Short Digital Signatures and ID-KEMs via Truncation Collision Resistance

Tibor Jager and Rafael Kurek

Paderborn University, Paderborn, Germany
{tibor.jager, rafael.kurek}@upb.de

Abstract. *Truncation collision resistance* is a simple non-interactive complexity assumption that seems very plausible for standard cryptographic hash functions like SHA-3. We describe how this assumption can be leveraged to obtain standard-model constructions of public-key cryptosystems that previously seemed to require a programmable random oracle. This includes the first constructions of identity-based key encapsulation mechanisms (ID-KEMs) and digital signatures over bilinear groups with full adaptive security and without random oracles, where a ciphertext or signature consists of only a single element of a prime-order group. We also describe a generic construction of ID-KEMs with full adaptive security from a scheme with very weak security ("selective and non-adaptive chosen-ID security"), and a similar generic construction for digital signatures.

Keywords: Identity-based encryption, digital signatures, random oracle model, extremely lossy functions, provable security.

1 Introduction

The random oracle model (ROM) [BR93] is often used to analyze the security of cryptosystems in a hypothetical setting, where a cryptographic hash function is modeled as an oracle that implements a truly random function. This provides a very strong handle for formal security proofs. For example, an adversary in this model has to explicitly query the oracle to evaluate the hash function, and it is possible to adaptively "program" the hash function to map certain input values to specific output values in the security proof. The random oracle is a hypothetical concept, used only in a security proof, but instantiated in practice with a standard cryptographic hash function, like SHA-3. This incurs the additional assumption that this hash function is "secure enough" for the given application.

Besides the well-known difficulty of instantiating random oracles [CGH98], the major drawback of this approach is that the random oracle essentially is a "perfect" hash function, which provides not only standard security properties, like onewayness and collision resistance, but essentially all imaginable security properties simultaneously. Therefore a security proof in the random oracle model does not explain which precise security properties of a hash function are actually

necessary or sufficient for a given application. This is very undesirable, as we want to understand the required security properties and we want to provide cryptanalysts with clearly-defined cryptanalytic goals to attack the "security" of cryptographic hash functions. Therefore the ROM is often seen as only a first step towards achieving provably-secure constructions in the standard model.

The only known security proofs for many important cryptographic constructions seem to inherently require an adaptively programmable random oracle [Nie02,FLR+10,HMS12,FF13]. There are many primitives for which it is still unknown whether and how they can be instantiated without random oracles, and where no standard-model security proofs based on classical complexity assumptions on cryptographic hash functions are known so far. Several previous works isolated specific properties of random oracles, such as programmability [HK08,HJK11,FHPS13,CFN15] or extreme lossyness [Zha16], and realized these with standard-model constructions of special-purpose functions and based on algebraic public-key techniques, which are relatively inefficient in comparison to standard cryptographic hash functions. In the instead, we ask:

Which reasonable, simple, and non-interactive complexity assumptions for standard cryptographic hash functions are sufficient to obtain instantiations of cryptographic tools that currently require the ROM?

Hence, we do not ask for new non-standard hash functions that realize specific properties of a random oracle, but for reasonable assumptions on *standard* hash functions that are sufficient to avoid the random oracle. In this sense, we follow a line of research initiated by Canetti in 1997 [Can97], and continued by the UCE approach introduced by Bellare *et al.* [BHK13] and continued by Farshim and Mittelbach [FM16].

Truncation collision resistance. Truncation collision resistance (TruCR) basically demands that there is no algorithm that finds collisions significantly faster than the standard birthday collision algorithm, even when (short) *prefixes* of hash values are considered. More precisely, let $H : \{0,1\}^* \to \{0,1\}^k$ be a cryptographic hash function, and write $H_j(x)$ to denote the first j bits of $H(x)$. Truncation collision resistance requires that two input values x, x' with $x \neq x'$ and $H_j(x) = H_j(x')$ cannot be found for *any* value of j with significantly better time-to-success ratio than the standard birthday collision algorithm.

In contrast to the ROM, this assumption provides an explicit and well-defined goal for the cryptanalysis of hash functions. It is a single simple assumption, rather than a complex family of UCE assumptions of [BHK13], and based on simple "symmetric-key techniques" (i.e., standard cryptographic hash functions).

Contributions. We show that truncation collision resistance enables several interesting applications that previously required a random oracle. This includes:

- Identity-based key encapsulation schemes (ID-KEMs) with very short ciphertexts (only a single group element) and full adaptive security.

– Short digital signatures over bilinear groups with very short signatures (only a single group element) and full adaptive security.
– Generic constructions of ID-KEMs and digital signatures with full adaptive security from ID-KEMs and signatures with extremely weak "selective and non-adaptive" security.

Due to the relatively large (but polynomially-bounded) security loss of our security proofs, the practical value of our constructions is limited, if tightness is taken into account. However, we see truncation collision resistance as a step towards avoiding the random oracle for cryptosystems with minimal overhead.

Leveraging truncation collision resistance. In order to sketch how truncation collision resistance can be used in a security proof, let us consider the case of short digital signatures as an example. We work in the bilinear group setting, where we have groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of prime order p, and an efficiently computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Signatures consist of only one element of $\mathbb{G}_1$. A secret key consist of $\ell = \log 4(k+1)$ elements $x_1, \ldots, x_\ell \in \mathbb{Z}_p$, where k is the security parameter. The corresponding public key consists of one element of $\mathbb{G}_1$ plus $4(k+1)$ elements of $\mathbb{G}_2$. Thus, public and secret keys are larger than for the random-oracle-based short signature scheme of Boneh, Lynn, and Shacham [BLS04], but the signature size is identical.

Computing a signature on a message m works as follows. For a cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^{4(k+1)}$, let us write $H_{2^j}(m)$ to denote the first 2^j bits of $H(m)$. In order to sign a message m, we first compute

$$G(m) = \prod_{j=1}^{\ell} (x_j + H_{2^j}(m)) \bmod p.$$

Note that one can perform this computation very efficiently, as it involves only elementary operations over $\mathbb{Z}_p$. Finally, the signature for m is

$$\sigma = g_1^{1/G(m)} \in \mathbb{G}_1,$$

where $g_1 \in \mathbb{G}_1$ is a generator. Thus, computing a signature requires to perform only a *single* exponentiation in $\mathbb{G}_1$, plus a small number of additional operations in $\mathbb{Z}_p$.

A signature can be verified by first computing $g_2^{G(m)} \in \mathbb{G}_2$ from the group elements contained in the public key, which involves $\mathcal{O}(k)$ operations in $\mathbb{Z}_p$, $\mathcal{O}(k)$ multiplications in $\mathbb{G}_2$, and then testing whether

$$e(\sigma, g_2^{G(m)}) \stackrel{?}{=} e(g_1, g_2).$$

Note that this test requires only a single application of the bilinear map e to compute $e(\sigma, g_2^{G(m)})$, because the term $e(g_1, g_2)$ is independent of the given message and signature, and can thus be precomputed.

In order to sketch how truncation collision resistance is used in the security proof, note that a signature has the form

$$\sigma = g_1^{1/\prod_{j=1}^{\ell}(x_j + H_{2^j}(m))}, \tag{1}$$

which can be viewed as an aggregation of ℓ signatures of the form

$$\sigma_j = g_1^{1/(x_j + H_{2^j}(m))}. \tag{2}$$

Let us view a signature σ as an ℓ-tuple $\sigma = (\sigma_1, \ldots, \sigma_\ell)$, where σ_j is as in (2). We describe later how these signatures can be aggregated to obtain our actual scheme. Note that each σ_j is a Boneh-Boyen signature [BB04c] over the first 2^j bits of $H(m)$. In the security proof, we will choose j such that it simultaneously achieves the following two properties.

1. The index j is *sufficiently small*. Let m^* be the message for which the assumed adversary $\mathcal{A}$ forges a signature. We want that j is small enough, such that that we can guess $H_{2^j}(m^*) \in \{0,1\}^{2^j}$ with reasonable success probability, even *before* the security experiment starts, and we are able to prepare signatures for *all other* values $\{0,1\}^{2^j} \setminus H_{2^j}(m^*)$.
2. At the same time, we will make sure that the index j is *sufficiently large*, such that it is "sufficiently unlikely" that the adversary finds a collision for H_{2^j}. More precisely, we want that it is "sufficiently unlikely" that the adversary ever requests a signature for a message m_i and then outputs a forgery for message m^* with $H_{2^j}(m_i) = H_{2^j}(m^*)$.

The main difficulty of our security analysis lies in the second property, therefore let us consider this one more closely. Truncation collision resistance basically guarantees that there is no algorithm that finds collisions with significantly better time-to-success ratio than the standard birthday collision algorithm, even when prefixes $H_{2^j}(x)$ of hash values $H(x)$ are considered. Of course we will not be able to choose j such that the probability that $\mathcal{A}$ finds a collision is *negligibly* small – at least not without sacrificing the first condition, which we cannot afford. However, we will be able to choose j such that we can argue that the probability that $\mathcal{A}$ finds a collision for H_{2^j} is at most $\epsilon/2$, where ϵ is the success probability of $\mathcal{A}$ in breaking our signature scheme. This is "sufficiently unlikely", because it means: while *sometimes* $\mathcal{A}$ may break the security of the signature scheme by finding a collision, at least *sometimes* (more precisely: with probability at least $\epsilon/2$) the adversary will also be able to break the signature scheme *without* finding a collision. This allows us to reduce the full EUF-CMA-security of our scheme to the SUF-naCMA-security of the underlying Boneh-Boyen scheme.

Since Boneh-Boyen signatures are not known to be efficiently aggregable in the sense of [BGLS03,LOS+06], we have to overcome another hurdle to obtain a short signature scheme. Note that computing a signature that satisfies (1) essentially yields a "polynomial in the exponent" in unkowns $x_1, \ldots, x_\ell$ of degree ℓ. In order to verify whether a given value σ indeed satisfies (1) using a bilinear

map, we therefore must to be able to compute group elements of the form

$$g^{x_1^{b_1} \cdots x_\ell^{b_\ell}}. \tag{3}$$

for all possible values of $b_1, \ldots, b_\ell \in \{0,1\}$. Note that these are 2^ℓ different values, but we have $\ell = \mathcal{O}(\log k)$ This allows us to include all required values of the form (3) in the public key, which yields a public key of size $\mathcal{O}(k)$.

On the necessity of using ℓ parallel copies. A signature σ in the construction described above essentially consists of ℓ aggregated copies of signatures σ_j, $j \in [\ell]$. The purpose of j is to find a balance the two properties described above, but the "right" choice of j depends on the given adversary: different adversaries may require a different value of j to achieve the right balance.

The fact that we essentially run several copies of the scheme in parallel may appear artificial, but it is the only way that we currently know to guarantee a suitable value of j for any possible adversary, and thus the only way we know to prove security against any efficient adversary. There are other schemes that essentially run many copies of an underlying scheme in parallel "only" to achieve provable security. This includes, for instance, pseudorandom functions [DS15,BH15] and digital signatures [BHJ+13,HW09a,BK10].

