Asymptotically Compact Adaptively Secure Lattice IBEs and Verifiable Random Functions via Generalized Partitioning Techniques

Shota Yamada(✉)

National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
yamada-shota@aist.go.jp

Abstract. In this paper, we focus on the constructions of adaptively secure identity-based encryption (IBE) from lattices and verifiable random function (VRF) with large input spaces. Existing constructions of these primitives suffer from low efficiency, whereas their counterparts with weaker guarantees (IBEs with selective security and VRFs with small input spaces) are reasonably efficient. We try to fill these gaps by developing new partitioning techniques that can be performed with compact parameters and proposing new schemes based on the idea.

- We propose new lattice IBEs with poly-logarithmic master public key sizes, where we count the number of the basic matrices to measure the size. Our constructions are proven secure under the LWE assumption with polynomial approximation factors. They achieve the best asymptotic space efficiency among existing schemes that depend on the same assumption and achieve the same level of security.
- We also propose several new VRFs on bilinear groups. In our first scheme, the size of the proofs is poly-logarithmic in the security parameter, which is the smallest among all the existing schemes with similar properties. On the other hand, the verification keys are long. In our second scheme, the size of the verification keys is poly-logarithmic, which is the smallest among all the existing schemes. The size of the proofs is sub-linear, which is larger than our first scheme, but still smaller than all the previous schemes.

1 Introduction

1.1 Background

In cryptography, we define appropriate security notions for cryptographic primitives, in order to capture real world attacks. For a cryptographic scheme to be useful, it is desirable that the scheme achieves security notions as realistic as possible. However, since natural and realistic security notions are hard to achieve in general, we sometimes are only able to prove ad-hoc and unrealistic security notions. Even when proving the former is possible, it sometimes comes with the

J. Katz and H. Shacham (Eds.): CRYPTO 2017, Part III, LNCS 10403, pp. 161–193, 2017.
DOI: 10.1007/978-3-319-63697-9_6

cost of longer parameters or stronger assumptions. In this paper, we focus on two such primitives: identity-based encryption (IBE) and verifiable random function (VRF).

Identity-Based Encryption. IBE [Sha85] is a generalization of public key encryption where the public key of a user can be any arbitrary string such as an e-mail address. The first realizations of IBE are given by [SOK00, BF01] on groups equipped with bilinear maps. Since then, realizations from bilinear maps [BB04a, BB04b, Wat05, Gen06, Wat09], from quadratic residues modulo composite [Coc01, BGH07], and from lattices [GPV08, CHKP10, ABB10a, Boy10] have been proposed.

Among the existing lattice IBE schemes in the standard model, the most efficient one is in [ABB10a]. However, the scheme only satisfies selective security, where an adversary must declare at the start of the game which identity it intends to target. Although schemes with a much more realistic adaptive security (or equivalently, full security) are known [CHKP10, ABB10a, Boy10], they are not as efficient as the aforementioned selectively secure scheme. In particular, all these schemes require master public keys longer by a factor $O(\lambda)$ than the selectively secure one, where λ is the security parameter. This stands in sharp contrast to pairing-based settings, in which we have adaptively secure IBE schemes [Wat09, CLL+12, JR13] that are as efficient as selectively secure ones [BB04a], up to a small constant factor.

There have been several studies that aim at reducing the sizes of the parameters in adaptively secure lattice IBEs [Yam16, AFL16, ZCZ16, KY16]. However, current state of affairs are not satisfactory. These schemes are either based on stronger assumptions [Yam16, KY16], or require still long public parameters [Yam16, KY16, AFL16], or only achieves weaker security guarantee [ZCZ16].

Verifiable Random Function. The notion of VRF was introduced by Micali, Rabin, and Vadhan [MRV99]. A VRF $\mathsf{V}_{\mathsf{sk}}(\cdot)$ is a pseudorandom function with the additional property that it is possible to create a non-interactive and publicly verifiable proof π that a given function value Y was computed correctly as $Y = \mathsf{V}_{\mathsf{sk}}(X)$. Since the introduction of this notion, several realizations have been proposed [MRV99, Lys02, Dod03, DY05, ACF09]. All these constructions only allow a polynomially bounded input space, or do not achieve full adaptive security without complexity leveraging, or are based on an interactive complexity assumption. Following [HJ16], in the sequel, we will say that a VRF has *all the desired properties*, if it has an exponential-sized input space and a proof of full adaptive security under a non-interactive complexity assumption.

The first VRF scheme with all the desired properties was proposed by Hohenberger and Waters [HW10]. Later, constructions from weaker assumptions have been studied [BMR10, ACF14, Jag15, HJ16]. Notably, the scheme in [HJ16] is secure under the standard decisional linear assumption. On the other hand, there has not been improvement on the efficiency since [HW10]. Namely, all existing VRF schemes with all the desired properties require $O(\lambda)$ group elements both in the verification keys and proofs. This is much more inefficient

than the scheme with a polynomial-size input space [DY05], which only requires $O(1)$ group elements for both.

The Gaps in Efficiency. As we have seen, there is a distinct gap in efficiency between the state of the art schemes and the desired schemes. Namely, both in lattice IBEs and VRFs, we loose efficiency when we want to achieve stronger security notions. This loss in efficiency is an artifact of the security proofs. Most of the schemes use the partitioning technique based on (an analogue of) Waters' hash [Wat05] or admissible hash functions [BB04b] to achieve adaptive security. However, these techniques typically require long parameters. The powerful framework of dual system encryption methodology, which was introduced by Waters [Wat09], does not seem to be applicable for these settings. In particular, we do not have lattice analogue of the dual system approach yet. Furthermore, the uniqueness property required for VRF seems to contradict the algebraic structure required to apply the dual system approach, as pointed out in [Jag15,HJ16].

1.2 Our Contributions

In this paper, we try to fill the above gaps by generalizing the partitioning technique and proposing new schemes with improved (asymptotic) efficiency. To do so, we first introduce the notion of *partitioning functions*, which can be thought of as a generalization of the standard admissible hash functions [BB04b,CHKP10,FHPS13,Jag15]. The notion of partitioning functions abstracts out the information theoretic properties that are required to perform the partitioning technique in the security proofs for IBE and VRF. Then, we propose two new partitioning functions that can be constructed by much more compact parameters than prior admissible hash functions. Our first construction is obtained by compressing the expression of the existing admissible hash functions by introducing a novel encoding technique, whereas the second construction is based on affine functions over a random modulus. We call the first partitioning function $\mathsf{F}_{\mathsf{MAH}}$ and the second $\mathsf{F}_{\mathsf{AFF}}$, where MAH and AFF stand for modified admissible hash function and affine function respectively. These functions provide us a framework to perform the security proofs in a more space efficient manner than previous ones.

One thing to note is that in order to use them to construct IBE and VRF schemes, we need a certain level of homomorphic capability on the underlying algebraic structures. In the lattice setting, we can implement the idea by carefully applying the powerful fully key homomorphic techniques of [BGG+14,GV15]. On the other hand, in the bilinear group setting, this technique may be inapplicable since we only have very limited amount of homomorphic capabilities. Namely, given group elements, which can be seen as encodings of the corresponding discrete logarithms, we can only compute encodings corresponding to quadratic multi-variate polynomials on them. However, in the special case of VRF, since the evaluator has full access to the secret key, it can evaluate any homomorphism on them to compute the function value. Based on this observation, we can implement the idea in this setting as well.

