner iPMAC processes its input, there is no restriction on the length. We put a bound of $2^{n/2}$ on the maximum length of any input. The bound on the PRF-advantage of iPMAC that we obtain later becomes meaningless for strings beyond this length. So, formally, iPMAC is a map
$$\mathsf{iPMAC} : \bigcup_{i \geq n}^{2^{n/2}} \{0,1\}^i \to \{0,1\}^n.$$
2. The function iPMAC_π accepts strings of length at least n bits. By fixing the first n bits of the input to a string fStr, we obtain a function which can handle strings of length greater than or equal to zero. By a slight abuse of notation, we denote this function by $\mathsf{iPMAC}_{\pi,\mathsf{fStr}}$. Later we show that iPMAC_π is a PRF and then using Proposition 4, we obtain the PRF-bound for $\mathsf{iPMAC}_{\pi,\mathsf{fStr}}$. The function $\mathsf{iPMAC}_{\pi,\mathsf{fStr}}$ can be used to obtain a MAC algorithm as described in Propositions 2 and 3.
3. To process $(\mathsf{str}, P_1, \ldots, P_m)$ iPMAC_π requires $(m+1)$ applications of π if $m > 1$; and 3 applications of π if $m = 1$. One application of π to str produces γ and there are m other applications; if $m = 1$, then one application of π to γ produces δ.
4. In $\mathsf{iPMAC}_{\pi,\mathsf{fStr}}$, the first n bits are fixed and so are γ and δ. Thus, the values of γ and δ can be computed once per session and cached. Further, the value of δ is required only if strings of lengths at most n bits are required to be processed in a session. Otherwise, this value need not be computed at all.

5. The mask δ is used to distinguish between strings of lengths at most n and strings of greater lengths. Masking the first $(m-1)$ blocks by $\phi_\gamma(1), \ldots, \phi_\gamma(m-1)$ ensures that with high probability the corresponding inputs to π are distinct. The masking or not of the last block by $\phi_\gamma(m)$ separates strings whose lengths are not multiples of n from strings whose lengths are multiples of n.

Processing of a 4-block message using iPMAC is shown in Figure 1. The same figure also describes the processing of a 4-block message using PMAC. The difference is in the interpretation of Λ.

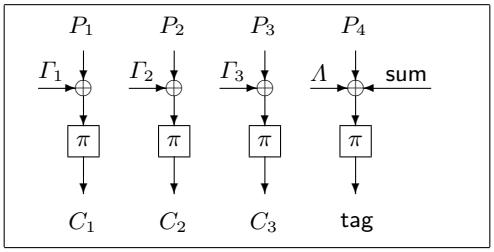

Fig. 1. Tag generation using iPMAC Here $\Gamma_i = \phi_\gamma(i)$; $\Lambda = 0^n$ if the last block is full and $\Lambda = \phi_\gamma(4)$ if the last block has been padded; and $\mathsf{sum} = C_1 \oplus C_2 \oplus C_3$.

iPMAC. If the last block is full, then $\Lambda = 0^n$ and if the last block is partial (and padded), then $\Lambda = \phi_\gamma(m)$.

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 [35]. For this scheme to be secure, the discrete logarithms of $(u \oplus 1)$ and $(u^2 \oplus u \oplus 1)$ to the base u have to be "large". The actual values depend on $\tau(x)$ and for $n = 128$ and the choice of $\tau(x)$ to be the first lexicographically first primitive polynomial the values are given in [35]. Changing $\tau(x)$ will require a re-computation of the discrete logarithm to ensure that it is "large"; for $n = 256$, computing the discrete logarithms will take a few hours and is not convenient

There are two advantages of iPMAC over PMAC. For one thing, iPMAC avoids the discrete logarithm computation required during the design stage. As a result, it is possible to simply change the field representation and obtain a distinct algorithm. So, iPMAC can be seen as providing an easily reconfigurable family of algorithms. The second advantage is the faster masking operations. The description of iPMAC does not depend on the actual choice of ϕ. As a result, one can use a tower representation of the underlying field and a word-oriented LFSR to instantiate ϕ (see Section 4).

Security. The security analysis of iPMAC is done using the general approach of domain extenders. Let iPHASH be the map which takes $x \mapsto D_m$. Following the approach outlined in Section 3, it is sufficient to bound the collision and disjointness probabilities of iPHASH. But, first let us consider this a little informally.

Suppose that q queries are made and let us take a look at the different inputs and outputs of π arising out of these queries. These can be tabulated as follows.

Inputs:

$\mathsf{str}^{(s)}, \gamma^{(s)}$

$$\begin{array}{ll}
D_1^{(s)} = P_1^{(s)} \oplus \delta^{(s)} & \text{if } (m^{(s)} = 1 \text{ and } r^{(s)} = n);\\
D_1^{(s)} = P_1^{(s)} \oplus \delta^{(s)} \oplus \Gamma_1^{(s)} & \text{if } (m^{(s)} = 1 \text{ and } r^{(s)} < n);\\
\left.\begin{array}{l} D_1^{(s)} = P_1^{(s)} \oplus \Gamma_1^{(s)}, \ldots, D_{m^{(s)}-1}^{(s)} = P_{m^{(s)}-1}^{(s)} \oplus \Gamma_{m^{(s)}-1}^{(s)}, \\ D_{m^{(s)}}^{(s)} = C_1^{(s)} \oplus \cdots \oplus C_{m^s-1}^{(s)} \oplus P_{m^{(s)}}^{(s)} \end{array}\right\} & \text{if } (m^{(s)} > 1 \text{ and } r^{(s)} = n);\\
\left.\begin{array}{l} D_1^{(s)} = P_1^{(s)} \oplus \Gamma_1^{(s)}, \ldots, D_{m^{(s)}-1}^{(s)} = P_{m^s-1}^{(s)} \oplus \Gamma_{m^{(s)}-1}^{(s)}, \\ D_{m^{(s)}}^{(s)} = C_1^{(s)} \oplus \cdots \oplus C_{m^s-1}^{(s)} \oplus P_{m^{(s)}}^{(s)} \oplus \Gamma_{m^{(s)}}^{(s)} \end{array}\right\} & \text{if } (m^{(s)} > 1 \text{ and } r^{(s)} < n).
\end{array}$$

Outputs:

$\gamma^{(s)}, \delta^{(s)}, C_1^{(s)}, \ldots, C_{m^{(s)}}^{(s)}$.

Note $\Gamma_i^{(s)} = \phi_{\gamma^{(s)}}(i)$. The formal analysis will be to show that the probability of $D_{m^s}^{(s)}$ and $D_{m^{(t)}}^{(t)}$ being equal is small and also that the probability of any $D_{m^s}^{(s)}$ being equal to any of the other inputs to π is also small. The form of the $D_{m^s}^{(s)}$'s show that this is indeed to be expected if we assume that $\gamma^{(s)}$, $\delta^{(s)}$ and the $C_i^{(s)}$ are independent and uniform random n-bit strings. (Since π is a permutation, these will not be independent and this will be taken care of in the analysis below.)

Note that for $m^{(s)} = 1$, the use of $\delta^{(s)}$ is quite crucial, as otherwise, one can achieve a trivial collision by setting $P_1^{(s)}$ to be equal to $\mathsf{str}^{(s)}$. (This point was missed in an earlier version of this paper.)

The next few lemmas state the required bounds on the collision and disjointness probabilities of iPHASH.

Lemma 5. *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 $m + m' - 3 \leq 2^n$. Then* $\Pr[D_m = D'_{m'}] \leq (m + m')/2^n \leq 2\max(m, m')/2^n$.

Proof. From the definition of the domain of iPMAC, $m, m' \leq 2^{n/2}$ so that $m+m'-3 \leq 2^n$. Assume without loss of generality that $m \geq m'$. First suppose $\mathsf{str} = \mathsf{str}'$ as this is the more difficult case. In this case, $\gamma = \gamma'$ and $\delta = \delta'$. We will denote these masks as γ and δ and the derived masks as Γ_i. We start by assuming that both m and m' are greater than one. The case when at least one of them is one is considered later. 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 $\mathcal{Y}$ and hence, the last expression is zero with probability $1/2^n$.

So, we can assume that either ($m > m' + 1$, $j = m' + 1$) or ($1 \leq j \leq m'$ and $m > m'$). In either case, $D_j = \phi_\gamma(j) + P_m$. 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 $D_i = P_i \oplus \phi_\gamma(i)$, $1 \leq i \leq m-1$; and $D'_k = P'_k \oplus \phi_\gamma(k)$, $1 \leq k \leq m'-1$. Let $\mathcal{E}$ be the event

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

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 $P_j \neq P'_j$, $D_j = P_j \oplus \phi_\gamma(j) \neq P'_j \oplus \phi_\gamma(j) = D'_j$ 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 $1/2^n$ by the properties of ϕ given in Definition 8. So,

$$\Pr[\mathcal{E}] \geq \left(1 - \frac{m + m' - 3}{2^n}\right).$$

We have

$$\begin{aligned} \Pr[D_m \neq D'_{m'}] &\geq \Pr[(D_m \neq D'_{m'}) \wedge \mathcal{E}] \\ &= \Pr[(D_m \neq D'_{m'}) | \mathcal{E}] \Pr[\mathcal{E}] \\ &\geq \Pr[(D_m \neq D'_{m'}) | \mathcal{E}] \times \left(1 - \frac{m + m' - 3}{2^n}\right) \end{aligned}$$

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. Note that $t \leq m+m'-3$. 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 $\chi_{t+1}(\mathcal{Y})$.

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

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

From this we have $\Pr[D_m \neq D'_{m'}] \geq (1-(m+m'-1)/2^n)$ and so $\Pr[D_m = D'_{m'}] \leq (m+m')/2^n$.

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

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

If $m = m'$, then the terms involving ϕ 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 str 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 $1 - (m + m' - 2)/2^n$. In a manner similar to the previous case, it can be shown $\Pr[D_m \neq D'_{m'}|\mathcal{E}] \geq 1 - 2/2^n$ so that we again have $\Pr[D_m \neq D'_{m'}] \leq (m + m')/2^n$.

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

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

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, $D_m \oplus D'_{m'} = \phi_\gamma(m)$ which is equal to 0 with probability $1/2^n$. 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.

Now we consider the two case which were left out earlier: $\mathsf{str} = \mathsf{str}'$ but one of m or m' is 1; and the case when $\mathsf{str} \neq \mathsf{str}'$.

$\mathsf{str} = \mathsf{str}'$ and one of m or m' is 1. If $m = m' = 1$, then $D_1 \oplus D'_1 = P_1 \oplus P'_1$. By the restriction that queries must be distinct, it follows that $\Pr[D_1 = D'_1] = 0$. If $m' = 1$ and $m > 1$, then $D'_1 = D_m \oplus P'_1 \oplus \delta$ and $D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \Lambda$, where Λ is either 0^n or Γ_m according as $r = n$ or $r < n$. In either case, δ is not involved in the expression for D_m. A simple computation now shows that $\Pr[D_m = D'_1] \leq 2m/2^n$.

$\mathsf{str} \neq \mathsf{str}'$. This is simpler than the case $\mathsf{str} = \mathsf{str}'$, since in this case (γ, γ') is uniformly distributed over $\chi_2(\mathcal{Y})$. The following two facts can be noted.

1. For any i, j, and any n-bit string arb, $\Gamma_i \oplus \Gamma'_j = \mathsf{arb}$ is the same as $\phi_\gamma(i) \oplus \phi_{\gamma'}(j) = \mathsf{arb}$ and from Definition 8, the probability of the last event is $1/(2^n - 1)$.
2. The event $\delta = \delta'$ occurs if and only if $\gamma = \gamma'$ and so in this case, the probability of $\delta = \delta'$ is 0.

