IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2017, No. 3, pp. 271–293. DOI:10.13154/tosc.v2017.i3.271-293

On Leakage-Resilient Authenticated Encryption with Decryption Leakages*

Francesco Berti, Olivier Pereira,
Thomas Peters and François-Xavier Standaert

Université catholique de Louvain (UCLouvain), Institute of Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM) – Crypto Group, B-1348 Louvain-la-Neuve, Belgium
francesco.berti@uclouvain.be,olivier.pereira@uclouvain.be,thomas.peters@uclouvain.be,fstandae@uclouvain.be

Abstract. At CCS 2015, Pereira et al. introduced a pragmatic model enabling the study of leakage-resilient symmetric cryptographic primitives based on the minimal use of a leak-free component. This model was recently used to prove the good integrity and confidentiality properties of an authenticated encryption scheme called DTE when the adversary is only given encryption leakages. In this paper, we extend this work by analyzing the case where decryption leakages are also available. We first exhibit attacks exploiting such leakages against the integrity of DTE (and variants) and show how to mitigate them. We then consider message confidentiality in a context where an adversary can observe decryption leakages but not the corresponding messages. The latter is motivated by applications such as secure bootloading and bitstream decryption. We finally formalize the confidentiality requirements that can be achieved in this case and propose a new construction satisfying them, while providing integrity properties with leakage that are as good as those of DTE.

Keywords: Leakage-resilience, authenticated encryption, secure bootloading.

1 Introduction

The study of authenticated encryption schemes that are resilient to side-channel leakages has been recently initiated by three complementary pieces of work. On the most practical side, Dobraunig et al. described how the combination of a fresh re-keying scheme (borrowed from [DKM+15, MSGR10]) and a sponge-based construction [BDPA08] leads to a natural candidate of authenticated encryption scheme with good features to mitigate concrete side-channel attacks. In particular, the authors exhibited an appealing heuristic connection between the capacity of the sponge construction and the amount of leakage that can be tolerated [DEM+17].

On the more theoretical side, Barwell et al. [BMOS17] investigated composition results for authenticated encryption in front of protocol and side-channel leakage. They showed strong positive results under the assumption that all the building blocks of their implementations are well protected against side-channel attacks, and proposed an instantiation of these building blocks based on pairings.

Eventually, Berti et al. studied alternative solutions in a pragmatic model combining the minimal use of an (expensive) leak-free component with much more efficient (less

*Thomas Peters is a postdoctoral researcher and François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work has been funded in parts by the INNOVIRIS projects SCAUT and C-Cure, the ERC project 724725 (acronym SWORD) and the FEDER-Cryptomedia Project.

Received: 2017-06-01, Accepted: 2017-08-01, Published: 2017-09-19

protected) implementations of symmetric-key building blocks. They put forward that some security properties do not seem to be reachable in this setting (e.g., full misuse-resistance with leakage), yet that a practically-relevant restriction, formalized as Ciphertext Integrity with Misuse and Leakage (CIML), can be obtained efficiently [BKP+16]. Intuitively, CIML captures the integrity requirement that the exploitation of randomness misuse and leakage does not allow forging valid ciphertexts. Since based on a leakage-resilient encryption mode borrowed from [PSV15], the instances of CIML-secure authenticated encryption scheme proposed in this previous work also inherit good confidentiality properties. That is the confidentiality of multiple message and blocks can be reduced to the security of a single message block (of which the formalization remains an open question).

In this paper, we pursue the analysis of Berti at al. to better understand the security guarantees that can be obtained with symmetric cryptographic primitives against side-channel attacks. In this respect, one important question left open by this previous work relates to decryption leakages that were excluded from the analysis. Indeed, while justified in certain case-studies (e.g., a smart card with a secure reader), it is of course generally desirable that an implementation maintains its integrity and confidentiality against an adversary who can also access a decryption device. Unfortunately, it was concluded in [BKP+16] that the Digest, Tag and Encrypt (DTE) instance of leakage-resilient authenticated encryption does not maintain CIML in this case.

Our contributions in this direction are twofold.

First, we present various attacks against natural variations of the DTE construction and show that none of these variations provide CIML security with decryption leakages, which we formalize as CIML2. In particular, our attacks suggest that the addition of more leak-free (possibly tweakable) PRFs in DTE does not help. By contrast, we show that by exploiting a leak-free strong PRP (rather than PRF), we can reach CIML2 with a very minor modification of DTE, that we denote as DTE2.

Second, we observe that CIML2 security does not prevent Differential Power Analysis (DPA) attacks targeting the decrypted ciphertext. While such a context may look strange theoretically (i.e., why would an adversary bother about leakages if he has access to a decryption device?), we argue that such attacks are particularly relevant in the case of bitstream decryption [MBKP11] or secure bootloading [OC15]. That is, while the adversary may eventually have access to a running device in this case, it remains that concretely, designing a side-channel disassembler in order to reverse engineer a piece of software or hardware code (as in [EPW10]) is significantly more challenging, and possibly impossible for some parts of the code such as memory accesses, than performing a DPA against the message (similar to [UWM17]). Hence, minimizing the attack surface appears as an interesting goal in this case as well. We formalize this requirement with the notion of Eavesdropper Security with Differential Leakage (EavDL), which captures the fact that the security of multiple decryptions with leakage can be reduced to the one of a single decryption. We then propose an authenticated encryption scheme denoted as EDT (for Encrypt, Digest and Tag), using only two calls to a leak-free block, that is in the same time CIML2- and EavDL-secure. Interestingly, EDT can be viewed as an instance of "Encrypt then MAC" scheme [BN00], which bridges the gap with the proposals of leakage-resilient authenticated encryption schemes by Dobraunig et al. and Barwell et al.

We finally conclude the paper by highlighting that contrary to DTE2, EDT does not achieve misuse-resistance (without leakage) which allow us to put forward the pros and cons of both schemes and state open problems regarding the combination of CIML2 and EavDL security with misuse-resistance without leakage, which seems to require an additional (third) call to the leak-free strong PRP.

Other related works In a distinct line of work, the problem of leakages resulting from decryption failures is investigated [BDPS13, ABL+14, HKR15]. The motivation of these works is that, when decryption fails (as a result of an incorrect ciphertext), the decryption software typically reveals more information than just this failure: for example, it often happens that different error codes are sent depending on the step at which decryption fails. In this setting, leakage naturally happens when decryption fails only, that is, correct decryption operations do not leak anything, since there is just no error message. However, in the context of side-channel attacks, this restriction becomes meaningless: decryption takes time, consumes power and produces electromagnetic radiations whether if succeeds or fails. (And we may even expect that an implementation will leak more in case of a successful decryption, since this is likely to be the case during which the largest amount of computation takes place.)

This distinction also puts the security notions that we propose in this work out of the unifying definition framework of Barwell et al. [BPS15], called Subtle Authenticated Encryption (SAE):

1. Leaking in case of successful decryption can have a disastrous effect for some leakage functions. For instance, if the decryption leakage function $\mathsf{L}_d(c;k)$ happens to be defined as "if the decryption of c with key k succeeds, then output k, else output $\perp$", then any scheme proven secure in the sense of any of the 24 security notions of the SAE framework will just become insecure as soon as leakage happens during a successful decryption.
2. We consider leakages happening during both encryption and decryption, while SAE focuses on decryption leakages (because encryption is not expected to fail). Again, depending on the kind of leakage that happens during encryption, this may have a disastrous effect.
3. In applications of the SAE framework [BPS15, AFL$^+$16], the leakage function is typically instantiated, as a worst case, as the (incorrect) plaintext produced by the decryption function, hence excluding leakages about the internal state and keys. While this makes sense in the context of software leakage, we will go further in our context of side-channel attacks and allow leakages about the device internal state, including keys.

In terms of our constructions, leakages about the internal state is indeed a core concern. As such, we make an intensive use of rekeying techniques, and also avoid to include key material in computation when unnecessary, all in order to minimize the side-channel observation of secret data. This concern is absent of the constructions of authenticated encryption schemes in the line of work focusing on decryption errors and, as such, these schemes would be insecure in the security model that we consider here, unless unreasonably restricted leakage functions are considered. We leave it as an open question to investigate whether these schemes could be adapted to satisfy our security notions, pending modifications that would use rekeying techniques. Apart from resilience to side-channel attacks, similar rekeying techniques have recently been shown to improve security bounds of block cipher modes of operation [GL17]: this provides an extra motivation.

2 Background

2.1 Definitions

We say that a probabilistic algorithm is (q,t)-bounded if it can make at most q queries to the oracles he is granted access to, and can perform computation bounded by running time t. We then define a collision-resistant hash function as follows.

Definition 1. A $(0,t,\epsilon_{\mathsf{cr}})$-*collision resistant hash function* $\mathsf{H}:\mathcal{S}\times\mathcal{M}\rightarrow\mathcal{B}$ is a function that is such that, for every $(0,t)$-bounded adversary $\mathcal{A}$, the probability that $\mathcal{A}(s)$ outputs a pair of distinct messages $(m_0,m_1)\in\mathcal{M}^2$ such that $\mathsf{H}^s(m_0)=\mathsf{H}^s(m_1)$ is bounded by $\varepsilon_{\mathsf{cr}}$, where $s\leftarrow\mathcal{S}$ is selected uniformly at random.

We next need the following definition of range-oriented preimage resistance. The usual notion of preimage resistance samples a random $m_0\leftarrow\mathcal{M}$ over the domain of

H^s and requires finding a preimage of $z = \mathsf{H}^s(m_0)$. The definition of $(1, t, \epsilon_{\mathsf{pr}})$-range-oriented preimage resistance [AS11] uniformly samples the target space q times instead: $(z_1, \ldots, z_q) \leftarrow \mathcal{B}^q$, and requires finding a preimage for any of q elements of $\mathcal{B}$. Note that q might be seen as the maximal number of adaptive target queries made by $\mathcal{A}$.

Definition 2. A $(q, t, \epsilon_{\mathsf{pr}})$-*range-oriented preimage resistant hash function* $\mathsf{H} : \mathcal{S} \times \mathcal{M} \to \mathcal{B}$ is a function such that, for every (q, t)-bounded adversary $\mathcal{A}$, there is a probability less than $\varepsilon_{\mathsf{pr}}$ that $\mathcal{A}(s, z)$ outputs a message $m \in \mathcal{M}$ such that $\mathsf{H}^s(m) = z_i$, where z_i is an element of $z = (z_1, \ldots, z_q)$, $z_i = z_j$ implies $i = j$, and each z_i and s are selected uniformly at random in $\mathcal{B}$ and $\mathcal{S}$ respectively.

In the following, we assume that the key s is not private, and refer to the hash function simply as H for simplicity, the key s being implicit.

We also need the following definitions of pseudorandom function/permutation.

Definition 3. A function $\mathsf{F} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a $(q, t, \epsilon_{\mathsf{F}})$-*pseudorandom function (PRF)* if for all (q, t)-bounded adversaries $\mathcal{A}$, the advantage

$$\left| \Pr\left[\mathcal{A}^{\mathsf{F}_k(.)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{f}(.)} \Rightarrow 1\right] \right|$$

is upper-bounded by ε_{F}, where k and f are chosen uniformly at random from their domains, namely $\mathcal{K}$ and the set of functions from $\mathcal{M}$ to $\mathcal{T}$.
In a similar way, F is $(q, t, \epsilon_{\mathsf{F}})$-*pseudorandom permutation (PRP)* if F_k is a permutation for all k and if the above advantage is ϵ_{F}-bounded when f is selected uniformly at random among the permutations on $\mathcal{M} = \mathcal{T}$.

