Efficient Public-Key Cryptography with Bounded Leakage and Tamper Resilience

Antonio Faonio[1] and Daniele Venturi[2]

[1] Department of Computer Science, Aarhus University, Aarhus, Denmark
[2] Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Abstract. We revisit the question of constructing public-key encryption and signature schemes with security in the presence of bounded leakage and tampering memory attacks. For signatures we obtain the first construction in the standard model; for public-key encryption we obtain the first construction free of pairing (avoiding non-interactive zero-knowledge proofs). Our constructions are based on generic building blocks, and, as we show, also admit efficient instantiations under fairly standard number-theoretic assumptions.

The model of bounded tamper resistance was recently put forward by Damgård *et al.* (Asiacrypt 2013) as an attractive path to achieve security against arbitrary memory tampering attacks without making hardware assumptions (such as the existence of a protected self-destruct or key-update mechanism), the only restriction being on the number of allowed tampering attempts (which is a parameter of the scheme). This allows to circumvent known impossibility results for unrestricted tampering (Gennaro *et al.*, TCC 2010), while still being able to capture realistic tampering attacks.

Keywords: public-key encryption, signatures, related-key attacks, tampering, leakage

1 Introduction

Motivated by the proliferation of memory tampering attacks and fault injection [11,13,46], a recent line of research—starting with the seminal work of Bellare and Kohno [8] on the related-key attack (RKA) security of blockciphers—aims at designing cryptographic primitives that provably resist such attacks. Briefly, memory tampering attacks allow an adversary to modify the secret key of a targeted cryptographic scheme, and later violate its security by observing the effect of such changes at the output. In practice such attacks can be implemented by several means, both in hardware and software.

This paper is focused on designing public-key primitives—i.e., public-key encryption (PKE) and signature schemes—with provable security guarantees against memory tampering attacks. In this setting, the modified secret key might be the signing key of a certification authority or of an SSL server, or the decryption key of a user. Informally, security of a signature scheme under tampering

attacks can be cast as follows. The adversary is given a target verification key vk and can observe signatures of adaptively chosen messages both under the original secret key sk and under related keys $sk' = T(sk)$, derived from sk by applying efficient tampering functions T chosen by the adversary; the goal of the adversary is to forge a signature on a "fresh message" (i.e., a message not asked to the signing oracle) under the original verification key. Tamper resistance of PKE schemes under chosen-ciphertext attacks (CCA) can be defined similarly, the difference being that the adversary is allowed to observe decryption of adaptively chosen ciphertexts under related secret keys sk', and its goal is now to violate semantic security.

Unrestricted tampering. The best we could hope for would be, of course, to allow the adversary to make any polynomial number of arbitrary, efficiently computable, tampering queries. Unfortunately, this type of "unrestricted tampering" is easily seen to be impossible without making further assumptions, as observed for the first time by Gennaro *et al.* [29]. The attack of [29] is simple enough to recall it here. The first tampering attempt defines sk'_1 to be equal to sk with the first bit set to zero, so that verifying a signature under sk'_1 essentially allows to learn the first bit b_1 of the secret key with overwhelming probability. The second tampering attempt defines sk'_2 to be equal to sk with the second bit set to zero, and with the first bit equal to b_1, and so on. This way each tampering attempt can be exploited to reveal one bit of the secret key, yielding a total security breach after $s(\kappa)$ queries, where $s(\kappa)$ is the bit-length of the secret key as a function of the security parameter.[3]

A possible way out to circumvent such an attack is to rely on the so-called self-destruct feature: Find a way how to detect tampering with high probability, and completely erase the memory or "blow-up the device" whenever tampering is detected. While this is indeed a viable approach, it has some shortcomings (at it can, e.g., be exploited for carrying out denial-of-service attacks), and so finding alternatives is an important research question. One natural such alternative is to simply restrict the power of the tampering functions T, in such a way that carrying out the above attack simply becomes impossible. This approach led to the design of several public-key primitives resisting an arbitrary polynomial number of *restricted* tampering attempts. All these schemes share the feature that the secret key belongs to some finite field, and the set of allowed modifications consist of all linear or affine functions, or all polynomials of bounded degree, applied to the key [7,54,10].

Bounded tampering. Unfortunately, the approach of restricting the tampering class only offers a partial solution to the problem; the main reason for this is that it is not a priori clear how the above mentioned algebraic relations capture realistic tampering attacks (where, e.g., a chip is shot with a laser). Motivated

[3] A similar attack works for PKE schemes, and more generally for a large class of cryptographic primitives that can be tested for malfunctioning [29]; one can also make the above attack completely stateless.

by this shortcoming, in a recent work, Damgård *et al.* [18] suggested the model of *bounded* tampering, where one assumes an upper-bound $\tau \in \mathbb{N}$ on the total number of tampering attempts the adversary is allowed to ever make; apart from this, and from the fact that the tampering functions T should be efficiently computable, there is no further restriction on the adversarial tampering. Arguably, such form of tamper-proof security is sufficient to capture realistic attacks in which tampering might anyway destroy the device under attack or it could be detected by auxiliary hardware countermeasures; moreover, this model allows to analyze the security of cryptographic primitives already "in the wild," *without* the need to modify the implementation to include, e.g., a self-destruct feature.

An important parameter in the model of bounded tampering is the so-called tampering rate $\rho(\kappa) := \tau(\kappa)/s(\kappa)$ defined to be the ratio between the number of allowed tampering attempts and the size $s(\kappa)$ of the secret key in bits. The attack of Gennaro *et al.* [29] shows that necessarily $\rho(\kappa) \leq 1 - 1/p(\kappa)$ for some polynomial $p(\cdot)$. The original work of [18] shows how to obtain signature schemes and PKE schemes tolerating linear tampering rate $\rho(\kappa) = O(1/\kappa)$. However, the signature construction relies on the so-called Fiat–Shamir heuristic [28], whose security can only be proven in the random oracle model; the PKE construction can be instantiated in the standard model, but requires an untamperable common reference string (CRS), being based on (true simulation-extractable) non-interactive zero-knowledge (NIZK) [20].

In a follow-up work [19], the same authors show that resilience against bounded tampering can be obtained via a generic transformation yielding tampering rate $\rho(\kappa) = O(1/\sqrt[3]{\kappa^2})$; however, the transformation only gives a weaker form of security against non-adaptive (or semi-adaptive [19]) tampering attacks.

1.1 Our Contribution

In this work we improve the current state of the art on signature schemes and PKE schemes provably resisting bounded memory tampering. In the case of signatures, we obtain the first constructions in the standard model based on generic building blocks; as we argue, this yields concrete signature schemes tolerating tampering rate $\rho(\kappa) = O(1/\kappa)$ under standard complexity assumptions such as the Symmetric External Diffie-Hellman (SXDH) [52,12] and the Decisional Linear (DLIN) [53,35] assumptions. In the case of PKE, we obtain a direct, pairing-free, construction based on certain hash-proof systems [17], yielding concrete PKE schemes tolerating tampering rate $\rho(\kappa) = O(1/\kappa)$ under a particular instantiation of the Refined Subgroup Indistinguishability (RSI) assumption [45].

More precisely, we show that *already existing* schemes can be proved secure against bounded tampering. We do not view this as a limitation of our result, as it confirms the perspective that the model of bounded tamper resilience allows to make statements about cryptographic primitives already used "in the wild" (that might have already been implemented and adopted in applications). Additionally, our security arguments are non-trivial, requiring significant modifications to the original proofs (more on this below). In what follows we explain

our contributions and techniques more in details. We refer the reader to Table 1 for a summary of our results and a comparison with previous work.

Reference	Type	Attack Class	Model	Tampering Rate	Assumption
BCM11 [7]	Sig./PKE	Linear	Standard	∞	DDHI [1]
Wee12 [54]	PKE	Linear	Standard	∞	BDDH/LWE
BPT12 [10]	Sig./PKE	Affine	Random Oracle	∞	BDH
		Polynomial	Standard	∞	EDBDH
DFMV13 [18]	Sig.	Any	Random Oracle	$O(1/\kappa)$	DLOG/Factoring
	PKE	Any	Standard	$O(1/\kappa)$	SXDH/DLIN
BMT14 [9]	Sig.	Affine	Standard	∞	DLOG
		Exponentiation	Standard	∞	RSA
		Addition	Standard	∞	LWE
DFMV15 [19]	Sig./PKE	Any	Standard†	$O(1/\sqrt[3]{\kappa^2})$	OWF/TDP
JW15 [37]	Sig./PKE	Poly-size Circuits	Standard	∞	OWF/TDP
QLY^{+}15 [51]	Sig./PKE	Polynomial	Standard	∞	DDH/DCR
Ours § 3	Sig.	Any	Standard	$O(1/\kappa)$	SXDH/DLIN
Ours § 4	PKE	Any	Standard	$O(1/\kappa)$	RSI

Table 1. Comparing known constructions of public-key primitives with security against related-key attacks (without self-destruct and key updating mechanisms). The value "∞" under the column "tampering rate" means that the scheme supports an arbitrary polynomial number of tampering queries. † Only achieves security against non-adaptive tampering.

Signatures. We prove that the leakage-resilient signature scheme by Dodis *et al.* [20] is secure against bounded tampering attacks. The scheme of [20] satisfies the property that it remains unforgeable even given bounded leakage on the signing key. The main idea for showing security against bounded tampering, is to reduce tampering to leakage. Notice that this is non-trivial, because in the tampering setting the adversary is allowed to see polynomially many signatures corresponding to each of the tampered secret keys (which are at most τ), and this yields a total amount of key-dependent information which is much larger than the tolerated leakage.

We now explain how to overcome this obstacle. The scheme exploits a so-called leakage-resilient hard relation R; such a relation satisfies the property that, given a statement y generated together with a witness x, it is unfeasible to compute a witness x^* for $(x^*, y) \in R$; moreover the latter holds even given bounded leakage on x. The verification key of the signature scheme consists of a random y, while the secret key is equal to x, where (x, y) is a randomly generated pair belonging to the relation R. In order to sign a message m, one simply outputs a non-interactive zero-knowledge proof of knowledge π of x, where the message m is used as a label in the proof. Verification of a signature can be done by verifying the accompanying proof.

In the security proof, by the zero-knowledge property, we can replace real proofs with simulated proofs. Moreover, by the proof of knowledge property, we can actually extract a valid witness x^* for (x^*, y) from the adversarial forgery π^*; note that, since the forger gets to see simulated proofs, the extractability requirement must hold even after seeing proofs generated via the zero-knowledge simulator. Finally, we can transform a successful forger for the signature scheme into an adversary breaking the underlying leakage-resilient relation; the trick is that the reduction can leak the statement y' corresponding to any tampered witness $x' = T(x)$, which allows to simulate an arbitrary polynomial number of signature queries corresponding to x' by running several independent copies of the zero-knowledge simulator upon input y'. Thus bounded tamper resilience follows by bounded leakage resilience.

A subtle technicality in the above argument is that the statement y' must be efficiently computable as a function of x'. We call a relation R satisfying this property a *complete* relation. As we define it, completeness additionally requires that any derived witness $x' = T(x)$ is a witness for a valid statement y' (i.e., $(x', y') \in R$); importantly this allows us to argue that simulated proofs are always for *true* statements, which leads to practical instantiations of the scheme. When we instantiate the signature scheme, of course, we need to make sure that the underlying relation meets our completeness requirement. Unfortunately, this is not directly the case for the constructions given in [20], but, as we show, such a difficulty can be overcome by carefully twisting the instantiation of the underlying relations.

Public-key encryption. Next, we prove that the PKE scheme by Qin and Liu [49] is secure against bounded tampering. The scheme is based on a variant of the classical Cramer-Shoup paradigm for constructing CCA-secure PKE [16,17]. Specifically, the PKE scheme combines a universal hash-proof system (HPS) together with a one-time lossy filter (OTLF) used to authenticate the ciphertext; the output of a randomness extractor is then used in order to mask the message in a one-time pad fashion. Since the OTLF is unkeyed, the secret key simply consists of the private evaluation key of the HPS, which makes it easier to analyze the security of the PKE scheme in the presence of memory tampering. The bulk of our proof is, indeed, to show that HPS with certain parameters already satisfy bounded tamper resilience.

More in details, every HPS is associated to a set $\mathcal{C}$ of ciphertexts and a subset $\mathcal{V} \subset \mathcal{C}$ of so-called *valid* ciphertexts, together with (the description of) a keyed hash function with domain $\mathcal{C}$. The hash function can be both evaluated privately (using a secret evaluation key) and publicly (on ciphertexts in $\mathcal{V}$, and using a public evaluation key). The main security guarantee is that for any $C \in \mathcal{C} \setminus \mathcal{V}$ the output of the hash function upon input C is unpredictable even given the public evaluation key. In the construction of [49] a ciphertext consists of an element $C \in \mathcal{V}$, from which we derive an hash value K which serves for two purposes: (i) To extract a random pad via a seeded extractor, used to mask the plaintext; (ii) To authenticate the ciphertext by producing an encoding Π of K via the OTLF. The decryption algorithm first derives the value K using the secret evaluation

key for the HPS, and then it uses this value to unmask the plaintext provided that the value Π can be verified correctly (otherwise decryption results in $\perp$).

In the reduction, the OTLF encoding will be programmed in such a way that, for all ciphertexts asked to the decryption oracle, the encoding is an injective function. This implies that, in order to create a ciphertext with a correct encoding Π, one has to know the underlying hash value K. To prove (standard) CCA security, one argues that all decryption queries with values $C \in \mathcal{V}$ do not reveal any additional information about the secret key, since the corresponding value K could be computed via the public evaluation procedure; as for decryption queries with values $C \in \mathcal{C} \setminus \mathcal{V}$, the corresponding value K is unpredictable, and therefore the decryption oracle will output $\perp$ with overwhelming probability which, again, does not reveal any additional information about the secret key.

