Authenticated Encryption Mode IAPM using SHA-3's Public Random Permutation

Charanjit Jutla
IBM T. J. Watson Research Center
New York 10598

Abstract. We study instantiating the random permutation of the block-cipher mode of operation IAPM (Integrity-Aware Parallelizable Mode) with the public random permutation of Keccak, on which the draft standard SHA-3 is built. IAPM and the related mode OCB are single-pass highly parallelizable authenticated-encryption modes, and while they were originally proven secure in the private random permutation model, Kurosawa has shown that they are also secure in the public random permutation model assuming the whitening keys are uniformly chosen with double the usual entropy. In this paper, we show a general composability result that shows that the whitening key can be obtained from the usual entropy source by a key-derivation function which is itself built on Keccak. We stress that this does not follow directly from the usual indifferentiability of key-derivation function constructions from Random Oracles. We also show that a simple and general construction, again employing Keccak, can also be used to make the IAPM scheme key-dependent-message secure. Finally, implementations on modern AMD-64 architecture supporting 128-bit SIMD instructions, and not supporting the native AES instructions, show that IAPM with Keccak runs three times faster than IAPM with AES.

1 Introduction

Symmetric key encryption of bulk data is usually performed using either a stream cipher or a block cipher. A long message is divided into small fixed-size blocks and encryption is performed by either a stream-cipher mode or a block-cipher mode employing a cryptographic primitive that operates on blocks. The block primitives have traditionally been keyed-primitives, i.e. the block primitives also take a secret key as input. However, stream-cipher modes are sometimes designed to work with key-less block primitives as the state itself can maintain or carry some secret information. Examples include random-oracle domain extensions and authenticated-encryption in the streaming mode [11] using the sponge construction [5], proven secure under the strong notion of indifferentiability [20].

Note that the only underlying assumption in the sponge construction is that the fixed-length (input and output) permutation is indeed as good as picking such a permutation randomly from all such permutations with the same domain and range. The random permutation is publicly available, yet it is deemed random enough in the sense that without actually computing the permutation P

on x (such that x was not the output of an earlier computation of $P^{-1}(y)$ for some y), its value $P(x)$ is random and un-predictable. Indeed, this is the model under which most cryptographic hash functions operate including SHA-3 [24] (a draft standardization of Keccak [4]). We will refer to this as the *public random-permutation (RP) model*. This should be contrasted with the *private random-permutation (RP) model*, where the random-permutation is not available to the public and it can only be accessed via an oracle, such as an encryption/decryption algorithm which is built using this private random-permutation. Moreover, taking AES[1] as an example, the model contends that the AES permutation keyed with a secret key becomes a private random-permutation. However, note that it requires that two (or more) such instantiations with randomly and independently chosen keys lead to completely independent private random-permutations, which is a strong requirement on the block primitive.

The challenge of designing such strong block-cipher primitives[1], and at the same time advances in designing good random permutations enjoying provable bounds on differential trails [1, 4], has led to many proposals of encryption schemes in the public random-permutation model. However, this has still been the case mostly in the stream-cipher mode. As mentioned above, the Keccak team has proved that one can build authenticated-encryption stream-cipher modes using the very same public Keccak permutation [6] on which SHA-3 (as a random oracle) is built. The question naturally arises if one can build authenticated-encryption block-cipher modes of operation using the Keccak permutation, i.e. in the public RP model.

In 2010, Kurosawa [19] showed that a modified version of the Integrity-aware-Parallelizable-Mode (IAPM) [15] authenticated encryption scheme is secure in the public RP model. Jutla in [15] had only shown that the IAPM scheme is secure in the private random-permutation model (e.g. instantiating it with keyed-AES). The result of Kurosawa shows that one can instantiate it (or at least the slightly modified version) by a public random-permutation, e.g. the key-less Keccak permutation. He also showed that the same applies to modified versions of OCB [23] which is a variant of IAPM that can also handle messages that are not of length exact multiples of block size. The main attraction of these schemes is that they provide single-pass authenticated-encryption, and in addition are fully-parallelizable. Essentially, both these properties were obtained in the private RP model by requiring two independent keys $k1$ and $k2$, the key $k1$ being say, the AES key, and $k2$ being a whitening key. The whitening key $k2$ is used to whiten the i-th block of input before encryption by AES under key $k1$, and also to whiten the output of the AES encryption in the same way. We will refer to this as pre- and post-whitening with $k2$. The whitening refers to obtaining n-bits of new randomness from $k2$ and block index i, and xor-ing it to the input block. The main idea here is that this randomness need only

[1] We remark that AES, which builds such a keyed-primitive, has never been shown to exhibit any weakness in this primitive. Nevertheless, the keyed-primitive of AES is a strong property or assumption.

be pair-wise independent, which makes this a rather simple operation, e.g. a linear-feedback-shift-register operation.

The result of Kurosawa shows that one can get rid of the permutation key, i.e. $k1$ by setting it to a randomly chosen public constant, and the scheme is still secure for authenticated encryption (just by the pre- and post- whitening due to $k2$ using a pair-wise independent random function). This is then reminiscent of the Even-Mansour construction [13], except that it uses a pair-wise independent function of the key $k2$. Further, its security bound has terms similar to the Even-Mansour bound, namely $z * q * (2^{-n} + 2^{-|k2|})$, where z is the number of encryption/decryption queries, q is the number of evaluations of the public permutation, and n is the block size of the primitive. Thus, as shown by Daemen [12], one must have large n, because of the "quadratic" nature of the bound. Thus, a 128-bit AES permutation (with a fixed key) is out of the question. However, this quadratic nature of the bound also applies to the sponge construction mentioned above, and hence Keccak actually uses a permutation on $n = 1600$ bits, in which case at least this concern goes away. We will refer to this version of IAPM that uses the key-less Keccak permutation as IAPM-Keccak.

However, once we are in the public random-permutation model, there are other issues which need to be addressed, which are usually swept aside in the (private) random-permutation model by making various independence assumptions (most likely valid, but still not entirely satisfying). In the public random-permutation model, such independence assumption are definitely not valid a priori, and one must prove that composition of various components of an end-to-end encryption paradigm, e.g. a secure channel, are secure, especially if they are all using the same public random-permutation.

In particular, while one may make the benign assumption that the whitening key $k2$ is chosen uniformly at random from all 256-bit strings (this is the minimum width required for $k2$ because of the above quadratic bound so as to match security obtained in the private RP model), it most likely was obtained from a wider, less-uniform random source and with lesser min-entropy (say, 128-bits) using a key-derivation function. Most likely, this key-derivation function itself is built using the same public random-permutation (e.g. Keccak of SHA-3).

Even if this key-derivation function is proven to be a random oracle in the indifferentiability sense, it does not prove that it can be composed "as is" with IAPM that is using the same key-less permutation Keccak. In fact, while [20] prove a composition theorem that says that a cryptosystem $\mathcal{C}$ can use an ideal primitive $\mathcal{I}$, instead of an algorithm ALG built using another *public* ideal primitive $\mathcal{F}$, and still be equally secure, this composition theorem *does not* hold if $\mathcal{C}$ itself is using $\mathcal{F}$ (in our case $\mathcal{F}$ is the Keccak permutation). We defer detailed discussion to Section 5.

However, in this work we prove that in some special situations of cryptosystems themselves accessing the public ideal primitive $\mathcal{F}$ a composition result still holds. This result should be of general interest, beyond application to using IAPM in the random-permutation model. In particular, we show that a key-derivation function that uses the Keccak permutation and which is shown

indifferentiable from a random oracle can indeed be securely used to generate the 256-bit uniformly random whitening key of IAPM-Keccak. The final security bound we obtain is of the form $q * 2^{\kappa} + z * q * (2^{-n} + 2^{-256})$, where κ is the min-entropy of the key-source. This matches the key-source security bound in the private RP model.

We also need to study security of secrecy under key-dependent message encryption (KDM-security) [7] as in the public RP model this could have ramifications usually ignored in the private RP model. Further, apart from security issues like accidental encryption of the key itself, KDM security can have other applications [7]. In the random oracle model, [7] also show an encryption scheme that is KDM-secure. However, constructions of arbitrary output length random oracles from small fixed length random oracles or random permutations tend to be sequential or at best tree-like, and do not offer fully parallelization of IAPM. Further, while IAPM operates at full rate, i.e. rate of encryption of 1600 bits per invocation of Keccak permutation, the random oracle constructions have a lesser ratio than the bit-size of the permutation. Finally, IAPM provides authentication almost for free.

Fortunately, we show that a construction similar to [7] can be used to obtain KDM-security for IAPM. The main idea is to apply, for each message, a random oracle H on $(k\|IV)$ but only to obtain 256-bits of a fresh 256-bit whitening key $k2$. Then, this key $k2$ can be used to do the IAPM authenticated-encryption in the public RP model. It is a non-trivial task to prove that the same public random-permutation can be used to build the random oracle H also. Our result is also general and applies to any cryptosystem that is chosen plaintext attack (CPA) secure in the public RP model. In particular, it also applies to IAPM in the private random-permutation model (i.e. using keyed-AES). We also show, using our earlier composition theorem, that the key k need not be the wider source from which the key $k2$ is obtained, but an already extracted key k from the wider source k' using a random oracle built from the same public RP, as long as the source k' is erased after extraction of k.

Finally, we prove that general IAPM like constructions, such as OCB and others which are based on pre- and post- whitening by pair-wise independent random numbers, are as secure in the public random-permutation model as in the private random-permutation model.

We also implement the KDM-secure IAPM scheme using the Keccak-1600 permutation and show that on modern Intel/AMD architectures supporting 128-bit SIMD operations (and *not* supporting native AES instructions) it runs at speeds 3 times faster than a similar IAPM scheme using keyed-AES.

2 Preliminaries

Throughout this paper, an algorithm will be called an N-oracle algorithm if it has access to N number of oracles. If it has only one oracle, we will just refer to it as an oracle algorithm.

Definition 1. (ϵ**-XOR-Universal Hash Function**) [18] For any finite set H, an H-keyed (m, n)-hash function $\mathcal{H}$ has signature $\mathcal{H} : H \times \{0,1\}^m \to \{0,1\}^n$. Such a hash function is called an ϵ-XOR-Universal hash function, if for every m-bit value M, and every n-bit value c, $\Pr_h[\mathcal{H}(h, M) = c] \leq \epsilon$, and further if for every pair of distinct m-bit values $M1$ and $M2$, and every n-bit value c, $\Pr_h[\mathcal{H}(h, M1) \oplus \mathcal{H}(h, M2) = c] \leq \epsilon$, where the probabilities are over choosing h uniformly from H.

Definition. For a random variable X defined on $\{0,1\}^n$, its **min-entropy** $H_\infty(X)$ is the minimum over all n-bit strings x of $\log(1/\Pr_X[X = x])$.