Table 1. Comparison of adaptively secure lattice IBE schemes

Schemes	$\lvert\mathsf{mpk}\rvert$ # of $\mathbb{Z}_q^{n\times m}$ mat.	$\lvert\mathsf{ct}\rvert$, $\lvert\mathsf{sk}\rvert$ # of $\mathbb{Z}_q^m$ vec.	LWE param $1/\alpha$	Reduction cost	Remarks
[CHKP10]	$O(\lambda)$	$O(\lambda)$	$\tilde{O}(n^{1.5})$	$O(\epsilon^{\nu+1}/Q^{\nu})^{\mathrm{b}}$	
[ABB10a]+[Boy10]	$O(\lambda)$	$O(1)$	$\tilde{O}(n^{5.5})$	$O(\epsilon^2/qQ)$	
[Yam16]	$O(\lambda^{1/\mu})^{\mathrm{a}}$	$O(1)$	$n^{\omega(1)}$	$O(\epsilon^{\mu+1}/kQ^{\mu})^{\mathrm{a}}$	
[ZCZ16]	$O(\log Q)$	$O(1)$	$\tilde{O}(Q^2\cdot n^{6.5})$	$O(\epsilon/kQ^2)$	Q-bounded
[AFL16][c]	$O(\lambda/\log^2\lambda)$	$O(1)$	$\tilde{O}(n^6)$	$O(\epsilon^2/qQ)$	
[BL16]	$O(\lambda)$	$O(1)$	$\mathsf{superpoly}(n)$	$O(\lambda)$	
[KY16][d]	$O(\lambda^{1/\mu})^{\mathrm{a,d}}$	$O(1)$	$O(n^{2.5+2\mu})^{\mathrm{a}}$	$O((\lambda^{\mu-1}\epsilon^{\mu}/Q^{\mu})^{\mu+1})^{\mathrm{a}}$	Ring-based
Sect. 5.2 + $\mathsf{F}_{\mathsf{MAH}}$	$O(\log^3\lambda)$	$O(1)$	$\tilde{O}(n^{11})$	$O(\epsilon^{\nu+1}/Q^{\nu})^{\mathrm{b}}$	
Sect. 5.2 + $\mathsf{F}_{\mathsf{AFF}}$[e]	$O(\log^2\lambda)$	$O(1)$	$\mathsf{poly}(n)$	$O(\epsilon^2/k^2Q)$	Need [BCH86, Bar89]

We compare with adaptively secure IBE schemes under the LWE assumption in the standard model. $\lvert\mathsf{mpk}\rvert$, $\lvert\mathsf{ct}\rvert$, and $\lvert\mathsf{sk}_{\mathsf{ID}}\rvert$ show the size of the master public keys, ciphertexts, and private keys, respectively. For both our schemes, we set $\eta = \log^2\lambda$. To measure the space efficiency, we count the number of basic components. Q and ϵ denote the number of key extraction queries and the advantage, respectively. $\mathsf{poly}(n)$ (resp. $\mathsf{superpoly}(n)$) represents fixed but large polynomial (super-polynomial) that does not depend Q and ϵ. To measure the reduction cost, we show the advantage of the LWE algorithm constructed from the adversary against the corresponding IBE scheme. To be fair, we calculate the reduction cost by employing the technique of Bellare and Ristenpart [BR09] for all schemes.

[a] $\mu \in \mathbb{N}$ is a constant number that can be chosen arbitrary. Since the reduction cost degrades exponentially as μ grows, we would typically set μ very small (e.g., $\mu = 2$ or 3).

[b] $\nu > 1$ is the constant satisfying $c = 1 - 2^{-1/\nu}$, where c is the relative distance of the underlying error correcting code $C : \{0,1\}^k \to \{0,1\}^{\ell}$. We can take ν as close to 1 as one wants, by choosing $c < 1/2$ appropriately and make ℓ large enough (See Appendix E.1 of [Gol08]).

[c] They also propose a variant of the scheme with constant-size master public key assuming the exponentially secure collision resistant hash function. Since the use of the exponential assumption can be considered as a certain kind of the complexity leveraging, we do not include the variant in the table.

[d] The scheme can only be instantiated over the rings $R_q = \mathbb{Z}_q[X]/(X^n+1)$. To measure the size of mpk we count the number of the basic vectors, instead of the basic matrices.

[e] The key generation and encryption algorithm of the scheme involves the heavy step of computing the description of the division circuit in NC^1 using the result of [BCH86] and converting it into a branching program by invoking the Barrington's theorem [Bar89].

New Lattice IBE Schemes. Based on the new partitioning functions, we propose two new adaptively secure lattice IBE schemes. For the overview and comparison, we refer to Table 1. Both our schemes achieve the best asymptotic space efficiency among existing schemes with the same assumption and security notion. In particular, the number of basic matrices in the master public keys are only polylogarithmic. Furthermore, the sizes of the ciphertexts and private keys are optimal, in the sense that they match those of the selectively secure schemes [ABB10a, Boy10] up to a constant factor.

– In our first scheme, the master public key consists of $\omega(\log^2\lambda)$ basic matrices[1], which is the smallest among all the previous schemes. The security of the scheme can be shown from the LWE assumption with approximation factor $\tilde{O}(n^{11})$, where n is the dimension of the lattices.

[1] In our paper, when we say that the size of a parameters is $\omega(f(\lambda))$, it means that the parameter can be set to be *any* (polynomially bounded) function that grows faster than $f(\lambda)$. The parameter can be as small as one wants, as long as it does not violate the lower-bound given by the ω-notation. In this case, we can choose the number of the matrices to be $\Theta(\log^3\lambda)$ or even $\Theta(\log^2\lambda\cdot\log\log\log\lambda)$ for instance.

– In our second scheme, the master public key consists of only $\omega(\log \lambda)$ basic matrices, which is even smaller than the one above. The security of the scheme can be shown from the LWE assumption with approximation factors $\mathsf{poly}(n)$, where $\mathsf{poly}(n)$ is some fixed polynomial that is determined by the depth of the circuit computing a certain function.

We constructed the above schemes in a modular way. We first define the notion of *compatible algorithms* for partitioning functions. Then, we propose a generic construction of an IBE scheme from a partitioning function with its associating compatible algorithms. We obtain our first scheme by instantiating this framework with $\mathsf{F}_{\mathsf{MAH}}$ and its compatible algorithms. We obtain our second scheme by instantiating it with $\mathsf{F}_{\mathsf{AFF}}$.

New VRF Schemes. We also obtain the following three new VRF schemes with all the desired properties. For the overview and comparison, we refer to Table 2. All our schemes are constructed on bilinear groups and proven secure under the L-DDH assumption,[2] as is the same as most of the previous schemes [ACF14, BMR10, Jag15]. In the following, to measure the sizes of the proofs and verification keys, we count the number of group elements. Note that in all existing VRF schemes with all the desired properties [HW10, ACF14, BMR10, Jag15, HJ16], the sizes of the verification keys and proofs are $O(\lambda)$.

– Our first scheme is based on $\mathsf{F}_{\mathsf{MAH}}$, and is parametrized by several parameters, which control the tradeoffs of the efficiency. In certain parameter settings, the scheme achieves the smallest proof-size among all existing VRF schemes that satisfy all the desired properties. The size of the proofs is $\omega(\log \lambda)$, whereas the size of the verification keys is $\omega(\lambda \log \lambda)$. The security is proven from the L-DDH assumption with $L = \tilde{O}(\lambda)$.
– Our second scheme is obtained by setting the parameters appropriately in our first scheme and modifying it slightly. The scheme achieves the smallest verification-key-size among all existing schemes with all the desired properties. The size of the verification keys is $\omega(\log \lambda)$, whereas the size of the proofs is $\omega(\sqrt{\lambda} \log \lambda)$. The size of the proofs is larger than our first scheme, but still smaller than all the previous schemes. The security is proven from the L-DDH assumption with $L = \tilde{O}(\lambda)$.
– Our third scheme is based on $\mathsf{F}_{\mathsf{AFF}}$. The size of the verification keys and the proofs are $\omega(\log \lambda)$ and $\mathsf{poly}(\lambda)$, respectively. The security of the scheme is proven from the L-DDH assumption with $L = \mathsf{poly}(\lambda)$. Here, $\mathsf{poly}(\lambda)$ is some fixed polynomial that is determined by the depth of the circuit computing a certain function.

Note that the main advantage of the third scheme over our first and second schemes is that the security reduction is tighter.

Finally, we note that even though our lattice IBE schemes achieve the best asymptotic space efficiency, it might not outperform [ABB10a, Boy10] in practical parameter settings, due to the large poly-logarithmic factors and the heavy

[2] The L-DDH assumption says that given elements g, h, $g^{\alpha}, \ldots, g^{\alpha^L}$ in a bilinear group, $e(g,h)^{1/\alpha}$ is pseudorandom for any PPT adversary.