The scenario in the case of tampering is more complicated. Consider a decryption oracle instantiated with a tampered secret key $sk' = T(sk)$. A decryption query containing a value $C \in \mathcal{V}$ might now reveal some information about the secret key; however, as we show, this information can be simulated by leaking the public key pk' corresponding to sk'. Decryption queries containing values $C \in \mathcal{C} \setminus \mathcal{V}$ are harder to simulate. This is because the soundness property of the HPS only holds for a uniformly chosen evaluation key, while sk', clearly, is not uniform. To overcome this obstacle we distinguish two cases:

- In case the value $T(sk)$ has low entropy, such a value does not reveal too much information on the secret key, and thus, at least intuitively, even if the decryption does not output $\perp$ the resulting plaintext should not decrease the entropy of the secret key by too much;
- In case the value $T(sk)$ has high entropy, we argue that it is safe to use this key within the HPS, i.e. we show that the soundness of the HPS is preserved as long as the secret key hash high entropy (even if it is not uniform).

With the above in mind, the security proof is similar to the ones in [44,49].

Trading tampering and leakage. Since our security arguments essentially reduce bounded tampering to bounded leakage (by individuating a short secret-key-dependent hint that allows to simulate polynomially many tampering queries for a given modified key), the theorems we get show a natural tradeoff between the obtained bounds for leakage and tamper resistance.

In particular, our results nicely generalizes previous work, in that we obtain the same bounds as in [20,49] by plugging $\tau = 0$ in our theorem statements.

1.2 Related Work

Bounded leakage. The signature scheme of Dodis *et al.* [20] generalizes and improves a previous construction by Katz and Vaikuntanathan [39]. Similarly, the PKE construction by Qin and Liu builds upon the seminal work of Naor and Segev [44]; the scheme was further improved in [50].

Related-key security. Related-key security was first studied in the context of symmetric encryption [8,43,30,3,2]. With time a number of cryptographic primitives with security against related-key attacks have emerged, including pseudorandom functions [6,40,4,1], hash functions [31], identity-based encryption [7,10], public-key encryption [7,54,10,42], signatures [7,10,9], and more [51,37,15].

All the above works achieve security against an unbounded number of restricted tampering attacks (typically, algebraic relations). Kalai, Kanukurthi, and Sahai [38], instead, show how to achieve security against unrestricted tampering without self-destruct, by assuming a protected mechanism to update the secret key of certain public-key cryptosystems (without modifying the corresponding public key).

Non-malleable codes. An alternative approach to achieve tamper-proof security of arbitrary cryptographic primitives against memory tampering is to rely on so-called non-malleable codes. While this solution yields security against an unbounded number of tampering queries, it relies on self-destruct and moreover it requires to further assume that the tampering functions are restricted in granularity (see, e.g., [22,41,25]) and/or computational complexity [26,37,5].

Tamper-proof computation. A related line of work (starting with [36,27]), finally, aims at constructing secure compilers protecting against tampering attacks targeting the computation carried out by a cryptographic device (typically in the form of boolean and arithmetic circuits).

2 Preliminaries

2.1 Notation

Notation. For $a, b \in \mathbb{R}$, we let $[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$; for $a \in \mathbb{N}$ we let $[a] = \{1, 2, \ldots, a\}$. If x is a string, we denote its length by $|x|$; if $\mathcal{X}$ is a set, $|\mathcal{X}|$ represents the number of elements in $\mathcal{X}$. When x is chosen randomly in $\mathcal{X}$, we write $x \leftarrow\!\!\$\, \mathcal{X}$. When A is an algorithm, we write $y \leftarrow\!\!\$\, \mathsf{A}(x)$ to denote a run of A on input x and output y; if A is randomized, then y is a random variable and $\mathsf{A}(x; r)$ denotes a run of A on input x and randomness r. An algorithm A is *probabilistic polynomial-time* (PPT) if A is randomized and for any input $x, r \in \{0,1\}^*$ the computation of $\mathsf{A}(x; r)$ terminates in at most $poly(|x|)$ steps.

Throughout the paper we let $\kappa \in \mathbb{N}$ denote the security parameter. We say that a function $\nu : \mathbb{N} \to \mathbb{R}$ is negligible in the security parameter κ if $\nu(\kappa) = \kappa^{-\omega(1)}$. For two ensembles $\mathcal{X} = \{\mathbf{X}_\kappa\}_{\kappa \in \mathbb{N}}$ and $\mathcal{Y} = \{\mathbf{Y}_\kappa\}_{\kappa \in \mathbb{N}}$, we write $\mathcal{X} \equiv \mathcal{Y}$ if they are identically distributed, $\mathcal{X} \approx_s \mathcal{Y}$ to denote that the corresponding distributions are statistically close, and $\mathcal{X} \approx_c \mathcal{Y}$ to denote that the two ensembles are computationally indistinguishable.

Languages and relations. A *decision problem* related to a language $L \subseteq \{0,1\}^*$ requires to determine if a given string y is in L or not. We can associate to any *NP*-language L a polynomial-time recognizable relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$

defining L itself, i.e. $L = \{y : \exists x \text{ s.t. } (x, y) \in R\}$ for $|x| \leqslant poly(|y|)$. The string y is called *theorem*, and the string x is called a *witness* for membership of $y \in L$.

Random variables. The min-entropy of a random variable $\mathbf{X}$, defined over a set $\mathcal{X}$, is $\mathbb{H}_\infty(\mathbf{X}) := -\log \max_{x \in \mathcal{X}} \mathbb{P}[\mathbf{X} = x]$, and it measures how $\mathbf{X}$ can be predicted by the best (unbounded) predictor. The average conditional min-entropy of a random variable $\mathbf{X}$ given a random variable $\mathbf{Y}$ and conditioned on an event E is defined as $\widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Y}, E) := -\log(\mathbb{E}_{y \leftarrow\!\$\, \mathbf{Y}} \left[2^{-\mathbb{H}_\infty(\mathbf{X}|\mathbf{Y}=y,E)}\right])$. We rely on the following basic facts.

Lemma 1 ([21]). *Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be random variables. If $\mathbf{Y}$ has at most 2^ℓ possible values, then $\widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) \geqslant \widetilde{\mathbb{H}}_\infty(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) - \ell \geqslant \widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Z}) - \ell$.*

Lemma 2. *Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be random variables such that $\mathbf{Y} = f(\mathbf{X}, \mathbf{Z})$ for an efficiently computable function f. Then $\widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Y}, \mathbf{Z}, E) \geqslant \widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Z}, E) - \beta$, where the event E is defined as $\{\forall z : \mathbb{H}_\infty(\mathbf{Y}|\mathbf{Z} = z) \leqslant \beta\}$.*

Proof. Let A be the best predictor for $\mathbf{X}$, given $\mathbf{Y}$ and $\mathbf{Z}$ and conditioned on the event E. Consider the predictor A' that upon input $\mathbf{Z}$ first samples an independent copy $\mathbf{X}'$ of the random variable $\mathbf{X}$ and then runs A upon input $f(\mathbf{X}', \mathbf{Z})$. Note that the event E holds for the inputs given to A', therefore the probability that $f(\mathbf{X}', \mathbf{Z}) = f(\mathbf{X}, \mathbf{Z})$ is bounded above by $2^{-\beta}$. This implies the lemma. □

2.2 Public-Key Encryption

A public-key encryption (PKE) scheme is a tuple of algorithms $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows. (1) Algorithm Setup takes as input the security parameter and outputs public parameters $pub \in \{0,1\}^*$; all algorithms are implicitly given pub as input. (2) Algorithm Gen takes as input the security parameter and outputs a public/secret key pair (pk, sk); the set of all secret keys is denoted by $\mathcal{SK}$ and the set of all public keys by $\mathcal{PK}$. (3) The randomized algorithm Enc takes as input the public key pk, a message $m \in \mathcal{M}$, and randomness $r \in \mathcal{R}$, and outputs a ciphertext $c = \mathsf{Enc}(pk, m; r)$; the set of all ciphertexts is denoted by $\mathcal{C}$. (4) The deterministic algorithm Dec takes as input the secret key sk and a ciphertext $c \in \mathcal{C}$, and outputs $m = \mathsf{Dec}(sk, c)$ which is either equal to some message $m \in \mathcal{M}$ or to an error symbol $\perp$.

Correctness. We say that $\mathcal{PKE}$ satisfies *correctness* if for all $pub \leftarrow\!\$\, \mathsf{Setup}(1^\kappa)$ and $(pk, sk) \leftarrow\!\$\, \mathsf{Gen}(1^\kappa)$ we have that $\mathbb{P}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m] = 1$ (where the randomness is taken over the internal coin tosses of algorithm Enc).

BLT Security. We now turn to defining indistinguishability under chosen-ciphertext attacks (IND-CCA) in the bounded leakage and tampering (BLT) setting.

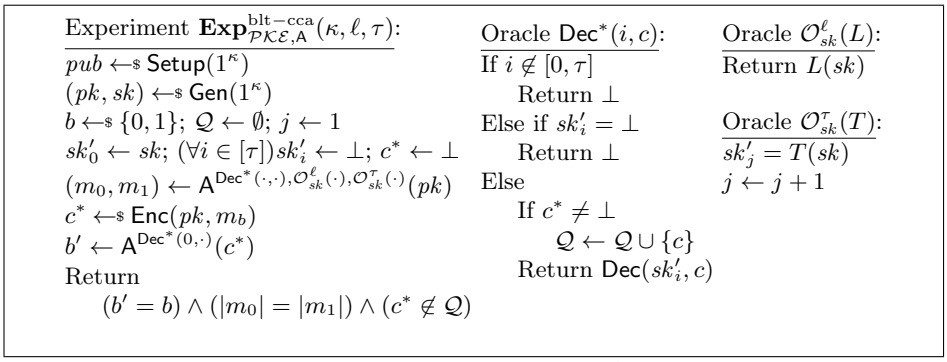

Fig. 1: Experiment defining BLT-IND-CCA security of $\mathcal{PKE}$.

Definition 1. *For $\kappa \in \mathbb{N}$, let $\ell = \ell(\kappa)$ and $\tau = \tau(\kappa)$ be parameters. We say that $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is (τ, ℓ)-BLT-IND-CCA if for all PPT adversaries A there exists a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that*

$$\left| \mathbb{P}\left[\mathbf{Exp}^{\text{blt-cca}}_{\mathcal{PKE},\mathsf{A}}(\kappa, \ell, \tau) = 1 \right] - \frac{1}{2} \right| \leq \nu(\kappa),$$

where the experiment $\mathbf{Exp}^{\text{blt-cca}}_{\mathcal{PKE},\mathsf{A}}(\kappa, \ell, \tau)$ is defined in Figure 1.

A few remarks on the definition are in order. In the specification of the BLT-IND-CCA security experiment, oracle $\mathcal{O}^\ell_{sk}$ takes as input (arbitrary polynomial-time computable) functions $L : \mathcal{SK} \to \{0,1\}^*$, and returns $L(sk)$ for a total of at most ℓ bits. In a similar fashion, oracle $\mathcal{O}^\tau_{sk}$ takes as input (arbitrary polynomial-time computable) functions $T : \mathcal{SK} \to \mathcal{SK}$, and defines the i-th tampered secret key as $sk'_i = T(sk)$; the oracle accepts at most τ queries. Oracle Dec^* can be used to decrypt arbitrary ciphertexts c under the i-th tampered secret key (or under the original secret key), provided that c is different from the challenge ciphertext.

Notice that A is not allowed to tamper with or leak from the secret key after seeing the challenge ciphertext. As shown in [18] this restriction is necessary already for the case $(\tau, \ell) = (1, 0)$. Finally, we observe that in case $(\tau, \ell) = (0, 0)$ we get, as a special case, the standard notion of IND-CCA security. Similarly, for $\tau = 0$ and $\ell > 0$, we obtain as a special case the notion of "semantic security against a-posteriori chosen-ciphertext ℓ-key-leakage attacks" from [44].

2.3 Signatures

A signature scheme is a tuple of algorithms $\mathcal{SIG} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ specified as follows. (1) Algorithm Setup takes as input the security parameter and outputs public parameters $pub \in \{0,1\}^*$; all algorithms are implicitly given pub as input. (2) Algorithm Gen takes as input the security parameter and outputs a public/secret key pair (vk, sk); the set of all signing keys is denoted by $\mathcal{SK}$. (3) The randomized algorithm Sign takes as input the signing key sk, a message

$m \in \mathcal{M}$, and randomness $r \in \mathcal{R}$, and outputs a signature $\sigma := \mathsf{Sign}(sk, m; r)$ on m. (4) The deterministic algorithm Vrfy takes as input the verification key vk and a pair (m, σ), and outputs a decision bit (indicating whether (m, σ) is a valid signature with respect to vk).

Correctness. We say that $\mathcal{SIG}$ satisfies *correctness* if for all messages $m \in \mathcal{M}$ and for all $pub \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$ and $(vk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$, algorithm $\mathsf{Vrfy}(vk, m, \mathsf{Sign}(sk, m))$ outputs 1 with all but negligible probability (over the coin tosses of the signing algorithm).

BLT Security. We now define what it means for a signature scheme to be existentially unforgeable against chosen-message attacks (EUF-CMA) in the bounded leakage and tampering (BLT) setting.

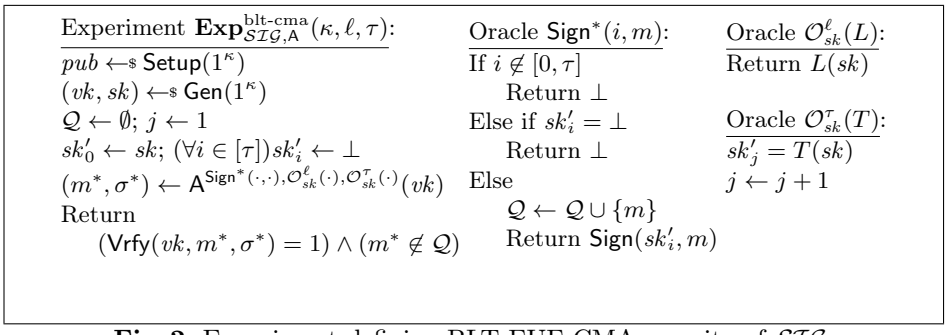

Fig. 2: Experiment defining BLT-EUF-CMA security of $\mathcal{SIG}$.