These two observations directly show that $\Pr[D_m = D'_{m'}] \le 2/2^n$ for all cases other than the case when both m and m' are greater than 1 and $r = r' = n$. In the last case, $D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m$ and $D'_{m'} = C'_1 \oplus \cdots \oplus C'_{m-1} \oplus P'_m$. Now $C_1 = \pi(P_1 \oplus \Gamma_1)$ and $C'_1 = \pi(P'_1 \oplus \Gamma'_1)$. Again, from Definition 8, the probability that $P_1 \oplus \Gamma_1$ equals $P'_1 \oplus \Gamma'_1$ is at most $1/(2^n - 1)$. From this a small calculation shows that $\Pr[D_m = D'_{m'}] \le 2\max(m, m')/2^n$. □

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

Lemma 6. *Let x and x' be two distinct messages having m and m' blocks respectively. Then*

1. $\Pr[D_m = D'_i] \le 2/2^n$ *for* $1 \le i \le m' - 1$;
2. $\Pr[D_m = D_i] \le 2/2^n$ *for* $1 \le i \le m - 1$;
3. $\Pr[D_m = \mathsf{str}] \le 1/2^n$;
4. $\Pr[D_m = \gamma] \le 1/2^n$.
5. $\Pr[D_m = \mathsf{str}'] \le 1/2^n$;
6. $\Pr[D_m = \gamma'] \le 1/2^n$.

Proof. First suppose $m = 1$. Then $D_1 = P_1 \oplus \delta$ or $D_1 = P_1 \oplus \delta \oplus \phi_\gamma(1)$ according as $r = n$ or $r < n$. Points 3 to 6 follow from this. For $1 \le i \le m' - 1$, $D'_i = P'_i \oplus \Gamma'_i$ which is independent of δ. So $\Pr[D_1 = D'_i] \le 2/2^n$ which proves Point 1. For $m = 1$, Point 2 is vacuous.

If $m > 1$, then $D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m$ or $D_m = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \Gamma_m$ according as $r = n$ or $r > n$. Here $\Gamma_m = \phi_\gamma(m)$ and $m > 1$. A straightforward analysis now shows the result. □

Consequently, $\Pr[\overline{\mathsf{Pairwise\text{-}Disjoint}(x, x')}] \le (m+m')/2^n$ and $\Pr[\overline{\mathsf{Self\text{-}Disjoint}(x)}] \le 2m/2^n$. Using Theorem 2 with $\varepsilon = \varepsilon_1 = \varepsilon_2 = 2/2^n$ and noting that in this case π is a uniform random permutation gives the following result.

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

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{iPMAC}}(q, \sigma) \le \frac{(7q + 2)\sigma}{2^n}.$$

As mentioned earlier, the modification in which the first n bits is fixed to fStr will be denoted by $\mathsf{iPMAC}_{\pi,\mathsf{fStr}}$. Using Proposition 4, the PRF-bound for $\mathsf{iPMAC}_{\pi,\mathsf{fStr}}$ is $(7q + 2)(\sigma + q)/2^n$.

Table 5. Description of iPMAC and its modified version. $\mathsf{Format}(P, n)$ defines m and r.

$\mathsf{iPMAC}_\pi(P)$:	$\mathsf{miPMAC}_\pi(P)$:
1. $(\mathsf{str}, P_1, \ldots, P_m) = \mathsf{Format}(P, n)$;	1. $(\mathsf{str}, P_1, \ldots, P_m) = \mathsf{Format}(P, n)$;
2. $\gamma = \pi(\mathsf{str})$;	2. $\gamma = \pi(\mathsf{str})$;
3. for $i = 1, \ldots, m$, $\Gamma_i = \phi_\gamma(i)$;	3. for $i = 1, \ldots, m$, $\Gamma_i = \phi_\gamma(i)$;
4. $(C_1, \ldots, C_{m-1})$ $= \mathrm{ECB}_\pi(P_1 \oplus \Gamma_1, \ldots, P_{m-1} \oplus \Gamma_{m-1})$;	4. $(C_1, \ldots, C_{m-1})$ $= \mathrm{ECB}_\pi(P_1 \oplus \Gamma_1, \ldots, P_{m-1} \oplus \Gamma_{m-1})$;
5. $\mathsf{sum} = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m$;	5. $\mathsf{sum} = C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m$;
6. if $(r < n)$ then $\mathsf{sum} = \mathsf{sum} \oplus \Gamma_m$;	6. if $(r < n)$ then $\boxed{\mathsf{sum} = \mathsf{sum} \oplus \Gamma_{m+1};}$
7. if $m = 1$, then	7. if $m = 1$, then
8. $\delta = \pi(\gamma)$; $\mathsf{sum} = \mathsf{sum} \oplus \delta$;	8. $\delta = \pi(\gamma)$; $\mathsf{sum} = \mathsf{sum} \oplus \delta$;
9. $\mathsf{tag} = \pi(\mathsf{sum})$;	9. $\boxed{\mathsf{tag} = \pi(\mathsf{sum} \oplus \Gamma_1 \oplus \cdots \oplus \Gamma_{m-1});}$
return tag.	return tag.

Table 5 provides an explicit description of iPMAC_π. In the presentation of the algorithm, we have chosen clarity over efficiency. For example, in actual implementation, the masks Γ_i are not actually required to be pre-computed. These will computed as required and Γ_i will be computed from Γ_{i-1} with one application of ψ. Further, if the first n bits are fixed to fStr, then γ needs to be computed only once per session and then cached. Similarly, δ will be computed only if messages of lengths less than or equal to n are required to be processed and then also, it will be computed only once per session.

This table also shows a variant called miPMAC. There is a reason for considering miPMAC. Later when we consider authenticated encryption, we will show that the authentication function of a particular construction becomes equal to miPMAC in a very natural manner.

The differences between iPMAC and miPMAC are small and are highlighted using boxes. It is easy to argue that these changes do not affect security. The changes are in the masking of the last block. Let $\xi_m = \Gamma_1 \oplus \cdots \oplus \Gamma_{m-1}$. In miPMAC, padded last blocks are masked by $\Gamma_{m+1} \oplus \xi_m$ instead of by Γ_m as in iPMAC; full last blocks are masked only by ξ_m while in iPMAC they are not masked at all. The following observations show that the collision analysis is not affected by these changes.

1. The masking of single block messages by δ does not change and so the collision analysis for single-block messages and blocks of other messages does not change. The XOR of a padded single-block message and a full single-block message is $P_1 \oplus \Gamma_2 \oplus P'_1$, which is zero with probability $1/2^n$.
2. Suppose the number of blocks in the messages are m and m' and by the first point assume that both are greater than 1. Further suppose that $\mathsf{str} = \mathsf{str}'$, so that $\Gamma_i = \Gamma'_i$ for all i. (If $\mathsf{str} \neq \mathsf{str}'$, then the analysis is easier.)
   (a) Consider the collision analysis of the last blocks. The structure of the last blocks are as follows.

$$D_m = \begin{cases} C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \xi_m & \text{if } r = n; \\ C_1 \oplus \cdots \oplus C_{m-1} \oplus P_m \oplus \Gamma_{m+1} \oplus \xi_m & \text{if } r < n; \end{cases}$$

$$D'_{m'} = \begin{cases} C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus P'_{m'} \oplus \xi_{m'} & \text{if } r = n; \\ C'_1 \oplus \cdots \oplus C'_{m'-1} \oplus P'_{m'} \oplus \Gamma_{m'+1} \oplus \xi_{m'} & \text{if } r' < n; \end{cases}$$

   If $m = m'$, then the mask ξ_m is used to mask the last block of both messages and has no effect on the collision analysis of the last block irrespective of whether they are padded or full. Suppose $m > m'$. There are, as before, four cases for the values of r and r'. The point here is that the probability of D_m being equal to $D'_{m'}$ can be shown to be small without involving the Γs. We consider $r = r' = n$, the consideration for the other three cases being similar. In this case,

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

   where Υ is the XOR of all the terms which depend on γ and Y is other part. The analysis of the distribution of Y is exactly the same as the case $r = r' = n$ in the proof of Theorem 4. Further, Y does not depend on γ and hence Y and Υ are independent.
   This shows that the collision analysis of the last blocks remain unaffected by the additional masking done in miPMAC.
   (b) Now consider the collision analysis of a last block (with $m > 1$) and an internal block. In this case also, it can be argued that the additional masking does not make any difference.

Note that it is possible that for some m, $\Gamma_{m+1} \oplus \xi_m = \Gamma_{m+1} \oplus \Gamma_1 \oplus \cdots \oplus \Gamma_{m-1}$ is zero even for a uniform random γ. This can happen if the minimal polynomial $\tau(x)$ of ψ over $\mathbb{F}_2$ divides $x^{m+1} \oplus x^{m-1} \oplus \cdots \oplus x_1$. But, this fact does not affect the collision analysis which remains unchanged from that of iPMAC as argued above. In a nutshell, this happens because Γ_{m+1} is used to rule out collisions only when the number of blocks in the two messages are equal and the last block of one is full while the last block of the other is partial. Since, the number of blocks in the two messages are equal, ξ_ms for both the messages are also equal and they cancel out leaving only Γ_{m+1}. We are then back to the analysis of iPMAC.

By the above argument, the PRF-bound obtained for iPMAC also holds for miPMAC. Let q and $\sigma \geq q$ be positive integers. Then

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{miPMAC}}(q, \sigma) \leq \frac{(7q+2)\sigma}{2^n}. \tag{25}$$

5.2 VPMAC

Fix positive integers ℓ and n with $\ell \geq n$ and let $\mathcal{U} = \{0,1\}^\ell$ and $\mathcal{Y} = \{0,1\}^n$. Let $\rho : \mathcal{U} \to \mathcal{Y}$ 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. For an n-bit string X, by $\mathsf{bot}(X)$ we will mean the n least significant bits of X; by $\mathsf{top}(X)$ we will mean the $(\ell - n)$ most significant bits of X. We define a function VPMAC which can handle strings of lengths greater than or equal to ℓ (and of length at most $2^{\ell/2}$).

Let ϕ be a function satisfying Definition 8. Given any binary string x, with $\mathsf{len}(x) \geq \ell$, let $(\mathsf{str}, P_1, \ldots, P_m)$ be the output of $\mathsf{Format}(x, \ell)$ given in Table 2 which also defines the values of m and r. Note that $\mathsf{len}(\mathsf{str}) = \ell$ and so it is possible that the other part consisting of $P_1, \ldots, P_m$ is the empty string. Let $\gamma = \rho(\mathsf{str}) \in \mathcal{Y}$ and $\delta = \rho(\gamma || 0^{\ell - n})$.

Define VPMAC_ρ to be a function

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

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

$$D_i = \begin{cases} \phi_\gamma(i) \oplus\!\!\rightarrow P_i & 1 \leq i \leq m-1; \\ (C_1 \oplus \cdots \oplus C_{m-1}) \oplus\!\!\rightarrow P_m & i = m > 1, r = \ell; \\ (C_1 \oplus \cdots \oplus C_{m-1} \oplus \phi_\gamma(m)) \oplus\!\!\rightarrow P_m & i = m > 1, r < \ell; \\ \delta \oplus\!\!\rightarrow P_1 & i = m = 1, r = \ell; \\ (\delta \oplus \phi_\gamma(1)) \oplus\!\!\rightarrow P_1 & i = m = 1, r < \ell. \end{cases} \tag{26}$$

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.

The following result is obtained in a manner similar to that of Theorem 4, the only difference being the fact that the uniform random permutation π is replaced by the uniform random function ρ. This results in a slightly better bound.