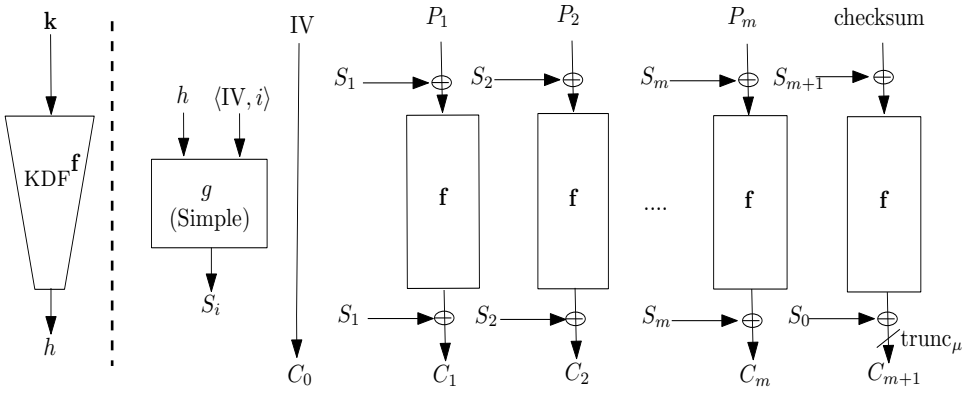


Fig. 1. IAPM in Public Random Permutation Model

3 Authenticated Encryption

We give definitions of authenticated encryption schemes in a public random permutation model. Let Coins be the set of infinite binary strings. Let $\mathcal{K} \subseteq \{0,1\}^*$ be the key space, and $\mathcal{D}$ be a distribution on the key space.

Definition A (2-oracle, probabilistic, symmetric, stateless) *authenticated-encryption* scheme, with block size n, key space $\mathcal{K}$, and distribution $\mathcal{D}$, consists of the following:

- **initialization:** All parties exchange information over private lines to establish a private key $k \in \mathcal{K}$. All parties store k in their respective private memories.
- **message sending with integrity:** Let E and D be efficient 2-oracle algorithms, with E taking as input a key k (in $\mathcal{K}$), COINS (in Coins), and a plaintext binary string and outputting a binary string, and D taking as input a key k and a ciphertext binary string and outputting either $\perp$ or a binary string. The two oracles take n-bits as input and produce n-bits as output.

In addition E and D have the property that if oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ implement inverse functions of each other, then for all $k \in \mathcal{K}$, for all COINS and P,

$$D^{\mathcal{O}_1,\mathcal{O}_2}(k, (E^{\mathcal{O}_1,\mathcal{O}_2}(k, \text{COINS}, P)) = P$$

We will usually drop the random argument to E as well, and just think of E as a probabilistic algorithm. The security of such a scheme is given by the following two definitions, the first defining confidentiality under chosen plaintext attacks, and the second defining message integrity. In the security definitions, we will count the length of plaintext inputs in terms of n-bit blocks. Thus, a plaintext input of length m bits will be considered to have length $\lceil m/n \rceil$ blocks.

Definition (*Chosen-Plaintext Attack Security[2]*)

For any $n > 0$, consider a 3-oracle probabilistic adversary A. Consider an authenticated-encryption scheme with key-space $\mathcal{K}$, key distribution $\mathcal{D}$ and 2-oracle algorithms E and D. For any n-bit permutation π, let $\mathsf{Real}^{\pi}_{\mathbf{k}}$ be the oracle that on input P returns $E^{\pi,\pi^{-1}}(\mathbf{k}, P)$, and $\mathsf{Ideal}^{\pi}_{\mathbf{k}}$ be the oracle that on input P returns $E^{\pi,\pi^{-1}}(\mathbf{k}, 0^{|P|})$. The IND-CPA advantage Adv_A of the adversary A in the *public random permutation model* is given by

$$|\Pr[\mathbf{k}\leftarrow\mathcal{D}; A^{\pi,\pi^{-1},\mathsf{Real}^{\pi}_{\mathbf{k}}} = 1] - \Pr[\mathbf{k}\leftarrow\mathcal{D}; A^{\pi,\pi^{-1},\mathsf{Ideal}^{\pi}_{\mathbf{k}}} = 1]|$$

where the probabilities are over choice of π as a random permutation on n-bits, and choice of k according to $\mathcal{D}$, other randomness used by E, and the probabilistic choices of A.

An authenticated-encryption scheme with block size n is said to be (t, $q1$, $q2$, m, ϵ)-secure against chosen plaintext attack in the *public random permutation model* if for any adversary A as above which runs in time at most t and asks at most $q1$ queries to π and π^{-1}, and at most $q2$ queries to the third oracle (these totaling at most m blocks), its advantage Adv_A is at most ϵ.

Definition (*Message Integrity):* Consider an adaptive 3-oracle (probabilistic) adversary A running in two stages. Adversary A has access to oracles $\mathcal{O}_1$, $\mathcal{O}_2$ and an encryption oracle $E^{\mathcal{O}_1,\mathcal{O}_2}(k, \cdot)$. In the first stage (*find*) A asks r queries of the encryption oracle. Let the oracle replies be $C^1, ..., C^r$. Subsequently in the second stage, A produces a cipher-text C', different from each C^i, $i \in [1..r]$. The adversary's success probability is given by

$$\text{Succ}_A \stackrel{\text{def}}{=} \Pr[D^{\pi,\pi^{-1}}(k, C') \neq \perp]$$

where the probability is over choice of $\mathcal{O}_1$ as a random permutation on n-bits (and $\mathcal{O}_2$ as its inverse), and choice of k according to $\mathcal{D}$, other randomness used by E, and the probabilistic choices of A.

An authenticated-encryption scheme with block size n is $(t, q1, q2, m, \epsilon)$-secure for message integrity in the *public random permutation model* if for any 3-oracle adversary A running in time at most t and making at most $q1$ queries to $\mathcal{O}_1$ and $\mathcal{O}_2$ and at most $q2$ queries to the encryption oracle (these totaling m blocks), its success probability is at most ϵ.

4 IAPM in Random Permutation Model

We will prove our results for more general (abstract) IAPM-like schemes, but to serve as a background we briefly review the definition of IAPM from [15, 16]. In the following, the operator "+" will stand for integer addition, and "$\oplus$" for n-bit exclusive-or. Since with wide permutations on n bits, the "MAC" tag produced by the permutation may need to be truncated, the authentication check in decryption is defined slightly differently (as in OCB [23] and [19]). In the following, when using n-bit permutations, we will refer to n-bit strings as a *block*.

Definition 2. Given a permutation f from n bits to n bits, an H-keyed $(2n, n)$-hash-function g, where H is the set of all ν-bit strings ($\nu \leq n$), the (deterministic) function E-IAPM$_{f,g}$: $H \times \{0,1\}^n \times (\{0,1\}^n)^* \rightarrow (\{0,1\}^n)^+$ is defined as follows:

- Let the input to E-IAPM$_{f,g}$ be $h \in H$, an n-bit (block) IV, and an m block string P $(= P_1, P_2, ..., P_m)$.
- Define $C_0 = IV$, and checksum $= 0 \oplus \bigoplus_{j=1}^{m} P_j$.
- Define for $j = 1$ to m:
  $C_j = g(h, \langle IV, j \rangle) \oplus f(P_j \oplus g(h, \langle IV, j \rangle))$.
- $C_{m+1} = g(h, \langle IV, 0 \rangle) \oplus f(\text{checksum} \oplus g(h, \langle IV, m+1 \rangle))$.
- The output of the function E-IAPM$_{f,g}$ is the $m+2$ block string $C_0, C_1, ..., C_{m+1}$. The last block can be truncated to the required "MAC" tag-length, say μ bits.

Definition 3. With the same parameters as above, the function D-IAPM$_{f,g}$: $H \times (\{0,1\}^n)^+ \rightarrow (\{0,1\}^n)^* \cup \{\perp\}$ is defined as follows:

- Let the input to D-IAPM$_{f,g}$ be an $h \in H$, an $((m+1)n + \mu)$-bit string C, which is divided into $(m+1)$ blocks IV, $C_1, ..., C_m$ and a tag T of μ bits.
- Define for $j = 1$ to m:
  $P_j = g(h, \langle IV, j \rangle) \oplus f^{-1}(C_j \oplus g(h, \langle IV, j \rangle))$.
- $T^* = g(h, \langle IV, 0 \rangle) \oplus f(\bigoplus_{j=1}^{m} P_j \oplus g(h, \langle IV, m+1 \rangle))$.
- if $(\text{trunc}_\mu(T^*) \neq T)$ return $\perp$, otherwise the output of D-IAPM$_{f,g}$ is the m block string $P_1, ..., P_m$.

See Fig. 1 (right of the dashed vertical line) for a schematic diagram. The left of the dashed line depicts key derivation using the same permutation, which is discussed in the next sub-section.

4.1 Public Random Permutation Model

If g is an efficiently computable function, the above two functions E-IAPM and D-IAPM can be computed efficiently given oracle access to f and f^{-1}. It is important to make this characterization as we intend to instantiate f and f^{-1} by public permutations. Further, the definition of an (authenticated) encryption scheme requires specifying the distribution from which the keys are sampled. While we

may assume a benign setting where the ν-bit key h above is chosen uniformly from H, it is most likely that this key is obtained using a key-derivation function (KDF) which in turn also used the same public permutation f. Thus, we will define a composite scheme which takes an arbitrarily long bit-string k as (key) input, uses a general-purpose KDF (with oracle access to f and f^{-1}) to obtain h from k, and then uses E-IAPM and D-IAPM as per Definitions 2, 3 with parameter g and with oracle access to f and f^{-1}.

Definition 4. (**IAPM in public random permutation model**)[Fig. 1] Let f be an n-bit permutation. Let g be an (efficiently computable) H-keyed $(2n, n)$-hash function, where H is the set of all ν-bit strings ($\nu \leq n$). Let KDF be an efficient (key-derivation) 2-oracle algorithm that takes arbitrary bit strings as input and produces ν-bit strings as output. The authenticated-encryption scheme IAPM(KDF, g, ν, μ, κ) with block size n, and oracles f and f^{-1} is given by the following key space, distribution, and 2-oracle encryption and decryption algorithms:

- The set $\mathcal{K}$ of keys is arbitrary bit strings. The distribution $\mathcal{D}$ on $\mathcal{K}$ is any distribution on $\mathcal{K}$ with min-entropy κ.
- Let $h = \text{KDF}^{f,f^{-1}}(k)$.
- The encryption under key k is given by $\text{E-IAPM}_g^{f,f^{-1}}(h, \cdot, \cdot)$, and the decryption by $\text{D-IAPM}_g^{f,f^{-1}}(h, \cdot)$.

It is easy to see that the decryption algorithm correctly inverts the encryption algorithm.

In Section 5.1 we prove a general composition result for application of key-derivation functions, and using that it will follow that all security properties related to the above composite scheme can be reduced to related security properties of the following IAPM scheme with uniformly chosen keys.

Definition 5. (**IAPM with uniform keys in public RP model**) Authenticated-Encryption scheme IAPM-uniform(g, ν, μ) with block size n, and oracle f and f^{-1} is given by a key space $\mathcal{K}$ that is the set of ν-bit strings, and a distribution $\mathcal{D}$ on keys that is the uniform distribution on $\mathcal{K}$. Moreover, the encryption and decryption algorithms under key k are given by $\text{E-IAPM}_g^{f,f^{-1}}(k, \cdot, \cdot)$, and $\text{D-IAPM}_g^{f,f^{-1}}(k, \cdot)$ resp.

Definition 6. (**Zero-IV IAPM**) An IAPM scheme is called a zero-IV scheme if IV is always set to zero. Thus, $C_0 = 0$ for all ciphertexts, and g function is computed with IV set to zero. As a consequence, the encryption function does not need the IV input.

