Efficient Adaptively-Secure IB-KEMs and VRFs via Near-Collision Resistance*

Tibor Jager[1][0000−0002−3205−7699], Rafael Kurek[1], and David Niehues[2][0000−0001−9169−7867]

[1] Bergische Universität Wuppertal, Wuppertal, Germany,
tibor.jager@uni-wuppertal.de, rafael.kurek@rub.de
[2] Paderborn University, Paderborn, Germany, david.niehues@upb.de

Abstract. We construct more efficient cryptosystems with provable security against *adaptive* attacks, based on simple and natural hardness assumptions in the standard model. Concretely, we describe:

- An adaptively-secure variant of the efficient, selectively-secure LWE-based identity-based encryption (IBE) scheme of Agrawal, Boneh, and Boyen (EUROCRYPT 2010). In comparison to the previously most efficient such scheme by Yamada (CRYPTO 2017) we achieve smaller lattice parameters and shorter public keys of size $\mathcal{O}(\log \lambda)$, where λ is the security parameter.
- Adaptively-secure variants of two efficient selectively-secure pairing-based IBEs of Boneh and Boyen (EUROCRYPT 2004). One is based on the DBDH assumption, has the same ciphertext size as the corresponding BB04 scheme, and achieves full adaptive security with public parameters of size only $\mathcal{O}(\log \lambda)$. The other is based on a q-type assumption and has public key size $\mathcal{O}(\lambda)$, but a ciphertext is only a single group element and the security reduction is quadratically tighter than the corresponding scheme by Jager and Kurek (ASIACRYPT 2018).
- A very efficient adaptively-secure verifiable random function where proofs, public keys, and secret keys have size $\mathcal{O}(\log \lambda)$.

As a technical contribution we introduce *blockwise partitioning*, which leverages the assumption that a cryptographic hash function is *weak near-collision resistant* to prove full adaptive security of cryptosystems.

1 Introduction

A very fundamental question in cryptography is to which extent idealizations like the random oracle model [7] are necessary to obtain practical constructions of cryptosystems. By advancing our techniques to prove security of schemes, we may

* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre On-The-Fly Computing (GZ: SFB 901/3) under the project number 160364472 and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement 802823.

eventually be able to obtain standard-model schemes that are about as efficient as corresponding schemes with security proofs in the ROM. From a practical perspective, it would be preferable to have security guarantees that are not based on an uninstantiable model [20]. From a theoretical perspective, it allows us to understand when a random oracle is necessary, and when not. For some primitives it is known that a programmable random oracle is indeed inherently necessary [25, 27, 31, 46]. But for many others, including those considered in this paper, there are no such impossibility results.

In the context of identity-based encryption the established standard security notion [16] considers an adversary which is able to choose the identities for which it requests secret keys or a challenge ciphertext *adaptively* in the security experiment. This yields much stronger security guarantees than so-called *selective* security definitions [14], where the adversary has to announce the "target identity" associated with a challenge ciphertext at the beginning of the security experiment, even before seeing the public parameters.

"Selective" security is much easier to achieve and therefore yields more efficient constructions. The random oracle model is then a useful tool to generically convert a selectively-secure scheme into an adaptively-secure one. This has negligible performance overhead, and thus yields an efficient and adaptively-secure construction. This generic construction is based on the fact that a random oracle is "programmable", which essentially means that it is possible to adaptively modify the mapping of function inputs to outputs in a way that is convenient for the security proof. While this is very useful to achieve efficient and adaptively-secure constructions, it is often considered a particularly unnatural property of the random oracle model, due to the fact that no fixed function can be as freely adaptively programmed as a random oracle.

There exist techniques to achieve adaptive security in the standard model by realizing certain properties of a random oracle with a concrete construction (i.e., in the standard model). This includes *admissible hash functions* [15], *programmable hash functions* [22, 28, 33, 34, 51], and *extremely lossy functions* [55]. However, these typically yield significantly less efficient constructions and are therefore less interesting for practical applications than corresponding constructions in the random oracle model.

A recent, quite different approach that addresses this issue is to use *truncation collision resistance* [37] of a cryptographic hash function to achieve adaptive security. In contrast to the aforementioned approaches, this does not introduce a new "algebraic" construction of a hash function. Instead, their idea is to formulate a concrete hardness assumption that on the one hand is "weak enough" to appear reasonable for *standard* cryptographic hash functions, such as SHA-3, but which at the same time is "strong enough" to be used to achieve adaptive security. It is shown that this indeed yields very efficient and adaptively-secure constructions, such as identity-based encryption with a single group element overhead and digital signatures that consist of a single group element. Notably, truncation collision resistance is also achieved by a non-programmable random oracle, even though this security notions is considered as a (non-standard, but

seemingly reasonable) security notion for standard-model cryptographic hash functions. However, the main disadvantages of the constructions in [37] are that very strong computational hardness assumptions (so-called q-type assumptions with very large q) are required, and that the reductions are extremely non-tight.

Our contributions. We introduce *blockwise partitioning* as a new approach to leverage the assumption that a cryptographic hash function is *weak near-collision resistant*. We informally say that a hash functions is weak near-collision resistant if the generic birthday attack is the fastest algorithm to find collisions on a subset of the output bits, where the subset has to be stated in advance. We formally introduce weak near-collision resistance in Definition 1. It can be seen as a new variant of truncation collision resistance [37], which essentially captures the same intuition and therefore can be considered equally reasonable. However, we will show that our technique yields more efficient and tighter constructions of identity-based encryption, based on lattices and on pairings, and a highly efficient new verifiable random function. We give a more detailed comparison between blockwise partitioning based on weak near-collision resistance and the results from [37] in Section 2.

Near-collision resistance of standardized hash functions. The near-collision resistance of hash functions has been studied in several works and has been shown to be an important property of hash functions [10, 11, 48]. Further, the Handbook of Applied Cryptography [44, Remark 9.22] lists near-collision resistance as a desired property of hash functions and a potential *certificational property*. Moreover, the sponge construction for hash functions, which SHA-3 is based on, has been shown to be indifferentiable from a random oracle [9], in a slightly idealized model. This immediately implies the near-collision resistance of the sponge construction in this model. Since weak near-collision resistance is an even weaker property, we view it as a natural property of modern hash functions.

Lattice-based IB-KEM. We apply our approach to construct a lattice-based IB-KEM with constant size ciphertexts and public keys of size $O(\log \lambda)$. This scheme has efficiency close to existing selectively-secure ones, which makes progress towards answering an open problem posed in Peikert's survey on lattice cryptography [47] on the existence of adaptively-secure schemes whose efficiency is comparable to selectively-secure ones. We compare the efficiency of existing schemes in Table 1, which is based on the respective table by Yamada [53], and discuss the used techniques in the full version [38].

Pairing-based IB-KEM. We also construct two new variants of the pairing-based identity-based encryption schemes of Boneh and Boyen [14] and Waters [51]. In comparison to [14] we achieve adaptive security instead of selective security. In comparison to [51] we have public parameters of size $O(\log \lambda)$ instead of $O(\lambda)$. Security is based on the same algebraic complexity assumption as the original schemes plus weak near-collision resistance. The security analysis is also much

Schemes	$\lvert\mathsf{mpk}\rvert$ **# of** $\mathbb{Z}_q^{n\times m}$ **matr.**	$\lvert\mathsf{usk}\rvert, \lvert\mathsf{ct}\rvert$ **# of** $\mathbb{Z}_q^m$**vec.**	**LWE param** $1/\alpha$	**Reduction Cost**	**Remarks**
[21]	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	$\tilde{\mathcal{O}}(n^{1.5})$	$\mathcal{O}(\varepsilon^{\nu+1}/Q^{\nu})^{\ddagger}$	
[2]+ [18]	$\mathcal{O}(\lambda)$	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}(n^{5.5})$	$\mathcal{O}(\varepsilon^2/qQ)$	
[52]	$\mathcal{O}(\lambda^{1/\mu})^{\dagger}$	$\mathcal{O}(1)$	$n^{\omega(1)}$	$\mathcal{O}(\varepsilon^{\mu+1}/kQ^{\mu})^{\dagger}$	
[56]	$\mathcal{O}(\log Q)$	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}(Q^2n^{6.5})$	$\mathcal{O}(\varepsilon/kQ^2)$	Q-bounded
[4]*	$\mathcal{O}(\lambda/\log^2\lambda)$	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}(n^6)$	$\mathcal{O}(\varepsilon^2/qQ)$	
[19]	$\mathcal{O}(\lambda)$	$\mathcal{O}(1)$	$\mathsf{superpoly}(n)$	$\mathcal{O}(\lambda)$	
[41]	$\mathcal{O}(\lambda^{1/\mu})^{\dagger}$	$\mathcal{O}(1)$	$\mathcal{O}(n^{2.5+2\mu})^{\dagger}$	$\mathcal{O}((\lambda^{\mu-1}\varepsilon^{\mu}/Q^{\mu})^{\mu+1})^{\dagger}$	Ring-based
[53] + $\mathsf{F}_{\mathsf{MAH}}$ §	$\mathcal{O}(\log^3\lambda)$	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}(n^{11})$	$\mathcal{O}(\varepsilon^{\nu+1}/Q^{\nu})^{\ddagger}$	
[53] + $\mathsf{F}_{\mathsf{AFF}}$ §	$\mathcal{O}(\log^2\lambda)$	$\mathcal{O}(1)$	$\mathsf{poly}(\lambda)$	$\mathcal{O}(\varepsilon^2/k^2Q)$	Expensive offline phase
Sec. 3	$\mathcal{O}(\log\lambda)$	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}(n^6)$	$\mathcal{O}(\varepsilon^2/t^2)$	

Table 1. Comparison of Adaptively Secure IBEs based on LWE in the standard model

We compare adaptively secure IBE schemes under the LWE assumption that do not use random oracles. We measure the size of ct and usk in the number of $\mathbb{Z}_q^m$ vectors and the size of mpk in the number of $\mathbb{Z}_q^{n\times m}$ matrices. Q, ε and t, respectively, denote the number of queries, the advantage against the security of the respective IBE, and the runtime of an adversary. We measure the reduction cost by the advantage of the algorithm solving the LWE problem that is constructed from the adversary against the IBE scheme. All reduction costs were computed using the technique of Bellare and Ristenpart [6].

† The constant $\mu \in \mathbb{N}$ can be chosen arbitrarily. However, the reduction cost degrades exponentially in μ and hence it should be chosen rather small.

‡ $\nu > 1$ is the constant satisfying $c = 1 - 2^{-1/\nu}$, where c is the relative distance of an underlying error correcting code. ν can be chosen arbitrarily close to one by choosing c closer to $1/2$ [30]. However, this comes with larger public keys as shown in [39].

$*$ The authors also propose an additional scheme that we do not include, because it relies on much stronger complexity assumptions.

§ Yamada [53] provides two instantiations of his IBE, one based on a modified admissible hash function ($\mathsf{F}_{\mathsf{MAH}}$) and one based on affine functions ($\mathsf{F}_{\mathsf{AFF}}$). When Yamada's scheme is used with the second instantiation, the key generation and encryption need to compute the description of a branching program that computes the division. This makes the construction less efficient.

simpler than in [51] or the simplified proof by Bellare and Ristenpart [6] and does not require an "artificial abort" [51]. To our best knowledge, this is the first adaptively-secure IBE scheme where ciphertexts consist only of *two* elements of a prime order algebraic group with logarithmic-size public parameters. The scheme also gives rise to an adaptively-secure (EUF-CMA) CDH-based digital signature scheme with logarithmic-size keys. See Table 2.

We also describe a new *adaptively-secure* variant of a scheme by Boneh and Boyen [14] based on a q-type assumption where a ciphertext consists only of a *single* group element. In comparison to the corresponding construction from [37], the q of the required q-type assumption is reduced *quadratically*, while the tightness of the reduction is improved *quadratically*, too. This scheme also gives rise to a signature scheme with adaptive security in the standard model, where a signature is only a single element from a prime-order group, which achieves the same quadratic improvement over a construction from [37].

Scheme	$\|\mathsf{mpk}\|$	$\|\mathsf{usk}\|$	$\|\mathsf{ct}\|$	**Security**	**Assumption**	**ROM**	**Security Loss**
[16]	2	1	1	adap.	DBDH	Yes	$O(q_{key})$
[51]	$\mathcal{O}(\lambda)$	2	2	adap.	DBDH	No	$O(t^2 + (\lambda \cdot q_{key} \cdot \varepsilon^{-1})^2)$
[50]	13	9	10	adap.	DLIN+DBDH	No	$O(q_{key})$
[43]	25	6	6	adap.	DLIN	No	$O(q_{key})$
[23]	9	4	4	adap.	SXDH	No	$O(q_{key})$
[5]	$O(\lambda)$	8	8	adap.	DLIN	No	$O(\log(\lambda))$
Sec. 4.1	$O(\log \lambda)$	2	2	adap.	DBDH	No	$O(t_{\mathcal{A}}^2/\varepsilon_{\mathcal{A}})$
[14]	4	2	2	selec.	qDBDHI	No	$O(1)$
[29]	3	2	3	adap.	qABDHE	No	$1 + \mathcal{O}(q_{key}^2)/t_{\mathcal{A}}$
[37]	$O(\lambda)$	1	1	adap.	qDBDHI	No	$O(t_{\mathcal{A}}^7/\varepsilon_{\mathcal{A}}^4)$
Sec. 4.2	$O(\lambda)$	1	1	adap.	qDBDHI	No	$O(t_{\mathcal{A}}^3/\varepsilon_{\mathcal{A}}^2)$

Table 2. Comparison of IB-KEMs based on pairings with prime order groups and short ciphertexts. $|\mathsf{mpk}|$ is the number of group elements in public keys (descriptions of groups and hash functions not included), λ the security parameter. All public keys include at least one element from the target group of the pairing, except for [16]. $|\mathsf{usk}|$ and $|\mathsf{ct}|$ are the respective numbers of group elements in the user secret keys and ciphertexts when viewed as a KEM. "adap." means adaptive IND-ID-CPA security as defined below, "selec." is selective security in the sense of [14]. The security loss is defined as the value L that satisfies $t_{\mathcal{B}}/\varepsilon_{\mathcal{B}} = L \cdot t_{\mathcal{A}}/\varepsilon_{\mathcal{A}}$, where $t_{\mathcal{A}}$,$\varepsilon_{\mathcal{A}}$ and $t_{\mathcal{B}}$,$\varepsilon_{\mathcal{B}}$ are the respective running time and advantage of the adversary and reduction, and we ignored negligible terms in the security loss. q_{key} is the number of identity key queries.