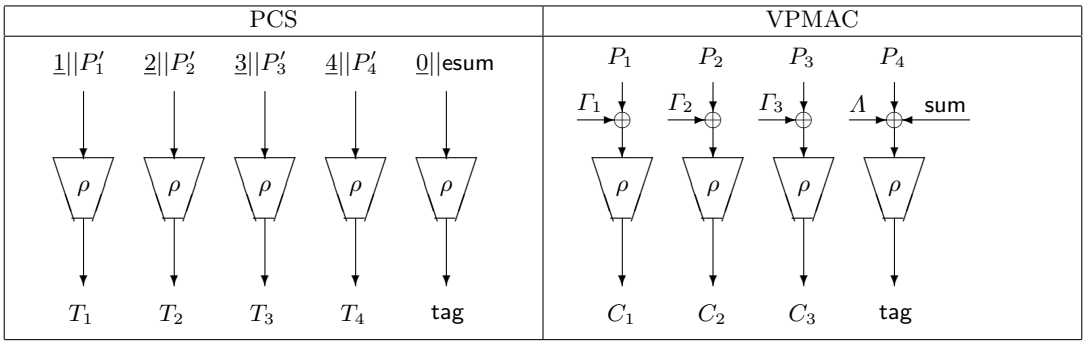

Fig. 2. Tag generation using PCS and VPMAC. In PCS, $\mathsf{esum} = T_1 \oplus T_2 \oplus T_3 \oplus T_4$. In VPMAC, $\Gamma_i = \phi_\gamma(i)$; $\Lambda = 0^n$ if the last block is full and $\Lambda = \phi_\gamma(4)$ if the last block has been padded; and $\mathsf{sum} = C_1 \oplus C_2 \oplus C_3$.

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

$$\mathbf{Adv}_{\mathrm{VPMAC}}(q, \sigma) \leq \frac{(6q+2)\sigma}{2^n}.$$

As in the case of iPMAC, Proposition 4 tells us the following. By fixing the first ℓ bits to a fixed string fStr, we obtain a PRF $\mathsf{VPMAC}_{\rho,\mathsf{fStr}}$ which can handle strings of lengths greater than or equal to 0 and has advantage upper bounded by $(6q+2)(\sigma+q)/2^n$.

Remark. The masking strategy that we use for VPMAC is not the only possible one. In fact, the tweak-based approach used for PMAC can be adapted to work with compressing functions. However, this would also require discrete logarithm computations. Since one of the objectives of this work is to do away with such computations, we do not adopt this approach.

6 Authenticated Encryption

Let $\mathcal{N}$ and $\mathcal{X}$ be finite non-empty sets of binary strings and let $\mathcal{F}_n[\mathcal{N}, \mathcal{X}]$ be the set of functions $f : \mathcal{N} \times \mathcal{X} \to \mathcal{X} \times \{0,1\}^n$ such that if $f(N, X) = (Y, \mathsf{tag})$, then $\mathsf{len}(X) = \mathsf{len}(Y)$. Here $\mathcal{N}$ is called the set of nonces. Typically, $\mathcal{N}$ will be the set of all binary strings of certain fixed length. Given a $f : \mathcal{N} \times \mathcal{X} \to \mathcal{X} \times \{0,1\}^n$, we define the following notions.

1. $f^{\mathrm{main}} : \mathcal{N} \times \mathcal{X} \to \mathcal{X}$ is defined to be $f^{\mathrm{main}}(N, X) = Y$ if $f(N, X) = (Y, \mathsf{tag})$.
2. The function f is said to be an AE-function if for every $N \in \mathcal{N}$, $f_N^{\mathrm{main}}(\cdot) \stackrel{\Delta}{=} f^{\mathrm{main}}(N, \cdot)$ is a length preserving permutation. The invertibility of f_N^{main} ensures that decryption is possible, i.e., given Y, it is possible to obtain X.
3. For an AE-function f, $\widetilde{f} : \mathcal{N} \times \mathcal{X} \to \{0,1\}^n$ is defined to be $\widetilde{f}(N, Y) = \mathsf{tag}$ if $f(N, X) = (Y, \mathsf{tag})$ for some $X \in \mathcal{X}$. Due to the invertibility of f_N^{main}, it follows that $\widetilde{f}$ is well defined. The function $\widetilde{f}$ is said to be the authentication function associated with f.
4. For an AE-function f, $f^- : \mathcal{N} \times \mathcal{X} \to \mathcal{X} \times \{0,1\}^n$ is defined to be $f^-(N, Y) = (X, \mathsf{tag})$ if $f(N, X) = (Y, \mathsf{tag})$. Note that the tag is in fact equal to $\widetilde{f}(N, Y)$.

An AE-function is required to satisfy two security properties – privacy and authenticity.

Let f be a random AE-function and f^* be a function distributed uniformly over $\mathcal{F}_n[\mathcal{N}, \mathcal{X}]$. Privacy is defined as indistinguishability from random strings. For defining privacy, an adversary $\mathcal{A}$

is assumed to have oracle access to f, i.e., for $1 \le i \le q$, $\mathcal{A}$ can adaptively query f on $(N^{(s)}, P^{(s)})$ and get back $(C^{(s)}, \mathsf{tag}^{(s)})$ in return. There is, however, a restriction on $\mathcal{A}$: the nonces of two different queries cannot be equal. Such an adversary is called nonce-respecting. Finally, $\mathcal{A}$ outputs a bit. As before, $\mathcal{A}^f \Rightarrow 1$ denotes the event that $\mathcal{A}$ produces 1 as output after interacting with the oracle f.

The advantage of $\mathcal{A}$ in breaking the privacy of f is defined to be

$$\mathbf{Adv}_f^{\text{priv}}(\mathcal{A}) = \Pr[\mathcal{A}^f \Rightarrow 1] - \Pr[\mathcal{A}^{f^*} \Rightarrow 1]. \tag{27}$$

By a (q, σ)-adversary we mean an adversary $\mathcal{A}$ which makes at most q queries and has query complexity at most σ. The resource bounded advantage $\mathbf{Adv}_f^{\text{priv}}(q, \sigma)$ is the maximum of $\mathbf{Adv}_f^{\text{priv}}(\mathcal{A})$ taken over all (q, σ)-adversaries $\mathcal{A}$.

We can think of privacy-advantage of f as the PRF-advantage of f with respect to nonce-respecting adversaries. We also define the privacy-advantage of f^{main} in a manner similar to that of (27). Note the following simple result.

Proposition 6. *Let f be an AE function. Then*

$$\mathbf{Adv}_{f^{\text{main}}}^{\text{priv}}(q, \sigma) \le \mathbf{Adv}_f^{\text{priv}}(q, \sigma).$$

In defining authenticity, an adversary $\mathcal{A}$ is given oracle access to f. Queries $(N^{(s)}, X^{(s)})$ to f are made adaptively by $\mathcal{A}$ with the restriction that no two queries are equal and that the nonces $N^{(i)}$ are distinct. The responses $(Y^{(s)}, \mathsf{tag}^{(s)})$ to the queries are provided to $\mathcal{A}$. Suppose that a total of $(q-1)$ such queries are made. Finally, $\mathcal{A}$ outputs a forgery $(N^{(q)}, Y^{(q)}, \mathsf{tag}^{(q)})$. The restriction on the forgery is that $(N^{(q)}, Y^{(q)}, \mathsf{tag}^{(q)}) \ne (N^{(s)}, Y^{(s)}, \mathsf{tag}^{(s)})$ for all $1 \le s \le q-1$. There is no restriction on $N^{(q)}$ and it can be equal to one of the earlier $N^{(s)}$'s.

If there is an $X^{(q)}$ such that $f(N^{(q)}, X^{(q)}) = (Y^{(q)}, \mathsf{tag}^{(q)})$ then $\mathcal{A}$ is said to be successful. Let $\mathsf{succ}(\mathcal{A})$ denote the event that $\mathcal{A}$ is successful. The advantage of $\mathcal{A}$ in breaking the authenticity of f is defined to be

$$\mathbf{Adv}_f^{\text{ae-auth}}(\mathcal{A}) = \Pr[\mathsf{succ}(\mathcal{A})]. \tag{28}$$

The resource bounded advantage $\mathbf{Adv}_f^{\text{ae-auth}}(q, \sigma)$ denotes the maximum of $\mathbf{Adv}_f^{\text{ae-auth}}(\mathcal{A})$ taken over all adversaries $\mathcal{A}$ making at most q queries and having query complexity at most σ.

The following result relates the authentication property of an AE-function to the privacy of f^{main} and the authenticity of $\widetilde{f}$. The function $\widetilde{f}$ maps $\mathcal{N} \times \mathcal{X}$ to $\{0,1\}^n$ and its authenticity is defined as in (13).

Proposition 7. *Given an AE function f, we have*

$$\mathbf{Adv}_f^{\text{ae-auth}}(q, \sigma) \le \mathbf{Adv}_{f^{\text{main}}}^{\text{priv}}(q, \sigma) + \mathbf{Adv}_{\widetilde{f}}^{\text{auth}}(q, \sigma).$$

Proof. Consider an adversary $\mathcal{A}$ attacking the authenticity of the AE-function f. Queries by $\mathcal{A}$ are of the form $(N^{(s)}, X^{(s)})$ and the outputs of the oracle are of the form $(Y^{(s)}, \mathsf{tag}^{(s)})$, $1 \le s \le q-1$, where q is the number of queries. The final query is a forging query of the type $(N^{(q)}, Y^{(q)}, \mathsf{tag}^{(q)})$. Let p_0 be the probability of $\mathsf{succ}(A)$.

Consider an adversary $\mathcal{B}$ which has $\widetilde{f}$ as oracle. $\mathcal{B}$ simulates $\mathcal{A}$'s queries as follows: given $(N^{(s)}, X^{(s)})$, $\mathcal{B}$ generates a uniform random string $Y^{(s)}$ of the same length as $X^{(s)}$ and queries

its oracle with $(N^{(s)}, Y^{(s)})$; in return it gets $\mathsf{tag}^{(s)}$ and sends $(Y^{(s)}, \mathsf{tag}^{(s)})$ to $\mathcal{A}$. Finally, $\mathcal{B}$ outputs whatever forgery that $\mathcal{A}$ outputs.

Let p_1 be the probability of $\mathsf{succ}(\mathcal{A})$ when the queries of $\mathcal{A}$ are simulated by $\mathcal{B}$ as mentioned above. Clearly, p_1 is also equal to $\mathsf{succ}(\mathcal{B})$. The difference between p_0 and p_1 is that Y is chosen uniformly at random instead of the being the output of f on (N, X). This shows that $p_0 - p_1 \leq \mathbf{Adv}^{\mathrm{priv}}_{f^{\mathrm{main}}}(q, \sigma)$. Now,

$$\begin{aligned}\mathbf{Adv}^{\text{ae-auth}}_{f}(q,\sigma) = p_0 &= (p_0 - p_1) + p_1 \\ &\leq \mathbf{Adv}^{\mathrm{priv}}_{f}(q,\sigma) + \mathsf{succ}(\mathcal{B}) \\ &\leq \mathbf{Adv}^{\mathrm{priv}}_{f}(q,\sigma) + \mathbf{Adv}^{\mathrm{auth}}_{\widetilde{f}}(q,\sigma).\end{aligned}$$

□

The advantage of Proposition 7 is that it reduces the task of proving the authenticity of an AE function f to that of proving the privacy of f and the authenticity of $\widetilde{f}$. Separated in this manner, the two individual tasks are easier to carry out compared to the monolithic task of directly proving the authenticity of f. At the same time, we also note that this approach may give us a slightly weaker security bound than what can be obtained directly: the degradation would be by at most a factor of 2.