5 Indifferentiability

In this section we briefly discuss the notion of indifferentiability introduced by Maurer et al [20] based on ideas of universal composability (UC) [9] and the model described in [21]. We refer the reader to [20, 11] for details.

A cryptosystem $\mathcal{C}$ is modeled as an interactive algorithm (or Turing Machine), and it is run by an *environment* $\mathcal{E}$. The cryptosystem $\mathcal{C}$ has a private interface $\mathcal{C}^{\mathsf{priv}}$ to the environment $\mathcal{E}$ and a public interface $\mathcal{C}^{\mathsf{pub}}$ to the adversary. The environment also controls the adversary. An *ideal primitive* is a cryptosystem whose interface just serves queries with answers. In this work, we focus on the notion of a *public ideal primitive* that has only a single interface which serves as both public and private interfaces. An important public ideal primitive is a *random oracle* (RO) which provides a random output to each query with the constraint that identical queries are replied with the same answer. We will refer to a random oracle that outputs exactly m-bits as an m-bit RO. Note that the input to an m-bit RO can be an arbitrarily long string.

Definition 7. An oracle algorithm ALG with its oracle instantiated by an ideal primitive $\mathcal{F}$ is said to be $(t_D, t_S, q1, q2, L, \epsilon)$-indifferentiable from a public ideal primitive $\mathcal{I}$ if there exists an oracle algorithm (called simulator) S that runs in time t_S and makes at most L oracle calls, and such that for any (2-oracle) distinguisher D the following holds:

$$|\Pr[D^{\mathrm{ALG}^{\mathcal{F}},\mathcal{F}} = 1] \; - \; \Pr[D^{\mathcal{I},S^{\mathcal{I}}} = 1]| < \epsilon$$

where D runs in time t_D and makes at most $q1$ ($q2$) calls to the first oracle (second oracle resp.). When the above property holds regardless of the run-time of D, we will say that $\mathrm{ALG}^{\mathcal{F}}$ is $(\infty, t_S, q1, q2, L, \epsilon)$-indifferentiable from $\mathcal{I}$.

Readers more familiar with the UC framework will note that the above is equivalent to saying that the public ideal functionality $\mathcal{I}$ is UC-realizable by ALG in the $\mathcal{F}$-hybrid model.

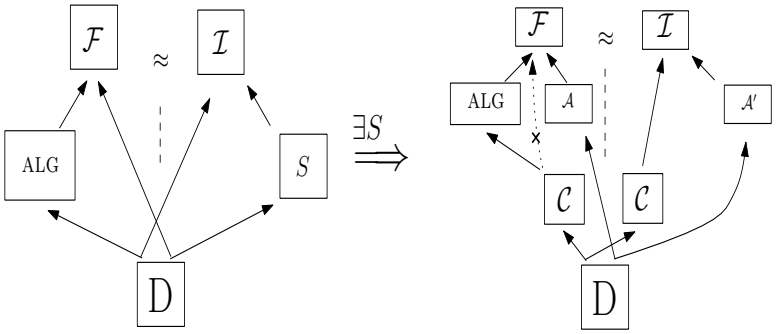
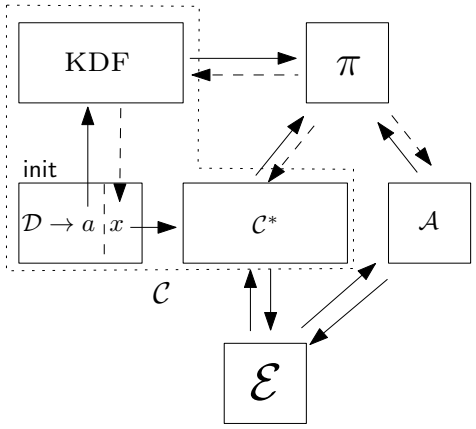


Fig. 2. Indifferentiability and Composition

When composing cryptosystems, it is important to note that if a cryptosystem $\mathcal{C}$ uses a cryptosystem $\mathcal{P}$ then the public interface of $\mathcal{C}$ includes the public interface of $\mathcal{P}$. One of the main results of [20] proves a composition theorem (see Fig. 2) which informally states that if an oracle algorithm ALG with oracle access

to a public ideal primitive $\mathcal{F}$ is indifferentiable from a public ideal primitive $\mathcal{I}$, then a cryptosystem $\mathcal{C}$ using $\text{ALG}^{\mathcal{F}}$ (with adversary having access to $\mathcal{F}$ by the above convention) is as secure as the cryptosystem $\mathcal{C}$ using $\mathcal{I}$ (with adversary having access to $\mathcal{I}$). However, if $\mathcal{C}$ itself accesses the public ideal primitive $\mathcal{F}$, then this composition theorem may not hold in general. In fact, $\mathcal{C}$ needs its oracle instantiated by either $\mathcal{F}$ or some other public ideal primitive in the $\mathcal{I}$-world as well. In this situation, for the composition theorem to hold in general it is well known that in the definition of indifferentiability the distinguisher may need access to the same primitive $\mathcal{F}$ in both worlds [10]. This, of course, would preclude programming of $\mathcal{F}$ using the simulator S.

However, we show that in some special situations of cryptosystems themselves accessing the public ideal primitive a composition result still holds. For the next definition, we will focus on cryptosystems that are themselves ideal primitives and further they use another public ideal primitive, say $\mathcal{F}$, as an oracle. Thus, the public interface of the former primitive is also $\mathcal{F}$. We now specialize the definition of "as secure as" [20] to cater to such cryptosystems.



Dashed arrows indicate oracle responses.

Fig. 3. Cryptosystem initialized using KDF

Definition 8. For public ideal primitives $\mathcal{F}_1$ and $\mathcal{F}_2$, a cryptosystem $\mathcal{C}_1^{\mathcal{F}_1}$ is said to be $(q1, q2, N, 1-\epsilon)$ as secure as a cryptosystem $\mathcal{C}_2^{\mathcal{F}_2}$ if for all environments $\mathcal{E}$ the following holds: for all adversary $\mathcal{A}_1$ making at most a total of $q1$ oracle calls there is an adversary $\mathcal{A}_2$ making at most a total of $q2$ oracle calls such that

$$|\Pr[\mathcal{E}(\mathcal{C}_1^{\mathcal{F}_1}, \mathcal{A}_1^{\mathcal{F}_1}) = 1] - \Pr[\mathcal{E}(\mathcal{C}_2^{\mathcal{F}_2}, \mathcal{A}_2^{\mathcal{F}_2}) = 1]| \; < \; \epsilon,$$

where both probabilities are conditioned on the total number of calls to $\mathcal{F}_1$ ($\mathcal{F}_2$ resp.) by $\mathcal{C}_1$ and $\mathcal{A}_1$ combined (by $\mathcal{C}_2$ and $\mathcal{A}_2$ combined resp.) being less than N.

5.1 KDF Composition

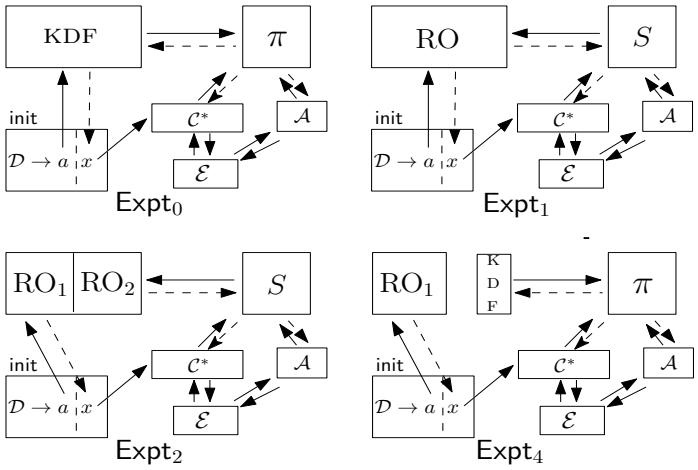


Fig. 4. Various Experiments in Theorem 1

Definition. We will say that a cryptosystem $\mathcal{C}$ has an initialization step init, if $\mathcal{C}$ can be split into two parts init and $\mathcal{C}^*$. Further, over all calls from $\mathcal{E}$ to $\mathcal{C}$, only the first call leads to execution of init and which results in a private state σ. The private state σ is used as an additional input by $\mathcal{C}^*$ in all calls from $\mathcal{E}$ to $\mathcal{C}$.

Theorem 1. *Let* KDF *be an oracle algorithm such that with its oracle instantiated with a public ideal primitive π, it is $(\infty, t_S, q_1, q_2, L, \epsilon)$-indifferentiable from an m-bit RO. Let $\mathcal{C}_1$ be a 1-oracle cryptosystem that has an initialization step that generates a private state by sampling m uniformly random bits. Let $\mathcal{D}$ be any distribution on finite length binary strings with min-entropy ν. Let $\mathcal{C}$ be a cryptosystem which is identical to $\mathcal{C}_1$ except that the initialization step is different and consists of running* KDF *on an input a sampled from $\mathcal{D}$, with the oracle calls of* KDF *redirected to the oracle of $\mathcal{C}_1$. The private state of the initialization step is the output of* KDF*. Then, for all q_3, and for all $(q_3 \leq) N < q_2$, cryptosystem $\mathcal{C}^\pi$ is $(q_3, q_3, N, 1 - L * N * 2^{-\nu} - 2 * \epsilon)$ as secure as cryptosystem $\mathcal{C}_1^\pi$.*

Remark 1. The cryptosystem $\mathcal{C}$ is depicted in Fig. 3 (and also in Expt_0 of Fig. 4). It is important to note that π is a public ideal primitive, and when proving security the adversary is allowed access to π. The cryptosystem $\mathcal{C}_1$ can be seen represented in Expt_4 of Fig. 4.

Remark 2. In most known realizations of RO such as the sponge construction [5], the simulator S makes at most $L = 1$ oracle calls.

Proof: Let $\mathcal{E}$ be any environment. Note that the public interfaces of $\mathcal{C}$ and $\mathcal{C}_1$ include the interface of public ideal primitive π. Let $\mathcal{C}_1$ consist of an initialization

phase of sampling a uniformly random m-bit string r and a second 1-oracle phase $\mathcal{C}^*$ running with additional input r. Let Ψ be a 2-oracle cryptosystem with oracles $\mathcal{O}_1$ and $\mathcal{O}_2$, with an initialization phase that samples a from $\mathcal{D}$, queries $\mathcal{O}_1$ with a to get x and runs the 1-oracle second phase $\mathcal{C}^*$ with additional input x and oracle $\mathcal{O}_2$. Note that Ψ makes at most one call to the first oracle $\mathcal{O}_1$. Moreover, if the two oracles of Ψ are instantiated by $\mathcal{O}_1 = \textsc{kdf}^\pi$ and $\mathcal{O}_2 = \pi$, then we get the cryptosystem $\mathcal{C}^\pi$ (see Fig. 3).