Pairing-based VRF. As our last contribution, we construct a new VRF based on the q-DBDHI assumption by using blockwise partitioning and techniques of Yamada's VRF [53]. Our VRF is the first to achieve both small public keys and small proofs at the same time. Furthermore, the size of the keys and proofs is not only asymptotically small but also concretely: for $\lambda = 128$, public keys of

Schemes	$\lvert\mathsf{vk}\rvert$	$\lvert\pi\rvert$	Assumption	Security loss
[35]	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda \cdot Q)$-DDHE	$\mathcal{O}(\lambda Q/\varepsilon)$
[17]	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$-DDH	$\mathcal{O}(\lambda)$
[36]	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	$\mathcal{O}(\log(Q/\varepsilon))$-DDH	$\mathcal{O}(Q^\nu/\varepsilon^{\nu+1})$
[32]	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	DLIN	$\mathcal{O}(\lambda \log(\lambda) Q^{2/c}/\varepsilon^3)$
[53] Sec. 6.1	$\omega(\lambda \log^2 \lambda)^\dagger$	$\omega(\log^2 \lambda)^\dagger$	$\tilde{\mathcal{O}}(\lambda)$-DDH	$\mathcal{O}(Q^\nu/\varepsilon^{\nu+1})$
[53] Sec. 6.2	$\omega(\log^2 \lambda)^\dagger$	$\omega(\sqrt{\lambda} \log^2 \lambda)^\dagger$	$\tilde{\mathcal{O}}(\lambda)$-DDH	$\mathcal{O}(Q^\nu/\varepsilon^{\nu+1})$
[54] App. C.	$\omega(\log^2 \lambda)^\dagger$	$\mathsf{poly}(\lambda)$	$\mathsf{poly}(\lambda)$-DDH	$\mathcal{O}(\lambda^2 Q/\varepsilon^2)$
[40] Sec. 5.1	$\omega(\log^2 \lambda)^\dagger$	$\omega(\lambda \log^2 \lambda)^\dagger$	$\omega(\log^2 \lambda)^\dagger$-DDH	$\mathcal{O}(Q^\nu/\varepsilon^{\nu+1})$
[40] Sec. 5.3	$\omega(\sqrt{\lambda} \log \lambda)^\dagger$	$\omega(\log \lambda)^\dagger$	$\omega(\log^2 \lambda)^\dagger$-DDH	$\mathcal{O}(Q^\nu/\varepsilon^{\nu+1})$
[49]	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	DLIN	$\mathcal{O}(\lambda \log(\lambda) Q^{2/c}/\varepsilon^3)$
[42]	$\omega(\lambda \log \lambda)^\dagger$	$\omega(\log \lambda)^\dagger$	DLIN	$\mathcal{O}(\lvert\pi\rvert \log(\lambda) Q^{2/\nu}/\varepsilon^3)$
[42]	$\omega(\lambda^{2+2\eta})$	$\omega(1)^\dagger$	DLIN	$\mathcal{O}(\lvert\pi\rvert \log(\lambda) Q^{2+2/\nu}/\varepsilon^3)$
[39]	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	$\mathcal{O}(t^2/\varepsilon)$-DDH	$\mathcal{O}(t^3/\varepsilon^2)$
Sec. 5	$\mathcal{O}(\log \lambda)$	$\mathcal{O}(\log \lambda)$	$\mathcal{O}(t^2/\varepsilon)$-DDH	$\mathcal{O}(t^2/\varepsilon^2)$

Table 3. Comparison of Adaptively Secure VRFs in the standard model

We compare adaptively secure VRF schemes in the standard model. We measure the size of vk and π in the number of the respective group. Q, ε and t respectively denote the number of queries an adversary makes, the adversaries advantage against the security of the respective VRF and the adversaries runtime. Most of the constructions use an error correcting code $C : \{0,1\}^\lambda \to \{0,1\}^n$ with constant relative minimal distance $c \leq 1/2$, where n,$\nu > 1$ can be chosen arbitrarily close to 1 by choosing c arbitrarily close to $1/2$ [30, Appendix E.1]. However, this leads to larger n and by that to larger public keys and/or proofs as shown in [39].

† Note that these terms only hold for "λ large enough" and therefore, key and proof sizes might have to be adapted with larger constants in order to guarantee adequate security.

our VRF consist of only 10 group elements and proofs of only 9 group elements. We compare the efficiency of existing schemes in Table 3, which is based on the respective table by Kohl [42], and discuss the used techniques in the full version [38].

2 Blockwise Partitioning via Near-Collision Resistance

High-level approach. *Confined guessing* [12,13] is a semi-generic technique to construct efficient and adaptively-secure digital signature schemes. It has been used for instance in [3, 24]. Unfortunately, it is only applicable to signatures, but neither to identity-based schemes such as identity-based key encapsulation mechanisms (IB-KEMs), nor to verifiable random functions (VRFs).

We propose *blockwise partitioning* as a new semi-generic technique, and show how it can be used to construct efficient IB-KEMs and VRFs with adaptive security. It is based on the *near-collision resistance* of a cryptographic hash function and similar in spirit to the closely related notion of *truncation collision resistance* [37].

Explained on an informal level using IB-KEMs as example, our approach is to let the reduction guess $n' = \mathcal{O}(\log \lambda)$, many bits of $H(\mathsf{id}^*)$, where λ is the security parameter, H is a collision resistant hash function and id^* is the challenge identity chosen by the adversary. Then, the reduction is successful if the guess matches $H(\mathsf{id}^*)$ on all n' guessed bits and the hash of every identity queried by the adversary differs in at least one bit from the guess. For this approach to yield a reduction with non-negligible loss, we have to choose n' such that it fulfills the following two conflicting goals: n' has to be small enough for the probability of guessing n' bits of $H(\mathsf{id}^*)$ correctly to be non-negligible, but we also have to choose n' large enough to ensure that it is unlikely, relative to the adversaries advantage, for the adversary to make a query id whose hash also matches on the n' guessed bits. Like [12, 13, 37], we balance these two goals by choosing n' depending on the runtime and advantage of the adversary. Following this approach thus yields an ideal choice of n' for each adversary. Constructions like [12,13,37], however, do not use this ideal choice but the next largest power of two as n' and then guess the first n' bits of $H(\mathsf{id}^*)$. This has the advantage that it leaves only $\mathcal{O}(\log \lambda)$ many possibilities for n' and hence yields small key sizes. Unfortunately, this comes at the cost of a larger security loss in the reduction because n' can be almost double the size of the ideal choice. Furthermore, choosing n' in this sub-optimal manner also requires stronger q-type assumptions and a hash function with longer outputs.

We address this issue by viewing the output of the hash function as the concatenation of blocks of exponentially growing length, i.e. the first bit is the first block, bits two and three are the second block, bits four, five, six and seven are the third block and so on. Our reduction then uses the ideal choice for n' and guesses the bits in the blocks whose lengths sum up to exactly n'. This more fine-grained guessing yields constructions with tighter security from weaker assumptions. Furthermore, it reduces the required output length of the hash

function from $4(\lambda+1)$ bits in [37] to only $2\lambda+3$ bits. Note that this is essentially optimal for a collision-resistant hash function. In particular, for many practical constructions one would probably use a collision resistant hash function, anyway, to map long identities to short strings. We compare our techniques to the ones of [37] in more detail after formally introducing blockwise partitioning.

In the remainder of this section we will describe the framework and assumptions for blockwise partitioning, give some more technical intuition, and state and prove a technical lemma that will be useful to use blockwise partitioning as modular as possible in security proofs.

Blockwise partitioning. Let $H : \{0,1\}^* \to \{0,1\}^n$ be a hash function. We will assume in the sequel that $n = \sum_{i=0}^{\ell} 2^i$ for simplicity and ease of exposition. One can generalize this to arbitrary n, but this would make the notation rather cumbersome without providing additional insight or clarity. Then we can view the output space $\{0,1\}^n$ of the hash function as a direct product of sets of exponentially-increasing size

$$\{0,1\}^n = \{0,1\}^{2^0} \times \cdots \times \{0,1\}^{2^\ell}.$$

For a hash function H we define functions $H_0, \ldots, H_\ell$ such that

$$H_i : \{0,1\}^* \to \{0,1\}^{2^i} \qquad \text{and} \qquad H(x) = H_0(x)|| \cdots ||H_\ell(x).$$

One can consider each $H_i(x)$ as one "block" of $H(x)$. Note that blocks have exponentially increasing size and there are $\lfloor \log n \rfloor + 1$ blocks in total.

Using blockwise partitioning. Let $t = t(\lambda)$ be a polynomial and let $\varepsilon = \varepsilon(\lambda)$ be a non-negligible function such that $\varepsilon > 0$ and $t/\varepsilon < 2^\lambda$ for all λ. Think of t and ε as (approximations of) the running time and advantage of an adversary in a security experiment. We define an integer n' depending on (t, ε) as

$$n' := \lceil \log(4t \cdot (2t-1)/\varepsilon) \rceil. \tag{1}$$

Note that if $n \geq 2\lambda+3$, then we have $0 \leq n' \leq n$ as we show in Lemma 2 below.

The value n' uniquely determines an index set $\mathcal{I} = \{i_1, \ldots, i_\omega\} \subseteq \{0, \ldots, \ell\}$ such that $n' = \sum_{i \in \mathcal{I}} 2^i$, where $\ell := \lfloor \log n \rfloor$. The key point in defining n' as in Equation (1) is that it provides the following two properties simultaneously:

Guessing from a polynomially-bounded range. In order to enable a reduction from adaptive to selective security, we will later have to "predict" a certain hash values $H(x^*)$. Think of x^* as the challenge identity in an IB-KEM security experiment, or the message from the forgery in a signature security experiment. Blockwise partitioning enables this as follows.
Consider the following probabilistic algorithm BPSmp, which takes as input λ, t, and ε, computes n' as in Equation (1), chooses $K_i \xleftarrow{\$} \{0,1\}^{2^i}$ uniformly random for $i \in \mathcal{I}$ and defines $K_i = \perp$ for all $i \notin \mathcal{I}$. Then it outputs

$$(K_0, \ldots, K_\ell) \xleftarrow{\$} \mathsf{BPSmp}(1^\lambda, t, \varepsilon).$$

The joint range of all hash functions H_i with $i \in \mathcal{I}$ is $\{0,1\}^{2^{i_1}} \times \cdots \times \{0,1\}^{2^{i_{|\mathcal{I}|}}}$, which has size

$$2^{n'} = 2^{\sum_{i \in \mathcal{I}} 2^i}.$$

Hence, we have that

$$\Pr\left[H_i(x^*) = K_i \text{ for all } i \in \mathcal{I}\right] = 2^{-n'}.$$

Note that $2^{n'}$ is *polynomially bounded*, due to the definition of n' in Equation (1).

Upper bound on the collision probability. In Lemma 2 below we will show that near-collision resistance of H guarantees that the probability that an adversary running in time t outputs any two values $x \neq x'$ such that

$$H_i(x) = H_i(x') \qquad \text{for all } i \in \mathcal{I} \tag{2}$$

is at most $\varepsilon/2$. Think of x and x' as values chosen adaptively by an adversary in a security experiment. In the context of IB-KEMs this would be chosen identities, in context of digital signatures chosen messages, for instance. Note that we do not argue that there is a *negligible* collision probability. This is not possible, because we consider a polynomially-bounded space, where an adversary will always be able to find collisions with non-negligible probability. However, we can guarantee that there will be no collision with probability at least $\varepsilon/2$. This means that an adversary that runs in some time t and has some advantage ε will sufficiently often be successful *without* finding a collision.

Hence, similar to confined guessing [12, 13] and truncation collision resistance [37], blockwise partitioning enables us to guess challenge identities from a *polynomially bounded* space. At the same time, it ensures that the space is large enough such that collisions are sufficiently unlikely, such that any adversary breaking a considered cryptosystem with some advantage ε must "sufficiently often" be successful without finding a collision.

Blockwise partitioning via weak near-collision resistance. We will now give a formal definition of weak near-collision resistance and then provide a technical lemma, which will be useful for security proofs based on blockwise partitioning of hash function outputs. Note that weak near-collision resistance is only required for the security of our constructions and we hence only require this property in the respective theorems and not in the constructions themselves.

