APE(X): Authenticated Permutation-Based Encryption with Extended Security Features

Elena Andreeva[1,2], Andrey Bogdanov[3], Atul Luykx[1,2], Bart Mennink[1,2], Nicky Mouha[1,2], and Kan Yasuda[1,4]

[1] Department of Electrical Engineering, ESAT/COSIC, KU Leuven, Belgium.
[2] iMinds, Belgium.
[3] Department of Mathematics, Technical University of Denmark, Denmark.
[4] NTT Secure Platform Laboratories, Japan.

Abstract. Lightweight cryptography deals with cryptographic algorithms suitable for extremely constrained devices. In such environments, it is often costly to avoid nonce reuse (secure PRNGs and non-volatile memory for counter storage might be too expensive). At the same time, a lot of cryptographic schemes actually require the nonce assumption for their security, including most authenticated encryption schemes.
This paper proposes APE, the first sponge based authenticated encryption scheme that is resistant against nonce misuse. We formally prove that APE is secure, based on the security of the underlying permutation. Furthermore, as a fully secure implementation is sometimes not feasible in lightweight devices, we consider the scenario that at some point, an attacker gets hold of the secret key. For this, we propose APEX (APE eXtended), which provides a countermeasure against key compromise. In order to decrypt, the attacker not only has to know the key, but also every associated data and ciphertext block and the tag. This property is related to the all-or-nothing encryption proposed by Rivest (FSE '97). However, APEX is the first design of an encryption mode with the all-or-nothing property that is both online and does not require a PRNG, and is therefore suitable for lightweight devices.
APE(X) can be instantiated with the permutations underlying such recent hash function designs as Quark, PHOTON or Spongent.

Keywords. APE, APEX, Authenticated Encryption, Sponge Function, Deterministic, Permutation-based, Misuse Resistant, All-or-Nothing, Key Leakage.

1 Introduction

Privacy and integrity are important cryptographic goals for building secure applications. However, constrained environments may make standard solutions prohibitively expensive to implement. Lightweight cryptography deals with cryptographic algorithms within the stringent requirements imposed by devices such as low-cost smart cards, sensor networks and electronic body implants where energy, power or hardware area consumption can be heavily restricted.

Although symmetric-key cryptography predominantly makes use of block cipher based solutions, the recently proposed framework of sponge functions [6] is quickly gaining acceptance in general and for lightweight devices in particular. Lightweight sponge based hash functions include Quark [1,2], PHOTON [13], and Spongent [7]. But not only lightweight devices benefit from the sponge construction: the SHA-3 winner Keccak [4] is a sponge based hash function that performs well on a wide variety of platforms.

Lightweight applications in practice require not only hash functions but also secret-key cryptographic functions, such as authenticated encryption (AE). AE is a cryptographic primitive that guarantees two security goals: privacy and integrity. The prevalent solutions in this direction

are block cipher based [20,24,29]. Permutation based AE schemes were only recently presented: the deterministic key-wrap scheme [14] of Khovratovich and SpongeWrap [3,5] of the KECCAK team.

These two constructions unfortunately have their limitations. With the key-wrap scheme [14], the message length is restricted to one block by design. While sufficient for key wrapping [25], this severely restricts the applicability of the scheme. SpongeWrap [3,5] can encrypt messages of varying lengths but relies on the uniqueness of the nonce value: failure to ensure so would result in the possibility to reuse the keystream in the encryption.

In Rogaway's security formalism of nonce based encryption [22,23], the nonce is considered to be unique for every evaluation. While this approach has theoretical merits, in practice it is challenging to ensure that a nonce is never reused. This is especially the case in lightweight cryptography, as a nonce is realized either by keeping a state (and correctly updating it) or by a secure pseudo-random number generator (PRNG).

Indeed, nonce misuse appears in plenty of practical applications, not necessarily limited to the lightweight setting. Examples include flawed implementations of nonces [8,10,17,19,30], bad management of nonces by the user, and backup resets or virtual machine clones when the nonce is stored as a counter.

Nonce *misuse resistance* has become an important criterion in the design of AE schemes. The upcoming CAESAR competition [9] considers misuse resistance in detail for their selection of a portfolio of AE algorithms. The problem of nonce misuse has also been addressed by the recent deterministic AE scheme SIV [25], by the online AE scheme McOE [11], and in part by the aforementioned deterministic key-wrap scheme [14]. However, no permutation based AE proposals that are resistant to nonce misuse and handle data of arbitrary length are known up to now.

APE

As a first contribution of this work, we introduce APE (Authenticated Permutation-based Encryption) in Sect. 3. APE is the first permutation based *and* nonce misuse resistant authenticated encryption scheme. Here, we initially focus on associated data and messages of an integral number of blocks. In Sect. 4 we prove that APE achieves privacy and integrity up to about $2^{c/2}$ queries, where c is the capacity parameter of APE. APE is derived from the sponge hash function design, yet tweaks are applied to make it nonce misuse resistant. In encryption APE processes data in an online manner but decryption is done backwards by requiring the inverse of the permutation as an artefact of the tweaks introduced to support nonce-misuse. We point out potential efficient implementations of permutations in both the forward and inverse direction in Sect. 3. In App. A, we show how APE can be generalized at almost no extra cost to handle fractional associated data and message blocks.

APEX

Cryptographic algorithms should not only be secure, but must also be implemented in a secure way. This is especially important for lightweight devices: often the end-user has access to the cryptographic device, but should not be able to extract the secret key K stored on it. However in an insecure implementation, K may be gleaned from unprotected memory, or recovered by a side channel attack [12,15,16].

In applications where secret keys are in the hands of the end-users, the user is often the weakest link. It is known that people often choose weak passwords, and reuse the same password for several accounts [28]. Or software may be unpatched against recent security vulnerabilities, leaving the door open for attackers [27]. Therefore, as a second contribution, we aim at strengthening APE, so that it still offers a reasonable level of security even if the attacker gets hold of the secret key K. Similar to the case of nonce reuse, this will obviously degrade the security of the AE scheme.

One possible countermeasure against key compromise is to prevent decryption if one ciphertext block, one associated data block, or the tag is missing. This property is related to the all-or-nothing encryption proposed by Rivest [21]. However, all existing all-or-nothing schemes either require a PRNG (as in Rivest's original scheme), or are not single-pass (as in [18]), thereby making them unsuitable for lightweight applications.

In Sect. 5, we introduce APEX (APE eXtended), which is a variant of APE that supports our aforementioned all-or-nothing property. More specifically, even if the secret key used by APEX is compromised, an attacker cannot decrypt if either the tag or any of the associated data or ciphertext blocks are unknown. APEX is a variant of APE where the last ciphertext block is replaced by the XOR of all ciphertext blocks and the keyed hash of the authenticated data.