Comparison to other prefix-guessing techniques. The main difference between our approach and the prefix-guessing technique of [HW09b,BK10] is that we essentially guess a short prefix of the hash of the "target message" *directly*, exploiting the truncation collision resistance to argue that this hash cannot be equal to the hash of any chosen-message query. We do not have to know any chosen-message queries of the adversary to do this, which makes the technique also applicable to identity-based schemes like ID-KEMs. In contrast, the prefix-guessing technique of [HW09b,BK10] guesses the shortest prefix of the target message that is not equal to a prefix of a chosen-message query, which depends on the chosen-message queries made by the adversary and therefore can only be used to construct non-adaptively secure signatures (adaptive security is then achieved in a second step, e.g. using [EGM96]).

2 Truncation Collision-Resistant Hashing

Intuitively, truncation collision resistance (TruCR) means that there exists no algorithm which finds collisions with significantly better *work factor* [BR09] than the trivial birthday collision algorithm. This must hold even if the output of the hash function is truncated. This notion is strongly related to standard collision resistance [RS04], but to our best knowledge has not yet been formally defined or used as a basis for formal security proofs of cryptosystems.

Computational model. We consider algorithms as Turing machines, which operate on a binary alphabet and write their output bit-by-bit to an output tape,

where each written bit takes one elementary machine operation.[1] The *running time* of an algorithm is defined as the number of elementary operations performed by this machine.

Truncation collision resistance. Let $\mathcal{H} = \{H : \{0,1\}^* \to \{0,1\}^\alpha\}$ be a family of keyed hash functions and let H_i denote the function obtained by evaluating H and truncating its output to the first i bits. Thus, $H_1(x)$ consists of the first bit of $H(x)$, and $H_\alpha(x) = H(x)$ for all $x \in \{0,1\}^*$.

Definition 1. *We say that adversary $\mathcal{B}$ i-breaks the truncation collision resistance of $\mathcal{H}$, if it runs in time $t_\mathcal{B}$ and*

$$\Pr_{H \xleftarrow{\$} \mathcal{H}} \left[\begin{array}{l} (x_0, \ldots, x_q) \xleftarrow{\$} \mathcal{B}(H) : \\ \exists u, v \ s.t. \ H_i(x_u) = H_i(x_v) \wedge x_u \neq x_v \end{array} \right] > \frac{t_\mathcal{B}(t_\mathcal{B} - 1)}{2^{i+1}}.$$

The bound $t_\mathcal{B}(t_\mathcal{B} - 1)/2^{i+1}$ in the above definition stems from the birthday bound, which states that an adversary which evaluates a random function with range $\{0,1\}^i$ at most q times will find a collision with probability at most $q(q-1)/2^{i+1}$. Observe here that an adversary in our computational model must perform at least $q \cdot \log_2 q$ computational operations in order to output q pairwise distinct bit strings. Hence, in order to break TruCR, an algorithm must be better than the trivial birthday attack algorithm by a factor larger than $\log_2 q$.

Constructing hash families from standard hash functions. Let H' be any standard cryptographic hash function, such as SHA-3, for example. We can construct a hash function family $\mathcal{H}$ as

$$\mathcal{H} := \{H : H(x) := H'(r||x), r \in \{0,1\}^k\}.$$

A uniformly random hash function H from the family is chosen by selecting a uniformly random bit string $r \in \{0,1\}^k$. $H(x)$ is evaluated by computing $H'(r||x)$.

Strength of the TruCR assumption. We view truncation collision resistance as a very natural security property for cryptographic hash functions. Note that truncated versions of SHA-256 (to 224 bits) and SHA-512 (to 384 bits) have been standardized by NIST [Nat15a]. Also the SHA-3 standard [Nat15b] defines two extendable-output-functions (XOFs), in which the output length can be adapted to any desired length.

Furthermore, recall that the standard way to determine the size of the output of a hash function in practice is to fix a security parameter k, and to take a hash function of output size $2k$. For example, choosing SHA-256 (which has 256-bit hash values) for $k = 128$ is considered an adequate choice in practice, if collision-resistance at "128-bit security level" is required. Note that one essentially assumes here already that there is no significantly better collision attack

[1] We explain below how our results can be extended to models of computation where an algorithm is able to output multiple bits per elementary machine operation.

than the generic birthday algorithm. TruCR generalizes this assumption to *prefixes* of the hash function output.

We also note that our applications will require hash functions with output length $4(k+1)$, rather than the "minimal" $2k$. For example, for $k = 127$ we would use SHA-512.

In the full version [JK17], we also sketch how TruCR hash functions can be constructed from standard assumptions.

Choice of computational model and weakening the TruCR assumption. Assume a computational model where an algorithm is able to output many pairwise distinct values $x_0, \ldots, x_q$ in a *single* elementary machine operation, and thus within a single time unit. Note that such an algorithm would be able to trivially break the TruCR assumption with a simple birthday attack. To overcome this technical obstacle, we are working in a computational model where algorithms are assumed to write their output bit-by-bit to an output tape.

In order to generalize this to a computational model where algorithms are able to output any constant number of bits in parallel in a single step, we can weaken Definition 1 by increasing the size of the prefix for which the adversary has to find a collision. To this end, one would replace the requirement

$$H_i(x_u) = H_i(x_v).$$

in Definition 1 with $H_{i+c}(x_u) = H_{i+c}(x_v)$ for some small constant value c (e.g., $c \in \{1, ..., 10\}$). This also allows to add some additional "safety margin" to the assumption, if desired, at the cost of an additional constant tightness loss factor of 2^c in the security proofs of our constructions. In the remainder of the paper, we will work with the original Definition 1, as it simplifies the exposition of our main results.

Relation to ELFs. From a high-level perspective, truncation collision-resistant hash functions are related to *extremely lossy functions* (ELFs), which were introduced by Zhandry [Zha16]. In some applications, TruCR and ELFs can be used in a very similar way to argue in a security proof, and it seems that some of our applications can also be achieved by using an ELF instead. ELFs furthermore allow for some additional applications, like the construction of *output-intractable* hash functions or a standard-model instantiation of full-domain hash (the latter in combination with indistinguishability obfuscation).

The main difference between Zhandry's work and ours is that [Zha16] gives new constructions of hash functions based on public-key techniques and the reasonable exponential hardness assumption of the decisional Diffie-Hellman problem in algebraic groups. Instead, following [Can97,BHK13,FM16], we use standard cryptographic hash functions that are already widely-used in practice, and a similarly reasonable complexity assumption for such functions. This partially resolves the open problem posed in [Zha16] of constructing ELFs using symmetric-key techniques: while we do not construct full ELFs, we show how certain potential applications of ELFs can be realized based on standard hash

functions. We furthermore show how TruCR can be used to obtain interesting new constructions: the first ID-KEM with full adaptive security and very short ciphertexts (only a single element from a bilinear group), and the first digital signature scheme with full adaptive security and very short signatures (again, only single element from a bilinear group). One can also show that ELFs and TruCR hash functions do not imply each other. Due to space limitations, this is deferred to the full version of this paper [JK17].

On assuming exponential hardness. Both the work of Zhandry [Zha16] and ours assume exponential hardness of the underlying computational problems. The construction of ELFs from [Zha16] assumes the exponential hardness of the DDH assumption in suitable algebraic groups. This is a strong assumption, but it appears reasonable, e.g., in certain elliptic curve groups, where the best known algorithms for solving the DDH problem have exponential complexity. Furthermore, note that this matches the choice of elliptic curve groups in practice, where typically a group of prime order $\approx 2^{2k}$ is expected to achieve "k-bit security". Similarly, we assume that for a cryptographic hash function there exists no significantly better collision attack than the generic birthday collision algorithm, which also has exponential complexity.

Useful technical lemma. To conclude our discussion of TruCR, we state a technical lemma, which will be useful to leverage truncation collision resistance in security proofs. Intuitively, the lemma will provide bounds to ensure in our security proofs that for each adversary with some running time t and success probability ϵ there always exists a "good" index j such that H_{2^j} is "sufficiently collision-resistant", but at the same time the range $\{0,1\}^{2^j}$ of H_{2^j} is "sufficiently small". As usual, all logarithms are to base 2 in the sequel.

Lemma 1. *Let $t \in \mathbb{N}$ and $\epsilon \in (0,1]$ with $t/\epsilon < 2^k$, and $j := \left\lfloor \log\log(4t^2/\epsilon) \right\rfloor + 1$. Then it holds that*

$$j \in \{1, \ldots, \log 4(k+1)\}, \qquad \frac{4t^2}{2^{2^j+1}} < \frac{\epsilon}{2} \qquad \text{and} \qquad 2^{2^j} \leq \left(\frac{4t^2}{\epsilon}\right)^2.$$

Proof. We first show that $j \in \{1, \ldots, \log 4(k+1)\}$. Since $\epsilon > 0$, we trivially have $j \geq 1$. Additionally using that $t/\epsilon < 2^k$, we obtain

$$\begin{aligned} j &= \left\lfloor \log\log(4t^2/\epsilon) \right\rfloor + 1 \leq \log\log(4t^2/\epsilon) + 1 \\ &< \log\log(2^{2k+2}) + 1 = \log 4(k+1). \end{aligned}$$

To show $4t^2/2^{2^j+1} < \epsilon/2$, we compute

$$\frac{4t^2}{2 \cdot 2^{2^j}} = \frac{4t^2}{2 \cdot 2^{2^{\lfloor \log\log(4t^2/\epsilon) \rfloor + 1}}} < \frac{4t^2}{2 \cdot 2^{2^{\log\log(4t^2/\epsilon)}}} = \frac{4t^2}{2 \cdot (4t^2/\epsilon)} = \frac{\epsilon}{2}.$$

Finally, we get $2^{2^j} \leq \left(4t^2/\epsilon\right)^2$ from

$$2^{2^j} = 2^{2^{\left\lfloor \log\log(4t^2/\epsilon) \right\rfloor + 1}} \leq 2^{2 \cdot 2^{\log\log(4t^2/\epsilon)}} = (4t^2/\epsilon)^2.$$

□

3 Identity-based key encapsulation

Recall that the commonly accepted standard security notion for identity-based key encapsulation (ID-KEM) is *adaptive chosen-ID* security (IND-ID-CPA), as introduced in [BFMLS08]. Here, the adversary in the security experiment may adaptively choose *both* the "challenge identity" (i.e., the identity for which it receives a challenge ciphertext) and the "key-query identities" (i.e., the identities for which it requests a user secret key). A much weaker common standard security notion is *selective challenge-ID security* (IND-sID-CPA) [CHK03,CHK04], where the adversary has to announce the challenge identity and at the very beginning of the security experiment, even before seeing the master public key, but may adaptively query for user secret keys. In this work, we consider the even weaker notion with *selective* challenge-identity and *non-adaptive* key-queries (IND-snaID-CPA), where the adversary has to announce *both* the challenge identity and all key-query identities at the very beginning of the security experiment, even before seeing the master public key (see Section 3.1 for formal definitions).