Definition 1 (Weak near-collision resistance). *Let $\mathcal{H} = \{H : \{0,1\}^* \to \{0,1\}^n\}$ be a family of hash functions. For $n' \in \{1, \ldots, n\}$, we say that an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ breaks the weak n'-near-collision resistance of $\mathcal{H}$, if it runs in time $t_{\mathcal{A}}$, and it holds that*

$$\Pr\left[n'\text{-}\mathsf{wNCR}^{\mathcal{H}}_{\mathcal{A}} = 1\right] \geq t_{\mathcal{A}}(t_{\mathcal{A}} - 1)/2^{n'+1},$$

$\underline{n'\text{-}\mathsf{wNCR}^{\mathcal{H}}_{\mathcal{A}}}$

$(\mathcal{J}, st) \xleftarrow{\$} \mathcal{A}_1(n')$
$H \xleftarrow{\$} \mathcal{H}$
$(X^{(1)}, \ldots, X^{(Q+1)}) \xleftarrow{\$} \mathcal{A}_2(H, st)$
If $|\mathcal{J}| = n'$ and $\exists x \neq y \in \{X^{(1)}, \ldots, X^{(Q+1)}\}$ with $H(x)[i] = H(y)[i]$ for all $i \in \mathcal{J}$:
return 1, else 0

Fig. 1. The security experiment for weak near-collision resistance, executed with a family of hash functions $\mathcal{H}$ and adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}_1$ outputs an index set $\mathcal{J} \subseteq [n]$ and $\mathcal{H} \subseteq \{h : \{0,1\}^* \to \{0,1\}^n\}$. We restrict $\mathcal{A}_1$ to only output index sets $\mathcal{J}$ with $|\mathcal{J}| = n'$. Note that $H(x)[i]$ denotes the i-th bit of $H(x)$.

where n'-wNCR *is the experiment defined in Figure 1 and the probability is over the randomness of* $\mathcal{A}$ *and choosing* H*. We say that* $\mathcal{H}$ *is* weak near-collision resistant*, if there exists no adversary* $\mathcal{A}$ *breaking the weak* n'*-near-collision resistance of* $\mathcal{H}$ *for any* $n' \in \{1, \ldots, n\}$*.*

The following lemma will be useful to apply blockwise partitioning in security proofs.

Lemma 2. *Let* $H : \{0,1\}^* \to \{0,1\}^n$ *be a hash function,* t *be a polynomial, and let* ε *be a non-negligible function such that* $\varepsilon > 0$ *and* $t/\varepsilon < 2^\lambda$ *for all* λ*. Let* $n' := \lceil \log(4t \cdot (2t-1)/\varepsilon) \rceil$ *as in Equation* (1) *and define set* $\mathcal{I}$ *such that* $n' = \sum_{i \in \mathcal{I}} 2^i$*. Let* $\mathcal{A}$ *be an algorithm that outputs* $(X^{(1)}, \ldots, X^{(Q)}, X^*)$ *and runs in time* t *and let*

$$(K_0, \ldots, K_\ell) \xleftarrow{\$} \mathsf{BPSmp}(1^\lambda, t, \varepsilon),$$

where BPSmp *is the algorithm described above. Then, we have that* $1 \leq n' \leq 2\lambda + 3$ *and the following statements hold.*

1.
2. *Let* coll *be the event that there exists* $x, x' \in \{X^{(1)}, \ldots, X^{(Q)}, X^*\}$ *such that*

$$H_i(x) = H_i(x') \text{ for all } i \in \mathcal{I}. \tag{3}$$

Let $\mathsf{badChal}$ *be the event that there exists* $i \in \mathcal{I}$ *such that* $\Pr[H_i(X^*) \neq K_i]$*. If* H *is drawn uniformly at random from a family of weak near-collision resistant hash functions in the sense of Definition 1, then we have*

$$(\varepsilon - \Pr[\mathsf{coll}]) \cdot \Pr[\neg\mathsf{badChal}] \geq \varepsilon^2/(32t^2 - 16t).$$

Moreover, coll *and* $\mathsf{badChal}$ *are independent of each other.*

3. *Let* $\mathsf{badEval}$ *be the event that there exists* $x \in \{X^{(1)}, \ldots, X^{(Q)}\}$ *with* $x \neq X^*$ *such that* $H_i(x) = K_i$ *for all* $i \in \mathcal{I}$*. Then we have*

$$\mathsf{badEval} \implies \mathsf{coll} \vee \mathsf{badChal}.$$

Proof. The proof uses the following inequalities and identities from [37, 39] and we therefore refer to [37, 39] and the full version [38] for the proof.

$$n' \in \{1, \ldots, 2\lambda + 3\}, \qquad \frac{2t(2t-1)}{2^{n'}} \leq \frac{\varepsilon}{2}, \text{ and} \qquad \frac{1}{2^{n'}} \geq \frac{\varepsilon}{16t^2 - 8t} \tag{4}$$

The statement that $1 \leq n' \leq 2\lambda + 3$ holds immediately follows from the first of the above equations. We start to prove Property 1 by showing $\Pr[\mathsf{coll}] < \varepsilon/2$. Assume an algorithm $\mathcal{A}$ running in time $t_{\mathcal{A}}$ that outputs $(X^{(1)}, \ldots, X^{(Q)}, X^*)$ such that there exist $x, x' \in \{X^{(1)}, \ldots, X^{(Q)}, X^*\}$ such that Equation (3) holds with probability at least $\varepsilon/2$. By the definition of $\mathcal{I}$ and the functions H_i, this yields that $H(x)$ and $H(x')$ agree on at least n' positions. We construct an algorithm $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ that uses $\mathcal{A}$ to break the weak n'-near-collision resistance of $\mathcal{H}$. Note that the choice of $\mathcal{I}$ is independent of $H \in \mathcal{H}$. $\mathcal{B}_1$ therefore just encodes $K = (K_0, \ldots, K_\ell)$ to $\mathcal{J} \subseteq \{1, \ldots, n\}$ with $|\mathcal{J}| = n'$. $\mathcal{B}_2$ simply relays $\mathcal{A}$'s output $(X^{(1)}, \ldots, X^{(Q)}, X^*)$. The runtime $t_{\mathcal{B}}$ of $\mathcal{B}$ is at most $2t_{\mathcal{A}}$, since $\mathcal{B}$ does nothing more than executing $\mathcal{A}$ and relaying its outputs. Therefore, we get

$$\Pr[\mathsf{coll}] > \varepsilon_{\mathcal{A}}/2 \geq \frac{2t_{\mathcal{A}}(2t_{\mathcal{A}} - 1)}{2^{n'}} \geq \frac{t_{\mathcal{B}}(t_{\mathcal{B}} - 1)}{2^{n'+1}},$$

where the second inequality follows from Equation (4). This contradicts the weak near-collision resistance of $\mathcal{H}$. Next, we determine $\Pr[\neg\mathsf{badChal}]$. We have that the events coll and $\mathsf{badChal}$ are independent of each other because $(K_0, \ldots, K_\ell)$ is chosen independently of $(X^{(1)}, \ldots, X^{(Q)}, X^*)$. Moreover, each K_i with $i \in \mathcal{I}$ is chosen uniformly at random from $\{0,1\}^{2^{2^i}}$ and thus we have

$$\Pr[\neg\mathsf{badChal}] = \Pr[H_i(X^*) = K_i \text{ for all } i \in \mathcal{I}] = \frac{1}{2^{\sum_{i \in \mathcal{I}} 2^i}} = 2^{-n'},$$

where the last equation follows by definition of n'. To prove Property 1, we then calculate

$$(\varepsilon_{\mathcal{A}} - \Pr[\mathsf{coll}])2^{-n'} \geq \left(\varepsilon_{\mathcal{A}} - \frac{\varepsilon_{\mathcal{A}}}{2}\right) \frac{\varepsilon_{\mathcal{A}}}{16t_{\mathcal{A}}^2 - 8t_{\mathcal{A}}} = \frac{\varepsilon_{\mathcal{A}}^2}{32t_{\mathcal{A}}^2 - 16t_{\mathcal{A}}},$$

where the first inequality follows from Equation (4). Finally, to show Property 2, we explain that if $\mathsf{badEval}$ occurs, then either $\mathsf{badChal}$ or coll *must occur*. This is because if there exists $x \in \{X^{(1)}, \ldots, X^{(Q)}\}$ with $x \neq X^*$ and $H_i(x) = K_i$ for all $i \in \mathcal{I}$, then we have either that also $H_i(X^*) = K_i$ for all $i \in \mathcal{I}$ and then coll occurs, or we have that there exists an index $i \in \mathcal{I}$ such that $H_i(X^*) \neq K_i$ and then $\mathsf{badChal}$ occurs. This concludes the proof.

Near-collision resistance and the non-programmable random oracle model. Near-collision resistance holds *unconditionally* in the non-programmable random oracle model [27]. Hence, all our results can also be viewed as a generic technique to obtain adaptively-secure cryptosystems in the non-programmable random oracle model without any additional assumptions. In this sense, our paper is in line with recent works that aim to avoid programmability, such as [26].

Relation to ELFs. Extremely lossy functions (ELFs), which were introduced by Zhandry in [55], are hash functions that allow the reductions to choose the hash function's image size depending on the adversary. For the adversary, the function with a small image is indistinguishable from the injective version. Blockwise partitioning uses the weak near-collision resistance of standard hash functions similarly by selecting the blocks the guess in depending on the adversaries runtime and advantage. Hence, ELFs might potentially enable constructions similar to ours. However, the known ELF construction from [55] relies on (exponential hardness of) DDH, and thus seems tied to a group based setting. Also, our approach can be seen as partially addressing the open problem from [55] of constructing ELFs based on symmetric key techniques.

Comparison to confined guessing and truncation collision resistance. Note that the index set $\mathcal{I}$ defined above may contain *multiple* indices. This is a major difference of our approach to confined guessing and truncation collision resistance, where always only *single* blocks are guessed.

The advantage of being able to guess multiple blocks is that we are now able to define n' in a much more fine-grained way, as *any* integer between 0 and n. In contrast, [12, 13] and [37] were only able to pick values n' of exponentially increasing size, such that $n' = 2^{2^j}$ for some j, which is the reason why our reductions can improve tightness and the strength of the required assumptions quadratically.

However, we cannot replace the approach of [12, 13] and [37] with blockwise partitioning in a black-box manner. Instead, we have to provide a new security analysis for cryptosystems, and show that there are reductions which are compatible with guessing multiple blocks.

3 Lattice-Based IB-KEM

We describe how blockwise partitioning can be applied in the context of lattice based cryptography, using an *Identity-Based Key-Encapsulation-Mechanism* (IB-KEM) based on LWE as example. We build our IB-KEM from Yamada's IBE [53], for which we describe how blockwise partitioning can be embedded into lattice trapdoors by describing "compatible algorithms" for blockwise partitioning in the lattice context. The notion is inspired by [53] and we use it as a generic building block to instantiate the IB-KEM. The instantiation then has ciphertexts and secret keys consisting of a constant number of matrices and vectors and public keys consisting of only $\mathcal{O}(\log(\lambda))$ many matrices and vectors. Furthermore, we are able to achieve better LWE-parameters. We provide preliminaries on lattices in the full version [38].

Definition 3. *An IB-KEM consists of the following four PPT algorithms:*

- $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ *takes as input the security parameter and outputs the public parameters* mpk *and the master secret key* msk.

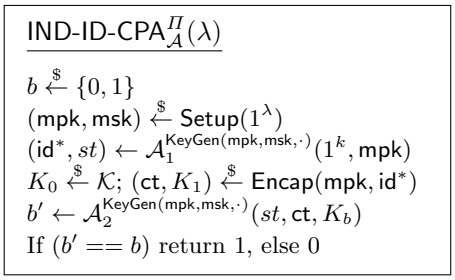

Fig. 2. The security experiment for IB-KEMs, executed with scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ and adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The oracle $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ returns $\mathsf{usk}_{\mathsf{id}} \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ with the restriction that $\mathcal{A}$ is not allowed to query oracle $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ for the target identity id^*.

- $\mathsf{usk}_{\mathsf{id}} \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ *returns the user secret key* $\mathsf{usk}_{\mathsf{id}}$ *for identity* $\mathsf{id} \in \{0,1\}^*$.
- $(\mathsf{ct}, K) \xleftarrow{\$} \mathsf{Encap}(\mathsf{mpk}, \mathsf{id})$ *returns a tuple* (ct, K)*, where* ct *is ciphertext encapsulating* K *with respect to identity* id*.*
- $K = \mathsf{Decap}(\mathsf{usk}_{\mathsf{id}}, \mathsf{ct}, \mathsf{id})$ *returns the decapsulated key* K *or an error symbol* $\perp$*.*

For correctness we require that for all $\lambda \in \mathbb{N}$*, all pairs* $(\mathsf{mpk}, \mathsf{msk})$ *generated by* $\mathsf{Setup}(1^\lambda)$*, all identities* $\mathsf{id} \in \{0,1\}^*$*, all* (K, ct) *output by* $\mathsf{Encap}(\mathsf{mpk}, \mathsf{id})$ *and all* $\mathsf{usk}_{\mathsf{id}}$ *generated by* $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$*:*

$$\Pr[\mathsf{Decap}(\mathsf{usk}_{\mathsf{id}}, \mathsf{ct}, \mathsf{id}) = K] \geq 1 - \mathsf{negl}(\lambda).$$

We use the standard IND-CPA-security notion for IB-KEMs from [8].

Definition 4. *Consider an adversary* $\mathcal{A}$ *with access (via oracle queries) to the procedures defined Figure 2. We say that* $\mathcal{A}$ *is* legitimate*, if* $\mathcal{A}$ *never queries* $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}^*)$*, where* id^* *is the output of* $\mathcal{A}_1$*. We define the advantage of* $\mathcal{A}$ *in breaking the* IND-ID-CPA *security of IB-KEM* Π *as*

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{IND\text{-}ID\text{-}CPA}^{\Pi}_{\mathcal{A}}(\lambda) = 1] - 1/2\right|$$

We include the running time of the security experiment into the running time $t_{\mathcal{A}}$ *of* $\mathcal{A}$*. This will later allow us to simplify our security analysis and the statement of theorems.*

Our construction is based on $\mathsf{dLWE}_{n,m+1,q,\alpha}$. The construction follows Yamada's construction of a lattice IBE [53] and requires "compatible algorithms" to be instantiated. We first define properties required from these compatible algorithms and then define our IB-KEM. We provide a concrete instantiation of compatible algorithms based on blockwise partitioning in Section 3.2.