Definition 4. A function $\mathsf{F} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a $(q, t, \epsilon_{\mathsf{F}})$- *strong pseudorandom permutation (SPRP)* if for all (q, t)-bounded adversaries $\mathcal{A}$ provided with oracle access to the function and its inverse, the advantage

$$\left| \Pr\left[\mathcal{A}^{\mathsf{F}_k(.), \mathsf{F}_k^{-1}(.)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{f}(.), \mathsf{f}^{-1}(.)} \Rightarrow 1\right] \right|$$

is upper-bounded by ε_{F}, where k and f are chosen uniformly at random from their domains, namely $\mathcal{K}$ and the set of permutations on $\mathcal{M} = \mathcal{T}$.

The oracle implemented either with F_k or with f will be called *image oracle*, the other one the *inverse oracle*. The tweakable variants of these definitions are given next:

Definition 5. A function $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \mathcal{M} \to \mathcal{T}$ is a $(q, t, \epsilon_{\mathsf{F}})$-*tweakable pseudorandom function (TPRF)* if for all (q, t)-bounded adversaries $\mathcal{A}$ provided with oracle access to the function, the advantage

$$\left| \Pr\left[\mathcal{A}^{\mathsf{F}_k^{(.)}(.)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{f}^{(.)}(.)} \Rightarrow 1\right] \right|$$

is upper-bounded by ε_{F}, where k and f are chosen uniformly at random from their domains, namely $\mathcal{K}$ and the set of functions from $\mathcal{TW} \times \mathcal{M}$ to $\mathcal{T}$.
Again, F is a $(q, t, \epsilon_{\mathsf{F}})$-*tweakable pseudorandom permutation (TPRP)* if F_k^{tw} is a permutation for all tw and k and if f^{tw} is a random permutation on $\mathcal{M} = \mathcal{T}$ chosen independently for each value of tw.

Definition 6. A function $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \mathcal{M} \to \mathcal{M}$ is a $(q, t, \epsilon_{\mathsf{F}})$-*strong tweakable pseudorandom permutation* (STPRP) if for all keys k and for all tweaks TW $\mathsf{F}_k^{TW} : \mathcal{M} \to \mathcal{M}$ is a permutation and if for all (q, t)-bounded adversaries $\mathcal{A}$ provided with oracle access to the function and its inverse, the advantage

$$\left| \Pr\left[\mathcal{A}^{\mathsf{F}_k^{(.)}(.), \mathsf{F}_k^{-1,(.)}(.)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{f}^{(.)}(.), \mathsf{f}^{-1,(.)}(.)} \Rightarrow 1\right] \right|$$

is upper-bounded by ε_{F}, where k is chosen uniformly at random in $\mathcal{K}$ and f^{tw} is an independent uniformly random permuation on $\mathcal{M}$ for each value of tw.

We will focus on authenticated encryption with the following formalism.

Definition 7. An *authenticated encryption scheme* is a tuple $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ s.t.:

- $\mathsf{Enc} : \mathcal{K} \times \mathcal{R} \times \mathcal{M} \to \mathcal{C}$ maps a key selected from $\mathcal{K}$, randomness selected from $\mathcal{R}$ and a message from $\mathcal{M}$ to a ciphertext in $\mathcal{C}$. We write $C \leftarrow \mathsf{Enc}_k(r, m) := \mathsf{Enc}(k, r, m)$.
- $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$ maps a key and a ciphertext to a message that is the decryption of that ciphertext, or to a special symbol $\perp$ if decryption fails.

Moreover, AE satisfies the *correctness* property: for any key k selected at random from $\mathcal{K}$, for any randomness $r \in \mathcal{R}$, and for any message $m \in \mathcal{M}$, we have $\mathsf{Dec}_k(\mathsf{Enc}_k(r, m)) = m$.

We finally use the security definition of misuse-resistant authenticated encryption of [BKP+16], which is directly inspired from Rogaway and Shrimpton [RS06] but tweaked for schemes that do not have an explicit IV, as will be the case for all our schemes (see further motivation in [BKP+16]):

Definition 8. An authenticated encryption scheme $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ offers (q, t, ε) *strong misuse-resistance* if, for every (q, t)-bounded adversary $\mathcal{A}$, the advantage

$$\mathbf{Adv}^{\mathsf{mr}}_{\mathsf{AE},\mathcal{A}} := \left| \Pr\left[\mathcal{A}^{\mathsf{Enc}_k(\cdot,\cdot),\mathsf{Dec}_k(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot),\perp(\cdot)} \Rightarrow 1\right]\right|$$

is upper-bounded by ε, where k is selected uniformly at random from $\mathcal{K}$, $\$(r, m)$ outputs c selected as a random bit string of length $\mathsf{Enc}_k(r, m)$ and the oracle $\perp(c)$ outputs $\perp$ except if c was output by the $\$(\cdot, \cdot)$ oracle earlier, in which case it returns the associated m.

2.2 DTE: Digest, Tag and Encrypt

We now recall the DTE-scheme introduced in [BKP+16], which is the starting point of our discussions. We suppose that $\mathcal{K} = \mathcal{T} = \mathcal{R} = \{0, 1\}^n$ and $\mathcal{M} = (\{0, 1\}^n)^*$, and we use n as a security parameter. It is based on an hash function H and on two block-ciphers F and F^*, both treated as PRF's, but with the distinction that F is assumed to be cheap and efficiently implemented but leaking, while F^* is assumed to be an expensive and leak-free component.[1]

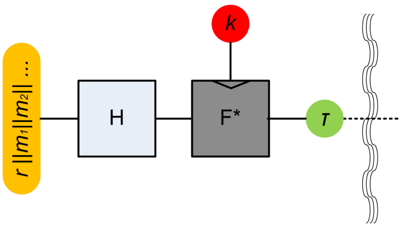

Figure 1: DTE leakage-resilient authenticated encryption (part I).

Part I, illustrated in Figure 1, is the authentication part of DTE and consists in generating a tag τ. Given the long term key k and an ℓ-block message $m = (m_1, \ldots, m_\ell)$, it picks $r \leftarrow \{0, 1\}^n$ and we will write $\mathsf{Tag}_k(r, m) = \tau$.

Part II is the encryption part of DTE which encrypts (r, m) based on an ephemeral key k_0 computed as $\mathsf{F}^*_k(\tau)$. From two fixed and distinct n-bit strings p_A and p_B, this part consists in generating $\ell + 1$ pseudorandom blocks $(y_0, y_1, \ldots, y_\ell)$ and to XOR them to $(r, m_1, \ldots, m_\ell)$. We will write $\mathsf{PSVEnc}_k(\tau, (r, m)) = c := (c_0, c_1, \ldots, c_\ell)$ to refer to a scheme due to [PSV15]. The full specification of DTE is provided in Table 1.

[1] We recall that, as discussed in [PSV15, BKP+16], this leak-free component is a just convenient model to separate the parts of the authenticated encryption that have to be well protected thanks to countermeasures, and the ones for which the mode enables some leakage-resilience by design.

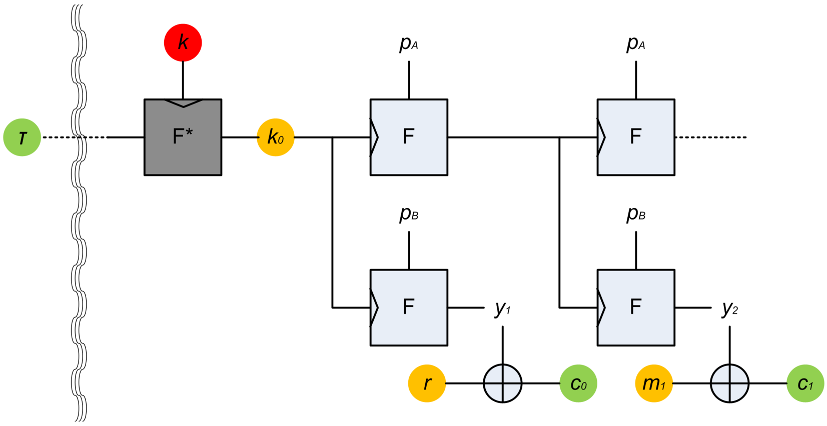

Figure 2: DTE leakage-resilient authenticated encryption (part II).

Table 1: DTE leakage-resilient authenticated encryption.

```
DTE
Enc_k(r, m): parse $m = (m_1, m_2, \ldots, m_\ell)$
  • $\tau \leftarrow \mathsf{Tag}_k(r, m)$:
      – $h \leftarrow \mathsf{H}(r||m)$                                  // digest
      – $\tau \leftarrow \mathsf{F}^*_k(h)$                              // tag
  • $c \leftarrow \mathsf{PSVEnc}_k(\tau, (r, m))$:                      // encrypt
      – $k_0 \leftarrow \mathsf{F}^*_k(\tau)$
      – $c_0 \leftarrow \mathsf{F}_{k_0}(p_B) \oplus r$
      – $k_i \leftarrow \mathsf{F}_{k_{i-1}}(p_A)$, $c_i \leftarrow \mathsf{F}_{k_i}(p_B) \oplus m_i$   $\forall i = 1, \ldots, \ell$
      – set $c = (c_0, c_1, \ldots, c_\ell)$
  • return $C \leftarrow (\tau, c)$

Dec_k(C): parse $C = (\tau, c)$
  • $(r, m) \leftarrow \mathsf{PSVEnc}_k(\tau, c)$
  • $\tau^c \leftarrow \mathsf{Tag}_k(r, m)$
  • if $\tau = \tau^c$ return $m$, return $\perp$.
```

3 DTE with decryption leakages

Ciphertext integrity with coin misuse and leakage (on encryption), CIML for short, is a strong flavor of ciphertext integrity due to [BKP+16]. Informally, this security notion requires that computing fresh valid ciphertexts must remain infeasible even if the adversary can get valid ciphertexts for any chosen randomness/message pair from the encryption algorithm, and even if each encryption run leaks.

DTE achieves CIML security in the so-called *unbounded leakage model* [BKP+16], in which the internal state of the device is leaked in full in each encryption run, except for the state of the leak-free component. However, it was pointed that, if the adversary is also able to get the ephemeral state computed during the decryption of any chosen ciphertext, this integrity guarantee completely falls down (see discussion below).

In this section, we first define CIML2, an extension of the CIML security notion that captures leakages during decryption. We then recall the attack against DTE with decryption leakages, together with attacks against natural attempts to mitigate decryption leakages with more leak-free (possibly tweakable) components.

Table 2: $\mathsf{CIML2}$ experiment.

$\mathsf{CIML2}_{\mathsf{AE},\mathsf{L}_e,\mathsf{L}_d,\mathcal{A}}(1^n)$ experiment	
Initialization:	Oracle $\mathsf{EncL}_k(r,m)$:
$k \leftarrow \mathcal{K}$ s.t. $\lvert k \rvert = n$	$C = \mathsf{Enc}_k(r,m)$
$\mathcal{S} \leftarrow \emptyset$	$\mathcal{S} \leftarrow \mathcal{S} \cup \{C\}$
Finalization:	return $(C, \mathsf{L}_e(r,m;k))$
$C \leftarrow \mathcal{A}^{\mathsf{EncL}_k(\cdot,\cdot),\mathsf{DecL}_k(\cdot)}$	
If $C \in \mathcal{S}$ or $\bot = \mathsf{Dec}_k(C)$, return 0	Oracle $\mathsf{DecL}_k(C)$:
Return 1	return $(\mathsf{Dec}_k(C), \mathsf{L}_d(C;k))$

3.1 CIML2: extending CIML with decryption leakage

We model the ability of an adversary to additionally get leakage on decryption by extending the CIML experiment into the $\mathsf{CIML2}$ experiment, by granting the adversary with an oracle access to DecL_k. The specifications of this experiment are detailed in Table 2.

