Ciphertext Integrity with Misuse and Leakage: Definition and Efficient Constructions with Symmetric Primitives

Francesco Berti, François Koeune, Olivier Pereira,
Thomas Peters, François-Xavier Standaert.
ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium
francesco.berti,francois.koeune,thomas.peters,olivier.pereira,fstandae@uclouvain.be

ABSTRACT

Leakage resilience (LR) and misuse resistance (MR) are two important properties for the deployment of authenticated encryption (AE) schemes. They aim at mitigating the impact of implementation flaws due to side-channel leakages and misused randomness. In this paper, we discuss the interactions and incompatibilities between these two properties.

We start from the usual definition of MR for AE schemes from Rogaway and Shrimpton, and argue that it may be overly demanding in the presence of leakages. As a result, we turn back to the basic security requirements for AE: ciphertext integrity (INT-CTXT) and CPA security, and propose to focus on a new notion of CIML security, which is an extension of INT-CTXT in the presence of misuse and leakages.

We discuss the extent to which CIML security is offered by previous proposals of MR AE schemes, conclude by the negative, and propose two new efficient CIML-secure AE schemes: the DTE scheme offers security in the standard model, while the DCE scheme offers security in the random oracle model, but comes with some efficiency benefits. On our way, we observe that these constructions are not trivial, and show for instance that the composition of a LR MAC and a LR encryption scheme, while providing a (traditional) MR AE scheme, can surprisingly lose the MR property in the presence of leakages and does not achieve CIML security. Eventually, we show the LR CPA security of DTE and DCE.

KEYWORDS

Authenticated encryption, leakage-resilient cryptography, misuse resistance

ACM Reference Format:
Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, François-Xavier Standaert.. 2018. Ciphertext Integrity with Misuse and Leakage: Definition and Efficient Constructions with Symmetric Primitives. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Authenticated Encryption (AE) has become the standard primitive for secure message transmission: after its introduction by Bellare and Namprempre [9] and Katz and Yung [26], a first round of ISO standardization in 2009 [25], and the ongoing CAESAR competition [14], emerging standards like TLS 1.3 only include AE schemes for record protection [39]. One of the main reason of this success is the ease of use of AE, compared to earlier solutions based on combinations of encryption and message authentication codes that led to numerous security issues [2, 18, 36].

It is then natural that important efforts have been focusing on making AE schemes more robust to various misuses and implementation issues. For example, a first line of work focuses on randomness misuses, and particularly on the use of poor quality IVs [42], making sure that the damages resulting from these are kept minimal. This offers substantial improvement compared to some schemes that can go as far as fully leaking their long-term secret key if the same randomness is used twice [35].

We explore the goal of misuse resistance (MR) taken in combination with a second desirable property: leakage resilience (LR). This property aims at making sure that cryptographic schemes behave as well as possible in the presence of leakages resulting from their implementation. While the original focus has been on leakages resulting of timings, power consumption, or electromagnetic radiations of embedded devices, LR has become a desirable feature for implementation in high(er)-end devices, following recent works on timing attacks against OpenSSL [1, 22], or power and electromagnetic analysis of powerful ARM cores running at high frequencies [5, 28].

Our work focuses on LR for leakages happening at encryption time. This is practically relevant, for instance in applications in which only one party (e.g., a power constrained smart-card), in charge of encryption, is susceptible of producing side-channel leakages, while the other party (e.g., a reader), in charge of decryption, can easily be physically shielded. Besides even in the case of bidirectional communications, many standards (e.g., TLS [39]) use a different key for each direction of communication, and it is therefore relevant to protect the encryption key, independently of the security of the decryption key. We also reckon that having

LR decryption (or just LR integrity verification) may be considerably harder, due to the deterministic nature of these functionalities: the question was discussed and left open in earlier works [37], or required the adoption of primitives that are considerably more complex and expensive than a block cipher [31].

This focus on encryption makes our work orthogonal to a third line of robustness improvements for AE, which focuses on the handling of information leakages that can result from decryption error messages and leakage of information computed during the decryption of incorrect ciphertexts [3, 7, 13, 24].

Our contributions. In the absence of security definitions for LR AE, a natural starting point for our work is to consider the definition of MR AE and try to enhance it with LR. Informally speaking, an AE scheme is MR-secure if it produces ciphertexts that all look random, even to an adversary controlling the random IV or nonce used during encryption, and if it is not feasible to produce a ciphertext that decrypts otherwise than as the result of the encryption of a message. The random-looking ciphertexts guarantee confidentiality, and the infeasibility to forge ciphertexts guarantees integrity.

Extending MR AE security to a notion of LR MR AE security seems very challenging, though: the natural path would be to augment with leakages the output of the encryption and random oracles (which are expected to be indistinguishable). The encryption oracle would provide the real leakages from the implementation. It is hard, however, to see how to define the leakages on the random side, as the output of this random oracle does not correspond to any real computation: how do we produce something that looks like the power consumption trace for a computation that does not exist and is completely inconsistent with its inputs?

As a result of this difficulty, we turn back to (a common formulation of) the original security requirements for AE: ciphertext integrity (INT-CTXT) and CPA security [9]. Focusing on INT-CTXT, we propose a new notion of ciphertext integrity in the presence of misuse and leakage (CIML). CIML security measures the hardness to produce a fresh valid ciphertext in the presence of an oracle that produces ciphertexts and the associated leakages when queried on adversarially chosen messages and IV's or nonces.

We then investigate the CIML security of several MR AE schemes. We observe that popular SIV-based constructions [42] offer very little protection in the presence of leakages. More surprisingly, even by combining recent LR encryption and MAC schemes from Pereira et al. [37] into a MR AE scheme, which we call PSV-AE, we still obtain a scheme that is quite sensitive to side-channel attacks and not CIML secure.

Given this state of affairs, we design two new AE schemes, which we show to be CIML secure. We follow a pragmatic design approach that combines the minimal use of an (expensive) leak-free component with much more efficient (less protected) implementations [37]. Such a model nicely matches the reality of modern embedded devices, where physical security against side-channel attacks is now a necessary condition for deployment, while cost constraints require to limit the overheads of the countermeasures against such attacks. Concretely, the leak-free component will typically be implemented by a block cipher (e.g., the AES Rijndael) protected with a combination of hardware and algorithmic techniques, e.g., noise addition [29], masking [40] and shuffling [45]. The latter ones usually increase the "code size × cycle count" metric (for software implementations) or the "throughput / area" metric (for hardware ones) by factors ranging from hundreds to thousands, hence motivating their minimal use.[1] In practice, this good tradeoff between security and performance is achieved by requiring a small constant number (1 or 2, depending on the scheme) of executions of the leak-free component, independently of the length of the message to be encrypted or authenticated. For long messages, the majority of the computational work can then be performed by weakly protected block-cipher implementations.

Our first scheme, which we call DTE for "Digest, Tag and Encrypt", is CIML secure in the standard model, based on a very permissive leakage model, which we call the *unbounded leakage model*. In this model, everything is leaked, except for the state of our leak-free component, which we need to use twice. We also show that it is a MR AE scheme (in the absence of leakages).

Our second scheme, which we call DCE for "Digest, Commit and Encrypt", is more efficient in the sense that it only requires a single use of the leak-free component. However, we can only prove its CIML security in the random oracle model (but still in the unbounded leakage model). Furthermore, DCE is not a MR AE scheme, as its ciphertexts do not look random, even in the absence of leakages. We insist that the introduction of random oracles in the analysis of LR constructions is questionable (since the random oracle abstraction excludes leakages). So this last proposal should be viewed as provocative and is mainly aimed at stimulating discussions and cryptanalysis.

We conclude by showing the LR CPA security of DTE and DCE. Given that we are interested in a confidentiality property, the unbounded leakage model is not suitable anymore: it would immediately leak the plaintext corresponding to each encrypted message. We then turn to the simulatable leakage model [43], which we extend in order to capture a notion of simulatable leakage for hash functions (only block ciphers have been considered in that model until now). In this context, the LR CPA security means that one can reduce the security of a primitive for many messages and blocks to the security of the primitive for one message block, without claims on the exact security of this iteration (which depends on the implementations) – see [37] and Section 7 for the details. While we are aware of the ongoing discussion about how to implement block ciphers ensuring simulatable leakages [27], this assumption remains the most realistic solution to reason about leakage we currently have (and in particular,

[1] See Table 4 in [37] for an illustration of these overheads.

the only one that can be challenged by hardware engineers). It is also shown by Fuller and Hamlin to be one of the least demanding assumptions available in the LR literature, compared to bounded leakages or indistinguishable leakages for instance [21]. Besides, we insist that we only use it to show that our CIML-secure schemes maintain the same LR CPA security level as previously published LR encryption schemes (i.e., that our improvements on the INT-CTXT side have no downside on message confidentiality compared to the state-of-the-art). The properties of our constructions are summarized in Table 1.

Table 1: Summary of our constructions. LMCPA = leakage resilient chosen plaintext attack security for multiple messages and blocks; (~~LR~~) MR = misuse-resistance in the absence of leakage; CIML = ciphertext integrity with misuse and leakage; LF executions counts the number of executions of the leak free component that are required for an encryption; the models are either the standard one, denoted std., or the random oracle model, denoted RO.

	LMCPA	(~~LR~~) MR	CIML	LF executions
PSV-AE	std.	std.	✗	2
DTE	std.	std.	std.	2
DCE	std.	✗	RO	1

Related works. Two recent (independent) reports proposed alternative constructions of LR AE schemes. The first one, by Dobraunig et al. [16], combines a concrete instance of fresh re-keying (borrowed from [17, 32]), with a sponge-based construction [12]. Due to the nature of these components, their security analysis is (so far) more heuristic. Yet, it comes with the nice and intuitive observation that one can naturally capture certain classes of leakage functions by reducing the capacity of the sponge.

The second one, by Barwell et al. [6], shares some goals with ours (as it also aims to combine both MR and LR) with a few significant differences though. First, and conceptually, this work is more focused on composition results, while we pay a particular attention to efficient instances of AE schemes. As a result of this choice, a second difference is that their instantiations require all the building blocks to be well protected against side-channel analysis, while we aim to minimize the use of a leak-free component. Concretely, this difference is reflected by different encryption modes: the instances in [6] are based on the standard Cipher Feed Back (CFB) mode, which is insecure in the simulatable leakage model that we use (because of the continuous reuse of a single long-term key), while we leverage the literature on LR stream ciphers in order to reduce the use of our leak-free component [19, 20, 38, 43, 46, 47]. Third, and more technically, we discuss what can be achieved by symmetric cryptographic building blocks, while the work by Barwell et al. would be implemented using elliptic curves operations for each message block. (Note the pairing-based LR PRF proposed in the latter work could be one more option to instantiate our leak-free component, So these two pieces of work are essentially complementary).

Paper structure. In Section 2, we review the main definitions and notations used in the paper. Section 3 defines and motivates our new notion of ciphertext integrity in the presence of misuse and leakages (CIML). Section 4 reviews some constructions of MRAE schemes, and shows how they do not achieve CIML security. Sections 5 and 6 introduce the DCE and DTE schemes, and show their CIML security. Eventually, Section 7 shows the LR CPA security of DTE and DCE.