Table 2. Comparison of VRF schemes with all the desired properties

Schemes	$\lvert\mathsf{vk}\rvert$ (# of $\mathbb{G}$)	$\lvert\pi\rvert$ (# of $\mathbb{G}$)	Assumption	Reduction cost
[ACF14]	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$-DDH	$O(\epsilon^{\nu+1}/Q^{\nu})^{\mathrm{a}}$
[BMR10]	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$-DDH	$O(\epsilon/\lambda)$
[HW10]	$O(\lambda)$	$O(\lambda)$	$O(Q\lambda/\epsilon)$-DDHE	$O(\epsilon^2/\lambda Q)$
[Jag15]	$O(\lambda)$	$O(\lambda)$	$O(\log(Q/\epsilon))$-DDH	$O(\epsilon^{\nu+1}/Q^{\nu})^{\mathrm{a}}$
[HJ16]	$O(\lambda)$	$O(\lambda)$	DLIN	$O(\epsilon^{\nu+1}/\lambda Q^{\nu})^{\mathrm{a}}$
Sect. 6.1 ($\ell_1 = \ell, \ell_2 = 1, \eta = \log^2\lambda$).	$O(\lambda \log^2\lambda)$	$O(\log^2\lambda)$	$\tilde{O}(\lambda)$-DDH	$O(\epsilon^{\nu+1}/Q^{\nu})^{\mathrm{a}}$
Sect. 6.2 ($\ell_1 = \ell_2 = \sqrt{\ell}, \eta = \log^2\lambda$)	$O(\log^2\lambda)$	$O(\sqrt{\lambda}\log^2\lambda)$	$\tilde{O}(\lambda)$-DDH	$O(\epsilon^{\nu+1}/Q^{\nu})^{\mathrm{a}}$
App. C of the full version	$O(\log^2\lambda)$	$\mathsf{poly}(\lambda)$	$\mathsf{poly}(\lambda)$-DDH	$O(\epsilon^2/\lambda^2 Q)$

We compare VRF schemes with all the desired properties. $\lvert\mathsf{vk}\rvert$ and $\lvert\pi\rvert$ show the size of the verification keys and proofs, respectively. To measure $\lvert\mathsf{vk}\rvert$ and $\lvert\pi\rvert$, we count the number of group elements. Q and ϵ denote the number of evaluation queries and the advantage, respectively. $\mathsf{poly}(\lambda)$ represents fixed polynomial that does not depend Q and ϵ. To measure the reduction cost, we show the advantage of the algorithm that solves the problem (which is L-DDH for some L except for [HJ16]) constructed from the adversary against the corresponding VRF scheme. To be fair, we measure the reduction cost by employing the technique of Bellare and Ristenpart [BR09] for all schemes.

[a] ν is the constant satisfying $c = 1 - 2^{-1/\nu}$, where c is the relative distance of the underlying error correcting code $C : \{0,1\}^k \to \{0,1\}^\ell$. We can take ν as close to 1 as one wants, by choosing $c < 1/2$ appropriately and make ℓ large enough (See Appendix E.1 of [Gol08]).

encryption algorithm. The construction of truly efficient adaptively secure lattice IBE still remains open.

Comparison with the Dual System Encryption Methodology. The dual system encryption methodology [Wat09,LW10] is a very powerful tool to prove the adaptive security of IBE and even advanced cryptographic primitives such as attribute-based encryption [LOS+10]. However, currently, the technique is not available in several settings. These include lattice-based cryptography and the construction of VRF. We notice that relatively high level of homomorphic capabilities are available in these settings and show that the partitioning technique can be performed more compactly by exploiting this fact. Our technique is somewhat limited in the sense that it requires some homomorphic capabilities and may not be available without them. However, in the settings where our technique does not apply, the dual system encryption methodology may apply. In this sense, they have mutual complementary relationship.

1.3 Related Works

Related Works on Lattice IBE. Yamada [Yam16] used the fully key homomorphic technique of [BGG+14] and asymptotically reduced the size of the master public key. However, it required super-polynomial size modulus. The subsequent work by Katsumata et al. [KY16] showed that for the ring version of Yamada's scheme, it is possible to prove the security for polynomial-size modulus. The scheme by Apon et al. [AFL16] also proposed a scheme with shorter master public keys using a different technique. These schemes require larger number of matrices in the master public keys than ours. The scheme by Zhang et al. [ZCZ16] achieved shorter master public key size than ours, however at the

cost of a weaker security guarantee. In particular, their scheme only achieves Q-bounded security, i.e., that the security of the scheme is not guaranteed any more if the number of key extraction queries that the adversary makes exceeds Q, where Q is a parameter that must be *determined at the setup phase* of the scheme. This restriction cannot be removed by just making Q super-polynomial, since the encryption algorithm of the scheme runs in time proportional to Q. Finally, Boyen and Li [BL16] proposed the first lattice IBE schemes with tight security reductions, where the schemes require long master public keys.

Related Works on VRF. Very recently, several works showed generic constructions of VRF from simpler cryptographic primitives [GHKW17,Bit17,BGJS17]. These constructions lead to VRF schemes from various assumptions, including schemes without bilinear maps. However, they cannot be efficiently instantiated because they require general NIWI and constrained PRF (for admissible hash). On the other hand, we focus on the efficient constructions of VRF from the specific number theoretic assumption. While our results are orthogonal to theirs, our definition of partitioning function is very similar to that of the "partitioning scheme" in the independent and concurrent work by Bitansky [Bit17].

2 Technical Overview

2.1 A Twist on the Admissible Hash

We first start with the review of the adaptively secure IBE schemes that use the admissible hash function [BB04b,CHKP10]. The security proofs of these schemes are based on the partitioning technique, a proof methodology that allows to secretly partition the identity space into two sets of exponential size, the uncontrolled set and the controlled set, so that there is a noticeable probability that the adversary's key extraction queries fall in the controlled set and the challenge identity falls in the uncontrolled set. Whether the identity is controlled or uncontrolled is determined by a function $\mathsf{F}_{\mathsf{ADH}}$ that on input a secret randomness K chosen during the simulation and an identity ID outputs 0 or 1. Here, 0 (resp. 1) indicates that ID is in the uncontrolled set (resp. controlled set). Concretely, the partitioning is made by the following specific function:

$$\mathsf{F}_{\mathsf{ADH}}(K, \mathsf{ID}) = \begin{cases} 0, & \text{if } \forall i \in [\ell] : \ C(\mathsf{ID})_i = K_i \quad \vee \quad K_i = \perp \\ 1, & \text{otherwise} \end{cases}$$

where $C(\cdot)$ is a public function that maps an identity to a bit string in $\{0,1\}^\ell$ and K is a string in $\{0,1,\perp\}^\ell$. $C(\mathsf{ID})_i$ and K_i represent the i-th bit of $C(\mathsf{ID})$ and the i-th component of K, respectively. In [BB04b,CHKP10], the master public keys are sufficiently long so that we can embed the secret randomness K into them in a component-wise manner in the security proof. Since $\ell = \Theta(\lambda)$, where λ is the security parameter, this results in large master public keys containing $O(\lambda)$ basic components. Due to the similar reasons, all constructions of VRFs using admissible hash functions [ACF14,BMR10,Jag15,HJ16] also suffer from

large public parameters. Our first step to address the problem is to observe that K is very "sparse" in the sense that it conveys only a small amount of information compared to its length. In the simulation, K is chosen uniformly at random from $\{0,1,\perp\}^\ell$, with $O(\log(Q/\epsilon))$ components being not $\perp$, where Q and ϵ are the number of key extraction queries and the advantage of the adversary, respectively. Since we assume an adversary that makes polynomial number of key extraction queries and has non-negligible advantage in the security proof, we have $O(\log(Q/\epsilon)) = O(\log\lambda)$. This means that $K_i = \perp$ for most $i \in [\ell]$.