As a first application of TruCR, we describe a simple generic construction of a fully adaptively IND-ID-CPA-secure ID-KEM from any ID-KEM which is only IND-snaID-CPA-secure. This shows that if TruCR hash functions exist, then ID-KEMs with full IND-ID-CPA-security are implied by IND-snaID-CPA-secure ID-KEMs. The latter are usually significantly easier to construct. This result also introduces a technique for leveraging truncation collision-resistance in security proofs. The generic conversion is relatively efficient: it increases the size of public parameters, user secret keys, and ciphertexts by a factor of only $\mathcal{O}(\log k)$, where k is the security parameter. The security reduction is non-tight, but polynomial-time. Previous standard techniques to build an adaptively secure ID-KEM from a selectively-secure one, e.g. by using admissible [BB04b] or programmable [HK08,Wat05] hash functions, work only non-generically for certain schemes with specific properties. The only previously known *generic* way to turn an IND-snaID-CPA-secure scheme into a fully IND-ID-CPA-secure one is to use the programmable ROM.

Then we show how TruCR can be used to obtain the first ID-KEM with full IND-ID-CPA-security without random oracles and with very short ciphertexts, where the ciphertext overhead is only a single element from a prime-order group. The only previously known ID-KEM with such short ciphertexts and adaptive security is the construction of Boneh and Franklin [BF01], which only has a security proof in the ROM. The adaptively secure standard-model scheme of Wee [Wee16] requires significantly larger composite-order groups. Our scheme is based on the selectively-secure Boneh-Boyen IBE scheme [BB04a] and proven secure under a q-type assumption, but it shows that adaptively secure ID-KEMs with such short ciphertext overhead can be constructed without random oracles.

On using complexity leveraging. Achieving adaptive security from selective security is sometimes also possible by directly assuming exponential hardness of breaking the underlying selectively-secure scheme, and then using complexity leveraging. For instance, in order to convert an IND-sID-CPA-secure ID-KEM

into an IND-ID-CPA-secure one, we can simply guess the challenge identity id^* chosen by the IND-ID-CPA adversary up front at the beginning of the security experiment, and use this to implement a straightforward reduction to the IND-sID-CPA of the considered scheme. Since the identity space usually has exponential size 2^k where k is the security parameter, then this incurs an exponential security loss of 2^k, which must be compensated by choosing larger parameters for the underlying scheme. If the underlying scheme is exponentially secure, then doubling the size of parameters, such as underlying algebraic groups for instance, usually suffices in this case.

However, note that this complexity leveraging approach is *not* useful if one starts from the weaker IND-snaID-CPA-security, as we do in this work. This is because here we would have to guess not only the challenge identity id^*, but also *all* q identities $id_1, \ldots, id_q$ of key-queries made by the IND-ID-CPA adversary up front, which yields a security loss of $\binom{2^k}{q+1} \geq 2^{kq-\log q} \approx 2^{kq}$. This cannot be easily compensated with larger parameters, because even if the underlying scheme is exponentially secure, then this would still require to increase the parameters by a factor of about q, which is completely impractical.

Relation to a result by Döttling and Garg. There exists several (semi-)generic constructions of strongly-secure signatures from signatures with weaker security [EGM96,BSW06,SPW07,HW09b,BK10]. All these works have in common that they can be applied only to signature scheme, but not to identity-based schemes like ID-KEMs, because they are either probabilistic (e.g., based on ephemeral one-time signatures or chameleon hash functions), or consider a setting with a non-adaptive adversary, which is forced to output all chosen-message queries before seeing the public key.

Döttling and Garg [DG17a] describe a generic construction of adaptively secure IBE from selectively-secure IBE. Their approach is completely different from ours, as it is based on the techniques introduced in [DG17b]. A ciphertext of the adaptively secure scheme consists of $n+1$ garbled circuits, ℓ corresponding labels, and ℓ ciphertexts of a specific type of encryption scheme called *one-time signature with encryption (OTSE)*, plus a few additional values. Here, n is the bit-length of identities and ℓ denotes the length of encrypted messages. A ciphertext of the OTSE scheme in turn consist of $2k$ ciphertexts of the underlying selectively-secure scheme, where k is the security parameter. The size of public and secret keys is similarly large.

In contrast, our generic construction is much more direct, and shows how to construct an adaptively secure scheme that requires only $\log k$ copies of the underlying scheme (with similarly short public and secret keys), but based on an additional security assumption on the hash function. Our approach also enables the construction of an adaptively secure scheme where a ciphertext consists only of a single group element, which currently seems out of reach of the approach of Döttling and Garg.

Another difference is that they start from the IND-sID-CPA security notion, while we start from the even weaker IND-snaID-CPA notion. It is unclear whether their results can be adopted to the same setting.

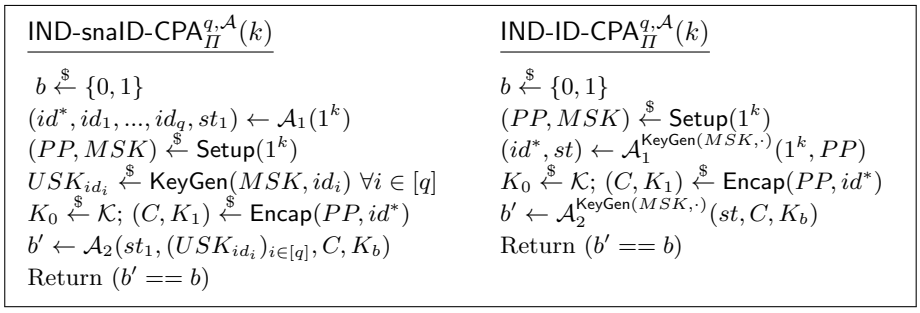

Fig. 1. The security experiments for ID-KEMs, executed with scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ and adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The oracle $\mathsf{KeyGen}(MSK, id)$ returns $USK_{id} \xleftarrow{\$} \mathsf{KeyGen}(MSK, id)$ with the restriction that $\mathcal{A}$ is not allowed to query oracle $\mathsf{KeyGen}(MSK, \cdot)$ for the target identity id^*.

In summary, the construction of Döttling and Garg has the advantage of not requiring an additional complexity assumption, but also seems rather far away from practical applicability and is therefore currently mostly of theoretical interest. In contrast, our construction requires an additional assumption, but is more direct and incurs significantly less overhead, such that it already yields very efficient schemes. Thus, both approaches achieve different goals.

3.1 Definitions and security notions

Definition 2. *An ID-KEM consists of the following four PPT algorithms:*

$\mathsf{Setup}(1^k)$ *returns the public parameters PP and the master secret key MSK. We assume that PP defines (implicitly or explicitly) an identity space $\mathcal{I}$, a key space $\mathcal{K}$ and a ciphertext space $\mathcal{C}$.*

$\mathsf{KeyGen}(MSK, id)$ *returns the user secret key USK_{id} for identity $id \in \mathcal{I}$.*

$\mathsf{Encap}(PP, id)$ *returns a tuple (C, K), where $K \in \mathcal{K}$ is a key and $C \in \mathcal{C}$ is a ciphertext encapsulating K with respect to identity id.*

$\mathsf{Decap}(USK_{id}, C, id)$ *returns the decapsulated key $K \in \mathcal{K}$ or an error symbol $\perp$.*

For perfect correctness we require that for all $k \in \mathbb{N}$, all pairs (PP, MSK) generated by $\mathsf{Setup}(1^k)$, all identities $id \in \mathcal{I}$, all (K, C) output by $\mathsf{Encap}(PP, id)$ and all USK_{id} generated by $\mathsf{KeyGen}(MSK, id)$:

$$\Pr[\mathsf{Decap}(USK_{id}, C, id) = K] = 1.$$

Adaptive security. Let us first recall the standard IND-CPA-security notion for ID-KEMs from [BFMLS08]. To this end, consider the IND-ID-CPA security experiment depicted in Figure 1.

Definition 3. *We say that adversary $\mathcal{A}$ $(t_{\mathcal{A}}, q, \epsilon_{\mathcal{A}})$-breaks the* IND-ID-CPA *security of Π, if $\Pr[\mathsf{IND\text{-}ID\text{-}CPA}^{q,\mathcal{A}}_{\Pi}(k) = 1] - \frac{1}{2} \geq \epsilon_{\mathcal{A}}$ and $t_{\mathcal{A}}$ is the running time of $\mathcal{A}$, including the security experiment.*

Remark 1. Including the running time of the security experiment into $t_{\mathcal{A}}$ will later allow us to simplify our security analysis and the statement of theorems significantly.

Selective and non-adaptive security. We also define a very weak security notion for ID-KEMs. Consider the IND-snaID-CPA security experiment depicted in Figure 1, where the attacker has to commit to both the challenge-ID id^* the key-query identities $id_1, \ldots, id_q$ non-adaptively and even before receiving the master public key PP.

Definition 4. *We say that $\mathcal{A}$ $(t_{\mathcal{A}}, q, \epsilon_{\mathcal{A}})$-breaks the* IND-snaID-CPA *security of Π, if it runs in time $t_{\mathcal{A}}$ and*

$$\Pr[\mathsf{IND\text{-}snaID\text{-}CPA}_{\Pi}^{q,\mathcal{A}}(k) = 1] - \frac{1}{2} \geq \epsilon_{\mathcal{A}}.$$

3.2 From weak security to adaptive security

Construction. Let $\mathcal{H} = \{H|\{0,1\}^* \to \{0,1\}^{4(k+1)}\}$ be a family of keyed hash functions and define

$$\ell := \log 4(k+1).$$

Let $\Pi' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Encap}', \mathsf{Decap}')$ be an ID-KEM. We construct our ID-KEM scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ as follows.

Setup. Compute $(PP_i, MPK_i) \xleftarrow{\$} \mathsf{Setup}'(1^k)$ for all $i \in \{1, \ldots, \ell\}$ and $H \leftarrow \mathcal{H}$ and define

$$PP = (H, PP_1, \ldots, PP_\ell) \qquad \text{and} \qquad MSK = (MSK_1, \ldots, MSK_\ell).$$

and output (PP, MSK).

User Key Generation. To create a private key for the identity id, compute $USK_i \xleftarrow{\$} \mathsf{KeyGen}'(MSK_i, H_{2^i}(id))$ for all $i \in \{1, \ldots, \ell\}$. Define

$$USK_{id} := (USK_1, \ldots, USK_\ell)$$

and output USK_{id}.