2 Notation

Set $\mathbf{R} := \{0,1\}^r$ and $\mathbf{C} := \{0,1\}^c$. Given two strings A and B, we use (A,B), $A\|B$ and AB interchangeably, so for example $(A,B) \in \mathbf{R}\times\mathbf{C} \cong \{0,1\}^{r+c} \ni A\|B = AB$. Given $X \in \mathbf{R}\times\mathbf{C}$, X_r denotes its rate part and X_c its capacity part. We write $0 \in \mathbf{R}$ for a shorthand for $00\cdots0 \in \mathbf{R}$ and $1 \in \mathbf{C}$ for $00\cdots01 \in \mathbf{C}$. The symbol $\oplus$ denotes the bitwise XOR operation of two (or more) strings.

An element of $\mathbf{R}$ is called a block. Let $\mathbf{R}^*$ denote the set of strings whose length is a multiple of r, at most $2^{c/2}$ blocks. Similarly, let $\mathbf{R}^+$ denote the set of strings whose length is a positive multiple of r, at most $2^{c/2}$ blocks. Given $M \in \mathbf{R}^+$, we divide it into blocks and write $M[1]M[2]\cdots M[w] \leftarrow M$, where each $M[i]$ is a block and w the block length of the string M.

Let $\mathcal{A}$ be some class of computationally bounded adversaries. For convenience, we use the notation

$$\Delta_{\mathcal{A}}[f,g] := \sup_{A\in\mathcal{A}} \Big|\Pr[A^f] - \Pr[A^g]\Big|$$

to denote the supremum of the distinguishing advantages over all adversaries distinguishing f and g. Providing access to multiple algorithms is denoted with a "$\wedge$", e.g. $\Delta[f_1 \wedge f_2, g_1 \wedge g_2]$ denotes distinguishing the combination of f_1 and f_2 from the combination of g_1 and g_2.

3 APE Authenticated Encryption Mode

We now define our APE mode for the case of plaintexts and associated data of length a multiple of the block size. We refer to App. A for the generalization of APE to fractional data blocks. APE iterates a fixed permutation $p : \mathbf{R}\times\mathbf{C} \to \mathbf{R}\times\mathbf{C}$ in a way similar to the sponge construction. The permutation p is the only underlying cryptographic primitive used by APE.

APE consists of two functionalities, encryption $\mathcal{E}$ and decryption $\mathcal{D}$. These are defined in Fig. 1. The definitions of the subroutines can be found in Fig. 4. Also, we provide pictorial representations in Figs. 2 and 3.

Algorithm 1: $\mathcal{E}_K(A, M)$

Input: $K \in \mathbf{C}$, $A \in \mathbf{R}^*$, $M \in \mathbf{R}^+$
Output: $C \in \mathbf{R}^+$, $T \in \mathbf{C}$

1 **if** $A = \varnothing$ **then**
2 $\quad IV \leftarrow (0, K) \in \mathbf{R} \times \mathbf{C}$
3 **else**
4 $\quad IV \leftarrow \mathsf{hash\text{-}data}_{(0,K)}(A)$
5 **end**
6 $(C, \widehat{V}_c) \leftarrow \mathsf{enc\text{-}message}_{IV}(M)$
7 $T \leftarrow \widehat{V}_c \oplus K$
8 **return** (C, T)

Algorithm 2: $\mathcal{D}_K(A, C, T)$

Input: $K \in \mathbf{C}$, $A \in \mathbf{R}^*$, $C \in \mathbf{R}^+$, $T \in \mathbf{C}$
Output: $M \in \mathbf{R}^+$ or $\perp$

1 **if** $A = \varnothing$ **then**
2 $\quad IV \leftarrow (0, K) \in \mathbf{R} \times \mathbf{C}$
3 **else**
4 $\quad IV \leftarrow \mathsf{hash\text{-}data}_{(0,K)}(A)$
5 **end**
6 $(M, V_c) \leftarrow \mathsf{dec\text{-}ctxt}_{IV_r}(C, T \oplus K)$
7 **if** $IV_c = V_c$ **then**
8 $\quad$ **return** M
9 **else**
10 $\quad$ **return** $\perp$
11 **end**

Fig. 1. The encryption $\mathcal{E}_K(A, M)$ and decryption $\mathcal{D}_K(A, C, T)$ algorithms of APE. In Fig. 4, the subroutines $\mathsf{enc\text{-}message}_{IV}(M)$, $\mathsf{dec\text{-}ctxt}_{IV_r}(C, \widehat{V}_c)$ and $\mathsf{hash\text{-}data}_V(A)$ are defined.

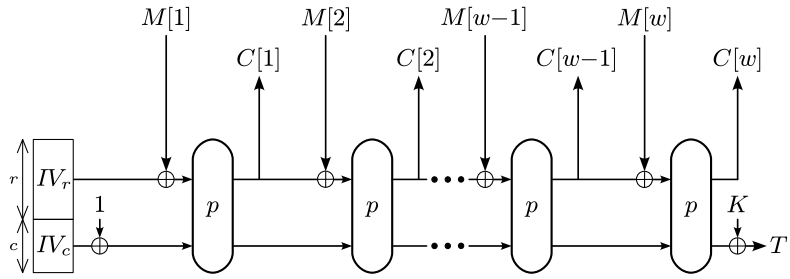

Fig. 2. The APE mode of operation (encryption): IV is computed from associated data (see Fig. 3). If there is no associated data ($A = \varnothing$), set $IV_r := 0$ and $IV_c := K$.

APE can be used with any permutation that is indistinguishable from a random permutation. For fast decryption, an efficiently invertible permutation is required. For lightweight applications where software code size or hardware area size are critical, we recommend to use a permutation with a (generalized) Feistel structure. Using such a permutation, implementing both the permutation and its inverse only slightly increases the size of the implementation.

The encryption algorithm $\mathcal{E}$ takes as input a key $K \in \mathcal{K} = \mathbf{C}$,[5] associated data $A \in \mathbf{R}^*$ and a message $M \in \mathbf{R}^+$ and returns a pair of ciphertext $C \in \mathbf{R}^+$ and a tag $T \in \mathbf{C}$, as $(C, T) \leftarrow \mathcal{E}_K(A, M)$. On the other hand, $\mathcal{D}$ takes as input a key $K \in \mathbf{C}$, associated data $A \in \mathbf{R}^*$, a ciphertext $C \in \mathbf{R}^+$, and a tag $T \in \mathbf{C}$, and returns either a message $M \in \mathbf{R}^+$ or the reject symbol $\perp$, as $M/\perp \leftarrow \mathcal{D}_K(A, C, T)$. The two functionalities are sound, in the sense that whenever we encrypt a message as $(C, T) \leftarrow \mathcal{E}_K(A, M)$, we always get the message back, not $\perp$, via the decryption process $M \leftarrow \mathcal{D}_K(A, C, T)$.

[5] The default key size is c bits, even though the security bounds given in Sect. 4 are only up to $c/2$ bits. However, we can use, for example, a $\lceil c/2 \rceil$-bit K and pad it as $0^{\lfloor c/2 \rfloor} \| K$. The security proofs can then be modified accordingly.