$K =$	$\perp$	$\perp$	1	$\perp$	0	$\perp$	$\perp$
	1	3	⑤	7	9	11	13
	2	4	6	8	⑩	12	14
$\mathsf{T} =$	{		5,		10,		}

$C(X)=$	0	1	1	0	0	1	0
	1	③	⑤	7	9	⑪	13
	②	4	6	⑧	⑩	12	⑭
$\mathsf{S}(X) =$	{2,	3,	5,	8,	10,	11,	14}

Fig. 1. Pictorial explanation of the definition of S and T.

Our key idea is to encode K into a much shorter bit-string. For $K \in \{0,1,\perp\}^\ell$, let us consider a set $\mathsf{T} \subseteq \{1,2,\ldots,2\ell\}$ as

$$\mathsf{T} := \{\ 2i - K_i \mid i \in [\ell],\ K_i \neq \perp\ \}. \tag{1}$$

See Fig. 1 for the illustrative example. Since an element in $\{1,2,\ldots,2\ell\}$ can be represented by a bit-string with length $\log 2\ell = O(\log\lambda)$ and T only consists of $O(\log\lambda)$ components, T can be represented by a bit-string with length $O(\log^2\lambda)$, which is much shorter than $\ell = \Theta(\lambda)$.

In the next step, we introduce a modified admissible hash function $\mathsf{F}_{\mathsf{MAH}}$ as

$$\mathsf{F}_{\mathsf{MAH}}(\mathsf{T},\mathsf{ID}) = \begin{cases} 0, & \text{if } \mathsf{T} \subseteq \mathsf{S}(\mathsf{ID}) \\ 1, & \text{otherwise} \end{cases} \qquad \text{where} \qquad \mathsf{S}(\mathsf{ID}) = \{\ 2i - C(\mathsf{ID})_i \mid i \in [\ell]\ \}.$$

Again, see Fig. 1 for the illustrative example. For T defined as above, we have

$$\mathsf{F}_{\mathsf{ADH}}(K,\mathsf{ID}) = \mathsf{F}_{\mathsf{MAH}}(\mathsf{T},\mathsf{ID}).$$

Namely, $\mathsf{F}_{\mathsf{ADH}}$ and $\mathsf{F}_{\mathsf{MAH}}$ are essentially the same functions, but they take different forms of inputs. The former takes K as the input, whereas the latter takes T, an encoded form of K, as the input. This fact suggests the possibility of the partitioning technique based on $\mathsf{F}_{\mathsf{MAH}}$, rather than $\mathsf{F}_{\mathsf{ADH}}$. Namely, we first choose $K \in \{0,1,\perp\}^\ell$ as specified, then set T as Eq. (1). The identity space is partitioned into two sets by $\mathsf{F}_{\mathsf{MAH}}(\mathsf{T},\cdot)$, which in turn is exactly the same partitioning made by $\mathsf{F}_{\mathsf{ADH}}(K,\cdot)$. Since the simulation strategy based on the function $\mathsf{F}_{\mathsf{MAH}}$ uses a much shorter secret randomness (i.e. T) than $\mathsf{F}_{\mathsf{ADH}}$, this opens up the possibility of constructing a much more compact IBE scheme.

Even given the above idea, the constructions of our IBE and VRF are not straightforward. Although the change is only in the encoding of the secret randomness, it might be the case that the construction of the function is incompatible with the underlying algebraic structures. In particular, $\mathsf{F}_{\mathsf{MAH}}$ seems to require more homomorphic capability than $\mathsf{F}_{\mathsf{ADH}}$. Indeed, even though we know how to construct IBE from bilinear maps using $\mathsf{F}_{\mathsf{ADH}}$ [BB04b], we do *not* know how to do it for $\mathsf{F}_{\mathsf{MAH}}$. In our lattice IBE, we can realize the idea by employing the fully key homomorphic technique introduced by [BGG+14]. However, we have to be careful when applying the technique, otherwise we will end up with a super polynomial LWE as in [Yam16], which is undesirable both from the security and efficiency perspectives. For our VRF based on bilinear maps, we employ the fact that we can compute the function value by highly non-linear operations in the exponent.

2.2 Our First Lattice IBE

Our proposed IBE scheme follows the general framework for constructing a lattice IBE scheme [CHKP10, ABB10a, Yam16, ZCZ16] that associates to each identity ID the matrix $[\mathbf{A}\|\mathbf{B}_{\mathsf{ID}}] \in \mathbb{Z}_q^{n\times 2m}$. In the template construction, the main part of the ciphertext for ID contains $\mathbf{s}^\top[\mathbf{A}\|\mathbf{B}_{\mathsf{ID}}]+\mathbf{x}^\top$, where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{x}$ is a small noise term. On the other hand, a private key for ID is a short vector $\mathbf{e}$ satisfying $[\mathbf{A}\|\mathbf{B}_{\mathsf{ID}}]\mathbf{e} = \mathbf{u}$ for a random public vector $\mathbf{u}$.

We compute the matrix $\mathbf{B}_{\mathsf{ID}}$ using the fully key homomorphic technique of [BGG+14]. Informally they showed that there exist algorithms $\mathsf{PubEval}$ and $\mathsf{TrapEval}$ that satisfy

$$\mathsf{PubEval}\left(\mathsf{F}, \{\mathbf{A}\mathbf{R}_i + y_i\mathbf{G}\}_{i\in[u]}\right) = \mathbf{A}\mathbf{R}_{\mathsf{F}} + \mathsf{F}(y)\cdot\mathbf{G}$$
$$\text{where } \mathbf{R}_{\mathsf{F}} = \mathsf{TrapEval}\left(\mathsf{F}, \mathbf{A}, \{\mathbf{R}_i, y_i\}_{i\in[u]}\right).$$

Here, $\mathsf{F} : \{0,1\}^u \to \{0,1\}$ is some function, $\mathbf{R}_i$ is a matrix with small coefficients, and y_i is the i-th bit of the bit-string y. Furthermore, $\mathbf{R}_{\mathsf{F}}$ has small coefficients.

For our construction, we prepare random matrices $\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_u$ in the master public key, where $u = \omega(\log^2 \lambda)$. Then, we set

$$\mathbf{B}_{\mathsf{ID}} = \mathsf{PubEval}(\ \mathsf{F}_{\mathsf{MAH}}(\ \cdot\ , \mathsf{ID}),\ \{\mathbf{B}_i\}_{i\in[u]}\ ).$$

Here, we consider $\mathsf{F}_{\mathsf{MAH}}(\cdot, \mathsf{ID})$ as a function that takes an *binary string* representing T as an input. This is necessary to apply the result of [BGG+14] without using the super-polynomial modulus. The security of the scheme is reduced to the LWE assumption, which says that given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{w} \in \mathbb{Z}_q^m$, it is hard to distinguish whether $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_q^m$ or $\mathbf{w}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{x}'^\top$ for some noise term $\mathbf{x}'$. To prove security, we set the matrices $\{\mathbf{B}_i\}$ in the master public key as

$$\mathbf{B}_i = \mathbf{A}\mathbf{R}_i + \mathsf{T}_i \cdot \mathbf{G}$$

where $\mathbf{A}$ is from the problem instance of the LWE, $\mathbf{R}_i$ is a random matrix with small coefficients, and $\mathsf{T}_i \in \{0,1\}$ is the i-th bit of the binary representation of T.