Definition 2. *For $\kappa \in \mathbb{N}$, let $\ell = \ell(\kappa)$ and $\tau = \tau(\kappa)$ be parameters. We say that $\mathcal{SIG} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is (τ, ℓ)-BLT-EUF-CMA if for all PPT adversaries A there exists a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that*

$$\mathbb{P}\left[\mathbf{Exp}^{\text{blt-cma}}_{\mathcal{SIG},\mathsf{A}}(\kappa, \ell, \tau) = 1\right] \leq \nu(\kappa),$$

where the experiment $\mathbf{Exp}^{\text{blt-cma}}_{\mathcal{SIG},\mathsf{A}}(\kappa, \ell, \tau)$ is defined in Figure 2.

The syntax of oracles $\mathcal{O}^\ell_{sk}$ and $\mathcal{O}^\tau_{sk}$ is the same as before. Oracle Sign^* can be used to sign arbitrary messages m under the i-th tampered signing key $sk'_i = T(sk)$, or under the original signing key sk; the goal of the adversary is to forge a signature on a "fresh" message, i.e. a message that was never queried to oracle Sign^*. Note that for $(\tau, \ell) = (0, 0)$ we obtain the standard notion of existential unforgeability under chosen-message attacks. Similarly, for $\tau = 0$ and $\ell > 0$, we obtain the definition of leakage-resilient signatures [39].

3 Signatures

In this section we give a generic construction of signature schemes with BLT-EUF-CMA in the standard model. In particular, we show that the construction by Dodis *et al.* [20] is already resilient to bounded leakage and tampering attacks.

3.1 The Scheme of Dodis, Haralambiev, Lòpez-Alt, and Wichs

The signature scheme is based on the following ingredients.

Hard relations. A leakage-resilient hard relation [20].

Definition 3. *A relation R is an ℓ-leakage-resilient hard relation, with witness space $\mathcal{X}$ and theorem space $\mathcal{Y}$, if the following requirements are met.*

Samplability: *There exists a PPT algorithm* SamR *such that for all pairs* $(x, y) \leftarrow\!\!\$\ \mathsf{SamR}(1^\kappa)$ *we have* $(x, y) \in R$*, with* $x \in \mathcal{X}$ *and* $y \in \mathcal{Y}$*.*

Verifiability: *There exists a PPT algorithm that decides if a given pair* (x, y) *satisfies* $(x, y) \in R$*.*

Completeness: *There exists an efficient deterministic function* ξ *that given as input any* $x \in \mathcal{X}$ *returns* $y = \xi(x) \in \mathcal{Y}$ *such that* $(x, y) \in R$*.*

Hardness: *For all PPT adversaries* A *there exists a negligible function* $\nu : \mathbb{N} \to [0, 1]$ *such that*

$$\mathbb{P}\left[(x^*, y) \in R : \ (x, y) \leftarrow\!\!\$\ \mathsf{SamR}(1^\kappa); x^* \leftarrow\!\!\$\ \mathsf{A}^{\mathcal{O}^\ell_x(\cdot)}(y)\right] \leq \nu(\kappa),$$

where the probability is taken over the random coin tosses of SamR *and* A*, and where oracle* $\mathcal{O}^\ell_x(\cdot)$ *takes as input efficiently computable functions* $L : \mathcal{X} \to \{0, 1\}^*$ *and returns* $L(x)$ *for a total of at most* ℓ *bits.*

NIZK. A true-simulation extractable non-interactive zero-knowledge (tSE NIZK) argument system $\mathcal{NIZK} = (\mathsf{I}, \mathsf{P}, \mathsf{V})$ for the relation R, supporting labels [20]. Recall that a NIZK argument system supporting labels has the following syntax: (i) Algorithm I takes as input the security parameter $\kappa \in \mathbb{N}$ and generates a common reference string (CRS) $crs \leftarrow\!\!\$\ \mathsf{I}(1^\kappa)$. (ii) Algorithm P takes as input the CRS, a label $\lambda \in \{0, 1\}^*$, and some pair $(x, y) \in R$, and returns a proof $\pi \leftarrow\!\!\$\ \mathsf{P}^\lambda(crs, x, y)$. (iii) Algorithm V takes as input the CRS, a label $\lambda \in \{0, 1\}^*$, and some pair (x, π), and returns a decision bit $\mathsf{V}^\lambda(crs, y, \pi)$. Moreover:

Definition 4. *We say that* $\mathcal{NIZK} = (\mathsf{I}, \mathsf{P}, \mathsf{V})$ *is a tSE NIZK for the relation R, supporting labels, if the following requirements are met.*

Correctness: *For all pairs* $(x, y) \in R$ *and for all labels* $\lambda \in \{0, 1\}^*$ *we have that* $\mathsf{V}^\lambda(crs, y, \mathsf{P}^\lambda(crs, x, y)) = 1$ *with overwhelming probability over the coin tosses of* P*,* V*, and over the choice of* $crs \leftarrow\!\!\$\ \mathsf{I}(1^\kappa)$*.*

Unbounded zero-knowledge: *There exists a PPT simulator* $\mathsf{S} := (\mathsf{S}_1, \mathsf{S}_2)$ *such that for all PPT adversaries* A *the following quantity is negligible:*[4]

$$\left|\mathbb{P}\left[b = b' : \begin{array}{c} b \leftarrow\!\!\$\ \{0,1\}; (crs, tk) \leftarrow\!\!\$\ \mathsf{S}_1(1^\kappa); (x, y, \lambda) \leftarrow\!\!\$\ \mathsf{A}(crs, tk) \\ \pi_0 \leftarrow\!\!\$\ \mathsf{P}^\lambda(crs, x, y); \pi_1 \leftarrow\!\!\$\ \mathsf{S}_2^\lambda(tk, y); b' \leftarrow\!\!\$\ \mathsf{A}(crs, tk, \pi_b) \end{array}\right] - \frac{1}{2}\right|.$$

True-simulation extractability: *There exists a PPT extractor* K *such that for all PPT adversaries* A *the following quantity is negligible:*

$$\mathbb{P}\left[\begin{array}{c}(\lambda^* \notin \mathcal{Q}) \wedge (\mathsf{V}^{\lambda^*}(crs, y^*, \pi^*) = 1) \\ \wedge ((x^*, y^*) \notin R)\end{array} : \begin{array}{c}(crs, tk) \leftarrow\!\!\$\ \mathsf{S}_1(1^\kappa) \\ (y^*, \pi^*, \lambda^*) \leftarrow\!\!\$\ \mathsf{A}^{\mathcal{O}_{\mathsf{S}_2,\tau}(\cdot,\cdot,\cdot)}(crs) \\ x^* \leftarrow\!\!\$\ \mathsf{K}^{\lambda^*}(tk, y^*, \pi^*)\end{array}\right],$$

where oracle $\mathcal{O}_{\mathsf{S}_2,\tau}$ *takes as input tuples* (x_i, y_i, λ_i) *and returns the same as* $\mathsf{S}_2^{\lambda_i}(tk, y_i)$ *as long as* $(x_i, y_i) \in R$ *(and* $\perp$ *otherwise), and* $\mathcal{Q}$ *is the set of all labels* λ_i *asked to oracle* $\mathcal{O}_{\mathsf{S}_2,\tau}$.

The signature scheme. Consider now the following signature scheme $\mathcal{SIG} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, based on a relation R, and on a non-interactive argument system $\mathcal{NIZK} = (\mathsf{I}, \mathsf{P}, \mathsf{V})$ for R, supporting labels.

- $\mathsf{Setup}(1^\kappa)$: Sample $crs \leftarrow\!\!\$\ \mathsf{I}(1^\kappa)$ and return $pub := (crs, R)$. (Recall that all algorithms implicitly take pub as input.)
- $\mathsf{Gen}(1^\kappa)$: Run $(x, y) \leftarrow\!\!\$\ \mathsf{SamR}(1^\kappa)$ and define $vk := y$ and $sk := x$.
- $\mathsf{Sign}(sk, m)$: Compute $\pi \leftarrow\!\!\$\ \mathsf{P}^m(crs, x, \xi(x))$ and return $\sigma := \pi$; note that the message m is used as a label in the argument system, and that the value $y = \xi(x)$ can be efficiently computed as a function of x.
- $\mathsf{Vrfy}(vk, m, \sigma)$: Parse (vk, σ) as $vk := y$ and $\sigma := \pi$, and output the same as $\mathsf{V}^m(crs, y, \pi)$.

Theorem 1. *For* $\kappa \in \mathbb{N}$*, let* $\ell := \ell(\kappa)$*,* $\ell' := \ell'(\kappa)$*,* $\tau := \tau(\kappa)$*, and* $n := n(\kappa)$ *be parameters. Assume that* R *is an* ℓ'*-leakage-resilient hard relation with theorem space* $\mathcal{Y} := \{0,1\}^n$*, and that* $\mathcal{NIZK}$ *is a tSE NIZK for* R*. Then the signature scheme* $\mathcal{SIG}$ *described above is* (ℓ, τ)*-BLT-EUF-CMA with* $\ell + (\tau + 1) \cdot n \leq \ell'$.

3.2 Security Proof

We consider a sequence of mental experiments, starting with the initial game $\mathbf{Exp}^{\text{blt-cma}}_{\mathcal{SIG},\mathsf{A}}(\kappa, \ell, \tau)$ which for simplicity we denote by $\mathbf{G}_0$.

Game $\mathbf{G}_0$. This is exactly the game of Definition 2, where the signature scheme $\mathcal{SIG}$ is the scheme described in the previous section. In particular, upon input the i-th tampering query T_i the modified secret key $x'_i = T_i(x)$ is computed. Hence, the answer to a query (i, m) to oracle Sign^* is computed by parsing $pub = (crs, R)$, computing the statement $y'_i = \xi(x'_i)$ corresponding to x'_i, and outputting $\sigma := \pi$ where $\pi \leftarrow\!\!\$\ \mathsf{P}^m(crs, x'_i, y'_i)$.

[4] Strictly speaking we should quantify the definition over all adversaries returning pairs $(x, y) \in R$; alternatively, we can slightly abuse notation and assume that both P and S_2 return $\perp$ if that is not the case.

Game $\mathbf{G}_1$. We change the way algorithm Setup generates the CRS. Namely, instead of sampling $crs \leftarrow\!\!\$\ \mathsf{I}(1^\kappa)$ we now run $(crs, tk) \leftarrow\!\!\$\ \mathsf{S}_1(1^\kappa)$ and additionally we replace the proofs output by oracle Sign^* by simulated proofs, i.e., $\pi \leftarrow\!\!\$\ \mathsf{S}_2(tk, y_i')$ where $y_i' = \xi(x_i')$.

Game $\mathbf{G}_2$. We change the winning condition of the previous game. Namely, the game now outputs one if and only if π^* is valid w.r.t. y (as before) and additionally $(x^*, y) \in R$ where the value x^* is computed from the proof π^* running the extractor K of the underlying argument system.

We now establish a series of lemmas, showing that the above games are computationally indistinguishable. The first lemma states that $\mathbf{G}_0$ and $\mathbf{G}_1$ are indistinguishable, down to the unbounded zero-knowledge property of the argument system.

Lemma 3. *For all PPT adversaries A there exists a negligible function $\nu_{0,1} : \mathbb{N} \to [0,1]$ such that $|\mathbb{P}[\mathbf{G}_0(\kappa) = 1] - \mathbb{P}[\mathbf{G}_1(\kappa) = 1]| \leq \nu_{0,1}(\kappa)$.*

Proof. We prove a stronger statement, namely that $\mathbf{G}_0(\kappa) \approx_c \mathbf{G}_1(\kappa)$. By contradiction, assume that there exists a PPT distinguisher $\mathsf{D}_{0,1}$ and a polynomial $p_{0,1}(\cdot)$ such that, for infinitely many values of $\kappa \in \mathbb{N}$, we have that $\mathsf{D}_{0,1}$ distinguishes between game $\mathbf{G}_0$ and game $\mathbf{G}_1$ with probability at least $1/p_{0,1}(\kappa)$. Let $q \in poly(\kappa)$ be the number of signature queries asked by $\mathsf{D}_{0,1}$. For an index $j \in [q+1]$ consider the hybrid game $\mathbf{H}_j$ that answers the first $j-1$ queries as in game $\mathbf{G}_0$ and all subsequent queries as in game $\mathbf{G}_1$. Note that $\mathbf{H}_1 \equiv \mathbf{G}_1$ and $\mathbf{H}_{q+1} \equiv \mathbf{G}_0$.

By a standard hybrid argument, we have that there exists an index $j^* \in [q]$ such that $\mathsf{D}_{0,1}$ tells apart $\mathbf{H}_{j^*}$ and $\mathbf{H}_{j^*+1}$ with non-negligible probability $1/q \cdot 1/p_{0,1}(\kappa)$. We build a PPT adversary $\mathsf{A}_{0,1}$ that (using distinguisher $\mathsf{D}_{0,1}$ and knowledge of $j^* \in [q]$) breaks the non-interactive zero-knowledge property of the argument system. A formal description of $\mathsf{A}_{0,1}$ follows.

Adversary $\mathsf{A}_{0,1}$:

- Receive (crs, tk) from the challenger, where $(crs, tk) \leftarrow\!\!\$\ \mathsf{S}_1(1^\kappa)$.
- Run $(x, y) \leftarrow\!\!\$\ \mathsf{SamR}(1^\kappa)$, set $pub := (crs, R)$, $vk := y$, $x_0' \leftarrow x$, $x_i' \leftarrow \perp$ for all $i \in [\tau]$, and send (pub, vk) to $\mathsf{D}_{0,1}$.
- Upon input a leakage query L return $L(x)$ to $\mathsf{D}_{0,1}$; upon input a tampering query T, set $x_i' = T(x)$.
- Upon input the j-th signature query of type (i, m), if $i \notin [0, \tau]$ or $x_i' = \perp$, answer with $\perp$. Otherwise, proceed as follows:
  - If $j \leq j^* - 1$, return $\sigma \leftarrow\!\!\$\ \mathsf{P}^m(crs, x_i', \xi(x_i'))$ to $\mathsf{D}_{0,1}$.
  - Else, if $j = j^*$, forward $(x_i', \xi(x_i'), m)$ to the challenger, receiving back a proof π_b; return $\sigma := \pi_b$ to $\mathsf{D}_{0,1}$.
  - Else, if $j \geq j^* + 1$, forward $\sigma \leftarrow\!\!\$\ \mathsf{S}_2^m(tk, \xi(x_i'))$ to $\mathsf{D}_{0,1}$.
- Output whatever D outputs.