2 BACKGROUND

We denote as a (q,t)-bounded algorithm a probabilistic algorithm that can make at most q queries to the oracles he is granted access to and can perform computation bounded by running time t.

2.1 Definitions

We first need the following definition of collision-resistant hash function.

Definition 2.1. 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.

Definition 2.2. A $(0,t,\epsilon_{\mathsf{pr}})$-*range-oriented preimage 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 $\mathcal{A}(s,y)$ outputs a message $m\in\mathcal{M}$ such that $\mathsf{H}^s(m)=y$ is bounded by $\varepsilon_{\mathsf{pr}}$, where $s\leftarrow\mathcal{S}$, $y\leftarrow\mathcal{B}$ are selected uniformly at random.

Note that the usual notion of preimage resistance samples a random $m_0\leftarrow\mathcal{M}$ over the *domain* of H^s and then sets $y=\mathsf{H}^s(m_0)$. Definition 2.2, which was introduced in [4], uniformly samples $y\leftarrow\mathcal{B}$ over the *range* of H^s.

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 definition of pseudorandom function.

Definition 2.3. A function $\mathsf{F}:\mathcal{K}\times\mathcal{B}\rightarrow\mathcal{T}$ is a $(q,t,\epsilon_{\mathsf{F}})$-*pseudorandom function* (PRF) 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 $\mathcal{B}$ to $\mathcal{T}$.

In order to capture authenticity, we introduce the notion of IV-based MAC. We use this variant of the standard definition of MAC (with no IV) because it gives compatibility with previous constructions of LR MAC's [37], which we will be using.

Definition 2.4. An IV-based MAC is a tuple $\mathsf{ivM} = (\mathcal{K}, \mathsf{Mac}, \mathsf{Vrfy})$ such that:

- $\mathsf{Mac} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \to \mathcal{T}$ takes a key, an IV, and a message and outputs a tag.
- $\mathsf{Vrfy} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \times \mathcal{T} \to \{\top \cup \bot\}$ and outputs $\top$ only if τ is a valid tag for IV, message m and key k.

We assume that, $\forall k \in \mathcal{K}$, $\forall IV \in \mathcal{IV}$, $\forall m \in \mathcal{M}$, it holds that $\mathsf{Vrfy}_k(IV, m, \mathsf{Mac}_k(IV, m)) = \top$.

We also define the probabilistic algorithm $\mathsf{MAC} : \mathcal{K} \times \mathcal{M} \to \mathcal{IV} \times \mathcal{T}$ which, on inputs $k \in \mathcal{K}$ and $m \in \mathcal{M}$ picks a random $IV \in \mathcal{IV}$ and outputs IV and $\tau \leftarrow \mathsf{Mac}_k(IV, m)$

While the traditional security property required from MACs is unforgeability, our constructions rely on a stronger property of the Mac function: we require Mac to be a pseudorandom function for the $(\mathcal{IV} \times \mathcal{M})$ input space.

Definition 2.5. ivM is $(q, t, \epsilon_{\mathsf{cip}})$ *chosen-IV pseudorandom* if the function $\mathsf{Mac} : \mathcal{K} \times (\mathcal{IV} \times \mathcal{M}) \to \mathcal{T}$ is a $(q, t, \epsilon_{\mathsf{cip}})$-pseudorandom function.

Our AE schemes will be based on IV-based encryption schemes, which we define following Rogaway and Shrimpton [41].

Definition 2.6. An IV-based encryption scheme is a tuple $\mathsf{ivE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ such that:

- $\mathsf{Enc} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \to \mathcal{C}$ maps a key selected from $\mathcal{K}$, an IV selected from $\mathcal{IV}$ and a message from $\mathcal{M}$ to a ciphertext from $\mathcal{C}$.
- $\mathsf{Dec} : \mathcal{K} \times \mathcal{IV} \times \mathcal{C} \to \mathcal{M}$ provides the decryption of a pair containing an IV and a ciphertext.

We also use $\mathsf{ENC} : \mathcal{K} \times \mathcal{M} \to \mathcal{IV} \times \mathcal{C}$ for the probabilistic function that picks a uniformly random IV and returns the ciphertext $(IV, \mathsf{Enc}(k, IV, m)) \leftarrow \mathsf{ENC}_k(m)$.

To capture message secrecy, we use the security definition of Namprempre et al. [34] and consider a distinguishing game in which the adversary tries to determine whether he is facing an encryption oracle or a random function.

Definition 2.7. An IV-based encryption scheme $\mathsf{ivE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ is $(q, t, \epsilon_{\mathsf{IV\text{-}sec}})$-IV-sec secure if for any $k \leftarrow \mathcal{K}$ and for every (q, t)-adversary $\mathcal{A}$, the advantage

$$\mathbf{Adv}^{\mathsf{IV\text{-}sec}}_{\mathsf{ivE},\mathcal{A}} := \left| \Pr\left[\mathcal{A}^{\mathsf{ENC}_k(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot)} \Rightarrow 1\right] \right|$$

is upper-bounded by $\epsilon_{\mathsf{IV\text{-}sec}}$, where $\$(m)$ picks a random $IV \leftarrow \mathcal{IV}$ and outputs (IV, σ), where σ is a random bit string of length $|\mathsf{Enc}_k(IV, m)|$.

Resistance against misuse then captures the security in front of an adversary controlling the generation of the randomness used for encryption. In the case of AE, the adversary is also granted access to a decryption oracle. We consider a definition of *misuse-resistant authenticated encryption* similar to the one appearing in [41].

Definition 2.8. An *authenticated encryption scheme* is a tuple $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ such that:

- $\mathsf{Enc} : \mathcal{K} \times \mathcal{R} \times \mathcal{M} \to \mathcal{C}$ maps a key selected from $\mathcal{K}$, randomness selected from $\mathcal{R}$ and a message from $\mathcal{M}$ to a ciphertext in $\mathcal{C}$.
- $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\bot\}$ maps a key and a ciphertext to a message that is the decryption of that ciphertext, or to a special symbol $\bot$ if decryption fails.

The associated probabilistic algorithm ENC first picks a random coin $r \in \mathcal{R}$ and returns $c = \mathsf{Enc}_k(r, m) := \mathsf{Enc}(k, r, m)$. We stress that Dec_k only needs c to recover m, which is the main difference between our definition and previous IV-based schemes for which an IV additionally needs to be provided. This slight variation allows for instance to embed an encryption of the IV in the ciphertext as done in our DTE scheme, which will be motivated by our improved LR goal as detailed in the next sections.

The definition of MR due to [41] is tailored for IV-based AE while our definition focuses on AE as in Definition 2.8.

Definition 2.9. 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

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

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 $\bot(c)$ outputs $\bot$ except if c was output by the $\$(\cdot, \cdot)$ oracle earlier, in which case it returns the associated m.

In the rest of the paper, we will simply refer to this notion as Misuse Resistance (MR).

Note that, for conciseness, we ignore the specific treatment of associated data in our constructions, which is orthogonal to the discussions on MR and LR that motivate our results.

2.2 Security parameter

We provide explicit adversary's advantages for all the constructions in the paper. Whenever instantiating our building blocks, we will consider $\mathcal{K} = \mathcal{T} = \mathcal{R} = \mathcal{B} = \mathcal{IV} = \{0,1\}^n$ using n as a security parameter, and $\mathcal{M} = \{0,1\}^{\ell n}$, (i.e., a message is made of on ℓ blocks of n bits) so that the advantages are negligible in n.

3 INT-CTXT WITH MISUSE AND LEAKAGES

Motivation. Definition 2.9 seems a natural starting point to define AE schemes that offer security in the presence of misuse and leakages.

It however makes a very strong requirement: it requires ciphertexts to be indistinguishable from random bits. This is a strengthening (which is already visible in Def. 2.7) of

usual confidentiality requirements of indistinguishable encryption, which require ciphertexts to be indistinguishable of the encryption of random messages, but not to have ciphertexts that are themselves indistinguishable from random bits. While this strengthening does not look overly constraining for practical schemes, it clearly rules out some schemes that look satisfactory from a security point of view. For instance, modifying a MR AE scheme by concatenting the bit "0" to all ciphertexts looks benign from a security point of view, but removes the MR property, since ciphertexts do not look uniformly distributed anymore.

This becomes a real issue if we want to use such a definition in the presence of leakages, and a similar difficulty was already faced in the early work of Micali and Reyzin [33] when they separated indistinguishability and unpredictability in the side-channel security of pseudorandom generators: in essence, leaking about a value prevents that value from looking random.

Concretely, an extension of Def. 2.9 to a world with leakages would focus on an advantage of the form:

$$\left|\Pr\left[\mathcal{A}^{\mathsf{EncL}_k(\cdot,\cdot),\mathsf{Dec}_k(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$\mathsf{L}(\cdot,\cdot),\perp(\cdot)} \Rightarrow 1\right]\right|$$

in which the EncL oracle is the usual Enc oracle modified in such a way that it also outputs the leakage happening during the computation of $\mathsf{Enc}_k(IV, m)$ (remember that we focus on leakages during encryption only). Now, the difficulty comes when defining $\$\mathsf{L}$: how do we define a leakage corresponding to an idealized computation that cannot be implemented physically, which would be an encryption of m that outputs a randomly chosen bit string instead of a ciphertext? And we cannot just ignore that leakage, as removing it would make it trivial for the adversary to distinguish the real world with leakages, from the random/ideal world.

In this context, one tempting solution is to posit that leakages can be simulated, namely, that there would be a leakage simulator that, given a pair (IV, m) and the random output of the $\$(IV, m)$ oracle, can produce a leakage that is indistinguishable from the one produced during the real encryption. This would somehow assume that leakages are zero-knowledge functions.

Such an assumption about leakages has been proposed by Standaert et al. [43], and variants have been explored by Fuller and Hamlin [21]. Informally, the implementation of a block cipher is said to have q-simulatable leakages if it is possible to simulate the leakages of this implementation, given public inputs and outputs but no key, for at most q evaluations of this block cipher with any key.

It is well known that such an assumption can only be fulfilled under strong restrictions. Indeed, a side-channel attack typically reduces the computational secrecy of the state manipulated by a device at a rate that is exponential in the number of leakages. In this context, the only hope to have simulatable leakages is to strongly limit their number (typically, $q = 2$) and to make them noisy. As discussed in [27], even simulating a small number of noisy leakages is difficult. Hence, assuming simulatability without such strong restrictions appears to be completely unrealistic.

Still, this is exactly what the context of MR would require: given that the adversary is in control of the IV, nothing prevents him from querying the $\mathsf{Enc}^{\mathsf{L}}$ oracle with a single (IV, m) pair as many times as desired, precisely in order to be able to remove all the noise of the leakages (through averaging), opening the door to attacks such as described in [8]. (In Section 7, we will analyze the LR CPA security of our schemes using the assumption of 2-simulatable leakages. The crucial difference, there, is that the adversary will not be in control of the IV anymore, which will make averaging strategies fail as long as there is no IV collision.)