Due to the leftover hash lemma, the master public key is correctly distributed. By the properties of $\mathsf{PubEval}$ and $\mathsf{TrapEval}$, we have

$$\begin{aligned}\mathbf{B}_{\mathsf{ID}} &= \mathbf{A}\mathbf{R}_{\mathsf{ID}} + \mathsf{F}_{\mathsf{MAH}}(\mathsf{T}, \mathsf{ID}) \cdot \mathbf{G} \\ &\text{where } \mathbf{R}_{\mathsf{ID}} = \mathsf{TrapEval}\left(\mathsf{F}_{\mathsf{MAH}}(\ \cdot\ , \mathsf{ID}), \mathbf{A}, \{\mathbf{R}_i, \mathsf{T}_i\}_{i\in[u]}\right).\end{aligned}$$

Furthermore, by the property of $\mathsf{F}_{\mathsf{MAH}}$, we have

$$\mathsf{F}_{\mathsf{MAH}}(\mathsf{T}, \mathsf{ID}^{(1)}) = \cdots = \mathsf{F}_{\mathsf{MAH}}(\mathsf{T}, \mathsf{ID}^{(Q)}) = 1 \ \wedge\ \mathsf{F}_{\mathsf{MAH}}(\mathsf{T}, \mathsf{ID}^{\star}) = 0 \tag{2}$$

with noticeable probability, where $\mathsf{ID}^{\star}$ is the challenge identity, and $\mathsf{ID}^{(1)}, \ldots, \mathsf{ID}^{(Q)}$ are identities for which the adversary has made key extraction queries. If this condition holds, the simulation will be successful. The key extraction queries for $\mathsf{ID} \in \{\mathsf{ID}^{(1)}, \ldots, \mathsf{ID}^{(Q)}\}$ can be handled by using $\mathbf{R}_{\mathsf{ID}}$ as a $\mathbf{G}$-trapdoor [MP12] for the matrix $[\mathbf{A}\|\mathbf{B}_{\mathsf{ID}}] = [\mathbf{A}\|\mathbf{A}\mathbf{R}_{\mathsf{ID}} + \mathbf{G}]$. The generation of the challenge ciphertext is also possible by computing

$$\mathbf{w}^{\top}[\mathbf{I}\|\mathbf{R}_{\mathsf{ID}^{\star}}] = \left(\mathbf{s}^{\top}\mathbf{A} + \mathbf{x}'^{\top}\right) \cdot [\mathbf{I}\|\mathbf{R}_{\mathsf{ID}^{\star}}] = \mathbf{s}^{\top}[\mathbf{A}\|\mathbf{B}_{\mathsf{ID}^{\star}}] + \underbrace{\mathbf{x}'^{\top}[\mathbf{I}\|\mathbf{R}_{\mathsf{ID}^{\star}}]}_{\text{noise term}}.$$

A subtle point here is that the noise term above is not correctly distributed. However, this problem can be resolved by the technique in [KY16].

Finally, we remark that our actual construction is different from the above in two points. First, we do not use the (general) fully key homomorphic algorithm of [BGG+14] to compute $\mathbf{B}_{\mathsf{ID}}$ and $\mathbf{R}_{\mathsf{ID}}$. If we use the algorithm in a naive way, the coefficients of $\mathbf{R}_{\mathsf{ID}}$ will become super-polynomial, which somewhat nullifies the merit of having smaller number of matrices. Instead, we show a direct algorithm to compute $\mathbf{B}_{\mathsf{ID}}$ and $\mathbf{R}_{\mathsf{ID}}$ using the technique of [GV15], such that the coefficients of $\mathbf{R}_{\mathsf{ID}}$ are polynomially bounded. The second difference is that we add a matrix $\mathbf{B}_0$ to the master public key and use the matrix $[\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{\mathsf{ID}}]$ in the encryption and the key generation, instead of $[\mathbf{A}\|\mathbf{B}_{\mathsf{ID}}]$. This change is introduced because of a subtle technical reason to make the security proof easier.

2.3 Our First VRF

Our VRF is constructed on bilinear maps and obtained by incorporating our technique with the previous inversion-based VRF schemes [DY05, BMR10]. In the scheme, we set the function as

$$\mathsf{V}_{\mathsf{sk}}(X) = e(g, h)^{1/\theta_X}, \tag{3}$$

where the value $\theta_X = \mathbb{Z}_p^*$ is deterministically computed by the input X. Let us ignore the problem of how we add the verifiability to the scheme for the time being and start with the overview of the security proof for the scheme as a (plain) PRF. The security will be proven under the L-DDH assumption, which says that given $(h, \hat{g}, \hat{g}^{\alpha}, \ldots \hat{g}^{\alpha^L}, \Psi)$, it is infeasible to distinguish whether

$\Psi \xleftarrow{\$} \mathbb{G}_T$ or $\Psi = e(\hat{g}, h)^{1/\alpha}$. As before, we sample T and partition the input space into two sets by $\mathsf{F}_{\mathsf{MAH}}$. By the property and definition of $\mathsf{F}_{\mathsf{MAH}}$, we have

$$\mathsf{T} \not\subseteq \mathsf{S}(X^{(1)}) \ \wedge \ \cdots \ \wedge \ \mathsf{T} \not\subseteq \mathsf{S}(X^{(Q)}) \ \wedge \ \mathsf{T} \subseteq \mathsf{S}(X^{\star})$$

with noticeable probability, where $X^{\star}$ is the challenge input and $X^{(1)}, \ldots, X^{(Q)}$ are the inputs for which the adversary has made evaluation queries. Our strategy to prove the security is to embed the problem instance and T into the parameters of the scheme so that we have

$$\theta_X = \mathsf{P}_X(\alpha) \qquad \text{and} \qquad g = \hat{g}^{\mathsf{Q}(\alpha)}.$$

Here, $\mathsf{P}_X(\mathsf{Z})$ is a polynomial in $\mathbb{Z}_p[\mathsf{Z}]$ that depends on X and $\mathsf{Q}(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ is some fixed polynomial. We want $\mathsf{P}_X(\mathsf{Z})$ and $\mathsf{Q}(\mathsf{Z})$ to satisfy the following property: There exist $\xi_X \in \mathbb{Z}_p^*$ and $\mathsf{R}_X(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ such that

$$\frac{\mathsf{Q}(\mathsf{Z})}{\mathsf{P}_X(\mathsf{Z})} = \begin{cases} \dfrac{\xi_X}{\mathsf{Z}} + \mathsf{R}_X(\mathsf{Z}) & \text{if} \quad \mathsf{T} \subseteq \mathsf{S}(X) \\ \mathsf{R}_X(\mathsf{Z}) & \text{if} \quad \mathsf{T} \not\subseteq \mathsf{S}(X) \end{cases}. \tag{4}$$

If the above holds, the simulation will be successful. To answer the evaluation query on input $X \in \{X^{(1)}, \ldots, X^{(Q)}\}$, we compute $e(\hat{g}^{\mathsf{R}_X(\alpha)}, h)$. This is a valid answer, since we have $\mathsf{T} \not\subseteq \mathsf{S}(X)$ and thus

$$e(\hat{g}^{\mathsf{R}_X(\alpha)}, h) = e(\hat{g}^{\mathsf{Q}(\alpha)/\mathsf{P}_X(\alpha)}, h) = e(g^{1/\mathsf{P}_X(\alpha)}, h) = e(g, h)^{1/\theta_X}.$$

To answer the challenge query, we compute $\Psi^{\xi_{X^\star}} \cdot e\left(\hat{g}^{\mathsf{R}_{X^\star}(\alpha)}, h\right)$. If $\Psi \xleftarrow{\$} \mathbb{G}_T$, it is a random element in $\mathbb{G}_T$, as desired. On the other hand, if $\Psi = e(\hat{g}, h)^{1/\alpha}$, we have

$$\Psi^{\xi_{X^\star}} \cdot e\left(\hat{g}^{\mathsf{R}_{X^\star}(\alpha)}, h\right) = e\left(\hat{g}^{\mathsf{Q}(\alpha)/\mathsf{P}_{X^\star}(\alpha)}, h\right) = e\left(g^{1/\mathsf{P}_{X^\star}(\alpha)}, h\right) = e(g, h)^{1/\theta_{X^\star}}$$

which is the correct value. Now we have to find the polynomials with the desired property (namely, Eq. (4)). Let us take $\mathsf{P}_X(\mathsf{Z})$ to be the following form:[3]

$$\mathsf{P}_X(\mathsf{Z}) = \prod_{i \in [\eta], j \in [\ell]} (\mathsf{Z} - t_i + s_j) \quad \text{where} \quad \mathsf{T} = \{t_1, \ldots, t_\eta\}, \ \mathsf{S}(X) = \{s_1, \ldots, s_\ell\}.$$

In some sense, $\mathsf{P}_X(\mathsf{Z})$ checks $(t_i \stackrel{?}{=} s_j)$ in a brute-force manner. We can see that $\mathsf{P}_X(\mathsf{Z})$ can be divided by Z exactly $|\mathsf{T} \cap \mathsf{S}(X)|$ times. Furthermore, we have $|\mathsf{T} \cap \mathsf{S}(X)| = |\mathsf{T}| = \eta \Leftrightarrow \mathsf{T} \subseteq \mathsf{S}(X)$. This motivates us to define $\mathsf{Q}(\mathsf{Z})$ as follows:

$$\mathsf{Q}(\mathsf{Z}) = \mathsf{Z}^{\eta - 1} \cdot \prod_{a \neq 0} (\mathsf{Z} + a), \tag{5}$$

where the product is taken for sufficiently many $a \neq 0$, so that the latter part of $\mathsf{Q}(\mathsf{Z})$ can be divided by any factor of $\mathsf{P}_X(\mathsf{Z})$ except for Z. It is easy to see that

[3] For simplicity, we use a polynomial that is slightly different from the actual proof.