Definition 9. An authenticated encryption $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ with encryption leakage function L_e and decryption leakage function L_d provides (q,t,ϵ)-*ciphertext integrity with coin misuse and leakage on encryption and decryption* (next denoted as $\mathsf{CIML2}$) if for all (q,t)-bounded adversaries $\mathcal{A}$, we have:

$$\Pr\left[\mathsf{CIML2}_{\mathsf{AE},\mathsf{L}_e,\mathsf{L}_d,\mathcal{A}} \Rightarrow 1\right] \leq \epsilon.$$

As usual q is an upper bound on the total number of queries made to the oracles.

Showing that a scheme satisfies this definition of course requires to model the leakages of an implementation, which is in general a hard problem. Interestingly, it has been observed in [BKP+16] that for integrity properties such as CIML, it is actually possible to reason based on a very permissive model, called unbounded leakage model.

Definition 10. An implementation of a scheme with leakage function L is said to offer a security property in the *unbounded leakage model* if that property is satisfied even if L yields all the internal states produced during each execution of the scheme, including all keys and random coins, at the exclusion of the internal state of leak-free components if there are any.

In the case of DTE, this means that, on each query, everything is leaked except for the long-term key k used by the leak-free component. Or, in an equivalent way, and in the context of CIML security, we can just assume that k_0 is leaked. Furthermore, it was shown that DTE is CIML-secure.

Theorem 1 ([BKP+16])**.** *Let* $\mathsf{H} : \{0,1\}^n \times \{0,1\}^\star \to \{0,1\}^n$ *be a* $(0,t',\varepsilon_{cr})$*-collision resistant and* $(1,t',\varepsilon_{pr})$*-range-oriented preimage resistant hash function. Let* $\mathsf{F}^* : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ *be a* $(2q+2,t',\varepsilon_{\mathsf{F}^*})$*-pseudorandom function. Then* DTE *provides* (q,t,ε)*-ciphertext integrity with coin misuse and unbounded leakage on encryption as long as* $t \leq t' - (q+1)(t_{\mathsf{H}} + (2l+1)t_{\mathsf{F}})$ *where* t_{H} *and* t_{F} *are the time needed to evaluate* H *and* F*, and we have*

$$\varepsilon \leq \varepsilon_{\mathsf{F}^\star} + \varepsilon_{cr} + 2q \cdot \varepsilon_{pr} + (q+1) \cdot 2^{-n}.$$

It can be observed that this result does not make any security assumption regarding F (but only regarding F^*). Indeed, the unbounded leakages associated to F make its expected pseudorandomness essentially useless (this theorem would hold even if $\mathsf{F}(m) = 0^n$ for every m), and it is actually sufficient for the function $\mathsf{PSV\text{-}ENC}_k(\tau, .)$ to be a permutation (for any k, τ), which is guaranteed, by the XOR operations. The pseudorandomness of F is of course most useful when turning to the confidentiality guarantees of DTE.

In the rest of this work, we will keep instatiating the encryption component of our schemes with $\mathsf{PSV\text{-}ENC}$ or variants of it, despite the fact that various other (and simpler) choices are possible if $\mathsf{CIML2}$ security is the only concern. This makes our constructions concrete, and also makes it possible to borrow previously proven results regarding confidentiality properties.

3.2 A first attack against DTE with decryption leakages

In the unbounded leakage model, we can summarize the outputs of the leakage functions for encryption and decryption as $\mathsf{L}_e(r, m; k) := k_0$ and $\mathsf{L}_d(C; k) := (k_0, \tau^c)$ since, from these values, the adversary is able to recompute all intermediate values and all keys used by the algorithms, apart from the key of the leak-free components.

As stated in [BKP+16], DTE is not secure in the presence of decryption leakages. The attack exploits the re-keying process in decryption in order to get a correct tag τ^c for any chosen couples (r, m), thanks to L_d. It proceeds as follows:

- Pick some randomness r and a message $m = m_1, ..., m_l$ and compute $h = \mathsf{H}(r\|m)$.
- Ask for the decryption of ciphertext $C^1 = (\tau^1, c^1)$ with $\tau^1 = h$ and any $c^1 = (c_0^1, ..., c_l^1)$. Recover the first ephemeral key k_0^1 via leakage.
- Ask for the decryption of ciphertext $C^2 = (\tau^2, c^2)$ with $\tau^2 = k_0^1$ and any $c^2 = (c_0^2, ..., c_l^2)$. Recover the first ephemeral key k_0^2 via leakage.
- From k_0^2, compute the ciphertext C in this way: $k_0 := k_0^2,\ c_0 = \mathsf{F}_{k_0}(p_B) \oplus r,\ k_1 = \mathsf{F}_{k_0}(p_A),\ c_1 = \mathsf{F}_{k_1}(p_B) \oplus m_1,\ ...,\ c_l = \mathsf{F}_{k_l}(p_B) \oplus m_l$. Compute its tag $\tau = k_0^1$. The ciphertext $C = (\tau, c)$, with $c = (c_0, ..., c_l)$ is valid.

The fundamental difference between encryption leakages and decryption leakages lies in the way the adversary can influence the generation of the ephemeral key k_0. In the former case the adversary can only obtain the k_0's associated to unpredictable random tags while in the latter case the adversary can obtain the k_0's from arbitrarily selected tags, which therefore allows forgeries of valid ciphertexts.

3.3 Tweaking DTE and another attack

A simple and efficient way to prevent the previous attack on DTE is to use a tweakable leak-free PRF leading to DTE' (see details in Table 3). The only difference with respect to the DTE encryption is in the computation of $\tau = \mathsf{F}_k^{*,0}(h)$ and $k_0 = \mathsf{F}_k^{*,1}(\tau)$ which tweaks F_k^* with one bit. This has the effect of viewing $\mathsf{F}_k^{*,0}$ and $\mathsf{F}_k^{*,1}$ as independent pseudorandom functions. As a result, the decryption leakages let the adversary controlling $\mathsf{F}_k^{*,1}$ but not $\mathsf{F}_k^{*,0}$ which computes the tag. We show next that this tweak is not enough and exhibit a more powerful forgery attack:

- Ask for the decryption of $C^1 = (\tau^1, c_0^1, ..., c_l^1)$ for random τ^1, $c^1 = (c_0^1, ..., c_l^1)$. During the decryption a randomness r^1 and a message $m^1 = m_1^1, ...m_l^1$ is computed. Then $h^1 = \mathsf{H}(r^1\|m^1)$ is computed and it is verified if $\tau^{1,c} = \mathsf{F}_k^{*,0}(h^1) \overset{?}{=} \tau^1$. Recover via leakage r^1, m^1 and $\tau^{1,c}$.
- Ask for the decryption of $(\tau^2, c_0^2, ..., c_l^2)$ for $\tau^2 = \tau^{1,c}$ and random $c_0^2, ..., c_l^2$ in order to get $k_0^2 = \mathsf{F}_k^{*,1}(\tau^2)$ via leakage.
- From τ^2 and k_0^2 it is possible to compute the valid encryption of (r^1, m^1) where $k_0 = k_0^2$, $c_0 := \mathsf{F}_{k_0}(p_B) \oplus r^1$, $k_1 = \mathsf{F}_{k_0}(p_A)$, $c_1 = \mathsf{F}_{k_1}(p_B) \oplus m_1^1, ..., c_l = \mathsf{F}_{k_l}(p_B) \oplus m_l^1)$. This is valid since $\tau^2 = \mathsf{F}_k^{*,0}(\mathsf{H}(r^1\|m^1))$ and $k_0 = \mathsf{F}_k^{*,1}(\tau^2)$.

As a result, DTE' is not $\mathsf{CIML2}$. Note that we will keep the tweaked versions of Tag_k and PSVEnc_k specified in Table 3, denoted Tag_k^{tw} and PSVEnc_k^{tw}, for later use, where the tweak tw refers to the leak-free component $\mathsf{F}_k^{*,tw}$ running inside the algorithm.

Table 3: DTE' leakage-resilient authenticated encryption.

```
$\mathsf{DTE}'$

$\mathsf{Enc}_k(r, m)$: parse $m = (m_1, \ldots, m_\ell)$
  • $\boxed{\tau \leftarrow \mathsf{Tag}^0_k(r, m):}$
      – $h \leftarrow \mathsf{H}(r||m)$                                   // digest
      – $\boxed{\tau \leftarrow \mathsf{F}^{*,0}_k(h)}$                                   // tag
  • $\boxed{c \leftarrow \mathsf{PSVEnc}^1_k(\tau, (r, m)):}$                    // encrypt
      – $\boxed{k_0 \leftarrow \mathsf{F}^{*,1}_k(\tau)}$
      – $c_0 \leftarrow \mathsf{F}_{k_0}(p_B) \oplus r$
      – $k_i \leftarrow \mathsf{F}_{k_{i-1}}(p_A)$, $c_i \leftarrow \mathsf{F}_{k_i}(p_B) \oplus m_i$   $\forall i = 1, \ldots, \ell$
      – set $c = (c_0, c_1, \ldots, c_\ell)$
  • return $C \leftarrow (\tau, c)$

$\mathsf{Dec}_k(C)$: parse $C = (\tau, c)$
  • $\boxed{(r, m) \leftarrow \mathsf{PSVEnc}^1_k(\tau, c)}$
  • $\boxed{\tau^c = \mathsf{Tag}^0_k(r, m)}$                                // check tag
  • if $\tau = \tau^c$ return $m$, return $\perp$.
```

3.4 More leak-free components do not help (for DTE)

Since tweaking DTE does not allow avoiding forgery attacks, another natural option to consider is the use of more leak-free components. Before arguing that such an approach is also unlikely to be effective, we first remind that the goal of DTE is to leverage the good security properties offered by leakage-resilient stream ciphers. This implies that for the encryption part of Figure 2, we anyway do not want to use leak-free components on all the blocks (or the interest of the whole construction vanishes). This leaves us with two main options which we discuss next: the addition of leak-free calls in the authentication part of Figure 1 and after the encryption part of Figure 2.

More leak-free components in authentication part of DTE. We argue that such a variation does not improve security by visual inspection. That is, say we add more leak-free components after each execution of an F^* in Figures 1 and 2. Independently of whether this change is operated for the first or the second execution of F^*, we obtain a composition of leak-free PRFs which can be viewed as a single (less efficient) PRF from the adversary's point of view. Indeed, even if we modify $\mathsf{Tag}^{tw}_k(h)$ of DTE' to return $\tau' = \mathsf{F}^*_i \circ \ldots \mathsf{F}^*_2 \circ \mathsf{F}^{*,0}_k(h)$ and $\mathsf{PSVEnc}^{tw}_k(\tau')$ to compute the ephemeral key $k'_0 = \mathsf{F}^*_l \circ \ldots \mathsf{F}^*_{j+1} \circ \mathsf{F}^{*,1}_k \circ \mathsf{F}^*_j \circ \ldots \mathsf{F}^*_{i+1}(\tau')$, for any non negative integers i, j, l, we simply have $\tau' = \mathsf{F}^*_0(h)$ and $k'_0 = \mathsf{F}^*_1(\tau')$ for some F^*_0 and F^*_1. This does not prevent the aforementioned forgery attacks.

More leak-free components after the encryption part of DTE. Starting from any $\mathsf{AE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we define $\mathsf{AE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ such that (1) Gen' returns the output of Gen and a (one-more) leak-free PRF G^*, (2) Enc' returns the ciphertext C output by Enc and $\tau' = \mathsf{G}^*(C)$, (3) Dec' first checks whether $\tau' = \mathsf{G}^*(C)$ and halts if it does not hold, otherwise it outputs $\mathsf{Dec}(C)$.[2]

Assuming there is a (q, t)-bounded adversary $\mathcal{A}$ against AE in $\mathsf{CIML2}$, we show how to build a $(2q + 1, t)$-bounded adversary $\mathcal{A}'$ against AE' in $\mathsf{CIML2}$ with the same advantage. The reduction is straightforward.