As a result of this central difficulty coming from MR, we need to adopt a different approach, one that would not include any requirement of random-looking ciphertexts. To this purpose, we turn back to the standard security requirements of AE schemes (without misuse), of which one formulation is the combination of ciphertext integrity (INT-CTXT) and indistinguishability under CPA [9].

CIML security. We propose a notion of ciphertext integrity in the presence of misuse and leakages. The traditional INT-CTXT property requires that an adversary, who can query an encryption oracle on chosen messages, is unable to produce a ciphertext that is different from those received from the encryption oracle, but would still pass the decryption algorithm without error.

Asking that this property remains satisfied in the presence of IV misuse can be expressed, by letting the adversary select the IV's that are submitted to the encryption oracle. And capturing encryption leakages can be expressed by letting the encryption oracle return the leakage corresponding to the encryption that is performed. This results in the following security definition.

Definition 3.1. An authenticated encryption AE with encryption leakage function L provides (q, t, ϵ)-*ciphertext integrity with coin misuse and leakage on encryption* if for all (q, t)-bounded adversaries $\mathcal{A}$, we have

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

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

$\mathsf{CIML}_{\mathsf{AE},\mathsf{L},\mathcal{A}}$ experiment	
Initialization: $k \xleftarrow{\$} \mathcal{K}$ $\mathcal{S} \leftarrow \emptyset$ Finalization: $C \leftarrow \mathcal{A}^{\mathsf{EncL}_k(\cdot,\cdot),\mathsf{Dec}_k(\cdot)}$ If $C \in \mathcal{S}$, return $\perp$ If $\perp = \mathsf{Dec}_k(C)$, return $\perp$ return 1	Oracle $\mathsf{EncL}_k(r, m)$: $C = \mathsf{Enc}_k(r, m)$ $\mathcal{S} \leftarrow \mathcal{S} \cup \{C\}$ return $(C, \mathsf{L}(r, m; k))$ Oracle $\mathcal{O}^{\mathsf{Dec}_k(C)}$: return $\mathsf{Dec}_k(C)$

As for LR CPA security, we will simply use the LMCPA security notion already defined in [37]. We defer the treatment of this second security goal to Section 7.

4 PREVIOUSLY PROPOSED CONSTRUCTIONS

In this section, we review some constructions of MR AE schemes, and explain how they would fail to offer CIML security.

Rather than focusing on a specific type of leakage functions (bounded leakages, indistinguishable leakages, simulatable leakages, hard to invert leakages, ...) [21] that would require much formalism, we explain how practical side-channel attacks could be mounted against these schemes, focusing on two standard attack methods: simple power analysis (SPA) and differential power analysis (DPA). Informally, DPAs are the most commonly exploited side-channel attacks and take advantage of the leakage about a secret from a computation based on multiple (different) inputs [30]. They reduce the computational secrecy of the state manipulated by a device at a rate that is exponential in the number of leakages, by combining the information of these different inputs (e.g., plaintexts). SPAs are side-channel attacks taking advantage of the leakage of a single input, possibly measured multiple times to reduce the measurement noise, e.g., by exploiting powerful (yet less practical) algebraic/analytical techniques [44].

We also consider that the adversary is in possession of a copy of the targeted device, which he can feed with any choice of plaintexts and keys in order to obtain outputs and leakages. This is traditionally used for profiling (i.e., learning how to interpret leakages) [15], but can also be used for efficient matching attacks. Indeed, even if it is sometimes difficult to *extract* a secret key from a power consumption trace (which can take gigabytes of data for the encryption of a single message block), it is typically easier to *recognize* whether a candidate secret is correct, by matching the leakage from the attacked device and the one obtained from the training device when fed with the candidate values [43]. Note that the latter can be viewed as a type of SPA with (much) simplified adversarial goal.

Summarizing, whenever a DPA is possible, it is the most devastating attack due to the exponentional rate at which it reduces the secrecy of the device state. When only SPA is possible, key extraction is more challenging than leakage matching, and the difference will be especially large when the noise is the measurements is limited.

4.1 The SIV construction

The SIV construction was introduced by Rogaway and Shrimpton [42] and is a popular approch for the construction of MR AE schemes. Encryption proceeds by applying a PRF (with a first key k_1) to the message (and to the associated data, if there are some) in order to obtain an IV, and using this IV as input to an IV-based encryption scheme, which uses a second key k_2 and returns a ciphertext c. The output is then the (IV, c) pair. This construction has been instantiated into the SIV and GCM-SIV modes for example [23, 42]. These two instances offer the same angle of attack.

First, a DPA is used to recover k_1. It is fairly easy on these schemes, because the constant value k_1 is used on each block of each message that is encrypted, and these blocks are adversarially chosen.

Then, based on k_1, it is possible to use the properties of the underlying PRF (or universal hash function in the case of GCM-SIV) to build two messages m, m^* that have the same IV. In the case of SIV, the PRF is CMAC, a close variant of CBC MAC. Simply put, if we have a message $m = (m_1 \| \cdots \| m_\ell)$ made of ℓ full blocks, a block m_0 is set to 0^n, tags $t_1, \ldots, t_\ell$ are computed as $t_i = F_{k_1}(m_i \oplus t_{i-1})$, and the output is $IV = t_\ell$. Now, if k_1 has been obtained through DPA, we can modify the i-th block into m'_i, compute the updated value t'_i, and adjust the $i+1$-th block m'_{i+1} as $m_{i+1} \oplus t_i \oplus t'_i$, which will guarantee that $t'_{i+1} = t_{i+1}$, and so on for all the next tags. A similar process can be applied to GCM-SIV, which uses the GHASH universal hash function instead of CMAC.

Eventually, since both SIV and GCM-SIV use the counter mode for their IV-based encryption part, we can adapt a ciphertext (IV, c) encrypting m into a different ciphertext $(IV, c \oplus m \oplus m^*)$ that decrypts to m^*, hence breaking CIML security.

As for LR CPA security, similar issues show up. Indeed, as soon as k_1 has been recovered by DPA, a leakage matching attack is easy to mount on the test query. Indeed, when the adversary asks to encrypt one message out of m_0 and m_1 and receives an encryption of m_b (for a random b) together with the corresponding leakage, the adversary can use his own device to produce the leakages corresponding to the evaluation of the PRF on m_0 and m_1: he can do so because he knows k_1. Then, he can compare these two leakages with the one received from his test query, and decide which is the correct one.

This strategy works even when nonce-based variants of these schemes are used, as the nonces are always returned as part of the ciphertext, and the adversary can therefore use them as part of his leakage matching attack.

These attacks are made easy through two main aspects:

(1) Long term keys are reused on each message block, which supports DPA attacks. This can be avoided by using LR operation modes, which use re-keying strategies to limit the number of leakages on any specific secret.
(2) Keyed functions are applied to values that are known to the adversary, which makes leakage matching attacks easy.

These suggests important ingredients for the design of CIML secure schemes.

4.2 Combining LR MAC and encryption modes

Given that the LR part is problematic in the MR AE constructions described above, one could be tempted to build

CIML secure schemes in the opposite direction, that is starting from LR primitives and turning them into a MR AE scheme, in the hope that CIML security will follow.

This is however not necessarily the case, as we demonstrate now from the combination of recent constructions of LR MACs and encryption schemes from Pereira et al. [37], which we call PSV-MAC and PSV-ENC.

PSV-MAC and PSV-ENC are based on two block-ciphers F and F^*, with the distinction that F is assumed to be cheap and efficiently implemented but leaking, while F^* is assumed to be an expensive (in terms of power and speed), heavily protected, and leak-free component. In other words, formally F^* is just a standard PRP without leakage while F is a leaking PRP. The purpose of this distinction is to design schemes that make minimal use of the expensive F^*: one or two calls per message to be encrypted, independently of the number of blocks of the message, while the bulk of the computation is performed by the cheap F.

We note that making a distinction between F and F^* would make little sense in the case of the SIV constructions above, because all message blocks are treated with the long-term keys, so that it would only be helpful to process all blocks using the expensive F^*.

$\mathsf{PSV\text{-}MAC}_k(IV, m)$ is an IV-based MAC, and is evaluated as follows if $m = m_1 \| \cdots \| m_\ell$:

- $k_0 \leftarrow \mathsf{F}^*_k(IV)$
- $k_i \leftarrow \mathsf{F}_{k_{i-1}}(m_i)$, $\forall i \in [1, \ell]$
- return $\tau \leftarrow k_\ell$

$\mathsf{Vrfy}_k(IV, m, \tau)$ proceeds in the natural way.

PSV-ENC is an IV-based encryption scheme, which we will be using in the next sections as well. Its description is available in Figure 1.

PSV-ENC
$\mathsf{Enc}_k(IV, m)$, where $m = m_1 \| \cdots \| m_\ell$ 1. $k_0 \leftarrow \mathsf{F}^*_k(IV)$ 2. $\forall i \in [1, \ell] : k_i \leftarrow \mathsf{F}_{k_{i-1}}(p_A)$, $y_i \leftarrow \mathsf{F}_{k_{i-1}}(p_B)$, $c_i \leftarrow y_i \oplus m_i$, where p_A, p_B are public and distinct constants 3. return $C = c_1 \| c_2 \| \cdots \| c_\ell$
$\mathsf{Dec}_k(IV, C)$ proceeds in the natural way

Figure 1: The PSV-ENC encryption scheme.

Based on our findings in the analysis of SIV, we build our MR AE scheme using a slightly different construction, which we call DIV, for "double IV". DIV takes an IV-based MAC $\mathsf{ivM} = (\mathcal{K}, \mathsf{Mac}, \mathsf{Vrfy})$ and an IV-based encryption scheme $\mathsf{ivE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$, and produces a scheme $\mathsf{AE}_{\mathsf{DIV}} = (\mathcal{K}^2, \mathsf{DIV.Enc}, \mathsf{DIV.Dec})$, defined as follows:

- $\mathsf{DIV.Enc}_{k_M, k_E}(IV, m)$ returns $\tau \leftarrow \mathsf{Mac}_{k_M}(IV, m)$ and $c \leftarrow \mathsf{Enc}_{k_E}(\tau, (IV, m))$.
- $\mathsf{DIV.Dec}_{k_M, k_E}(\tau, c)$ computes $(IV, m) \leftarrow \mathsf{Dec}_{k_E}(\tau, c)$ and returns m if $\mathsf{Vrfy}_{k_M}(IV, m, \tau)$ succeeds. The error symbol $\perp$ is returned otherwise.

The differences with SIV are:

- the use of an IV-based MAC, which offers the possibility to perform re-keying already in the MAC part of the computation (as in PSV-MAC for instance), and
- the encryption of the IV used in the MAC, which mitigates the leakage matching attacks described for SIV.

In terms of efficiency, and compared to SIV, DIV requires to encrypt one more block (the IV), but does not increase the size of the ciphertext. We show, in the full version of this paper, that $\mathsf{AE}_{\mathsf{DIV}}$ is a MR AE as long as (1) ivM is chosen-IV pseudorandom (2) ivE is IV-sec-secure [10].