Encapsulation. On input $PP = (H, PP_1, \ldots, PP_\ell)$ and id, compute $(K_i, C_i) \xleftarrow{\$} \mathsf{Encap}'(PP_i, H_{2^i}(id))$ for all $i \in \{1, \ldots, \ell\}$. Then define $K := \bigoplus_{i=1}^{\ell} K_i$, where $\bigoplus$ denotes the XOR-operation and output $(C, K) = ((C_1, ..., C_\ell), K)$.

Decapsulation. On input $C = (C_1, ...C_\ell)$ and USK_{id}, compute

$$K_i = \mathsf{Decap}'(USK_i, C_i)$$

for all $i \in \{1, \ldots, \ell\}$ and output $K := \bigoplus_{i=1}^{\ell} K_i$.

The correctness of Π follows immediately from the correctness of Π'.

Security analysis. Recall that we have $\ell := \log 4(k+1)$, and that Lemma 1 shows that for each adversary $\mathcal{A}$ with $t_{\mathcal{A}}/\epsilon_{\mathcal{A}} < 2^k$, there exists an index $j \in \{1, \ldots, \ell\}$ such that

$$j = \left\lfloor \log \log 4t_{\mathcal{A}}^2/\epsilon_{\mathcal{A}} \right\rfloor + 1. \tag{4}$$

Theorem 1. *Let $\mathcal{A}$ be an adversary that $(t_{\mathcal{A}}, q_{\mathcal{A}}, \epsilon_{\mathcal{A}})$-breaks the* IND-ID-CPA*-security of Π such that $t_{\mathcal{A}}/\epsilon_{\mathcal{A}} < 2^k$ and let j be an index such that (4) is satisfied. Given $\mathcal{A}$ and j, we can either construct an adversary $\mathcal{B}_j$ that $(t_{\mathcal{B}}, q_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks the* IND-snaID-CPA*-security of Π' with*

$$t_{\mathcal{B}} = \mathcal{O}(t_{\mathcal{A}}^4/\epsilon_{\mathcal{A}}^2), \qquad q_{\mathcal{B}} < 4t_{\mathcal{A}}^4/\epsilon_{\mathcal{A}}^2 \quad \textit{and} \quad \epsilon_{\mathcal{B}} \geq \frac{\epsilon_{\mathcal{A}}^3}{32t_{\mathcal{A}}^4}$$

or an adversary $\mathcal{C}$ that 2^j-breaks the truncation collision resistance of $\mathcal{H}$.

Note that the theorem considers adversaries that for a given security parameter k have "work factor" $t_{\mathcal{A}}/\epsilon_{\mathcal{A}}$ below 2^k. This deviates slightly from the common asymptotic definition, where it is required that $t_{\mathcal{A}}$ is polynomially-bounded and $\epsilon_{\mathcal{A}}$ is non-negligible. Assuming $t_{\mathcal{A}}/\epsilon_{\mathcal{A}} < 2^k$ is an alternative way of expressing that a cryptosystem is secure with respect to a given security parameter k that originates from the "concrete security" approach of Bellare and Ristenpart [BR09]. Note also that the security loss of reduction $\mathcal{B}$ is polynomially-bounded, but relatively large.

Proof. We say that event coll_j occurs in the $\mathsf{IND\text{-}ID\text{-}CPA}_{\Pi}^{q,\mathcal{A}}(k)$ security experiment, when $\mathcal{A}$ ever queries a user secret key for identity id_z such that $H_{2^j}(id_z) = H_{2^j}(id^*)$, where id^* is the challenge identity. We distinguish between two cases.

Case 1: $\Pr[\mathsf{coll}_j] > \epsilon_{\mathcal{A}}/2$**.** In this case we are able to construct an adversary $\mathcal{C}$ on the truncation collision resistance of H.

Construction of $\mathcal{C}$. $\mathcal{C}$ runs the $\mathsf{IND\text{-}ID\text{-}CPA}_{\Pi}^{q,\mathcal{A}}(k)$ experiment. Whenever $\mathcal{A}$ issues a query id_i to KeyGen, then $\mathcal{C}$ additionally outputs id_i. When $\mathcal{A}$ outputs the challenge identity id^*, then $\mathcal{C}$ outputs id^*. When $\mathcal{A}$ terminates, then $\mathcal{C}$ terminates, too.

Running time of $\mathcal{C}$. To bound the running time of $\mathcal{C}$, note that it consists of the time required to run $\mathcal{A}$, plus the time required to simulate the experiment, plus the time required to output the queries $id_1, \ldots, id_q, id^*$ made by $\mathcal{A}$. Thus, since $t_{\mathcal{A}}$ already includes the running time of the experiment, the total running time of $\mathcal{C}$ is bounded by $t_{\mathcal{C}} \leq 2 \cdot t_{\mathcal{A}}$.

Success probability of $\mathcal{C}$. To analyze the success probability of $\mathcal{C}$, let $i := 2^j$. Note that the success probability of $\mathcal{C}$ is equal to $\Pr[\mathsf{coll}_j]$, and by definition of j and Lemma 1 we have

$$\Pr[\mathsf{coll}_j] > \epsilon_{\mathcal{A}}/2 > 4t_{\mathcal{A}}^2/2^{i+1} \geq 2t_{\mathcal{A}}(2t_{\mathcal{A}} - 1)/2^{i+1} \geq t_{\mathcal{C}}(t_{\mathcal{C}} - 1)/2^{i+1}$$

so that indeed $\mathcal{C}$ breaks the truncation collision resistance of H.

Case 2: $\Pr[\mathsf{coll}_j] \leq \epsilon_{\mathcal{A}}/2$**.** In this case we construct an algorithm $\mathcal{B}_j$ on the IND-snaID-CPA security of Π'. Before we are able to describe $\mathcal{B}_j$, we will describe a short sequence of games, which gradually modifies the original $\mathsf{IND\text{-}ID\text{-}CPA}_{\Pi}^{q,\mathcal{A}}(k)$ experiment. In the sequel let G_i denote the event that Game i outputs 1.

Game 0. This is the original $\mathsf{IND\text{-}ID\text{-}CPA}_{\Pi}^{q,\mathcal{A}}(k)$ security experiment. By definition, we have

$$\Pr[G_0] = \frac{1}{2} + \epsilon_{\mathcal{A}}.$$

Game 1. Game 1 is identical to Game 0, except that if coll_j occurs, then Game 1 outputs a random bit and aborts.

Using that $G_1 \wedge \neg\mathsf{coll}_j \iff G_0 \wedge \neg\mathsf{coll}_j$ and that $\Pr[\mathsf{coll}_j] \leq \epsilon_{\mathcal{A}}/2$, we obtain

$$\Pr[G_1] \geq \Pr[G_1 \wedge \neg\mathsf{coll}_j] = \Pr[G_0 \wedge \neg\mathsf{coll}_j] \geq \Pr[G_0] - \Pr[\mathsf{coll}_j] \geq \frac{1}{2} + \frac{\epsilon_{\mathcal{A}}}{2}.$$

Game 2. In Game 2, we additionally guess a string $ID^* \xleftarrow{\$} \{0,1\}^{2^j}$ uniformly random. We raise event $\mathsf{abort}_{\mathsf{chal}}$ if adversary $\mathcal{A}$ requests a challenge ciphertext for identity id^* with $H_{2^j}(id^*) \neq ID^*$.

We stress that this game merely *defines* event $\mathsf{abort}_{\mathsf{chal}}$, as a preparation for the analysis in the following game, but we do not make any changes to the experiment. Thus, we have

$$\Pr[G_2] = \Pr[G_1].$$

Note also that $\mathcal{A}$ receives no information about ID^* and thus the events G_2 and $\mathsf{abort}_{\mathsf{chal}}$ are independent.

Game 3. This game is identical to Game 2, except that we output a random bit and abort the game, if $\mathsf{abort}_{\mathsf{chal}}$ occurs.

Using that $G_3 \wedge \neg\mathsf{abort}_{\mathsf{chal}} \iff G_2 \wedge \neg\mathsf{abort}_{\mathsf{chal}}$ we first get

$$\begin{aligned}
\Pr[G_3] &= \Pr[G_3 \mid \mathsf{abort}_{\mathsf{chal}}] \cdot (1 - \Pr[\neg\mathsf{abort}_{\mathsf{chal}}]) + \Pr[G_3 \wedge \neg\mathsf{abort}_{\mathsf{chal}}] \\
&= \frac{1}{2} \cdot (1 - \Pr[\neg\mathsf{abort}_{\mathsf{chal}}]) + \Pr[G_2 \wedge \neg\mathsf{abort}_{\mathsf{chal}}] \\
&= \frac{1}{2} \cdot (1 - \Pr[\neg\mathsf{abort}_{\mathsf{chal}}]) + \Pr[G_2 \mid \neg\mathsf{abort}_{\mathsf{chal}}] \cdot \Pr[\neg\mathsf{abort}_{\mathsf{chal}}] \\
&= \frac{1}{2} + \Pr[\neg\mathsf{abort}_{\mathsf{chal}}] \cdot \left(\Pr[G_2 \mid \neg\mathsf{abort}_{\mathsf{chal}}] - \frac{1}{2} \right).
\end{aligned}$$

Using now that $\Pr[G_2 \mid \neg\mathsf{abort}_{\mathsf{chal}}] = \Pr[G_2]$, since G_2 and $\mathsf{abort}_{\mathsf{chal}}$ are independent, the bound on $\Pr[G_2]$ from Game 2, and finally using that our choice

of j and Lemma 1 yield that $2^{2^j} \leq 16t_{\mathcal{A}}^4/\epsilon_{\mathcal{A}}^2$, we obtain

$$\begin{aligned}\Pr[G_3] &= \frac{1}{2} + \Pr[\neg\mathsf{abort}_{\mathsf{chal}}] \cdot \left(\Pr[G_2 \mid \neg\mathsf{abort}_{\mathsf{chal}}] - \frac{1}{2}\right) \\ &= \frac{1}{2} + \Pr[\neg\mathsf{abort}_{\mathsf{chal}}] \cdot \left(\Pr[G_2] - \frac{1}{2}\right) \\ &\geq \frac{1}{2} + \frac{1}{2^{2^j}} \cdot \frac{\epsilon_{\mathcal{A}}}{2} \geq \frac{1}{2} + \frac{\epsilon_{\mathcal{A}}^3}{32t_{\mathcal{A}}^4}.\end{aligned}$$

Construction of $\mathcal{B}_j$. Now we are ready to construct $\mathcal{B}_j$, which simulates Game 3 as follows.

Initialization. At the beginning of the experiment, $\mathcal{B}_j$ samples a random bit string $ID^* \leftarrow \{0,1\}^{2^j}$ and defines $2^{2^j}-1$ identities $ID_1, \ldots, ID_{2^{2^j}-1}$, consisting of all values in $\{0,1\}^{2^j} \setminus \{ID^*\}$. It outputs these values to the $\mathsf{IND\text{-}snaID\text{-}CPA}$ experiment, which then generates and responds with a key pair $(PP', MSK') \xleftarrow{\$} \mathsf{Setup}'(1^k)$, user secret keys USK'_{ID_s}, $s \in \{1, \ldots, 2^{2^j}-1\}$, for the requested identities, and a challenge ciphertext (C', K'), where $(C', K) \xleftarrow{\$} \mathsf{Encap}'(PP', ID^*)$ and either $K' = K$ or K' is uniformly random.