Compatible Algorithms. Let $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ be the gadget matrix as introduced in [45, Theorem 1]. That is, $\mathbf{G}$ is a full rank matrix for which there is an efficient algorithm $\mathbf{G}^{-1}$ that on input $\mathbf{U} \in \mathbb{Z}_q^{n \times m}$ outputs a matrix $\mathbf{V} \in \{-1, 1\}^{m \times m}$ such that $\mathbf{GV} = \mathbf{U}$. We do not provide a formal definition of $\mathbf{G}$ due to space limitations and instead refer to [45] or the full version [38] for a formal definition. We then say that the algorithms $\mathsf{Encode}, \mathsf{PubEval}$ and $\mathsf{TrapEval}$ are *compatible with blockwise partitioning* if they combine the *vanishing trapdoors technique* from [1, 18] with blockwise partitioning. That is, that Encode encodes $(K_0, \ldots, K_\ell) \xleftarrow{\$} \mathsf{BPSmp}(1^\lambda, t(\lambda), \varepsilon(\lambda))$ into matrices $\mathbf{B}, (\mathbf{B}_i)_{0 \leq i \leq \ell}$ and trapdoors $\mathbf{R}, (\mathbf{R})_{0 \leq i \leq \ell}$
such that $\mathsf{PubEval}(H, \mathsf{id}, \mathbf{B}, (\mathbf{B}_i)_{0 \leq i \leq \ell})$ computes $\mathbf{B}_{\mathsf{id}}$ with

$$\mathbf{B}_{\mathsf{id}} = \begin{cases} \mathbf{AR}_{\mathsf{id}} + \mathbf{H}_{\mathsf{id}}\mathbf{G} & \text{if } H_i(\mathsf{id}) = K_i \text{ for all } i \in \mathcal{I} \\ \mathbf{AR}_{\mathsf{id}} & \text{otherwise,} \end{cases}$$

where $\mathbf{R}_{\mathsf{id}}$ is a matrix of small maximum norm that can be computed from the trapdoors using $\mathsf{TrapEval}$ and $\mathbf{H}_{\mathsf{id}}$ is a invertible matrix that depends on id. Note that we denote the infinity norm of a matrix $\mathbf{R}$ by $\|\mathbf{R}\|_\infty$.

Given these properties, the reduction can generate user secret keys for all identities id with $H_i(\mathsf{id}) \neq K_i$ for some $i \in \mathcal{I}$ by using a gadget trapdoor described in the full version [38]. At the same time, if id^* is such that $H_i(\mathsf{id}^*) = K_i$ for all $i \in \mathcal{I}$, then the reduction can extract a solution to its LWE instance using the adversary. By this, compatible algorithms allow us to apply blockwise partitioning in the context of lattices. We formally define these conditions as follows.

Definition 5. *We say that the algorithms* $(\mathsf{Encode}, \mathsf{PubEval}, \mathsf{TrapEval})$ *are* δ-compatible with blockwise partitioning using a family of hash functions $\mathcal{H}$*, if they are efficient and for all* $\lambda \in \mathbb{N}, t = t(\lambda) = \mathsf{poly}(\lambda)$ *and* $\varepsilon = \varepsilon(\lambda)$ *non-negligible in* λ *with* $t(\lambda)/\varepsilon(\lambda) \leq 2^\lambda$*, they satisfy the following properties:*

- *For some matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $(K_i)_{0 \leq i \leq \ell} \xleftarrow{\$} \mathsf{BPSmp}(1^\lambda, t(\lambda), \varepsilon(\lambda))$ *we have that* $\mathsf{Encode}(\mathbf{A}, (K_i)_{0 \leq i \leq \ell}) = ((\mathbf{B}, \mathbf{R}), (\mathbf{B}_i, \mathbf{R}_i)_{0 \leq i \leq \ell})$ *with* $\mathbf{B}, \mathbf{B}_i \in \mathbb{Z}_q^{n \times m}$ *and* $\mathbf{R}, \mathbf{R}_i \in \{-1, 1\}^{m \times m}$ *for all* $1 \leq i \leq \ell$*.*
- *For* $H \in \mathcal{H}$*,* $\mathsf{id} \in \{0, 1\}^*$ *and* $(\mathbf{B}, (\mathbf{B}_i)_{0 \leq i \leq \ell})$ *with* $\mathbf{B}_i \in \mathbb{Z}_q^{n \times m}$ *for all* $0 \leq i \leq \ell$ *it holds that* $\mathsf{PubEval}(H, \mathsf{id}, \mathbf{B}, (\mathbf{B}_i)_{0 \leq i \leq \ell}) = \mathbf{B}_{\mathsf{id}} \in \mathbb{Z}_q^{n \times m}$*.*
- *For* $H \in \mathcal{H}, \mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $\mathbf{R}_i \in \mathbb{Z}_q^{m \times m}$ *for all* $0 \leq i \leq \ell$*, and all* $\mathsf{id} \in \{0, 1\}^*$ *it holds that* $\mathsf{TrapEval}(H, \mathsf{id}, \mathbf{R}, (\mathbf{R}_i)_{0 \leq i \leq \ell}) = \mathbf{R}_{\mathsf{id}} \in \mathbb{Z}^{m \times m}$*.*

We require that for all $\mathsf{id} \in \{0, 1\}^*, \mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and* $H \in \mathcal{H}$ *it holds that*

$$\mathsf{PubEval}(H, \mathsf{id}, (\mathbf{B}_i)_{0 \leq i \leq \ell}) \begin{cases} \mathbf{AR}_{\mathsf{id}} & \text{if } H_i(\mathsf{id}) = K_i \text{ for all } i \in \mathcal{I} \\ \mathbf{AR}_{\mathsf{id}} + \mathbf{H}_{\mathsf{id}}\mathbf{G} & \text{otherwise} \end{cases}$$

for some invertible matrix $\mathbf{H}_{\mathsf{id}} \in \mathbb{Z}_q^{n \times n}$ *and that*

$$\|\mathbf{R}_{\mathsf{id}}\|_\infty \leq \delta,$$

where $(K_i)_{0\le i\le \ell}$ *is sampled as* $(K_i)_{0\le i\le \ell} \xleftarrow{\$} \mathsf{BPSmp}(1^\lambda, t, \varepsilon)$ *and we have that* $\mathsf{Encode}(\mathbf{A}, (K_i)_{0\le i\le \ell}) = ((\mathbf{B}, \mathbf{R}), (\mathbf{B}_i, \mathbf{R}_i)_{0\le i\le \ell})$. *Further* $\mathbf{R}_{\mathsf{id}}$ *is computed as* $\mathbf{R}_{\mathsf{id}} = \mathsf{TrapEval}(H, \mathsf{id}, \mathbf{R}, (\mathbf{R}_i)_{0\le i\le \ell})$. *Finally, we require, that for* $\mathbf{A}, \mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ *and all* $0 \le i \le \ell$ *the distributions* $(\mathbf{A}, \mathbf{A}')$ *and* $(\mathbf{A}, \mathbf{B}_i)$ *and the distributions* $(\mathbf{A}, \mathbf{A}')$ *and* $(\mathbf{A}, \mathbf{B})$ *have only negligible statistical difference in* λ.

The construction. Let $\mathcal{H} = \{H : \{0,1\}^* \to \{0,1\}^{2\lambda+3}\}$ be a family of hash functions, let $\ell = \lfloor \log(n) \rfloor$. Further, let $D_{\mathbb{Z}^m,\sigma}$ be the Gaussian distribution over $\mathbb{Z}^m$ with parameter $\sigma > 0$. Moreover, let $\mathsf{GenTrap}(1^n, 1^m, q)$ be an algorithm that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that is indistinguishable from a random matrix and a trapdoor $\mathbf{A}_{\sigma_0}^{-1}$ for $\sigma_0 = \omega(n \log q \log m)$. Note that for arbitrary $m' \ge m$, $\mathbf{u} \in \mathbb{Z}_q^n$ and $\mathbf{B} \in \mathbb{Z}_q^{n\times(m'-m)}$, the trapdoor $\mathbf{A}_{\sigma_0}^{-1}$ allows sampling vectors $\mathbf{v} \in \mathbb{Z}_q^{m'}$ from $D_{\mathbb{Z}^{m'},\sigma}$ conditioned on $[\mathbf{A} \mid \mathbf{B}]\mathbf{v} = \mathbf{u}$ for $\sigma' > \sigma_0$. We denote this as sampling from $[\mathbf{A} \mid \mathbf{B}]_\sigma^{-1}(\mathbf{u})$ and formalize it in the full version [38].

We now construct our IB-KEM scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ similar to [53] and based on LWE as follows.

Setup. $\mathsf{Setup}(1^\lambda)$ chooses parameters $n, m, q, \ell, \sigma, \alpha$ and α' as specified in Remark 6, where q is a prime. It runs $(\mathbf{A}, \mathbf{A}_{\sigma_0}^{-1}) \xleftarrow{\$} \mathsf{GenTrap}(1^n, 1^m, q)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\sigma_0 = \omega(\sqrt{n \log(q) \log(m)})$ and then samples $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$. Finally, it samples $H \xleftarrow{\$} \mathcal{H}$ and $\mathbf{B}, (\mathbf{B}_i)_{0,\le i\le \ell}, \mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m\times m}$ and then outputs

$$\mathsf{mpk} = (H, \mathbf{A}, \mathbf{B}, (\mathbf{B}_i)_{0\le i\le \ell}, \mathbf{C}, \mathbf{u}) \text{and} \qquad \mathsf{msk} := \mathbf{A}_{\sigma_0}^{-1}.$$

Key Generation. The algorithm KeyGen receives $(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ as input and computes $\mathbf{B}_{\mathsf{id}} := \mathsf{PubEval}(H, \mathsf{id}, \mathbf{B}, (\mathbf{B}_i)_{0\le i\le \ell})$ such that $\mathbf{B} \in \mathbb{Z}_q^{m\times m}$. It then computes $[\mathbf{A} \mid \mathbf{C} + \mathbf{B}_{\mathsf{id}}]_\sigma^{-1}$ from $\mathbf{A}_{\sigma_0}^{-1}$ and samples $\mathbf{e} \xleftarrow{\$} [\mathbf{A} \mid \mathbf{C} + \mathbf{B}_{\mathsf{id}}]_\sigma^{-1}(\mathbf{u})$. It then outputs $\mathsf{usk}_{\mathsf{id}} := \mathbf{e} \in \mathbb{Z}^{2m}$.

Encapsulation. The Encap algorithm receives an identity $\mathsf{id} \in \{0,1\}^*$ and mpk as input. It computes $\mathbf{B}_{\mathsf{id}} := \mathsf{PubEval}(H, \mathsf{id}, \mathbf{B}, (\mathbf{B}_i)_{0\le i\le \ell})$ such that $\mathbf{B}_{\mathsf{id}} \in \mathbb{Z}_q^{n\times m}$. It then samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, x_0 \xleftarrow{\$} D_{\mathbb{Z},\alpha q}, \mathbf{x}_1, \mathbf{x}_2 \xleftarrow{\$} D_{\mathbb{Z}^m,\alpha' q}$ and $K \xleftarrow{\$} \{0,1\}$ and computes

$$c_0 = \mathbf{s}^\mathsf{T}\mathbf{u} + x_0 + K \cdot \lceil q/2 \rceil \in \mathbb{Z}_q, \quad \mathbf{c}_1^\mathsf{T} = \mathbf{s}^\mathsf{T}[\mathbf{A} \mid \mathbf{C} + \mathbf{B}_{\mathsf{id}}] + [\mathbf{x}_1^\mathsf{T} \mid \mathbf{x}_2^\mathsf{T}] \in \mathbb{Z}_q^{2m}.$$

It then returns $(\mathsf{ct} = (c_0, \mathbf{c}_1), K)$.

Decapsulation. In order to decapsulate a ciphertext $\mathsf{ct} = (c_0, \mathbf{c}_1)$, the algorithm Decap receives the user secret key $\mathsf{usk}_{\mathsf{id}} = \mathbf{e}$ and computes $w = c_0 - \mathbf{c}_1^\mathsf{T} \cdot \mathbf{e} \in \mathbb{Z}_q$. It then returns $K := 1$ if $|w - \lceil q/2 \rceil| < \lceil q/4 \rceil$ and $K := 0$ otherwise.

Error term. We deduce the error term as Yamada in [54]. We have

$$w = c_0 - \mathbf{c}_1^\mathsf{T} \cdot \mathbf{e} = K \cdot \lceil q/2 \rceil + x_0 - \left[\mathbf{x}_1^\mathsf{T} \mid \mathbf{x}_2^\mathsf{T}\right] \cdot \mathbf{e},$$

where $x_0 - \left[\mathbf{x}_1^\mathsf{T} \mid \mathbf{x}_2^\mathsf{T}\right] \cdot \mathbf{e}$ is the error term. Assuming $\alpha' \geq \alpha$, the error term is then bounded as follows

$$\begin{aligned}\left|x_0 - \left[\mathbf{x}_1^\mathsf{T} \mid \mathbf{x}_2^\mathsf{T}\right] \mathbf{e}\right| &\leq |x_0| + \left|\left[\mathbf{x}_1^\mathsf{T} \mid \mathbf{x}_2^\mathsf{T}\right] \cdot \mathbf{e}\right| \\ &\leq |x_0| + \left\|\left[\mathbf{x}_1^\mathsf{T} \mid \mathbf{x}_2^\mathsf{T}\right]\right\|_2 \cdot \|\mathbf{e}\|_2 \\ &\leq \alpha q\sqrt{m} + (\alpha'\sqrt{2m}) \cdot \sigma\sqrt{2m} \\ &= \mathcal{O}(\alpha'\sigma m q)\end{aligned}$$

with overwhelming probability, where the first inequality follows from the triangle inequality, the second one follows from the Cauchy-Schwartz inequality, and the third follows from properties of the algorithm $\mathsf{GenTrap}$ and the fact that for $x_0 \xleftarrow{\$} D_{\mathbb{Z},\alpha q}$ it holds that $|x_0| \leq \alpha q\sqrt{m}$ with overwhelming probability. We provide formal theorems for both of these claims in the full version [38]. This then implies the correctness of the scheme.