Proposition 8. *Given an AE-function f, define another AE function h as follows: $h(N, X) = (Y, g(\mathsf{tag}))$, where $f(N, X) = (Y, \mathsf{tag})$ and $g : \{0,1\}^n \to \{0,1\}^t$ is a regular function. Then*

$$\mathbf{Adv}^{\text{ae-auth}}_{h}(q,\sigma) \leq \frac{1}{2^t} + \mathbf{Adv}^{\mathrm{priv}}_{f^{\mathrm{main}}}(q,\sigma) + \mathbf{Adv}^{\mathrm{prf}}_{\widetilde{f}}(q,\sigma).$$

Proof. From Propositions 2, 3 and 7 we have

$$\begin{aligned}\mathbf{Adv}^{\text{ae-auth}}_{h}(q,\sigma) &\leq \mathbf{Adv}^{\mathrm{priv}}_{h^{\mathrm{main}}}(q,\sigma) + \mathbf{Adv}^{\mathrm{auth}}_{\widetilde{h}}(q,\sigma) \\ &= \mathbf{Adv}^{\mathrm{priv}}_{f^{\mathrm{main}}}(q,\sigma) + \mathbf{Adv}^{\mathrm{auth}}_{g\circ\widetilde{f}}(q,\sigma) \\ &= \frac{1}{2^t} + \mathbf{Adv}^{\mathrm{priv}}_{f^{\mathrm{main}}}(q,\sigma) + \mathbf{Adv}^{\mathrm{prf}}_{g\circ\widetilde{f}}(q,\sigma) \\ &= \frac{1}{2^t} + \mathbf{Adv}^{\mathrm{priv}}_{f^{\mathrm{main}}}(q,\sigma) + \mathbf{Adv}^{\mathrm{prf}}_{\widetilde{f}}(q,\sigma).\end{aligned}$$

□

Note. In analysing the PRF-property of $\widetilde{f}$ we have not constrained the adversary to be nonce-respecting. The security model for authentication of AE protocol, however, requires the nonces in the queries to be distinct and only the nonce in the forgery attempt is allowed to be equal to one of the nonces in the queries. So, requiring $\widetilde{f}$ to be a PRF is an overkill. It is sufficient for $\widetilde{f}$ to satisfy a weaker security requirement as discussed below.

6.1 PRF Against Almost Nonce-Respecting Adversaries

Let f be an AE function and consider $\widetilde{f}$. An adversary attacking the PRF property of $\widetilde{f}$ has only one restriction on the queries, namely, two queries $(N^{(s)}, Y^{(s)})$ and $(N^{(t)}, Y^{(t)})$ cannot be the

same. Now suppose, that the following additional restriction is made: for $1 \leq s < t \leq q-1$, $N^{(s)} \neq N^{(t)}$. Note that there is no restriction on $N^{(q)}$ which may or may not be equal to one of $N^{(s)}$ for $1 \leq s \leq q-1$. Adversaries of the above type will be called almost nonce-respecting (ANR). (If the restriction of distinctness is also imposed on $N^{(q)}$, then the adversary is nonce-respecting.) The ANR-PRF-advantage of f with respect to an almost nonce-respecting adversary $\mathcal{A}$ is defined to be

$$\mathbf{Adv}_f^{\text{anr-prf}}(\mathcal{A}) = \Pr[\mathcal{A}^f \Rightarrow 1] - \Pr[\mathcal{A}^{f^*} \Rightarrow 1]. \tag{29}$$

The resource bounded advantage is defined as usual to be $\mathbf{Adv}_f^{\text{anr-prf}}(q, \sigma)$. In the definition of authentication security of an AE function, an adversary is actually restricted to be ANR. In view of this, the following weaker version of Proposition 8 can be obtained.

Proposition 9. *Given an AE-function f, define another AE function h as follows: $h(N, X) = (Y, g(\mathsf{tag}))$, where $f(N, X) = (Y, \mathsf{tag})$ and $g : \{0,1\}^n \to \{0,1\}^t$ is a regular function. Then*

$$\mathbf{Adv}_h^{\text{ae-auth}}(q, \sigma) \leq \frac{1}{2^t} + \mathbf{Adv}_{f^{\text{main}}}^{\text{priv}}(q, \sigma) + \mathbf{Adv}_{\widetilde{f}}^{\text{anr-prf}}(q, \sigma).$$

Suppose $f_1 : \mathcal{N} \times \mathcal{X} \to \{0,1\}^n$ and $f_2 : \mathcal{H} \to \{0,1\}^n$ are independent random functions. Consider the function $f_3 : \mathcal{N} \times \mathcal{H} \times \mathcal{X} \to \{0,1\}^n$ defined as $f_3(N, H, X) \stackrel{\Delta}{=} f_1(N, X) \oplus f_2(H)$. Since f_1 and f_2 are independent functions, it is easy to show that

$$\mathbf{Adv}_{f_3}^{\text{prf}}(q, \sigma) \leq \mathbf{Adv}_{f_1}^{\text{prf}}(q, \sigma) + \mathbf{Adv}_{f_2}^{\text{prf}}(q, \sigma). \tag{30}$$

Now consider the following more complicated scenario. Let

$$\mathcal{S} = (\mathcal{N} \times \mathcal{X}) \bigcup (\mathcal{N} \times \mathcal{H} \times \mathcal{X})$$

and consider a function $f_4 : \mathcal{S} \to \{0,1\}^n$ defined as follows:

$$\begin{aligned} f_4(N, X) &= f_1(N, X); \\ f_4(N, H, X) &= f_3(N, H, X) = f_1(N, X) \oplus f_2(H). \end{aligned}$$

The f_4 so defined is not a PRF and is easily demonstrated by four queries:

1. $(N^{(1)}, X^{(1)})$ returning $Y^{(1)} = f_1(N^{(1)}, X^{(1)})$;
2. $(N^{(1)}, H^{(1)}, X^{(1)})$ returning $Y^{(2)} = f_1(N^{(1)}, X^{(1)}) \oplus f_2(H^{(1)})$;
3. $(N^{(2)}, X^{(2)})$ with $(N^{(2)}, X^{(2)}) \neq (N^{(1)}, X^{(1)})$, returning $Y^{(3)} = f_1(N^{(2)}, X^{(2)})$;
4. $(N^{(2)}, H^{(1)}, X^{(2)})$ returning $Y^{(4)} = f_1(N^{(2)}, X^{(2)}) \oplus f_2(H^{(1)})$.

Then $f_2(H^{(1)}) = Y^{(1)} \oplus Y^{(2)} = Y^{(3)} \oplus Y^{(4)}$ showing that f_4 is not a PRF. The problem arises due to the fact that it is allowed to query f_4 on $(N^{(1)}, X^{(1)})$, $(N^{(1)}, H^{(1)}, X^{(1)})$ and $(N^{(2)}, X^{(2)})$, $(N^{(2)}, H^{(1)}, X^{(2)})$. Such an adversary is certainly not non-respecting and since two nonces have been repeated, it is also not almost nonce-respecting. The following, however, can be proved.

$$\mathbf{Adv}_{f_4}^{\text{anr-prf}}(q, \sigma) \leq \mathbf{Adv}_{f_1}^{\text{prf}}(q, \sigma) + \mathbf{Adv}_{f_2}^{\text{prf}}(q, \sigma). \tag{31}$$

In other words, if we restrict to almost nonce-respecting adversaries, then the ANR-PRF-bound for f_4 is upper bounded by the sum of the PRF-bounds for f_1 and f_2. From Proposition 9, this is sufficient to reason about the authentication security of an AE function.

Note. The discussion in this section will become relevant when we analyse the authentication of AEAD constructions.

6.2 Constructions

Two schemes for block cipher based parallel authenticated encryption are described. As in the case of authentication, the description is in terms of a uniform random permutation π of $\{0,1\}^n$. The constructions are called PAE and PAE-1. Table 6 shows the encryption algorithms while Table 7 shows the decryption algorithms. *By only* PAE *and* PAE*-1, we will denote the encryption algorithms which are AE functions.*

As in the case of presentation of the explicit algorithm for iPMAC, we have chosen clarity over efficiency and the efficiency issues mentioned in that context are also applicable here. The Γ_is would not be pre-computed, instead, Γ_i will be computed from Γ_{i-1} by a single application of ϕ. If the application requires to handle strings of length less than or equal to n, then only the value of δ will be computed.

The difference between PAE and PAE-1 is marked by boxes and consists in applying either π^{-1} or π. The difference though small has two consequences.

1. The authentication function $\widetilde{\mathsf{PAE}}$ of PAE is essentially the function miPMAC defined earlier. On the other hand, the authentication function $\widetilde{\mathsf{PAE}}$-1 is a different one. Thus, in the two cases, the authentications are achieved in different ways.
2. The encryption algorithm of PAE requires both π and π^{-1}, but, the decryption algorithm requires only π^{-1}. In contrast, the encryption algorithm of PAE-1 requires only π and the decryption algorithm requires both π and π^{-1}. This can have consequences for actual implementation. If the decryption module of the AE scheme is desired to be implemented in small size hardware then PAE is preferable, whereas if the encryption module is desired to be implemented in small size hardware, then PAE-1 is preferable.

Theorem 6. *Let q and $\sigma \geq q$ be positive integers. Then* $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{PAE}}(q,\sigma) \leq \dfrac{2(\sigma+2q)^2}{2^n}$.

Consequently, $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{PAE}^{\mathrm{main}}}(q,\sigma) \leq \dfrac{2(\sigma+2q)^2}{2^n}$.

Proof. Let $\mathcal{A}$ be a (q,σ)-adversary, i.e., $\mathcal{A}$ makes a total of q queries and provides a total of σ n-bit blocks in all the queries. This also includes the n-bit blocks for the nonces. Recall that $\mathcal{A}$ is restricted to be nonce-respecting, i.e., $\mathcal{A}$ cannot repeat a nonce.

The s-th query is of the form $(N^{(s)}, P^{(s)})$ and gets back $(C^{(s)}, \mathsf{tag}^{(s)})$ where $\mathsf{len}(P^{(s)}) = \mathsf{len}(C^{(s)})$. Note that the output of $\mathsf{Format}(P^{(s)}, n)$ is $(P^{(s)}_1, \ldots, P^{(s)}_{m^{(s)}})$ and the output of $\mathsf{Format}(C^{(s)}, n)$ is $(C^{(s)}_1, \ldots, C^{(s)}_{m^{(s)}})$.

For $1 \leq s \leq q$ and $0 \leq i \leq m^{(s)}+1$, define

$$A^{(s)}_i = \begin{cases} \gamma^{(s)} & \text{if } i = 0; \\ P^{(s)}_i \oplus \Gamma^{(s)}_i & \text{if } 1 \leq i \leq m^{(s)}-1; \\ P^{(s)}_{m^{(s)}} \oplus \Gamma^{(s)}_{m^{(s)}} & \text{if } i = m^{(s)} \text{ and } r^{(s)} = n; \\ P^{(s)}_{m^{(s)}} \oplus D^{(s)}_{m^{(s)}} & \text{if } i = m^{(s)} \text{ and } r^{(s)} < n; \\ \mathsf{tag}^{(s)} & \text{if } i = m^{(s)}+1; \end{cases}$$

Table 6. Two schemes for parallel authenticated encryption. Here N is an n-bit string and P is a binary string of length greater than or equal to 0. The call to $\mathsf{Format}(P, n)$ defines the values of m and r.