Simulation of the public key. In order to simulate the public key, $\mathcal{B}_j$ generates $H \leftarrow \mathcal{H}$ and $\ell - 1$ additional key pairs by running $(PP_i, MPK_i) \xleftarrow{\$} \mathsf{Setup}'$ for all $i \in \{1, \ldots, \ell\} \setminus \{j\}$. Then it sets

$$PP = (H, PP_1, ..., PP_{j-1}, PP', PP_{j+1}, ..., PP_l).$$

Finally, $\mathcal{B}_j$ outputs PP to $\mathcal{A}$. Note that this is a correctly distributed master public key for scheme Π.

Simulation of key queries. Note that the two abort conditions introduced in Games 1 and 3 together imply in particular that $\mathcal{B}_j$ aborts and outputs a random bit, if $\mathcal{A}$ ever issues a key query for identity id_z with $H_{2^j}(id_z) = ID^*$. This is because in this case either it holds that $H_{2^j}(id_z) = ID^* \neq H_{2^j}(id^*)$, in which case $\mathsf{abort}_{\mathsf{chal}}$ occurs, or it holds that $H_{2^j}(id_z) = ID^* = H_{2^j}(id^*)$, which means that coll_j occurs. Thus, we only have to consider the case $H_{2^j}(id_z) \neq ID^*$ in the sequel. Note also that $\mathcal{B}_j$ knows MSK_i for all $i \neq j$ and user secret keys for USK'_{ID_s} for all $ID_s \in \{0,1\}^{2^j} \setminus \{ID^*\}$. Therefore it is able to compute and return valid user secret keys to $\mathcal{A}$ for all identities id_z with $H_{2^j}(id_z) \neq ID^*$.

Whenever $\mathcal{A}$ requests a user secret key for an identity $id_z \in \{0,1\}^*$, $\mathcal{B}_j$ proceeds as follows. If there is no abort, then $\mathcal{B}_j$ computes

$$(USK_i) \xleftarrow{\$} \mathsf{KeyGen}'(MSK_i, H_{2^i}(id_z))$$

for all $i \in \{1, \ldots, \ell\} \setminus \{j\}$. Recall that $\mathcal{B}_j$ has requested user secret keys for all values $H_{2^j}(id_z) \in \{0,1\}^{2^j}$ with $H_{2^j}(id_z) \neq ID^*$, in particular for $ID_s \in \{0,1\}^{2^j}$ such that $ID_s = H_{2^j}(id_z)$. Therefore it is able to efficiently determine and output

$$USK_{id_z} = (USK_1, ..., USK_{j-1}, USK'_{ID_s}, USK_{j+1}, ..., USK_l).$$

Computing the challenge ciphertext. Recall that we only have to consider the case $H_{2^j}(id^*) = ID^*$, as otherwise $\mathcal{B}_j$ outputs a random bit and aborts. Thus, when adversary $\mathcal{A}$ outputs a challenge identity id^* with $H_{2^j}(id^*) = ID^*$, then $\mathcal{B}_j$ computes $(C_i, K_i) \xleftarrow{\$} \mathsf{Encap}'(PP_i, H_{2^i}(id^*))$ for all $i \in \{1, \ldots, \ell\} \setminus \{j\}$, and then

$$K := \bigoplus_{i=1, i\neq j}^{\ell} K_i \oplus K' \text{ and } C := (C_1, \ldots, C_{j-1}, C', C_{j+1}, \ldots, C_\ell),$$

where (C', K') is the tuple received from the IND-snaID-CPA-experiment. B_j returns (C, K) to $\mathcal{A}$ and outputs whatever $\mathcal{A}$ outputs. Note that if K' is a "real" key, then so is K, while if K' is "random", then so is K.

Success probability of $\mathcal{B}_j$. Since $\mathcal{B}_j$ provides a perfect simulation of Game 3 for $\mathcal{A}$, we have

$$\Pr\left[\mathsf{IND\text{-}snaID\text{-}CPA}^{q,\mathcal{B}}_{\Pi'}(k) = 1\right] = \Pr\left[G_3\right] \geq 1/2 + \frac{\epsilon^3_{\mathcal{A}}}{32t^4_{\mathcal{A}}}.$$

Running time of $\mathcal{B}_j$. The running time $t_{\mathcal{B}}$ of $\mathcal{B}_j$ consists of the time needed to execute $\mathcal{A}$, the time required to simulate the IND-ID-CPA security experiment, and the time required to request the $2^{2^j} - 1$ user secret keys from the IND-snaID-CPA experiment, plus a minor number of additional operations. Making use of Lemma 1, we get

$$t_{\mathcal{B}} \approx t_{\mathcal{A}} + \mathcal{O}\big(2^{2^j} - 1\big) \approx t_{\mathcal{A}} + \mathcal{O}\left(\frac{t^4_{\mathcal{A}}}{\epsilon^2_{\mathcal{A}}}\right) = \mathcal{O}\left(\frac{t^4_{\mathcal{A}}}{\epsilon^2_{\mathcal{A}}}\right).$$

Remark 2. We remark also that $t_{\mathcal{B}} \approx t_{\mathcal{A}}$ if we instead consider the generic construction of an IND-ID-CPA-secure ID-KEM from an IND-sID-CPA-secure one. This is because in this case the reduction $\mathcal{B}_j$ does not have to issue all KeyGen queries at the beginning of the experiment. Instead, it can make all queries "on demand", whenever $\mathcal{A}$ issues such a query. Their number is identical to the number q of KeyGen made by $\mathcal{A}$, so that we get a reduction which runs in strict polynomial time.

Note also that $\mathcal{B}_j$ issues $q_{\mathcal{B}} = 2^{2^j} - 1 < 4t^4_{\mathcal{A}}/\epsilon^2_{\mathcal{A}}$ user key queries. This completes the proof of Theorem 1.

□

3.3 Adaptively secure ID-KEM with short ciphertexts

Now we show that it is possible to construct an ID-KEM with full adaptive security, where a ciphertext consists of only a *single* element of a bilinear group. A comparison to previous IBE-schemes, viewed as ID-KEMs, is given in Table 2.

Scheme	$\|\mathbb{G}\|$	$\|pk\|$	$\|C\|$	Security	Assumption	ROM	Security Loss
[BF01]	Prime	2	1	Full	BDDH	Yes	$\mathcal{O}(q_{key})$
[BB04a]	Prime	4	2	Selective	strong q-DH	No	$\mathcal{O}(1)$
[Wat05]	Prime	$n+3$	2	Full	BDDH	No	$\tilde{\mathcal{O}}(t^2 + (n \cdot q_{key} \cdot \epsilon^{-1})^2)$
[Wee16]	Comp.	3	1	Full	Dec. Subgrp.	No	$\mathcal{O}(q_{key})$
Ours	Prime	$\mathcal{O}(k)$	1	Full	q-DH	No	$\mathcal{O}(t_{\mathcal{A}}^7/\epsilon_{\mathcal{A}}^4)$

Fig. 2. Comparison of ID-based encryption schemes with short ciphertexts. The column $|\mathbb{G}|$ refers to the order of the underlying group (prime or composite), $|pk|$ is the number of group elements in public keys (common descriptions of groups and hash functions not included), where n is the length of identities and k the security parameter. All public keys include one element from the target group of the pairing, except for [BF01]. $|C|$ is the number of group elements in the ciphertexts when viewed as a KEM. "Full" security means IND-ID-CPA security as defined below, "selective" security is from [BB04a]. The remaining columns state the hardness assumption in the security proof, whether the Random Oracle Model is used, and the security loss of the reduction, where q_{key} is the number of identity key queries, $t_{\mathcal{A}}$ and $\epsilon_{\mathcal{A}}$ the running time and advantage of the adversary, and the loss is the value L that satisfies $t_{\mathcal{B}}/\epsilon_{\mathcal{B}} = L \cdot t_{\mathcal{A}}/\epsilon_{\mathcal{A}}$, where $t_{\mathcal{B}}$ and $\epsilon_{\mathcal{B}}$ are the success probability and running time of the reduction.

Building block: simplified Boneh-Boyen ID-KEM. The following ID-KEM is based on the IBE scheme of Boneh and Boyen [BB04a]. Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be groups of prime order p with generators g_1, g_2, g_T, respectively, and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an efficiently computable pairing. We will use the implicit notation of Escala *et al.* [EHK+13], and write $[x]_s$ shorthand for g_s^x for all $s \in \{1, 2, T\}$.

Simple ID-KEM based on the Boneh-Boyen IBE scheme. We use the following scheme as a building block for our adaptively secure ID-KEM.

Setup Choose two random elements $x, y \xleftarrow{\$} \mathbb{Z}_p$. Then define $\nu = e([1]_1, [y]_2)$. The public parameters PP and the master secret key MSK are defined as $PP = ([1]_1, [x]_1, \nu)$ *and* $MSK = (x, y)$.

Key Generation To create a private key for identity $id \in \mathbb{Z}_p$, compute and return $USK_{id} = [y/(id + x)]_2$.

Encapsulation. To encapsulate a key $K \in \mathbb{G}_T$ under public key $id \in \mathbb{Z}_p$, pick a random $r \in \mathbb{Z}_p$ and output

$$(C, K) = ([id + x]_1^r, \nu^r) \in \mathbb{G}_1 \times \mathbb{G}_T.$$

Decapsulation. To decapsulate C using the private key USK_{id}, compute and output $e(C, USK_{id})$.

Proving security of the simplified Boneh Boyen IBE. Consider the following experiment q-BDDHI(1^k), which was generalized to asymmetric bilinear groups in [BB11]. With regard to the security parameter k, the challenger generates an

asymmetric pairing group and chooses $x \in \mathbb{Z}_p$ uniformly at random. Then it chooses $T \xleftarrow{\$} \mathbb{G}_T$ and defines

$$T_0 := ([1]_1, [x]_1, [1]_2, [x]_2, \ldots [x^q]_2, T),$$
$$T_1 := ([1]_1, [x]_1, [1]_2, [x]_2, \ldots [x^q]_2, e([1]_1, [1]_2)^{\frac{1}{x}}).$$

Finally, it flips a fair coin $\beta \xleftarrow{\$} \{0,1\}$ and outputs T_β to the adversary. The task of adversary $\mathcal{B}$ is to determine β.

Definition 5. *We say that adversary $\mathcal{B}$ (t, ϵ)-solves the q-BDDHI problem, if it runs in time t and*

$$|\Pr[\mathcal{B}(T_0)] - \Pr[\mathcal{B}(T_1)]| \geq \epsilon.$$

It is straightforward to prove the IND-snaID-CPA-security of our simplified Boneh-Boyen scheme using standard techniques from [BB04a,BB11], therefore we state the following theorem without proof.