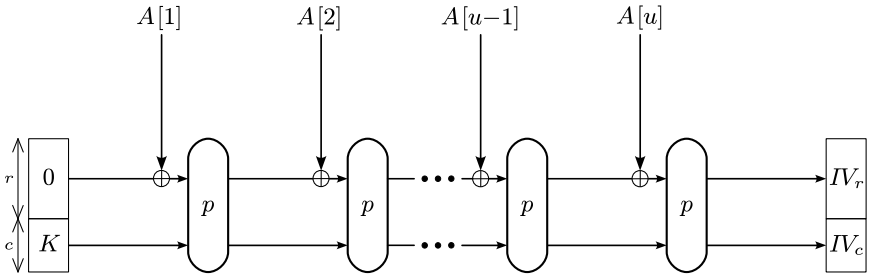

Fig. 3. Hashing associated data A in APE: IV used in Fig. 2 is computed as in this figure for encrypting the message M.

4 Privacy and Integrity of APE

In this section, we prove that APE of Sect. 3 satisfies privacy (CPA) and integrity security up to about $c/2$ bits. Firstly, in Sect. 4.1, we present the security model, where we formalize the notion of an ideal online function, and where we introduce the CPA and integrity security definitions. Then, privacy is proven in Sect. 4.2 and integrity in Sect. 4.3.

4.1 Security Model

Let $\mathrm{Perm}(n)$ be the set of all permutations on n bits. By $\perp$, we denote a function that returns $\perp$ on every input. When writing $x \xleftarrow{\$} X$ for some finite set X we mean that x is sampled uniformly from X. To avoid confusion, for $X \in \mathbf{R} \times \mathbf{C}$ we will sometimes write $[X]_c := X_c$ to denote the projection of X onto $\mathbf{C}$.

Definition 1 (Ideal Online Function). *Let $g : \mathbf{R}^+ \to \mathbf{R}$ and $g' : \mathbf{R}^+ \to \mathbf{C}$ be random functions. Let $h : \mathbf{R}^* \to \mathbf{R} \cup \{\varnothing\}$ be a random function with the property that $h(X) = \varnothing$ if and only if $X = \varnothing$. Then, on input of (A, M) with $w = |M|/r$, we define $\$: \mathbf{R}^* \times \mathbf{R}^+ \to \mathbf{R}^+ \times \mathbf{C}$ as*

$$\$(A, M[1], M[2], \ldots, M[w]) = (C[1], C[2], \ldots, C[w], T),$$

where

$$\begin{aligned} V &= h(A) \\ C[1] &= g(V, M[1]) \\ C[2] &= g(V, M[1], M[2]) \\ &\vdots \\ C[w] &= g(V, M[1], \ldots, M[w]) \\ T &= g'(V, M[1], \ldots, M[w]). \end{aligned}$$

We use the notions of CPA security and integrity of authenticated encryption schemes from Rogaway and Zhang [26] and Fleischmann et al. [11].

Algorithm 3: $\mathsf{enc\text{-}message}_{IV}(M)$

Input: $IV \in \mathbf{R} \times \mathbf{C}$, $M \in \mathbf{R}^+$
Output: $C \in \mathbf{R}^+$, $\widehat{V}_c \in \mathbf{C}$

$V \leftarrow IV \oplus (0,1)$
$M[1]M[2]\cdots M[w] \leftarrow M$
for $i = 1$ **to** w **do**
  $\widehat{V} \leftarrow p\big(M[i] \oplus V_r, V_c\big)$
  $C[i] \leftarrow \widehat{V}_r$
  $V \leftarrow \widehat{V}$
end
return $\big(C[1]C[2]\cdots C[w], \widehat{V}_c\big)$

Algorithm 4: $\mathsf{dec\text{-}ctxt}_{IV_r}(C, \widehat{V}_c)$

Input: $IV_r \in \mathbf{R}$, $C \in \mathbf{R}^+$, $\widehat{V}_c \in \mathbf{C}$
Output: $M \in \mathbf{R}^+$, $V_c \in \mathbf{C}$

$C[1]C[2]\cdots C[w] \leftarrow C$
$C[0] \leftarrow IV_r$
for $i = w$ **to** 1 **do**
  $V \leftarrow p^{-1}\big(C[i], \widehat{V}_c\big)$
  $M[i] \leftarrow C[i-1] \oplus V_r$
  $\widehat{V}_c \leftarrow V_c$
end
return $\big(M[1]M[2]\cdots M[w], V_c \oplus 1\big)$

Algorithm 5: $\mathsf{hash\text{-}data}_{V}(A)$

Input: $V \in \mathbf{R} \times \mathbf{C}$, $A \in \mathbf{R}^+$
Output: $\widehat{V} \in \mathbf{R} \times \mathbf{C}$

$A[1]A[2]\cdots A[u] \leftarrow A$
for $i = 1$ **to** u **do**
  $\widehat{V} \leftarrow p\big(A[i] \oplus V_r, V_c\big)$
  $V \leftarrow \widehat{V}$
end
return $\widehat{V}$

Fig. 4. The subroutines $\mathsf{enc\text{-}message}_{IV}(M)$, $\mathsf{dec\text{-}ctxt}_{IV_r}(C, \widehat{V}_c)$ and $\mathsf{hash\text{-}data}_{V}(A)$ which are used in Fig. 1 of the APE algorithms.

Definition 2. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ denote an authenticated encryption scheme. The CPA advantage of a distinguisher D is defined as*

$$\mathbf{Adv}^{\mathrm{cpa}}_{\Pi}(D) = \left| \begin{array}{r} \Pr\left[p \xleftarrow{\$} \mathrm{Perm}(n)\,,\, K \xleftarrow{\$} \mathcal{K} \;:\; D^{\mathcal{E}_K, \perp, p, p^{-1}} = 1\right] - \\ \Pr\left[p \xleftarrow{\$} \mathrm{Perm}(n) \;:\; D^{\$, \perp, p, p^{-1}} = 1\right] \end{array} \right| .$$

By $\mathbf{Adv}^{\mathrm{cpa}}_{\Pi}(q, m)$ we denote the supremum taken over all distinguishers making q queries of total length m blocks.

Definition 3. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ denote an authenticated encryption scheme. The integrity advantage of a distinguisher D is defined as*

$$\mathbf{Adv}^{\mathrm{int}}_{\Pi}(D) = \left| \begin{array}{r} \Pr\left[p \xleftarrow{\$} \mathrm{Perm}(n)\,,\, K \xleftarrow{\$} \mathcal{K} \;:\; D^{\mathcal{E}_K, \mathcal{D}_K, p, p^{-1}} = 1\right] - \\ \Pr\left[p \xleftarrow{\$} \mathrm{Perm}(n) \;:\; D^{\mathcal{E}_K, \perp, p, p^{-1}} = 1\right] \end{array} \right| .$$

By $\mathbf{Adv}^{\mathrm{int}}_{\Pi}(q, m)$ we denote the supremum taken over all distinguishers making q queries of total length m blocks.

In the above definitions, we consider the strongest possible type of distinguishers: D is an information-theoretic distinguisher which has unbounded computational power and whose

complexity is measured solely by the number of queries it makes to its oracles. Without loss of generality we may restrict ourselves to distinguishers which do not ask "trivial" queries. Trivial queries are either repeated queries, or inverse queries for which the forward query has previously been asked. For example, a query $p(x)$ is never followed by a query $p^{-1}(y) = x$, and a query $\mathcal{D}_K(C,T)$ is never followed by a query $\mathcal{E}_K(A,M) = (C,T)$. This allows us to perform a PRP-PRF switch as a first step: both the CPA and integrity advantages of D are defined with access to a permutation (p, p^{-1}), which we can switch to access to random functions (f, f^{-1}). Here, f and f^{-1} are defined to provide a random answer to every new query (independently of each other).