$\mathsf{PAE.Encrypt}_\pi(N, P)$:	$\mathsf{PAE\text{-}1.Encrypt}_\pi(N, P)$:
1. $(P_1, \ldots, P_m) = \mathsf{Format}(P, n)$;	1. $(P_1, \ldots, P_m) = \mathsf{Format}(P, n)$;
2. $\gamma = \pi^{-1}(N)$; $\Gamma_i = \phi_\gamma(i)$ for $1 \le i \le m$;	2. $\boxed{\gamma = \pi(N);}$ $\Gamma_i = \phi_\gamma(i)$ for $1 \le i \le m$;
3. $(C_1, \ldots, C_{m-1}) = \mathrm{ECB}_\pi(P_1 \oplus \Gamma_1, \ldots, P_{m-1} \oplus \Gamma_{m-1}) \oplus (\Gamma_1, \ldots, \Gamma_{m-1})$;	3. $(C_1, \ldots, C_{m-1}) = \mathrm{ECB}_\pi(P_1 \oplus \Gamma_1, \ldots, P_{m-1} \oplus \Gamma_{m-1}) \oplus (\Gamma_1, \ldots, \Gamma_{m-1})$;
4. if $(r = n)$ then	4. if $(r = n)$ then
5. $\quad C_m = \pi(P_m \oplus \Gamma_m) \oplus \Gamma_m$;	5. $\quad C_m = \pi(P_m \oplus \Gamma_m) \oplus \Gamma_m$;
6. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m$;	6. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m$;
7. else	7. else
8. $\quad \mathsf{tmp} = \pi^{-1}(\mathsf{bin}_n(r) \oplus \Gamma_m)$;	8. $\quad \boxed{\mathsf{tmp} = \pi(\mathsf{bin}_n(r) \oplus \Gamma_m);}$
9. $\quad D_m = P_m \oplus \mathsf{tmp}$; $T_m = \mathsf{First}_r(D_m)$;	9. $\quad D_m = P_m \oplus \mathsf{tmp}$; $T_m = \mathsf{First}_r(D_m)$;
10. $\quad C_m = T_m \| (10^{n-r-1})$;	10. $\quad C_m = T_m \| (10^{n-r-1})$;
11. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m \oplus \Gamma_{m+1}$;	11. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m \oplus \Gamma_{m+1}$;
12. if $m = 1$, then	12. if $m = 1$, then
13. $\quad \delta = \pi^{-1}(\gamma)$; $\mathsf{sum} = \mathsf{sum} \oplus \delta$;	13. $\quad \boxed{\delta = \pi(\gamma);}$ $\mathsf{sum} = \mathsf{sum} \oplus \delta$;
14. $\mathsf{tag} = \pi^{-1}(\mathsf{sum})$;	14. $\boxed{\mathsf{tag} = \pi(\mathsf{sum});}$
return $(C_1, \ldots, C_{m-1}, T_m, \mathsf{tag})$.	return $(C_1, \ldots, C_{m-1}, T_m, \mathsf{tag})$.

Table 7. Decryption algorithms for the two schemes shown in Table 6. The call to $\mathsf{Format}(C, n)$ defines the values of m and r.

$\mathsf{PAE.Decrypt}_\pi(N, C, \mathsf{tag})$:	$\mathsf{PAE\text{-}1.Decrypt}_\pi(N, C, \mathsf{tag})$:
1. $(C_1, \ldots, C_m) = \mathsf{Format}(C, n)$;	1. $(C_1, \ldots, C_m) = \mathsf{Format}(C, n)$;
2. $\gamma = \pi^{-1}(N)$; $\Gamma_i = \phi_\gamma(i)$ for $1 \le i \le m$;	2. $\gamma = \pi(N)$; $\Gamma_i = \phi_\gamma(i)$ for $1 \le i \le m$;
3. $(P_1, \ldots, P_{m-1}) = \mathrm{ECB}_{\pi^{-1}}(C_1 \oplus \Gamma_1, \ldots, C_{m-1} \oplus \Gamma_{m-1}) \oplus (\Gamma_1, \ldots, \Gamma_{m-1})$;	3. $(P_1, \ldots, P_{m-1}) = \mathrm{ECB}_{\pi^{-1}}(C_1 \oplus \Gamma_1, \ldots, C_{m-1} \oplus \Gamma_{m-1}) \oplus (\Gamma_1, \ldots, \Gamma_{m-1})$;
4. if $(r = n)$ then	4. if $(r = n)$ then
5. $\quad P_m = \pi^{-1}(C_m \oplus \Gamma_m) \oplus \Gamma_m$;	5. $\quad P_m = \pi^{-1}(C_m \oplus \Gamma_m) \oplus \Gamma_m$;
6. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m$;	6. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m$;
7. else	7. else
8. $\quad \mathsf{tmp} = \pi^{-1}(\mathsf{bin}_n(r) \oplus \Gamma_m)$;	8. $\quad \mathsf{tmp} = \pi(\mathsf{bin}_n(r) \oplus \Gamma_m)$;
9. $\quad S_m = \mathsf{First}_r(C_m \oplus \mathsf{tmp})$;	9. $\quad S_m = \mathsf{First}_r(C_m \oplus \mathsf{tmp})$;
10. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m \oplus \Gamma_{m+1}$;	10. $\quad \mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m \oplus \Gamma_{m+1}$;
11. if $m = 1$, then	11. if $m = 1$, then
12. $\quad \delta = \pi^{-1}(\gamma)$; $\mathsf{sum} = \mathsf{sum} \oplus \delta$;	12. $\quad \delta = \pi(\gamma)$; $\mathsf{sum} = \mathsf{sum} \oplus \delta$;
13. $\mathsf{tag}' = \pi^{-1}(\mathsf{sum})$;	13. $\mathsf{tag}' = \pi(\mathsf{sum})$;
14. if $(\mathsf{tag}' \ne \mathsf{tag})$ return $\perp$;	14. if $(\mathsf{tag}' \ne \mathsf{tag})$ return $\perp$;
return $(P_1, \ldots, P_{m-1}, S_m)$.	return $(P_1, \ldots, P_{m-1}, S_m)$.

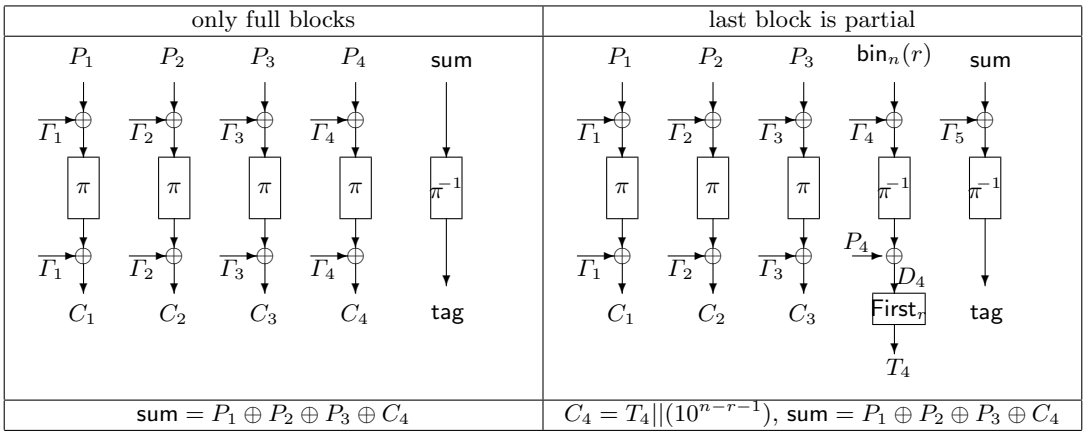

Fig. 3. Encryption using PAE: $\gamma = \pi^{-1}(N)$; $\Gamma_i = \phi_\gamma(i)$.

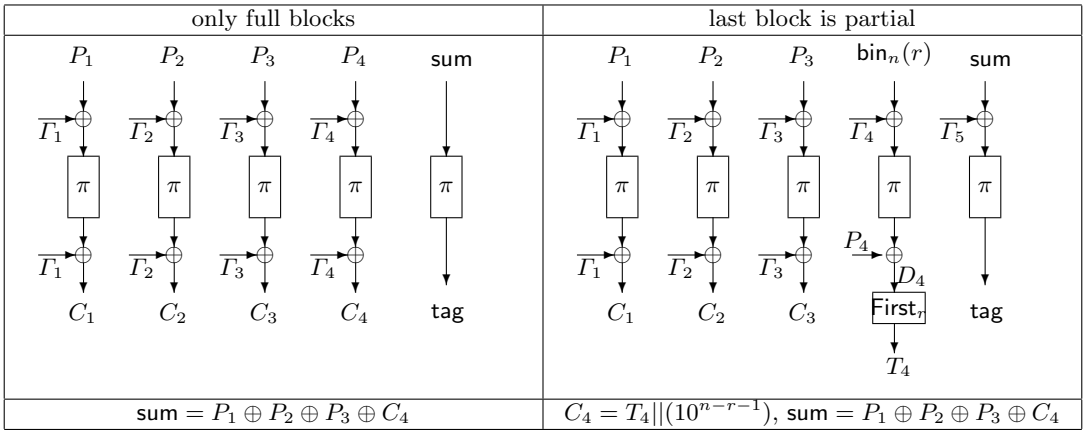

Fig. 4. Encryption using PAE-1: $\gamma = \pi(N)$; $\Gamma_i = \phi_\gamma(i)$.

$$B_i^{(s)} = \begin{cases} N^{(s)} & \text{if } i = 0; \\ C_i^{(s)} \oplus \Gamma_i^{(s)} & \text{if } 1 \le i \le m^{(s)} - 1; \\ C_{m^{(s)}}^{(s)} \oplus \Gamma_{m^{(s)}}^{(s)} & \text{if } i = m^{(s)} \text{ and } r^{(s)} = n; \\ \mathsf{bin}_n(r^{(s)}) \oplus \Gamma_{m^{(s)}}^{(s)} & \text{if } i = m^{(s)} \text{ and } r^{(s)} < n; \\ \left.\begin{array}{l} A_1^{(s)} \oplus \cdots \oplus A_{m^{(s)}-1}^{(s)} \\ \oplus \Gamma_1^{(s)} \oplus \cdots \oplus \Gamma_{m^{(s)}-1}^{(s)} \oplus C_{m^{(s)}}^{(s)} \end{array}\right\} & \text{if } i = m^{(s)} + 1, m^{(s)} > 1 \text{ and } r^{(s)} = n; \\ \left.\begin{array}{l} A_1^{(s)} \oplus \cdots \oplus A_{m^{(s)}-1}^{(s)} \\ \oplus \Gamma_1^{(s)} \oplus \cdots \oplus \Gamma_{m^{(s)}-1}^{(s)} \oplus C_{m^{(s)}}^{(s)} \oplus \Gamma_{m^{(s)}+1}^{(s)} \end{array}\right\} & \text{if } i = m^{(s)} + 1, m^{(s)} > 1 \text{ and } r^{(s)} < n; \\ C_1^{(s)} \oplus \delta^{(s)} & \text{if } i = 2, m^{(s)} = 1 \text{ and } r^{(s)} = n; \\ C_1^{(s)} \oplus \delta^{(s)} \oplus \Gamma_1^{(s)} & \text{if } i = 2, m^{(s)} = 1 \text{ and } r^{(s)} < n. \end{cases}$$

If $m^{(s)} = 1$, then $\delta^{(s)}$ is required and is obtained as $\pi^{-1}(\gamma^{(s)})$ so that $\pi(\delta^{(s)}) = \gamma^{(s)}$. We let $\mathcal{S}$ be the set of all such $\delta^{(s)}$ corresponding to $m^{(s)} = 1$ and let $\mathcal{T}$ be the set of corresponding $\gamma^{(s)}$. If $m^{(s)}$ is different from 1 for every s, then the sets $\mathcal{S}$ and $\mathcal{T}$ are empty.

From the description of PAE it follows that $B_i^{(s)} = \pi(A_i^{(s)})$. We define the following sets of random variables.

$$\mathcal{D}^{(s)} = \left\{A_0^{(s)}, \ldots, A_{m^{(s)}+1}\right\}; \quad \mathcal{R}^{(s)} = \left\{B_0^{(s)}, \ldots, B_{m^{(s)}+1}^{(s)}\right\};$$

$$\mathcal{D} = \mathcal{S} \cup \bigcup_{s=1}^{q} \mathcal{D}^{(s)}; \quad \mathcal{R} = \mathcal{T} \cup \bigcup_{s=1}^{q} \mathcal{R}^{(s)}.$$

The number of elements in either of $\mathcal{D}$ or $\mathcal{R}$ equals $\sum_{s=1}^{q}(m^{(s)}+2) + \#\mathcal{S} \leq \sigma + 2q$. Note that σ is the query complexity which is the total number of n-bit blocks provided by the adversary in all its queries and so $\sigma = \sum_{s=1}^{q}(m^{(s)}+1)$.