[2] We omit the use of obvious inputs such as keys, messages and so for readability. Note also that one could additionally hash the ciphertext in Step (2) which would not affect our argument.

Table 4: DTE2 leakage-resilient authenticated encryption.

```
DTE2
$\mathsf{Enc}_k(r, m)$:
    • $\tau \leftarrow \mathsf{Tag}_k^0(r, m)$
    • $c \leftarrow \mathsf{PSVEnc}_k^1(\tau, (r, m))$
    • return $C \leftarrow (\tau, c)$
$\mathsf{Dec}_k(C)$: parse $C = (\tau, c)$
    • $(r, m) \leftarrow \mathsf{PSVEnc}_k^1(\tau, c)$
    • $h \leftarrow \mathsf{H}(r||m)$
    • $\boxed{h^c = (\mathsf{F}_k^{*,0})^{-1}(\tau)}$                // check digest
    • $\boxed{\text{if } h = h^c}$ return $m$, return $\perp$
```

- $\mathcal{A}$ queries an encryption of m with randomness r to Enc: $\mathcal{A}'$ queries an encryption on (r, m) to Enc' and gets back (C, τ') along with $\mathsf{L}'_e(C, \tau')$. Since τ' is given in the ciphertext, we simply have $\mathsf{L}'_e(C, \tau') = \mathsf{L}_e(C)$. Then, $\mathcal{A}'$ hands $\mathcal{A}$ with C and $\mathsf{L}_e(C)$.
- $\mathcal{A}$ queries a decryption of C to Dec: $\mathcal{A}'$ conducts two steps,
  1. $\mathcal{A}'$ picks a dummy tag τ'_0 and queries a decryption of (C, τ'_0) to Dec'. The check $\tau'_0 = \tau'$, where $\tau' = G^*(C)$, may fail but leaks the right tag τ' to $\mathcal{A}'$;
  2. $\mathcal{A}'$ queries a decryption of (C, τ') to Dec'. Since the check passes, now $\mathcal{A}'$ learns $\mathsf{Dec}(C)$ and $\mathsf{L}_d(C)$ and forwards them to $\mathcal{A}$.

Eventually $\mathcal{A}$ outputs a forgery C^* with some probability. Then, $\mathcal{A}'$ runs the subroutine in item 1 on C^* to get the right tag τ'^* and finally outputs (C^*, τ'^*), which is a forgery with the same probability.

Despite heuristic, the latter observations suggest that adding a leak-free-PRF anywhere inside an CIML2-insecure authenticated encryption scheme will not directly help preventing forgeries. We leave the proof of a more formal statement (e.g., based on an induction of the leak-free calls) as an interesting scope for further research, and for now we use these observations to motivate the positive result in the next section.

4 A first CIML2-secure construction

We now turn DTE into a CIML2-secure authenticated encryption scheme that we denote as DTE2. This modified scheme simply changes the way the DTE$'$ decryption algorithm works in order to prevent the decryption leakages to enable valid forgeries.

4.1 DTE2 specification

In a nutshell, the DTE2 scheme starts from DTE$'$ but further assumes the tweakable leak-free F^* to be a permutation. This allows us to invert the way the authentication check is performed during decryption. Once the digest h is recomputed, instead of verifying if the F^* evaluation matches the tag τ, we now check if the F^* preimage of τ matches h. The description of DTE2 is available in Table 4.

DTE2 uses tweakable PRPs, with two distinct tweaks. Moving back to standard PRP's (or identical tweaks) would expose DTE2 to the same attacks as before.

4.2 DTE2 security proof

The idea behind the design of DTE2 is to never leak the right answer $\tau^c = \mathsf{F}_k^{*,0}(h)$ during the verification step of an invalid ciphertext. Instead, the decryption leakage only gives what would have to be the right question $h^c = (\mathsf{F}_k^{*,0})^{-1}(\tau)$. The next theorem states that this slight modification is enough to reach $\mathsf{CIML2}$ in the unbounded leakage model as long as h^c looks random.

Theorem 2. *Let* $\mathsf{H} : \{0,1\}^n \times \{0,1\}^\star \to \{0,1\}^n$ *be a* $(0, t', \varepsilon_{cr})$*-collision resistant and* $(q+1, t', \varepsilon_{pr})$*-range-oriented preimage resistant hash function. Let* $\mathsf{F}^* : \{0,1\}^n \times \{0,1\} \times \{0,1\}^n \to \{0,1\}^n$ *be a* $(2q+2, t', \varepsilon_{\mathsf{F}^*})$*-strong tweakable pseudorandom permutation. Then* $\mathsf{DTE2}$ *provides* (q, t, ε)*-ciphertext integrity with coin misuse and unbounded leakage on encryption and decryption as long as* $t \leq t' - (q+1)(t_\mathsf{H} + (2\ell+1)t_\mathsf{F})$*, where* t_H *and* t_F *are the time needed to evaluate* H *and* F*, and we have*

$$\varepsilon = \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr}.$$

Just as Theorem 1, and for the same reasons, Theorem 2 does not make any assumption about F. The pseudorandomness of F is however central to the proof that $\mathsf{DTE2}$ is also a misuse-resistance authenticated encryption scheme in the sense of Definition 8. Indeed, DTE already satisfies this security notion and tweaking as well as inverting F^* as no effect on the original proof [BKP+16].

The proof of Theorem 2 follows the flow of the proof of Theorem 1 given in [BKP+16]. In the unbounded model the leakage functions are $\mathsf{L}_e(r, m; k) := k_0$ and $\mathsf{L}_d(C; k) = (k_0, h^c)$ since all the intermediate values can be computed from these outputs.

Proof. Let $\mathcal{A}$ be a (q, t)-bounded adversary against $\mathsf{DTE2}$. We have to show that

$$\Pr\left[\mathsf{CIML2}_{\mathsf{DTE2},\mathsf{L}_e,\mathsf{L}_d,\mathcal{A}} \Rightarrow 1\right] \leq \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr},$$

as long as $q_e + q_d \leq q$, where q_e denotes the number of encryption queries made to EncL and q_d denotes the number of decryption queries made to DecL. To do so, we will use $\mathcal{A}$ in a sequence of games beginning with the real game $\mathsf{CIML2}_{\mathsf{DTE2}}$ and ending with a game where all the ciphertexts are invalid except those returned by EncL. Each transition between the games will be reduced to an efficient algorithm against either the STPRP of $\mathsf{F}^{*,\cdot}$ or the collision resistance of H or to the range oriented preimage resistance of H. For the sake of simplicity $C^{q_d+1} = (\tau^{q_d+1}, c^{q_d+1})$ denotes the $\mathcal{A}$'s output in each game and we consider it as a last decryption query which is not among the answers to EncL.

Without loss of generality we also assume that any returned ciphertext by EncL_k is never sent to DecL_k. Indeed, the reduction should only stores the query (r, m) and the answer $(C, k_0) = (\mathsf{Enc}_k(r, m), \mathsf{L}_e(r, m; k))$ to emulate the answer $(m, (k_0, h^c)) = (\mathsf{Dec}_k(C), \mathsf{L}_d(C; k))$ to $\mathsf{DecL}_k(C)$ since $h^c = h$ always holds for such C in all the following games.

In the i-th $\mathsf{CIML2}$ game, $\mathsf{Game}\text{-}i$, the adversary is face with a modified version of $\mathsf{CIML2}_{\mathsf{DTE2}}$ which has been subject to i modification(s). Each modification has the effect of changing the way the corresponding oracles respond to encryption and decryption queries. Let E_i be the event whereby $\mathcal{A}$ eventually outputs a valid forgery, namely $\mathsf{Game}\text{-}i_\mathcal{A} \Rightarrow 1$.

$\mathsf{Game}\text{-}0$: this is the real $\mathsf{CIML2}$ game where $\mathcal{A}$ attacks $\mathsf{DTE2}$. More precisely, once the n-bit key k is fixed, the answer (C, k_0) to an encryption query (r, m) is performed as $\mathsf{EncL}_k(r, m)$: (1) compute $\tau = \mathsf{Tag}_k^0(r, m)$ as $\tau = \mathsf{F}_k^{*,0}(h)$ where $h = \mathsf{H}(r\|m)$, (2) compute $c = \mathsf{PSVEnc}_k^1(\tau, (r, m))$ which first computes $k_0 = \mathsf{F}_k^{*,1}(\tau)$ and then $c = (c_0, c_1, \ldots, c_\ell)$ from the ephemeral key k_0 and (r, m) viewed as a plaintext (see Figure 2 for an illustration), (3) set $C = (\tau, c)$. The answer $(m, (k_0, h^c))$ or $(\perp, (k_0, h^c))$ to a decryption query $C = (\tau, c)$ is performed as $\mathsf{DecL}_k(C)$: (1) compute $(r, m) = \mathsf{PSVEnc}_k^1(\tau, c)$ which first computes $k_0 = \mathsf{F}_k^{*,1}(\tau)$ and then r and $m = (m_1, \ldots, m_\ell)$ with the same process as in encryption, (2) compute $h = \mathsf{H}(r\|m)$ and $h^c = (\mathsf{F}_k^{*,0})^{-1}(\tau)$, (3) return $(m, (k_0, h^c))$ if $h = h^c$ and $(\perp, (k_0, h^c))$, otherwise.

$\mathsf{Game}\text{-}1$: we replace all the occurrences of $\mathsf{F}_k^{*,tw}$ and its inverse function in $\mathsf{DTE2}$ by a truly random permutation f^{tw} and its inverse, with tweak $tw \in \{0,1\}$. The modifications of

EncL_k and DecL_k in $\mathsf{CIML2}$ are straightforward. We simply switch the leak-free strong PRP, with tweak $tw \in \{0,1\}$, with the random permutation with the same tweak tw. We apply the same switch to the inverse functions.

Transition from Game-0 to Game-1: we build a $(2q+2, t_1)$-bounded challenger $\mathcal{B}_1$ against the strong tweakable pseudorandom permutation $\mathsf{F}^{*,\cdot}$ whose advantage is $|\Pr[E_0] - \Pr[E_1]|$. The challenger $\mathcal{B}_1$ is given the key length n and access to $\mathcal{A}$ as well as to an oracle which, on any input $p \in \{0,1\}^n$, any tweak $tw \in \{0,1\}$ and an implicit sign ± 1, always computes either $(\mathsf{F}_k^{*,tw})^{\pm 1}(p)$ or $f_{tw}^{\pm 1}(p)$. The goal of $\mathcal{B}_1$ is to distinguish which evaluation is performed by the oracle. For that purpose, $\mathcal{B}_1$ must emulate the encryption and the decryption oracles of $\mathsf{CIML2}$ in front of $\mathcal{A}$. To do so, $\mathcal{B}_1$ first picks $p_A, p_B \leftarrow \{0,1\}^n$ at random which are involved in the computation of PSVEnc (see Figure 2 for instance). Then, $\mathcal{B}_1$ selects H and responds to an encryption query (r, m) by (C, k_0) performed as: (1) call its own oracle on $h = \mathsf{H}(r\|m)$ with tweak $tw = 0$ to get τ, (2) call its own oracle on τ with tweak $tw = 1$ to get k_0 and then $c = (c_0, c_1, \ldots, c_\ell)$ from the ephemeral key k_0 and (r, m) as in both encryption algorithms of the both games, (3) set $C = (\tau, c)$. To decryption query $C = (\tau, c)$, $\mathcal{B}_1$ responds with $(m, (k_0, h^c))$ performed as: (1) call its own oracle on τ with tweak $tw = 1$ to get k_0 and then to recover r and $m = (m_1, \ldots, m_\ell)$ with the same process as in encryption, (2) compute $h = \mathsf{H}(r\|m)$ and query its own oracle on the inverse function on τ with tweak $tw = 0$ to get h^c, (3) return $(m, (k_0, h^c))$ if $h = h^c$ and $(\perp, (k_0, h^c))$, otherwise. Eventually, when $\mathcal{A}$ outputs its final ciphertext C^{q_d+1}, $\mathcal{B}_1$ simply outputs 1 as its guess if C^{q_d+1} is a valid forgery and outputs 0 otherwise.
Obviously, depending on whether $\mathcal{B}_1$ is given oracle access to either $\mathsf{F}_k^{*,tw}$ or f^{tw}, $\mathcal{A}$ is playing Game-0 or Game-1. Therefore, any difference between $\Pr[E_0]$ and $\Pr[E_1]$ leads to the same difference in distinguishing the leak-free strong tweakable PRP from the random tweakable permutation, making $\mathcal{B}_1$ a $(2q+2, t_1)$-adversary against the STPRP $\mathsf{F}^{*,\cdot}$ with $t_1 = t + (q+1)(t_\mathsf{H} + (2\ell+1)t_\mathsf{F})$, since two evaluations of the oracles are needed in each encryption and each decryption emulation as well as in the validity check on C^{q_d+1}, and where t_H and t_F are the time needed to evaluate H and F. Consequently, we have $t_1 \leq t'$ by assumption. Therefore, since $\mathsf{F}^{*,\cdot}$ is a $(2q+2, t', \varepsilon_{\mathsf{F}^*})$-strong tweakable pseudorandom permutation we find $|\Pr[E_0] - \Pr[E_1]| \leq \varepsilon_{\mathsf{F}^*}$.