For any adversary $\mathcal{A}$, consider a composite 2-oracle algorithm D that is a composition of $\mathcal{E}$, the 1-oracle adversary $\mathcal{A}$ and Ψ as defined above. The oracle calls of 2-oracle Ψ are directed to the two oracles of D respectively, and the oracle calls of the 1-oracle $\mathcal{A}$ are directed to the second oracle of D. The algorithm D also outputs a single bit which is same as the bit output by $\mathcal{E}$. Now consider two worlds: a *real world* where the first oracle is instantiated by $\textsc{kdf}^\pi$ and the second oracle by π, and an *ideal world* where the first oracle of D is instantiated by an m-bit RO and the the second oracle by S (which itself has oracle access to the same m-bit RO). Here S is the simulator as stipulated in the indifferentiability hypothesis of $\textsc{kdf}^\pi$ and m-bit RO. More formally, we will say that D is taking part in the real world experiment or the ideal world experiment. The real and the ideal world experiments will also be denoted by Expt_0 and Expt_1 respectively (see Fig. 4). We will denote probabilities in Expt_i by a subscript i. Let N be any number less than q_2. Note that the total number of calls to the second oracle of D is the sum of the total number of calls of Ψ to its second oracle and the total number of calls of $\mathcal{A}$ to its oracle. By the indifferentiability hypothesis, and conditioned on D making at most $N(< q_2)$ calls to the second oracle, the algorithm D cannot distinguish between the real world experiment and the ideal world experiment with probability more than ϵ. In other words, $|\Pr_0[D = 1] - \Pr_1[D = 1]| \leq \epsilon$.

Let BAD be the event that in Expt_1, the simulator S makes a call to its oracle (the m-bit RO) which is identical to the single call made to the first oracle by D. Recall, in Expt_1 the first oracle of D is same as the m-bit RO oracle of S. Now, the probability of D outputting 1 in Expt_1 is at most the sum of the following two values: (a) the probability of D outputting 1 *and* event BAD *not* happening, and (b) the probability of event BAD happening. Thus, $\Pr_1[D = 1 \wedge \neg\text{BAD}] \leq \Pr_1[D = 1] \leq \Pr_1[D = 1 \wedge \neg\text{BAD}] + \Pr_1[\text{BAD}]$.

Now, consider another experiment Expt_2 (see fig. 4) which differs from the ideal world experiment Expt_1 in that the common m-bit RO oracle of S and D is replaced by two independent m-bit random oracles RO_1 and RO_2 (RO_1 for the first oracle of D and RO_2 for the oracle of S; see Fig. 4).

From the definition of a random oracle, i.e. the fact that it outputs random and independent values on different inputs, it is not difficult to see that the first probability (a) remains same in Expt_2 as in Expt_1. More formally, this is proved by induction over a sequence of hybrid games, starting from Expt_1 and ending in Expt_2, where in each subsequent game one additional call of S to its oracle (going backward from last call to first) is made to the new independent m-bit random

oracle RO_2. Thus, $\Pr_2[D = 1 \wedge \neg\mathrm{BAD}] \leq \Pr_1[D = 1] \leq \Pr_2[D = 1 \wedge \neg\mathrm{BAD}] + \Pr_1[\mathrm{BAD}]$.

Now, consider experiment Expt_3 which is same as experiment Expt_2 except that the single call to the first oracle is replaced by just generating a uniform m-bit random value independently. This is just a syntactic change by definition of m-bit RO, and hence the probability (a) remains the same. Since the first oracle call does not access any m-bit RO, the m-bit RO oracle of S is the only RO that remains in Expt_3. Thus the above inequalities continue to hold with subscript 2 replaced by 3. It also follows that $\Pr_3[D = 1] - \Pr_3[\mathrm{BAD}] \leq \Pr_1[D = 1] \leq \Pr_3[D = 1] + \Pr_1[\mathrm{BAD}]$.

Next, consider Expt_4 which is same as Expt_3 except that the second oracle of D is instantiated by primitive π. Again, by the indifferentiability hypothesis of KDF^π and m-bit RO, the probability $\Pr_3[D = 1]$ differs from $\Pr_4[D = 1]$ by at most ϵ. Now, note that experiment Expt_4 is identical to $\mathcal{E}$ running $\mathcal{C}_1^\pi$ and adversary $\mathcal{A}^\pi$. Since D outputs the same bit that is output by $\mathcal{E}$ it follows that $|\Pr_4[\mathcal{E}() = 1] - Pr_0[\mathcal{E}() = 1]| \leq 2 * \epsilon + \max\{\Pr_1[\mathrm{BAD}], \Pr_2[\mathrm{BAD}]\}$.

Since in both Expt_1 and Expt_2, the value x is independent of a (by definition of random oracle), it follows that all oracle calls of simulator S in both Expt_1 and Expt_2 are independent of a. Moreover, for each invocation of S, S itself makes at most L oracle calls. Since $\mathcal{D}$ has min-entropy ν, it follows by union bound that both $\Pr_1[\mathrm{BAD}]$ and $\Pr_2[\mathrm{BAD}]$, conditioned on total number of calls to the second oracle being less than N, are upper bounded by $L * N * 2^{-\nu}$ and that completes the proof. ■

6 Key-Dependent Message Security

In this section we show that IAPM in public RP model (Def. 4) can be slightly modified by introducing a random nonce so that it even becomes key-dependent message (KDM) secure. KDM security was introduced and formalized in [7], extending the notion of circular security from [8]. Informally, KDM security means that an Adversary cannot distinguish between an encryption of some function ϕ of the key itself from encryption of a constant message. The function ϕ is also allowed to be picked by the adversary adaptively.

6.1 KDM Security Definition

In this work, we will follow the definition of KDM security from [7] in the random oracle model, and adapt it to the public RP model, but will focus on a single key instead of a set of keys. One interesting feature of this definition is that the Adversary can ask for encryptions of the key under any function ϕ of its choice, and even a function ϕ whose description is given by an oracle-algorithm with the oracle to be instantiated by the very same public random-permutation.

In the following, we will restrict the Adversary's choice of oracle-algorithms ϕ to *fixed-output-length* algorithms, i.e. for all oracles π, $|\phi^{\pi}(k)|$ is same for all k.

Definition (*Key-Dependent Message Security*) For any $n > 0$, consider a 3-oracle probabilistic adversary A. Consider an (authenticated) encryption scheme with key-space $\mathcal{K}$, key distribution $\mathcal{D}$ and 2-oracle-algorithms E and D. For any n-bit permutation π, Let $\mathsf{Real}^{\pi}_{\mathbf{k}}$ be the oracle that on input a description of a 2-oracle fixed-output-length algorithm ϕ returns $E^{\pi,\pi^{-1}}(\mathbf{k}, \phi^{\pi,\pi^{-1}}(\mathbf{k}))$, and $\mathsf{Ideal}^{\pi}_{\mathbf{k}}$ be the oracle that on input P returns $E^{\pi,\pi^{-1}}(\mathbf{k}, \text{ZERO})$, where ZERO is a bit-string of zeroes of length $|\phi^{\pi,\pi^{-1}}(\mathbf{k})|$. The IND-KDM advantage $\mathrm{Adv}^{\mathtt{kdm}}_{A}$ of the adversary A in the *public random-permutation model* is given by

$$|\Pr[\mathbf{k}\leftarrow\mathcal{D}; A^{\pi,\pi^{-1},\mathsf{Real}^{\pi}_{\mathbf{k}}} = 1] - \Pr[\mathbf{k}\leftarrow\mathcal{D}; A^{\pi,\pi^{-1},\mathsf{Ideal}^{\pi}_{\mathbf{k}}} = 1]|$$

where the probabilities are over choice of π as a random permutation on n-bits, and choice of k according to $\mathcal{D}$, other randomness used by E, and the probabilistic choices of A.

An (authenticated) encryption scheme with block size n is said to be (t, $q1$, $q2$, $t3$, $q3$, m, ϵ)-secure against key-dependent message attack in the *public random-permutation model* if for any adversary A as above that restricts its queries to description of 2-oracle-algorithms ϕ that run in time $t3$ and make at most $q3$ oracle calls, and which itself (i.e. A) runs in time at most t and asks at most $q1$ queries to π and π^{-1}, and at most $q2$ queries to the third oracle (these totaling at most m blocks), its advantage $\mathrm{Adv}^{\mathtt{kdm}}_{A}$ is at most ϵ.

6.2 General Construction

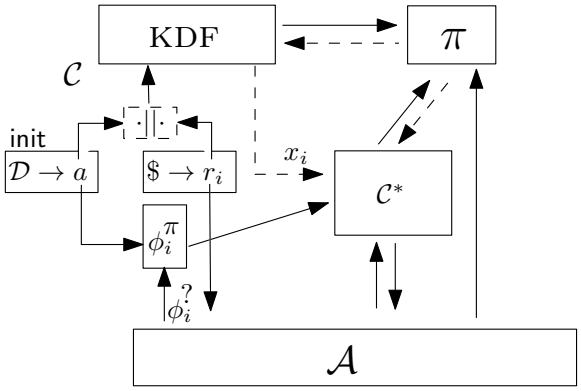


Fig. 5. KDM Secure General Construction in Public RP Model

Definition 9. Let $\mathcal{C}^*$ be a 2-oracle stateless authenticated encryption scheme with block size n, with key space $\mathcal{K}^*$ and distribution $\mathcal{D}^*$ on $\mathcal{K}^*$ given by uniform distribution on all ν-bit strings, and encryption and decryption algorithms E^* and D^*. Let KDF be an efficient (key-derivation) 2-oracle-algorithm that takes arbitrary bit strings as input and produces ν-bit strings as output. Then, define another 2-oracle stateless probabilistic authenticated encryption scheme $\mathcal{C}$ with block size n as follows (let $\mathcal{O}_1$ and $\mathcal{O}_2$ be its oracles):

- The set $\mathcal{K}$ of keys is arbitrary bit strings. The distribution $\mathcal{D}$ on $\mathcal{K}$ is any distribution on $\mathcal{K}$ with min-entropy κ.
- The probabilistic encryption algorithm under key a, takes input P, chooses ρ-bit r uniformly at random, obtains $x = \text{KDF}^{\mathcal{O}_1,\mathcal{O}_2}(a||r)$, and outputs $\langle r, E^{*\mathcal{O}_1,\mathcal{O}_2}(x, P)\rangle$.
- The decryption algorithm under key a, takes as input $\langle r, C\rangle$, obtains $x = \text{KDF}^{\mathcal{O}_1,\mathcal{O}_2}(a||r)$, and outputs $D^{*\mathcal{O}_1,\mathcal{O}_2}(x, C)$.

Theorem 2. *Let $\mathcal{C}^*$ as above be $(t, q1, (q2 =) 1, m, \epsilon_1)$-secure against chosen plaintext attacks in the public random-permutation model. Let β be such that, For each l (n-bit) block plaintext input, $\beta * l$ is the maximum number of calls that E^* makes to its oracles. Let* KDF *as above with its oracle instantiated with a public random-permutation on n bits be $(\infty, t_S, q3, q4, L, \epsilon_2)$-indifferentiable from a ν-bit RO. Then, the authenticated encryption scheme $\mathcal{C}$ as defined above is $(t', q1', q2', t'_3, q3', m, \delta)$ KDM-secure in the public random-permutation model, for*

- $t' + t'_3 + (q1' + q3') * t_S < t$, *and*
- $\beta * m + q1' + q3' < q4$, *and where*
- $\delta = 4 * \epsilon_2 + 2 * \epsilon_1 + (\beta * m + q1' + q3') * L * (q2' * 2^{-\rho} + 2^{-\kappa})$.