$\mathsf{Q}(\mathsf{Z})$ can be divided by Z exactly $\eta - 1$ times. These imply that $\mathsf{Q}(\mathsf{Z})$ can be divided by $\mathsf{P}_X(\mathsf{Z})$, if and only if the multiplicity of Z in $\mathsf{P}_X(\mathsf{Z})$ is at most $\eta - 1$. This fact allows us to prove Eq. (4).

Finally, we go back and see how our actual construction works. We set the verification key as $\mathsf{vk} = (g, h, \{W_i = g^{w_i}\}_{i\in[\eta]})$ and choose θ_X as

$$\theta_X = \prod_{(i,j)\in[\eta]\times[\ell]} \underbrace{(w_i + s_j)}_{:=\theta_{i,j}} = \prod_{i\in[\eta]} \underbrace{\left(\prod_{j\in[\ell]} (w_i + s_j)\right)}_{\phi_i} \tag{6}$$

and set the function value as $\mathsf{V}_{\mathsf{sk}}(X) = e(g,h)^{1/\theta_X}$. The form of θ_X reflects the "brute-force structure" that has appeared in $\mathsf{P}_X(\mathsf{Z})$. To generate a proof for the function value, we take the "step ladder approach" [Lys02, ACF09, HW10]. Namely, we publish values of the form $g^{1/\theta_{1,1}}, g^{1/\theta_{1,1}\theta_{1,2}}, \ldots, g^{1/\theta_{1,1}\cdots\theta_{\eta,\ell}} = g^{1/\theta_X}$. The correctness of the function value can be verified by the pairing computations using these terms. While this scheme achieves very short verification key, the proofs for the function values are very long. We can make the proofs much shorter by a simple trick. We introduce additional helper components $\{g^{w_i^j}\}_{(i,j)\in[\eta]\times[\ell]}$ to the verification key. Instead of publishing the proof above, we publish $g^{1/\phi_1}, g^{1/\phi_1\phi_2}, \ldots, g^{1/\phi_1\cdots\phi_\eta} = g^{1/\theta_X}$ as a proof. Thanks to the helper components, we can verify whether the function value is correct using the proof.

2.4 Other Constructions

Partitioning with Yet Another Function. We propose another function $\mathsf{F}_{\mathsf{AFF}}$, which is also useful to perform the partitioning technique. The main advantage of the function over $\mathsf{F}_{\mathsf{MAH}}$ is that it achieves even shorter secret randomness K of length $\omega(\log\lambda)$. Here, we begin by reviewing $\mathsf{F}_{\mathsf{WAT}}$, a slight variant of the celebrated Waters' hash [Wat05], and then gradually modify it to our $\mathsf{F}_{\mathsf{AFF}}$. Let the identity space of IBE (or input space of VRF) be $\{0,1\}^k$. The function $\mathsf{F}_{\mathsf{WAT}}$ is defined as

$$\mathsf{F}_{\mathsf{WAT}}(K = (\{\alpha_i\}_{i\in[k]}, \beta), \mathsf{ID}) = \begin{cases} 0, & \text{if } (\sum_{i\in[k]} \alpha_i \mathsf{ID}_i) + \beta = 0 \\ 1, & \text{otherwise} \end{cases}$$
$$\text{where } \alpha_i, \beta \in \mathbb{Z},\ \mathsf{ID} \in \{0,1\}^k$$

Here, ID_i is the i-th bit of ID. In order for the function to be useful, we should choose the random secret K so that

$$\Pr_K\left[\mathsf{F}_{\mathsf{WAT}}(K, \mathsf{ID}^{(1)}) = 1 \ \wedge\ \cdots\ \wedge\ \mathsf{F}_{\mathsf{WAT}}(K, \mathsf{ID}^{(Q)}) = 1 \ \wedge\ \mathsf{F}_{\mathsf{WAT}}(K, \mathsf{ID}^\star) = 0\right]$$

is noticeable. By a standard analysis, one can show that it suffices to satisfy the following two requirements:

(A) $\Pr_K[\mathsf{F}_{\mathsf{WAT}}(K, \mathsf{ID}^\star) = 0]$ is noticeable.
(B) $\Pr_K[\mathsf{F}_{\mathsf{WAT}}(K, \mathsf{ID}^{(i)}) = 0 \mid \mathsf{F}_{\mathsf{WAT}}(K, \mathsf{ID}^\star) = 0]$ is sufficiently small for all $i \in [Q]$.

In order to satisfy the requirements, one way to choose is $\alpha_1, \ldots, \alpha_k \xleftarrow{\$} [1, 4Q]$ and $\beta \xleftarrow{\$} [-4kQ, 0]$. As for requirement (A), we have

$$\Pr_K[\mathsf{F}_{\mathsf{WAT}}(K, \mathsf{ID}^\star) = 0] = \Pr_{\alpha,\beta}\left[\beta = -\sum_{i\in[k]} \alpha_i \mathsf{ID}_i^\star\right] = \frac{1}{4kQ+1}$$

where the second equality follows from $-4kQ \le \sum_{i\in[k]} \alpha_i \mathsf{ID}_i^\star \le 0$. We can see that the probability is noticeable as desired. The main observation here is that since the value of each α_i is polynomially bounded and $\mathsf{ID}_i^\star \in \{0,1\}$, the total sum is also confined within the polynomially bounded range and thus can be guessed with noticeable probability. Requirement (B) can be proven by exploiting a certain kind of pairwise independence of $\mathsf{F}_{\mathsf{WAT}}(K, \cdot)$.

The problem of the above function is that it requires long secret randomness K, whose length is linear in k. As the first attempt to shorten this, we could consider a modified function $\mathsf{F}'_{\mathsf{WAT}}$ defined as

$$\mathsf{F}'_{\mathsf{WAT}}(K = (\alpha, \beta), \mathsf{ID}) = \begin{cases} 0, & \text{if } \alpha\mathsf{ID} + \beta = 0 \\ 1, & \text{otherwise} \end{cases} \quad \text{where} \quad \alpha, \beta \in \mathbb{Z}, \mathsf{ID} \in [2^k - 1]$$

where we interpret $\mathsf{ID} \in \{0,1\}^k$ as an integer in $[2^k - 1]$ by the natural bijection. While it is easy to satisfy requirement (B), we no longer know how to satisfy requirement (A) at the same time. Even if the size of α is polynomially bounded, $\alpha \cdot \mathsf{ID}$ can be very large, and we can not guess the value better than with exponentially small probability.