Both Defs. 2 and 3 can be viewed as a distinguisher comparing a real world and an ideal world. If (R, p) denotes a real world, (I, p) an ideal world then

$$\Delta_D[R \wedge p, I \wedge p] \leq \Delta_D[R \wedge p, R \wedge f] + \Delta_D[R \wedge f, I \wedge p],$$

where $\Delta_D[R \wedge p, R \wedge f]$ is a PRP-PRF switch, and is bounded by at most $m^2/2^{b+1}$. We can perform a similar switch with (I, p) to get

$$\Delta_D[R \wedge p, I \wedge p] \leq \frac{m^2}{2^b} + \Delta_D[R \wedge f, I \wedge f], \tag{1}$$

where we recall that the distinguisher makes at most q queries of total length m blocks (each block corresponds to a new (p, p^{-1})-query). Now, in order to analyze the CPA and integrity security of our authenticated encryption scheme, it suffices to consider the distance $\Delta_D[R \wedge f, I \wedge f]$, and our result is obtained via (1).

4.2 Privacy

In this section, we present a privacy security proof for our APE authenticated encryption scheme.

Theorem 1. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ denote the APE authenticated encryption scheme of Sect. 3. Then,*

$$\mathbf{Adv}^{\mathrm{cpa}}_{\Pi}(q, m) \leq \frac{m^2}{2^b} + \frac{m^2}{2^c}.$$

Proof. We consider a CPA distinguisher which has oracle access to one of the two worlds, either $(\mathcal{E}_K, f, f^{-1})$ or $(\$, f, f^{-1})$ (for simplicity and without loss of generality we can drop out the $\perp$). Here, in the real world R represents $\mathcal{E}_K$ and in the ideal world I represents $\$$. Note that by construction f^{-1} is independent of $\mathcal{E}_K$ and f, hence no distinguisher will gain any advantage from querying f^{-1}. In our analysis we will not consider f^{-1}-queries and will represent both worlds by (F, f), for $F \in \{\mathcal{E}_K, \$\}$.

If f is called by D then we call this a direct f-query; a call of f by $\mathcal{E}_K$ (as a result of D calling $\mathcal{E}_K$) is called an indirect f-query. Every indirect f-query has a sequence of associated data blocks and message blocks leading up to it (from the $\mathcal{E}_K$-query calling it); we call this sequence the message chain associated to the indirect f-query. When we do not specify whether an f-query is indirect or direct, we mean that it could be either.

Let $\mathcal{Q}_i$ denote the set of all prefixes of all queries made by D to its F-oracle before the ith f-query, where an F-query (A, M) results in prefixes $\{A[1], A[1]\|A[2], \ldots, A\|M[1], \ldots, A\|M\}$.

Furthermore, we denote by X_i^{d} the set of all capacity values input to *direct* f-queries before the ith f-query, and by X_i^{i} the set of all capacity values input to *indirect* f-queries before the ith f-query. We write $X_i = X_i^{\mathrm{d}} \cup X_i^{\mathrm{i}}$, and initialize $X_0^{\mathrm{i}} = \{K\}$.

For the analysis of the real world $(\mathcal{E}_K, f)$, we define event $\mathsf{E}_i = \mathsf{E}_i^{\mathrm{d}} \cup \mathsf{E}_i^{\mathrm{i}}$, where

$\mathsf{E}_i^{\mathrm{d}}$: direct query $f(x)$ satisfies $[x]_c \in X_i^{\mathrm{i}} \cup X_i^{\mathrm{i}} \oplus 1$,

$\mathsf{E}_i^{\mathrm{i}}$: indirect query $f(x)$ with corresponding message chain $(A, M) \notin \mathcal{Q}_i$ satisfies $[f(x)]_c \in X_i \cup X_i \oplus 1$.

We furthermore define

$$\hat{\mathsf{E}}_i := \mathsf{E}_i \cap \textstyle\bigcap_{j=1}^{i-1} \overline{\mathsf{E}_j}, \text{ and } \mathsf{E} := \textstyle\bigcup_{i=1}^{m} \hat{\mathsf{E}}_i, \tag{2}$$

where $\overline{\mathsf{E}_j}$ is the complement of E_j.

Now, the remainder of the proof is divided as follows. Firstly, in Lem. 1 (in App. B.1) we will prove that, as long as E does not occur, $(\mathcal{E}_K, f)$ and $(\$, f)$ are indistinguishable. By (1) and the fundamental lemma of game playing this implies

$$\mathbf{Adv}_{\Pi}^{\mathrm{cpa}}(q, m) = \Delta_D[R \wedge p, I \wedge p] \leq \frac{m^2}{2^b} + \Pr[\mathsf{E}].$$

Then, in Lem. 2 (in App. B.1), we will prove that $\Pr[\mathsf{E}] \leq \dfrac{m^2}{2^c}$, which completes the proof. □

4.3 Integrity

In this section, we present an integrity security proof for our APE authenticated encryption scheme.

Theorem 2. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ denote the APE authenticated encryption scheme of Sect. 3. Then,*

$$\mathbf{Adv}_{\Pi}^{\mathrm{int}}(q, m) \leq \frac{m^2}{2^b} + \frac{3m^2}{2^c}.$$

Proof. The basic idea of the proof is the same as for Thm. 1, but we need to take into account inverse queries too.

We consider an integrity distinguisher which has query access to one of the two worlds, either $(\mathcal{E}_K, \mathcal{D}_K, f, f^{-1})$ or $(\mathcal{E}_K, \perp, f, f^{-1})$. Here, in the real world R represents $(\mathcal{E}_K, \mathcal{D}_K)$ and in the ideal world I represents $(\mathcal{E}_K, \perp)$.

If f (resp. f^{-1}) is called by D then we call this a direct f-query (resp. f^{-1}-query); a call of f by $\mathcal{E}_K$ or $\mathcal{D}_K$ (as a result of D calling them) is called an indirect f-query, and similar for f^{-1}. Every indirect f-query has a sequence of associated data blocks and/or message blocks leading up to it (from the $\mathcal{E}_K$- or $\mathcal{D}_K$-query calling it); we call this sequence the message chain associated to the indirect f-query. Every indirect f^{-1}-query has a tag and a sequence of ciphertext blocks leading up to it, and we call this sequence the associated ciphertext chain.