When considering the relative costs of F and F^*, the difference of efficiency becomes considerably more important. If we assume that F has a cost a and F^* a cost b, then the DIV composition applied to PSV-MAC and PSV-ENC comes at a cost of $(3\ell + 2)a + 2b$ for a message of ℓ blocks, and the SIV mode requires $(2\ell + 1)b$ with each block protected. If we assume that $b = 100a$ (which is consistent with the table given in [37]), then the DIV composition is already cheaper for a single message block and, for long messages, the DIV composition will be ≈ 67 times cheaper.

Given that PSV-MAC and PSV-ENC satisfy these security notions and are LR taken individually, one may hope that $\mathsf{PSV\text{-}AE} = \mathrm{DIV}(\mathsf{PSV\text{-}MAC}, \mathsf{PSV\text{-}ENC}) = (\mathcal{K}^2, \mathsf{PSVAEnc}, \mathsf{PSVADec})$ would offer CIML security. This is unfortunately not the case.

A CIML attacker can proceed as follows. First, select a random IV, and query $\mathsf{PSVAEnc}_{k_M, k_E}(IV, m)$ with various messages of $\ell > 1$ blocks. Keeping IV constant ensures that the same $k_0 \leftarrow \mathsf{F}^*_{k_M}(IV)$ is computed every time, and to mount a DPA attack that recovers k_0 when $\mathsf{F}_{k_0}(m_1)$ is computed with the first block of each message.

The rest of the attack is similar to the one against SIV. Let $(\tau, C) \leftarrow \mathsf{PSVAEnc}_{k_M, k_E}(IV, m)$ for the same IV as above and a chosen ℓ block message m. The adversary can (i) select $\ell - 1$ blocks $m^*_1, \cdots, m^*_{\ell-1}$ that were not part of a previous encryption query, (ii) define $k^*_0 = k_0$, and compute $k^*_i = F_{k^*_{i-1}}(m^*_i)$ for $i \in [1, l-1]$, (iii) compute $m^*_\ell = F^{-1}_{k'_{\ell-1}}(\tau)$. This guarantees that (IV, m) and $(IV, m^* = m^*_1 \| \ldots \| m^*_\ell)$ have the same MAC τ. Now, the adversary can define $C^* = C \oplus (IV \| m) \oplus (IV \| m^*)$ and return (τ, C^*) as a fresh ciphertext that decrypts to m^*, hence violating the CIML security.

In the full version of this paper, we additionally show that the alternative LR MAC proposed in [37] could be broken by similar (though slightly more elaborate) attacks [10].

These negative results for CIML security lead us to the design of new schemes, which we present and analyze in the following sections.

5 DIGEST, TAG AND ENCRYPT

In this section, we build a MR AE scheme that provably achieves CIML security in a very permissive leakage model, which we call the unbounded leakage model. As previously mentioned, the confidentiality analysis with leakage is deferred to Section 7 (and will have to rely on a less permissive leakage assumption).

5.1 Specification of DTE

The attacks against the CIML property in the previous section essentially result from the fact that revealing a (long-term or ephemeral) key is sufficient to break the collision resistance of the tags. Hence, a natural way to address them is to combine PSV-ENC with a modified IV-based MAC that first hashes the IV with the plaintext and then runs a leak-free PRF on the hash value to compute the tag τ (see Figure 2, where long-term secrets are in red, ephemeral ones in orange and public values in green).

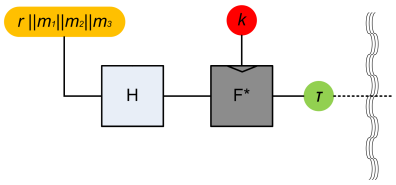

Figure 2: DTE leakage-resilient AE: Authentication part.

Then, PSV-ENC encrypts both the IV and the plaintext using the tag as its own IV (see Figure 3). As a result our scheme produces a digest from its input, generates the tag from it, and encrypts, hence the name DTE.

Note that our presentation explicitly uses a hash function here, because we need collision resistance and preimage resistance. However, from an implementation point of view, it is not necessary to use a different functionality, and a hash function based on the block cipher F could be used.

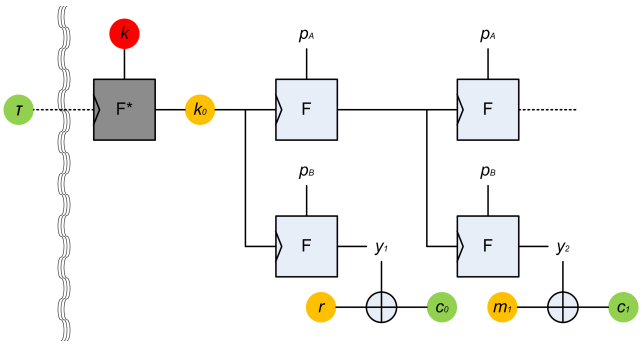

Figure 3: DTE leakage-resilient AE: Encryption part (PSV-ENC).

The full description of DTE is given in Figure 4. As before, the values p_A and p_B are two public distinct constants in $\mathcal{B} = \{0,1\}^n$. The key k is drawn at random over $\mathcal{K}$ as usual. In order to stress that the IV used in the MAC part of the scheme is not public, we refer to it with the letter r.

We point that DTE is the result of the DIV composition (see Section 4.2) applied to the IV-based MAC of Figure 2 and PSV-ENC, with the important difference that the same key k is used in the authentication part as well as in the encryption part (which either reduces the length of the key, or avoids a key expansion step).

DTE

$\mathsf{Enc}_k(r, m)$, where $m = (m_1, m_2, \ldots, m_\ell)$:
- $h \leftarrow \mathsf{H}(r||m)$ // digest
- $\tau \leftarrow \mathsf{F}^*_k(h)$ // tag
- $k_0 \leftarrow \mathsf{F}^*_k(\tau)$ // ...and encrypt
- $c_0 \leftarrow \mathsf{F}_{k_0}(p_B) \oplus r$ // $y_i := \mathsf{F}_{k_i}(p_B)$
- $k_i \leftarrow \mathsf{F}_{k_{i-1}}(p_A)$, $c_i \leftarrow \mathsf{F}_{k_i}(p_B) \oplus m_i$ $\forall i \in [1, \ell]$
- return $C \leftarrow (\tau, c_0, c_1, c_2, \ldots, c_\ell)$

$\mathsf{Dec}_k(C)$, where $C = (\tau, c_0, c_1, c_2, \ldots, c_\ell)$:
- $k_0 \leftarrow \mathsf{F}^*_k(\tau)$
- $r \leftarrow \mathsf{F}_{k_0}(p_B) \oplus c_0$
- $k_i \leftarrow \mathsf{F}_{k_{i-1}}(p_A)$, $m_i \leftarrow \mathsf{F}_{k_i}(p_B) \oplus c_i$ $\forall i \in [1, \ell]$
- $h \leftarrow \mathsf{H}(r||m)$
- if $\tau = \mathsf{F}^*_k(h)$ return $(m_1, ..., m_\ell)$, else return $\perp$.

Figure 4: DTE - Full description.

5.2 Efficiency of DTE

Ignoring the cost of the protection against leakages, the computational costs of DTE are increased by 50% compared to the original MR AE SIV scheme of Rogaway and Schrimpton: both perform two passes on the message, but DTE's iterations are slightly more expensive due to the leakage-resilient encryption, which requires two block cipher executions per message block instead of one. However, as soon as leakage is included in the game, protecting the SIV MR AE scheme would require to have all block cipher executions equally well protected (i.e., as F^*), since they all compute with the same long-term key. Denoting the overhead factor of F^* compared to F by α and the number of blocks to be encrypted by ℓ, this roughly implies an approximate cost of $2\alpha + 3\ell$ for DTE and $2\alpha\ell$ for SIV. This means that the encryption cost of DTE is favorable against the one of SIV as soon as we need to encrypt $\ell \geq 2$ message blocks and $\alpha \geq 3$, and that the gain will tend to $2\alpha/3$ when ℓ increases. So, given that α typically ranges from hundreds to thousands (as discussed in the introduction), the performance of an implementation of DTE is expected to gradually outperform SIV by two or three orders of magnitude when the size of the messages increases, if security against side-channel attacks has to be guaranteed. Similar improvement factors can be obtained when comparing with recent improvements/refinements on the original SIV scheme, and similar gains will also be obtained for the DCE scheme that will be discussed in the next section.

5.3 Misuse resistance without leakage

As a first security analysis we show that DTE is a MR AE, in the sense of Definition 2.9.

THEOREM 5.1. *Let* $\mathsf{H} : \{0,1\}^n \times \{0,1\}^\star \to \{0,1\}^n$ *be a* $(0, t_1, \varepsilon_{cr})$*-collision resistant and* $(0, t_1, \varepsilon_{pr})$*-range-oriented preimage resistant hash function. Let* $\mathsf{F}^\star : \{0,1\}^n \times \{0,1\}^n \to$

$\{0,1\}^n$ *be a* $(2q, t_1, \varepsilon_{\mathsf{F}^\star})$*-pseudorandom function and* $\mathsf{F} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ *be a* $(2, t_2, \varepsilon_{\mathsf{F}})$*-pseudorandom function. Then the* DTE *authenticated encryption scheme which encrypts* ℓ*-block messages is* (q, t, ε)*-MR as long as* $t \leq \min\{t_1 - q(t_{\mathsf{H}} + (2\ell+1)t_{\mathsf{F}}), t_2 - q_e(t_{\mathsf{H}} + (2\ell+1)t_{\mathsf{F}})\}$ *with* $0 \leq q_e + q_d \leq q$*, where* q_e *(resp.* q_d*) is the number of encryption (resp. decryption) queries, where* t_{H} *and* t_{F} *are the time needed to evaluate* H *and* F*, and we have*

$$\varepsilon \leq \varepsilon_{\mathsf{F}^\star} + \varepsilon_{cr} + 2q \cdot \varepsilon_{pr} + q(\ell+1) \cdot \varepsilon_{\mathsf{F}} + \left(q_d + q_e^2 + q_e^2(\ell+1)^2\right) \cdot 2^{-n}. \quad (1)$$

The guideline of the proof is as follows: first, we start by arguing that all decryption queries can be answered by $\perp$. Then, proceeding block by block, we gradually show that the answers to encryption queries can be replaced by random outputs.

The easiest transition relies on the pseudorandomness of F^*, which is replaced by a truly random function f. Therefrom, we can move to show the invalidity of the first fresh decryption query $C = (\tau, c)$, where $c = (c_0, c_1, \ldots, c_\ell)$. Since (τ, c) is fresh, we will see that the decrypted tuple $(r, m = (m_1, \ldots, m_\ell))$ is fresh. Thereby, the collision resistance ensures that $h = \mathsf{H}(r\|m)$ is not the output of any previous evaluation of H during the encryption queries. If h never appeared until the first decryption query, then $\mathsf{f}(h) \neq \tau$ except by chance. However, we must also consider the event by which $h = \tau'$, where τ' is the returned tag associated to a previous fresh encryption query. Hence the need of the range-oriented preimage resistance of H since $\tau' = f(h')$ is random over $\{0,1\}^n$, for some $h' \neq h$. As a side note on the proof, the unlikelihood of $h = \tau'$ also plays an important role to ensure the random-looking of the ciphertexts. Indeed, if the adversary managed to query an encryption on the a pair (r, m) such that $\tau' = \mathsf{H}(r\|m)$, the answer (τ, c) of the (modified) encryption oracle would reveal the ephemeral key $k'_0 = \tau = f(h')$ of the ciphertext containing the tag τ'.
The proof of Theorem 5.1 is available in the full version of the paper [10].