To resolve the problem, we further modify the function and obtain our final function $\mathsf{F}_{\mathsf{AFF}}$ defined as follows:

$$\mathsf{F}_{\mathsf{AFF}}(K = (\alpha, \beta, \rho), \mathsf{ID}) = \begin{cases} 0, & \text{if } \alpha\mathsf{ID} + \beta \equiv 0 \mod \rho \\ 1, & \text{otherwise} \end{cases}$$
$$\text{where } \alpha, \beta, \rho \in \mathbb{Z},\ \mathsf{ID} \in [2^k - 1].$$

Here, we choose ρ to be a random polynomial-size prime. Now, we can satisfy requirement (A), since we only have to guess $(\alpha \cdot \mathsf{ID} \mod \rho)$, for which there are only a polynomial number of candidates. However, making the size of ρ polynomial causes a subtle problem regarding requirement (B). Let us consider the case where an adversary makes queries such that $\mathsf{ID}^\star = \mathsf{ID}^{(1)} + \rho$. In such a case, we have $\mathsf{F}_{\mathsf{AFF}}(K, \mathsf{ID}^\star) = \mathsf{F}_{\mathsf{AFF}}(K, \mathsf{ID}^{(1)})$ and the simulation fails with probability 1, no matter how we choose α and β. Such queries can be made with noticeable probability, since ρ is polynomial-size and the adversary can guess the value with noticeable probability. However a small subtlety is that the probability does not need to be negligible in order to satisfy requirement (B).

Due to this observation, by choosing ρ randomly from a large enough domain (concretely, from $[kQ^2/\epsilon, 4kQ^2/\epsilon]$), we can make the probability of such queries being made sufficiently small, hence satisfying requirement (A) and (B).

New IBE and VRF Based on the Function. Based on the function $\mathsf{F}_{\mathsf{AFF}}$, we propose a lattice based IBE scheme and a VRF scheme on bilinear groups. To construct a lattice based IBE scheme, we follow the same template as the case of $\mathsf{F}_{\mathsf{MAH}}$ and set $\mathbf{B}_{\mathsf{ID}} = \mathsf{PubEval}(\mathsf{F}_{\mathsf{AFF}}(\cdot\,, \mathsf{ID}), \{\mathbf{B}_i\}_{i\in[u]})$. Again, if we use the fully key homomorphic algorithm of [BGG+14] naively, the scheme will require super polynomial modulus q. To avoid this, to compute $\mathbf{B}_{\mathsf{ID}}$, we first compute a description of a log-depth circuit corresponding to $\mathsf{F}_{\mathsf{AFF}}$. Such a circuit exists by the classical result of Beam, Cook, and Hoover [BCH86], who showed that the computation of division can be performed in $\mathbf{NC}^1$, since division implies modulo ρ arithmetic. Then, we convert the log-depth circuit into a branching program using the Barrington's theorem [Bar89]. Finally, we use the key homomorphic algorithm for branching programs in [GV15]. Note that similar approach was also taken in [BL16] to homomorphically evaluate a PRF. To construct a VRF based on bilinear groups, we again take advantage of the fact that $\mathsf{F}_{\mathsf{AFF}}$ can be computed by a log-depth circuit. This fact is necessary for our VRF to be proven secure under a polynomial-size assumption, since our security proof requires 2^d-DDH assumption, where d is the depth of the circuit.

3 Preliminaries

Due to the space limitation, we omit most of the proofs for the lemmas presented in the paper. They can be found in the full version [Yam17].

Notation. We denote by $[a]$ a set $\{1, 2, \ldots, a\}$ for any integer $a \in \mathbb{N}$. For a set S, $|S|$ denotes its size. We treat a vector as a column vector. If $\mathbf{A}_1$ is an $n \times m$ and $\mathbf{A}_2$ is an $n \times m'$ matrix, then $[\mathbf{A}_1 \| \mathbf{A}_2]$ denotes the $n \times (m + m')$ matrix formed by concatenating $\mathbf{A}_1$ and $\mathbf{A}_2$. We use similar notation for vectors. For a vector $\mathbf{u} \in \mathbb{Z}^n$, $\|\mathbf{u}\|$ and $\|\mathbf{u}\|_\infty$ denote its ℓ_2 and ℓ_∞ norm respectively. Similarly, for a matrix $\mathbf{R}$, $\|\mathbf{R}\|_\infty$ denotes its infinity norm. $\|\mathbf{R}\|_2$ is the operator norm of $\mathbf{R}$. Namely, $\|\mathbf{R}\|_2 := \sup_{\|\mathbf{x}\|=1} \|\mathbf{R}\mathbf{x}\|$. For a function $f(\cdot) : \mathbb{N} \to \mathbb{R}_{\geq 0}$, we say that the function is negligible when for every polynomial $g(\cdot)$ and all sufficiently large λ we have $f(\lambda) < |1/g(\lambda)|$. We say that the function is noticeable when there exists a polynomial $g(\cdot)$ such that we have $f(\lambda) \geq |1/g(\lambda)|$ for all λ.

3.1 Cryptographic Primitives

IBE and VRF. We use the standard syntax of IBE [BF01] and VRF with large input spaces [HW10]. We require standard notion of the correctness for both. For VRF, we also require unique provability. As for the security, we require adaptive anonymity for IBE and pseudorandomness for VRF. We refer to the full version for the formal definitions. These security notions are defined by games between the challenger and the adversary. In the games, we use two random variables coin

and $\widehat{\mathsf{coin}}$ in $\{0,1\}$ for defining the security. coin refers to the random value chosen by the challenger and $\widehat{\mathsf{coin}}$ refers to the guess for coin output by the adversary. We have the following general statement concerning coin and $\widehat{\mathsf{coin}}$.

Lemma 1 (Lemma 8 in [KY16], See also Lemma 28 in [ABB10a]). *Let us consider an IBE (resp. VRF) scheme and an adversary $\mathcal{A}$ that breaks the adaptively-anonymous security (resp. pseudorandomness) with advantage ϵ. Let the identity space (resp. input space) be $\mathcal{X}$ and consider a map γ that maps a sequence of elements in $\mathcal{X}$ to a value in $[0,1]$. We consider the following experiment. We first execute the security game for $\mathcal{A}$. Let $X^\star$ be the challenge identity (resp. challenge input) and $X_1,\ldots,X_Q$ be the identities (resp. inputs) for which key extraction queries (resp. evaluation queries) were made. We denote $\mathbb{X}=(X^\star,X_1,\ldots,X_Q)$. At the end of the game, we set $\mathsf{coin}'\in\{0,1\}$ as $\mathsf{coin}'=\widehat{\mathsf{coin}}$ with probability $\gamma(\mathbb{X})$ and $\mathsf{coin}'\xleftarrow{\$}\{0,1\}$ with probability $1-\gamma(\mathbb{X})$. Then, the following holds.*

$$\left|\Pr[\mathsf{coin}'=\mathsf{coin}]-\frac{1}{2}\right|\geq\gamma_{\min}\cdot\epsilon-\frac{\gamma_{\max}-\gamma_{\min}}{2}$$

where $\gamma_{\min}$ and $\gamma_{\max}$ are the maximum and the minimum of $\gamma(\mathbb{X})$ taken over all possible $\mathbb{X}$, respectively.

Though the lemma was proven only for IBE in [KY16], the same proof works also for VRF.

3.2 Preliminaries on Lattices and Bilinear Maps

For an integer $m>0$, let $D_{\mathbb{Z}^m,\sigma}$ be the discrete Gaussian distribution over $\mathbb{Z}^m$ with parameter $\sigma>0$.

Learning with Errors (LWE) Assumption. We define the learning with errors (LWE) problem, which was introduced by Regev [Reg05].

Definition 1 (LWE). *For an integers $n=n(\lambda)$, $m=m(n)$, a prime integer $q=q(n)>2$, a real number $\alpha\in(0,1)$, and a PPT algorithm $\mathcal{A}$, an advantage for the learning with errors problem $\mathsf{dLWE}_{n,m,q,\alpha}$ of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{dLWE}_{n,m,q,\alpha}}=\left|\Pr\left[\mathcal{A}(\mathbf{A},\mathbf{s}^\top\mathbf{A}+\mathbf{x}^\top)\to 1\right]-\Pr\left[\mathcal{A}(\mathbf{A},\mathbf{w}^\top+\mathbf{x}^\top)\to 1\right]\right|$$

where $\mathbf{A}\xleftarrow{\$}\mathbb{Z}_q^{n\times m}$, $\mathbf{s}\xleftarrow{\$}\mathbb{Z}_q^n$, $\mathbf{x}\xleftarrow{\$}D_{\mathbb{Z}^m,\alpha q}$, $\mathbf{w}\xleftarrow{\$}\mathbb{Z}_q^m$. We say that $\mathsf{dLWE}_{n,m,q,\alpha}$ assumption holds if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{dLWE}_{n,m,q,\alpha}}$ is negligible for all PPT $\mathcal{A}$.