Remark 3. For authenticated encryption schemes such as IAPM, β is at most 2. Moreover, for most ν-bit RO constructions such as the sponge construction L is at most 1. Also, note that in the theorem statement $\mathcal{C}^*$ is required to be only *single-use* secure, i.e. $q2 = 1$ or only one encryption query is allowed. Informally, this suffices as the encryption key x for $\mathcal{C}^*$ is obtained as $x = \text{KDF}(a||r)$, for a fresh r for each message.

Proof: We will focus on the proof for a single encryption query by the Adversary A. Proof for multiple queries follows by induction by considering hybrid experiments. See Fig. 5 for a depiction of this setting along with the construction of $\mathcal{C}$. We will denote both the public random permutation and its inverse as a single public ideal primitive π which offers both interfaces. All random variables will be denoted in boldface.

The real world experiment where encryption of $\phi(\mathbf{a})$ is returned will be called Expt_0. We will define a sequence of experiments, with the last being the one in which a constant string is encrypted. We will show that in each subsequent experiment, the probability of A outputting 1 is only negligibly different from the previous experiment.

In Expt_1, we replace KDF and π by ν-bit RO and the simulator S as stipulated in the indifferentiability of KDF from ν-bit RO. By the indifferentiability claim the difference in the probability of A outputting 1 is at most ϵ_2. We will use subscript i to denote probabilities in experiment Expt_i. Thus, $|\Pr_1[A = 1] - \Pr_0[A = 1]| < \epsilon_2$.

Let BAD be the event that in Expt_1, the simulator S makes a call to its oracle (the ν-bit RO) which is identical to the (single) call made to the ν-bit RO by $\mathcal{C}$, i.e. $(\mathbf{a}||\mathbf{r})$, where $\mathbf{r}$ is a ρ-bit uniform and independent random value. Now, $\Pr_1[A = 1]$ is at most the sum of $\Pr_1[A = 1 \wedge \neg\text{BAD}]$ and $\Pr_1[BAD]$.

Now, consider experiment Expt_2 where we split the RO into two independent random oracles RO_1 and RO_2, where the call $(\mathbf{a}||\mathbf{r})$ is served by RO_1 and all calls by S are served by RO_2. This is similar to the situation depicted in Expt_2 in Fig. 4. It is clear that $\Pr_2[A = 1 \wedge \neg\text{BAD}]$ remains same as in Expt_1.

We, also consider Expt_3 where the call $(\mathbf{a}||\mathbf{r})$ to RO_1 is replaced by just using a random and independent ν bit value $\mathbf{x}$. By definition of RO, this is same as Expt_2.

Next, we switch to Expt_4 where we go back to KDF and public random permutation π, except that there is no call to the KDF now (similar to as shown in Expt_4 in Fig. 3). Now, note that the encryption of $\phi(\mathbf{a})$ is being performed under a key $\mathbf{x}$, which is a ν-bit uniformly random value independent of all other variables including $\mathbf{a}$ and $\mathbf{r}$. Thus, by CPA security of $\mathcal{C}^*$, we can consider Expt_5 where we replace the encryption of $\phi(\mathbf{a})$ by a constant string of the same length, and the Adversary will not be able to distinguish with probability more than ϵ_1. Thus, similar to proof of theorem 1, $|\Pr_5[A() = 1] - Pr_0[A() = 1]| \leq 2 * \epsilon_2 + \epsilon_1$ + $\max\{\Pr_1[\text{BAD}], \Pr_2[\text{BAD}]\}$.

We now bound both $\Pr_1[\text{BAD}]$ and $\Pr_2[\text{BAD}]\}$. We first focus on the former. First note that $\mathbf{r}$ is only revealed to the Adversary A at the end of encryption by $\mathcal{C}^*$, while $\mathcal{C}^*$ runs independent of $\mathbf{r}$. Thus, all calls by $\mathcal{C}^*$ to S are independent of $\mathbf{r}$, and similarly all calls by A to to S before $\mathcal{C}$ outputs $\mathbf{r}$ are independent of $\mathbf{r}$. Thus the probability of any of these calls leading to event BAD is at most $L * 2^{-\rho}$ (recall, L is the maximum number of calls by S to RO in any invocation of S). Let there be a total of q' such calls to S.

So, we now focus on calls by A to S after $\mathbf{r}$ is output by $\mathcal{C}$ to A. Let there be q'' such calls. We will also split BAD as a disjunction of BAD$'$ and BAD$''$, where BAD$'$ is BAD restricted to the q' calls above, and BAD$''$ is conjunction of BAD$'$ *not* happening and BAD restricted to the q'' calls of the latter kind. Consider the i-th such call by A to S. We can write BAD$''$ as a disjunction of $(\texttt{COL}_i \wedge \neg\text{BAD}' \wedge \forall j < i : \neg\, \texttt{COL}_i)$ with i ranging from 1 to q'', where $\texttt{COL}_i$ stands for collision in oracle calls of S with $(\mathbf{a}||\mathbf{r})$ in A's i-th invocation of S. Further, since these q'' disjuncts are mutually exclusive, the probability of BAD$''$ is exactly the sum of the probability of each disjunct. We will refer to each disjunct as BAD''_i. We now show that $Pr_1[\text{BAD}''_i] = \Pr_2[\text{BAD}''_i[$. Since the view of the adversary A at the point it makes the i-th call is completely determined by earlier calls of A to S and all calls of $\mathcal{C}^*$, and given that the Expt_1 and Expt_2 are identically

distributed till that point conditioned on $\text{BAD}' \wedge \forall j < i : \neg\ \texttt{COL}_i$, the claim follows.

Again, since the events BAD''_i are mutually exclusive, we get $\Pr_1[\text{BAD}''] = \Pr_2[\text{BAD}'']$. Now, $\Pr_2[\text{BAD}'']$ is easier to upper bound, as we now show. First note that $\Pr_2[\text{BAD}''] = \Pr_3[\text{BAD}'']$, as the two experiments Expt_2 and Expt_3 are identically distributed.

Recall, in Expt_3, S is a simulator stipulated for each distinguisher in the indifferentiability claim, and thus it is defined given A, A and $\mathcal{C}^*$. It may also be a probabilistic algorithm. However, for fixed algorithms $\mathcal{C}^*$, A and A, it is also a fixed probabilistic algorithm.

Now, consider a 2-oracle distinguisher D which is built as follows by also using the uninstantiated 1-oracle S as a component (not to be confused with it being used as an oracle). The distinguisher D consists of composition of the 2-oracle $\mathcal{C}$ and 1-oracle A as in Expt_3, except for the following change: for each of the $i \in [1..q'']$ calls of A to its oracle, it also uses S internally to see if S's L oracle calls collide with $(\mathbf{a}||\mathbf{r})$. Finally, the distinguisher D outputs 1 iff event BAD'' happens, with its two oracles instantiated by RO and S^{RO}.

Now by indifferentiability of KDF^π and π from RO and S^{RO}, the above probability of D outputting 1 remains same if we go back to using KDF^π and π as the two oracles of D.

Next, consider D' which is same as D but replaces the encryption of $\phi(\mathbf{a})$ by $\mathcal{C}^*$ by a constant string of the same length. Since in D and D', $\mathcal{C}^*$ is using a random and independent ν-bit value as key (i.e. independent of $\mathbf{a}$), by CPA-security of $\mathcal{C}^*$, $|\Pr[D = 1] - \Pr[D' = 1]| < \epsilon_1$.

Since as component of D', the view of A is independent of $\mathbf{a}$, the probability of $D' = 1$ is at most $q'' * L * 2^{-\kappa}$, recalling that the min-entropy of $\mathbf{a}$ (or its distribution $\mathcal{D}$) is κ.

Thus, $\Pr_2[\text{BAD}''] = \Pr_4[\text{BAD}''] < \epsilon_1 + q'' * (L * 2^{-\kappa})$. Hence $\Pr_1[\text{BAD}] \leq 2 * \epsilon_2 + \epsilon_1 + q' * L * 2^{-\rho} + q'' * L * 2^{-\kappa}$. ■

7 Reducing Public to Private Random-Permutation Model

We start by showing that the cryptosystem IAPM-uniform (Definition 5) in the public random-permutation (RP) model is as secure as the cryptosystem IAPM-uniform in the private random-permutation model. Later, in Section 7.2, we will use Theorem 1 to prove security of IAPM in the public RP model (i.e. as per Definition 4). Recall that in the public RP model, the adversary has access to oracles f and f^{-1} which the IAPM scheme uses. Security is proven under the probability of choosing f uniformly from all random permutations on n bits, where n is the block size of the IAPM scheme. In the private RP model, the adversary *does not* have access to either f or f^{-1}.

Theorem 3. *Let g be any ϵ-xor-universal hash function from $2n$ bits to n bits. The cryptosystem IAPM-uniform(g, ν, μ) in the n-bit public random-permutation*

model is $(q, q, N, 1 - q * 2^{-n} - (2 * q * N + N(N+1)) * \epsilon)$ *as secure as the cryptosystem IAPM-uniform*(g, ν, μ) *in the* n*-bit private random-permutation model, if the environment makes at most one call to the decryption algorithm.*

Remark 4. Since all invocations of f and f^{-1} in both E-IAPM$_{f,g}$ and D-IAPM$_{f,g}$ are "guarded" by xor-universal whitening function g keyed with secret key h, it would seem that it is easy matter to show that adversarial calls to f and f^{-1} do not collide with such calls from IAPM. However, the adversary has access to the ciphertexts from the various calls the environment makes to IAPM, and it needs to be shown that the adversary gains only negligible information about the secret key h from the adaptively obtained ciphertext transcripts.

Remark 5. If the cryptosystem IAPM-uniform(g, ν, μ) with block size n in the private RP model is $(t,\ q1,\ q2,\ m,\ \epsilon_1)$-secure for message integrity, then the above restriction in the theorem statement of only a single call to the decryption algorithm D can be removed. This is so because if D is called with a ciphertext not returned by an earlier call to the encryption algorithm E, then in the private RP model it will return $\perp$ with overwhelming probability $(1 - \epsilon_1)$. Therefore, by induction, even in the public RP model $\perp$ will be returned with overwhelming probability. Hence, the environment need not make this call at all. Further, it is well-known that in the private RP model, if an authenticated-encryption scheme is IND-CPA secure and secure for message integrity, then it is IND-CCA secure (i.e. secure against chosen-ciphertext attacks) [3, 17]. Hence from the above unrestricted version of the theorem it follows that if IAPM-uniform is IND-CPA secure and secure for message integrity in the private RP model then it is also IND-CCA secure in the public RP mode.

Remark 6. While the actual IAPM encryption scheme truncates the last block to obtain the "MAC tag", for the purpose of studying security, this truncation is not required, and we can assume that the whole last block is returned to the environment. Thus, the truncation is only performed to save on the space required to represent the tag and is not a security requirement. Similarly, in OCB, ciphertext stealing is used to represent the final non-full-block ciphertext by truncating an invocation of f. Again, for security purposes, the whole output of this invocation of f can be returned.