For the analysis, note that the only difference between game $\mathbf{H}_{j^*}$ and game $\mathbf{H}_{j^*+1}$ is on how the j^*-th signature query is answered. In particular, in case

the hidden bit b in the definition of non-interactive zero-knowledge equals zero, $\mathsf{A}_{0,1}$'s simulation produces exactly the same distribution as in $\mathbf{H}_{j^*}$, and otherwise $\mathsf{A}_{0,1}$'s simulation produces exactly the same distribution as in $\mathbf{H}_{j^*+1}$. Hence, $\mathsf{A}_{0,1}$ breaks the NIZK property with non-negligible advantage $1/q \cdot 1/p_{0,1}(\kappa)$, a contradiction. This concludes the proof. □

The second lemma states that $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable, down to the true-simulation extractability property of the argument system.

Lemma 4. *For all PPT adversaries* A *there exists a negligible function* $\nu_{1,2} : \mathbb{N} \to [0,1]$ *such that* $|\mathbb{P}[\mathbf{G}_1(\kappa) = 1] - \mathbb{P}[\mathbf{G}_2(\kappa) = 1]| \leq \nu_{1,2}(\kappa)$.

Proof. We prove a stronger statement, namely that $\mathbf{G}_1(\kappa) \approx_c \mathbf{G}_2(\kappa)$. Define the following "bad event" *Bad*, in the probability space of game $\mathbf{G}_1$: The event becomes true if the adversarial forgery $(m^*, \sigma^* := \pi^*)$ is valid (i.e., the proof π^* is valid w.r.t. statement y and label m^*), but running the extractor $\mathsf{K}(tk, \cdot, \cdot)$ on (y, π^*) yields a value x^* such that $(x^*, y) \notin R$.

Notice that $\mathbf{G}_1(\kappa)$ and $\mathbf{G}_2(\kappa)$ are identically distributed conditioning on *Bad* not happening. Hence, by a standard argument, it suffices to bound the probability of provoking event *Bad* by all PPT adversaries A. By contradiction, assume that there exists a PPT adversary $\mathsf{A}_{1,2}$ and a polynomial $p_{1,2}(\cdot)$ such that, for infinitely many values of $\kappa \in \mathbb{N}$, we have that $\mathsf{A}_{1,2}$ provokes event *Bad* with probability at least $1/p_{1,2}(\kappa)$. We build an adversary A' that (using $\mathsf{A}_{1,2}$) breaks true-simulation extractability of the argument system. A formal description of A' follows.

Adversary A':
- Receive crs from the challenger, where $(crs, tk) \leftarrow\!\!\$\ \mathsf{S}_1(1^\kappa)$.
- Sample $(x, y) \leftarrow\!\!\$\ \mathsf{SamR}(1^\kappa)$, set $pub := (crs, R)$, $vk := y$, $x'_0 \leftarrow x$, $x'_i \leftarrow \perp$ (for all $i \in [\tau]$), and forward (pub, vk) to $\mathsf{A}_{1,2}$.
- Upon input a leakage query L return $L(x)$ to $\mathsf{A}_{1,2}$; upon input a tampering query T, set $x'_i = T(x)$.
- Upon input the j-th signature query of type (i, m), if $i \notin [0, \tau]$ or $x'_i = \perp$, answer with $\perp$. Otherwise, forward $(x'_i, \xi(x'_i), m)$ to the challenger obtaining a proof π as a response, and return $\sigma := \pi$ to $\mathsf{A}_{1,2}$.
- Whenever $\mathsf{A}_{1,2}$ returns a pair (m^*, σ^*), define $\pi^* := \sigma^*$ and output (y, π^*, m^*).

For the analysis, we note that A' perfectly simulates signature queries. In fact, by completeness of the underlying relation, the pair $(x'_i, \xi(x'_i))$ is always in the relation R, and thus the proof π obtained by the reduction is always for a true statement and has exactly the same distribution as in game $\mathbf{G}_1$. As a consequence, $\mathsf{A}_{1,2}$ will provoke event *Bad* with probability $1/p_{1,2}(\kappa)$, and thus the pair (y, π^*) output by the reduction violates the tSE property of the non-interactive argument with non-negligible probability $1/p_{1,2}(\kappa)$. This finishes the proof. □

Finally, we show that the advantage of any PPT adversary in game $\mathbf{G}_2$ must be negligible, otherwise one could violate the hardness of the underlying leakage-resilient relation.

Lemma 5. *For all PPT adversaries* A *there exists a negligible function* $\nu_2 : \mathbb{N} \to [0,1]$ *such that* $\mathbb{P}[\mathbf{G}_2 = 1] \leq \nu_2(\kappa)$.

Proof. By contradiction, assume there exists a PPT adversary A_2 and a polynomial $p_2(\cdot)$ such that, for infinitely many values of $\kappa \in \mathbb{N}$, adversary A_2 makes game $\mathbf{G}_2$ output 1 with probability at least $1/p_2(\kappa)$. We construct a PPT adversary A'' (using A_2) breaking hardness of the leakage-resilient relation R. A description of A'' follows.

Adversary A'':
- Receive y from the challenger, where $(x, y) \leftarrow\!\!\$\ \mathsf{SamR}(1^\kappa)$.
- Sample $(crs, tk) \leftarrow\!\!\$\ \mathsf{S}_1(1^\kappa)$, set $pub := (crs, R)$, $y'_i \leftarrow \perp$ (for all $i \in [\tau]$), $vk := y$, and forward (pub, vk) to A_2.
- Define the leakage function $L_\xi(x) := \xi(x)$ and forward L_ξ to the target leakage oracle $\mathcal{O}^\ell_x$, obtaining a value y'_0.
- Upon input a leakage query L, forward L to the target leakage oracle $\mathcal{O}^\ell_x$ and return to A_2 the answer received from the oracle.
- Upon input the i-th tampering query T, define the function $L_{T,\xi}(x) := \xi(T(x))$, and forward $L_{T,\xi}$ to the target leakage oracle $\mathcal{O}^\ell_x$; set the value y'_i equal to the answer obtained from the oracle.
- Upon input the j-th signature query of type (i, m), if $i \notin [0, \tau]$ or $y'_i = \perp$, answer with $\perp$. Otherwise, run $\pi \leftarrow\!\!\$\ \mathsf{S}_2^m(tk, y'_i)$ and return $\sigma := \pi$ to A_2.
- Whenever $\mathsf{A}_{1,2}$ returns a forgery (m^*, σ^*), define $\pi^* := \sigma^*$ and output x^* such that $x^* \leftarrow\!\!\$\ \mathsf{K}^{m^*}(tk, y, \pi^*)$.

For the analysis, note that A'' perfectly simulates signature queries. In fact, for each tampering query T the reduction obtains the statement y'_i corresponding to $x'_i := T(x)$ via a leakage query; given this value a signature for key x'_i is computed by running the zero-knowledge simulator (as defined in $\mathbf{G}_2$). Moreover, the total leakage asked by A'' equals ℓ (as A_2 leaks at most ℓ bits from the secret key) plus $n \cdot \tau$ (as for each tampering function T the reduction leaks n bits, and A_2 makes at most τ such queries), plus n bits (as the value $y'_0 = \xi(x)$ is needed for simulating signature queries w.r.t. the original secret key), and by assumption $\ell + (\tau + 1) \cdot n \leq \ell'$. Hence, A'' breaks the hardness of the leakage-resilient relation with non-negligible probability $1/p_2(\kappa)$. This concludes the proof. □

The proof of the theorem follows by combining the above lemmas.

3.3 Concrete Instantiations

We now explain how to instantiate the signature scheme from the previous section using standard complexity assumptions. We need two ingredients: (i) A

leakage-resilient hard relation R; (ii) A tSE NIZK for the same relation R, supporting labels. For the latter component, we rely on the construction due to Dodis *et al.* [20] that allows to obtain a tSE NIZK for arbitrary relations, based on a standard (non-extractable) NIZK for a related relation (see below) and an IND-CCA-secure PKE scheme supporting labels.

Let $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CCA-secure PKE scheme supporting labels, with message space $\mathcal{X}$. Plugging in the construction from [20] a signature has the form $\sigma := (c, \pi)$, where $c \leftarrow\!\!\$\ \mathsf{Enc}^{\lambda}(pk, x)$ and π is a standard NIZK argument for the following derived relation:

$$R^* := \{((y, c, pk, m), (x, r)) : \ (x, y) \in R \wedge c = \mathsf{Enc}^m(pk, x; r)\}\,. \tag{1}$$

Diffie-Hellman Assumptions. In what follows, let $\mathbb{G}$ be a group with prime order q and with generator g. Also, let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be groups of prime order q and $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a non-degenerate, efficiently computable, bilinear map.

Discrete Logarithm. Let $g \leftarrow\!\!\$\ \mathbb{G}$ and $x \leftarrow\!\!\$\ \mathbb{Z}_q$. The Discrete Logarithm (DL) assumption holds in $\mathbb{G}$ if it is computationally hard to find $x \in \mathbb{Z}_q$ given $y = g^x \in \mathbb{G}$.

Decisional Diffie-Hellman. Let $g_1, g_2 \leftarrow\!\!\$\ \mathbb{G}$ and $x_1, x_2, x \leftarrow\!\!\$\ \mathbb{Z}_q$. The Decisional Diffie-Hellman (DDH) assumption holds in $\mathbb{G}$ if the following distributions are computationally indistinguishable: $(\mathbb{G}, g_1, g_2, g_1^{x_1}, g_2^{x_2})$ and $(\mathbb{G}, g_1, g_2, g_1^{x}, g_2^{x})$.

Symmetric External Diffie-Hellman. The Symmetric External Diffie-Hellman (SXDH) assumption states that the DDH assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$. Such an assumption is not satisfied in case $\mathbb{G}_1 = \mathbb{G}_2$, but it is believed to hold in case there is no efficiently computable mapping between $\mathbb{G}_1$ and $\mathbb{G}_2$ [52,12].

D-Linear [53,35]. Let $D \geq 1$ be a constant, and let $g_1, \ldots, g_{D+1} \leftarrow\!\!\$\ \mathbb{G}$ and $x_1, \ldots, x_D \leftarrow\!\!\$\ \mathbb{Z}_q$. We say that the D-linear assumption holds in $\mathbb{G}$ if the following distributions are computationally indistinguishable: $(\mathbb{G}, g_1^{x_1}, \ldots, g_D^{x_D}, g_{D+1}^{x_{D+1}})$ and $(\mathbb{G}, g_1^{x_1}, \ldots, g_D^{x_D}, g_{D+1}^{\sum_{i=1}^{D} x_i})$. Note that for $D = 1$ we obtain the DDH assumption, and for $D = 2$ we obtain the so-called Linear assumption [53].

Construction based on SXDH. The first instantiation is based on the SXDH assumption, working with asymmetric pairing based groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$. The construction below is similar to the one given in [20, Section C.2.2], except that we had to modify the underlying hard relation, in that the one used by Dodis *et al.* does not meet our completeness requirement.[5]

[5] In particular, a pair $(x, y) \in R$ is computed by sampling random exponents $r_1, \ldots, r_N \leftarrow\!\!\$\ \mathbb{Z}_q$ and outputting $x_i := g^{r_i}$ and $y := \prod_{i=1}^{N} g_i^{r_i}$, where g is a generator of $\mathbb{G}_2$ and $g_1, \ldots, g_N$ are generators of $\mathbb{G}_1$; thus, by the SXDH assumption, it is hard to compute y given only $x_1, \ldots, x_N$, without knowledge of the randomness $r_1, \ldots, r_N$.

Hard relation: Let $N \geq 2$, and $g_1, \ldots, g_N \leftarrow_{\$} \mathbb{G}_1$ be generators. The sampling algorithm chooses a random $x := (x_1, \ldots, x_N) \leftarrow_{\$} \mathbb{G}_2^N$ and defines $y := \prod_{i=1}^{N} \mathsf{e}(g_i, x_i) \in \mathbb{G}_T$. Notice that the relation satisfies completeness, with mapping function $\xi(\cdot)$ defined by $\xi(x) := \prod_{i=1}^{N} \mathsf{e}(g_i, x_i)$. In the full version [24], we argue that this relation is leakage-resilient under the SXDH assumption.

Lemma 6. *Under the SXDH assumption in $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$, the above defined relation is an ℓ-leakage-resilient hard relation for $\ell \leq (N-1)\log q$.*

PKE: We use the Cramer-Shoup PKE scheme in $\mathbb{G}_2$ [16], optimized as described in [20]. The public key consists of random generators $(h_1, h_2, h_{3,1}, \ldots, h_{3,N}, h_4, h_5)$ of $\mathbb{G}_2$, and in order to encrypt $x = (x_1, \ldots, x_N) \in \mathbb{G}_2^N$ under label $m \in \{0,1\}^*$ we return a ciphertext:

$$c := (c_1, \ldots, c_{N+3}) = (h_1^r, h_2^r, h_{3,1}^r \cdot x_1, \ldots, h_{3,N}^r \cdot x_N, (h_4 \cdot h_5^t)^r)$$

with $r \leftarrow_{\$} \mathbb{Z}_q$, and where $t := H(c_1 || \cdots || c_{N+2} || m)$ is computed using a standard collision-resistant hash function.

NIZK: We use the Groth-Sahai proof system [32]. In order to prove that a given pair $x^* := (x, r)$ and $y^* := (y, c, pk, m)$ belongs to the relation of Eq. (1), we first prove that $(x, y) \in R$. This requires to show satisfiability of a one-sided pairing product equation, which can be done with a proof consisting of $2N+16$ elements in $\mathbb{G}_1$ and 2 elements in $\mathbb{Z}_q$ (under the SXDH assumption). Next, we prove validity of a ciphertext which requires to show satisfiability of a system of $N+3$ one-sided multi-exponentiation equations; the latter can be done with a proof consisting of $(N+3)+2N = 3N+3$ group elements (under the SXDH assumption).