We use similar notation as in the proof of Thm. 1, but as the proof now also involves inverse queries, slightly more involved definitions are needed, and we re-introduce them. Let $\mathcal{Q}_i$ denote

the set of all prefixes of all queries made by D to its F-oracle before the ith (f, f^{-1})-query, where an F-query (A, M) results in prefixes $\{A[1], A[1]\|A[2], \ldots, A\|M\}$. In this set, we also include $\{A[1], \ldots, A\}$ for an F^{-1}-query (A, C, T). Let $\mathcal{Q}_i^{-1}$ denote the set of all suffixes of all queries made by D to its F^{-1}-oracle before the ith query, where an F^{-1}-query (A, C, T) results in suffixes $\{C[w]\|T, C[w-1]\|C[w]\|T, \ldots, C\|T\}$. (The tag value T is included here for technical reasons.) Furthermore, regarding all *direct* queries before the ith query, we denote by X_i^{d} the set of all capacity values input to f-queries or output of f^{-1}-queries, and by Y_i^{d} the set of all capacity values input to f^{-1}-queries or output of f-queries. For example, a direct forward query $f(x) \to y$ adds x to X_i^{d} and y to Y_i^{d}, and a direct inverse query $f^{-1}(y) \to x$ adds x to X_i^{d} and y to Y_i^{d}. The sets X_i^{i} and Y_i^{i} are defined similarly. We write $X_i = X_i^{\mathrm{d}} \cup X_i^{\mathrm{i}}$ and $Y_i = Y_i^{\mathrm{d}} \cup Y_i^{\mathrm{i}}$, and initialize $X_0^{\mathrm{i}} = Y_0^{\mathrm{i}} = \{K\}$.

We define event $\mathsf{E}_i = \mathsf{E}_i^{\mathrm{d}} \cup \mathsf{E}_i^{\mathrm{i}} \cup \mathsf{E}_i^{\mathrm{d\text{-}}} \cup \mathsf{E}_i^{\mathrm{i\text{-}}}$, where $\mathsf{E}_i^{\mathrm{d}}$ and $\mathsf{E}_i^{\mathrm{i}}$ are as in the proof of Thm. 1 with the renewed definitions of the sets, and where

$\mathsf{E}_i^{\mathrm{d\text{-}}}$: direct query $f^{-1}(y)$ satisfies $[y]_c \in Y_i^{\mathrm{i}} \cup Y_i^{\mathrm{i}} \oplus 1$,

$\mathsf{E}_i^{\mathrm{i\text{-}}}$: indirect query $f^{-1}(y)$ with corresponding ciphertext chain $(C, T) \notin \mathcal{Q}_i^{-1}$ satisfies $[f^{-1}(y)]_c \in Y_i \cup Y_i \oplus 1 or [y]_c \in Y_i^{\mathrm{d}} \oplus K$.

Definitions $\hat{\mathsf{E}}_i$ and E are defined as before. The latter condition of $\mathsf{E}_i^{\mathrm{i\text{-}}}$, $[y]_c \in Y_i^{\mathrm{d}} \oplus K$ covers the case the distinguisher obtains the key by making a direct inverse query and a $\mathcal{D}_K$-query.

Now, the remainder of the proof is divided as follows. Firstly, in Lem. 3 (in App. B.2) we will prove that, as long as E does not occur, $(\mathcal{E}_K, \mathcal{D}_K, f, f^{-1})$ and $(\mathcal{E}_K, \perp, f, f^{-1})$ are indistinguishable. By (1) and the fundamental lemma of game playing this implies

$$\mathbf{Adv}_{\Pi}^{\mathrm{int}}(q, m) = \Delta_D[R \wedge p, I \wedge p] \leq \frac{m^2}{2^b} + \Pr[\mathsf{E}].$$

Then, in Lem. 4 (in App. B.2), we will prove that $\Pr[\mathsf{E}] \leq \dfrac{3m^2}{2^c}$, which completes the proof. □

5 APEX: an Extension of APE

The APE Authenticated Encryption mode proposed in Sect. 3 is designed to be resistant against nonce misuse: if a nonce is reused, the adversary cannot learn any information about the plaintext up to a common prefix.

We can go even further by considering the worst possible failure of the encryption system: compromise of the secret key K. This can happen either because of an insecure implementation, or because the user (on purpose or inadvertently) leaks K. Surprisingly, APE provides resistance against this scenario. An attacker who knows the secret key K but not the tag T or the last ciphertext block $C[w]$, cannot gain any information about the plaintext. This is a useful feature of the backwards decryption requirement.

APEX (APE eXtended) is an extension of APE, with the last ciphertext block changed to make the scheme even more robust against key compromise. More specifically, the last ciphertext block is replaced by an XOR of all ciphertext blocks and IV_r, as shown in Fig. 5 (the subroutine $\mathsf{hash\text{-}data}_{0,K}$ remains unchanged). Suppose an adversary recovers the key K at some point, but is missing the tag, at least one of the associated data blocks, or one of the ciphertext blocks.

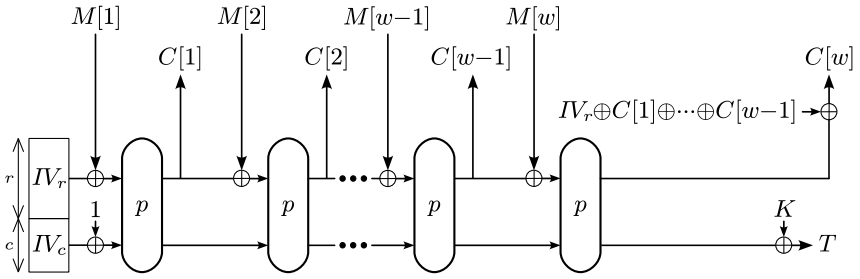

Fig. 5. APEX: an all-or-nothing variant of APE. If K is compromised, decryption is infeasible unless the adversary obtains the tag, all ciphertext blocks and all associated data blocks.

Due to the characteristic design of APEX, the adversary cannot gain any information about the plaintext. This is similar to the all-or-nothing encryption proposed by Rivest [21].

Clearly, an adversary who knows K and message blocks $M[i],\ 0 \leq i \leq r$, can also calculate ciphertext blocks $C[i],\ 0 \leq i \leq r$. Otherwise, APEX would not support online encryption. Therefore, our setting assumes that if K is known, at least one message block $M[i]$ and one ciphertext block $C[j]$ is not known, where $i \leq j$.

Regarding privacy and integrity, the proofs of Thms. 1 and 2 directly carry over. A small remark needs to be made regarding the privacy proof: note that, once IV_r and $C[1], \ldots, C[w-1]$ are fixed, and the rate part of the output of the wth p-call is random, then so is $C[w]$ (here w is the length of the input message). The integrity proof carries over without any significant modifications.

6 Conclusion

We introduced two new permutation based schemes, APE and APEX. APE is secure as an online deterministic AE scheme, with privacy and integrity up to the birthday bound of the capacity, i.e. $c/2$-bit security. To achieve misuse-resistance the decryption of APE as a sponge based construction requires using the inverse permutation to decrypt in a backwards manner. The advantage of having backwards decryption is that if the tag or last ciphertext block is missing, then decryption is impossible. Our second scheme APEX extends this extra security feature to the case of any missing ciphertext block, tag, or associated data block. We argue that at present the schemes we introduce are particularly suitable for lightweight applications. This is nowadays the case with the use of permutations based on a (generalized) Feistel structure where the inverse permutation call adds only minimal implementation cost, and decryption with any AE scheme keeps anyway the entire message in memory until the message has been verified. Finally, the added security features of APE and APEX lend themselves well to the environments in which lightweight cryptography is applied nowadays.