Proof: Note that since the environment $\mathcal{E}$ and adversary $\mathcal{A}$ are not computationally bounded, we can assume that they are deterministic. Also, note that underlying probability distribution is the key h chosen uniformly from H (the ν-bit keys of g), and the choice of f as a random permutation. Thus, the space for the probability distribution is the set of pairs h and f. Any variable which is a function of h and f, will be called a random variable, and for clarity will be depicted in bold-face or capital. By the same convention, from now on, we will also denote f and h in bold-face, i.e. $\mathbf{f}$ and $\mathbf{h}$. We will refer to $\mathbf{f}$ as *the permutation*, and $\mathbf{h}$ as *the key*. Fixed values of any random variables will be denoted by small-case letters.

W.l.o.g. we can assume that the environment never repeats queries, and moreover it never calls D-IAPM with a ciphertext returned by an earlier call to

E-IAPM. All queries by $\mathcal{E}$ to E-IAPM will be called plaintexts, and the i-th such query will be denoted P^i. Individual blocks in P^i will be denoted by subscripts. All replies to such queries will be called ciphertexts, and the i-th ciphertext will be denoted by C^i, and similarly, the j-th block on C^i will be denoted C^i_j. All the C^i together will be called C. The i-th query by $\mathcal{A}$ to $\mathbf{f}$ will be denoted V^i, and i-th query to $\mathbf{f}^{-1}$ will be denoted X^i. The results of these queries will be denoted by W^i and Y^i resp. We will call the ciphertexts, W^i and Y^i together as the *transcript* $\tilde{C}$. Since, $\mathcal{A}$ and $\mathcal{E}$ are deterministic, all queries of $\mathcal{E}$ and $\mathcal{A}$ are a function of the transcript alone. The transcript itself is a random variable as it is a function of $\mathbf{f}$ and $\mathbf{h}$.

The (single) query to D-IAPM will be denoted by C' and will be called the *forged ciphertext*. It is also a function of the transcript $\tilde{C}$. Thus, given a fixed value $\tilde{c}$ of the transcript, all the plaintexts and the forged ciphertext are fixed as well (and in particular, do not depend on $\mathbf{f}$ and $\mathbf{h}$). We will call all variables which are either part of the transcript or are a function of the transcript alone (i.e. are independent of $\mathbf{f}$ and $\mathbf{h}$) as *visible variables* (these are visible to the environment). Thus, C, P, V, W, X, Y and C' are visible variables. We will refer to P' (which is the decryption of C') as a *hidden variable*, as it may not be output if the authentication test fails. However, it is computed by D-IAPM, and indeed D-IAPM further computes $T^* = \mathbf{f}(\bigoplus_{j=1}^{m} P'_j \oplus g(\mathbf{h}, \langle IV', m+1\rangle))$ to compare it (more precisely, $\mathrm{trunc}_\mu(T^*)$) with the tag T given as part of C'. We will also refer to $\bigoplus_{j=1}^{m} P'_j$ as a hidden variable P'_{m+1}. Note that hidden variables are not a function of the transcript alone, and these may also depend on $\mathbf{f}$ and $\mathbf{h}$.

We will denote values that are invoked on $\mathbf{f}$ in E-IAPM as M^i_j, and its output as N^i_j. Note, $M^i_j = P^i_j \oplus g(\mathbf{h}, \langle IV^i, j\rangle)$, and $N^i_j = C^i_j \oplus g(\mathbf{h}, \langle IV^i, j\rangle)$. Similarly, the values invoked on $\mathbf{f}^{-1}$ in D-IAPM will be denoted N'_j and its output by M'_j. Note, $N'_j = C'_j \oplus g(\mathbf{h}, \langle IV', j\rangle)$. Since P^i_j, C^i_j, C'_j (and also the IVs) are visible variables, each of these M^i_j, N^i_j and N'_j can be written as a function of $\tilde{C}$ and $\mathbf{h}$.

Thus, all inputs to invocations of $\mathbf{f}$ and $\mathbf{f}^{-1}$ in both E-IAPM and D-IAPM, except for the one used to compute T^*, have the property that they are exclusive-or of a visible variable and $g(\mathbf{h}, a)$, where a is itself a visible variable. Associate to each such invocation of $\mathbf{f}$ and $\mathbf{f}^{-1}$ a value a (for now, disregard the invocation of $\mathbf{f}$ to compute T^*). Clearly, if the IV for all the queries to E-IAPM are different, then the a values across different queries are different. Further, the a values within a query are different by design. For the forged ciphertext query to D-IAPM, if IV$'$ is different from all the IV in the E-IAPM queries, then the a values used in the D-IAPM query are also different within the D-IAPM query and different from all a values used in E-IAPM.

We will say that a block C'_j in the forged ciphertext C' is **in-place** if IV$' =$ IV^i for some i, and $C'_j = C^i_j$, and C^i_j is not the MAC tag block of ciphertext C^i. Note, in this case $N'_j = N^i_j$, and we will refer to N'_j as also being in-place.

As for the computation of T^* in D-IAPM, we will denote the input to $\mathbf{f}$ to compute T^* as M'_{m+1}. For now, we just observe that it is an exclusive-or of a hidden variable and $g(\mathbf{h}, a)$ for some visible variable a.

Now, given a fixed value of the transcript $\tilde{c}$, and a fixed value h of the key $\mathbf{h}$, define the event $\texttt{iCOL}(h, \tilde{c})$ (for internal collisions) as disjunction of some two M_j^i being same, *or* some two N_j^i being same, *or* some two N'_j being same. Define $\texttt{xCOL}(h, \tilde{c})$ (for external collision) as disjunction of some M_j^i being same as some $V^{i'}$ or some $Y^{i'}$, *or* some N_j^i being same as some $W^{i'}$ or some $X^{i'}$, *or* some $N'_{j'}$ being same as some $W^{i'}$ or some $X^{i'}$, *or* some $N'_{j'}$ that is *not* in-place being same as some N_j^i, *or* all $N'_{j'}$ are in-place and M'_{m+1} is same as some $V^{i'}$ or some $Y^{i'}$. We will refer to disjunction of iCOL and xCOL as simply COL. Finally, if we also fix a value f for $\mathbf{f}$, define $\texttt{hCOL}(f, h, \tilde{c})$ (for hidden collision) as disjunction of some M'_j ($j = 1$ to $m + 1$) being same as some $V^{i'}$ or some $Y^{i'}$.

Now, we are interested in the probability of the event $\texttt{COL}(\mathbf{h}, \tilde{C})$ or $\texttt{hCOL}(\mathbf{f}, \mathbf{h}, \tilde{C})$ happening. When neither of these events happen the view of $\mathcal{E}$ is identical in the public and private RP model. Thus, its distinguishing probability is upper bounded by the sum of the two collision probabilities[2]. The bound on the collision probabilities follows by the following lemmas 1, 2, 3 and 4.

For $\tilde{c} = (c, w, y)$, define u_c to be the number of blocks in c, u_w to be the number of blocks (queries) in w and u_y be the number of blocks (queries) in y. For any fixed $\tilde{c}$, h, define $F_{\tilde{c},h}$ to be the set of permutations as follows: If $\texttt{COL}(h, \tilde{c})$ holds then this set is empty. Otherwise, the set contains all permutations f with the following restrictions:

1. $\forall i, j : f(M_j^i(h, \tilde{c})) = N_j^i(h, \tilde{c})$,
2. $\forall i \in [1..u_w] : f(V^i(\tilde{c})) = w^i$,
3. $\forall i \in [1..u_y] : X^i(\tilde{c}) = f(y^i)$,

Define $|\tilde{c}| = u_c + u_w + u_y$. Then, for $\tilde{c}$, h, such that $\texttt{COL}(h, \tilde{c})$ does not hold, the probability $\Pr_{\mathbf{f}}[\mathbf{f} \in F_{h\tilde{c}}]$ depends only on $|\tilde{c}|$, and in particular is independent of h. Thus, for the rest of this paragraph, for any fixed $\tilde{c}$, consider any h such that $\neg\texttt{COL}(h, \tilde{c})$ holds. Moreover, define $\texttt{num}(\tilde{c})$ to be the ratio of number of permutations on 2^n blocks and $|F_{h,\tilde{c}}|$, which is same as $(2^n)!/(2^n - |\tilde{c}| - 1)!$. Note that $\Pr_{\mathbf{f}}[\mathbf{f} \in F_{h,\tilde{c}}]$ is same as $1/\texttt{num}(\tilde{c})$. In the following lemma, recall that for each fixed and deterministic adversary, the transcript $\tilde{C}$ is a function of permutation f and key h. Hence, it should more precisely be written as $\tilde{C}(f, h)$.

Lemma 1. *For any fixed $\tilde{c} = (c, x, z)$, any fixed h such that $\neg\texttt{COL}(h, \tilde{c})$, and any fixed f, $\tilde{C}(f, h) = \tilde{c}$ is equivalent to $f \in F_{h,\tilde{c}}$.*

Proof: That $\tilde{C}(f, h) = \tilde{c}$ implies $f \in F_{h,\tilde{c}}$. follows from the definition of the set $F_{h,\tilde{c}}$. The reverse direction is proved by induction over the order of adversarial queries. Since the adversary is deterministic, the first query, whether P^i or V^1 or X^1 is fixed. In case the first query was P^1 (and first IV is IV^i), given the fixed h, it also fixes M^1, which then leads to $C_j^1 = g(h, \langle IV^1, j\rangle) \oplus f(M_j^1)$. But,

[2] Actually, the distinguishing probability is upper bounded by sum of xCOL and hCOL, but it will be difficult to bound this probability without also bounding iCOL.

$N_j^1(h, \tilde{c})$ is defined as $g(h, \langle \mathrm{IV}^1, j \rangle) \oplus c_j^1$. Thus, $C_j^1 = N_j^i(h, \tilde{c}) \oplus c_j^1 \oplus f(M_j^1)$. Since, $f \in F_{h,\tilde{c}}$, by definition of the set $F_{h,\tilde{c}}$, we have $N_j^i(h, \tilde{c}) = f(M_j^1)$, and hence $C_j^1 = c_j^1$.

A similar but simpler argument also shows that $W^1 = w^1$ or $Y^1 = y^1$ (in case the first query was V^1 or X^1 resp.). In other words $\tilde{C}^1 = \tilde{c}^1$. This in turn fixes the next query, and we continue the argument inductively. ■

Lemma 2. *For any* $\tilde{c} = (c, x, z)$,

$$\Pr_{\mathbf{f},\mathbf{h}}[\tilde{C} = \tilde{c} \wedge \neg \mathtt{COL}(\mathbf{h}, \tilde{c})] \;=\; \frac{1}{\mathsf{num}(\tilde{c})} * \Pr_{\mathbf{h}}[\neg \mathtt{COL}(\mathbf{h}, \tilde{c})]$$

The proof of this lemma follows easily by applying lemma 1. *Proof:*

$$\begin{aligned}
&\Pr_{\mathbf{f},\mathbf{h}}[\tilde{C} = \tilde{c} \wedge \neg \mathtt{COL}(\mathbf{h}, \tilde{c})] \\
&= \sum_h \Pr_{\mathbf{f},\mathbf{h}}[\mathbf{h} = h \wedge \tilde{C} = \tilde{c} \wedge \neg \mathtt{COL}(h, \tilde{c})] \\
&= \sum_h \Pr_{\mathbf{f},\mathbf{h}}[\mathbf{h} = h \wedge \mathbf{f} \in F_{h,\tilde{c}} \wedge \neg \mathtt{COL}(h, \tilde{c})] \\
&= \sum_h \Pr_{\mathbf{f}}[\mathbf{f} \in F_{h,\tilde{c}}] * \Pr_{\mathbf{h}}[\mathbf{h} = h \wedge \neg \mathtt{COL}(h, \tilde{c})] \\
&= \frac{1}{\mathsf{num}(\tilde{c})} * \Pr_{\mathbf{h}}[\neg \mathtt{COL}(\mathbf{h}, \tilde{c})]
\end{aligned}$$

where the first equality follows from lemma 1, and the second equality follows as $\mathbf{f}$ and $\mathbf{h}$ are independent. ■

Let u'_c be the number of blocks in C' (which is completely determined by $\tilde{c}$).