5.4 The Unbounded Leakage Model

Before turning to our proof of the CIML security of DTE, we need to introduce a leakage model. For this proof, we can adopt a very permissive leakage model, which we call the unbounded leakage model.

Definition 5.2. 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 encryption 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, since r and m are known to the adversary anyway, and all the internal variables can be recomputed from k_0. Indeed, given k_0 the adversary is able to derive all the ephemeral keys $(k_1, \ldots, k_\ell)$ used during each encryption query, which in turns gives him all the $(y_0, y_1, \ldots, y_\ell)$ values from Fig. 3.

5.5 CIML Security of DTE

We now prove that DTE satisfies the CIML notion in the unbounded leagage model.

Theorem 5.3. *Let* $\mathsf{H} : \{0,1\}^n \times \{0,1\}^\star \to \{0,1\}^n$ *be a* $(0, t', \varepsilon_{cr})$*-collision resistant and* $(0, t', \varepsilon_{pr})$*-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 *which encrypts* ℓ*-block messages provides* (q, t, ε)*-CIML security in the unbounded leakage model 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*

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

An interesting observation about this statement is that it shows that the pseudorandomness of F has no impact on the success probability of the CIML adversary.

The proof of Theorem 5.3 is given in Appendix A.

6 DIGEST, COMMIT AND ENCRYPT

The previous construction reaches different types of guarantees with and without leakages (namely, MR and CIML). As a (more balanced) alternative, we now present a construction that drops the requirement of MR (without leakage), and only focuses on CIML security. This construction has the advantage of only requiring one execution of the leak-free function, but at the expense of relying on the random oracle model in its proof of CIML security (yet not for its LR CPA security, as will be shown in Section 7).

We acknowledge that the use of a random oracle when analyzing implementation weaknesses is questionable, since the random oracle is an abstraction, and therefore does not offer a simple model for physical leakages.

In order to overcome this difficulty, we assume that the random oracle has unbounded leakages: it leaks all of its inputs and outputs. Of course, such a leakage model would be too strong for proving any confidentiality property of a construction that would hash a secret value. But we show that, even in a such a strong leakage model, CIML can be achieved.

Besides, and as discussed in [47], the random oracle model sometimes comes in handy in order to argue about the security of natural constructions of which the LR seems hard to reach in the standard model. In view of the practical interest of the DCE construction, we therefore include a proof in this model in our treatment and suggest the further investigation of DCE instances as an interesting scope for further research. We note that our proof does not make use of the programmability of the random oracle, which is a common source of gaps in the

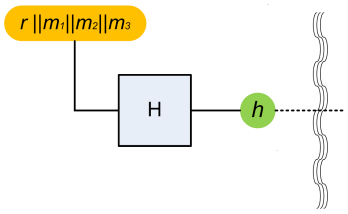

Figure 5: DCE leakage-resilient AE (part I). Part II is identical to Fig. 3.

DCE

$\mathsf{Enc}_k(r, m)$, where $m = (m_1, m_2, \ldots, m_\ell)$:
- $\mathsf{h} \leftarrow \mathsf{H}(r\|m)$
- $k_0 \leftarrow \mathsf{F}^*_k(\mathsf{h})$
- $c_0 \leftarrow \mathsf{F}_{k_0}(p_B) \oplus r$
- $k_i \leftarrow \mathsf{F}_{k_{i-1}}(p_A)$, $c_i \leftarrow \mathsf{F}_{k_i}(p_B) \oplus m_i$ $(\forall i = 1, \ldots, \ell)$
- return $C = (\mathsf{h}, c_0, c_1, c_2, \ldots, c_\ell)$

$\mathsf{Dec}_k(C)$, where $C = (\mathsf{h}, c_0, c_1, c_2, \ldots, c_\ell)$:
- $k_0 \leftarrow \mathsf{F}^*_k(\mathsf{h})$
- $r \leftarrow \mathsf{F}_{k_0}(p_B) \oplus c_0$
- $k_i \leftarrow \mathsf{F}_{k_{i-1}}(p_A)$, $m_i \leftarrow \mathsf{F}_{k_i}(p_B) \oplus c_i$ $(\forall i = 1, \ldots, \ell)$
- if $\mathsf{h} = \mathsf{H}(r\|m)$ return $m = (m_1, ..., m_\ell)$, else return $\perp$.

Figure 6: The DCE scheme.

soundness of schemes that are proven to be secure in this model but are insecure for any instantiation of the random oracle.

6.1 Specifications

The authentication part of the DCE scheme is outlined in Figure 5 which is then plugged to the encrypting part of Figure 3. The full specification is available in Figure 6. There, H is a hash function and p_A and p_B are two public distinct constants in $\mathcal{B} = \{0,1\}^n$. The key k is picked randomly from $\mathcal{K}$, as usual.

6.2 Security analysis

Theorem 6.1. *Let* $\mathsf{H} : \{0,1\}^n \times \{0,1\}^\star \longmapsto \{0,1\}^n$ *be modeled as a random oracle. Let* $\mathsf{F}^* : \{0,1\}^n \times \{0,1\}^n \longmapsto \{0,1\}^n$ *be* $(q+1, t', \varepsilon_{\mathsf{F}^*})$*-pseudorandom. Then,* DCE *provides* (q, t, ε)*-CIML security in the unbounded leakage model, where* $t \leq t' - (q+1)(t_{\mathsf{H}} + (2\ell+1)t_{\mathsf{F}})$*,* ℓ *is the number of blocks of the encrypted messages,* t_{H} *and* t_{F} *are the time needed to evaluate* H *and* F*, and we have*

$$\varepsilon \leq \varepsilon_{\mathsf{F}^*} + 4(q+1)^2/2^n + (q+1)/2^n.$$

The proof of Theorem 6.1 is given in Appendix B. The CPA security of DCE without leakage (or misuse) in the random oracle model is immediate.

7 LEAKAGE-RESILIENT CPA SECURITY

The ciphertext integrity properties discussed in the previous sections do not imply anything about the confidentiality of the messages that are encrypted with DTE and DCE.

This section shows the leakage-resilient CPA security of these schemes, which is measured by the probability that an adversary distinguishes between playing the $\mathsf{PrivK}^{\mathsf{lmcpa},0}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}$ and $\mathsf{PrivK}^{\mathsf{lmcpa},1}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}$ games, defined in Figure 7 and borrowed from PSV [37]. This is essentially the traditional CPA game, with the addition that the challenger provides leakages for any computation it performs, including the test query at Step 3), and that the adversary can access a leakage oracle L that gives him leakages from the attacked circuit on chosen inputs (which makes it possible to run matching attacks, as described in Section 4). This oracle was formally omitted in the previous sections, as it was meaningless in the unbounded leakage model.

We recall that the lmcpa superscript in the notation $\mathsf{PrivK}^{\mathsf{lmcpa},0}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}$ stands for multiple messages and blocks leakage-resilient CPA security, which relates to the remark in introduction that our following proofs only guarantee that the security of our constructions for multiple messages and blocks reduces to their security for one block, and then depends on what can be guaranteed for this single block. As discussed in [37], this is the best that can be achieved given the impossibility of leakage-resilient CPA security with negligible advantage (due the the fact that even a single bit of plaintext leakage trivially breaks the semantic security game).

$\mathsf{PrivK}^{\mathsf{lmcpa},\mathsf{b}}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}$, with $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$, is the output of the following experiment:
(1) Select $k \xleftarrow{\$} \mathcal{K}$
(2) $\mathcal{A}^{\mathsf{L}}$ gets access to a leaking encryption oracle that, when queried on a message m of arbitrary block length, returns $\mathsf{Enc}_k(m)$ together with the leakage resulting from the encryption process.
(3) $\mathcal{A}^{\mathsf{L}}$ submits two messages m_0 and m_1 of identical block length, to which he is replied with $\mathsf{Enc}_k(m_b)$ and the corresponding leakage.
(4) $\mathcal{A}^{\mathsf{L}}$ can keep accessing the leaking encryption oracle.
(5) $\mathcal{A}^{\mathsf{L}}$ outputs a bit b'.

Figure 7: The $\mathsf{PrivK}^{\mathsf{lmcpa},\mathsf{b}}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}$ game

The $\mathsf{PrivK}^{\mathsf{leav},\mathsf{b}}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}$ game [37], modeling leakage-resilient eavesdropper security, is defined just in the same way, except that the encryption oracles from steps 2 and 4 disappear.

Definition 7.1. An AE scheme $\mathsf{AE} = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ with leakage function L is (q, t, ϵ) lmcpa-secure (resp. leav-secure) if, for every (q, t)-bounded adversary $\mathcal{A}^{\mathsf{L}}$, the lmcpa (resp. leav) advantage $|\mathsf{PrivK}^{\mathsf{lmcpa},0}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}} - \mathsf{PrivK}^{\mathsf{lmcpa},1}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}|$ (resp. $|\mathsf{PrivK}^{\mathsf{leav},0}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}} - \mathsf{PrivK}^{\mathsf{leav},1}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}|$) is bounded by ϵ.

7.1 Background: LMCPA security of PSV-ENC

Observing that the encryption part of all our schemes essentially follows the PSV-ENC scheme, we can hope to import the results of the previous analyzes of that scheme.

The security of an implementation of the PSV-ENC scheme relies on the assumption that the block cipher implementation that it uses has 2-simulatable leakages.

The notion of simulatable leakages is based on the q-sim-game below, from which q-simulatable leakages are defined. This game essentially measures the capability of a simulator to produce leakages that look consistent with given inputs and outputs of a block cipher, without knowing the key used in the computation.