Assume that for any query, the quantities $C_1^{(s)}, \ldots, C_{m^{(s)}}^{(s)}, \mathsf{tag}^{(s)}$ are chosen uniformly at random and independent of the previous choices. These are then returned to the adversary (only the first $r^{(s)}$ bits of $C_{m^{(s)}}^{(s)}$ is provided). Let $\mathsf{Coll}(\mathcal{D})$ be the event that two random variables in $\mathcal{D}$ take the same value and similarly define $\mathsf{Coll}(\mathcal{R})$. Further, let $\mathsf{Coll} = \mathsf{Coll}(\mathcal{D}) \vee \mathsf{Coll}(\mathcal{R})$. As is standard, it is possible to show that

$$\mathsf{Adv}(\mathcal{A}) \leq \Pr[\mathsf{Coll}].$$

The task now reduces to bounding the probability of Coll. Note that, $\gamma^{(s)} = \pi(N^{(s)})$. Since the adversary is nonce-respecting, the values $N^{(s)}$ are distinct so that applying the uniform random permutation π on these q values ensures that each $\gamma^{(s)}$ is uniformly distributed over $\mathbb{F}$ and the joint distribution of the $\gamma^{(s)}$s is uniform over $\chi_q(\mathbb{F})$. So, the probability that two of the Γs are equal is at most $1/(2^n - 1) \leq 1/2^{n-1}$. Further, $\Gamma_i^{(s)} = \phi_{\gamma^{(s)}}(i)$, i.e., $\Gamma_i^{(s)}$ depends on the actual value of the nonce $N^{(s)}$ provided in the s-th query. The XOR-universality of ϕ shows that for $1 \leq i < j \leq 2^n - 2$ and for any $\beta \in \mathbb{F}$, $\Pr[\Gamma_i^{(s)} \oplus \Gamma_j^{(s)} = \beta] = 1/2^n$.

Using the randomness of the γs, the randomness of the Cs and the randomness of the tags, it is possible to show that for any two elements in $\mathcal{D}$, the probability that they are equal is at most $1/2^{n-1}$. This is a routine case analysis and depends on the XOR universality of ϕ. Since the number of elements in $\mathcal{D}$ is $\sigma + 2q$, the probability of $\mathsf{Coll}(\mathcal{D})$ is at most $(\sigma + 2q)(\sigma + 2q - 1)/(2 \times 2^{n-1})$. In a similar manner, the same bound on the probability of $\mathsf{Coll}(\mathcal{R})$ can be obtained so that $\Pr[\mathsf{Coll}] \leq (\sigma + 2q)^2/2^{n-1}$. □

Usually authentication of an AE scheme is difficult to analyse. Proposition 7, however, makes the task easier. It reduces the task of arguing about authentication of the entire scheme to the task of analysing the privacy of the scheme and the PRF-property of the associated authentication function. Theorem 6 bounds the privacy advantage of $\mathsf{PAE}^{\mathrm{main}}$. So, we only have to analyse the PRF-property of $\widetilde{\mathsf{PAE}}$.

This task is made easy by the simple observation that $\widetilde{\mathsf{PAE}}_\pi$ is equal to $\mathsf{miPMAC}_{\pi^{-1}}$. In fact, PAE was designed so as to achieve this equality. Checking this equality is quite routine and can be carried out by an inspection of the algorithms given in Tables 6 and 5. Table 7 provides the description of the decryption algorithm of PAE. Letting $T_i = \pi^{-1}(C_i \oplus \Gamma_i)$, we have

$$P_1 \oplus \cdots \oplus P_{m-1} = T_1 \oplus \cdots \oplus T_{m-1} \oplus \Gamma_1 \oplus \cdots \oplus \Gamma_{m-1}.$$

Thus the quantity $\Gamma_1 \oplus \cdots \oplus \Gamma_{m-1}$ is part of sum in both $\widetilde{\mathsf{PAE}}_\pi$ and $\mathsf{miPMAC}_{\pi^{-1}}$ The masking of a padded block by Γ_{m+1} is also present in both cases. In the algorithm in Table 7, consider only the part required to regenerate the tag, i.e., to compute tag'. This means that Steps 8 and 9 can be omitted and the algorithm ends at Step 13. Now, if we replace each occurrence of π^{-1} by π in the reduced algorithm, then we get exactly $\mathsf{miPMAC}_{\pi^{-1}}$.

As a result of this observation, the PRF-bound for miPMAC (which is the PRF-bound for iPMAC) is also the PRF bound for $\widetilde{\mathsf{PAE}}$. This is formally stated in the following result.

Proposition 10. *For every $N \in \{0,1\}^n$ and binary string C, with $\mathsf{len}(C) \geq 0$,*

$$\widetilde{\mathsf{PAE}}_\pi(N, C) = \mathsf{miPMAC}_{\pi^{-1}}(N, C).$$

Consequently, for $q > 0$ and $\sigma \geq q$,

$$\mathbf{Adv}^{\mathrm{prf}}_{\widetilde{\mathsf{PAE}}}(q, \sigma) = \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{miPMAC}}(q, \sigma) \leq \frac{\sigma(7q+2)}{2^n}.$$

This also answers a question attributed to Rivest in [34] which asks whether an AE scheme such as OCB can be used for authentication. The answer in case of PAE is simple and straightforward. Fix an n-bit string fStr and then the function $\widetilde{\mathsf{PAE}}_\pi$ is a PRF. This function can be used for authentication in the usual manner.

For $1 \leq t \leq n$, let t-PAE denote the AE function obtained from PAE by truncating the tag to (the first) t bits. So, n-PAE is in fact PAE. The privacy of t-PAE follows from the privacy of PAE. The authenticity of t-PAE is given by the following result.

Theorem 7. *Let $\sigma \geq q \geq 1$. Then*

$$\mathbf{Adv}^{\text{ae-auth}}_{t\text{-}\mathsf{PAE}}(q, \sigma) \leq \frac{1}{2^t} + \frac{2(\sigma + 2q)^2}{2^n} + \frac{\sigma(7q+2)}{2^n}.$$

Proof. Using Proposition 8, we have

$$\mathbf{Adv}^{\text{ae-auth}}_{t\text{-}\mathsf{PAE}}(q, \sigma) \leq \frac{1}{2^t} + \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PAE}^{\mathrm{main}}}(q, \sigma) + \mathbf{Adv}^{\mathrm{prf}}_{\widetilde{\mathsf{PAE}}}(q, \sigma).$$

Now the result follows from Theorem 6 and Proposition 10. □

The privacy of PAE-1 follow in a similar manner and the same bound holds. Following our approach of authentication analysis, we need to study the PRF-property of $\widetilde{\mathsf{PAE}}$-1. This function is shown in Table 8. Note that $\widetilde{\mathsf{PAE}}$-1 uses both π and π^{-1} which is unlike $\widetilde{\mathsf{PAE}}$ which uses only π^{-1}. Thus, the authentication functions of $\widetilde{\mathsf{PAE}}$ and $\widetilde{\mathsf{PAE}}$-1 are different. In fact, the authentication function for OCB is similar to that of $\widetilde{\mathsf{PAE}}$-1 in the sense that OCB also uses both π and π^{-1} in the decryption algorithm. This makes the manner in which authentication is achieved in OCB rather different from the manner in which it is achieved in $\widetilde{\mathsf{PAE}}$.

The differences between $\widetilde{\mathsf{PAE}}$ and $\widetilde{\mathsf{PAE}}$-1 are in Steps 1, 8 and 9, where π is used instead of π^{-1}. For producing the masks γ and δ it does not matter whether π or π^{-1} is applied. For the two functions, tag is produced by applying π^{-1} or π. This also does not cause any additional difficulty. In each case, the argument boils down to showing that the different values of $\mathsf{sum} \oplus \Gamma_1 \oplus \cdots \oplus \Gamma_{m-1}$ are distinct and are also different from the different values of $C_i \oplus \Gamma_i$. This analysis remains the same for both algorithms and so we omit the details. The final result on PAE-1 is given below and the bounds are the same as that of PAE.

Table 8. Description of $\widetilde{\mathsf{PAE}}$-1. The call to $\mathsf{Format}(C, n)$ defines m and r.

$\widetilde{\mathsf{PAE}}\text{-}1_\pi(N, C)$:
1. $\gamma = \pi(\mathsf{fStr})$;
2. for $i = 1$ to m $\Gamma_i = \phi_\gamma(i)$;
3. $(C_1, \ldots, C_m) = \mathsf{Format}(C, n)$;
4. $(P_1, \ldots, P_{m-1})$
   $= \mathrm{ECB}_{\pi^{-1}}(C_1 \oplus \Gamma_1, \ldots, C_{m-1} \oplus \Gamma_{m-1})$;
5. $\mathsf{sum} = P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m$;
6. if $(r < n)$ then $\mathsf{sum} = \mathsf{sum} \oplus \Gamma_{m+1}$;
7. if $m = 1$, then
8. $\quad \delta = \pi(\gamma)$; $\mathsf{sum} = \mathsf{sum} \oplus \delta$;
9. $\mathsf{tag} = \pi(\mathsf{sum} \oplus \Gamma_1 \oplus \cdots \oplus \Gamma_{m-1})$;

return tag.

Theorem 8. *Let $\sigma \geq q \geq 1$. Then*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{PAE}\text{-}1}(q, \sigma) \leq \frac{2(\sigma + 2q)^2}{2^n},$$
$$\mathbf{Adv}^{\text{ae-auth}}_{t\text{-}\mathsf{PAE}\text{-}1}(q, \sigma) \leq \frac{1}{2^t} + \frac{(\sigma + 2q)^2 + \sigma^2}{2^n} + \frac{\sigma(7q + 2)}{2^n}.$$

7 Authenticated Encryption with Associated Data

In many applications (such as encryption of IP packets), along with the message and the nonce, there is an additional binary string called the associated data (or header). The requirement is to authenticate the header but not to encrypt it. The input to the encryption algorithm is a triple (H, N, P), where H is the header, N is the nonce, P is the message and the output is (C, tag), where C is the encryption of P and tag authenticates both H and P.

The notion of AE can be easily extended to obtain the formal definition of an AEAD scheme. Let $\mathcal{H}, \mathcal{N}$ and $\mathcal{X}$ be sets of binary strings and let $\mathcal{F}_n[\mathcal{N}, \mathcal{H}, \mathcal{X}]$ be the set of functions $f : \mathcal{N} \times \mathcal{H} \times \mathcal{X} \to \mathcal{X} \times \{0,1\}^n$ such that if $f(N, H, P) = (C, \mathsf{tag})$, then $\mathsf{len}(P) = \mathsf{len}(C)$. Here, $\mathcal{H}$ is the set of all possible headers and $\mathcal{N}$ is the set of all possible nonces. Typically, $\mathcal{H}$ can consist of variable length strings while $\mathcal{N}$ is $\{0,1\}^n$.

If we define $\mathcal{N}' = \mathcal{N} \times \mathcal{H}$ to be the set of nonces, then we go back to the formal framework for AE functions. In this case, we have the set of nonces $\mathcal{N}'$ to consist of possibly variable length strings. The security notions of privacy and authentication for $\mathcal{F}_n[\mathcal{N}', \mathcal{X}]$ are exactly the notions for $\mathcal{F}_n[\mathcal{N}, \mathcal{H}, \mathcal{X}]$. These coincide exactly with the security notion of AEAD schemes introduced in [34]. We will use the notation $\mathbf{Adv}^{\text{aead-auth}}$ to denote the authentication security of an AEAD scheme.

In this case, the query complexity σ also counts the number of n-bit blocks formed from the headers provided as part of the different queries. We divide the query complexity into two parts σ_H and σ_P, where σ_H is the number of n-bit blocks obtained from the headers and σ_P is the number of n-bit blocks obtained from the nonces and the actual messages.

A simple AEAD scheme can be obtained by combining PAE and iPMAC and we call this PAEAD. A similar construction is also obtained by combining PAE-1 and iPMAC which we call PAEAD-1. The descriptions are given in Tables 9 and 10. A t-bit tag is produced and consequently, we will

refer to the PAEAD scheme as t-PAEAD scheme and similarly for PAEAD-1. Considering PAE to be an AE-function, the notation PAE^- is defined as in Section 6. Similarly for PAE-1.