Lemma 3. *For every constant transcript* $\tilde{c}$,

$$\Pr_{\mathbf{h}}[\mathtt{COL}(\mathbf{h}, \tilde{c})] < (2(u_w + u_y) * (u_c + u'_c) + u_c(u_c + 1)) * \epsilon$$

Proof: We will assume that $\mathcal{E}$ does not repeat queries to E-IAPM, and further each such query uses a distinct IV. Since g is an xor-universal hash function, and $\mathbf{h}$ is chosen uniformly from its set of keys H (which is just all ν-bit strings), the result follows by noting that each disjunct in $\mathtt{iCOL}$ and $\mathtt{xCOL}$ compares either $g(\mathbf{h}, a)$ (for some constant value a determined by $\tilde{c}$) or $(g(\mathbf{h}, a) \oplus g(\mathbf{h}, a'))$ (for some distinct constant values a and a') with some constant value determined by $\tilde{c}$. ■

Lemma 4. *For every constant transcript* $\tilde{c}$, *and every constant* h *such that* $\neg \mathtt{COL}(h, \tilde{c})$

$$\Pr_{\mathbf{f}}[\mathtt{hCOL}(\mathbf{f}, h, \tilde{c}) \mid \mathbf{f} \in F_{h,\tilde{c}}] < (u_w + u_y) * 2^{-n}$$

Proof: Recall that event $\mathtt{hCOL}$ is the disjunction of any M'_i being same as either some $V^{i'}$ or some $Y^{i'}$. Moreover, recall that for $j = 1$ to m, $M'_j = \mathbf{f}^{-1}(N'_j)$, where $N'_j = C'_j \oplus g(\mathbf{h}, \langle \mathrm{IV}', j\rangle)$, and $M'_{m+1} = \bigoplus_{j=1}^{m} P'_j \oplus g(\mathbf{h}, \langle \mathrm{IV}', m+1\rangle)$.

Now, by lemma 1, $\mathbf{f} \in F_{h,\tilde{c}}$ implies that the random variable transcript $\tilde{C}$ is fixed to $\tilde{c}$. Since, C' is completely determined by $\tilde{C}$, the value of C' is also fixed. We will denote this fixed value of C' by c'. For each $j' = 1$ to m, there are two cases to consider:
(a) either $N'_{j'}$ is in-place and same as some N^i_j But, then $M'_{j'} = M^i_j$, and from being in-place it also follows that $P'_{j'} = P^i_j$. In addition, since $\neg\mathtt{xCOL}(h, \tilde{c})$ holds, then $M'_{j'}$ does not collide with any $V^{i'}$ or $Y^{i'}$.
(b) If $N'_{j'}$ is not in-place, then since $\neg\mathtt{COL}(h, \tilde{c})$ holds, $N'_{j'}$ does not collide with any $W^{i'}$ or $X^{i'}$, or with any other $N'_{j''}$, or with any N^i_j. Thus, $M'_{j'} = \mathbf{f}^{-1}(N'_{j'})$ is different from all $V^{i'}$ and $Y^{i'}$ (as $\mathbf{f}$ is a permutation). Also, it is uniformly random n-bit value even conditioned on $\mathbf{f} \in F_{h,\tilde{c}}$.

We also need to determine the probability of M'_{m+1} colliding with $V^{i'}$ or $Y^{i'}$. If all $N'_{j'}$ satisfied (a) above, then all $N'_{j'}$ are in place and by $\neg\mathtt{COL}(h, \tilde{c})$, it follows that M'_{m+1} does not collide with $V^{i'}$ or $Y^{i'}$. If some $N'_{j'}$ is not in-place, then by (b) above, $M'_{j'}$ is a uniformly random value, and also independent of all other $M'_{j''}$ (as $\neg\mathtt{COL}(h, \tilde{c})$ implies that $N'_{j'}$ does not collide with other $N'_{j''}$). Thus, M'_{m+1} is uniformly random n-bit value, and probability of it colliding with any $V^{i'}$ or $Y^{i'}$ is 2^{-n}. ■

Coming back to the proof of theorem 3,

$$\begin{aligned}
&\Pr[\neg\mathtt{COL}(\mathbf{h}, \tilde{C}) \ \wedge \ \neg\mathtt{hCOL}(\mathbf{f}, \mathbf{h}, \tilde{C})] \\
&= \sum_{\tilde{c},h} \Pr[\tilde{C} = \tilde{c} \wedge \mathbf{h} = h \wedge \neg\mathtt{COL}(h, \tilde{c}) \wedge \neg\mathtt{hCOL}(\mathbf{f}, h, \tilde{c})] \\
&= \sum_{\tilde{c},h} \Pr[\mathbf{f} \in F_{h,\tilde{c}} \wedge \mathbf{h} = h \wedge \neg\mathtt{COL}(h, \tilde{c}) \wedge \neg\mathtt{hCOL}(\mathbf{f}, h, \tilde{c})]
\end{aligned}$$

where the last equality follows from lemma 1. Now, each of these probabilities is the product of $\Pr[\mathbf{f} \in F_{h,\tilde{c}} \wedge \mathbf{h} = h \wedge \neg\mathtt{COL}(h, \tilde{c})]$ and $\Pr[\neg\mathtt{hCOL}(\mathbf{f}, h, \tilde{c}) | \mathbf{f} \in F_{h,\tilde{c}}]$. The latter is lower bounded by $1 - (u_w + u_y) * 2^{-n}$ by lemma 4. The former is again, by lemma 1 same as $\Pr[\tilde{C} = \tilde{c} \wedge \mathbf{h} = h \wedge \neg\mathtt{COL}(h, \tilde{c})]$. Thus, continuing the

above equations,

$$
\begin{aligned}
&\Pr[\neg\mathtt{COL}(\mathbf{h},\tilde{C}) \ \wedge \ \neg\mathtt{hCOL}(\mathbf{f},\mathbf{h},\tilde{C})] \\
&\geq (1-(u_w+u_y)*2^{-n})*\sum_{\tilde{c}}\Pr[\tilde{C}=\tilde{c}\wedge\neg\mathtt{COL}(\mathbf{h},\tilde{c})] \\
&= (1-(u_w+u_y)*2^{-n})*\sum_{\tilde{c}}\frac{1}{\mathsf{num}(\tilde{c})}*\Pr_{\mathbf{h}}[\neg\mathtt{COL}(\mathbf{h},\tilde{c})] \\
&\geq (1-(u_w+u_y)*2^{-n})* \\
&\quad (1-(2(u_w+u_y)*(u_c+u'_c)+u_c(u_c+1))*\epsilon)* \\
&\quad \sum_{\tilde{c}}\frac{1}{\mathsf{num}(\tilde{c})} \\
&\geq (1-(u_w+u_y)*2^{-n})* \\
&\quad (1-(2(u_w+u_y)*(u_c+u'_c)+u_c(u_c+1))*\epsilon)
\end{aligned}
$$

where the equality above follows by lemma 2, the second last inequality by lemma 3, and the last inequality by definition of $\mathsf{num}(\tilde{c})$, which we recall is same as $(2^n)!/(2^n-|\tilde{c}|-1)!$.

Thus, the probability of either COL or hCOL happening is at most $(u_w+u_y)*2^{-n}+(2(u_w+u_y)*(u_c+u'_c)+u_c(u_c+1))*\epsilon$. In the definition of "as secure as" (Definition 8), the cryptosystems $\mathcal{C}_1$ and $\mathcal{C}_2$ have public ideal primitives $\mathcal{F}_1$ and $\mathcal{F}_2$ resp. Here, $\mathcal{C}_1$ is IAPM in the public RP model, and $\mathcal{F}_1$ is just $\mathbf{f}$ and $\mathbf{f}^{-1}$ combined in one interface. Also, $\mathcal{C}_2$ here is IAPM in the private RP model and it does not need access to any public oracle. However, the adversary continues to have access to a different public random permutation (and its inverse). Now, note that adversary $\mathcal{A}_1$'s queries to the public oracle are (u_w+u_y) in number. This number remains the same as $\mathcal{A}_2$ is exactly the same as $\mathcal{A}_1$. The number N of total queries to $\mathcal{F}_1$ (i.e. $\mathbf{f}$ and $\mathbf{f}^{-1}$) is of course upper bounded by $(u_c+u'_c+u_w+u_y)$. Thus, IAPM-uniform in public RP model is $(q,q,N,\ 1-q*2^{-n}+(2*q*N+N(N+1))*\epsilon)$ as secure as IAPM-uniform in the private RP model. ■

7.1 General Schemes

From the structure of the proof of Theorem 3, the theorem is easily generalizable to different variants of IAPM such as OCB [23], the authenticated-encryption with associated data (AEAD) scheme due to Hawkes and Rose [14], the OCB variant for associated data OCB-AEAD [22], and the modified IAPM and OCB schemes due to Kurasawa [19]. Note that the proof in [19] just estimates the probability of event $\mathtt{COL}(\mathbf{h},\tilde{C})$, where as to upper bound it correctly it requires the detailed consideration above (see Remark 2 above).

7.2 Corollaries

In this section we state the various corollaries that obtain from the combination of theorems in Sections 5, 6, 7, and results from earlier works in the private

random-permutation model. To start with, we state a theorem from [16], which states the security of IAPM for message integrity in the private RP model.

Theorem 4. *[16] Let g be an ϵ-xor-universal H-keyed $(2n, n)$-hash function, where H is the set of all ν-bit strings ($\nu \leq n$). Let A be an adaptive adversary in the message integrity experiment in the private RP model for the authenticated-encryption scheme IAPM-uniform(g, ν, μ) with block size n. Let A make at most z queries, these totaling at most m blocks. Let A make a query with at most v blocks in the second stage. If $4m^2 < 2^n$ and $4v^2 < 2^n$, then*

$$Succ_A \leq 2^{-\mu} + (m^2 + 3v) \cdot (\epsilon + 2^{-n})$$

This theorem along with theorem 3 implies that IAPM-uniform(g, ν, μ) is secure for message integrity in the public random-permutation model, with

$$\mathrm{Succ}_A \leq 2^{-\mu} + (m^2 + 3v) \cdot (\epsilon + 2^{-n}) + q * 2^{-n} + (2 * q * m + m(m+1)) * \epsilon$$

where A makes at most z queries to the encryption oracle, these totaling at most m blocks, and A makes at most q queries to the public random permutation.