Theorem 2. *From an adversary $\mathcal{A}$ that $(t_\mathcal{A}, q_s, \epsilon_\mathcal{A})$-breaks the* IND-snaID-CPA*-security of the simplified Boneh-Boyen ID-KEM one can construct an algorithm $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B})$-solves the q-BDDHI problem with $q = q_s + 1$ such that*

$$t_\mathcal{B} \approx t_\mathcal{A} \quad \textit{and} \quad \epsilon_\mathcal{B} = \epsilon_\mathcal{A}.$$

Adaptively secure construction.

Encoding elements of $\{0,1\}^{4(k+1)}$ as $\mathbb{Z}_p$-elements. In order to simplify the notation and description of the construction and its security analysis, we will henceforth make the implicit assumption that elements of $\{0,1\}^{4(k+1)}$ can be injectively encoded as elements of $\mathbb{Z}_p$. This is of course easily possible by choosing p large enough, such that $p > 4(k+1)$. However, this would yield an unnaturally large group order (a typical choice in practice is $2k$). In practice, one would map elements of $\{0,1\}^{4(k+1)}$ to elements in $\mathbb{Z}_p$ by using a collision-resistant hash function $h : \{0,1\}^{4(k+1)} \to \mathbb{Z}_p$, which for our purposes is as good as an injective map. However, to simplify the description of our scheme and its security proof we do not make h explicit in the sequel.

The construction. In the sequel, let $\mathcal{H} = \{H | \{0,1\}^* \to \{0,1\}^{4(k+1)}\}$ be a family of keyed hash functions and define

$$\ell := \log 4(k+1).$$

We construct ID-KEM scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ as follows.

Setup. Sample random generators $[1]_1 \in \mathbb{G}_1$, $[1]_2 \in \mathbb{G}_2$, elements $y, x_1, \ldots, x_\ell \in \mathbb{Z}_p$ and a hash function $H \leftarrow \mathcal{H}$ and define the master secret key MSK as

$$MSK = (y, x_1, \ldots, x_\ell) \in \mathbb{Z}_p^{\ell+1}.$$

Define $b_i(n)$ for positive integers i as the function that, on input of integer $n \geq 0$, outputs the i-th bit of the binary representation of n. Let $F(MSK, n)$ be the function that on input of $MSK = (x_1, \ldots, x_\ell)$ and an integer $n \geq 0$ outputs

$$F(MSK, n) = \prod_{i=1}^{\ell} x_i^{b_i(n)}.$$

The public parameters are defined as

$$PP = (H, [F(MSK, 0)]_1, \ldots, [F(MSK, 2^\ell - 1]_1, [1]_2, \nu),$$

where $\nu = e([1]_1, [y]_2)$.

Key Generation. The private key for identity id is computed as

$$USK_{id} = [y/u(id)]_2,$$

where

$$u(id) = \prod_{i=1}^{\ell} (H_{2^i}(id) + x_i) \in \mathbb{Z}_p. \tag{5}$$

Encapsulation. Observe that

$$u(id) = \prod_{i=1}^{\ell} (H_{2^i}(id) + x_i) = d_0 + \sum_{n=1}^{2^\ell - 1} \left(d_n \prod_{i=1}^{\ell} x_i^{b_i(n)}\right),$$

where the constants d_i are efficiently computable from $H(id)$.
To encapsulate a key, first $[u(id)]_1$ is computed. Note that this is possible from $H(id)$ and the values $F(MSK, n)$ contained in the public parameters (in particular, without knowing $x_1, \ldots, x_\ell$ explicitly), by computing

$$[u(id)]_1 = \left[d_0 + \sum_{n=1}^{2^\ell - 1} \left(d_n \prod_{i=1}^{\ell} x_i^{b_i(n)}\right) \right]_1 = [d_0]_1 \cdot \prod_{n=1}^{2^\ell - 1} [F(MSK, n)]_1^{d_n}.$$

Finally, the ciphertext and key are computed as

$$(C, K) = ([u(id)]_1^r, \nu^r) \in \mathbb{G}_T^2$$

for uniformly random $r \xleftarrow{\$} \mathbb{Z}_p$.

Decapsulation. To recover K from a ciphertext C for identity id and a matching user secret key $[y/(u(id))]_2$, compute and output $e(C, USK_{id})$.

Correctness. The correctness follows from

$$e(C, USK_{id}) = e([u(id)]_1^r, [y/u(id)]_2) = e([1]_1, [y]_2)^r = \nu^r.$$

Note that the scheme described above has extremely short ciphertexts, consisting of only one element of $\mathbb{G}_1$, and also very efficient decapsulation, which takes only a single pairing evaluation.

Theorem 3. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, q_\mathcal{A}, \epsilon_\mathcal{A})$-breaks the* IND-ID-CPA*-security of Π such that $t_\mathcal{A}/\epsilon_\mathcal{A} < 2^k$ and let j be an index such that (4) is satisfied. Given $\mathcal{A}$ and j, we can either construct an adversary $\mathcal{B}_j$ that $(t_\mathcal{B}, q_\mathcal{B}, \epsilon_\mathcal{B})$-breaks the* IND-snaID-CPA*-security of Π' with*

$$t_\mathcal{B} = \mathcal{O}(t_\mathcal{A}^4/\epsilon_\mathcal{A}^2), \quad q_\mathcal{B} < 4t_\mathcal{A}^4/\epsilon_\mathcal{A}^2 \quad \text{and} \quad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}^3}{32t_\mathcal{A}^4}$$

or an adversary $\mathcal{C}$ that 2^j-breaks the truncation collision resistance of $\mathcal{H}$.

Proof. The proof of Theorem 3 is almost identical to the proof of Theorem 1. The main difference is that we additionally use the algebraic structure of the underlying Boneh-Boyen ID-KEM to achieve short ciphertexts.

Setup and initial input. Just like in the proof of Theorem 1, $\mathcal{B}$ picks a random value $ID^* \xleftarrow{\$} \{0,1\}^{2^j}$ and requests a challenge ciphertext for identity ID^* and user secret keys for all $2^{2^j} - 1$ identities in the set $\{0,1\}^{2^j} \setminus \{\}$. In response, $\mathcal{B}$ receives public parameters $PP' = ([1]_1, [x_j]_1, \nu)$ from the IND-snaID-CPA experiment, as well as user secret keys $[y/(ID + x_j)]_2$ for all $ID \neq ID^*$ and a challenge ciphertext (C', K').

Additionally, $\mathcal{B}$ chooses $\ell - 1$ integers x_i for all $i \in \{1, \ldots, \ell\} \setminus \{j\}$ and a hash function $H \leftarrow \mathcal{H}$.

Simulation of the public parameters. Note that $\mathcal{B}$ is not able to compute the function $F((x_1, \ldots, x_\ell), n) = \prod_{i=1}^{\ell} x_i^{b_i(n)}$ for all values of n efficiently, since it does not know x_j. However, $\mathcal{B}$ is able to efficiently compute

$$[F((x_1, \ldots, x_\ell), n)]_1 = \left[\prod_{i=1}^{\ell} x_i^{b_i(n)}\right]_1$$

for all values of n from $[x_j]_1$ and the x_i, $i \in \{1, \ldots, \ell\} \setminus \{j\}$. This is sufficient to properly simulate a public key of scheme Π.

Simulation of user secret keys. Using the user secret keys received from the IND-snaID-CPA challenger, $\mathcal{B}$ is able to answer all secret key queries for all identities id with $H_{2^j}(id) \neq ID^*$. To this end, it computes

$$USK_{id} = \left[y/\prod_{i=1}^{\ell}(H_{2^i}(id) + x_i)\right]_2 = [y/(H_{2^j}(id) + x_j)]_2^{1/u_j(id)},$$

where

$$u_j(id) = \prod_{i=1, i\neq j}^{\ell} (H_{2^i}(id) + x_i)).$$

Creating the challenge ciphertext. $\mathcal{B}$ creates the challenge ciphertext as follows. If $\mathcal{A}$ has selected a target identity id^* with $H_{2^j}(id^*) = ID^*$, then $\mathcal{B}$ computes $C := (C')^{u_j(id^*)}$ and outputs (C, K). Note that

$$C = [(H_{2^j}(id^*) + x_j)]^r \prod_{i=1, i\neq j}^{\ell} (H_{2^i}(id^*)+x_i) = \left[\prod_{i=1}^{\ell} (H_{2^i}(id^*) + x_i)\right]_1^r$$

such that C is a correctly distributed challenge ciphertext, and K is either "real" or "random", depending on the choice of the IND-snaID-CPA security experiment.

Analysis. The analysis of the success probability of $\mathcal{B}$ is identical to the analysis from the proof of Theorem 1, and yields identical bounds. □

4 Digital signatures

Recall that the commonly accepted security notion for digital signatures is *existential unforgeability under adaptive chosen-message attacks*, as introduced by Goldwasser, Micali, and Rivest [GMR88] (EUF-CMA, see Section 4.1 for formal definitions). There are several different ways to turn signatures schemes with weaker security properties into ones with full EUF-CMA-security, even without random oracles. These are either based on one-time signatures [EGM96] or chameleon hash functions [KR00,BSW06,SPW07], and work generically for any signature scheme. However, all these generic constructions start from an *existentially-unforgeable* scheme, where the adversary has to select the "chosen-message queries", for which it requests a signature, even before seeing the public key, but is able to choose the "target-message" for which it forges a signatures adaptively (EUF-naCMA-security).

We consider the even weaker notion of *selective unforgeability under non-adaptive chosen-message attacks* (SUF-naCMA) [HW09b,BK10], where the adversary has to select both the "target-message" for which it forges a signatures and the chosen-message queries for which it requests a signature already before seeing the public key. We describe a generic construction of EUF-CMA-secure digital signatures from signatures that are only SUF-naCMA-secure. This construction is also relatively efficient: it increases the size of public keys, secret keys, and signatures by a factor of only $\mathcal{O}(\log k)$, where k is the security parameter. Again, the security reduction is non-tight, but polynomial-time.