Note. PAEAD requires both π and π^{-1} during encryption and only π^{-1} during decryption; whereas PAEAD-1 requires only π during encryption and both π and π^{-1} during decryption. This difference is inherited from a similar difference between PAE and PAE-1 and the fact that iPMAC uses π^{-1} in PAEAD whereas iPMAC uses π in PAEAD-1. For implementation in hardware or resource constrained devices, the decryption algorithm of PAEAD will be smaller while the encryption algorithm of PAEAD-1 will be smaller. The actual strategy to be adopted will depend on the application.

Table 9. Parallel AEAD schemes. PAEAD is obtained by combining PAE and iPMAC, while, PAEAD-1 is obtained by combining PAE-1 and iPMAC. Here $1 \le t \le n$ is a fixed value; $g : \{0,1\}^n \to \{0,1\}^t$ is a regular function; and fStr is a fixed n-bit string.

$\mathsf{PAEAD.Encrypt}_{\pi,\mathsf{fStr}}(N,H,P)$	$\mathsf{PAEAD\text{-}1.Encrypt}_{\pi,\mathsf{fStr}}(N,H,P)$
1. if H is null, return $\mathsf{PAE.Encrypt}_\pi(N,P)$;	1. if H is null, return $\mathsf{PAE\text{-}1.Encrypt}_\pi(N,P)$;
2. $(C,\mathsf{tag}_1) = \mathsf{PAE.Encrypt}_\pi(N,P)$;	2. $(C,\mathsf{tag}_1) = \mathsf{PAE\text{-}1.Encrypt}_\pi(N,P)$;
3. $\upsilon = \pi^{-1}(\mathsf{fStr})$;	3. $\upsilon = \pi(\mathsf{fStr})$;
4. $\mathsf{tag}_2 = \mathsf{iPMAC}_{\pi^{-1},\upsilon}(H)$;	4. $\mathsf{tag}_2 = \mathsf{iPMAC}_{\pi,\upsilon}(H)$;
5. return $(C, g(\mathsf{tag}_1 \oplus \mathsf{tag}_2))$.	5. return $(C, g(\mathsf{tag}_1 \oplus \mathsf{tag}_2))$.

Table 10. Decryption algorithms for the schemes shown in Table 9.

$\mathsf{PAEAD.Decrypt}_{\pi,\mathsf{fStr}}(N,H,C,\mathsf{tag})$	$\mathsf{PAEAD\text{-}1.Decrypt}_{\pi,\mathsf{fStr}}(N,H,C,\mathsf{tag})$
1. if H is null then return $\mathsf{PAE.Decrypt}_\pi(N,C,\mathsf{tag})$;	1. if H is null then return $\mathsf{PAE\text{-}1.Decrypt}_\pi(N,C,\mathsf{tag})$;
2. $(P,\mathsf{tag}_1) = \mathsf{PAE}^-_\pi(N,C)$;	2. $(P,\mathsf{tag}_1) = \mathsf{PAE\text{-}1}^-_\pi(N,C)$;
3. $\upsilon = \pi^{-1}(\mathsf{fStr})$;	3. $\upsilon = \pi(\mathsf{fStr})$;
4. $\mathsf{tag}_2 = \mathsf{iPMAC}_{\pi^{-1},\upsilon}(H)$;	4. $\mathsf{tag}_2 = \mathsf{iPMAC}_{\pi,\upsilon}(H)$;
5. if $(\mathsf{tag} \ne g(\mathsf{tag}_1 \oplus \mathsf{tag}_2))$ return $\perp$;	5. if $(\mathsf{tag} \ne g(\mathsf{tag}_1 \oplus \mathsf{tag}_2))$ return $\perp$;
6. return P.	6. return P.

Privacy of the construction is easy to obtain and the analysis is similar to that of PAE.

Theorem 9. *Let q and $\sigma \ge q$ be positive integers. Then* $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{PAEAD}}(q,\sigma) \le \dfrac{2(\sigma+2q)^2}{2^n}$.

Consequently, $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{PAEAD}^{\mathrm{main}}}(q,\sigma) \le \dfrac{2(\sigma+2q)^2}{2^n}$.

For authentication we need to consider the function $\widetilde{\mathsf{PAEAD}}$. Let $\mathsf{PAEAD}_{\pi,\mathsf{fStr}}(N,H,P) = (C,\mathsf{tag})$ and $\upsilon = \pi^{-1}(\mathsf{fStr})$. Then from the definition of PAEAD, the following holds.

- If H is null, then

$$\left.\begin{aligned}\widetilde{\mathsf{PAEAD}}_{\pi,\delta}(N,H,P) &= \widetilde{\mathsf{PAE}}_\pi(N,P)\\ &= \mathsf{miPMAC}_{\pi^{-1}}(N,C).\end{aligned}\right\} \tag{32}$$

- If H is not null, then

$$\left.\begin{aligned}\widetilde{\mathsf{PAEAD}}_{\pi,\mathsf{fStr}}(N,H,P) &= \widetilde{\mathsf{PAE}}_\pi(N,C) \oplus \mathsf{iPMAC}_{\pi^{-1},\upsilon}(H)\\ &= \mathsf{miPMAC}_{\pi^{-1}}(N,C) \oplus \mathsf{iPMAC}_{\pi^{-1}}(,\upsilon,H)\end{aligned}\right\} \tag{33}$$

Similar equations can be written for PAEAD-1.

The analysis of $\widetilde{\mathsf{PAEAD}}$ is based on ideas in Section 6.1. The functions miPMAC and iPMAC are both PRFs. However, they are not independent functions in $\widetilde{\mathsf{PAEAD}}$, since the same π^{-1} is used for both of them. We will see how to tackle this difficulty a bit later and for the moment suppose that these are independent. Then using (31), we get a upper bound on the ANR-PRF-advantage of $\widetilde{\mathsf{PAEAD}}$. Using Proposition 9 this is sufficient to show the authentication security bound of t-PAEAD.

Now we turn to the question of how the issue of non-independence of miPMAC and iPMAC can be tackled. Let $\mathcal{E}$ be the event that the set of inputs to π^{-1} in miPMAC is disjoint from the set of inputs to π^{-1} in iPMAC. (Consequently, the set of inputs to π in miPMAC will also be disjoint from the set of inputs to π in iPMAC.) Then the PRF bounds for the individual functions would hold and using standard arguments we obtain

$$\mathbf{Adv}^{\text{anr-prf}}_{\widetilde{\mathsf{PAEAD}}}(q,\sigma) \leq \mathbf{Adv}^{\text{prf}}_{\mathsf{miPMAC}}(q,\sigma) + \mathbf{Adv}^{\text{prf}}_{\mathsf{iPMAC}}(q,\sigma) + \Pr[\overline{\mathcal{E}}]. \tag{34}$$

Proposition 9 gives

$$\begin{aligned}\mathbf{Adv}^{\text{aead-auth}}_{t\text{-}\mathsf{PAEAD}}(q,\sigma) &\leq \frac{1}{2^t} + \mathbf{Adv}^{\text{priv}}_{\mathsf{PAEAD}^{\text{main}}}(q,\sigma) + \mathbf{Adv}^{\text{anr-prf}}_{\widetilde{\mathsf{PAEAD}}}(q,\sigma) \\ &\leq \frac{1}{2^t} + \mathbf{Adv}^{\text{priv}}_{\mathsf{PAEAD}^{\text{main}}}(q,\sigma) \\ &\quad + \mathbf{Adv}^{\text{prf}}_{\mathsf{miPMAC}}(q,\sigma) + \mathbf{Adv}^{\text{prf}}_{\mathsf{iPMAC}}(q,\sigma) + \Pr[\overline{\mathcal{E}}]. \end{aligned} \tag{35}$$

The task, thus, reduces to bounding $\Pr[\mathcal{E}]$. The event $\mathcal{E}$ represents the separation of the inputs for π^{-1} in the message and header part. The literature provides different techniques for such separation of inputs. These include using independent keys [34], and using tweakable block ciphers [35].

In our case, however, this is achieved differently. In the PAE part, the masks are obtained from γ which is obtained as $\pi^{-1}(N)$. On the other hand, in iPMAC part the masks are obtained from $\pi^{-1}(\upsilon) = \pi^{-1}(\pi^{-1}(\mathsf{fStr}))$. Since, the probability that N is equal to $\pi^{-1}(\mathsf{fStr})$ is $1/2^n$, we obtain an effective separation of the masks. We consider this in more details.

First consider the inputs and outputs to π^{-1} determined by $\mathsf{miPMAC}_{\pi^{-1}}(N^{(s)}, C^{(s)})$. For the s-th query, let $A_i^{(s)}$ and $B_i^{(s)}$ $(1 \leq i \leq m^{(s)})$ be the different inputs and outputs to π, so that $\pi^{-1}(B_i^{(s)}) = A_i^{(s)}$. The expressions for $A_i^{(s)}$ and $B_i^{(s)}$ are given in the proof of Theorem 6. Note that each $B_i^{(s)}$ is masked with either $\delta^{(s)}$ or with the XOR of one or more of the $\Gamma_i^{(s)}$s.

Next consider the inputs and outputs to π^{-1} determined by $\mathsf{iPMAC}_{\pi^{-1}}(\upsilon, H^{(s)})$. Such calls are made only if $H^{(s)}$ is non-null. Let the number of n-bit blocks in $H^{(s)}$ be $k^{(s)}$ and let the length of the last block before padding be $p^{(s)}$. Denote the blocks as $H_1^{(s)}, \ldots, H_{k^{(s)}}^{(s)}$. These blocks are the output of $\mathsf{Format}(H^{(s)}, n)$ which also defines the values of $k^{(s)}$ and $p^{(s)}$. Let $T_i^{(s)} = \pi^{-1}(H_i^{(s)})$ for $1 \leq i \leq k^{(s)} - 1$; and let $\mathsf{iPMAC}_{\pi^{-1}}(\upsilon, H^{(s)})$ be denoted by $\mathsf{htag}^{(s)}$.

Let $\omega = \pi^{-1}(\upsilon)$, $\vartheta = \pi^{-1}(\omega)$ and $\Omega_i = \phi_\omega(i)$. Since fStr does not depend on the queries, neither do the Ω_is or υ or ϑ. For $1 \leq s \leq q$, if $H^{(s)}$ is non-null, then let $E_i^{(s)}$ for $1 \leq i \leq k^{(s)}$ be the different inputs to π and $F_i^{(s)}$ be the different outputs of π (and so are inputs to π^{-1}), i.e., $\pi(E_i^{(s)}) = F_i^{(s)}$.

The different $F_i^{(s)}$s are as follows.

$$\begin{array}{ll}
H_1^{(s)} \oplus \delta^{(s)} & \text{if } (k^{(s)} = 1 \text{ and } r^{(s)} = n); \\
H_1^{(s)} \oplus \delta^{(s)} \oplus \Gamma_1^{(s)} & \text{if } (k^{(s)} = 1 \text{ and } p^{(s)} < n); \\
\left.\begin{array}{l} H_1^{(s)} \oplus \Gamma_1^{(s)}, \ldots, H_{k^s-1}^{(s)} \oplus \Gamma_{k^{(s)}-1}^{(s)}, \\ E_1^{(s)} \oplus \cdots \oplus E_{k^s-1}^{(s)} \oplus H_{m^{(s)}}^{(s)} \end{array}\right\} & \text{if } (k^{(s)} > 1 \text{ and } p^{(s)} = n); \\
\left.\begin{array}{l} H_1^{(s)} \oplus \Gamma_1^{(s)}, \ldots, H_{k^s-1}^{(s)} \oplus \Gamma_{k^{(s)}-1}^{(s)}, \\ E_1^{(s)} \oplus \cdots \oplus E_{k^s-1}^{(s)} \oplus H_{k^{(s)}}^{(s)} \oplus \Gamma_{k^{(s)}}^{(s)} \end{array}\right\} & \text{if } (k^{(s)} > 1 \text{ and } p^{(s)} < n).
\end{array}$$

Note that other than the case when $k^{(s)} > 1$ and $p^{(s)} = n$, in all other cases, either ϑ, or some Ω_i or a XOR of Ωs occur as a component of $F_i^{(s)}$. We are interested in the event $\overline{\mathcal{E}}$ which holds if one of the following occur.