Corollary 1. *Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be asymmetric pairing based groups with prime order q. Under the SXDH assumption there exists a signature scheme satisfying BLT-EUFCMA with tampering rate $\rho(\kappa) = O(1/\kappa)$. For $N \geq 2$, the public key consists of a single group element, the secret key consists of N group elements, and a signature consists of $6N + 22$ group elements and 2 elements in $\mathbb{Z}_q$.*

Construction based on DLIN. The second instantiation is based on the DLIN assumption, working with symmetric pairing based groups $(\mathbb{G}, \mathbb{G}_T)$. The construction below is similar to one of the instantiations given in [20, Section C.2.3], except that we had to modify the underlying hard relation, in that the one used by Dodis *et al.* does not meet our completeness requirement.

Hard relation: Let $N \geq 3$, and $g_1, \ldots, g_N, g'_1, \ldots, g'_N \leftarrow_{\$} \mathbb{G}$ be generators. The sampling algorithm chooses a random $x := (x_1, \ldots, x_N) \leftarrow_{\$} \mathbb{G}$ and defines $y_1 := \prod_{i=1}^{N} \mathsf{e}(g_i, x_i) \in \mathbb{G}_T$ and $y_2 := \prod_{i=1}^{N} \mathsf{e}(g'_i, x_i)$. Notice that the relation satisfies completeness, with mapping function $\xi(\cdot)$ defined by $\xi(x) := (\prod_{i=1}^{N} \mathsf{e}(g_i, x_i), \prod_{i=1}^{N} \mathsf{e}(g'_i, x_i))$. In the full version [24], we argue that this relation is leakage-resilient under the DLIN assumption.

Lemma 7. *Under the DLIN assumption in $(\mathbb{G}, \mathbb{G}_T)$, the above defined relation is an ℓ-leakage-resilient hard relation for $\ell \leq (N-2)\log q$.*

PKE: We use the Linear Cramer-Shoup PKE scheme in $\mathbb{G}$ [53], optimized as described in [20]. The public key consists of random generators $(h_0, h_1, h_2, h_{3,1}, \ldots, h_{3,N}, h_{4,1}, \ldots, h_{4,N}, h_{5,1}, h_{5,2}, h_{6,1}, h_{6,2})$ of $\mathbb{G}$, and in order to encrypt $x = (x_1, \ldots, x_N) \in \mathbb{G}^N$ under label $m \in \{0,1\}^*$ we return a ciphertext:

$$c := (c_1, \ldots, c_{N+4}) = (h_0^{r_1+r_2}, h_1^{r_1}, h_2^{r_2}, h_{3,1}^{r_1} \cdot h_{4,1}^{r_2} \cdot x_1, \\ \ldots, h_{3,N}^{r_1} \cdot h_{4,N}^{r_2} \cdot x_N, (h_{4,1} \cdot h_{5,1}^t)^{r_1} \cdot (h_{4,2} \cdot h_{5,2}^t)^{r_2})$$

with $r_1, r_2 \leftarrow_\$ \mathbb{Z}_q$, and where $t := H(c_1 || \cdots || c_{N+3} || m)$ is computed using a standard collision-resistant hash function.

NIZK: We use again the Groth-Sahai proof system. In order to prove that a given pair $x^* := (x, r)$ and $y^* := ((y_1, y_2), c, pk, m)$ belongs to the relation of Eq. (1), we first prove that $(x, (y_1, y_2)) \in R$. This requires to show satisfiability of two one-sided pairing product equations, which can be done with a proof consisting of $3N + 42$ elements in $\mathbb{G}$ and 6 elements in $\mathbb{Z}_q$ (under the DLIN assumption). Next, we prove validity of a ciphertext which requires to show satisfiability of a system of $N+4$ one-sided multi-exponentiation equations; the latter can be done with a proof consisting of $2(N+4)+3N = 5N+8$ group elements (under the DLIN assumption).

Corollary 2. *Let $(\mathbb{G}, \mathbb{G}_T)$ be symmetric pairing based groups with prime order q. Under the DLIN assumption there exists a signature scheme satisfying BLT-EUFCMA with tampering rate $\rho(\kappa) = O(1/\kappa)$. For $N \geq 3$, the public key consists of two group elements, the secret key consists of N group elements, and a signature consists of $9N + 54$ group elements and 6 elements in $\mathbb{Z}_q$.*

4 Public-Key Encryption

We give a construction of an efficient PKE scheme satisfying BLT-IND-CCA security in the standard model. In particular, we prove that the PKE scheme of Qin and Liu [49] is already resilient to bounded leakage and tampering attacks.

4.1 The Scheme of Qin and Liu

The encryption scheme is a twist of the well-known Cramer-Shoup paradigm for CCA security [17], and is based on the following ingredients.

Hash-proof systems. An ϵ-universal hash-proof system (HPS) $\mathcal{HPS} = (\mathsf{Gen}_{\mathrm{hps}}, \mathsf{Pub}, \mathsf{Priv})$. Recall that a HPS has the following syntax: (i) Algorithm $\mathsf{Gen}_{\mathrm{hps}}$ takes as input the security parameter, and outputs public parameters $pub := (aux, \mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{SK}, \mathcal{PK}, \Lambda_{(\cdot)} : \mathcal{C} \to \mathcal{K}, \mu : \mathcal{SK} \to \mathcal{PK})$ where aux might contain additional structural parameters, and where Λ_{sk} is a hash function and, for any $sk \in \mathcal{SK}$, the function $\mu(sk)$ defines the action of Λ_{sk} over the subset $\mathcal{V}$ of valid ciphertexts (i.e., Λ_{sk} is projective). Moreover the function Λ_{sk} is ϵ-almost universal:

Definition 5. *A projective hash function $\Lambda_{(\cdot)}$ is* ϵ-almost universal*, if for all pk, $C \in \mathcal{C} \setminus \mathcal{V}$, and all $K \in \mathcal{K}$, it holds that $\mathbb{P}\left[\Lambda_{\mathbf{SK}}(C) = K | \mathbf{PK} = pk, C\right] \leqslant \epsilon$, where $\mathbf{SK}$ is uniform over $\mathcal{SK}$ conditioned on $\mathbf{PK} = \mu(\mathbf{SK})$.*

(ii) Algorithm Pub takes as input a public key $pk = \mu(sk)$, a *valid ciphertext* $C \in \mathcal{V}$, and a witness w for $C \in \mathcal{V}$, and outputs the value $\Lambda_{sk}(C)$. (iii) Algorithm Priv take as input the secret key sk and a ciphertext $C \in \mathcal{C}$, and outputs the value $\Lambda_{sk}(C)$.

Definition 6. *A hash-proof system $\mathcal{HPS}$ is ϵ-almost universal if the following holds:*

1. *For all sufficiently large $\kappa \in \mathbb{N}$, and for all possible outcomes of $\mathsf{Gen}_{\mathrm{hps}}(1^\kappa)$, the underlying projective hash function is $\epsilon(\kappa)$-almost universal.*
2. *The underlying set membership problem is hard. Specifically, for any PPT adversary A the following quantity is negligible:*

$$\mathbf{Adv}^{\mathrm{smp}}_{\mathcal{HPS},\mathsf{A}} := |\mathbb{P}[\mathsf{A}(\mathcal{C}, \mathcal{V}, C_0) = 1 |\ C_0 \leftarrow\!\!\$\ \mathcal{V})] - \mathbb{P}[\mathsf{A}(\mathcal{C}, \mathcal{V}, C_1) = 1 |\ C_1 \leftarrow\!\!\$\ \mathcal{C} \setminus \mathcal{V})]|.$$

The lemma below directly follows from the definition of hash-proof system and the notion of min-entropy.

Lemma 8. *Let $\Lambda_{(\cdot)}$ be ϵ-almost universal. Then for all pk and $C \in \mathcal{C} \setminus \mathcal{V}$ it holds that $\mathbb{H}_\infty(\Lambda_{\mathbf{SK}}(C) | \mathbf{PK} = pk, C) \geqslant -\log \epsilon$ where $\mathbf{SK}$ is uniform over $\mathcal{SK}$ conditioned on $\mathbf{PK} = \mu(\mathbf{SK})$.*

One-time lossy filters [49]. A One-Time Lossy Filter (OTLF) $\mathcal{LF} = (\mathsf{Gen}_{\mathrm{lf}}, \mathsf{Eval}, \mathsf{LTag})$ is a family of functions $\mathsf{LF}_{\phi,t}(X)$ indexed by a public key ϕ and a tag t. Recall that a OTLF has the following syntax: (i) Algorithm $\mathsf{Gen}_{\mathrm{lf}}$ takes as input the security parameter, and outputs a public key ϕ and a trapdoor key ψ. The public key ϕ defines a tag space $\mathcal{T} := \{0,1\}^* \times \mathcal{T}_c$ that contains two disjoint subsets $\mathcal{T}_{\mathrm{inj}}$ and $\mathcal{T}_{\mathrm{loss}}$ and a domain space $\mathcal{D}$. (ii) Algorithm Eval takes as input ϕ, a tag $t = (t_a, t_c) \in \mathcal{T}$ (where we call t_a the auxiliary tag and t_c the core tag), and $X \in \mathcal{D}$, and outputs $\mathsf{LF}_{\phi,t}(X)$. (iii) Algorithm LTag takes as input ψ and an auxiliary tag $t_a \in \{0,1\}^*$, and outputs a core tag t_c such that $t = (t_a, t_c) \in \mathcal{T}_{\mathrm{loss}}$.

Definition 7. *We say that $\mathcal{LF} = (\mathsf{Gen}_{\mathrm{lf}}, \mathsf{Eval}, \mathsf{LTag})$ is an ℓ_{lf}-OTLF with domain $\mathcal{D}$ if the following proprieties hold:*

Lossiness: *In case the tag t is injective (i.e., $t \in \mathcal{T}_{\mathrm{inj}}$), so is the function $\mathsf{LF}_{\phi,t}(\cdot) := \mathsf{Eval}(\phi, t, \cdot)$. In case t is lossy (i.e., $t \in \mathcal{T}_{\mathrm{loss}}$), then $\mathsf{LF}_{\phi,t}(\cdot)$ has image size at most $2^{\ell_{\mathrm{lf}}}$.*

Indistinguishability: *No PPT adversary A is able to distinguish lossy tags from random tags, i.e. the following quantity is negligible:*

$$\mathbf{Adv}^{\mathrm{ind}}_{\mathcal{LF},\mathsf{A}} := \left|\mathbb{P}\left[\mathsf{A}(\phi, (t_a, t_c^0)) = 1\right] - \mathbb{P}\left[\mathsf{A}(\phi, (t_a, t_c^1)) = 1\right]\right|$$

where $(\phi, \psi) \leftarrow\!\!\$\ \mathsf{Gen}_{\mathrm{lf}}(1^\kappa), t_a \leftarrow\!\!\$\ \mathsf{A}(\phi), t_c^0 \leftarrow\!\!\$\ \mathcal{T}_c$ and $t_c^1 \leftarrow\!\!\$\ \mathsf{LTag}(\psi, t_a)$.

Evasiveness: *No PPT adversary* A *is able to generate a non-injective tag even given a lossy tag, i.e. the following quantity is negligible:*

$$\mathbf{Adv}^{\text{evasive}}_{\mathcal{LF},\mathsf{A}} := \mathbb{P}\left[\begin{array}{c}(t'_a,t'_c)\neq(t_a,t_c)\\(t'_a,t'_c)\in\mathcal{T}\setminus\mathcal{T}_{\text{inj}}\end{array} : \begin{array}{c}(\phi,\psi)\leftarrow\!\!\$\ \mathsf{Gen}_{\text{lf}}(1^\kappa);\\ t_a\leftarrow\!\!\$\ \mathsf{A}(\phi);t_c\leftarrow\!\!\$\ \mathsf{LTag}(\psi,t_a);\\(t'_a,t'_c)\leftarrow\!\!\$\ \mathsf{A}(\phi,(t_a,t_c))\end{array}\right].$$

Randomness extractors. An average-case strong randomness extractor.

Definition 8. *An efficient function* $\mathsf{Ext}:\mathcal{X}\times\mathcal{S}\to\mathcal{Y}$ *is an average-case* (δ,ϵ)*-strong extractor if for all pair of random variables* $(\mathbf{X},\mathbf{Z})$*, where* $\mathbf{X}$ *is defined over a set* $\mathcal{X}$ *and* $\widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Z})\geqslant\delta$*, we have*

$$(\mathbf{Z},\mathbf{S},\mathsf{Ext}(\mathbf{X},\mathbf{S}))\approx_\epsilon(\mathbf{Z},\mathbf{S},\mathbf{U}),$$

with $\mathbf{S}$ *uniform over* $\mathcal{S}$ *and* $\mathbf{U}$ *uniform over* $\mathcal{Y}$*.*

The encryption scheme. Consider now the following PKE scheme $\mathcal{PKE}=(\mathsf{Setup},\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ with message space $\mathcal{M}:=\{0,1\}^m$, based on a HPS $\mathcal{HPS}=(\mathsf{Gen}_{\text{hps}},\mathsf{Pub},\mathsf{Priv})$, on a OTLF $\mathcal{LF}=(\mathsf{Gen}_{\text{lf}},\mathsf{Eval},\mathsf{LTag})$ with domain $\mathcal{K}$, and on an average-case strong extractor $\mathsf{Ext}:\mathcal{K}\times\{0,1\}^d\to\{0,1\}^m$.

- $\underline{\mathsf{Setup}(1^\kappa)}$: Sample $pub_{\text{hps}}:=(aux,\mathcal{C},\mathcal{V},\mathcal{K},\mathcal{SK},\mathcal{PK},\Lambda_{(\cdot)},\mu)\leftarrow\!\!\$\ \mathsf{Gen}_{\text{hps}}(1^\kappa)$ and compute $(\phi,\psi)\leftarrow\!\!\$\ \mathsf{Gen}_{\text{lf}}(1^\kappa)$. Return $pub:=(pub_{\text{hps}},\phi)$. (Recall that all algorithms implicitly take pub as input.)
- $\underline{\mathsf{Gen}(1^\kappa)}$: Choose a random $sk\leftarrow\!\!\$\ \mathcal{SK}$, define $pk=\mu(sk)$, and return (pk,sk).
- $\underline{\mathsf{Enc}(pk,M)}$: Sample $C\leftarrow\!\!\$\ \mathcal{V}$ (with witness w), $S\leftarrow\!\!\$\ \{0,1\}^d$, and a core tag $t_c\leftarrow\!\!\$\ \mathcal{T}_c$. Compute $K:=\mathsf{Pub}(pk,C,w)$, $\Phi:=\mathsf{Ext}(K,S)\oplus M$, and $\Pi:=\mathsf{Eval}(\phi,(t_a,t_c),K)$ where $t_a:=(C,S,\Phi)$. Output $\hat{C}:=(C,S,\Phi,\Pi,t_c)$.
- $\underline{\mathsf{Dec}(sk,\hat{C})}$: Parse $\hat{C}:=(C,S,\Phi,\Pi,t_c)$. Compute $\hat{K}:=\mathsf{Priv}(sk,C)$ and check if $\mathsf{Eval}(\phi,t,\hat{K})=\Pi$ where $t:=((C,S,\Phi),t_c)$. If the check fails, reject and output $\perp$; else output $M:=\Phi\oplus\mathsf{Ext}(\hat{K},S)$.