Acknowledgments. This work has been funded in part by the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II, in part by the Research Council KU Leuven: GOA TENSE, and in part by the Research Fund KU Leuven, OT/08/027. Elena Andreeva is supported by a Postdoctoral Fellowship from the Flemish Research Foundation

(FWO-Vlaanderen). Bart Mennink is supported by a Ph.D. Fellowship from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

1. Aumasson, J.P., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A Lightweight Hash. In: Mangard, S., Standaert, F.X. (eds.) CHES. Lecture Notes in Computer Science, vol. 6225, pp. 1–15. Springer (2010)
2. Aumasson, J.P., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A Lightweight Hash. J. Cryptology 26(2), 313–339 (2013)
3. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers (July 2012)
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak SHA-3 submission. Submission to the NIST SHA-3 Competition (Round 3) (2011), http://keccak.noekeon.org/Keccak-submission-3.pdf
5. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In: Miri, A., Vaudenay, S. (eds.) Selected Areas in Cryptography 2011. Lecture Notes in Computer Science, vol. 7118, pp. 320–337. Springer (2012)
6. Bertoni, G., Daemen, J., Peeters, M., Assche, G.: Sponge functions (ECRYPT Hash Function Workshop 2007), http://sponge.noekeon.org/SpongeFunctions.pdf
7. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: Spongent: A Lightweight Hash Function. In: Preneel, B., Takagi, T. (eds.) CHES. Lecture Notes in Computer Science, vol. 6917, pp. 312–325. Springer (2011)
8. Borisov, N., Goldberg, I., Wagner, D.: Intercepting mobile communications: the insecurity of 802.11. In: Rose, C. (ed.) MOBICOM. pp. 180–189. ACM (2001)
9. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (April 2013), http://competitions.cr.yp.to/caesar.html
10. Cantero, H.M., Peter, S., Bushing, Segher: Console Hacking 2010 – PS3 Epic Fail. 27th Chaos Communication Congress (December 2010)
11. Fleischmann, E., Forler, C., Lucks, S.: McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In: Canteaut, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 7549, pp. 196–215. Springer (2012)
12. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic Analysis: Concrete Results. In: Çetin Kaya Koç, Naccache, D., Paar, C. (eds.) CHES. Lecture Notes in Computer Science, vol. 2162, pp. 251–261. Springer (2001)
13. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 6841, pp. 222–239. Springer (2011)
14. Khovratovich, D.: Key Wrapping with a Fixed Permutation. Cryptology ePrint Archive, Report 2013/145 (2013)
15. Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1109, pp. 104–113. Springer (1996)
16. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M.J. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999)
17. Kohno, T.: Attacking and repairing the winZip encryption scheme. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) ACM Conference on Computer and Communications Security. pp. 72–81. ACM (2004)
18. Kuwakado, H., Tanaka, H.: Secure Length-Preserving All-or-Nothing Transform. Information and Media Technologies 1(1), 112–120 (2006)
19. Lenstra, A.K., Hughes, J.P., Augier, M., Bos, J.W., Kleinjung, T., Wachter, C.: Public Keys. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO. Lecture Notes in Computer Science, vol. 7417, pp. 626–642. Springer (2012)
20. McGrew, D.A., Viega, J.: The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT. Lecture Notes in Computer Science, vol. 3348, pp. 343–355. Springer (2004)
21. Rivest, R.L.: All-or-Nothing Encryption and the Package Transform. In: Biham, E. (ed.) FSE. Lecture Notes in Computer Science, vol. 1267, pp. 210–218. Springer (1997)
22. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM Conference on Computer and Communications Security 2002. pp. 98–107. ACM (2002)
23. Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B.K., Meier, W. (eds.) FSE 2004. Lecture Notes in Computer Science, vol. 3017, pp. 348–359. Springer (2004)

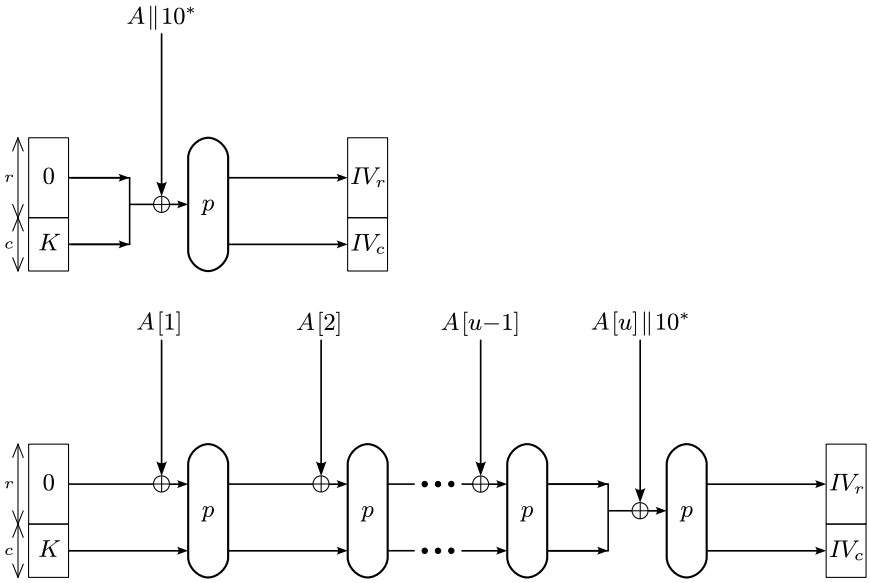

Fig. 6. A generalization of APE that can handle fractional associated data blocks.

24. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Trans. Inf. Syst. Secur. 6(3), 365–403 (2003)
25. Rogaway, P., Shrimpton, T.: A Provable-Security Treatment of the Key-Wrap Problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 373–390. Springer (2006)
26. Rogaway, P., Zhang, H.: Online Ciphers from Tweakable Blockciphers. In: CT-RSA 2011. Lecture Notes in Computer Science, vol. 6558, pp. 237–249. Springer, Heidelberg (2011)
27. SANS Institute: The Top Cyber Security Risks (2009), `http://www.sans.org/top-cyber-security-risks/`
28. Scarfone, K., Souppaya, M.: Guide to Enterprise Password Management. NIST special publication 800-118 (draft), National Institute of Standards and Technology (NIST) (April 2009), `http://csrc.nist.gov/publications/drafts/800-118/draft-sp800-118.pdf`
29. Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM). Request For Comments 3610 (2003)
30. Wu, H.: The Misuse of RC4 in Microsoft Word and Excel. Cryptology ePrint Archive, Report 2005/007

A APE for Fractional Data

The APE description of Sect. 3 applies only to plaintexts and associated data whose length is a multiple of the block size r. In this section, we demonstrate how to adjust APE to handle fractional messages and associated data. The extension of $\mathsf{hash\text{-}data}_{0,K}$ to fractional associated data is given in Fig. 6, and for fractional messages the extension of $\mathsf{enc\text{-}message}_{IV}$ is given in Fig. 7. For both the processing of authenticated data and message data, we require the last block ($A[u]$ or $M[w]$) to be of length at most $r-1$ bits. Note that in Fig. 7, the ciphertext $C[w-1]$ is of size equal to $M[w]$. The reason we opt for this design property is the following: despite $M[w]$ being smaller than r bits, we require its corresponding ciphertext to be r bits for decryption to be possible. As a toll, the extended APE generates ciphertext $C[w-1]$ to be of size equal to $M[w]$. Clearly, this has no influence on the decryption algorithm $\mathcal{D}$.