Then, using theorem 1, we get the following corollary for the composite IAPM scheme (Definition 4) that uses a key derivation function with oracle access to the same public random permutation.

Corollary 1. *Let* KDF *be an oracle algorithm such that with its oracle instantiated with a public ideal primitive π, it is $(\infty, t_S, q_1, q_2, L, \epsilon_1)$-indifferentiable from a ν-bit RO. Let g be an ϵ-xor-universal H-keyed $(2n, n)$-hash function, where H is the set of all ν-bit strings ($\nu \leq n$). Let A be a 3-oracle adaptive adversary in the message integrity experiment in the public RP model for the authenticated-encryption scheme IAPM(*KDF*$, g, \nu, \mu, \kappa)$ with block size n. Let A make at most z encryption queries, these totaling at most m blocks. Let A make a query with at most v blocks in the second stage. Let A make at most q queries to its first two oracles (the public random permutation). If $4m^2 < 2^n$ and $4v^2 < 2^n$, and $(m + q) < q_2$, then $Succ_A$ is at most*

$$2^{-\mu} + (q + m^2 + 3v) * 2^{-n} + (2 * q * m + 2m^2 + 3v) * \epsilon + L * (m + q) * 2^{-\kappa} + \epsilon_1$$

A similar corollary (with similar bounds) holds for IND-CPA security of IAPM(KDF, g, ν, μ, κ) in the public random-permutation model, again by using theorems 3 and 1, and the known result from [16] about message secrecy of IAPM-uniform in the private RP model.

As for the IND-KDM security of IAPM, we have two options. One is to consider a scheme which has arbitrarily long bit-strings as key space as long as they have min-entropy κ, or one can consider KDM security with the keys chosen randomly and uniformly from ν-bit strings. The latter is a realistic model if we assume that after applying the key-derivation function, the original κ-entropy key source is immediately and permanently deleted. This would also lead to a more efficient implementation, since for KDM security we must apply the key-derivation function to $(\mathbf{a}||\mathbf{r})$ afresh for each encryption. If $\mathbf{a}$ is the compact ν-bit

string (typically ν is either 256 bits, or 512 bits or a maximum of 1024 bits), then applying the sponge-style random oracle implementation to $(\mathbf{a}||\mathbf{r})$ with $\mathbf{r}$ at most 512 bits would only need a single application of a 1600-bit permutation to get 1024 bits random oracle output (with 576-bit security, also known as capacity). Thus, we only formally state the corollary for KDM-security of the IAPM-uniform instance. Moreover, by our general composition theorem 1, *we can continue to use the a key-derivation function built using the same public random permutation to derive this short ν-bit uniform key.* Note that theorem 2 only requires a single-use encryption scheme (see Remark 1 after that theorem). This means that we can instantiate with an IAPM scheme that does not require IVs, or the IV can be permanently set to zero.

Corollary 2. *Let* KDF *be an oracle algorithm such that with its oracle instantiated with a public ideal primitive π, it is $(\infty, t_S, q_1, q_2, L, \epsilon_1)$-indifferentiable from a ν-bit RO. Let g be an ϵ-xor-universal H-keyed $(2n, n)$-hash function, where H is the set of all ν-bit strings ($\nu \leq n$). Let A be a 3-oracle adaptive adversary in the IND-KDM experiment in the public RP model for the authenticated-encryption scheme obtained from zero-IV IAPM-uniform(g, ν, μ) with block size n and* KDF *as per Definition 9. Let A make at most z encryption queries, these totaling at most m blocks. Let A make at most q queries to its first two oracles (the public random permutation). Let A only make (kdm) queries with description of 2-oracle algorithms ϕ that make at most q_3 oracle calls. If $4m^2 < 2^n$ and $(m + q + q_3) < q_2$, then*

$$\begin{aligned} Adv_A^{\mathtt{kdm}} \leq & 2 * (q + m^2) * 2^{-n} + 2 * (2 * q * m + 2m^2) * \epsilon + \\ & 4 * \epsilon_1 + (m + q + q_3) * L * (z * 2^{-\rho} + 2^{-\nu}) \end{aligned}$$

We also need to prove that the scheme $\mathcal{C}$ as per Definition 9 instantiated with zero-IV IAPM-uniform(g, ν, μ) is secure for message-integrity. This is proven by first noting that the the adversary in the message-integrity experiment' find stage cannot distinguish between the real-world and the ideal world by Corollary 2. Thus, we can consider the adversary to be in the usual message-integrity experiment as in Section 3 for the scheme $\mathcal{C}$ (i.e. with no key-dependent message queries). The rest of the proof follows by showing that for each encryption query in the find stage, the key to IAPM-uniform is a uniformly random and independent ν-bit value. This is proven similarly to the analysis in the proof of Theorem 2. The adversary's probability of success Succ_A is same as $\text{Adv}_A^{\mathtt{kdm}}$ but with additional terms $2^{-\mu} + v * L * (z * 2^{-\rho} + 2^{-\nu})$, where v is the number of blocks in the second stage. Recall, μ is the length of the MAC tag.

8 Concrete Instance

We will instantiate the public random permutation by the permutation underlying SHA-3 [24], which in its draft standardization uses the Keccak hash function [4]. This hash function is built on a "cryptographic" permutation on 1600-bits called KECCAK $- f[1600]$, and which we will just call KECCAK from now

on. During and after the SHA-3 selection process, KECCAK has undergone extensive cryptanalysis, and is considered indistinguishable from a public random permutation. We will instantiate the public random permutation by KECCAK.

Thus, we consider block size $n = 1600$. The key source $\mathcal{K}$ min-entropy can be kept just as in encryption modes using private random permutations such as keyed-AES. This is justified by the security bounds obtained for message-integrity (and similar bounds for message secrecy) in Corollary 1. Thus, we let $\kappa = 128$ to be the min-entropy of the key-source. The ϵ-XOR-universal hash function g must have $\epsilon \leq 2^{-256}$, as there are quadratic terms $q * m * \epsilon$ in both Corollary 1 and 2. Thus, the size of the key ν for IAPM-uniform should be at least 256 as well, and we will set $\nu = 256$. We also let $\mu = 128$ to be the MAC tag length. For KDM security ρ should be 256 bits as well, though 128 bits may be enough. In the security bound obtained in Corollary 2 the dependence on ρ is given by the term $(m + q + q_3) * z * 2^{-\rho}$. Thus, the quadratic term comes from z, the total number of encryptions, and it does not lead to key-recovery, but just the possible loss of secrecy of that particular message.

The ϵ-xor-universal (2n,n)-hash function g is as follows. Let $\mathbb{F}$ be the Galois field GF(2^{256}). The key 256-bit key h to g is considered as an element of $\mathbb{F}$. The function $g(h, IV, i)$, where IV and i are less than 128-bits long and are considered elements of $\mathbb{F}$ is computed as $g(h, IV, i) = h * (IV * 2^{128} + i)$ in $\mathbb{F}$. It is extended to $n = 1600$ bits by prefixing zero bits. Note in zero-IV IAPM, this just becomes $h * i$ in $\mathbb{F}$. It is easy to see that this yields an ϵ-xor-universal hash function for inputs restricted to 128-bits, with $\epsilon = 2^{-256}$.

To be precise, here is the complete KDM-secure authenticated encryption scheme IAPM:

- In the initialization stage, let k be a key sampled from a source $\mathcal{D}$ with min-entropy κ. Run a KDF with 256-bits output on k to obtain k'. Permanently erase k.
- To encrypt a message P, choose a fresh random 256-bit R, and compute $h = \mathsf{trunc}_{256}$ (KECCAK$(k'||R)$). Run zero-IV IAPM-uniform encryption function on P with key h to obtain ciphertext C. Output $\langle R, C \rangle$.
- To decrypt a ciphertext $\langle R, C \rangle$, compute $h = \mathsf{trunc}_{256}$ (KECCAK$(k'||R)$), and run the zero-IV IAPM-uniform decryption function on C with key h. Output the result.

The KDF above can be implemented using the sponge construction [5] using KECCAK. Note that h above is obtained using a simple modification (optimization) of the sponge construction restricted to inputs that are at most 1600-bits.

8.1 Implementation

We implemented the above scheme on an Intel Xeon X5570 processor running at 3GHz, with SSE4 SIMD-instruction set and *no* native AES instruction. The above KDM-secure IAPM algorithm achieved 3250 mbps (mega-bits per sec.) on a single core on messages of size 16000 bytes. Our implementation used a double-permutation implementation of KECCAK from the Keccak pacakge, which utilizes

the 128-bit SIMD-instructions. In contrast, IAPM running with keyed-AES using the fastest AES implementation available (as per SUPERCOP [25] profiling on the machine) achieved only 968 mbps performance (note, there is no native AES support on this processor).

References

1. Advanced encryption standard (aes). National Institute of Standards and Technology (NIST), FIPS PUB 197, U.S. Department of Commerce, Nov. 2001.
2. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, Oct. 1997.
3. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, Dec. 2000.
4. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Keccak. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 313–314. Springer, May 2013.
5. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. On the indifferentiability of the sponge construction. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197. Springer, Apr. 2008.
6. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In A. Miri and S. Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 320–337. Springer, Aug. 2011.
7. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In K. Nyberg and H. M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Aug. 2002.
8. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, May 2001.
9. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, Oct. 2001.
10. R. Canetti, Y. Dodis, R. Pass, and S. Walfish. Universally composable security with global setup. In S. P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 61–85. Springer, Feb. 2007.
11. J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgård revisited: How to construct a hash function. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, Aug. 2005.
12. J. Daemen. Limitations of the Even-Mansour construction (rump session). In H. Imai, R. L. Rivest, and T. Matsumoto, editors, *ASIACRYPT'91*, volume 739 of *LNCS*, pages 495–498. Springer, Nov. 1991.
13. S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. In H. Imai, R. L. Rivest, and T. Matsumoto, editors, *ASIACRYPT'91*, volume 739 of *LNCS*, pages 210–224. Springer, Nov. 1991.
14. P. Hawkes and G. G. Rose. A mode of operation with partial encryption and message integrity. *IACR Cryptology ePrint Archive*, 2003:1, 2003.

15. C. S. Jutla. Encryption modes with almost free message integrity. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 529–544. Springer, May 2001.
16. C. S. Jutla. Encryption modes with almost free message integrity. *Journal of Cryptology*, 21(4):547–578, Oct. 2008.
17. J. Katz and M. Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In B. Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 284–299. Springer, Apr. 2000.
18. H. Krawczyk. LFSR-based hashing and authentication. In Y. Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 129–139. Springer, Aug. 1994.
19. K. Kurosawa. Power of a public random permutation and its application to authenticated encryption. *IEEE Transactions on Information Theory*, 56(10):5366–5374, 2010.
20. U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Feb. 2004.
21. B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In S. Jajodia and P. Samarati, editors, *ACM CCS 00*, pages 245–254. ACM Press, Nov. 2000.
22. P. Rogaway. Authenticated-encryption with associated-data. In V. Atluri, editor, *ACM CCS 02*, pages 98–107. ACM Press, Nov. 2002.
23. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *ACM CCS 01*, pages 196–205. ACM Press, Nov. 2001.
24. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Draft FIPS 202, 2014.
25. SUPERCOP:eBASC. http://bench.cr.yp.to/primitives-stream.html.