Theorem 2. *Let* $\kappa\in\mathbb{N}$ *be the security parameter. Assume that* $\mathcal{HPS}$ *is* ϵ*-almost universal,* $\mathcal{LF}$ *is an* ℓ_{lf}*-OTLF with domain* $\mathcal{K}$*, and* Ext *is an average-case* (δ,ϵ')*-strong extractor for a negligible function* ϵ'*. Let* $s=s(\kappa)$ *and* $p=p(\kappa)$ *be parameters such that* $s\leqslant\log|\mathcal{SK}|$ *and* $p\geqslant\log|\mathcal{PK}|$ *for any* $\mathcal{SK},\mathcal{PK}$ *generated by* $\mathsf{Gen}_{\text{hps}}(1^\kappa)$*, and define* $\alpha=-\log\epsilon$ *and* $\beta=s-\alpha$*.*

For any $\delta\leqslant\alpha-\tau(p+\beta+\kappa)-\ell_{\text{lf}}-\ell$ *the PKE scheme* $\mathcal{PKE}$ *described above is* (τ,ℓ)*-BLT-IND-CCA with* $\ell+\tau(p+\beta+\kappa)\leqslant\alpha-\ell_{\text{lf}}$*.*

4.2 Security Proof

We consider a sequence of mental experiments, starting with the initial game $\mathbf{Exp}^{\text{blt-cca}}_{\mathcal{PKE},\mathsf{A}}(\kappa,\ell,\tau)$ which for simplicity we denote by $\mathbf{G}_0$.

Game $\mathbf{G}_0$. This is exactly the game of Definition 1, where $\mathcal{PKE}$ is the PKE scheme described above. In particular, upon input the i-th tampering query T_i the modified secret key $sk'_i = T_i(sk)$ is computed (where sk is the original secret key). Hence, the answer to a query $(i, \hat{C})$ to oracle Dec^* is computed by parsing $\hat{C} := (C, S, \Phi, \Pi, t_c)$, computing $\hat{K} := \mathsf{Priv}(sk'_i, C)$, and checking $\Pi = \mathsf{Eval}(\phi, ((C, S, \Phi), t_c), \hat{K})$; if the check fails the answer is $\perp$ and otherwise the answer is $M := \Phi \oplus \mathsf{Ext}(\hat{K}, S)$.

Game $\mathbf{G}_1$. We change the way the tag t_c^* corresponding to the challenge ciphertext is computed, namely we now let $t_c^* \leftarrow \mathsf{LTag}(\psi, t_a^*)$ (i.e., the tag $t^* = (t_a^*, t_c^*) \in \mathcal{T}_{\text{loss}}$ is now lossy).

Game $\mathbf{G}_2$. We add an extra check to the decryption oracle. Namely, upon input a decryption query $(i, (C, S, \Phi, \Pi, t_c))$ we check whether $t_a := (C, S, \Phi)$ and t_c satisfy $(t_a, t_c) = (t_a^*, t_c^*)$ (where t_a^* and t_c^* are the auxiliary and core tag corresponding to the challenge ciphertext). If the check succeeds, the oracle returns $\perp$. Notice that t_a^* and t_c^* are initially set to $\perp$, and remain equal to $\perp$ until the challenge ciphertext is generated.

Game $\mathbf{G}_3$. We change the way the challenge ciphertext is computed. Namely, we now compute the value K^* as $K^* := \mathsf{Priv}(sk, C^*)$.

Game $\mathbf{G}_4$. We change the way the challenge ciphertext is computed. Namely, we now sample C^* as $C^* \leftarrow\!\!\$\ \mathcal{C} \setminus \mathcal{V}$.

Game $\mathbf{G}_5$. We add an extra check to the decryption oracle; the check is performed only for decryption queries corresponding to tampered secret keys (i.e., $i \geq 1$). At setup, the experiment initializes an additional set $\mathcal{Q}' \leftarrow \emptyset$. Denote by $\mathbf{V}$ the random variable containing all the answers from the decryption and leakage oracles, and define the quantity

$$\gamma_i(\kappa) := \mathbb{H}_\infty(\mathbf{SK}'_i | \mathbf{V} = v, \{\mathbf{SK}'_j = sk'_j\}_{j \in \mathcal{Q}'}, \{\mathbf{PK}'_j = pk'_j\}_{j \in [\tau] \cup \{0\}})$$

where we write $\mathbf{SK}'_i$ for the random variable of the i-th tampered secret key and $\mathbf{PK}'_i$ for the random variable of the corresponding public key (by default $pk'_i = \perp$ if sk'_i is undefined and $pk'_0 = pk$).
Upon input a decryption query $(i, (C, S, \Phi, \Pi, t_c))$ such that $i \geq 1$ we proceed exactly as in $\mathbf{G}_4$ but, for all ciphertexts such that $C \in \mathcal{C} \setminus \mathcal{V}$, in case the decryption oracle did not already return $\perp$, we additionally check whether $\gamma_i(\kappa) \leq \beta(\kappa) + \log^2 \kappa$; if that happens, we add the index i to the set $\mathcal{Q}'$ and otherwise we do not modify $\mathcal{Q}'$ and we additionally answer the decryption query with $\perp$.

Game $\mathbf{G}_6$. We change the way decryption queries corresponding to the original secret key are answered. Namely, upon input a decryption query $(0, (C, S, \Phi, \Pi, t_c))$ we proceed as in $\mathbf{G}_5$ but, in case $C \in \mathcal{C} \setminus \mathcal{V}$, we answer the query with $\perp$.

Game $\mathbf{G}_7$. We change the way the challenge ciphertext is computed. Namely, we now sample $\Phi^* \leftarrow\!\!\$\ \{0,1\}^m$. Notice that the challenge ciphertext is now independent of the message being encrypted.

Next, we turn to showing that the above defined games are indistinguishable. In what follows, given a ciphertext $\hat{C} = (C, S, \Phi, \Pi, t_c)$, we say that $\hat{C}$ is *valid* if $C \in \mathcal{V}$ (i.e., if C is a valid ciphertext for the underlying HPS).

Lemma 9. *For all PPT adversaries* A *there exists a negligible function* $\nu_{0,1} : \mathbb{N} \to [0,1]$ *such that* $|\mathbb{P}[\mathbf{G}_0(\kappa) = 1] - \mathbb{P}[\mathbf{G}_1(\kappa) = 1]| \leq \nu_{0,1}(\kappa)$.

Proof. We prove a stronger statement, namely that $\mathbf{G}_0(\kappa) \approx_c \mathbf{G}_1(\kappa)$. By contradiction, assume there exists a PPT distinguisher $\mathsf{D}_{0,1}$ and a polynomial $p_{0,1}(\cdot)$ such that, for infinitely many values of $\kappa \in \mathbb{N}$, we have that $\mathsf{D}_{0,1}$ distinguishes between $\mathbf{G}_0$ and $\mathbf{G}_1$ with probability at least $\geq 1/p_{0,1}(\kappa)$. We construct an adversary $\mathsf{A}_{0,1}$ breaking the indistinguishability property of the underlying OTLF $\mathcal{LF}$. At the beginning, adversary $\mathsf{A}_{0,1}$ receives the evaluation key ϕ from its own challenger, and simulates the entire experiment $\mathbf{G}_0$ with $\mathsf{D}_{0,1}$ by sampling all other parameters by itself; notice that this can be done because $\mathbf{G}_0$ does not depend on the secret trapdoor ψ. Whenever $\mathsf{D}_{0,1}$ outputs (M_0, M_1), adversary $\mathsf{A}_{0,1}$ samples t_a^* as defined in $\mathbf{G}_0$ and returns t_a^* to its own challenger. Upon receiving a value t_c^* from the challenger, $\mathsf{A}_{0,1}$ embeds t_c^* in the challenge ciphertext, and keeps simulating all queries done by $\mathsf{D}_{0,1}$ as before. Finally, $\mathsf{A}_{0,1}$ outputs the same as $\mathsf{D}_{0,1}$.

We observe that $\mathsf{A}_{0,1}$ perfectly simulates the decryption oracle (which is identical in both $\mathbf{G}_0$ and $\mathbf{G}_1$). Moreover, depending on the challenge tag t_c^* being random or lossy, the distribution of the challenge ciphertext produced by $\mathsf{A}_{0,1}$ is identical to that of either $\mathbf{G}_0$ or $\mathbf{G}_1$. Thus, $\mathsf{A}_{0,1}$ retains the same advantage as that of $\mathsf{D}_{0,1}$. This concludes the proof. □

Lemma 10. $\mathbf{G}_1 \equiv \mathbf{G}_2$.

Proof. Notice that $\mathbf{G}_1$ and $\mathbf{G}_2$ only differ in how decryption queries such that $(t_a, t_c) = (t_a^*, t_c^*)$ are answered. Clearly, such queries are answered identically in the two games for all decryption queries before the generation of the challenge ciphertext. As for decryption queries after the challenge ciphertext has been computed, we distinguish two cases: (i) $\Pi = \Pi^*$, and (ii) $\Pi \neq \Pi^*$. In case (i) we get that $\hat{C} = \hat{C}^*$, and thus both games return $\perp$. In case (ii), note that $\mathbf{G}_1$ checks whether $\Pi = \mathsf{Eval}(\phi, (t_a^*, t_c^*), \mathsf{Priv}(sk_i', C^*))$ and thus it returns $\perp$ whenever $\Pi \neq \Pi^*$. Hence, the two games are identically distributed. □

Lemma 11. $\mathbf{G}_2 \equiv \mathbf{G}_3$.

Proof. The difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ is only syntactical, as $\mathsf{Priv}(sk, C^*) = K^* = \mathsf{Pub}(pk, C^*, w)$ by correctness of the underlying HPS. □

Lemma 12. *For all PPT adversaries* A*, there exists a negligible function* $\nu_{3,4} : \mathbb{N} \to [0,1]$ *such that* $|\mathbb{P}[\mathbf{G}_3(\kappa) = 1] - \mathbb{P}[\mathbf{G}_4(\kappa) = 1]| \leq \nu_{3,4}(\kappa)$.

Proof. We prove a stronger statement, namely that $\mathbf{G}_3(\kappa) \approx_c \mathbf{G}_4(\kappa)$. By contradiction, assume there exists a PPT distinguisher $\mathsf{D}_{3,4}$ and a polynomial $p_{3,4}(\cdot)$ such that, for infinitely many values of $\kappa \in \mathbb{N}$, we have that $\mathsf{D}_{3,4}$ distinguishes

between $\mathbf{G}_3$ and $\mathbf{G}_4$ with probability at least $\geq 1/p_{3,4}(\kappa)$. We construct a PPT adversary $\mathsf{A}_{3,4}$ solving the set membership problem of the underlying HPS. $\mathsf{A}_{3,4}$ receives as input pub_{hps} and a challenge C^* such that either $C^* \leftarrow\!\!\$\ \mathcal{V}$ or $C^* \leftarrow\!\!\$\ \mathcal{C} \setminus \mathcal{V}$. Hence, $\mathsf{A}_{3,4}$ perfectly simulates the challenger for $\mathsf{D}_{3,4}$, by sampling all required parameters by itself, and embeds the value C^* in the challenge ciphertext. In case $C^* \leftarrow\!\!\$\ \mathcal{V}$ we get exactly the same distribution as in $\mathbf{G}_3$, and in case $C^* \leftarrow\!\!\$\ \mathcal{C} \setminus \mathcal{V}$ we get exactly the same distribution as in $\mathbf{G}_4$. Hence, $\mathsf{A}_{3,4}$ retains the same advantage as that of $\mathsf{D}_{3,4}$. This finishes the proof. □

For the j-th query $(i, \hat{C})$ to the decryption oracle, such that $\hat{C} = (C, S, \Phi, \Pi, t_c)$, we let Inj_j be the event that the corresponding core tag t_c is injective. We also define $Inj := \bigwedge_{j\in[q]} Inj_j$ where $q \in poly(\kappa)$ is the total number of decryption queries asked by the adversary.

Lemma 13. *For all PPT adversaries* A *there exists a negligible function* $\nu_4 : \mathbb{N} \to [0, 1]$ *such that:* $|\mathbb{P}[\mathbf{G}_4(\kappa) = 1] - \mathbb{P}[\mathbf{G}_4(\kappa) = 1|Inj]| \leq \nu_4(\kappa)$.

Proof. The lemma follows by a simple reduction to the evasiveness property of the OTLF $\mathcal{LF}$. By contradiction, assume there exists a PPT adversary A_4 and a polynomial $p_4(\cdot)$ such that $|\mathbb{P}[\mathbf{G}_4(\kappa) = 1] - \mathbb{P}[\mathbf{G}_4(\kappa) = 1|Inj]| \geq 1/p_4(\kappa)$ for infinitely many values of $\kappa \in \mathbb{N}$. This implies:

$$1/p_4(\kappa) \leq |\mathbb{P}[\mathbf{G}_4(\kappa) = 1] - \mathbb{P}[\mathbf{G}_4(\kappa) = 1|Inj]| \leq \mathbb{P}[Inj].$$

We build a PPT adversary B_4 with non-negligible advantage in the evasiveness game. The adversary B_4 receives as input a public key ϕ for the OTLF and perfectly simulates a run of game $\mathbf{G}_4$ for A_4 by sampling all parameters by itself. After A_4 returns (M_0, M_1), adversary B_4 samples t_a^* as defined in $\mathbf{G}_4$, and forwards t_a^* to its own challenger. Upon receiving t_c^* from the challenger, B_4 embeds t_c^* in the challenge ciphertext for A_4.