Game q-$\mathsf{sim}(\mathcal{A}, \mathsf{F}, \mathsf{L}, \mathcal{S}, b)$ [43, Section 2.1].		
The challenger selects two random keys $k, k^* \xleftarrow{\$} \mathcal{K}$. The output of the game is a bit b' computed by $\mathcal{A}^{\mathsf{L}}$ based on the challenger responses to a total of at most q adversarial queries of the following type:		
Query	Response if $b = 0$	Response if $b = 1$
$\mathsf{Enc}(x)$	$\mathsf{F}_k(x)$, $\mathsf{L}(k, x)$	$\mathsf{F}_k(x)$, $\mathcal{S}^{\mathsf{L}}(k^*, x, \mathsf{F}_k(x))$
and one query of the following type:		
Query	Response if $b = 0$	Response if $b = 1$
$\mathsf{Gen}(z, x)$	$\mathcal{S}^{\mathsf{L}}(z, x, k)$	$\mathcal{S}^{\mathsf{L}}(z, x, k^*)$

Definition 7.2. [q-simulatable leakages [43, Def. 1]] Let F be a PRF having leakage function L. Then F has $(q_{\mathcal{S}}, t_{\mathcal{S}}, q_{\mathcal{A}}, t_{\mathcal{A}}, \epsilon_{q\text{-}sim})$ q-simulatable leakages if there is a $(q_{\mathcal{S}}, t_{\mathcal{S}})$-bounded simulator $\mathcal{S}^{\mathsf{L}}$ such that, for every $(q_{\mathcal{A}}, t_{\mathcal{A}})$-bounded adversary $\mathcal{A}^{\mathsf{L}}$, we have

$$|\Pr[q\text{-}\mathsf{sim}(\mathcal{A}, \mathsf{F}, \mathsf{L}, \mathcal{S}^{\mathsf{L}}, 1) = 1] - \Pr[q\text{-}\mathsf{sim}(\mathcal{A}, \mathsf{F}, \mathsf{L}, \mathcal{S}^{\mathsf{L}}, 0) = 1]| \leq \epsilon_{q\text{-}sim}.$$

Based on this definition, the eavesdropper security of PSV-ENC can be summarized as follows.

Theorem 7.3 ([37], Thm 3.). *Let F be a $(q, t, \epsilon_{\mathsf{F}})$-PRF whose implementation has running time t_{F} and a leakage function L_{F} with $(q_{\mathcal{S}}, t_{\mathcal{S}}, q, t, \epsilon_{2\text{-}sim})$ 2-simulatable leakages.*

The advantage of every $(q - q_r, t - t_r)$-bounded $\mathcal{A}^{\mathsf{L}_{\mathsf{F}}}$ playing the $\mathsf{PrivK}^{\mathsf{leav},\mathsf{b}}_{\mathsf{PSV-ENC}}$ game is bounded by $\epsilon^{\mathsf{eav}}_{\mathsf{PSV-ENC}} = \ell(\mathbf{Adv}_s + 4(\epsilon_{\mathsf{F}} + \epsilon_{2\text{-}sim}))$ where $\mathbf{Adv}_s$ is a bound on the eavesdropper advantage of a $(q - q_{r'}, t - t_{r'})$-bounded adversary trying to distinguish the encryptions of two single-block messages encrypted with the PSV-ENC scheme, q_r, $q_{r'}$ are $\mathcal{O}(\ell q_{\mathcal{S}})$ and $t_r, t_{r'}$ are $\mathcal{O}(\ell(t_{\mathcal{S}} + t_{\mathsf{F}}))$.

This result relates the eavesdropper security of the PSV-ENC scheme to the security that is offered in front of an adversary who can only get a single encryption of a single block messages, which is expected to be simpler to evaluate (see discussion in [37]). Note that, in our analysis below, we will not need to use any result about the CPA security of PSV-ENC.

7.2 Bounding hash function leakages

The security of the PSV-ENC scheme is going to be helpful for the encryption part of the DTE and DCE modes, but the first parts of our modes also include the evaluation of a hash function running on the message to be encrypted, which may in turn leak information about the message and help win the $\mathsf{PrivK}^{\mathsf{lmcpa},\mathsf{b}}_{\mathcal{A}^{\mathsf{L}},\mathsf{AE}}$ game: if the implementation of the hash function just leaks its input in full, we can obviously not hope for any confidentiality. We therefore turn to the definition of our security assumption about the hash function implementation, before analyzing DCE and DTE.

Concretely, we need a bound on the distinguishing probability of an adversary who would see the leakages resulting from hashing something containing a message m_0 and those resulting from hashing something containing m_1. Simply assuming the indistinguishability of leakages on adversarially chosen m_0 and m_1 would be way too strong from a physical point-of-view: if an adversary knows m_0 and m_1, he can obtain leakages computed on these two values directly from the hash function implementation, and compare those leakages with the leakage returned by the challenger, in a leakage matching attack.

However, our adversary faces a more difficult problem, since he is not able to predict what message is hashed when he gets leakages to distinguish. More precisely, the adversary may be able to choose 2 messages m_0 and m_1, but must then decide the value of b when he gets $\mathsf{H}(r\|m_b), \mathsf{L}_{\mathsf{H}}(r\|m_b)$ in return, where r is a fresh random value and $\mathsf{L}_{\mathsf{H}}(x)$ is the leakage resulting from evaluating the hash function on x. Since DTE and DCE encrypt $(r\|m_b)$ with PSV-ENC, r is unknown to the adversary, and he cannot feed his device with $(r\|m_0)$ or $(r\|m_1)$ in order to match the leakages, and is bound to run a more sophisticated SPA attack due to the partially unknown state.

The DCE and DTE schemes also hash $(r\|m_b)$ and not $(m_b\|r)$. While equivalent in theory, this makes sure that, when using an iterating hash function, the block containing the randomness r is processed before the blocks containing the message. This again prevents the adversary from performing a matching attack on the first block of the hash function implementation only, because that first block will already have an unknown input, and will in turn make unknown the inputs of all further blocks.

These observations lead us to the following definition.

Definition 7.4. A hash function $\mathsf{H} : \mathcal{R} \times \mathcal{M} \to \mathcal{B}$ with leakage function L is (q, t, ϵ)-leakage-resilient if, for every (q, t)-bounded adversary $\mathcal{A}^{\mathsf{L}}$, the advantage $\left|\mathsf{Hash}^0_{\mathcal{A}^{\mathsf{L}},\mathsf{H}} - \mathsf{Hash}^1_{\mathcal{A}^{\mathsf{L}},\mathsf{H}}\right|$ is bounded by ϵ, where $\mathsf{Hash}^b_{\mathcal{A}^{\mathsf{L}},\mathsf{H}}$ is defined as the probability

that $\mathcal{A}^{\mathsf{L}}$ outputs 1 when, after a query $(m_0, m_1) \in \mathcal{M}^2$, he is returned with the pair $(\mathsf{H}(r\|m_b), \mathsf{L}(r\|m_b))$ with $r \xleftarrow{\$} \mathcal{R}$.

Based on these definitions of leakage resilient PRF and hash function, the following section shows the LMCPA security of DCE and DTE. Admittedly, these results should be understood similarly to the ones in [37], where it was argued that semantic security *with negligible advantage* is impossible to achieve even if the leakage of an encryption would be as low as a single message bit (in contrast, leaking a bit of the secret key may not be an issue). So informally, what we show next is that the execution of our leakage-resilient authentication scheme for many messages does not significantly degrade the security compared to the situation with a single message, and that the security degradation resulting from the encryption of a long multi-block message is not significantly worse than if this message had been encrypted block by block, with fresh independent keys for each block. Concretely though, it always remains that manipulating the message leaks some information that can be exploited via SPA (as just explained), because of the initial hashing of Figures 2 and 5 and the stream encryption of Figure 3.

7.3 LMCPA Security of the DCE and DTE schemes

We start by focusing on the LMCPA security of the DCE scheme. The leakage function $\mathsf{L}(k, r, m)$ for DCE is defined by the pair $(\mathsf{L}_{\mathsf{H}}(r, m), \mathsf{L}_{\mathsf{PSV}}(k_0, r\|m))$, where k_0 is defined as $\mathsf{F}_k(\mathsf{H}(r\|m))$. The L_{H} component of this leakage contains the leakage occurring during the evaluation of the hash function in DCE encryption, and the $\mathsf{L}_{\mathsf{PSV}}$ component contains the leakage of the encryption part of the DCE as depicted in Figure 3, which we refer to as the "PSV-encryption component" of DCE. The $\mathsf{L}_{\mathsf{PSV}}$ function itself returns leakages that are made of individual leakages by each PRF and XOR operation, as defined in [37], but this is irrelevant for our analysis.

THEOREM 7.5. *Let* $\mathsf{H} : \mathcal{R} \times \mathcal{M} \to \mathcal{B}$ *be a* $(0, t, \epsilon_{cr})$*-collision resistant and* $(q, t, \epsilon_{\mathsf{L}_{\mathsf{H}}})$*-leakage-resilient hash function. Let* F *be a* $(q, t, \epsilon_{\mathsf{F}})$*-pseudorandom function. Let* DCE *be implemented with a PSV-encryption component that is* $(q, t, \epsilon^{\mathsf{leav}}_{\mathsf{PSV}-\mathsf{ENC}})$*-*leav *secure.*

Then, the DCE *scheme with the leakage function* L *described above is* $(q', t', \epsilon_{\mathsf{lmcpa}})$*-secure. Here:* $q' \leq q - q_e - 1$ *where* q_e *is the number of encryption queries made by the* (q', t')*-bounded LMCPA adversary;* $t' \leq t_1 - t_c - t_{sc}$*, where* t_c *is the running time needed to run the LMCPA challenger in front of a* (q, t')*-bounded adversary,* t_{sc} *is the time needed to determine whether a list of* q_e *hash values contains a collision; and* $\epsilon_{\mathsf{lmcpa}} \leq 2\frac{q_e^2}{|\mathcal{R}|} + 2\epsilon_{cr} + 4\epsilon_{\mathsf{F}} + \epsilon_{\mathsf{L}_{\mathsf{H}}} + \epsilon^{\mathsf{leav}}_{\mathsf{PSV}-\mathsf{ENC}}$.

The proof of Theorem 7.5 is given in Appendix C.

The leakage-resilient CPA security of the DTE scheme can be shown in an almost identical way.

THEOREM 7.6. *Let* $\mathsf{H} : \mathcal{R} \times \mathcal{M} \to \mathcal{B}$ *be a* $(0, t, \epsilon_{cr})$*-collision resistant and* $(q, t, \epsilon_{\mathsf{L}_{\mathsf{H}}})$*-leakage-resilient hash function. Let* F *be a* $(2q, t, \epsilon_{\mathsf{F}})$*-pseudorandom function. Let* DTE *be implemented with a PSV-encryption component that is* $(q, t, \epsilon^{\mathsf{leav}}_{\mathsf{PSV}-\mathsf{ENC}})$*-*leav *secure.*

Then, the DTE *scheme with the leakage function* L *described above is* $(q', t', \epsilon_{\mathsf{lmcpa}})$*-secure. Here:* $q' \leq q - q_e - 1$ *where* q_e *is the number of encryption queries made by the* (q', t')*-bounded LMCPA adversary;* $t' \leq t_1 - t_c - t_{sc}$*, where* t_c *is the running time needed to run the LMCPA challenger in front of a* (q', t')*-bounded adversary,* t_{sc} *is the time needed to determine whether a list of* q_e *hash values contains a collision; and* $\epsilon_{\mathsf{lmcpa}} \leq 2\frac{q_e^2}{|\mathcal{R}|} + 4\frac{(q_e+1)^2}{|\mathcal{B}|} + 2\epsilon_{cr} + 4\epsilon_{\mathsf{F}} + \epsilon_{\mathsf{L}_{\mathsf{H}}} + \epsilon^{\mathsf{leav}}_{\mathsf{PSV}-\mathsf{ENC}}$.