Remark 6. We select the parameters as described by Yamada (only in the full version [54]) with the additional constraint of n to be large enough to allow for blockwise partitioning. That is, we require

- that n' as chosen in Lemma 2 is at most n, that is $n \geq 2\lambda + 3$ as explained in Section 2,
- $\ell = \lfloor \log(n) \rfloor$ in order to use blockwise partitioning.
- the error term is less than $q/5$ with overwhelming probability, that is $q > \Omega(\alpha'\sigma m q)$,
- that $\mathsf{GenTrap}$ can operate, that is $m > 6n \lceil \log q \rceil$,
- that the leftover hash lemma can be applied, meaning $m \geq (n+1)\log(q) + \omega(\log(n))$ (we provide a formal definition of the leftover hash lemma in the full version [38]),
- σ has to be large enough such that the distribution of private keys in the actual scheme and in the reduction is the same, that is $\sigma > \sigma_0 = \omega(\sqrt{n\log(q)\log(m)})$ and $\sigma > m(1+\delta)\omega(\sqrt{\log(m)})$,
- that the ReRand algorithm can operate in the reduction, that is $\alpha'/2\alpha > \sqrt{2}\cdot m(\delta+1)$ and $\alpha q > \omega(\sqrt{\log(m)})$. We formally define the ReRand algorithm and the requirements for its application in the full version [38].
- that the worst to average case reduction works, that is $\alpha q > 2\sqrt{2n}$.

To satisfy the above requirements, we set the parameters as follows:

$$\begin{aligned} &n = 2\lambda + 3, && m = \mathcal{O}(n\log(q)), && q = n^{7/2} \cdot \delta^2 \omega(\log^{7/2}(n)) \\ &\sigma = m \cdot \delta \cdot \omega(\sqrt{\log(m)}) && \alpha q = 3\sqrt{n}, && \alpha' q = 5\sqrt{n} \cdot m \cdot \delta \end{aligned}$$

Note that our compatible algorithms have $\delta = 1 + (\ell+1)m$ compared to $\delta' = m^3\mathcal{O}(\log^2(\lambda))(\mathcal{O}(\lambda)+1)$ for Yamada's compatible algorithms for the modified admissible hash function and $\delta'' = \mathsf{poly}(\lambda)$ for his partitioning based on affine functions. This allows us to use much smaller q and σ.

3.1 Security of the IB-KEM

Our construction is secure when used in conjunction with the compatible algorithms we describe below in Section 3.2 under the $\mathsf{dLWE}_{n,m+1,q,\alpha}$ assumption.

Theorem 7. *If* $\Pi := (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ *from above is instantiated with a family* $\mathcal{H}$ *of weak near-collision resistant hash functions in the sense of Definition 1, then for any legitimate attacker* $\mathcal{A}$ *that breaks the* IND-ID-CPA *security of* Π *in time* $t_{\mathcal{A}}$ *with advantage* $\varepsilon_{\mathcal{A}} := \mathsf{Adv}^{\Pi}_{\mathcal{A}}(\lambda)$*, there exists an algorithm* $\mathcal{B}$ *that, given (sufficiently close approximations of)* $t_{\mathcal{A}}$ *and* $\varepsilon_{\mathcal{A}}$*, breaks the* $\mathsf{dLWE}_{n,m+1,q,\alpha}$ *assumption in time* $t_{\mathcal{B}} \approx t_{\mathcal{A}}$ *and with*

$$\mathsf{Adv}^{\mathsf{dLWE}_{n,m+1,q,\alpha}}_{\mathcal{B}}(\lambda) \geq \varepsilon^2_{\mathcal{A}}/(32t^2_{\mathcal{A}} - 16t_{\mathcal{A}}) - \mathsf{negl}(\lambda),$$

for some negligible term negl*.*

The proof of Theorem 7 mostly follows the proof from [54]. We therefore only provide it in the full version [38] for completeness.

3.2 Compatibility of Blockwise Partitioning and Lattice IBE

In this section we describe the main technical novelty of our lattice based construction: how blockwise partitioning can be applied in the context of lattices. We first discuss how a hash function output $H_i(X)$ is encoded as a matrix using the full-rank-difference encoding from Agrawal *et al.* [1] and adapt it to our needs. We then proceed to describe compatible algorithms using this encoding that fulfill all requirements of Definition 5 and can thus be used to instantiate our IB-KEM.

Encoding identities as full rank difference matrices. In our construction, we will first hash each $\mathsf{id} \in \{0,1\}^*$ with a weak near-collision resistant hash function $H \xleftarrow{\$} \mathcal{H}$ and then encode each $H_i(\mathsf{id})$ as an invertible matrix as described by Agrawal *et al.* [1]. In the following, we define the *full rank difference encoding function* of [1] and show how it can be adopted to fit blockwise partitioning. Informally, for a binary string $a \in \{0,1\}^{2^i}$, meaning a is a potential output of H_i, we pad a with zeros to be of length n by first padding it $\sum_{j=0}^{i-1} 2^j$ zeros in the front and with $\sum_{j=i+1}^{\ell} 2^j$ zeros in the end. We then canonically interpret it as a vector in $\mathbb{Z}_q^n$ and encode it with the full-rank difference encoding of [1]. We formalize this process in the following definition.

Definition 8. *Let* $f(\mathsf{Z})$ *be an irreducible polynomial of degree* n *in* $\mathbb{Z}_q^n[\mathsf{Z}]$ *and for* $\mathbf{a} \in \mathbb{Z}_q^n$*, let* $g_a(\mathsf{Z}) := \sum_{k=0}^{n-1} a_{k+1}\mathsf{Z}^k \in \mathbb{Z}_q^n[\mathsf{Z}]$*. Then the function* $\mathsf{FRD}(\mathbf{a})$:

$\mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ *from [1] is defined as*

$$\mathsf{FRD}(\mathbf{a}) := \begin{bmatrix} \mathsf{coeffs}(g_{\mathbf{a}} \bmod f) \\ \mathsf{coeffs}(\mathsf{Z} \cdot g_{\mathbf{a}} \bmod f) \\ \mathsf{coeffs}(\mathsf{Z}^2 \cdot g_{\mathbf{a}} \bmod f) \\ \vdots \\ \mathsf{coeffs}(\mathsf{Z}^{n-1} \cdot g_{\mathbf{a}} \bmod f) \end{bmatrix} \in \mathbb{Z}_q^{n\times n},$$

where coeffs *denotes the coefficients of a polynomial in* $\mathbb{Z}_q^n[\mathsf{Z}]$*. For all* $0 \le i \le \ell$ *we define* $\mathsf{FRD}_i : \{0,1\}^{2^i} \to \mathbb{Z}_q^{n\times n}$ *to be the function that behaves as follows.*

1. *For an input* $(a_1, \ldots, a_{2^i}) \in \{0,1\}^{2^i}$*,* FRD_i *lets* $\mathsf{offset}_i := \sum_{j=0}^{i-1} 2^j$ *and sets* $\mathbf{b}^T := [b_1, \ldots, b_n] \in \mathbb{Z}_q^n$*, where*

$$b_k := \begin{cases} a_{k-\mathsf{offset}_i} & \text{if } \mathsf{offset}_i < k \le \mathsf{offset}_i + 2^i \\ 0 & \text{otherwise} \end{cases}$$

   for all $1 \le k \le n$*.*
2. *It then outputs* $\mathsf{FRD}_i(a) := \mathsf{FRD}(\mathbf{b})$*.*

Agrawal *et al.* [1] prove some properties of FRD that immediately imply the following properties of FRD_i.

Lemma 9 (Section 5 in [1]). *Let* $\mathsf{FRD}_i : \{0,1\}^{2^i} \to \mathbb{Z}_q^{n\times n}$ *be as defined in Definition 8, then the following holds:*

1. FRD_i *is injective.*
2. *There is an additive group* $\mathbb{G} \subset \mathbb{Z}_q^{n\times n}$ *such that each* $\mathbf{H} \in \mathbb{G}\setminus\{0\}$ *is invertible and the range of* FRD_i *is a subset of* $\mathbb{G}$ *for all* $1 \le i \le \ell$*.*

We refer to [1, Section 5] for the proofs of the underlying facts used in Lemma 9. Our definition of FRD_i serves some further purposes that allows us to use it in conjunction with blockwise partitioning. We detail these properties in the following lemma.

Lemma 10. *Let* BPSmp *be as defined in Section 2 and let* $t \in \mathbb{N}, \varepsilon \in (0,1]$ *with* $t/\varepsilon < 2^\lambda$*. Then for* $(K_0, \ldots, K_\ell) \xleftarrow{\$} \mathsf{BPSmp}(1^\lambda, t, \varepsilon)$*,* $\mathcal{I} = \{i : K_i \neq \perp\} \subseteq \{0, \ldots, \ell\}$ *and* $X \in \{0,1\}^*$ *it holds that*

$$-\left(\sum_{i\in\mathcal{I}} \mathsf{FRD}_i(K_i)\right) + \left(\sum_{i\in\mathcal{I}} \mathsf{FRD}_i(H_i(X))\right) = 0 \Leftrightarrow K_i = H_i(X) \text{ for all } i \in \mathcal{I}.$$

We do not present the proof here due to space limitations. However, we present it in the full version [38]. Next, we describe the algorithms ($\mathsf{Encode}, \mathsf{PubEval}$, $\mathsf{TrapEval}$) and how they use FRD_i. Afterwards, we prove that the algorithms are compatible and can thus be used in our IB-KEM. The algorithms behave as follows:

$\mathsf{Encode}(\mathbf{A}, K_0, \ldots, K_\ell)$**:** The algorithm samples $\mathbf{R}, \mathbf{R}_i \xleftarrow{\$} \{-1,1\}^{m\times m}$ for all $0 \le i \le \ell$ and sets

$$\mathbf{B}_i := \begin{cases} \mathbf{A}\mathbf{R}_i + \mathbf{G} & \text{if } K_i \neq \perp \\ \mathbf{A}\mathbf{R}_i & \text{if } K_i = \perp \end{cases}$$

and $\mathbf{B} := \mathbf{A}\mathbf{R} - \left(\sum_{i\in\mathcal{I}} \mathsf{FRD}_i(K_i)\mathbf{G}\right)$. It then outputs the matrices $((\mathbf{B}, \mathbf{R}), (\mathbf{B}_i, \mathbf{R}_i)_{0\le i\le \ell})$.

$\mathsf{PubEval}(H, \mathsf{id}, \mathbf{B}, (\mathbf{B}_i)_{0,\le i\le \ell})$**:** The algorithm computes $\mathbf{H}_i := \mathsf{FRD}_i(H_i(\mathsf{id}))$ for all $0 \le i \le \ell$ and sets $\mathbf{B}'_i := \mathbf{B}_i\mathbf{G}^{-1}(\mathbf{H}_i\mathbf{G})$. It then outputs $\mathbf{B}_{\mathsf{id}} := \mathbf{B} + \sum_{i=0}^{\ell} \mathbf{B}'_i$.

$\mathsf{TrapEval}(H, \mathsf{id}, \mathbf{R}, (\mathbf{R}_i)_{0\le i\le \ell})$**:** The algorithm computes $\mathbf{H}_i := \mathsf{FRD}_i(H_i(\mathsf{id}))$ for all $0 \le i \le \ell$ and sets $\mathbf{R}'_i := \mathbf{R}_i\mathbf{G}^{-1}(\mathbf{H}_i\mathbf{G})$. It then outputs $\mathbf{R}_{\mathsf{id}} := \mathbf{R} + \sum_{i=0}^{\ell} \mathbf{R}'_i$.