Without going into detail, we note that the security results of Sect. 4 directly carry over to APE with fractional data. Here, we rely on the fact that $A[u]$ and $M[w]$ are of size at most $r-1$ bits.

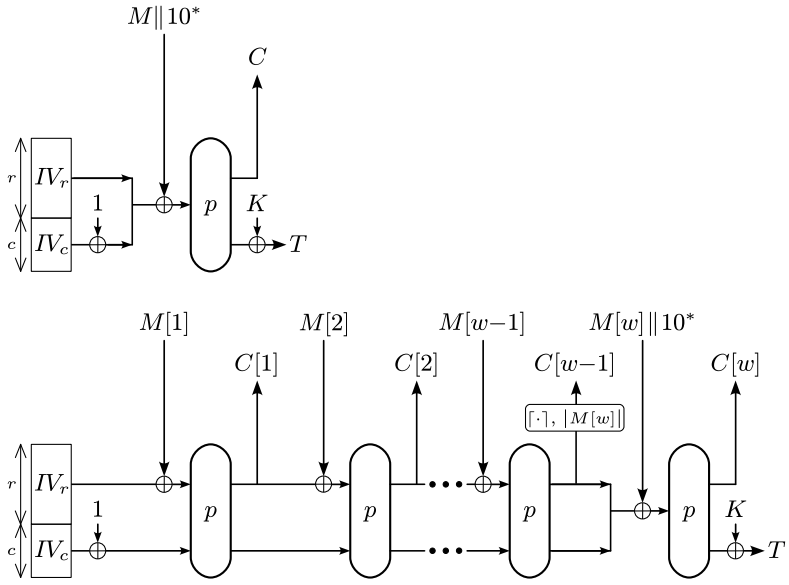

Fig. 7. A generalization of APE that can handle fractional message blocks.

B Lemmas for Privacy and Integrity of APE

B.1 Lemmas for Privacy of APE

Lemma 1. *Given that* E *does not occur,* $(\mathcal{E}_K, f)$ *and* $(\$, f)$ *are indistinguishable.*

Proof. Note that in the ideal world, each direct f-query is new, and is answered with a uniformly randomly drawn response. Now, consider a direct query $f(x)$ in the real world. As the distinguisher does not make trivial queries, it does not coincide with any previous direct query. Additionally, if $[x]_c \in X_i^{\mathrm{i}} \cup X_i^{\mathrm{i}} \oplus 1$, where $f(x)$ is the ith f-query, then this would trigger $\mathsf{E}_i^{\mathrm{d}}$, hence we can assume $[x]_c \notin X_i^{\mathrm{i}} \cup X_i^{\mathrm{i}} \oplus 1$. This means that the query $f(x)$ is truly new, and its value is independently and uniformly distributed.

Therefore, we only need to consider queries to the big oracle $F \in \{\mathcal{E}_K, \$\}$. Let (A, M) be a query made by the distinguisher. Denote by w the number of blocks of M. Denote the corresponding ciphertext and tag by (C, T).

First consider the case $(A, M') \in \mathcal{Q}_i$ for some M' (we will come back to the case of $(A, *) \notin \mathcal{Q}_i$ later in the proof), and let M' be such that it maximizes the value j such that $M[1]\| \cdots \|M[j] = M'[1]\| \cdots \|M'[j]$. Denote the corresponding ciphertext and tag by (C', T'), and the block length by w'.

Clearly, in the ideal world $(\$, f)$, we have $C[i] = C'[i]$ for $i = 1, \ldots, j$, but $C[i]$ for $i = j + 1, \ldots, w$ and T are uniformly randomly drawn. We will consider how these values are distributed in the real world $(\mathcal{E}_K, f)$. We first consider the general case $j < w$, the case $j = w$ is discussed afterwards.

1. $C[1], \ldots, C[j]$. Also in the real world, these values are equal to $C'[1], \ldots, C'[j]$, which follows clearly from the specification of $\mathcal{E}_K$. Note that in particular, the state value V equals V' after the jth round.

2. $C[j+1]$. We make a distinction between $j > 0$ and $j = 0$, and start with the former case. Write the indirect query corresponding to the jth round as $f(x)$. The input of the $(j+1)$th query will be $f(x) \oplus (M[j+1], 0)$. Suppose this query has already been made before, then either $[f(x)]_c \in X^{\mathrm{d}} \cup X^{\mathrm{i}}$, or $(A, M' \| M[j+1]) \in \mathcal{Q}$. Since $[f(x)]_c \notin X^{\mathrm{d}} \cup X^{\mathrm{i}}$, it must be that $(A, M' \| M[j+1]) \in \mathcal{Q}$, but this contradicts the fact that j is maximal. This query has been made at an earlier point in time (it may even date from before the evaluation of (A, M')), but at this particular time, the capacity part $[f(x)]_c$ did not hit any element from $X^{\mathrm{d}} \cup X^{\mathrm{i}}$ (otherwise it would have triggered E^{i}). After this query has been made, there has not been any newer indirect query or any newer direct query whose capacity part hit $[f(x)]_c$ (both cases would have triggered $\mathsf{E}^{\mathrm{d}} \cup \mathsf{E}^{\mathrm{i}}$). Thus, the query corresponding to the $(j+1)$th round is generated independently and uniformly at random.
   Now, in the case $j = 0$, denote the state coming from $\mathsf{hash\text{-}data}_{0,K}(A)$ by IV (if $A = \varnothing$, $IV = (0, K)$). The same story as before applies with the difference that now the input to the $(j+1)$th query is $IV \oplus (M[j+1], 1)$. Here we use that by $\overline{\mathsf{E}}$, no other query hit $X^{\mathrm{i}}_j \oplus 1$ (for direct queries) or $X_j \oplus 1$ (for indirect queries) in the meanwhile, and that X^{i} is initialized with $\{K\}$.
3. $C[j+2], \ldots, C[w]$. By the above argument, the indirect query made in the $(j+1)$th round of (A, M), say $f(x)$ for the sake of presentation, is responded with a uniformly random answer. This query would have triggered E^{i} if $[f(x)]_c \in X_i$. Therefore, we know that also the $(j+2)$th query is truly random and so is $C[j+2]$. The same reasoning applies up to $C[w]$.
4. T. The same reasoning applies: the previous query is responded with a truly random answer $f(x)$. Consequently $T = [f(x)]_c \oplus K$ is random too.