The proof shares almost all features of the one for the DCE scheme, and the handling of adversarial queries is the same. The double use of F_k just loosens the bounds of Thm. 7.5 by constant factors, by increasing the probability of collisions and doubling the number of queries that are needed when replacing the evaluation of F with the selection of random values (which is included in the t_c bound on the challenger running time). It is given in Appendix D.

8 CONCLUSIONS

To conclude this paper, we first observe that our analyzes focus on the leakages occurring during AE, so far excluding the possibility to target a decryption device. Interestingly, this limitation is very strong in the LR MAC and encryption schemes of [37] because random IVs are strictly needed for LR security, and a decryption oracle allows the adversary to control the IV. As discussed in Section 4, contradicting this requirement directly enables devastating forgery attacks based on a standard DPA. By contrast, our notion of CIML aims at mitigating the impact of IV control. So it is natural to investigate whether it formally rules out any attack against the decryption oracle. Unfortunately, and despite CIML security indeed rules out many realistic attacks against a decryption device, our schemes remain susceptible to strong attacks when the decryption leaks. Taking the case of DTE, we can for example show that it is possible to forge valid ciphertexts thanks to decryption leakages as follows: (1) Pick a random r and message m and compute $h = \mathsf{H}(r\|m)$. (2) Ask decryption of ciphertext $C^i = (\tau, c^i)$ with $\tau = h$ and a random c^i and recover k_0 thanks to leakage. (3) Ask decryption of ciphertext $C^j = (\tau', c^j)$ with $\tau' = k_0$ and a random c^j and recover k_0' thanks to leakage. (4) From k_0', compute the ciphertext c produced using the encryption part of DTE from the ephemeral key k_0', the random r and the message m, so that $C = (k_0, c)$ is valid (and has decryption m). A similar attack can be performed against DCE. Note that this attack (which essentially exploits SPA to recover ephemeral keys) is admittedly more challenging than the standard DPAs in Section 4. Yet, it is also impossible to argue why such attacks should not be covered by our threat model. A formal treatment of CIML extended with decryption leakages has been recently proposed in [11].

We finally mention that our work is focused on CIML which is an integrity notion primarily intended for authentication (or the authentication part of an AE scheme). By contrast,

our treatment of LR CPA security so far excludes randomness misuse. The main reason for this separate treatment is that CIML can be achieved in a very liberal (mostly unbounded) leakage model. We leave the extension of these definitions towards a complete definition of LR MR AE as an interesting scope for further research.

Acknowledgments. 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 European Union (EU) through the ERC projects CRASH (280141) and SWORD (724725), the INNOVIRIS projects SCAUT and C-Cure, the ARC project NANOSEC and the European Union and Walloon Region FEDER USERMedia project 501907-379156.

REFERENCES

[1] M. R. Albrecht and K. G. Paterson. Lucky microseconds: A timing attack on amazon's s2n implementation of TLS. In *EUROCRYPT*, pages 622–643, 2016.
[2] M. R. Albrecht, K. G. Paterson, and G. J. Watson. Plaintext recovery attacks against SSH. In *S&P*, pages 16–26. IEEE Computer Society, 2009.
[3] E. Andreeva, A. Bogdanov, A. Luykx, B. Mennink, N. Mouha, and K. Yasuda. How to securely release unverified plaintext in authenticated encryption. In *ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 105–125. Springer, 2014.
[4] E. Andreeva and M. Stam. The symbiosis between collision and preimage resistance. In L. Chen, editor, *IMACC*, volume 7089 of *LNCS*, pages 152–171. Springer, 2011.
[5] J. Balasch, B. Gierlichs, O. Reparaz, and I. Verbauwhede. DPA, bitslicing and masking at 1 GHz. In T. Güneysu and H. Handschuh, editors, *CHES*, volume 9293 of *LNCS*, pages 599–619. Springer, 2015.
[6] G. Barwell, D. P. Martin, E. Oswald, and M. Stam. Authenticated encryption in the face of protocol and side channel leakage. In T. Takagi and T. Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 693–723. Springer, 2017.
[7] G. Barwell, D. Page, and M. Stam. Rogue decryption failures: Reconciling AE robustness notions. In *IMACC 2015*, volume 9496 of *LNCS*, pages 94–111. Springer, 2015.
[8] S. Belaïd, V. Grosso, and F. Standaert. Masking and leakage-resilient primitives: One, the other(s) or both? *Cryptography and Communications*, 7(1):163–184, 2015.
[9] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptology*, 21(4):469–491, 2008.
[10] F. Berti, F. Koeune, O. Pereira, T. Peters, and F. Standaert. Leakage-resilient and misuse-resistant authenticated encryption. *IACR Cryptology ePrint Archive*, 2016:996, 2016.
[11] F. Berti, O. Pereira, T. Peters, and F. Standaert. On leakage-resilient authenticated encryption with decryption leakages. *IACR Trans. Symmetric Cryptol.*, 2017(3):271–293, 2017.
[12] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. On the indifferentiability of the sponge construction. In *EUROCRYPT*, pages 181–197, 2008.
[13] A. Boldyreva, J. P. Degabriele, K. G. Paterson, and M. Stam. On symmetric encryption with distinguishable decryption failures. In *FSE 2013*, volume 8424 of *LNCS*, pages 367–390. Springer, 2013.
[14] CAESAR. Competition for authenticated encryption: Security, applicability, and robustness. https://competitions.cr.yp.to/caesar.html, 2012.
[15] S. Chari, J. R. Rao, and P. Rohatgi. Template attacks. In B. S. K. Jr., Ç. K. Koç, and C. Paar, editors, *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, 2002.
[16] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, and T. Unterluggauer. ISAP - towards side-channel secure authenticated encryption. *IACR Trans. Symmetric Cryptol.*, 2017(1):80–105, 2017.
[17] C. Dobraunig, F. Koeune, S. Mangard, F. Mendel, and F. Standaert. Towards fresh and hybrid re-keying schemes with beyond birthday security. In *CARDIS*, pages 225–241, 2015.
[18] T. Duong and J. Rizzo. Cryptography in the web: The case of cryptographic design flaws in ASP.NET. In *S&P*, pages 481–489. IEEE Computer Society, 2011.
[19] S. Dziembowski and K. Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008.
[20] S. Faust, K. Pietrzak, and J. Schipper. Practical leakage-resilient symmetric cryptography. In *CHES*, pages 213–232, 2012.
[21] B. Fuller and A. Hamlin. Unifying leakage classes: Simulatable leakage and pseudoentropy. In *ICITS*, pages 69–86, 2015.
[22] D. Gruss, R. Spreitzer, and S. Mangard. Cache template attacks: Automating attacks on inclusive last-level caches. In *USENIX Security*, pages 897–912, 2015.
[23] S. Gueron and Y. Lindell. GCM-SIV: full nonce misuse-resistant authenticated encryption at under one cycle per byte. In *CCS*, pages 109–119. ACM, 2015.
[24] V. T. Hoang, T. Krovetz, and P. Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In *EUROCRYPT*, volume 9056 of *LNCS*, pages 15–44. Springer, 2015.
[25] ISO/IEC 19772:2009. Information technology – security techniques – authenticated encryption. https://www.iso.org/standard/46345.html, 2009.
[26] J. Katz and M. Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In *FSE*, pages 284–299, 2000.
[27] J. Longo, D. P. Martin, E. Oswald, D. Page, M. Stam, and M. Tunstall. Simulatable leakage: Analysis, pitfalls, and new constructions. In *ASIACRYPT, Part I*, pages 223–242, 2014.
[28] J. Longo, E. D. Mulder, D. Page, and M. Tunstall. SoC it to EM: electromagnetic side-channel attacks on a complex system-on-chip. In *CHES*, pages 620–640, 2015.
[29] S. Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In *CT-RSA*, pages 222–235, 2004.
[30] S. Mangard, E. Oswald, and F. Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.
[31] D. P. Martin, E. Oswald, M. Stam, and M. Wójcik. A leakage resilient mac. In *IMACC 2015*, pages 295–310, 2015.
[32] M. Medwed, F. Standaert, J. Großschädl, and F. Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In *AFRICACRYPT*, pages 279–296, 2010.
[33] S. Micali and L. Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.
[34] C. Namprempre, P. Rogaway, and T. Shrimpton. Reconsidering generic composition. In *EUROCRYPT*, pages 257–274, 2014.
[35] NIST. FIPS PUB 186-4 Digital Signature Standard (DSS). http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf, 2013.
[36] K. G. Paterson and N. J. AlFardan. Plaintext-recovery attacks against datagram TLS. In *NDSS*, 2012.
[37] O. Pereira, F. Standaert, and S. Vivek. Leakage-resilient authentication and encryption from symmetric cryptographic primitives. In *ACM CCS*, pages 96–108, 2015.
[38] K. Pietrzak. A leakage-resilient mode of operation. In *EUROCRYPT*, pages 462–482, 2009.
[39] E. Rescorla. The transport layer security (tls) protocol version 1.3. https://tlswg.github.io/tls13-spec/draft-ietf-tls-tls13.html, July 2017.
[40] M. Rivain and E. Prouff. Provably secure higher-order masking of AES. In *CHES*, pages 413–427, 2010.
[41] P. Rogaway and T. Shrimpton. Deterministic authenticated-encryption: A provable-security treatment of the key-wrap problem. *IACR Cryptology ePrint Archive*, 2006:221, 2006.
[42] P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In *EUROCRYPT*, pages 373–390, 2006.
[43] F. Standaert, O. Pereira, and Y. Yu. Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In *CRYPTO*, pages 335–352, 2013.
[44] N. Veyrat-Charvillon, B. Gérard, and F. Standaert. Soft analytical side-channel attacks. In *ASIACRYPT*, pages 282–296, 2014.
[45] N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F. Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In *ASIACRYPT*, pages 740–757, 2012.

[46] Y. Yu and F. Standaert. Practical leakage-resilient pseudorandom objects with minimum public randomness. In *CT-RSA*, pages 223–238, 2013.
[47] Y. Yu, F. Standaert, O. Pereira, and M. Yung. Practical leakage-resilient pseudorandom generators. In *ACM CCS*, pages 141–151, 2010.

A PROOF OF THEOREM 5.3

Let $\mathcal{A}$ be a (q,t)-CIML adversary against DTE making $q_e + q_d \leq q$ queries, where q_e is the number of encryption queries and q_d the number of decryption queries. We say that the final output ciphertext $(\tau^\dagger, c^\dagger)$ is the $(q+1)$-th query of the game. Without loss of generality we assume that any answer to some encryption query is never sent as a decryption query and conversely. We also assume that the final output is not an answer to some encryption query, otherwise the adversary loses anyway.

Since we are in the same condition as in the proof of misuse resistant, we name by $\bar{E}_i$ the event where the winning condition of CIML is satisfied which can be viewed as the analogue of E_i with an additional decryption query: the $(q+1)$-th query which is the last of the game. We thus have to focus on proving that the $(q+1)$-th query is also invalid even when all the ephemeral key k_0's associated to the encryption queries only are given in E_i.