Lemma 11. *The algorithms* $(\mathsf{Encode}, \mathsf{PubEval}, \mathsf{TrapEval})$ *above are* $\delta = 1 + (\ell+1)m$*-compatible with blockwise partitioning using the family of weak near-collision resistant hash functions* $\mathcal{H}$ *described in Section 2.*

Proof. We first observe that the algorithms described above fulfill the syntactical requirements. We next show that

$$\mathsf{PubEval}(H, \mathsf{id}, (\mathbf{B}_i)_{0\le i\le \ell}) \begin{cases} \mathbf{A}\mathbf{R}_{\mathsf{id}} & \text{if } H_i(\mathsf{id}) = K_i \text{ for all } i \in \mathcal{I} \\ \mathbf{A}\mathbf{R}_{\mathsf{id}} + \mathbf{H}_{\mathsf{id}}\mathbf{G} & \text{otherwise} \end{cases}$$

for some invertible matrix $\mathbf{H}_{\mathsf{id}} \in \mathbb{Z}_q^{n\times n}$. For $\mathbf{H}_i := \mathsf{FRD}_i(H_i(\mathsf{id}))$ and $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{H}\mathbf{G}$, where $x_i = 1$ if $i \in \mathcal{I}$ and 0 otherwise, we observe that $\mathbf{B}'_i = \mathbf{B}_i\mathbf{G}^{-1}(\mathbf{H}_i\mathbf{G}) = \mathbf{A}\mathbf{R}_i\mathbf{G}^{-1}(\mathbf{H}_i\mathbf{G}) + x_i \cdot \mathbf{H}_i\mathbf{G} = \mathbf{A}\mathbf{R}'_i + x_i\mathbf{H}_i\mathbf{G}$, where $\mathbf{R}'_i$ is as defined by $\mathsf{TrapEval}$. We then have that

$$\begin{aligned} \mathbf{B}_{\mathsf{id}} = \mathbf{B} + \sum_{i=0}^{\ell} \mathbf{B}'_i &= \mathbf{A}\mathbf{R} - \left(\sum_{i\in\mathcal{I}} \mathsf{FRD}_i(K_i)\mathbf{G}\right) + \left(\sum_{i=0}^{\ell} \mathbf{A}\mathbf{R}'_i + x_i\mathbf{H}_i\mathbf{G}\right) \\ &= \mathbf{A}\left(\mathbf{R} + \sum_{i=0}^{\ell} \mathbf{R}'_i\right) - \left(\sum_{i\in\mathcal{I}} \mathsf{FRD}_i(K_i)\mathbf{G}\right) + \left(\sum_{i\in\mathcal{I}} \mathsf{FRD}_i(H_i(\mathsf{id}))\mathbf{G}\right) \\ &= \mathbf{A}\mathbf{R}_{\mathsf{id}} - \mathbf{H}_{\mathsf{id}}\mathbf{G}, \end{aligned}$$

where $\mathbf{R}_{\mathsf{id}}$ is as in the description of $\mathsf{TrapEval}$ and $\mathbf{H}_{\mathsf{id}} = -\left(\sum_{i\in\mathcal{I}} \mathsf{FRD}_i(K_i)\right) + \left(\sum_{i\in\mathcal{I}} \mathbf{H}_i\right)$. Observe that $\mathbf{H}_{\mathsf{id}} = 0$ is equivalent to $K_i = H_i(\mathsf{id})$ for all $i \in \mathcal{I}$ by Lemma 10. Furthermore, we have by Lemma 9 that if $\mathbf{H}_{\mathsf{id}} \neq 0$, then $\mathbf{H}_{\mathsf{id}}$ is invertible. We proceed by proving the upper bound on $\|\mathbf{R}_{\mathsf{id}}\|_\infty$. First, observe $\|\mathbf{R}'_i\|_\infty = \|\mathbf{R}_i\mathbf{G}^{-1}(\mathbf{H}_iG)\|_\infty \le m$ since $\mathbf{R}_i, \mathbf{G}^{-1}(\mathbf{H}_i\mathbf{G}) \in \{-1,1\}^{m\times m}$ and therefore their product $\mathbf{R}'_i \in \mathbb{Z}_q^{m\times m}$ can not contain any element of absolute value larger than m. We then have

$$\|\mathbf{R}_{\mathsf{id}}\|_\infty = \left\|\mathbf{R} + \sum_{i=0}^{\ell} \mathbf{R}'_i\right\|_\infty \le \|\mathbf{R}\|_\infty + \sum_{i=0}^{\ell} \|\mathbf{R}'_i\|_\infty \le 1 + (\ell+1)m = \delta,$$

where the last inequality follows from $\mathbf{R} \in \{-1,1\}^{m\times m}$ and $\|\mathbf{R}_i'\|_\infty \leq m$. Finally, we have that for $\mathbf{A}, \mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ it holds that for all $0 \leq i \leq \ell$ the distributions $(\mathbf{A}, \mathbf{A}')$ and $(\mathbf{A}, \mathbf{B}_i)$ have only negligible statistical difference by the leftover hash lemma, which we formally provide in the full version [38]. The same holds for the distributions $(\mathbf{A}, \mathbf{A}')$ and $(\mathbf{A}, \mathbf{B})$.

4 IB-KEMs from Pairings

In this section, we show how to use blockwise partitioning to create two variants of the IB-KEMs of Boneh and Boyen [14] and Waters [51], respectively. In comparison to [14], we achieve adaptive security instead of selective security. Additionally, we get ciphertexts of only a *single* element. In comparison to the corresponding construction from [37], the q of the required q-type assumption is reduced *quadratically*, while the tightness of the reduction is improved *quadratically*. In comparison to [51], we have public parameters of size $O(\log \lambda)$ instead of $O(\lambda)$. The security analysis is also much simpler than in [51] or the simplified proof by Bellare and Ristenpart [6]. To our best knowledge, this is the first adaptively-secure IBE scheme where ciphertexts consist only of *two* elements of a prime order algebraic group with logarithmic-size public parameters. For a better understanding we instantiate both constructions with symmetric pairings. However, asymmetric pairings work as well as it can be seen in [37].

Definition 12 (Definition 1 from [32]). *A* Bilinear Group Generator *is a probabilistic polynomial-time algorithm* GrpGen *that takes as input a security parameter* λ *(in unary) and outputs* $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, \circ, \circ_T, e, \phi(1)) \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$ *such that the following requirements are satisfied.*

1. p *is a prime and* $\log(p) \in \Omega(\lambda)$
2. $\mathbb{G}$ *and* $\mathbb{G}_T$ *are subsets of* $\{0,1\}^*$*, defined by algorithmic descriptions of maps* $\phi : \mathbb{Z}_p \to \mathbb{G}$ *and* $\phi_T : \mathbb{Z}_p \to \mathbb{G}_T$*.*
3. $\circ$ *and* $\circ_T$ *are algorithmic descriptions of efficiently computable (in the security parameter) maps* $\circ : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ *and* $\circ_T : \mathbb{G}_T \times \mathbb{G}_T \to \mathbb{G}_T$*, such that*
   a) $(\mathbb{G}, \circ)$ *and* $(\mathbb{G}_T, \circ_T)$ *form algebraic groups,*
   b) ϕ *is a group isomorphism from* $(\mathbb{Z}_p, +)$ *to* $(\mathbb{G}, \circ)$ *and*
   c) ϕ_T *is a group isomorphism from* $(\mathbb{Z}_p, +)$ *to* $(\mathbb{G}_T, \circ_T)$*.*
4. e *is an algorithmic description of an efficiently computable (in the security parameter) bilinear map* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$*. We require that* e *is non-degenerate, that is,*

$$x \neq 0 \Rightarrow e(\phi(x), \phi(x)) \neq \phi_T(0).$$

Encoding elements of $\{0,1\}^{2\lambda+3}$ *as* $\mathbb{Z}_p$*-elements.* Furthermore, in order to simplify the notation and description of the construction and its security analysis, we assume that elements of $\{0,1\}^{2\lambda+3}$ can be injectively encoded as elements of $\mathbb{Z}_p$.

4.1 Compact IB-KEM from Decisional Bilinear Diffie-Hellman

In this section we describe a variant of the IBE scheme of Waters [51], which has public parameters of size $O(\log \lambda)$ instead of $O(\lambda)$.

The construction. Let $\mathcal{H} = \{H : \{0,1\}^* \to \{0,1\}^{2\lambda+3}\}$ be a family of hash functions, let $\ell = \lfloor \log(2\lambda+3) \rfloor$, and let GrpGen be a bilinear group generator. We construct IB-KEM scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ as follows.

Setup. Choose a group description $\mathcal{BG} \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$, a random hash function $H \xleftarrow{\$} \mathcal{H}$, random generators $[1], [h] \in \mathbb{G}$, $\ell + 2$ random group elements $[u'], [u_0], \ldots, [u_\ell]$, and $x \xleftarrow{\$} \mathbb{Z}_p$. Compute $e([1],[hx]) = [hx]_T$ The master secret key is $\mathsf{msk} := [hx]$. The public parameters are defined as

$$\mathsf{mpk} = ([1], [u'], [u_0], \ldots, [u_\ell], [hx]_T).$$

Key Generation. Let

$$[u(\mathsf{id})] := [u'] \prod_{i=0}^{\ell} [u_i]^{H_i(\mathsf{id})} = \left[u' + \sum_{i=0}^{\ell} u_i H_i(\mathsf{id}) \right]$$

To compute the private key for identity id, choose $s \xleftarrow{\$} \mathbb{Z}_p$ and compute and return

$$\mathsf{usk}_{\mathsf{id}} = ([s], [hx] \cdot [u(\mathsf{id})]^s = [hx + u(\mathsf{id})s])$$

Encapsulation. To encapsulate a key, choose $r \xleftarrow{\$} \mathbb{Z}_p$ and compute and return

$$\mathsf{ct} := ([r], [u(\mathsf{id})]^r = [u(\mathsf{id})r]) \in \mathbb{G}^2 \qquad \text{and} \qquad K := [hx]_T^r = [hxr]_T$$

Decapsulation. To recover K from a ciphertext $\mathsf{ct} = ([r], [u(\mathsf{id})r])$ and a matching user secret key $([s], [hx + u(\mathsf{id})s])$, compute and output

$$\frac{e([hx + u(\mathsf{id})s], [r])}{e([u(\mathsf{id})r], [s])} = \frac{[hxr + u(\mathsf{id})sr]_T}{[u(\mathsf{id})sr]_T} = [hxr]_T$$

Security Analysis. The security of this construction is based on the Decisional Bilinear Diffie-Hellman assumption, which is the same assumption as for schemes of [14,51]. In addition, we assume that the hash function H is weak near-collision resistant.

Definition 13 (Decisional Bilinear Diffie-Hellman [14]). *The advantage of an adversary $\mathcal{A}$ in solving the* Decisional Bilinear Diffie-Hellman Problem *(*DBDH*) with respect to a Bilinear Group Generator* GrpGen *is*

$$\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{A},\mathcal{BG}}(\lambda) := \left| \Pr\left[\mathcal{A}\left([\alpha],[\beta],[\gamma],V_0\right) = 1\right] - \Pr\left[\mathcal{A}\left([\alpha],[\beta],[\gamma],V_1\right) = 1\right] \right|,$$

where $\mathcal{BG} \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda), \alpha, \beta, \gamma \xleftarrow{\$} \mathbb{Z}_p, V_0 = [\alpha\beta\gamma]_T$ and $V_1 \xleftarrow{\$} \mathbb{G}_T$. We say that the DBDH *assumption holds with respect to* GrpGen*, if $\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{A}}(\lambda)$ is negligible for every PPT $\mathcal{A}$.*

Theorem 14. *If Π is instantiated with a family $\mathcal{H}$ of weak near-collision resistant hash functions in the sense of Definition 1, then for any legitimate attacker $\mathcal{A}$ that breaks the* IND-ID-CPA *security of Π in time $t_{\mathcal{A}}$ with advantage $\varepsilon_{\mathcal{A}} := \mathsf{Adv}^{\Pi}_{\mathcal{A}}(\lambda)$, there exists an algorithm $\mathcal{B}$ that, given (sufficiently close approximations of) $t_{\mathcal{A}}$ and $\varepsilon_{\mathcal{A}}$, breaks the* DBDH *assumption in time $t_{\mathcal{B}} \approx t_{\mathcal{A}}$ and with*

$$\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{B}}(\lambda) \geq \frac{\ell}{\ell+1} \cdot \frac{\varepsilon^2_{\mathcal{A}}}{32t^2_{\mathcal{A}} - 16t_{\mathcal{A}}} - \mathsf{negl}(\lambda)$$

for some negligible term negl.

Proof. Consider the following sequence of games, where we denote with G_i the event that Game i outputs 1 and with $E_i = \Pr\left[1 \xleftarrow{\$} G_i\right] - 1/2$ the advantage of $\mathcal{A}$ in Game i.

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

$$E_0 = \Pr[\mathsf{IND\text{-}ID\text{-}CPA}^{\Pi}_{\mathcal{A}}(\lambda) = 1] - 1/2 = \varepsilon_{\mathcal{A}}$$

Game 1. In this game, we additionally run algorithm

$$(K_0, \ldots, K_\ell) \xleftarrow{\$} \mathsf{BPSmp}(1^\lambda, t_{\mathcal{A}}, \varepsilon_{\mathcal{A}})$$

at the beginning of the experiment, where algorithm BPSmp is from Lemma 2.

Furthermore, we define $\mathcal{I} := \{i : K_i \neq \perp\}$. Let $\mathcal{Q}$ be the set of all identities that the adversary queries to $\mathsf{KeyGen}(\mathsf{mpk}, msk, \cdot)$, and let $\mathcal{Q}^* := \mathcal{Q} \cup \{\mathsf{id}^*\}$, where id^* is the identity of the challenge ciphertext. We raise event coll, abort the experiment, and output a random bit, if there exists $i \in \mathcal{I}$ and $\mathsf{id}, \mathsf{id}' \in \mathcal{Q}^*$ such that $\mathsf{id} \neq \mathsf{id}'$, but $H_i(\mathsf{id}) = H_i(\mathsf{id}')$ for all $i \in \mathcal{I}$. Note that coll is defined exactly as in Lemma 2 and that we have

$$E_1 \geq E_0 - \Pr[\mathsf{coll}] = \varepsilon_{\mathcal{A}} - \Pr[\mathsf{coll}].$$

Game 2. We raise event badChal, output a random bit, and abort the game, if there exist $i \in \mathcal{I}$ such that $K_i \neq H(\mathsf{id}^*)$. Note that badChal is defined exactly as in Lemma 2 and that we have

$$E_2 = E_1 \cdot \Pr[\neg\mathsf{badChal}] = (\varepsilon_{\mathcal{A}} - \Pr[\mathsf{coll}]) \cdot \Pr[\neg\mathsf{badChal}] \geq \varepsilon^2_{\mathcal{A}}/(32t^2_{\mathcal{A}} - 16t_{\mathcal{A}})$$

where the last inequality is from Property 1 of Lemma 2.

Game 3. This game deviates from the security proofs of other constructions in this paper. We need to deal with an event dlog, which is defined below, in order to apply blockwise partitioning to the Boneh-Boyen/Waters scheme.