4.1 Definitions and security notions

Definition 6. *A digital signature scheme consists of three PPT algorithms with the following syntax.*

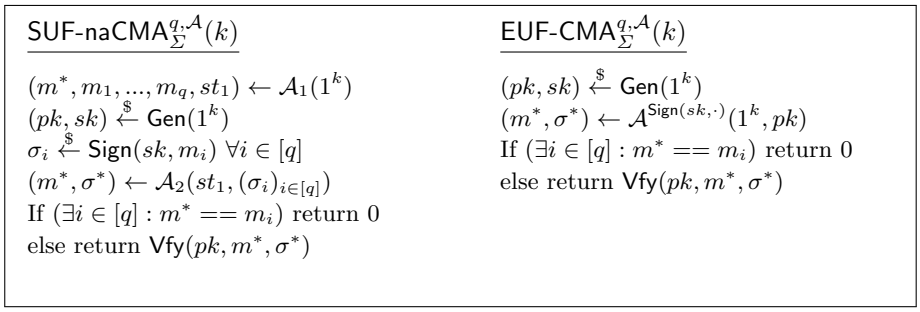

Fig. 3. The security experiments for digital signature schemes, executed with scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ and adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The oracle $\mathsf{Sign}(sk, m)$ returns $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, m)$ with the restriction that $\mathcal{A}$ is not allowed to query oracle $\mathsf{Sign}(sk, m^*)$ for m^*.

$\mathsf{Gen}(1^k)$ *outputs a key pair* (pk, sk)*. We assume that* pk *implicitly or explicitly defines a message space* $\mathcal{M}$*.*
$\mathsf{Sign}(sk, m)$ *on input of* sk *and message* $m \in \mathcal{M}$ *outputs a signature* σ*.*
$\mathsf{Vfy}(pk, m, \sigma)$ *outputs* 1 *if* σ *is a valid signature for* m *with respect to* pk *and else* 0*.*

Adaptive security. We recall the standard security notion *existential unforgeability under adaptive chosen message attack* (EUF-CMA) depicted in Figure 3. Note that the adversary may choose the challenge-message m^* after it has received the public key pk and may adaptively query signatures for messages $m_i \neq m^*$

Definition 7. *We say that adversary* $\mathcal{A}$ $(t_{\mathcal{A}}, q, \epsilon_{\mathcal{A}})$*-breaks the* EUF-CMA *security of* $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$*, if* $\Pr[\mathsf{EUF\text{-}CMA}^{q,\mathcal{A}}_{\Sigma}(k) = 1] \geq \epsilon_{\mathcal{A}}$ *and* $t_{\mathcal{A}}$ *is the running time of* $\mathcal{A}$ including *the* EUF-CMA *security experiment.*

Selective and non-adaptive security. We also define a very weak security notion for digital signature schemes. Consider the SUF-naCMA security experiment depicted in Figure 3, where the attacker has to commit to both the challenge-message m^* the signing-query messages $m_1, \ldots, m_q$ non-adaptively and even before receiving the public key pk.

Definition 8. *We say that* $\mathcal{A}$ $(t_{\mathcal{A}}, q, \epsilon_{\mathcal{A}})$*-breaks the* SUF-naCMA *security of* Σ*, if it runs in time* $t_{\mathcal{A}}$ *and* $\Pr[\mathsf{SUF\text{-}naCMA}^{q,\mathcal{A}}_{\Sigma}(k) = 1] \geq \epsilon_{\mathcal{A}}$.

4.2 From weak security to adaptive security

Construction. Let $\mathcal{H} = \{H|\{0,1\}^* \rightarrow \{0,1\}^{4(k+1)}\}$ be a family of keyed hash functions and $\Sigma' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vfy}')$ a digital signature scheme. In the sequel, let $\ell := \log 4(k+1)$. We construct our digital signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ as follows.

– **Key Generation.** Algorithm Gen computes $(pk_i, sk_i) \xleftarrow{\$} \mathsf{Gen}'(1^k)$ for all $i \in \{1, \ldots, \ell\}$ and $H \leftarrow \mathcal{H}$, defines

$$pk := (H, pk_1, \ldots, pk_\ell) \text{ and } sk = (sk_1, \ldots, sk_\ell)$$

and outputs (pk, sk).
– **Signing.** To sign a message m, compute $\sigma_i \xleftarrow{\$} \mathsf{Sign}'(sk_i, H_{2^i}(m))$ for all $i \in \{1, \ldots, \ell\}$, and return the signature $\sigma = (\sigma_1, \ldots, \sigma_\ell)$.
– **Verification.** To verify a signature $\sigma = (\sigma_1, \ldots, \sigma_\ell)$, return 1 if and only if $\mathsf{Vfy}'(pk_i, H_{2^i}(m)) = 1$ for all $i \in [\ell]$.

Theorem 4. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, q_\mathcal{A}, \epsilon_\mathcal{A})$-breaks the* EUF-CMA*-security of $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ such that $t_\mathcal{A}/\epsilon_\mathcal{A} < 2^k$ and let j be an index such that (4) is satisfied. Given $\mathcal{A}$ and j, we can either construct an adversary $\mathcal{B}_j$ that $(t_\mathcal{B}, q_\mathcal{B}, \epsilon_\mathcal{B})$-breaks the* SUF-naCMA*-security of $\Sigma' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vfy}')$ with*

$$t_\mathcal{B} = \mathcal{O}(t_\mathcal{A}^4/\epsilon_\mathcal{A}^2), \quad q_\mathcal{B} < 4t_\mathcal{A}^4/\epsilon_\mathcal{A}^2 \quad \text{and} \quad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}^3}{32t_\mathcal{A}^4}$$

or an adversary $\mathcal{C}$ that 2^j- breaks the truncation collision resistance of $\mathcal{H}$.

The proof of Theorem 4 is nearly identical to the proof of Theorem 1, except that some arguments and computing some bounds works *slightly* differently, because in the ID-KEM setting from Theorem 1 we are considering an "indistinguishability" security experiment, while in the digital signature setting of Theorem 4 we consider a "search problem". The full proof is contained in the full version [JK17].

4.3 Very short signatures with adaptive security

The generic construction of adaptively secure digital signature schemes described in Section 4.2 increases the size of keys and signatures by a factor of $\mathcal{O}(\log k)$. Again it is possible to obtain a more efficient scheme based on specific, number-theoretic constructions. In this section we describe a variant of the Boneh-Boyen signature scheme [BB04c] that applies a truncation collision resistant hash function to achieve adaptive security without random oracles, and where a signature consists of only a *single* group element. A comparison to previous short signature schemes is given in Table 4.

Building block: simplified Boneh-Boyen signatures. Again we let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be groups of prime order p with generators g_1, g_2, g_T, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an efficiently computable pairing. Recall that write $[x]_s$ shorthand for g_s^x for all $s \in \{1, 2, T\}$, following [EHK+13]. The Boneh-Boyen signature scheme [BB04c] consists of the following algorithms $\Sigma' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vfy}')$.

Key generation. Algorithm $\mathsf{Gen}'(k)$ chooses a random integer $x \xleftarrow{\$} \mathbb{Z}_p$ and defines $\nu = e([1]_1, [1]_2)$. The public keys and the secret key are defined as $pk := ([1]_1, [x]_1, [1]_2, \nu)$ and $sk := x$.

Scheme	$\|\mathbb{G}\|$	$\|pk\|$	$\|\sigma\|$	Security	Assumption	ROM	Security Loss
[BLS04]	Prime	2	1	Full	CDH	Yes	$\mathcal{O}(q_{Sig})$
[BB04c]	Prime	5	2	Selective	strong q-DH	No	$\mathcal{O}(1)$
[Wat05]	Prime	$n+3$	2	Full	CDH	No	$\mathcal{O}(n \cdot q_{Sig})$
[HJK11]	Prime	$n+k+3$	2	Full	q-DH	No	$\mathcal{O}(n^2 \cdot q_{Sig})$
[BHJ$^+$13]	Prime	$\mathcal{O}(\log k)$	3	Full	CDH	No	$\mathcal{O}((\epsilon^{-1} \cdot q_{Sig}^{m+1})^{c/m})$
[Wee16]	Comp.	3	1	Full	Dec. Subgrp.	No	$\mathcal{O}(q_{Sig})$
Ours	Prime	$\mathcal{O}(k)$	1	Full	q-DH	No	$\mathcal{O}(t_{\mathcal{A}}^7/\epsilon_{\mathcal{A}}^4)$

Fig. 4. Comparison of short signature schemes, instantiated with asymmetric pairings. The column $|\mathbb{G}|$ refers to the order of the underlying groups (prime or composite), $|pk|$ is the number of group elements in public keys, where common descriptions of groups and hash functions are not included, n is the length of messages, and k the security parameter. All public keys include one element from the target group of the pairing, except for [BLS04,HJK11,BHJ$^+$13]. The column $|\sigma|$ refers to the number of group elements in the signature. "Full" security means EUF-CMA security as defined below, "selective" security is from [BB04c]. The remaining columns state the assumption the proof is based on, whether the Random Oracle Model is used, and the security loss of the reduction, where q_{Sig} is the number of signing queries, $t_{\mathcal{A}}$ and $\epsilon_{\mathcal{A}}$ the running time and advantage of the adversary, and the loss is computed as explained in Figure 2. The values m and c are system parameters influencing keys and signature sizes. Note that [HJK11] present also other trade-offs with larger public keys consisting and shorter signatures, but always strictly larger than one group element.

Signing. Algorithm Sign' receives as input $sk = x$ and message $m \in \mathbb{Z}_p$, and computes and returns $\sigma := [1/(x+m)]_2 \in \mathbb{G}_2$.

Verification. Algorithm Vfy' takes as input a public key $pk = ([1]_1, [x]_1, \nu) \in \mathbb{G}_1^2 \times \mathbb{G}_2$, message $m \in \mathbb{Z}_p$, and $\sigma \in \mathbb{G}_2$. It returns 1 if and only if

$$e\left([x]_1 \cdot [1]_1^m, \sigma\right) = \nu.$$

Security. The original paper by Boneh and Boyen [BB04c] proves security of this scheme in the sense of *existential unforgeability under non-adaptive chosen-message attacks* (EUF-naCMA), under the *strong* (or "flexible") q-Diffie-Hellman assumption. We will require only a weaker notion of security, in the sense of *selective unforgeability against non-adaptive chosen message attacks* (SUF-naCMA), which is achievable under a weaker, "non-flexible" q-type assumption.

Definition 9. *We say that adversary $\mathcal{A}$ $(\epsilon_{\mathcal{A}}, t_{\mathcal{A}})$-breaks the* q-Diffie-Hellman assumption *in group $\mathbb{G}$ of order p, if it runs in time $t_{\mathcal{A}}$ and*

$$\Pr\left[x \xleftarrow{\$} \mathbb{Z}_p; h \xleftarrow{\$} \mathcal{A}([1], [x], [x^2], \ldots, [x^q]) : h = [1/x]\right] \geq \epsilon_{\mathcal{A}}.$$

The above assumption is also known as the q-Diffie-Hellman Inversion assumption [ZSS04]. By using the "generator-shifting" technique of [HJK11], one can prove the following theorem along the lines of the original proof of Boneh and Boyen [BB04c].