Let's see what happens in E_1 where F^* was replaced by a random function f if $\mathsf{f}(\tau) = k_0$ was given to the adversary, where $\tau = \mathsf{f}(\mathsf{H}(r\|m))$ for the encryption query (r,m). Obviously, k_0 gives nothing more since in E_1 the encryption algorithm from k_0 is run honestly as in E_0. We then get an adversary against F^* in $\bar{E}_1$ making at most $2(q+1)$ queries since we must count the $(q+1)$-th query and running in time bounded by $t + (q+1)(t_{\mathsf{H}} + (2\ell+1)t_{\mathsf{F}}) \leq t'$. Nevertheless, we assume F^* to be $(2q+2, t', \varepsilon_{\mathsf{F}^*})$-pseudorandom and we find $|\Pr[\bar{E}_0] - \Pr[\bar{E}_1]| \leq \varepsilon_{\mathsf{F}^*}$.

Likewise with E_1, we consider the partition $\bar{E}_1 \cap (\bar{F}_1 \cup \bar{F}_2)$ and $\bar{E}_1 \cap \bar{F}_3$, where $\bar{F}_1$ is the analogue of F_1 meaning that collision on *associated* digests occurs, where $\bar{F}_2$ is an extended version of F_2 where some *associated* digest $\mathsf{H}(r,m) = h$ is equal to some *associated* τ' or to some *associated* k_0'' (which simply has the form $\mathsf{f}(\tau'')$ for some associated τ''), and where $\bar{F}_3$ is the complement of $\bar{F}_1 \cup \bar{F}_2$. We stress that the fact that the k_0's associated to encryption queries leak does not affect the emulations made in F_1, F_2 and F_3 since we remain in the same game. It is now straightforwards that $\Pr[\bar{F}_1] \leq \varepsilon_{cr}$ since we get an adversary against the $(0, t', \varepsilon_{cr})$-collision resistance of H running in the time bounded by t'. Moreover, since in F_2' we already put targets of the range-oriented preimage resistance of H in place of all the associated tags *and* the associated ephemeral key k_0's we also have an adversary here (built from $\mathcal{A}$), for $\bar{F}_2$, asking/receiving at most $(2q+2)$ targets and running in time bounded by t'. By assumption on H, we must have $\Pr[\bar{F}_2] \leq \varepsilon_{cr}$ and we are thus left with bounding $\Pr[\bar{E}_1|\bar{F}_3]$.

We are ready for the last transition from $\bar{E}_1|\bar{F}_3$ to $\bar{E}_2$ where we reach the game where all the decryption queries including the $(q+1)$-th one are answered by $\perp$. It is straightforward to show that $|\Pr[\bar{E}_1|\bar{F}_3] - \Pr[\bar{E}_2]| \leq (q+1)/2^n$, which concludes the proof. □

B PROOF OF THEOREM 6.1

Let $\mathcal{A}$ be a (q,t)-CIML adversary against DCE making $q_e + q_d \leq q$ queries, where q_e is the number of encryption queries and q_d the number of decryption queries. We have to bound the probability $\Pr[\mathsf{CIML}_{\mathsf{DCE},\mathcal{A}} = 1]$. Without loss of generality we assume that any answer to some encryption query is never sent as a decryption query and conversely. We also assume that the final output is not an answer to some encryption query, otherwise the adversary looses anyway.

The proof is in the spirit of the proof of Theorem 5.3 except that $\mathcal{A}$ cannot compute H itself: it must query the random oracle to get h. However, since h is random here, the distribution of $\mathsf{F}_k^*(\mathsf{H}(r\|m))$ in DCE is similar to the distribution of $\mathsf{F}_k^* \circ \mathsf{F}_k^*(\mathsf{H}(r\|m))$ in DTE by relying on the pseudorandomness of F^*. Then, all the ephemeral keys k_0 *associated* to encryption queries are random (See the proof of Theorem 5.3).

Let us assume that the final output ciphertext $(\tau^\dagger, c^\dagger)$ is the $(q+1)$-th query of the game. Then we only need to replace $q+1$ outputs of F_k^* by random values (instead of computing k_0's). By reusing the argument detailed in the proof of Theorem 5.3, we obtain that the $(q+1, t', \varepsilon_{\mathsf{F}^*})$-pseudorandomness of F^* is sufficient to bound the gap resulting from this transition by $\varepsilon_{\mathsf{F}^*}$: we can easily build an adversary running in time $t + (q+1)(2\ell+1)t_{\mathsf{F}} \leq t'$, since all the h's are already random.

The probability that some collision occurs among all the h's and the k_0's is bounded by $4(q+1)^2/2^n$. Therefore, assuming that no collision happens, if a decryption query (h, c) is valid it must be the case that $\mathsf{H}(r\|m)$ returned by the random oracle where r and m are computed during decryption matches h which has a probability bounded by $1/2^n$ for each query. Thus all the ciphertexts of the encryption queries including the $(q+1)$-th one are invalid except with probability $(q+1)/2^n$. □

C PROOF OF THEOREM 7.5

We start by defining Game 0 as the $\mathsf{PrivK}^{\mathsf{lmcpa},0}_{\mathcal{A}^{\mathsf{L}},\mathsf{DCE}}$ game.

Game 1 is equal to Game 0, except that we abort if, when processing the queries of $\mathcal{A}^{\mathsf{L}}$, the same randomness r is picked twice. The probability of this event is bounded by $q_e^2/|\mathcal{R}|$.

Game 2 is equal to Game 1, except that we abort if, when processing the queries of $\mathcal{A}^{\mathsf{L}}$, a collision happens on the hash function, that is, it the adversary provides messages m and m' such that, when performing their encryption, it happens that $\mathsf{H}(r|m) = \mathsf{H}(r'|m')$ (note that $r \neq r'$, because of the failure condition of Game 1). The gap between Game 2 and Game 1 is bounded by ϵ_{cr}: a collision resistance adversary can run $\mathcal{A}^{\mathsf{L}}$ and its LMCPA challenger (in time t_c, and using q_e+1 leakage queries), and search for a collision (in time t_{sc}), placing us within the bounds of the hash function security.

Game 3 is equal to Game 2 except that, for all queries, the challenger replaces the computation of the key $k_0 = \mathsf{F}_k(\mathsf{h})$ with the selection of a random key $k_0 \xleftarrow{\$} \mathcal{B}$ (we assume that this does not increase its running time). Since the previous

failure conditions guarantee that h is always fresh, the gap between Game 3 from Game 2 is bounded by ϵ_{F}: a PRF adversary can run $\mathcal{A}^{\mathsf{L}}$ and its LMCPA challenger (within (q_e+1, t_c) bounds), except that it queries the PRF challenger with all the h values that it computes.

Game 4 is equal to Game 3 except that, during the test query of the LMCPA game, the computation of $H(r\|m_0)$ (and the corresponding leakage) is replaced by the computation of $\mathsf{H}(r\|m_1)$. Here the probability of distinguishing Game 4 from Game 3 is bounded by ϵ_{LH}: an adversary against the leakage resilience of H can run $\mathcal{A}^{\mathsf{L}}$ and its LMCPA challenger (as tweaked in Game 3, and within (q_e+1, t_c) bounds), except that it hands the computation of h to the leakage resilient hash function challenger during the test query.

Game 5 is equal to Game 4 except that, during the test query of the LMCPA game, the selection of a random k_0 (from Game 3) is replaced by the selection of a random h^* and the computation of $k_0 = \mathsf{F}_k(\mathsf{h}^*)$. The gap between Game 5 from Game 4 is bounded by ϵ_{F}: a PRF adversary can run $\mathcal{A}^{\mathsf{L}}$ and its LMCPA challenger (within (q_e+1, t_c) bounds), except that it queries the PRF challenger with the h^* value that it computes.

To sum up, at this stage, $\mathcal{A}^{\mathsf{L}}$ sees:

- During an encryption query: the expected hash and leakage, and an encryption component encrypting that hash and leakage, but with a randomly chosen k_0 (hence independent of the long-term key k).
- During the test query: the hash and leakage of $(r\|m_1)$, followed by a PSV encryption of $(r\|m_0)$ with key k.

The presence of this isolated PSV encryption makes it possible to use the leakage resilient eavesdropper security of that scheme.

Game 6 is equal to Game 5 except that, during the test query of the LMCPA game, we encrypt $(r\|m_1)$ instead of $(r\|m_0)$. The gap between Game 6 and Game 5 is bounded by $\epsilon^{\mathsf{eav}}_{\mathsf{PSV}-\mathsf{ENC}}$, since we can build an EAV adversary running $\mathcal{A}^{\mathsf{L}}$ and the LMCPA challenger (within (q_e+1, t_c) bounds), except that it hands the two messages $(r\|m_0)$ and $(r\|m_1)$ to the leavchallenger and returns the corresponding ciphertext to $\mathcal{A}^{\mathsf{L}}$.

Game 7 now hops to the $\mathsf{PrivK}^{\mathsf{lmcpa},1}_{\mathcal{A}^{\mathsf{L}},\mathsf{DCE}}$ game by undoing most of the hops that we made before, introducing the same gaps again, but keeping m_1 in place:

- We go back to a uniformly random k_0 by undoing the Game 4-5 transform.
- We go back to the selection of random k_0's everywhere to the use of a PRF as in the Game 2-3 transform.
- We stop aborting if the same randomness r is picked twice or if a collision happens in the hash function, as in the Game 0-2 transforms.

To sum-up we observe that the total gap introduced by our sequence of games is bounded by $2\frac{q_e}{|\mathcal{R}|} + 2\epsilon_{cr} + 4\epsilon_{\mathsf{F}} + \epsilon_{\mathsf{L_H}} + \epsilon^{\mathsf{eav}}_{\mathsf{PSV}-\mathsf{ENC}}$. Besides, none of our reductions requires more leakage function queries than those needed to run the LMCPA challenger, and time more than the one needed to run that challenger and look for a collision in the outputs of the evaluation of the hash function that result from answering the adversary's queries in the LMCPA game (in Game 2). □

D PROOF OF THEOREM 7.6

We only detail the steps that differ from the proof of Thm. 7.5.

We split Game 3 into two steps, in order to be able to replace the tag τ and key k_0 values with random values. In the first step, we replace F_k with a random function f, bringing an ϵ_{F} gap as before. In the second step, we replace the evaluation of f by the selection of random values, which is only equivalent if f is never queried on the same value twice. This is actually the case, except with probability less than $4(q_e+1)^2/|\mathcal{B}|$. Indeed: a collision between two hashes is precluded by Games 1 and 2; a collision between two τ values can only happen with probability bounded by $(q_e+1)^2/|\mathcal{B}|$ (this upper-bounds the probability of a collision in the range of f invoked on distinct values); and a collision between a hash and a τ value is also bounded by by $(q_e+1)^2/|\mathcal{B}|$ (the τ's a selected at random by f, and each of them will collide with one of the q_e+1 distinct hashes with probability $(q_e+1)/\mathcal{B}$.

In a similar way, we add a step in Game 7, in order to revert the transform above, bringing a second $2(q_e+1)^2/|\mathcal{B}|$ gap. □