A special treatment is needed for $j = w$. In this case, $C[1], \ldots, C[w]$ equals $C'[1], \ldots, C'[w]$ by construction, but the query producing T is not new. Yet, the distinguisher never made that query itself by virtue of $\overline{\mathsf{E}^{\mathrm{d}}}$, so it never learnt $T \oplus K$. Besides, due to the absence of indirect capacity collisions, $\overline{\mathsf{E}^{\mathrm{i}}}$, every f-query will produce a tag at most once. This means that T will look uniformly random to the distinguisher, as it would look if it were produced by \$.

Finally, we consider the case $(A, *) \notin \mathcal{Q}_i$, hence this is the first time a query for this particular associated data A is made. Then, the above reasoning carries over for $i = 0$ with the simplification that if $A \neq \varnothing$, the value IV_c can be considered new. □

Lemma 2. $\Pr[\mathsf{E}] \leq \dfrac{m^2}{2^c}$.

Proof. Inspired by (2), we start bounding $\Pr[\mathsf{E}_i \cap \bigcap_{j=1}^{i-1} \overline{\mathsf{E}_j}]$ for $i \in \{1, \ldots, m\}$. Clearly,

$$\Pr[\mathsf{E}_i \cap \textstyle\bigcap_{j=1}^{i-1} \overline{\mathsf{E}_j}] \leq \Pr[\mathsf{E}_i \mid \textstyle\bigcap_{j=1}^{i-1} \overline{\mathsf{E}_j}].$$

Therefore, we assume $\bigcap_{j=1}^{i-1} \overline{\mathsf{E}_j}$ and consider the probability the ith query triggers E_i.

If the ith query is a direct query, it triggers $\mathsf{E}^{\mathrm{d}}_i$ if the distinguisher "guesses" a capacity part in $X^{\mathrm{i}}_i \cup X^{\mathrm{i}}_i \oplus 1$, which happens with probability at most $2|X^{\mathrm{i}}_i|/2^c$. On the other hand, if it is an indirect query that is new (hence, for which $(A, M) \notin \mathcal{Q}_i$) it triggers $\mathsf{E}^{\mathrm{i}}_i$ if $[f(x)]_c \in X_i \cup X_i \oplus 1$, hence with probability at most $2|X_i|/2^c$.

By construction, $X^{\mathrm{i}}_i \leq X_i \leq i$, henceforth we find:

$$\Pr[\mathsf{E}_i \mid \textstyle\bigcap_{j=1}^{i-1} \overline{\mathsf{E}_j}] \leq \dfrac{2i}{2^c}.$$

The result is now obtained by summing over $i = 1, \ldots, m$ (as in (2)). □

B.2 Lemmas for Integrity of APE

Lemma 3. *Given that* E *does not occur,* $(\mathcal{E}_K, \mathcal{D}_K, f, f^{-1})$ *and* $(\mathcal{E}_K, \perp, f, f^{-1})$ *are indistinguishable.*

Proof. For direct f-queries, the analysis of Lem. 1 carries over. Similarly, for inverse direct f^{-1}-queries (note that f and f^{-1} behave independently), the same reasoning applies with the difference that $[y]_c \in Y_i^{\mathsf{i}} \cup Y_i^{\mathsf{i}} \oplus 1$ would trigger event $\mathsf{E}^{\mathsf{d}\text{-}}$.

Also, queries made to $\mathcal{E}_K$ in the real and ideal world are handled the same. Here, we use that in the real world, indirect f-queries coming from $\mathcal{D}_K$ (corresponding to the evaluation of $\mathsf{hash\text{-}data}_{0,K}$), do not ruin the distribution of the responses from $\mathcal{E}_K$ by $\overline{\mathsf{E}^{\mathsf{i}}}$, where we now deal with a larger set X_i^{i}.

Therefore, we only need to consider queries to the big oracle $F \in \{\mathcal{D}_K, \perp\}$, and let (A, C, T) be a query made by the distinguisher. Denote by w the number of blocks of C. Denote the state coming from $\mathsf{hash\text{-}data}_{0,K}(A)$ by IV, and the corresponding message and capacity value by (M, V_c). Here, $V_c \oplus 1$ equals the capacity part of the call to f^{-1} corresponding to $C[1]$.

Clearly, in the ideal world $(\mathcal{E}_K, \perp, f, f^{-1})$, the query is responded with $\perp$. Therefore, the distinguisher has no advantage in the real world unless $IV_c = V_c$. We make a distinction between whether or not $(C, T) \in \mathcal{Q}_i^{-1}$.

- $(C, T) \notin \mathcal{Q}_i^{-1}$. A similar reasoning as for the value T in Lem. 1 subcase "$i < w$" results in the observation that the indirect f^{-1}-query corresponding to the ciphertext block $C[1]$ is new and the response is uniformly randomly drawn. The only fundamental difference lies in the fact that we now use events $\mathsf{E}^{\mathsf{d}\text{-}}$ and $\mathsf{E}^{\mathsf{i}\text{-}}$, and rely on the fact the first indirect query never matches a direct query $\oplus K$ (by $\overline{\mathsf{E}_i^{\mathsf{i}\text{-}}}$) or vice versa (by $\overline{\mathsf{E}_i^{\mathsf{d}\text{-}}}$). We skip the details. Now, D succeeds if the capacity part of this value, say $[f^{-1}(y)]_c$, equals $IV_c \oplus 1$, but then this indirect query would have triggered $\mathsf{E}^{\mathsf{i}\text{-}}$. We note that this value IV_c may be an older value, e.g., if $A \in \mathcal{Q}_i$, but this does not invalidate the analysis.
- $(C, T) \in \mathcal{Q}_i^{-1}$. The further reasoning depends on whether $(A, M') \in \mathcal{Q}_i$ for some M'.
  - $(A, *) \notin \mathcal{Q}_i$. If $A = \varnothing$, the distinguisher succeeding would mean that $V_c = K \oplus 1$. But this means that at some earlier point in time an indirect f^{-1}-query has hit $K \oplus 1$ and thus invalidated $\mathsf{E}^{\mathsf{d}\text{-}} \cup \mathsf{E}^{\mathsf{i}\text{-}}$. Now, if $A \neq \varnothing$, the analysis of Lem. 1 for the same case "$(A, *) \notin \mathcal{Q}_i$" carries over: the value IV_c can be considered new and if it hits $V_c \oplus 1$, the query would trigger E^{i}.
  - $(A, M') \in \mathcal{Q}_i$ for some M'. By assumption, (A, M') and (C, T) cannot correspond to one and the same query. Without loss of generality, (A, M') corresponds to an earlier query than (C, T). Now, the distinguisher's query is valid if $IV_c = V_c \oplus 1$. If this is the case, the (C, T) query must have triggered $\mathsf{E}^{\mathsf{i}\text{-}}$. Also, in case $A = \perp$, this equation could not hold as $V_c = K \oplus 1$ would have triggered $\mathsf{E}^{\mathsf{i}\text{-}}$.

This completes the proof of Lem. 3. □

Lemma 4. $\Pr[\mathsf{E}] \leq \dfrac{3m^2}{2^c}$.

Proof. The analysis is fairly similar to the proof of Lem. 2, and is skipped for conciseness. Note that $\mathsf{E}_i^{\mathsf{i}\text{-}}$ contains an additional condition, compared with $\mathsf{E}_i^{\mathsf{i}}$. □