Theorem 5. *From an adversary $\mathcal{A}$ that $(t_{\mathcal{A}}, q_s, \epsilon_{\mathcal{A}})$-breaks the* SUF-naCMA*-security of Σ' chosen-message queries, one can canstruct an adversary $\mathcal{B}$ that $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks the q-Diffie-Hellman assumption with $q = q_s + 1$, $t_{\mathcal{B}} \approx t_{\mathcal{A}}$ and $\epsilon_{\mathcal{B}} = \epsilon_{\mathcal{A}}$.*

Encoding elements of $\{0,1\}^{4(k+1)}$ as $\mathbb{Z}_p$-elements. In order to simplify the notation and description of the construction and its security analysis, we will henceforth make the implicit assumption that elements of $\{0,1\}^{4(k+1)}$ can be injectively encoded as elements in $\mathbb{Z}_p$ (see also the corresponding, more detailed comment in Section 3.3).

Construction. Let $\mathcal{H} = \{H|\{0,1\}^* \to \{0,1\}^{4(k+1)}\}$ be a family of keyed hash functions and $\ell := \log 4(k+1)$. We construct signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ as follows.

Key generation. Algorithm $\mathsf{Gen}(k)$ chooses ℓ random integers $x_1, \ldots, x_\ell \xleftarrow{\$} \mathbb{Z}_p$ and $H \leftarrow \mathcal{H}$. It defines the secret key as $sk := (x_1, \ldots, x_\ell) \in \mathbb{Z}_p^\ell$. Note that sk contains only $\ell = \log 4(k+1)$ elements of $\mathbb{Z}_p$.

The public key is computed as follows. For a positive integer $i \geq 1$, let $b_i(n)$ be the function that, on input of integer $n \geq 0$, outputs the i-th bit of the (canonical) binary representation of n. Let $F(sk, n)$ be the function that, on input of $sk = (x_1, \ldots, x_\ell)$ and integer $n \geq 0$, outputs

$$F(sk, n) := \prod_{i=1}^{\ell} x_i^{b_i(n)}.$$

The public key is defined as $pk := (H, [F(sk,0)]_1, \ldots, [F(sk, 2^\ell - 1)]_1, [1]_2, \nu)$, where $\nu = e([1]_1, [1]_2)$.

Signing. Algorithm Sign receives as input $sk = (x_1, \ldots, x_\ell)$ and message $m \in \{0,1\}^*$. Let $u(m)$ be the function

$$u(m) := \prod_{i=1}^{\ell} (x_i + H_{2^i}(m)) \in \mathbb{Z}_p, \tag{6}$$

where bit strings $H_{2^i}(m)$ are interpreted canonically as integers in $\mathbb{Z}_p$. Recall here that by our assumption on p this is injective for all $i \in \{1, \ldots, \ell\}$.

The signing algorithm computes and returns $\sigma := [1/u(m)]_2 \in \mathbb{G}_1$.

Note that computing signatures is *extremely* efficient. It involves only the computation of $1/u(m) \in \mathbb{Z}_p$, which can be performed over the integers modulo p, where p is the group order, and then a *single* exponentiation in $\mathbb{G}_1$ to compute $g_1^{1/u(m)} \in \mathbb{G}_1$.

Verification. Algorithm Vfy takes as input a public key

$$pk = ([F(sk,0)]_1, \ldots, [F(sk, 2^\ell - 1)]_1, [1]_2, \nu),$$

message $m \in \{0,1\}^*$ and $\sigma \in \mathbb{G}_2$. Note here that $[F(sk,0)]_1 = [1]_1$. The algorithm returns 1 if and only if

$$e\left([u(m)]_1, \sigma\right) = \nu. \tag{7}$$

Here $[u(m)]_1$ is computed as follows. Viewing $u(m) = \prod_{i=1}^{\ell}(x_i + H_{2^i}(m))$ as a polynomial in ℓ unknowns $x_1, \ldots, x_\ell$, we can expand the product from (6) to obtain the equation

$$u(m) = \prod_{i=1}^{\ell}(x_i + H_{2^i}(m)) = d_0 + \sum_{n=1}^{2^\ell - 1}\left(d_n \prod_{i=1}^{\ell} x_i^{b_i(n)}\right) \tag{8}$$

for integers d_i, which are efficiently computable from $H(m)$. This yields the equation

$$[u(m)]_1 = \left[d_0 + \sum_{n=1}^{2^\ell - 1}\left(d_n \prod_{i=1}^{\ell} x_i^{b_i(n)}\right)\right]_2 = [d_0]_2 \cdot \prod_{n=0}^{2^\ell - 1}[F(sk,n)]_2^{d_n}. \tag{9}$$

Therefore the verification algorithms proceeds as follows:

1. From $H(m)$ it computes the integers d_i as in (8) .
2. Then it computes $[u(m)]_1$ as in (9) from the group elements $[F(sk,n)]_1$ contained in the public key.
3. Finally, it outputs 1 if and only if Equation (7) holds.

Theorem 6. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, q_\mathcal{A}, \epsilon_\mathcal{A})$-breaks the* EUF-CMA*-security of Σ such that $t_\mathcal{A}/\epsilon_\mathcal{A} < 2^k$ and let j be an integer that such that (4) is satisfied. Given $\mathcal{A}$ and j we can either construct an adversary $\mathcal{B}_j$ that $(t_\mathcal{B}, q_\mathcal{B}, \epsilon_\mathcal{B})$-breaks the* SUF-naCMA *security of the Boneh-Boyen signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vfy}')$ with*

$$t_\mathcal{B} = \mathcal{O}(t_\mathcal{A}^4/\epsilon_\mathcal{A}^2), \qquad q_\mathcal{B} < 4t_\mathcal{A}^4/\epsilon_\mathcal{A}^2 \qquad \text{and} \qquad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}^3}{32t_\mathcal{A}^4}$$

or and adversary $\mathcal{C}$ that 2^j-breaks the truncation collision resistance of $\mathcal{H}$.

The proof of Theorem 6 is almost identical to the proofs of Theorems 3 and 4. It is contained in the full version [JK17].

5 Conclusion

Truncation collision resistance enables very efficient generic constructions of adaptively-secure cryptographic primitives from building blocks with very weak selective and non-adaptive security. We showed this for identity-based encryption and digital signatures, but expect further useful applications to other cryptographic primitives.

Two particularly interesting applications are the first standard-model constructions of an ID-KEM where a ciphertext consist of only a single group element of a prime-order group, and a digital signature scheme where signatures consist of only a single prime-order group element. Both achieve full adaptive security. Previously, it was not clear that this is possible without random oracles and based on simple, non-interactive hardness assumptions.

References

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany.

[BB04c] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[BB11] Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, October 2011.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

[BFMLS08] Kamel Bentahar, Pooya Farshim, John Malone-Lee, and Nigel P. Smart. Generic constructions of identity-based and certificateless KEMs. *Journal of Cryptology*, 21(2):178–199, April 2008.

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.

[BH15] Itay Berman and Iftach Haitner. From non-adaptive to adaptive pseudorandom functions. *Journal of Cryptology*, 28(2):297–311, April 2015.

[BHJ+13] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, Jae Hong Seo, and Christoph Striecks. Practical signatures from standard assumptions. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 461–485, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 398–415, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[BK10] Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model.

Cryptology ePrint Archive, Report 2010/086, 2010. http://eprint.iacr.org/2010/086.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.

[BR09] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424, Cologne, Germany, April 26–30, 2009. Springer, Heidelberg, Germany.

[BSW06] Dan Boneh, Emily Shen, and Brent Waters. Strongly unforgeable signatures based on computational Diffie-Hellman. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006*, volume 3958 of *LNCS*, pages 229–240, New York, NY, USA, April 24–26, 2006. Springer, Heidelberg, Germany.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 455–469, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany.

[CFN15] Dario Catalano, Dario Fiore, and Luca Nizzardo. Programmable hash functions go private: Constructions and applications to (homomorphic) signatures with shorter public keys. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 254–274, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218, Dallas, TX, USA, May 23–26, 1998. ACM Press.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[DG17a] Nico Döttling and Sanjam Garg. From selective ibe to full ibe and selective hibe. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography*, pages 372–408, Cham, 2017. Springer International Publishing.

[DG17b] Nico Döttling and Sanjam Garg. Identity-based encryption from the diffie-hellman assumption. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 537–569, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[DS15] Nico Döttling and Dominique Schröder. Efficient pseudorandom functions via on-the-fly adaptation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 329–350, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[EGM96] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[FF13] Marc Fischlin and Nils Fleischhacker. Limitations of the meta-reduction technique: The case of Schnorr signatures. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 444–460, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[FHPS13] Eduarda S. V. Freire, Dennis Hofheinz, Kenneth G. Paterson, and Christoph Striecks. Programmable hash functions in the multilinear setting. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 513–530, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[FLR+10] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 303–320, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany.

[FM16] Pooya Farshim and Arno Mittelbach. Modeling random oracles under unpredictable queries. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 453–473, Bochum, Germany, March 20–23, 2016. Springer, Heidelberg, Germany.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[HJK11] Dennis Hofheinz, Tibor Jager, and Eike Kiltz. Short signatures from weaker assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 647–666, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany.

[HK08] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 21–38, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany.

[HMS12] Goichiro Hanaoka, Takahiro Matsuda, and Jacob C. N. Schuldt. On the impossibility of constructing efficient key encapsulation and programmable hash functions in prime order groups. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 812–831, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

[HW09a] Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 333–350, Cologne, Germany, April 26–30, 2009. Springer, Heidelberg, Germany.

[HW09b] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 654–670, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Heidelberg, Germany.

[JK17] Tibor Jager and Rafael Kurek. Short digital signatures and ID-based KEMs via truncation collision resistance. Cryptology ePrint Archive, Report 2017/061, 2017. Full version of an ASIACRYPT 2018 paper, `https://eprint.iacr.org/2017/061`.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*, San Diego, CA, USA, February 2–4, 2000. The Internet Society.

[LOS+06] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, and Brent Waters. Sequential aggregate signatures and multisignatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 465–485, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.

[Nat15a] National Institute of Standards and Technology. *FIPS PUB 180-4: Secure Hash Standard*. August 2015. `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf`.

[Nat15b] National Institute of Standards and Technology. *FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. August 2015. `https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf`.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 111–126, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Heidelberg, Germany.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 371–388, New Delhi, India, February 5–7, 2004. Springer, Heidelberg, Germany.

[SPW07] Ron Steinfeld, Josef Pieprzyk, and Huaxiong Wang. How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In Masayuki Abe, editor, *CT-RSA 2007*, volume 4377 of *LNCS*, pages 357–371, San Francisco, CA, USA, February 5–9, 2007. Springer, Heidelberg, Germany.

[Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.

[Wee16] Hoeteck Wee. Déjà Q: Encore! Un petit IBE. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 237–258, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.

[Zha16] Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 479–508, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[ZSS04] Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. An efficient signature scheme from bilinear pairings and its applications. In Feng Bao, Robert Deng, and Jianying Zhou, editors, *PKC 2004*, volume 2947 of *LNCS*, pages 277–290, Singapore, March 1–4, 2004. Springer, Heidelberg, Germany.