Let $\mathcal{Q}$ be the list of decryption queries made by A_4. At the end of the simulation, adversary $\mathcal{B}_4$ picks uniformly at random a ciphertext $\hat{C} = (C, S, \Phi, t_c)$ from the list $\mathcal{Q}$ and outputs the tuple $(t_a := (C, S, \Phi), t_c)$. Clearly, the advantage of B_4 in the evasiveness game is equal to the probability of event Inj happening times the probability of guessing one of the ciphertexts containing a non-injective tag. Let $q(\kappa) \in poly(\kappa)$ be the total number of decryption queries made by A_4. We have obtained,

$$\mathbf{Adv}^{\text{evasive}}_{\mathcal{LF},\mathsf{B}_4}(\kappa) \geq \mathbb{P}[Inj]/q(\kappa) \geq 1/q(\kappa) \cdot 1/p_4(\kappa),$$

which is a non-negligible quantity. This concludes the proof. □

From now on, all of our arguments will be solely information-theoretic, and hence we do not mind if the remaining experiments will no longer be efficient.

Lemma 14. *For all (possibly unbounded) adversaries* A *making polynomially many decryption queries, there exists a negligible function* $\nu_{4,5} : \mathbb{N} \to [0, 1]$ *such that* $|\mathbb{P}[\mathbf{G}_4(\kappa) = 1|Inj] - \mathbb{P}[\mathbf{G}_5(\kappa) = 1|Inj]| \leq \nu_{4,5}(\kappa)$.

Proof. Recall that $\mathbf{G}_4$ and $\mathbf{G}_5$ differ only in the way decryption queries are handled. In particular, upon input a query $(i,(C,S,\Phi,\Pi,t_c))$ such that $i \geq 1$ and $C \in \mathcal{C} \setminus \mathcal{V}$, the decryption oracle in $\mathbf{G}_5$ checks whether $\gamma_i(\kappa) \leqslant \beta(\kappa) + \log^2 \kappa$. In case that happens, $\mathbf{G}_5$ proceeds identically to $\mathbf{G}_4$ and additionally updates the set $\mathcal{Q}'$ by including the index i; otherwise $\mathbf{G}_5$ answers the query with $\perp$. Intuitively, the set $\mathcal{Q}'$ keeps track of the tampered secret keys that did not return $\perp$ upon input an *invalid* ciphertext; the variable $\gamma_i(\kappa)$, instead, measures the conditional min-entropy of the i-th tampered secret key conditioned on all values returned by the decryption and leakage oracles, all tampered secret keys within the set $\mathcal{Q}'$, and all public keys corresponding to the tampered secret keys generated so far.

It follows that the distribution of the two games differ only in case the adversary makes a decryption query $(i,(C,S,\Phi,\Pi,t_c))$ such that: (i) $\gamma_i(\kappa) > \beta(\kappa) + \log^2 \kappa$; (ii) $C \in \mathcal{C} \setminus \mathcal{V}$; (iii) $\Pi = \mathsf{Eval}(\phi,(t_a,t_c),\mathsf{Priv}(sk'_i,C))$. Let Bad be the event that any (possibly unbounded) adversary makes a decryption query as above. Clearly,

$$|\mathbb{P}\left[\mathbf{G}_4(\kappa)=1|Inj\right] - \mathbb{P}\left[\mathbf{G}_5(\kappa)=1|Inj\right]| \leq \mathbb{P}\left[Bad|Inj\right].$$

For all $j \in [q]$, let Bad_j be the event that Bad happens for the j-th decryption query, which as usual we denote by $(i,(C,S,\Phi,\Pi,t_c)))$. Since we are conditioning on Inj, we have that there exists a unique value K that is the pre-image of Π under function $\mathsf{Eval}(\phi,(t_a,t_c),\cdot)$. Thus, by averaging over all the possible views for the adversary, we obtain:

$$\begin{aligned}
\mathbb{P}\left[Bad_j|Inj\right] &= \mathbb{P}\left[\mathsf{Priv}(\mathbf{SK}'_i,C)) = K\right] \\
&= \sum_{v,pk} \mathbb{P}\left[\mathbf{V}=v,\mathbf{PK}=pk\right] \cdot \mathbb{P}\left[\mathsf{Priv}(\mathbf{SK}'_i,C)) = K|\mathbf{V}=v,\mathbf{PK}=pk\right] \\
&\leqslant \sum_{v,pk} \mathbb{P}\left[\mathbf{V}=v,\mathbf{PK}=pk\right] \cdot 2^{-\mathbb{H}_\infty(\mathsf{Priv}(\mathbf{SK}'_i,C)|\mathbf{V}=v,\mathbf{PK}=pk)}.
\end{aligned}$$

Define the set $\mathcal{SK}^*_{K,C} := \{sk : \ \mathsf{Priv}(sk,C) = K \ \wedge \ pk = \mu(sk)\}$. We can write:

$$\begin{aligned}
&2^{-\mathbb{H}_\infty(\mathsf{Priv}(\mathbf{SK}'_i,C)|\mathbf{V}=v,\mathbf{PK}=pk)} \\
&= \max_K \mathbb{P}\left[\mathsf{Priv}(\mathbf{SK}'_i,C) = K|\mathbf{V}=v,\mathbf{PK}=pk\right] \\
&= \max_K \mathbb{P}\left[\mathbf{SK}'_i \in \mathcal{SK}^*_{K,C}|\mathbf{V}=v,\mathbf{PK}=pk\right] \\
&\leqslant \max_{K,sk'_i} |\mathcal{SK}^*_{K,C}| \cdot \mathbb{P}\left[\mathbf{SK}'_i = sk'_i|\mathbf{V}=v,\mathbf{PK}=pk\right] \\
&= \max_K |\mathcal{SK}^*_{K,C}| \cdot 2^{-\mathbb{H}_\infty(\mathbf{SK}'_i|\mathbf{V}=v,\mathbf{PK}=pk)} \\
&= \max_K \frac{|\mathcal{SK}^*_{K,C}|}{|\mathcal{SK}|} \cdot |\mathcal{SK}| \cdot 2^{-\mathbb{H}_\infty(\mathbf{SK}'_i|\mathbf{V}=v,\mathbf{PK}=pk)} \\
&\leqslant \epsilon \cdot |\mathcal{SK}| \cdot 2^{-\mathbb{H}_\infty(\mathbf{SK}'_i|\mathbf{V}=v,\mathbf{PK}=pk)} \leqslant \epsilon \cdot |\mathcal{SK}| \cdot 2^{-\beta(\kappa)-\log^2\kappa} = 2^{-\log^2\kappa},
\end{aligned}$$

where in the last line we used the ϵ-almost universality of the underlying HPS, together with the fact that $\gamma_i(\kappa) > \beta(\kappa) + \log^2 \kappa$. Finally, by a union bound over all decryption queries, we obtain that there exists a negligible function $\nu_{4,5} : \mathbb{N} \to [0,1]$ such that $\mathbb{P}[Bad|Inj] \leqslant q \cdot 2^{-\log^2 \kappa} \leq \nu_{4,5}(\kappa)$, which concludes the proof of the lemma. □

Lemma 15. *For all (possibly unbounded) adversaries* A*, there exists a negligible function* $\nu_{5,6} : \mathbb{N} \to [0,1]$ *such that* $|\mathbb{P}[\mathbf{G}_5(\kappa) = 1|Inj] - \mathbb{P}[\mathbf{G}_6(\kappa) = 1|Inj]| \leq \nu_{5,6}(\kappa)$.

Proof. Let *Bad* be the event that the adversary submits a decryption query $(0, (C, S, \Phi, \Pi, t_c))$ such that: (i) $C \in \mathcal{C} \setminus \mathcal{V}$; (ii) $\Pi = \mathsf{Eval}(\phi, (t_a, t_c), \mathsf{Priv}(sk, C))$. Similarly to the proof of the previous lemma, it suffices to bound the probability of the event *Bad* conditioned on *Inj*. Denote by $(0, (C, S, \Phi, \Pi, t_c))$ the first decryption query (w.r.t. the original secret key) that triggers event *Bad*. Recall that the view of adversary A in a run of game $\mathbf{G}_5$ consists of its own coin tosses, the public key pk, the answers to all queries to the decryption and leakage oracles, and the challenge ciphertext $\hat{C}^*$. In what follows, we write $\mathbf{L}$ for the random variable corresponding to the leakage queries; furthermore, for an index $i \in [\tau]$, we denote with $\mathbf{D}_i$ the random variable corresponding to all decryption queries relative to the i-th tampered secret key. Note that we can partition $\mathbf{D}_i$ in two parts: $\mathbf{D}_i^-$ for all decryption queries (w.r.t. the i-th tampered secret key) with an invalid ciphertext, and $\mathbf{D}_i^+$ for all decryption queries (w.r.t. the i-th tampered secret key) with a valid ciphertext. We also write $\mathbf{W}$ for the random variable corresponding to the overall view in game $\mathbf{G}_5$.

As in the previous lemma, since we are conditioning on event *Inj*, it suffices to analyze the conditional average min-entropy of $\mathsf{Priv}(\mathbf{SK}, C)$ conditioned on the adversarial view.

$$
\begin{aligned}
&\widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{W}) \\
&\geqslant \widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i\}_{i\in[\tau]}, \mathbf{L}, \hat{\mathbf{C}}^*) && (2)\\
&\geqslant \widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i\}_{i\in[\tau]}) - \ell_{\mathrm{lf}} - \ell && (3)\\
&= \widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i^+\}_{i\in[\tau]}, \{\mathbf{D}_i^-\}_{i\in\mathcal{Q}'}, \mathcal{Q}') - \ell_{\mathrm{lf}} - \ell && (4)\\
&\geqslant \widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i^+\}_{i\in[\tau]}, \{\mathbf{SK}_i'\}_{i\in\mathcal{Q}'}, \mathcal{Q}') - \ell_{\mathrm{lf}} - \ell. && (5)
\end{aligned}
$$

Here, Eq. (2) uses the fact that the coin tosses of the adversary are independent of $\mathbf{SK}$, Eq. (3) follows by the chain rule for conditional average min-entropy (cf. Lemma 1), Eq. (4) uses the fact that, by definition of $\mathbf{G}_5$, all decryption queries for keys outside $\mathcal{Q}'$ and with an invalid ciphertext are answered with $\perp$, and Eq. (5) follows by the fact that $\mathbf{D}_i^-$ is a deterministic function of $\mathbf{SK}_i'$.

Let $\mathcal{Q}' = \{i_1, \ldots, i_{q'}\}$, as defined in game $\mathbf{G}_5$. Since the fact that $sk_{i_{q'}} \in \mathcal{Q}'$ implies that $\mathbb{H}_\infty(\mathbf{SK}'_{i_{q'}}|\mathbf{W}) \leqslant \beta(\kappa) + \log^2 \kappa$, we can first apply Lemma 2 and then Lemma 1 to obtain

$$
\begin{aligned}
&\widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i^+\}_{i\in[\tau]}, \{\mathbf{SK}_i'\}_{i\in\mathcal{Q}'}, \mathcal{Q}') \\
&\geqslant \widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i^+\}_{i\in[\tau]}, \{\mathbf{SK}_i'\}_{i\in\mathcal{Q}''}, \mathcal{Q}'') - \beta(\kappa) - \log^2 \kappa - \log|\mathcal{Q}'|,
\end{aligned}
$$

where $\mathcal{Q}'' := \mathcal{Q}' \setminus \{i_{q'}\}$. Notice to apply Lemma 2 we need to condition on $sk_{i_{q'}} \in \mathcal{Q}'$, however, such condition holds with probability 1 and by conditioning on a sure event the min-entropy does not change. By iterating the above argument for each key in $\mathcal{Q}'$:

$$\begin{aligned}\widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i^+\}_{i\in[\tau]}, \{\mathbf{SK}'_i\}_{i\in\mathcal{Q}'}, \mathcal{Q}') \\ \geqslant \widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i^+\}_{i\in[\tau]}) - \tau \cdot (\beta + \log^2 \kappa + \log \tau),\end{aligned} \tag{6}$$

and relying on the fact that the answer to decryption queries for a valid ciphertext and w.r.t. index $j \in [\tau]$ can be computed using the "tampered" projection key $pk'_i = \mu(sk'_i)$, we obtain

$$\begin{aligned}\widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{D}_i^+\}_{i\in[\tau]}) &\geqslant \widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{PK}, \{\mathbf{PK}'_i\}_{i\in[\tau]}) \\ &\geqslant \alpha - \tau \cdot p,\end{aligned} \tag{7}$$

where Eq. (7) follows by Lemma 1 and Lemma 8. Combining together Eq. (5), Eq. (6), and Eq. (7), yields:

$$\widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C)|\mathbf{W}) \geqslant \alpha - \tau \cdot (p + \beta + \log^2 \kappa + \log \tau) - \ell_{\mathrm{lf}} - \ell.$$

It follows that the decryption oracle in game $\mathbf{G}_5$ does not reject the first invalid ciphertext with probability at most $\epsilon \cdot 2^{\tau(p+\beta+\log^2\kappa+\log\tau)+\ell_{\mathrm{lf}}+\ell}$. A generalization of this argument implies that, for all $j \in [q]$, the probability that the decryption oracle does not reject the j-th decryption query of type $(0, \cdot)$ containing an invalid ciphertext is at most $2^{\tau(p+\beta+\log^2\kappa+\log\tau)+\ell_{\mathrm{lf}}+\ell}/(1/\epsilon - q(\kappa))$. Finally, by a union bound over the total number of decryption queries, there exists a negligible function $\nu_{5,6} : \mathbb{N} \to [0, 1]$ such that:

$$\begin{aligned}\mathbb{P}\left[Bad|Inj\right] &\leq \frac{q \cdot 2^{\tau(p+\beta+\log^2\kappa+\log\tau)+\ell_{\mathrm{lf}}+\ell}}{1/\epsilon - q} \\ &\leq \epsilon \cdot e^{-q\epsilon} \cdot 2^{\tau(p+\beta+\log^2\kappa+\log\tau)+\ell_{\mathrm{lf}}+\ell+\log q} \\ &\leqslant 2^{-(\alpha - q\epsilon(\kappa) - \tau(p+\beta+\log^2\kappa\log\tau) - \ell_{\mathrm{lf}} - \ell - \log q)} \\ &\leq \nu_{5,6}(\kappa).\end{aligned}$$

where the last inequality follows by the fact that $\alpha \geqslant \ell + \ell_{\mathrm{lf}} + \tau(p + \beta + \kappa)$ and additionally $\kappa - \log^2 \kappa - \log \tau - \log q/\tau - q\epsilon/\tau \in \omega(\log \kappa)$. □

Lemma 16. *For all (possibly unbounded) adversaries* A*, there exists a negligible function* $\nu_{6,7} : \mathbb{N} \to [0, 1]$ *such that* $|\mathbb{P}\left[\mathbf{G}_6(\kappa) = 1|Inj\right] - \mathbb{P}\left[\mathbf{G}_7(\kappa) = 1|Inj\right]| \leq \nu_{6,7}(\kappa)$.