1. Some $B_j^{(t)}$ is equal to some $F_{k^{(s)}}^{(s)}$, where $k^{(s)} > 1$ and $p^{(n)} = n$.
2. Some $B_j^{(t)}$ is equal to some $F_i^{(s)}$,
3. Some $B_j^{(t)}$ is equal to either υ or ω or ϑ.

Each $B_j^{(t)}$ has a component which is either $\delta^{(t)}$, or $\Gamma_j^{(t)}$ or a XOR of some of the $\Gamma^{(t)}$s. For a fixed j, $\Gamma_j^{(t)}$ uniquely determines $\gamma^{(t)}$. Similarly, for a fixed i, Ω_i uniquely determines ω. Here both i and j are greater than 1.

Consider the event $\Gamma_j^{(t)} = \upsilon$, i.e., $\phi_{\gamma^{(t)}}(j) = \upsilon$. If $N^{(t)} = \mathsf{fStr}$, then $\gamma^{(t)} = \upsilon$; since $j \geq 1$, from Definition 8, we have $\Pr[\phi_{\gamma^{(t)}}(j) = \upsilon] = 1/2^n$. If $N^{(t)} \neq \mathsf{fStr}$, then since $\gamma^{(t)} = \pi^{-1}(N^{(t)})$ and $\upsilon = \pi^{-1}(\mathsf{fStr})$, the pair $(\gamma^{(t)}, \upsilon)$ is uniformly distributed over $\chi_2(\mathbb{F})$. Again from Definition 8, it follows that $\Pr[\phi_{\gamma^{(t)}}(j) = \upsilon] = 1/(2^n - 1)$. So, in both cases, $\Pr[\Gamma_j^{(t)} = \upsilon] \leq 1/(2^n - 1)$.

Now consider the event $\Gamma_j^{(t)} = \omega$, i.e., $\phi_{\gamma^{(t)}}(j) = \omega$. The analysis is similar, the difference being that in this case $\omega = \pi^{-1}(\upsilon)$ and so the event $N^{(t)} = \upsilon$ holds with probability $1/2^n$. Using this it is possible to show that $\Pr[\Gamma_j^{(t)} = \upsilon] \leq 1/2^{n-1}$. A similar analysis holds for the events $\Gamma_j^{(t)} = \vartheta$ and $\Gamma_j^{(t)} = \Omega_i$.

These show that the events in Points 2 and 3 above hold with probability at most $1/2^{n-1}$. For the event in Point 1, $F_{k^{(s)}}^{(s)} = E_1^{(s)} \oplus \cdots \oplus E_{k^s-1}^{(s)} \oplus H_{m^{(s)}}^{(s)}$. Since $k^{(s)} > 1$, there is at least one $E_i^{(s)}$ in the expression for $F_{k^{(s)}}^{(s)}$. The probability that this is equal to $\Gamma_j^{(t)}$ can again be shown to be bounded above by $1/2^{n-1}$. So, the event in Point 1 also holds with probability at most $1/2^{n-1}$.

As a result of this analysis, we obtain $\Pr[\mathcal{E}] \leq \sigma_H \sigma_P / 2^{n-1} \leq \sigma^2 / 2^{n-1}$. This finally leads to the following result.

Theorem 10. *Let $\sigma \geq q \geq 1$. Then*

$$\mathbf{Adv}^{\text{aead-auth}}_{t\text{-}\mathsf{PAEAD}}(q, \sigma) \leq \frac{1}{2^t} + \frac{1}{2^{n-1}} \times \left(\sigma(2\sigma + 11q + 2) + 4q^2\right).$$

The analysis of PAEAD-1 is similar and results in the same bounds.

Theorem 11. *Let $\sigma \geq q \geq 1$. Then*

$$\mathbf{Adv}^{\text{priv}}_{\mathsf{PAEAD\text{-}1}}(q,\sigma) \leq \frac{2(\sigma+2q)^2}{2^n};$$
$$\mathbf{Adv}^{\text{aead-auth}}_{t\text{-}\mathsf{PAEAD\text{-}1}}(q,\sigma) \leq \frac{1}{2^t} + \frac{1}{2^{n-1}} \times \left(\sigma(2\sigma+11q+2)+4q^2\right).$$

8 Discussion and Comparison

There are several [16, 12, 35] other single-pass schemes for authenticated encryption with or without the option of supporting associated data. These include sequential as well as parallelizable designs. Among these, the most famous is OCB [35] which is a parallelizable design. All parallel schemes, including the ones proposed in this work, share a common feature. Each message block is masked before being encrypted by the block cipher and the output is also masked. There are differences in the manner that the masks are generated and in the way a possible partial last block is handled.

Authentication and authenticated encryption. Our basic premise is that the authentication security of an AE scheme can essentially be derived from the PRF-property of an associated function. As a result, we first designed an authentication scheme and showed that it is a PRF. Then the AE scheme was designed so that the authentication function of the AE scheme can be easily seen to be (a simple modification of) the previously designed PRF. This makes the analysis of authentication security of the AE scheme almost trivial. We note that in schemes such as IACBC, IAPM [16] and OCB [35], it is the analysis of authentication security which is the complicated part.

Avoiding design stage discrete logarithm computation. It has already been mentioned that our approach does not require the computation of any discrete logarithms. Further, avoiding this computation does not cause any slow down in the generation of the masks. In any implementation, the representation of $\mathbb{F}$ as a tower field will be provided by two polynomials as described in Section 4. Changing these to any other pair (subject to the second one being primitive over the intermediate field) will provide a secure algorithm. Thus, the new AEAD schemes provide a *family of easily reconfigurable* designs with the same efficiency.

Efficiency. The masks in the new AEAD schemes are to be generated using the word oriented LFSR approach described in Section 4. OCB, on the other hand, uses the powering up method which is about two times slower than the word oriented LFSR approach. As discussed in Section 4, this leads to a small speed-up in the overall time for processing a message.

More Variants. For OCB, the encryption algorithm requires only E_K and the decryption algorithm requires both E_K and E_K^{-1}. Here two AE schemes have proposed – PAE and PAE-1 as also two AEAD schemes – PAEAD and PAEAD-1. For PAE and PAEAD the encryption algorithms require both E_K and E_K^{-1} whereas the decryption algorithms require only E_K^{-1}. In contrast, for PAE-1 and PAEAD-1 the encryption algorithms require only E_K whereas the decryption algorithms require both E_K and E_K^{-1}. OCB corresponds to the second case. For hardware and resource constrained implementations, the first case will lead to a smaller decryption module whereas the second case will lead to a smaller encryption module. Efficiency and security for both cases are same. This aspect provides a designer with more flexible choices and underlines the fact that the new approach has significant differences to OCB.

Handling of the last block. If the message consists of only full blocks, then the new schemes do not distinguish between the last block and the other blocks. The distinction arises only when the last block is partial. For many applications, messages consist of only full blocks and the new schemes will be simpler to implement for such applications. In contrast, OCB handles the last block differently irrespective of whether it is full or partial. Thus, an inherent asymmetry is built into the design even when the application has messages consisting of only full blocks.

9 Conclusion

We have analysed pseudo-random functions for use in symmetric key message authentication. Starting from a useful result by Vaudenay [40] and Bernstein [6], we proved a general result on bounding the advantage of a PRF built using a uniform random permutation or a uniform random function. This result is used to analyse parallelizable PRF constructions which improve upon existing constructions.

We took a re-look at the problem of constructing and analysing AE and AEAD schemes. On the theoretical issue, we highlighted several subtle aspects of showing authentication of an AE scheme and how to use a simple masking technique to securely combine an AE and a authentication scheme. From a more practical point of view, the new constructions that we describe offer certain improvements over the previously known constructions.

Acknowledgments

We would like to thank Mridul Nandi for pointing out Lemma 22 of [40] to us. We also thank the anonymous reviewers for their comments.

References

1. http://csrc.nist.gov/CryptoToolkit/modes/.
2. 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.
3. Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.
4. 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.
5. Mihir Bellare, Phillip Rogaway, and David Wagner. The EAX mode of operation. In Bimal K. Roy and Willi Meier, editors, *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2004.
6. Daniel J. Bernstein. How to stretch random functions: The security of protected counter sums. *J. Cryptology*, 12(3):185–192, 1999.
7. Daniel J. Bernstein and Peter Schwabe. New AES software speed records. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT*, volume 5365 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2008.
8. John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Knudsen [19], pages 384–397.
9. 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.
10. Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES – The Advanced Encryption Standard (Information Security and Cryptography)*. Springer, Heidelberg, 2002.

11. 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.
12. Virgil D. Gligor and Pompiliu Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. In Mitsuru Matsui, editor, *FSE*, volume 2355 of *Lecture Notes in Computer Science*, pages 92–108. Springer, 2001.
13. Shay Gueron. White paper on intel's advanced encryption standard (AES) instructions set. April 2003. `http://software.intel.com/en-us/articles/advanced-encryption-standard-aes-instructions-set/`.
14. Version 4 Internet Protocol. From Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/IPv4`.
15. 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.
16. Charanjit S. Jutla. Encryption modes with almost free message integrity. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 529–544. Springer, 2001.
17. 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.
18. Jonathan Katz and Moti Yung. Complete characterization of security notions for probabilistic private-key encryption. In *STOC*, pages 245–254, 2000.
19. 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.
20. R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications, revised edition.* Cambridge University Press, 1994.
21. 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.
22. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
23. Stefan Lucks. Two-pass authenticated encryption faster than generic composition. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes in Computer Science*, pages 284–298. Springer, 2005.
24. Mitsuru Matsui and Junko Nakajima. On the power of bitslice implementation on Intel Core2 processor. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 121–134. Springer, 2007.
25. Ueli M. Maurer. Indistinguishability of random systems. In Knudsen [19], pages 110–132.
26. David A. McGrew and John Viega. The security and performance of the Galois/Counter Mode (GCM) of operation. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.
27. 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.
28. 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.
29. Mridul Nandi. Fast and secure CBC-type MAC algorithms. In Orr Dunkelman, editor, *FSE*, volume 5665 of *Lecture Notes in Computer Science*, pages 375–393. Springer, 2009.
30. Mridul Nandi and Avradip Mandal. Improved Security Analysis of PMAC. Cryptology ePrint Archive, Report 2007/031, 2007. `http://eprint.iacr.org/`.
31. Erez Petrank and Charles Rackoff. CBC MAC for Real-Time Data Sources. *J. Cryptology*, 13(3):315–338, 2000.
32. Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. Full version of the Asiacrypt 2004 paper available from the author's home page, `http://www.cs.ucdavis.edu/~rogaway/papers/index.html`.
33. Phillip Rogaway. OCB Mode. `http://www.cs.ucdavis.edu/~rogaway/ocb/`.
34. Phillip Rogaway. Authenticated-encryption with associated-data. In Vijayalakshmi Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 98–107. ACM, 2002.
35. 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.
36. Phillip Rogaway, Mihir Bellare, and John Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, 2003.
37. Palash Sarkar. A general mixing strategy for the ECB-Mix-ECB mode of operation. *Inf. Process. Lett.*, 109(2):121–123, 2008.

38. Palash Sarkar. A simple and generic construction of authenticated encryption with associated data. Cryptology ePrint Archive, Report 2009/215, 2009. `http://eprint.iacr.org/`.
39. 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.
40. Serge Vaudenay. Decorrelation: A theory for block cipher security. *J. Cryptology*, 16(4):249–286, 2003.
41. Kan Yasuda. A one-pass mode of operation for deterministic message authentication- security beyond the birthday barrier. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 316–333. Springer, 2008.