Game-2: this game is defined as Game-1 except that we introduce a failure event F_1, depending on some efficiently checkable properties, which force Game-2 to abort and return 0. We define F_1 as the event that the *associated digest* $h^i = \mathsf{H}(r^i\|m^i)$ related to some encryption query (r^i, m^i) is equal to the *draft digest* $h^j = \mathsf{H}(r^j\|m^j)$ involved in some decryption query $C^j = (\tau^j, c^j)$ such that $(r^i, m^i) \neq (r^j, m^j)$, and so even for $j = q_d + 1$.

Transition from Game-1 to Game-2: we have $\Pr[E_1] \leq \Pr[F_1] + \Pr[E_1 \wedge \neg F_1]$ in Game-1 and $\Pr[E_2] = \Pr[E_2 \wedge \neg F_1]$ in Game-2. Therefore, we find $|\Pr[E_1] - \Pr[E_2]| \leq \Pr[F_1]$ in Game-1, which upper-bounds the probability that the new abort rule would have occurred before.
We use the $(0, t', \epsilon_{cr})$-collision resistance of H to bound $\Pr[F_1]$ in Game-1. For that purpose, we build a $(0, t_2)$-bounded challenger $\mathcal{B}_2$ which, given H and access to $\mathcal{A}$, attempts to output a collision on H. To emulate $\mathsf{CIML2}$ during the interactions with $\mathcal{A}$, $\mathcal{B}_2$ simply picks $p_A, p_B \leftarrow \{0,1\}^n$ and the random permutations f_0, f_1 as before and responds to all the queries as in Game-1. If F_1 occurs it happens that $\mathsf{H}(r^i\|m^i) = \mathsf{H}(r^j\|m^j)$, for some ith encryption query and some jth decryption query, and $(r^i, m^i) \neq (r^j, m^j)$. At the end of the game, $\mathcal{B}_2$ ran in time $t_2 = t_1 \leq t'$ (assuming that computing the random permutations is free) and outputs the collision $(r^i\|m^i, r^j\|m^j)$. Therefore, $\Pr[F_1] \leq \epsilon_{cr}$.

Game-3: this game is identical to Game-2 except that we introduce yet another failure event F_2, depending on some efficiently checkable properties, which forces Game-3 to abort and return 0. We define F_2 as the event that in some decryption query $C^i = (\tau^i, c^i)$ the *check digest* $h^{c,i} = f_0^{-1}(\tau^i)$ is equal to the *draft digest* $h^i = \mathsf{H}(r^i\|m^i)$ as long as τ^i first appears in a decryption query in the game, and so even for $i = q_d + 1$.

Transition from $\mathsf{Game}\text{-}2$ to $\mathsf{Game}\text{-}3$: we have $\Pr[E_2] \leq \Pr[F_2] + \Pr[E_2 \wedge \neg F_2]$ in $\mathsf{Game}\text{-}2$ and $\Pr[E_3] = \Pr[E_3 \wedge \neg F_2]$ in $\mathsf{Game}\text{-}2$. Therefore, we find $|\Pr[E_2] - \Pr[E_3]| \leq \Pr[F_2]$ in $\mathsf{Game}\text{-}2$, which upper-bounds the probability that the new abort rule would have occurred before.

We use the $(q+1, t', \epsilon_{pr})$-range-oriented preimage resistance of H to bound $\Pr[F_2]$ in $\mathsf{Game}\text{-}2$. For that purpose, we build a (q_d+1, t_3)-bounded challenger $\mathcal{B}_3$ which, given H and access to $\mathcal{A}$, attempts to output a range-oriented preimage on H. Since the q_d+1 targets $z_1, \ldots, z_{q_d+1}$ requested by $\mathcal{B}_3$ are random over $\{0,1\}^n$, they are independent of $\mathcal{B}_3$'s behavior, and we assume that they are given to $\mathcal{B}_3$ at the outset of the game. To emulate $\mathsf{CIML2}$ during the interactions with $\mathcal{A}$, $\mathcal{B}_3$ simply picks $p_A, p_B \leftarrow \{0,1\}^n$ and the random permutations f_0, f_1 as before and responds to all the queries as in $\mathsf{Game}\text{-}2$ except in the following case: from a decryption query $C^j = (\tau^j, c^j)$ with a fresh tag τ^j at that time in the game, instead of computing $h^{c,j} = f_0^{-1}(\tau^j)$ as before, $\mathcal{B}_3$ redefines the input-output pair $(f_0^{-1}(\tau^j), \tau^j)$ of f_0 by (z_j, τ^j), namely $\mathcal{B}_3$ sets $h^{c,j} = z_j$. However the distribution of all the $h^{c,j}$ as well as all the other values in the game remain unchanged. This modification is purely conceptual. Now, considering that F_2 occurs, it happens that the draft digest h^i for some ith decryption query is equal to the check digest $h^{c,i}$ which is among the at most q_d+1 targets. Indeed, either τ^i was fresh or $\tau^i = \tau^j$ for some smallest index j but where τ^j was fresh at the time the corresponding jth decryption query $C^j = (\tau^j, c^j)$ was made. Then, once $\mathcal{B}_3$ figures out that $\mathsf{H}(r^i \| m^i) = h^i = h^{c,i}$ it simply outputs a desired preimage $(r^i \| m^i)$ of $z_i = h^{c,i}$. Since $\mathcal{B}_3$ still runs in time $t_3 = t_2 = t_1 \leq t'$, we have $\Pr[F_2] \leq \varepsilon_{pr}$.

$\mathsf{Game}\text{-}4$: this game is identical to $\mathsf{Game}\text{-}3$ except that we force $\mathsf{Game}\text{-}4$ to abort and return 0 if one of the decryption query on some $C^i = (\tau^i, c^i)$ turns out to be a valid ciphertext, and so even if $i = q_d + 1$. Clearly $\Pr[E_4] = 0$.

Transition from $\mathsf{Game}\text{-}3$ to $\mathsf{Game}\text{-}4$: let F_3 be the event that some ciphertext on which $\mathcal{A}$ queried decryption in $\mathsf{Game}\text{-}3$ is valid while it is now deemed invalid in $\mathsf{Game}\text{-}4$. Clearly, we have $|\Pr[E_3] - \Pr[E_4]| \leq \Pr[F_3]$. We will now argue that $\Pr[F_3] = 0$

Let $C^i = (\tau^i, c^i)$ be the valid ciphertext with the smallest index on which $\mathcal{A}$ queries decryption. We consider several cases.

- Case 1: τ^i appears in a response to some encryption query (r^j, m^j) where j is the smallest index satisfying this property.
  - Case 1.a: the jth encryption query happens before the ith decryption query.
    In that case, we know that the response $C^j = (\tau^j, c^j)$ differs from C^i and then $c^i \neq c^j$ since we assumed that no decryption query recycles such response. Moreover, we must have $k_0^i = k_0^j$ and $h^{c,i} = h^{c,j}$ in decryption because of the use of permutations. Therefore, $(r^i, m^i) \neq (r^j, m^j)$ (see Figure 2). Furthermore, we have $\mathsf{H}(r^i \| m^i) = h^{c,i} = h^{c,j} = \mathsf{H}(r^j \| m^j)$ since both ciphertexts are valid. So, in both games this situation does not happen since the introduction of the failure event F_1.
  - Case 1.b: the jth encryption query happens after the ith decryption query.
    In both games this situation does not happen since the introduction of the failure event F_2.
- Case 2: τ^i does not appear in any response to encryption query.
  In that case C^i is already not valid in both games since the introduction of the failure event F_2, which leads to a contradiction.

As a result we find that F_3 is an impossible event.

Taking all together we thus find $\Pr[E_0] \leq |\Pr[E_0] - \Pr[E_1]| + \Pr[F_1] + \Pr[F_2] + \Pr[E_4]$. Hence the announced result $\Pr[\mathsf{CIML2}_{\mathsf{DTE2},\mathsf{L}_e,\mathsf{L}_d,\mathcal{A}}] \leq \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr}$. □

□

Table 5: EavDL experiments.

$\mathsf{EavDL}_{\mathsf{AE},\mathsf{L}_d,\mathcal{A}}(1^n)$ experiment	
Initialization: $k \leftarrow \mathcal{K}$ s.t. $\lvert k\rvert = n$ $b \leftarrow \{0,1\}$ Oracle $\mathsf{LDec}_k(c)$: Return $\mathsf{L}_d(c;k)$	Finalization: $(\mathsf{st}, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathsf{LDec}_k(\cdot)}(1^n)$ $c^* \leftarrow \mathsf{Enc}_k(m_b)$ $b' \leftarrow \mathcal{A}_2^{\mathsf{LDec}_k(\cdot)}(\mathsf{st}, c^*)$ If $b = b'$ and $\lvert m_0\rvert = \lvert m_1\rvert$, return 1 else return 0

5 Towards secure bootloading and bitstream decryption

The previous section showed how to obtain strong integrity guarantees from an implementation of authenticated encryption where the adversary can both exploit coin misuse and leakage, in encryption and decryption. Yet, and as discussed in introduction, it may be that even if an adversary has control of a decryption device, he is interested to break the confidentiality of the messages. This typically happens in the case of secure bootloading or bitstream decryption. More generally, this happens any time a piece of code is decrypted and the legitimate user is only supposed to use the code, not to access the sources (multimedia content and video games are other typical examples). In this section, we argue that such a context may be captured with a variant of the eavesdropper security introduced in [PSV15] and then show that $\mathsf{DTE2}$ does not reach this goal.

5.1 Eavesdropper Security with Differential Leakage

We extend the notion of indistinguishability of ciphertexts in the presence of an eavesdropper to an environment where decryption leakage can be observed. This leakage is unavoidable in practice when the adversary against the (authenticated) encryption owns a decryption device. Even if decrypted plaintexts might remain hidden into the device, the adversary can run it as many times as desired on any chosen ciphertext and collect more and more leakages adaptively.

The definition below then captures confidentiality through a new experiment which, with respect to the usual eavesdropping experiment, additionally grants the adversary with an unbounded access to LDec. In the new experiment, called EavDL and defined in Table 5, LDec returns the decryption leakage on any chosen ciphertext.