First, we modify the way how the experiment samples the group elements that determine the function $[u(\mathsf{id})]$. The experiment first chooses a generator

$[\alpha] \xleftarrow{\$} \mathbb{G}$ and $r', r_1, \ldots, r_\ell \xleftarrow{\$} \mathbb{Z}_p$ uniformly random. Then it sets

$$[u_i] := \begin{cases} [\alpha r_i] & \text{if } K_i \neq \bot \\ [r_i] & \text{if } K_i = \bot \end{cases} \qquad \text{and} \qquad [u'] := \left[r' - \sum_{i \in \mathcal{I}} \alpha r_i K_i \right] \tag{5}$$

Note that the distribution of these values is still uniform, and therefore identical to Game 2. Game 3 now raises event dlog and aborts, if there exists $\mathsf{id} \in \mathcal{Q}$ such that

$$\sum_{i \in \mathcal{I}} \alpha r_i K_i = \sum_{i \in \mathcal{I}} \alpha r_i H_i(\mathsf{id}) \iff \sum_{i \in \mathcal{I}} r_i (K_i - H_i(\mathsf{id})) = 0 \tag{6}$$

We claim that there exists an algorithm $\mathcal{B}_1$ that breaks the Decisional Bilinear Diffie-Hellman assumption with success probability $\Pr[\mathsf{dlog}] / |\mathcal{I}|$. $\mathcal{B}_1$ receives as input $(\mathcal{BG}, [r], [\beta], [\gamma], V)$. It will compute the discrete logarithm r and use this to break the DBDH assumption.[3]

$\mathcal{B}_1$ picks $j \xleftarrow{\$} \mathcal{I}$ at random and defines $[r_j] := [r]$. Then it proceeds exactly like Game 3. If dlog occurs, then Equation (6) holds. Due to Game 2 we can be certain that $K_i = H_i(\mathsf{id}^*)$ holds for all $i \in \mathcal{I}$, and due to Game 1 we know that for any $\mathsf{id} \in \mathcal{Q}$ there must be at least one $i \in \mathcal{I}$ such that $H_i(\mathsf{id}) \neq H_i(\mathsf{id}^*)$. These two together yield that for any $\mathsf{id} \in \mathcal{Q}$ there must exist at least one $i \in \mathcal{I}$ such that $H_i(\mathsf{id}) \neq K_i$.

Let id be the first identity for which Equation (6) holds. With probability at least $1/|\mathcal{I}|$ we have $j = i$. In this case $\mathcal{B}_1$ is able to compute

$$r = r_j = \frac{\sum_{i \in \mathcal{I} \setminus \{j\}} r_i (K_i - H_i(\mathsf{id}))}{H_j(\mathsf{id}) - K_j}$$

which immediately enables $\mathcal{B}_1$ to test whether $V = e([\beta], [\gamma])^r$ holds. With $|\mathcal{I}| \leq \ell$ we thus get

$$E_3 \geq E_2 - \mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{B}_1}(\lambda)/\ell$$

Reduction. Now we are able to describe our final reduction $\mathcal{B}_2$ to the Decisional Bilinear Diffie-Hellman assumption. $\mathcal{B}_2$ that receives $(\mathcal{BG}, [\alpha], [\beta], [\gamma], V)$ and simulates the $\mathsf{IND\text{-}ID\text{-}CPA}$ experiment as follows.

Setup. $\mathcal{B}_2$ defines $[x] := [\alpha]$, $[h] := [\beta]$, and uses $[\alpha]$ to compute $[u'], [u_0], \ldots, [u_\ell]$ exactly as in Game 3, Equation (5). The master public parameters are defined as

$$\mathsf{mpk} = ([1], [u'], [u_0], \ldots, [u_\ell], e([\alpha], [\beta]))$$

note that this is a correctly distributed master public key. The secret key is implicitly defined as $[\alpha\beta]$.

[3] We could alternatively reduce to the weaker discrete logarithm problem, but we will later reduce to DBDH anyway, so omitting the additional definition saves trees.

Key Generation. In the sequel let us write

$$a(\mathsf{id}) := \sum_{i \in \mathcal{I}} r_i (H_i(\mathsf{id}) - K_i) \qquad \text{and} \qquad b(\mathsf{id}) := r' + \sum_{i \notin \mathcal{I}} r_i H_i(\mathsf{id})$$

such that we have $[u(\mathsf{id})] = [\alpha a(\mathsf{id}) + b(\mathsf{id})]$.
$\mathcal{B}_2$ needs to compute a secret key of the form

$$[s], [\alpha\beta + u(\mathsf{id})s]$$

such that s is uniform over $\mathbb{Z}_p$. To this end, it picks $s' \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$[s] := [\beta]^{-1/a(\mathsf{id})} \cdot [s'],$$

which is correctly distributed and implicitly defines $s := -\beta/a(\mathsf{id}) + s'$. Then it computes

$$\begin{aligned}
[z] &:= [\beta]^{-b(\mathsf{id})/a(\mathsf{id})} \cdot [\alpha]^{a(\mathsf{id})s'} \cdot [b(\mathsf{id})s'] \\
&= [\alpha\beta - \alpha\beta - \beta b(\mathsf{id})/a(\mathsf{id}) + \alpha a(\mathsf{id})s' + b(\mathsf{id})s'] \\
&= [\alpha\beta + (\alpha a(\mathsf{id}) + b(\mathsf{id}))(-\beta/a(\mathsf{id}) + s')] \\
&= [\alpha\beta + u(\mathsf{id})s]
\end{aligned}$$

Note here that we have $a(\mathsf{id}) \neq 0$ for all $\mathsf{id} \in \mathcal{Q}$, as otherwise we raise event dlog and abort due to Game 3, Equation (6). Then it returns $([s], [z])$.

Encapsulation. Given a challenge identity id^*, $\mathcal{B}_2$ has to create a challenge ciphertext of the form

$$\mathsf{ct} := ([r], [u(\mathsf{id}^*)r])$$

$\mathcal{B}_2$ sets $[r] := [\gamma]$, where $[\gamma]$ is from the DBDH challenge. Note that we have $a(\mathsf{id}^*) = 0$, as otherwise we raise event guess and abort due to Game 2, and thus

$$[u(\mathsf{id}^*)\gamma] = [b(\mathsf{id}^*)\gamma] = [\gamma]^{b(\mathsf{id}^*)}$$

such that $\mathsf{ct}^* = ([\gamma], [\gamma]^{b(\mathsf{id}^*)})$ is a consistent ciphertext.
Finally, it sets $K^* := T$ and returns (ct^*, K^*).

Note that if $T = [\alpha\beta\gamma]_T$, then this is a correct key, since for any valid user key $([s], [hx + u(\mathsf{id}^*)s])$ for the challenge identity id^* we have

$$\frac{e([\alpha\beta + u(\mathsf{id})s], [\gamma])}{e([u(\mathsf{id})\gamma], [s])} = \frac{[\alpha\beta\gamma + u(\mathsf{id})s\gamma]_T}{[u(\mathsf{id})s\gamma]_T} = [\alpha\beta\gamma]_T$$

while if T is random, then so is K^*. Hence, $\mathcal{B}_2$ provides a perfect simulation of Game 3. It returns whatever $\mathcal{A}$ returns, and thus we have that

$$\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{B}_2}(\lambda) \geq E_3.$$

By collecting probability across all games, we get

$$\frac{\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{B}_1}(\lambda)}{\ell} + \mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{B}_2}(\lambda) \geq \frac{\varepsilon_{\mathcal{A}}^2}{32t_{\mathcal{A}}^2 - 16t_{\mathcal{A}}}.$$

4.2 IB-KEM with Short Ciphertexts

In this section, we present a new IB-KEM that is adaptively secure and where the ciphertext consists of only a *single* element. Compared to the only other construction with these properties ([37]), the q of the required q-type assumption is reduced *quadratically*, while the tightness of the reduction is improved *quadratically*, as well. Due to weak near-collision resistance, we are also able to reduce the output length of the hash function to roughly half of the output length required in [37], which reduces computational costs while guaranteeing the same level of security. **The construction.** Let $\mathcal{H} = \{H : \{0,1\}^* \to \{0,1\}^{2\lambda+3}\}$ be a family of hash functions, let $\ell = \lfloor \log(2\lambda+3) \rfloor$, and let GrpGen be a bilinear group generator. We construct the IB-KEM scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ as follows.

Setup. Choose a group description $\mathcal{BG} \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$, a random hash function $H \xleftarrow{\$} \mathcal{H}$, a random generator $[1] \in \mathbb{G}_1$, and random elements $x_0, \ldots, x_\ell \in \mathbb{Z}_p^*$. Define the master secret key msk as

$$\mathsf{msk} = (x_0, \ldots, x_\ell) \in \mathbb{Z}_p^{\ell+1}.$$

For $i \in \mathbb{N}$ and $m \in \mathbb{N}_0$ define $b_i(m)$ as the function that, on input of integer m, outputs the i-th bit of the binary representation of m. For $\mathsf{msk} = (x_0, \ldots, x_\ell)$ and $m = 0, \ldots, 2^{\ell+1} - 1$ define

$$F(\mathsf{msk}, m) := \prod_{i=0}^{\ell} x_i^{b_i(m)}. \tag{7}$$

The public parameters are defined as

$$\mathsf{mpk} = ([F(\mathsf{msk}, 0)], \ldots, [F(\mathsf{msk}, 2^{\ell+1} - 1]).$$

Key Generation. Let

$$u(\mathsf{id}) = \prod_{i=0}^{\ell} (H_i(\mathsf{id}) + x_i) \in \mathbb{Z}_p. \tag{8}$$

Then the private key for identity id is computed as $\mathsf{usk}_{\mathsf{id}} = [1/u(\mathsf{id})]$.

Encapsulation. Observe that

$$u(\mathsf{id}) = \prod_{i=0}^{\ell} (H_i(\mathsf{id}) + x_i) = d_0 + \sum_{m=1}^{2^\ell - 1} \big(d_m \prod_{i=0}^{\ell} x_i^{b_i(n)} \big),$$

where the constants d_i are efficiently computable from $H(\mathsf{id})$. Using $H(\mathsf{id})$ and mpk first $[u(\mathsf{id})]$ is computed as

$$[u(\mathsf{id})] = \left[d_0 + \sum_{m=1}^{2^\ell - 1} \big(d_m \prod_{i=0}^{\ell} x_i^{b_i(n)} \big) \right] = [d_0] \cdot \prod_{m=1}^{2^\ell - 1} [F(\mathsf{msk}, m)]^{d_m}.$$

Note that this does not require knowledge of $x_0, \ldots, x_\ell$ explicitly.
Finally, the ciphertext and key are computed as

$$(\mathsf{ct}, K) = ([u(\mathsf{id})]^r, e([1],[1])^r) \in \mathbb{G}_1 \times \mathbb{G}_T.$$

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

Decapsulation. To recover K from a ciphertext ct for identity id and a matching user secret key $[1/(u(\mathsf{id}))]$, compute and output $e(C, \mathsf{usk}_{\mathsf{id}})$.

Security Analysis. The security of this construction is based on the q-DBDHI assumption, which is the same assumption as for the scheme of [14]. In addition, we assume that the hash function H is weak near-collision resistant.

Definition 15 (q-Decision Bilinear Diffie-Hellman Inversion Assumption [14]). *For a PPT algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ in solving the* q-Decision Bilinear Diffie-Hellman Inversion Problem *(q-DBDHI) with respect to a Bilinear Group Generator* GrpGen *is*

$$\begin{aligned}\mathsf{Adv}^{q\text{-DBDHI}}_{\mathcal{A}}(\lambda) := \big|&\Pr\left[\mathcal{A}\left(\mathcal{BG}, [y], [y\alpha], [y\alpha^2], \ldots, [y\alpha^q], V_0\right) = 1\right] - \\ &\Pr\left[\mathcal{A}\left(\mathcal{BG}, [y], [y\alpha], [y\alpha^2], \ldots, [y\alpha^q], V_1\right) = 1\right]\big|,\end{aligned}$$

where $\mathcal{BG} \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda), \alpha \xleftarrow{\$} \mathbb{Z}_p^*, [y] \xleftarrow{\$} \mathbb{G}, V_0 = e([y],[y])^{1/\alpha}$ and $V_1 \xleftarrow{\$} \mathbb{G}_T$. The probability is over the randomness of $\mathcal{A}$, GrpGen and sampling $\alpha, \tilde{g}, h$ and V_1. We say that the q-DBDHI assumption holds with respect to GrpGen if $\mathsf{Adv}^{q\text{-DBDHI}}_{\mathcal{A}}(\lambda)$ is negligible for every PPT $\mathcal{A}$.

We start by defining the strength of the q-DBDHI assumption and set $q := 4\lambda + 7 + j + 2\sum_{\substack{i \in [\lfloor \log(2\lambda+3) \rfloor]_0 \\ K_i \neq \perp}} \left(2^{2^i} - 1\right)$. Using the following lemma, we immediately obtain $q \leq 4\lambda + 8 + \lfloor \log(2\lambda+3) \rfloor + 32t^2_{\mathcal{A}}/\varepsilon_{\mathcal{A}}$ because $j \leq \lfloor \log(2\lambda+3) \rfloor + 1$.

Lemma 16. *Let $\mathcal{I} = \{i : K_i \neq \perp\}$ be as above, then*

$$2 \cdot \sum_{\substack{i \in [\lfloor \log(2\lambda+3) \rfloor]_0 \\ i \in \mathcal{I}}} \left(2^{2^i} - 1\right) \leq \frac{32t^2_{\mathcal{A}}}{\varepsilon_{\mathcal{A}}}.$$

The proof of Lemma 16 consists only of simple arithmetic and we therefore provide it in the full version [38].