Proof. We analyze the conditional average min-entropy of $\mathsf{Priv}(\mathbf{SK}, C^*)$ conditioned on the view of the adversary. By a previous argument, we can write:

$$\widetilde{\mathbb{H}}_\infty(\mathsf{Priv}(\mathbf{SK}, C^*)|\mathbf{W}) \geqslant \alpha - \tau \cdot (p + \beta + \log^2 \kappa + \log \tau) - \ell_{\mathrm{lf}} - \ell,$$

and thus the statement follows by our choice of parameters for the strong average-case extractor. □

The statement of the theorem now follows by combining the above lemmas together with the fact that in $\mathbf{G}_7$ the challenge ciphertext is independent of the hidden bit b, and thus $\mathbb{P}\left[\mathbf{G}_7(\kappa)|Inj\right] = 1/2$ for all (even unbounded) adversaries. This finishes the proof.

4.3 Concrete Instantiations

The ratio $\frac{\alpha-\ell-\ell_{\mathrm{lf}}}{p+\beta}$ plays an important role in evaluating the tampering rate of a given instantiation. Ideally, we would like to have an HPS where α is as big as possible while p and $\beta = \alpha - s$ are as small as possible. Below, we give an instantiation based on the Refined Subgroup Indistinguishability (RSI) assumption.

Instantiation based on RSI. Let $\xi \in \mathbb{N}$ be a parameter. For security parameter $\kappa \in \mathbb{N}$, let p and q be primes of size respectively κ bits and $\xi \cdot \kappa$ bits and define $\bar{p} = 2pq+1$. For this choice of parameters, we have that $\mathbb{Z}_{\bar{p}}^*$ has a unique subgroup of order $N = pq$. Denote by $\mathbb{QR}_{\bar{p}}$ the set of quadratic residues modulo $\bar{p}$; the group $\mathbb{QR}_{\bar{p}}$ can be decomposed as a direct product of $\mathbb{G}_p \times \mathbb{G}_q$ where $\mathbb{G}_p$ and $\mathbb{G}_q$ are cyclic groups of prime order p and q (respectively).

For random $x, y \leftarrow\!\!\$\, \mathbb{Z}_{\bar{p}}^*$, one can show that, with overwhelming probability, $g = x^q \bmod \bar{p}$ and $h = y^p \bmod \bar{p}$ are generators of $\mathbb{G}_p$ and $\mathbb{G}_q$ (respectively). Let $pub_{\mathrm{rsi}} := (\mathbb{QR}_{\bar{p}}, \bar{p}, g, h)$. The RSI assumption over $\mathbb{QR}_{\bar{p}}$ states that for all PPT adversary A the following quantity is negligible in the security parameter:

$$\left|\mathbb{P}\left[\mathsf{A}(pub_{\mathrm{rsi}}, g^x \bmod \bar{p}) :\ x \leftarrow\!\!\$\, \mathbb{Z}_{\bar{p}}\right] - \mathbb{P}\left[\mathsf{A}(pub_{\mathrm{rsi}}, y) :\ y \leftarrow\!\!\$\, \mathbb{QR}_{\bar{p}}\right]\right|.$$

The RSI assumption over $\mathbb{QR}_{\bar{p}}$ is conjectured to hold if factoring $N = pq$ is hard [45]. We can derive a HPS as follow. We set $\mathcal{C} := \mathbb{QR}_{\bar{p}}$, $\mathcal{V} := \mathbb{G}_p$, $\mathcal{SK} := \mathbb{Z}_{\bar{p}}$, and $\mathcal{PK} := \mathbb{G}_p$. Given a random secret key $sk \leftarrow\!\!\$\, \mathcal{SK}$, the corresponding public key pk is computed as $\mu(sk) := g^{sk} \bmod \bar{p}$. Algorithm Pub, upon input $C := g^w$ (where w is the witness for $C \in \mathcal{V}$) and pk outputs $pk^w \bmod \bar{p}$. Algorithm Priv, upon input C and sk, outputs $\Lambda_{sk}(C) := C^{sk} \bmod \bar{p}$. It was shown in [50] that the above construction defines a $1/q$-almost universal HPS based on the RSI assumption. The work of [50] additionally presents a construction of a OTLF achieving $\ell_{\mathrm{lf}} := \log p$ based on the RSI assumption.

Finally, by instantiating the average-case strong extractor using universal hash functions as required by the left-over hash lemma [34] we note that the PKE scheme allows to encrypt messages with bit-length $m = O(\xi\kappa - \tau\kappa - \ell - \kappa)$. We obtain the following result:

Corollary 3. *Let $\bar{p}$ be as above. Under the RSI assumption over $\mathbb{QR}_{\bar{p}}$, for any $\xi(\kappa) = \omega(1)$, there exists a PKE scheme satisfying (τ, ℓ)-BTL-IND-CCA with tampering rate $\rho(\kappa) = O(1/\kappa - \frac{\ell}{\xi^2\kappa})$. The size of the secret key is $\Omega(\xi\kappa)$, and the PKE scheme allows to encrypt messages with bit-length $m = O(\xi\kappa - \tau\kappa - \ell - \kappa)$.*

5 Conclusions and Open Problems

We have shown new constructions of public-key cryptosystems with provable security guarantees against bounded leakage and tampering attacks. The proposed schemes are in the standard model, and can be instantiated efficiently under standard complexity assumptions.

There are several interesting problems left open by our work. First, our constructions only achieve sub-optimal tampering rate $\rho(\kappa) = O(1/\kappa)$, so it would be interesting to find alternative constructions achieving optimal rate in the standard model. Second, it would be interesting to combine related-key attacks with related-randomness attacks [47,48], where the adversary might force a cryptographic scheme to re-use (functions of) its own random coins; a promising idea in this direction is to combine our leakage-to-tamper reduction to so called fully leakage-resilient signatures [14,23], where the adversary can additionally leak on the random coins of the signature algorithm. Third, it remains open how to obtain CCA security for PKE against "after-the-fact" tampering and leakage, where both tampering and leakage can still occur after the challenge ciphertext is generated (in the spirit of [33]). Finally, one could try to come-up with new hash-proof systems meeting the requirements needed for our PKE instantiation under alternative hardness assumptions.

Acknowledgments. The authors would like to thank Jesper Buus Nielsen for an interesting conversation regarding the result in Section 4. Antonio Faonio was supported by European Research Council Starting Grant 279447.

References

1. Abdalla, M., Benhamouda, F., Passelègue, A.: An algebraic framework for pseudorandom functions and applications to related-key security. In: CRYPTO. pp. 388–409 (2015)
2. Applebaum, B.: Garbling XOR gates "for free" in the standard model. In: TCC. pp. 162–181 (2013)
3. Applebaum, B., Harnik, D., Ishai, Y.: Semantic security under related-key attacks and applications. In: Innovations in Computer Science. pp. 45–60 (2011)
4. Applebaum, B., Widder, E.: Related-key secure pseudorandom functions: The case of additive attacks. IACR Cryptology ePrint Archive 2014, 478 (2014), `http://eprint.iacr.org/2014/478`
5. Ball, M., Dachman-Soled, D., Kulkarni, M., Malkin, T.: Non-malleable codes for bounded depth, bounded fan-in circuits. In: EUROCRYPT. pp. 881–908 (2016)
6. Bellare, M., Cash, D.: Pseudorandom functions and permutations provably secure against related-key attacks. In: CRYPTO. pp. 666–684 (2010)
7. Bellare, M., Cash, D., Miller, R.: Cryptography secure against related-key attacks and tampering. In: ASIACRYPT. pp. 486–503 (2011)
8. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: EUROCRYPT. pp. 491–506 (2003)
9. Bellare, M., Meiklejohn, S., Thomson, S.: Key-versatile signatures and applications: RKA, KDM and joint Enc/Sig. In: EUROCRYPT. pp. 496–513 (2014)

10. Bellare, M., Paterson, K.G., Thomson, S.: RKA security beyond the linear barrier: IBE, encryption and signatures. In: ASIACRYPT. pp. 331–348 (2012)
11. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: CRYPTO. pp. 513–525 (1997)
12. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO. pp. 41–55 (2004)
13. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults (extended abstract). In: EUROCRYPT. pp. 37–51 (1997)
14. Boyle, E., Segev, G., Wichs, D.: Fully leakage-resilient signatures. J. Cryptology 26(3), 513–558 (2013)
15. Chen, Y., Qin, B., Zhang, J., Deng, Y., Chow, S.S.M.: Non-malleable functions and their applications. In: PKC. pp. 386–416 (2016)
16. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: CRYPTO. pp. 13–25 (1998)
17. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: EUROCRYPT. pp. 45–64 (2002)
18. Damgård, I., Faust, S., Mukherjee, P., Venturi, D.: Bounded tamper resilience: How to go beyond the algebraic barrier. In: ASIACRYPT. pp. 140–160 (2013)
19. Damgård, I., Faust, S., Mukherjee, P., Venturi, D.: The chaining lemma and its application. In: ICITS. pp. 181–196 (2015)
20. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. ASIACRYPT 2010, 613–631
21. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. 38(1), 97–139 (2008)
22. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In: Innovations in Computer Science. pp. 434–452 (2010)
23. Faonio, A., Nielsen, J.B., Venturi, D.: Mind your coins: Fully leakage-resilient signatures with graceful degradation. In: ICALP. pp. 456–468 (2015)
24. Faonio, A., Venturi, D.: Efficient public-key cryptography with bounded leakage and tamper resilience. Cryptology ePrint Archive, Report 2016/529 (2016), `http://eprint.iacr.org/2016/529`
25. Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: Continuous non-malleable codes. In: TCC. pp. 465–488 (2014)
26. Faust, S., Mukherjee, P., Venturi, D., Wichs, D.: Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In: EUROCRYPT. pp. 111–128 (2014)
27. Faust, S., Pietrzak, K., Venturi, D.: Tamper-proof circuits: How to trade leakage for tamper-resilience. In: ICALP. pp. 391–402 (2011)
28. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: CRYPTO. pp. 186–194 (1986)
29. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In: TCC. pp. 258–277 (2004)
30. Goldenberg, D., Liskov, M.: On related-secret pseudorandomness. In: TCC. pp. 255–272 (2010)
31. Goyal, V., O'Neill, A., Rao, V.: Correlated-input secure hash functions. In: TCC. pp. 182–200 (2011)
32. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: EUROCRYPT. pp. 415–432 (2008)

33. Halevi, S., Lin, H.: After-the-fact leakage in public-key encryption. In: TCC. pp. 107–124 (2011)
34. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. 28(4), 1364–1396 (1999)
35. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: CRYPTO. pp. 553–571 (2007)
36. Ishai, Y., Prabhakaran, M., Sahai, A., Wagner, D.: Private circuits II: Keeping secrets in tamperable circuits. In: EUROCRYPT. pp. 308–327 (2006)
37. Jafargholi, Z., Wichs, D.: Tamper detection and continuous non-malleable codes. In: TCC. pp. 451–480 (2015)
38. Kalai, Y.T., Kanukurthi, B., Sahai, A.: Cryptography with tamperable and leaky memory. In: CRYPTO. pp. 373–390 (2011)
39. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: ASIACRYPT. pp. 703–720 (2009)
40. Lewi, K., Montgomery, H.W., Raghunathan, A.: Improved constructions of PRFs secure against related-key attacks. In: ACNS. pp. 44–61 (2014)
41. Liu, F., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: CRYPTO. pp. 517–532 (2012)
42. Lu, X., Li, B., Jia, D.: Related-key security for hybrid encryption. In: Information Security. pp. 19–32 (2014)
43. Lucks, S.: Ciphers secure against related-key attacks. In: FSE. pp. 359–370 (2004)
44. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: CRYPTO. pp. 18–35 (2009)
45. Nieto, J.M.G., Boyd, C., Dawson, E.: A public key cryptosystem based on a subgroup membership problem. Des. Codes Cryptography 36(3), 301–316 (2005)
46. Otto, M.: Fault Attacks and Countermeasures. Ph.D. thesis, University of Paderborn, Germany (2006)
47. Paterson, K.G., Schuldt, J.C.N., Sibborn, D.L.: Related randomness attacks for public key encryption. In: PKC. pp. 465–482 (2014)
48. Paterson, K.G., Schuldt, J.C.N., Sibborn, D.L., Wee, H.: Security against related randomness attacks via reconstructive extractors. In: IMACC. pp. 23–40 (2015)
49. Qin, B., Liu, S.: Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In: ASIACRYPT. pp. 381–400 (2013)
50. Qin, B., Liu, S.: Leakage-flexible CCA-secure public-key encryption: Simple construction and free of pairing. In: PKC. pp. 19–36 (2014)
51. Qin, B., Liu, S., Yuen, T.H., Deng, R.H., Chen, K.: Continuous non-malleable key derivation and its application to related-key security. In: PKC. pp. 557–578 (2015)
52. Scott, M.: Authenticated ID-based key exchange and remote log-in with simple token and PIN number. IACR Cryptology ePrint Archive 2002, 164 (2002), `http://eprint.iacr.org/2002/164`
53. Shacham, H.: A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. IACR Cryptology ePrint Archive 2007, 74 (2007), `http://eprint.iacr.org/2007/074`
54. Wee, H.: Public key encryption against related key attacks. In: PKC. pp. 262–279 (2012)