Definition 11. An authenticated encryption $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ with security parameter n and decryption leakage function L_d provides (q,t,ϵ)-indisinguishability of ciphertexts against *eavesdropping with differential leakage attacks*, or is (q,t,ϵ)-eavdl secure for short, if for all (q,t)-bounded adversaries $\mathcal{A}$ we have

$$\Pr\big[\,\mathsf{EavDL}_{\mathsf{AE},\mathsf{L}_d,\mathcal{A}}(1^n) \Rightarrow 1\big] \leq \frac{1}{2} + \epsilon,$$

where q is an upper bound on the number of queries made to the LDec_k oracle. The EavDL experiment is given in Table 5.

As highlighted by the attack against $\mathsf{DTE2}$ in the next subsection, achieving a (strong) form of authenticity does not prevent invalid ciphertexts from leaking information related to the challenge encrypted message. Obviously, the EavDL security requires at least that an authenticated encryption scheme AE does not degrade with the number of queries made to LDec. Informally, more access to LDec should not help the adversary in winning the experiment with better probability even on invalid ciphertexts. Therefore, a first useful step towards proving the EavDL security of AE might be to reduce an adversary in the

EavDL experiment to an adversary in a restricted-EavDL experiment, where decryption leakage is only given for the challenge ciphertext. The restricted-EavDL experiment, next denoted as $\mathsf{r\text{-}EavDL}$, is defined in Table 6.

Table 6: Restricted EavDL experiments.

$\mathsf{r\text{-}EavDL}_{\mathsf{AE},\mathsf{L}_d,\mathcal{A}}(1^n)$ experiment	
Initialization:	Finalization:
$k \leftarrow \mathcal{K}$ s.t. $\lvert k \rvert = n$	$(\mathsf{st}, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathsf{LDec}_k^*(\cdot)}(1^n)$
$b \leftarrow \{0,1\}$	$c^* \leftarrow \mathsf{Enc}_k(m_b)$
Oracle $\mathsf{LDec}_k^*(c)$:	$b' \leftarrow \mathcal{A}_2^{\mathsf{LDec}_k^*(\cdot)}(\mathsf{st}, c^*)$
If $c \neq c^*$, return $\perp$	If $b = b'$ and $\lvert m_0 \rvert = \lvert m_1 \rvert$, return 1
else return $\mathsf{L}_d(c; k)$	else return 0

Definition 12. An authenticated encryption $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ with security parameter n and decryption leakage function L_d provides (q, t, ϵ)-indisinguishability of ciphertexts against *eavesdropping with restricted differential leakage attacks*, or "AE is $\mathsf{r\text{-}eavdl}$-secure" for short, if for all (q, t)-bounded adversaries $\mathcal{A}$ we have

$$\Pr\big[\,\mathsf{r\text{-}EavDL}_{\mathsf{AE},\mathsf{L}_d,\mathcal{A}}(1^n) \Rightarrow 1\big] \leq \frac{1}{2} + \epsilon,$$

where q is an upper bound on the number of queries made to the LDec_k^* oracle. The $\mathsf{r\text{-}EavDL}$ experiment is given in Table 6.

The difference between the EavDL and $\mathsf{r\text{-}EavDL}$ games is reminiscent of the security notions adopted by Barwell et al. [BMOS17] – we refer to it as "BMOS". The BMOS security definitions separate encryption and decryption queries from leakage queries and prevent forwarding beween them. As a result, in the BMOS definitional approach, an implementation that leaks plaintexts in full for each successful leaking encryption or decryption query could still be considered secure against leakages, with tight security bounds. However, such an implementation would be completely insecure when considering the EavDL or $\mathsf{r\text{-}EavDL}$ games: the adversary can just make one LDec query on c^* and win the game. The BMOS definitions share some features with the gap between the EavDL and $\mathsf{r\text{-}EavDL}$ games, though, as this gap measures the advantage that can be gained from accessing decryption leakages on ciphertexts that are not forwarded from the encryption query.

5.2 A differential side-channel attack against DTE2

First observe that the first decryption step in $\mathsf{DTE2}$ consists in performing $\mathsf{PSVEnc}_k^1(\tau, c)$ (see Table 4). Looking back at Table 1, this implies successively computing $k_0 \leftarrow \mathsf{F}_k^*(\tau)$ and $r \leftarrow \mathsf{F}_{k_0}(p_B) \oplus c_0$, $k_1 \leftarrow \mathsf{F}_{k_0}(p_A)$, $m_1 \leftarrow \mathsf{F}_{k_1}(p_B) \oplus c_1, \ldots$ As a result, an adversary can easily use invalid ciphertexts made of a correct tag τ (which allows setting k_0, k_1,… and all the other intermediate keys to their correct value) and incorrect chosen $\tilde{c}_0$'s. By observing the leakage of the operation $\tilde{r} \leftarrow \mathsf{F}_{k_0}(p_B) \oplus \tilde{c}_0$ for multiple $\tilde{c}_0$'s, he can then gradually learn $\mathsf{F}_{k_0}(p_B)$ in full – and therefore r – by Differential Power Analysis (DPA). We informally define DPA (which is typically instantiated by [MOS11]) as a side-channel attack taking advantage of the leakage of multiple (different) inputs. This is in contrast with Simple Power Analysis (SPA) which take advantage of a single input. The message blocks m_1, m_2, … can then be recovered similarly, implying that $\mathsf{DTE2}$ is not EavDL.[3]

[3] A similar attack was shown in [UWM17] in the case of an encryption device. While it does not contradict the proof of DTE, it shows that the confidentiality of a single message block can be compromised

Table 7: EDT leakage-resilient authenticated encryption.

```
EDT
$\mathsf{Enc}_k(r, m)$: parse $m = (m_1, \ldots, m_\ell)$
  • $c = (c_1, \ldots, c_\ell) \leftarrow \mathsf{PSVEnc}_k^0(r, m)$        // encrypt
  • $\tau \leftarrow \mathsf{Tag}_k^1(r, c)$:
      – $h \leftarrow \mathsf{H}(r \| c)$                                  // digest
      – $\tau \leftarrow \mathsf{F}_k^{*,1}(h)$                             // tag
  • return $C \leftarrow (r, c, \tau)$
$\mathsf{Dec}_k(C)$: parse $C = (r, c, \tau)$
  • $h \leftarrow \mathsf{H}(r \| c)$
  • $h^c = (\mathsf{F}_k^{*,1})^{-1}(\tau)$                               // check first
  • if $h \neq h^c$ return $\perp$
  • return $m \leftarrow \mathsf{PSVEnc}_k^0(r, c)$                        // recover after
```

6 A CIML2- and EavDL-secure construction

We now present a new authenticated encryption scheme, called EDT, bridging the gap between EavDL security and r-EavDL security, while preserving the CIML2 security. The design of EDT aims at minimizing the decryption leakages given by invalid ciphertexts in order to restrict the impact of DPA attacks on message confidentiality, with bitstream decryption and secure bootloading as typical applications.

Note that as in previous works (e.g., [PSV15, BKP+16]), our proofs leave an "SPA gap" in the confidentiality reductions (plus the standard black-box gap resulting from the computational security of the encryption scheme), making explicit that we reduce the security of multiple decryptions to the security of a single decryption, yet do not claim that the latter one is achievable with exponentially small advantage. As for the use of a leak-free component in our constructions, we believe such a gap makes the parts of our designs and guarantees that require special care (i.e., hardware-level countermeasures) explicit.

By contrast, we do not remind the additional step of reducing the security of multiple decrypted blocks to the one of a single decrypted block given in these previous works, which is essentially unchanged since the encryption part of EDT is the same as DTE. Concretely, it improves readability by allowing us to avoid re-introducing heavier formalisms for the leakage function (such as the simulatability framework) and to prove our results without further specification of this leakage function.

We finally note that the following constuction can be viewed as an instance of "Encrypt then MAC" scheme, which also brings our designs closer to the previous proposals of Dobraunig et al. [DEM+17] and Barwell et al. [BMOS17].

6.1 EDT specifications

In a nutshell, the EDT scheme combines a tweaked version of PSVEnc with an "Hash-then-MAC" scheme. This Hash-then-Mac produces a digest and a tag, hence the name Encrypt, Digest, Tag. During decryption the validity of the ciphertext is verified before recomputing the plaintext. This reduces the leakage given on invalid ciphertexts in the confidentiality game EavDL. See Table 7 and Figure 3 for a description.

by an SPA encrypting the same message block under different fresh keys (by inverting the role of the plaintext and key in a DPA – which is only possible if the plaintext can be kept constant). The latter relates to the general question of how to formalize the security guarantees that one can hope for a single message block mentioned in introduction, which is one of the important challenges in the field.

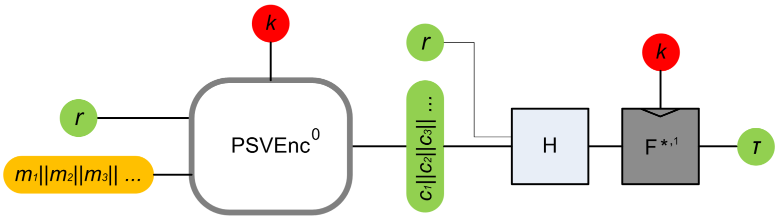

Figure 3: EDT leakage-resilient authenticated encryption.

Compared to DTE2, EDT essentially reverses Tag and PSVEnc and no more encrypts the randomness. Yet, the authentication step in decryption still inverts F^* and checks whether the recomputed digest matches the F^*-preimage of τ. This allows to maintain the CIML2 security.

6.2 EDT security proof

EDT is CIML2 secure and reduces the advantage in EavDL to the advantage in r-EavDL.

Theorem 3. *Let* $\mathsf{H} : \{0,1\}^n \times \{0,1\}^\star \to \{0,1\}^n$ *be a* $(0, t', \varepsilon_{cr})$*-collision resistant and* $(q+1, t', \varepsilon_{pr})$*-range-oriented preimage resistant hash function. Let* $\mathsf{F}^* : \{0,1\}^n \times \{0,1\} \times \{0,1\}^n \to \{0,1\}^n$ *be a* $(2q+2, t', \varepsilon_{\mathsf{F}^*})$*-strong tweakable pseudorandom permutation. Then* EDT *provides* (q, t, ε)*-ciphertext integrity with coin misuse and unbounded leakage on encryption and decryption as long as* $t \le t' - (q+1)(t_\mathsf{H} + (2\ell - 1)t_\mathsf{F})$*, where* t_H *and* t_F *are the time needed to evaluate* H *and* F*, and we have*

$$\varepsilon = \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr}.$$

Only the bound on t slightly differs relatively to Theorem 2. This is due to the fact r is not encrypted in EDT. So, the reduction never has to compute $k_1 = F_{k_0}(p_A)$ and $c_0 = F_{k_0}(p_B) \oplus r$ (see Figure 2). To simplify the notation we then assume that PSVEnc_k^0 first outputs the ephemeral key k_1, and no more k_0 (see Figure 3).

Proof. The proof and all the transition games from the real game, Game-0, to the final game where only the ciphertext computed by EncL are valid, Game-4, are very similar to the proof of Theorem 2. Let q_e denotes the number of encryption queries made to EncL and q_d denotes the number of decryption queries made to DecL made by the adversary $\mathcal{A}$. We then assume $q_e + q_d \le q$. We still denotes $C^{q_d+1} = (\tau^{q_d+1}, c^{q_d+1})$ the $\mathcal{A}$'s output in each game and we consider it as a last decryption query which is not among the answers to EncL. Without loss of generality we also assume that any returned ciphertext by EncL_k is never sent to DecL_k.