Theorem 17. *If Π is instantiated with a family $\mathcal{H}$ of weak near-collision resistant hash functions in the sense of Definition 1, then for any legitimate attacker $\mathcal{A}$ that breaks the* IND-ID-CPA *security of Π in time $t_{\mathcal{A}}$ with advantage $\varepsilon_{\mathcal{A}} := \mathsf{Adv}^{\Pi}_{\mathcal{A}}(\lambda)$, there exists an algorithm $\mathcal{B}$ that, given (sufficiently close approximations of) $t_{\mathcal{A}}$ and $\varepsilon_{\mathcal{A}}$, breaks the q-DBDHI assumption with $q \leq 4\lambda + 9 + \lfloor \log(2\lambda+3) \rfloor + 32t^2_{\mathcal{A}}/\varepsilon_{\mathcal{A}}$ in time $t_{\mathcal{B}} = O(32t^2_{\mathcal{A}}/\varepsilon_{\mathcal{A}})$ and with*

$$\mathsf{Adv}^{q\text{-DBDHI}}_{\mathcal{B}}(\lambda) \geq \varepsilon^2_{\mathcal{A}}/(32t^2_{\mathcal{A}} - 16t_{\mathcal{A}}) - \mathsf{negl}(\lambda),$$

for some negligible term negl.

The proof of Theorem 17 adapts the techniques from the poof of Theorem 14 to the q-DBDHI assumption. For a complete overview, we provide the proof in the full version [38].

5 Verifiable Random Functions from Pairings

In this section, we use blockwise partitioning in order to construct the first verifiable random function without random oracles that has both, short proofs and short public keys. Compared to previous VRF constructions that also achieve small proof sizes, like [40, 42], we achieve much better concrete proof sizes or much smaller public keys. We provide preliminaries to VRFs in the full version [38]. The construction of the VRF roughly follows the construction of the IB-KEM. The major difference is that including group elements in the proof of the VRF allows us to only include the group elements $[x_i]$ instead of all possible combinations $F(\mathsf{msk}, m)$, as in the IB-KEM, in the public keys. Instead of viewing the VRF as an adaptation of our previous IB-KEM in Section 4.2, it can also be viewed as an adaptation of Yamada's VRF [53] to blockwise partitioning.

The construction. Let $\mathcal{H}_\lambda = \{H : \{0,1\}^* \to \{0,1\}^{2\lambda+3}\}$ be a family of hash functions, let $\ell = \lfloor \log(2\lambda+3) \rfloor$, let GrpGen be a certified bilinear group generator, and let $\mathcal{VRF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Vfy})$ be the following algorithms.

Key generation. $\mathsf{Gen}(1^\lambda)$ chooses a group description $\mathcal{BG} \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$, a random hash function $H \xleftarrow{\$} \mathcal{H}_\lambda$, a random generator $[1] \xleftarrow{\$} \mathbb{G}^*$. Then it samples $w_i \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $W_i := [w_i]$ for all $i = 0, \ldots, \ell$. It returns

$$\mathsf{vk} := ([1], \mathcal{BG}, W_0, \ldots, W_\ell, H) \qquad \text{and} \qquad \mathsf{sk} := (w_0, \ldots, w_\ell).$$

Evaluation. $\mathsf{Eval}(\mathsf{sk}, X)$ computes for $i = 0, \ldots, \ell$

$$\Theta_i(X) := \prod_{i'=0}^{i} (w_{i'} + H_{i'}(X)).$$

If there is an index $0 \le i \le \ell$ such that $\Theta_i(X) \equiv 0 \bmod p$ it sets $Y := 1_{\mathbb{G}_T}$ and $\pi_i = 1_{\mathbb{G}}$ for all $i = 0, \ldots, \ell$. Otherwise, it computes

$$Y := e([1],[1])^{1/\Theta_\ell(X)} \qquad \text{and} \qquad \pi_i := g^{1/\Theta_i(X)}$$

for all $i = 0, \ldots, \ell$. It outputs $(Y, \pi = (\pi_0, \ldots, \pi_\ell))$.

Verification. $\mathsf{Vfy}(\mathsf{vk}, X, Y, \pi)$ checks if the following conditions are met and outputs 0 if not, otherwise it outputs 1.

1. We have that $X \in \{0,1\}^*$.
2. vk has the form $([1], \mathcal{BG}, W_0, \ldots, W_\ell, H)$ and $\mathsf{sk} = (w_0, \ldots, w_\ell)$.
3. $\mathcal{BG}$ is a certified encoding of a bilinear group: $\mathsf{GrpVfy}(1^\lambda, \mathcal{BG}) = 1$ Further, all group elements can be verified $\mathsf{GrpElemVfy}(1^\lambda, \mathcal{BG}, [1]) = 1$, $\mathsf{GrpElemVfy}(1^\lambda, \mathcal{BG}, h) = 1$, $\mathsf{GrpElemVfy}(1^\lambda, \mathcal{BG}, W_i) = 1$ and also $\mathsf{GrpElemVfy}(1^\lambda, \mathcal{BG}, \pi_i) = 1$ for all $0 \le i \le \ell$.

4. If there is an index $\leq i \leq \ell$ such that $W_i \cdot [H_i(X)] = 1_{\mathbb{G}}$, then it holds that $Y = 1_{\mathbb{G}_T}$ and $\pi_i = 1_{\mathbb{G}}$ for all $i = 0, \ldots, \ell$.
5. If we have $W_i \cdot [H_i(X)] \neq 1_{\mathbb{G}}$ for all $i = 0, \ldots, \ell$, then for all of these i it holds that $e(\pi_i, W_i \cdot [H_i(X)]) = e([1], \pi_{i-1})$.
6. It holds that $e(\pi_\ell, [1]) = Y$.

$\mathcal{VRF}$ as specified above is correct and fulfills the unique provability requirements as can be proven with standard arguments. Also note that using a hash function does not affect unique provability because the hash function deterministically maps each input to an output. Like the IB-KEM we present in Section 4.2, our VRF is based on the q-DBDHI assumption. We set $q := \log(2\lambda + 3) + 2 + 2\sum_{\substack{i \in [\lfloor \log(2\lambda+3) \rfloor]_0 \\ K_i \neq \perp}} \left(2^{2^i} - 1\right)$ which is at most $\log(2\lambda + 3) + 2 + \frac{32t - \mathcal{A}^2}{\varepsilon_{\mathcal{A}}}$ by Lemma 16.

Theorem 18. *If $\mathcal{VRF}$ is instantiated with a family $\mathcal{H} = \{H : \{0,1\}^* \to \{0,1\}^{2\lambda+3}\}$ of weak near-collision resistant hash functions from Definition 1, then for any legitimate attacker $\mathcal{A}$ that breaks the pseudorandomness of $\mathcal{VRF}$ in time $t_{\mathcal{A}}$ with advantage $\varepsilon_{\mathcal{A}} := \mathsf{Adv}^{\mathsf{RoR}}_{\mathcal{A}}(\lambda)$, there exists an algorithm $\mathcal{B}$ that, given (sufficiently close approximations of) $t_{\mathcal{A}}$ and $\varepsilon_{\mathcal{A}}$, breaks the q-DBDHI assumption with $q \leq \lfloor \log(2\lambda + 3) \rfloor + 2 + 32t^2_{\mathcal{A}}/\varepsilon_{\mathcal{A}}$ in time $t_{\mathcal{B}} = O(t^2_{\mathcal{A}}/\varepsilon_{\mathcal{A}})$ and with*

$$\mathsf{Adv}^{q\text{-}\mathsf{DBDHI}}_{\mathcal{A}}(\lambda) \geq \varepsilon^2_{\mathcal{A}}/(32t^2_{\mathcal{A}} - 16t_{\mathcal{A}}) - \mathsf{negl}(\lambda),$$

for some negligible term negl.

The proof of Theorem 18 follows the proof of Theorem 17 and we thus provide it in the full version [38].

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: EUROCRYPT 2010 3, 3.2, 3.2, 8, 3.2, 9, 3.2
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: CRYPTO 2010 1
3. Alperin-Sheriff, J.: Short signatures with short public keys from homomorphic trapdoor functions. In: PKC 2015 2
4. Apon, D., Fan, X., Liu, F.H.: Compact identity based encryption from LWE. Cryptology ePrint Archive, Report 2016/125, `http://eprint.iacr.org/2016/125` 1
5. Attrapadung, N., Hanaoka, G., Yamada, S.: A framework for identity-based encryption with almost tight security. In: ASIACRYPT 2015, Part I 1
6. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In: EUROCRYPT 2009 1, 1, 4
7. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM CCS 93 1
8. Bentahar, K., Farshim, P., Malone-Lee, J., Smart, N.P.: Generic constructions of identity-based and certificateless KEMs. Journal of Cryptology 3

9. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: EUROCRYPT 2008 1
10. Biham, E., Chen, R.: Near-collisions of SHA-0. In: CRYPTO 2004 1
11. Biham, E., Chen, R., Joux, A., Carribault, P., Lemuet, C., Jalby, W.: Collisions of SHA-0 and reduced SHA-1. In: EUROCRYPT 2005 1
12. Böhl, F., Hofheinz, D., Jager, T., Koch, J., Seo, J.H., Striecks, C.: Practical signatures from standard assumptions. In: EUROCRYPT 2013 2, 2, 2
13. Böhl, F., Hofheinz, D., Jager, T., Koch, J., Striecks, C.: Confined guessing: New signatures from standard assumptions. Journal of Cryptology 2, 2, 2
14. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: EUROCRYPT 2004 1, 1, 2, 4, 4.1, 13, 4.2, 15
15. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: CRYPTO 2004 1
16. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: CRYPTO 2001 1, 1, 2
17. Boneh, D., Montgomery, H.W., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: ACM CCS 2010 1
18. Boyen, X.: Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In: PKC 2010 1, 3
19. Boyen, X., Li, Q.: Towards tightly secure lattice short signature and id-based encryption. In: ASIACRYPT 2016, Part II 1
20. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th ACM STOC 1
21. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: EUROCRYPT 2010 1
22. Catalano, D., Fiore, D., Nizzardo, L.: Programmable hash functions go private: Constructions and applications to (homomorphic) signatures with shorter public keys. In: CRYPTO 2015, Part II 1
23. Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter ibe and signatures via asymmetric pairings. In: Pairing-Based Cryptography – Pairing 2012 1
24. Ducas, L., Micciancio, D.: Improved short lattice signatures in the standard model. In: CRYPTO 2014, Part I 2
25. Fischlin, M., Fleischhacker, N.: Limitations of the meta-reduction technique: The case of Schnorr signatures. In: EUROCRYPT 2013 1
26. Fischlin, M., Harasser, P., Janson, C.: Signatures from sequential-OR proofs. In: EUROCRYPT 2020, Part III 2
27. Fischlin, M., Lehmann, A., Ristenpart, T., Shrimpton, T., Stam, M., Tessaro, S.: Random oracles with(out) programmability. In: ASIACRYPT 2010 1, 2
28. Freire, E.S.V., Hofheinz, D., Paterson, K.G., Striecks, C.: Programmable hash functions in the multilinear setting. In: CRYPTO 2013, Part I 1
29. Gentry, C.: Practical identity-based encryption without random oracles. In: EUROCRYPT 2006 1
30. Goldreich, O.: Computational complexity - a conceptual perspective 1, 3
31. Hanaoka, G., Matsuda, T., Schuldt, J.C.N.: On the impossibility of constructing efficient key encapsulation and programmable hash functions in prime order groups. In: CRYPTO 2012 1
32. Hofheinz, D., Jager, T.: Verifiable random functions from standard assumptions. In: TCC 2016-A, Part I 1, 12
33. Hofheinz, D., Jager, T., Kiltz, E.: Short signatures from weaker assumptions. In: ASIACRYPT 2011 1

34. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: CRYPTO 2008 1
35. Hohenberger, S., Waters, B.: Constructing verifiable random functions with large input spaces. In: EUROCRYPT 2010 1
36. Jager, T.: Verifiable random functions from weaker assumptions. In: TCC 2015, Part II 1
37. Jager, T., Kurek, R.: Short digital signatures and ID-KEMs via truncation collision resistance. In: ASIACRYPT 2018, Part II 1, 1, 1, 2, 2, 2, 2, 4, 4.2
38. Jager, T., Kurek, R., Niehues, D.: Efficient adaptively-secure ib-kems and vrfs via near-collision resistance (full version of this publication). Cryptology ePrint Archive, Report 2021/160, `https://eprint.iacr.org/2021/160` 1, 1, 2, 3, 3, 3, 3, 6, 3.1, 3.2, 3.2, 4.2, 4.2, 5, 5
39. Jager, T., Niehues, D.: On the real-world instantiability of admissible hash functions and efficient verifiable random functions. In: SAC 2019 1, 1, 3, 2
40. Katsumata, S.: On the untapped potential of encoding predicates by arithmetic circuits and their applications. In: ASIACRYPT 2017, Part III 1, 5
41. Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In: ASIACRYPT 2016, Part II 1
42. Kohl, L.: Hunting and gathering - verifiable random functions from standard assumptions with short proofs. In: PKC 2019, Part II 1, 1, 5
43. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: EUROCRYPT 2012 1
44. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. 5th edn. 1
45. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: EUROCRYPT 2012 3
46. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: CRYPTO 2002 1
47. Peikert, C.: A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939, `http://eprint.iacr.org/2015/939` 1
48. Polak, I., Shamir, A.: Using random error correcting codes in near-collision attacks on generic hash-functions. In: INDOCRYPT 2014 1
49. Rosie, R.: Adaptive-secure VRFs with shorter keys from static assumptions. In: CANS 18 1
50. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: CRYPTO 2009 1
51. Waters, B.R.: Efficient identity-based encryption without random oracles. In: EUROCRYPT 2005 1, 1, 4, 4.1, 4.1
52. Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: EUROCRYPT 2016, Part II 1
53. Yamada, S.: Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In: CRYPTO 2017, Part III 1, 1, 1, 1, 1, 3, 3, 3, 5
54. Yamada, S.: Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. Cryptology ePrint Archive, Report 2017/096, `http://eprint.iacr.org/2017/096` 1, 3, 6, 3.1
55. Zhandry, M.: The magic of ELFs. In: CRYPTO 2016, Part I 1, 2
56. Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: Short signatures and IBEs with small key sizes. In: CRYPTO 2016, Part III 1