Game-0: this is the real CIML2 game where $\mathcal{A}$ attacks EDT. More precisely, once the n-bit key k is fixed, the answer $(C, k_1) = \mathsf{EncL}_k(r, m)$ is performed as: (1) compute $c = \mathsf{PSVEnc}_k^0(r, m)$ which first computes $k_1 = \mathsf{F}_k^{*,0}(r)$ and then $c = (c_1, \ldots, c_\ell)$ from the ephemeral key k_1 and the plaintext $m = (m_1, \ldots, m_\ell)$ (which slightly differs from Figure 2), (2) compute $\tau = \mathsf{Tag}_k^1(r, c)$ as $\tau = \mathsf{F}_k^{*,1}(h)$ where $h = \mathsf{H}(r\|c)$ is the *associated digest*, (3) set $C = (r, c, \tau)$. The answer to a decryption query $C = (r, c, \tau)$ is performed as $\mathsf{DecL}_k(C)$: (1) compute the *draft digest* $h = \mathsf{H}(r\|c)$ and the *check digest* $h^c = (\mathsf{F}_k^{*,1})^{-1}(\tau)$, (2) if $h \ne h^c$ output $(\perp, h^c)$, else, (3) compute $m = \mathsf{PSVEnc}_k^0(r, c)$ which first computes $k_1 = \mathsf{F}_k^{*,1}(r)$ and then $m = (m_1, \ldots, m_\ell)$ with he same process as in encryption, then return $(m, (k_1, h^c))$.

Game-1: we replace all the occurrences of $\mathsf{F}_k^{*,tw}$ and its inverse function in EDT by a truly random permutation f^{tw} and its inverse, with tweak $tw \in \{0,1\}$. The modifications of EncL_k and DecL_k in CIML2 are straightforwards. We simply switch the leak-free PRP, with tweak $tw \in \{0,1\}$, with the random permutation with the same tweak tw. We apply the same switch to the inverse functions.

Transition from Game-0 to Game-1: as easily as in the proof of Theorem 2 we might build a $(2q+2, t_1)$-bounded challenger $\mathcal{B}_1$ with $t_1 = t + (q+1)(t_{\mathsf{H}} + (2\ell - 1)t_{\mathsf{F}})$ against the strong tweakable pseudorandom permutation $\mathsf{F}^{*,\cdot}$ such that $|\Pr[E_0] - \Pr[E_1]| \leq \varepsilon_{\mathsf{F}^*}$.

Game-2: this game is defined as Game-1 except that we introduce the following failure event F_1: some *associated digest* $h^i = \mathsf{H}(r^i \| c^i)$ is equal to the *draft digest* $h^j = \mathsf{H}(r^j \| c^j)$ involved in some decryption query $C^j = (r^j, c^j, \tau^j)$ such that $(r^i, c^i) \neq (r^j, c^j)$, and so even for $j = q_d + 1$.

Transition from Game-1 to Game-2: we have $|\Pr[E_1] - \Pr[E_2]| \leq \Pr[F_1] \leq \epsilon_{cr}$ from a $(0, t_2)$-bounded challenger $\mathcal{B}_2$ against the $(0, t', \epsilon_{cr})$-collision resistance of H since $t_2 \leq t'$.

Game-3: this game is identical to Game-2 except that we introduce the following failure event F_2: the check digest $h^{c,i} = f_1^{-1}(\tau^i)$ of some decryption query $C^i = (r^i, c^i, \tau^i)$ is equal to the draft digest $h^i = \mathsf{H}(r^i \| c^i)$ and τ^i first appears in a decryption query in the game, and so even for $i = q_d + 1$.

Transition from Game-2 to Game-3: we have $|\Pr[E_2] - \Pr[E_3]| \leq \Pr[F_2] \leq \epsilon_{pr}$ from a $(q_d + 1, t_3)$-bounded challenger $\mathcal{B}_3$ against the $(q+1, t', \epsilon_{pr})$-range-oriented preimage resistance of H, since $t_3 = t_2 = t_1 \leq t'$.

Game-4: this game is identical to Game-3 except that we introduce a last failure event F_3: some decryption query on $C^i = (r^i, c^i, \tau^i)$ is valid, and so even if $i = q_d + 1$.

Transition from Game-3 to Game-4: we have $|\Pr[E_3] - \Pr[E_4]| \leq \Pr[F_3] = 0$. Indeed, let $C^i = (r^i, c^i, \tau^i)$ be the valid ciphertext with the smallest index on which $\mathcal{A}$ queries decryption. We consider several cases.

- Case 1: τ^i appears in a response to some encryption query (r^j, m^j) where j is the smallest index satisfying this property.
  - Case 1.a: the jth encryption query happens before the ith decryption query. In that case, we know that the response $C^j = (r^j, c^j, \tau^j)$ differs from C^i and then $(r^i, c^i) \neq (r^j, c^j)$ since we assumed that no decryption query recycles such response. Furthermore, we have $\mathsf{H}(r^i \| c^i) = h^{c,i} = h^{c,j} = \mathsf{H}(r^j \| c^j)$ since both ciphertexts are valid. Then, such a case cannot happen in both games due to the introduction of the failure event F_1.
  - Case 1.b: the jth encryption query happens after the ith decryption query. In both games this situation does not happen since the introduction of the failure event F_2.
- Case 2: τ^i does not appear in any response to encryption query. In that case C^i is already not valid in both games since the introduction of the failure event F_2, which leads to a contradiction.

As a conclusion, $\Pr[\mathsf{CIML2}_{\mathsf{DTE2},\mathsf{L}_e,\mathsf{L}_d,\mathcal{A}}] = \Pr[E_0] \leq |\Pr[E_0] - \Pr[E_1]| + \Pr[F_1] + \Pr[F_2] + \Pr[F_3] + \Pr[E_4] \leq \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr}$. □

□

Theorem 4. *Let* $\mathsf{H} : \{0,1\}^n \times \{0,1\}^\star \to \{0,1\}^n$ *be a* $(0, t', \varepsilon_{cr})$*-collision resistant and* $(q, t', \varepsilon_{pr})$*-range-oriented preimage resistant hash function. Let* $\mathsf{F}^* : \{0,1\}^n \times \{0,1\} \times \{0,1\}^n \to \{0,1\}^n$ *be a* $(2q+2, t', \varepsilon_{\mathsf{F}^*})$*-strong tweakable pseudorandom leakfree permutation. Then,* EDT *is* (q, t, ε)-eavdl *secure with decryption leakage* L_d *such that*

$$\varepsilon \leq \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr} + \epsilon_{\mathsf{SPA}},$$

assuming that EDT *is* $(q, t, \varepsilon_{\mathsf{SPA}})$-$\mathsf{r}$-$\mathsf{eavdl}$ *secure with the decryption leakage* L_d*, and as long as* $t \leq t' - (q+1)(t_{\mathsf{H}} + (2\ell - 1)t_{\mathsf{F}})$ *where* t_{H} *and* t_{F} *are the time needed to evaluate* H *and* F.

We may partially reuse the previous proof of Theorem 3 since L_d cannot leak more than what is leaked in the unbounded leakage model. Roughly, the challenge ciphertext $C^* = (r^*, c^*, \tau^*)$ can be seen as the response to a single encryption query. Then, we can gradually replace the answers to leakage decryption queries by the decryption leakage of invalid ciphertexts. Namely, except for C^*, only the h^c values leaks and they are random on (r, c) due to the strong pseudorandomness of F^*. Then any $h^i = (\mathsf{F}_k^{*,1})^{-1}(\tau^i)$ in the reduction is independent of C^* and $\mathsf{L}_d(C^*)$ as long as $\tau^i \neq \tau^*$. If $\tau^i = \tau^*$, the adversary gets nothing more than the decryption leakage of C^*.

The security of F appears implicitly in this statement, through the security bound associated to the r-eavdl game. Indeed, the r-eavdl game is an extension of the indistinguishable encryption game in the presence of an eavesdropper, and an adversary who can break the eavesdropper security of a scheme can win the r-eavdl game as well (it can just completely ignore the leakages). And a weak function F would make EDT insecure in front of an eavesdropper.

Proof. Let $\mathcal{A}$ be a (q, t)-bounded adversary against EDT. We have to show that

$$\Pr\left[\mathsf{EavDL}_{\mathsf{EDT},\mathsf{L}_d,\mathcal{A}} \Rightarrow 1\right] - \frac{1}{2} \leq \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr} + \varepsilon_{\mathsf{SPA}}.$$

To do so, we will use $\mathcal{A}$ in a sequence of games beginning with the real game $\mathsf{EavDL}_{\mathsf{EDT}}$ with decryption leakage function L_d and ending with the last game $\mathsf{r\text{-}EavDL}_{\mathsf{EDT}}$ with the same decryption leakage function. Each transition between the games will be reduced to an efficient algorithm against either the STPRP of $\mathsf{F}^{*,\cdot}$ or the collision resistance of H or to the range oriented preimage resistance of H, in the unbounded leakage model. In the i-th EavDL game, Game-i, the adversary is face with a modified version of $\mathsf{EavDL}_{\mathsf{EDT}}$ which has been subject to i modification(s). Each modification has the effect of changing the way the oracle responds to the decryption leakage queries. Let E_i be the event whereby $\mathcal{A}$ eventually guesses the right bit b used in the challenge ciphertext C^* which encrypts $m_b \in \{m_0, m_1\}$, namely $\mathsf{Game}\text{-}i_{\mathcal{A}} \Rightarrow 1$.

Game-0: this is the real EavDL game where $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ attacks EDT. More precisely, once the n-bit key k is fixed, the adversary $\mathcal{A}_1$ may request the decryption leakage on any chosen ciphertext C before it outputs $\{m_0, m_1\}$. We will simply say that $\mathcal{A}$ made *pre-challenge* decryption leakage queries. In the challenge phase the adversary receives back $C^* \leftarrow \mathsf{Enc}(m_b)$ for some hidden uniform bit b. Given the challenge ciphertext, the adversary $\mathcal{A}_2$ may request the decryption leakage on any chosen ciphertext C before it outputs its guess b'. We will simply say that $\mathcal{A}$ made *post-challenge* queries. Then $\Pr[E_0]$ is the probability that $\mathcal{A}$ outputs $b' = b$ in this game.

At any time, when $\mathcal{A}$ makes a decryption leakage query on some $C = (r, c, \tau)$, the answer $\mathsf{LDec}_k(C)$ is performed as in the unbounded model as: (1) compute the *draft digest* $h = \mathsf{H}(r\|c)$ and the *check digest* $h^c = (\mathsf{F}_k^{*,1})^{-1}(\tau)$, (2) if $h \neq h^c$ output h^c, else, (3) compute $m = \mathsf{PSVEnc}_k^0(r, c)$ which first computes $k_1 = \mathsf{F}_k^{*,1}(r)$ and then $m = (m_1, \ldots, m_\ell)$ (see Figure 2), then return (k_1, h^c). Note that from k_1 and c the adversary can get m. As a summary, $\mathcal{A}$ either receives (k_1, h^c), if C is valid, or only h^c, if C is not valid.

Game-1: we replace all the occurrences of $\mathsf{F}_k^{*,tw}$ and its inverse function in EDT by a truly random permutation f^{tw} and its inverse, with tweak $tw \in \{0, 1\}$. The modifications of the generations of the challenge ciphertext and the answers to LDec_k in EavDL are straightforwards. We simply switch the leak-free PRP, with tweak $tw \in \{0, 1\}$, with the random permutation with the same tweak tw. We apply the same switch for the inverse functions.

Transition from Game-0 to Game-1: as easily as in the proof of Theorem 3 we might build a $(2q+2, t_1)$-bounded challenger $\mathcal{B}_1$ with $t_1 = t + (q+1)(t_{\mathsf{H}} + (2\ell - 1)t_{\mathsf{F}}) \leq t'$ against $\mathsf{F}^{*,\cdot}$. Indeed, $\mathcal{B}_1$ should make at most 2 calls to its oracle to answer the at most q decryption leakage queries and two more calls to compute the challenge ciphertext. Since $\mathsf{F}^{*,\cdot}$ is a $(2q+2, t', \varepsilon_{\mathsf{F}^*})$-strong tweakable pseudorandom permutation, $|\Pr[E_0] - \Pr[E_1]| \leq \varepsilon_{\mathsf{F}^*}$.

Game-2: this game is defined as Game-1 except that we introduce the following failure event F_1: the *associated digest* $h^* = \mathsf{H}(r^*\|c^*)$ of the challenge ciphertext $C^* = (r^*, c^*, \tau^*)$ is equal to the *draft digest* $h^j = \mathsf{H}(r^j\|c^j)$ involved in some decryption leakage query $C^j = (r^j, c^j, \tau^j)$ such that $(r^i, c^i) \neq (r^j, c^j)$.

Transition from Game-1 to Game-2: we have $|\Pr[E_1] - \Pr[E_2]| \leq \Pr[F_1] \leq \epsilon_{cr}$ from a $(0, t_2)$-bounded challenger $\mathcal{B}_2$ against the $(0, t', \epsilon_{cr})$-collision resistance of H since $t_2 \leq t'$.

Game-3: this game is identical to Game-2 except that we introduce the following failure event F_2: the check digest $h^{c,i} = f_1^{-1}(\tau^i)$ of some decryption leakage query $C^i = (r^i, c^i, \tau^i) \neq C^*$ equals the draft digest $h^i = \mathsf{H}(r^i\|c^i)$ and τ^i first appears in a decryption leakage query.

Transition from Game-2 to Game-3: we have $|\Pr[E_2] - \Pr[E_3]| \leq \Pr[F_2] \leq \epsilon_{pr}$ from a (q_3, t_3)-bounded challenger $\mathcal{B}_3$ against the (q, t', ϵ_{pr})-range-oriented preimage resistance of H, since $t_3 = t_2 = t_1 \leq t'$ and $q_3 \leq q$ (because q_3 is the number of range targets to introduce to each fresh tags in the decryption queries).

Game-4: this game is identical to Game-3 except that we introduce a last failure event F_3: some decryption leakage query on $C^i = (r^i, c^i, \tau^i) \neq C^*$ is valid.

Transition from Game-3 to Game-4: we have $|\Pr[E_3] - \Pr[E_4]| \leq \Pr[F_3] = 0$. Indeed, let $C^i = (r^i, c^i, \tau^i)$ be the valid ciphertext with the smallest index on which $\mathcal{A}$ queries decryption leakage. We consider several cases.

- Case 1: $\tau^i = \tau^*$.
  - Case 1.a: The ith decryption leakage query is a post-challenge query.
    In that case, since $C^j \neq C^*$, we must have $(r^i, c^i) \neq (r^j, c^j)$. Furthermore, we have $\mathsf{H}(r^i\|c^i) = h^{c,i} = h^{c,*} = \mathsf{H}(r^*\|c^*)$ since both ciphertexts are valid. But this contradicts the fact that F_1 no more occurs.
  - Case 1.b: The ith decryption leakage query is a pre-challenge query.
    In both games this situation no more happen since the introduction of the failure event F_2.
- Case 2: $\tau^i \neq \tau^*$. As in the previous case 1.b.

Now, we argue that the probability that $\mathcal{A}$ wins in Game-4 with the decryption leakage function L_d is bounded by $\varepsilon_{\mathsf{SPA}}$: we build a (q, t)-bounded challenger $\mathcal{B}_{\mathsf{SPA}}$ against the $\mathsf{r\text{-}eavdl}$-security of EDT whose advantage is $\Pr[E_4] \leq 1/2 + \varepsilon_{\mathsf{SPA}}$.

The challenger $\mathcal{B}_{\mathsf{SPA}}$ is given the key length n and access to $\mathcal{A}$ as well as the decryption leakage oracle of EDT in the *restricted* experiment, namely to LDec_k^* (see Table 6). The challenger $\mathcal{B}_{\mathsf{SPA}}$ has to emulate the decryption leakage oracle LDec_k in front of $\mathcal{A}$. After the pre-challenge phase, $\mathcal{A}$ sends $\{m_0, m_1\}$ and $\mathcal{B}_{\mathsf{SPA}}$ simply forwards these messages as its own choice. Once $\mathcal{B}_{\mathsf{SPA}}$ receives the challenge ciphertext $C^* = (r^*, c, \tau^*)$ in $\mathsf{r\text{-}EavDL}$ it gives it to $\mathcal{A}$ as the challenge ciphertext in EavDL as done in Game-4. When $\mathcal{A}$ queries the decryption leakage on $C^i = (r^i, c^i, \tau^i) \neq C^*$, $\mathcal{B}_1$ check if τ^i is fresh. If so, $\mathcal{B}_{\mathsf{SPA}}$ simply picks a random $h^i \xleftarrow{\$} \{0,1\}^n$, stores (h^i, τ^i), and sends h^i as the decryption leakage. If not, $\mathcal{B}_{\mathsf{SPA}}$ finds the stored pair (h^j, τ^j) with $\tau^j = \tau^i$ and gives back $h^i := h^j$. Furthermore, each time $\mathcal{A}$ requests the decryption leakage on C^*, $\mathcal{B}_{\mathsf{SPA}}$ asks $\mathsf{LDec}_k^*(C^*)$ and receives $\mathsf{L}_d(C^*)$ which it gives to $\mathcal{A}$ as $\mathsf{LDec}_k(C^*)$. Eventually, $\mathcal{A}$ outputs b' and $\mathcal{B}_{\mathsf{SPA}}$ returns b' as its own guess.

Clearly, the distribution of all the values are the identical to the one of Game-4. Moreover, $\mathcal{B}_{\mathsf{SPA}}$ is indeed a (q, t)-bounded adversary against DTE with the leakage function L_d. Then, by the assumption on the $\mathsf{r\text{-}eavdl}$-security we thus find $\Pr[E_4] \leq 1/2 + \varepsilon_{\mathsf{SPA}}$.

By summarizing all the above results, we find that $\Pr[\mathsf{EavDL}_{\mathsf{EDT},\mathsf{L}_d,\mathcal{A}}] = \Pr[E_0] \leq |\Pr[E_0] - \Pr[E_1]| + \Pr[F_1] + \Pr[F_2] + \Pr[F_3] + \Pr[E_4] \leq \epsilon_{\mathsf{F}^*} + \epsilon_{cr} + \epsilon_{pr} + \varepsilon_{\mathsf{SPA}} + \frac{1}{2}$, which concludes the proof. □ □

Remark 1. Unlike DTE2, EDT is not a misuse resistant authenticated encryption scheme [RS06, BKP+16]. Indeed, using a common r the encryption on distinct messages $m = (m_1, m_2, \dots, m_\ell)$ and $m' = (m_1, m'_2, \dots, m'_\ell)$ share the same $c_1 \leftarrow \mathsf{F}_{k_1}(p_B) \oplus m_1$, since $k_1 \leftarrow \mathsf{F}^{*,0}(r)$. Therefore, ciphertexts are not pseudorandom on random-message pair (r, m).

7 Conclusion

To conclude this work, we observe that DTE2 provides CIML2 security (as shown in this paper) and misuse-resistance without leakages (as shown in [BKP+16]) but not EavDL security. By contrast, EDT provides CIML2 and EavDL security, in the model and based on the assumptions discussed in this paper, but not misuse-resistance without leakage (as just mentioned in Remark 2). Hence, they can be viewed as two complementary solutions to reach different levels of leakage-resilience and misuse-resistance.

Interestingly, adding the missing property to each of these constructions seems to require an additional pass on the (message or ciphertext) blocks. That is, DTE2 could be extended towards EavDL security by adding a “Hash then MAC" on the ciphertexts. Similarly, EDT could be extended towards misuse-resistance without leakage by adding a second encryption pass on the message blocks. In both cases, it would require an additional call to the leak-free strong PRP. Finding solutions to reach these goals with less leak-free blocks (or showing impossibility) is an interesting scope for further research.

References

[ABL+14] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to securely release unverified plaintext in authenticated encryption. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *LNCS*, pages 105–125. Springer, 2014.

[AFL+16] Farzaneh Abed, Christian Forler, Eik List, Stefan Lucks, and Jakob Wenzel. RIV for robust authenticated encryption. In *Fast Software Encryption - 23rd International Conference, FSE 2016*, volume 9783 of *LNCS*, pages 23–42. Springer, 2016.

[AS11] Elena Andreeva and Martijn Stam. The symbiosis between collision and preimage resistance. In Liqun Chen, editor, *Cryptography and Coding - 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*, volume 7089 of *Lecture Notes in Computer Science*, pages 152–171. Springer, 2011.

[BDPA08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[BDPS13] Alexandra Boldyreva, Jean Paul Degabriele, Kenneth G. Paterson, and Martijn Stam. On symmetric encryption with distinguishable decryption failures. In *FSE 2013*, volume 8424 of *LNCS*, pages 367–390. Springer, 2013.

[BKP$^+$16] Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Leakage-resilient and misuse-resistant authenticated encryption. *IACR Cryptology ePrint Archive*, 2016:996, 2016.

[BMOS17] Guy Barwell, Daniel P. Martin, Elisabeth Oswald, and Martijn Stam. Authenticated encryption in the face of protocol and side channel leakage. *IACR Cryptology ePrint Archive*, 2017:68, 2017.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.

[BPS15] Guy Barwell, Daniel Page, and Martijn Stam. Rogue decryption failures: Reconciling AE robustness notions. In *IMACC 2015*, volume 9496 of *LNCS*, pages 94–111. Springer, 2015.

[DEM$^+$17] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, and Thomas Unterluggauer. ISAP - towards side-channel secure authenticated encryption. *IACR Trans. Symmetric Cryptol.*, 2017(1):80–105, 2017.

[DKM$^+$15] Christoph Dobraunig, François Koeune, Stefan Mangard, Florian Mendel, and François-Xavier Standaert. Towards fresh and hybrid re-keying schemes with beyond birthday security. In Naofumi Homma and Marcel Medwed, editors, *Smart Card Research and Advanced Applications - 14th International Conference, CARDIS 2015, Bochum, Germany, November 4-6, 2015. Revised Selected Papers*, volume 9514 of *Lecture Notes in Computer Science*, pages 225–241. Springer, 2015.

[EPW10] Thomas Eisenbarth, Christof Paar, and Björn Weghenkel. Building a side channel based disassembler. *Trans. Computational Science*, 10:78–99, 2010.

[GL17] Shay Gueron and Yehuda Lindell. Better bounds for block cipher modes of operation via nonce-based key derivation. Cryptology ePrint Archive, Report 2017/702, 2017. http://eprint.iacr.org/2017/702.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In *EUROCRYPT*, volume 9056 of *LNCS*, pages 15–44. Springer, 2015.

[MBKP11] Amir Moradi, Alessandro Barenghi, Timo Kasper, and Christof Paar. On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from xilinx virtex-ii fpgas. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011*, pages 111–124. ACM, 2011.

[MOS11] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.

[MSGR10] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In Daniel J. Bernstein and Tanja Lange, editors, *Progress in Cryptology - AFRICACRYPT 2010, Third International Conference on*

Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2010. Proceedings, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.

[OC15] Colin O'Flynn and Zhizhang (David) Chen. Side channel power analysis of an AES-256 bootloader. In *IEEE 28th Canadian Conference on Electrical and Computer Engineering, CCECE 2015, Halifax, NS, Canada, May 3-6, 2015*, pages 750–755. IEEE, 2015.

[PSV15] Olivier Pereira, François-Xavier Standaert, and Srinivas Vivek. Leakage-resilient authentication and encryption from symmetric cryptographic primitives. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS*, pages 96–108. ACM, 2015.

[RS06] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *LNCS*, pages 373–390. Springer, 2006.

[UWM17] Thomas Unterluggauer, Mario Werner, and Stefan Mangard. Side-channel plaintext-recovery attacks on leakage-resilient encryption. In David Atienza and Giorgio Di Natale, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1318–1323. IEEE, 2017.