Regev [Reg05] (see also [GKV10]) showed that solving $\mathsf{dLWE}_{n,m,q,\alpha}$ for $\alpha q>2\sqrt{2n}$ is (quantumly) as hard as approximating the SIVP and GapSVP problems to within $\tilde{O}(n/\alpha)$ factors in the ℓ_2 norm, in the worst case. In the subsequent works, (partial) dequantumization of the Regev's reduction were achieved [Pei09, BLP+13].

Gadget Matrix. Let $m > n\lceil \log q \rceil$. There is a fixed full-rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ such that there exists a deterministic polynomial-time algorithm $\mathbf{G}^{-1}$ which takes the input $\mathbf{U} \in \mathbb{Z}_q^{n \times m}$ and outputs $\mathbf{V} = \mathbf{G}^{-1}(\mathbf{U})$ such that $\mathbf{V} \in \{0,1\}^{m \times m}$ and $\mathbf{GV} = \mathbf{U}$.

Trapdoors. Here, we follow the presentation of [BV16]. Let $n, m, q \in \mathbb{N}$ and consider a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. For all $\mathbf{V} \in \mathbb{Z}_q^{n \times m'}$, we let $\mathbf{A}_\sigma^{-1}(\mathbf{V})$ be a distribution that is a Gaussian $(D_{\mathbb{Z}^m,\sigma})^{m'}$ conditioned on $\mathbf{A} \cdot \mathbf{A}_\sigma^{-1}(\mathbf{V}) = \mathbf{V}$. A σ-trapdoor for $\mathbf{A}$ is a procedure that can sample from the distribution $\mathbf{A}_\sigma^{-1}(\mathbf{V})$ in time $\mathsf{poly}(n, m, m', \log q)$, for any $\mathbf{V}$. We slightly overload notation and denote a σ-trapdoor for $\mathbf{A}$ by $\mathbf{A}_\sigma^{-1}$. The following properties had been established in a long sequence of works [GPV08, ABB10a, CHKP10, ABB10b, MP12, BLP+13].

Lemma 2 (Properties of Trapdoors). *Lattice trapdoors exhibit the following properties.*

1. *Given* $\mathbf{A}_\sigma^{-1}$*, one can obtain* $\mathbf{A}_{\sigma'}^{-1}$ *for any* $\sigma' \geq \sigma$.
2. *Given* $\mathbf{A}_\sigma^{-1}$*, one can obtain* $[\mathbf{A}\|\mathbf{B}]_\sigma^{-1}$ *and* $[\mathbf{B}\|\mathbf{A}]_\sigma^{-1}$ *for any* $\mathbf{B}$.
3. *For all* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and* $\mathbf{R} \in \mathbb{Z}^{m \times m}$*, with* $m \geq n\lceil \log q \rceil$*, one can obtain* $[\mathbf{AR} + \mathbf{G}\|\mathbf{A}]_\sigma^{-1}$ *for* $\sigma = m \cdot \|\mathbf{R}\|_\infty \cdot \omega(\sqrt{\log m})$.
4. *There exists an efficient procedure* $\mathsf{TrapGen}(1^n, 1^m, q)$ *that outputs* $(\mathbf{A}, \mathbf{A}_{\sigma_0}^{-1})$ *where* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *for some* $m = O(n \log q)$ *and is* 2^{-n}*-close to uniform, where* $\sigma_0 = \omega(\sqrt{n \log q \log m})$.
5. *For* $\mathbf{A}_\sigma^{-1}$ *and* $\mathbf{u} \in \mathbb{Z}_q^n$*, it follows* $\Pr[\ \|\mathbf{A}_\sigma^{-1}(\mathbf{u})\| > \sqrt{m}\sigma\ ] = \mathsf{negl}(n)$.

Certified Bilinear Group Generators. We define certified bilinear group generators following [HJ16]. We require that there is an efficient bilinear group generator algorithm GrpGen that on input 1^λ and outputs a description Π of bilinear groups $\mathbb{G}, \mathbb{G}_T$ with prime order p and a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. We also require that GrpGen is certified, in the sense that there is an efficient algorithm GrpVfy that on input a (possibly incorrectly generated) description of the bilinear groups and outputs whether the description is valid or not. Furthermore, we require that each group element has unique encoding, which can be efficiently recognized. For the precise definitions, we refer to [HJ16] and the full version.

L-Diffie-Hellman Assumptions

Definition 2 (*L*-Diffie-Hellman Assumptions). *For a PPT algorithm* $\mathcal{A}$*, an advantage for the decisional L-Diffie Hellman problem L-*DDH *of* $\mathcal{A}$ *with respect to* GrpGen *is defined as follows:*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{L\text{-}\mathsf{DDH}} = |\ &\Pr[\mathcal{A}(\Pi, \hat{g}, h, \hat{g}^\alpha, \hat{g}^{\alpha^2}, \ldots \hat{g}^{\alpha^L}, \Psi_0) \to 1] \\ &- \Pr[\mathcal{A}(\Pi, \hat{g}, h, \hat{g}^\alpha, \hat{g}^{\alpha^2}, \ldots \hat{g}^{\alpha^L}, \Psi_1) \to 1]|\end{aligned}$$

where $\Pi \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$*,* $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$*,* $\hat{g}, h \xleftarrow{\$} \mathbb{G}$*,* $\Psi_0 = e(\hat{g}, h)^{1/\alpha}$*, and* $\Psi_1 \xleftarrow{\$} \mathbb{G}_T$*. We say that L-*$\mathsf{DDH}$ *assumption holds if* $\mathsf{Adv}_{\mathcal{A}}^{L\text{-}\mathsf{DDH}}$ *is negligible for all PPT* $\mathcal{A}$.

4 Partitioning Functions

In this section, we introduce the notion of *partitioning functions*. The notion abstracts out the information theoretic properties that are useful in the security proofs based on the partitioning techniques. Then, we proceed to recap the specific partitioning function that was given by [Jag15]. Then, we propose two new constructions of partitioning functions. The first one is obtained by introducing a simple but novel twist to the construction by [Jag15]. The second one is based on the affine-functions on random modulus. In the later sections, we will construct new lattice IBEs and VRFs based on these partitioning functions.

4.1 Definition

In the security proofs based on the partitioning technique [BB04b, Wat05], the simulations are successful only with noticeable probabilities. As observed by Waters [Wat05], this causes a subtle problem when considering the reduction to the decisional assumptions (such as the L-DDH). He resolved the problem by introducing the artificial abort step, where the simulator intentionally aborts with certain probability even when the simulation is successful. Later, Bellare and Ristenpart [BR09] showed that by requiring reasonable *upper bound* on the probability that the simulation is successful in addition to the *lower bound*, this step can be removed. In the subsequent work, Jager [Jag15] incorporated the idea of [BR09] into the notion of the admissible hash function [BB04b, CHKP10, FHPS13] to define *balanced admissible hash function*. The notion is a useful tool to perform the security proofs based on the partitioning technique. In addition, it is compatible with the decisional assumptions in the sense that it does not require the artificial abort step. Here, we define the notion of the partitioning function by slightly generalizing the balanced admissible hash function [Jag15].

Definition 3. *Let* $\mathsf{F} = \{\mathsf{F}_\lambda : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \{0,1\}\}$ *be an ensemble of function families. We say that* F *is a partitioning function, if there exists an efficient algorithm* $\mathsf{PrtSmp}(1^\lambda, Q, \epsilon)$*, which takes as input polynomially bounded* $Q = Q(\lambda) \in \mathbb{N}$ *and noticeable* $\epsilon = \epsilon(\lambda) \in (0, 1/2]$ *and outputs* K *such that:*

1. *There exists* $\lambda_0 \in \mathbb{N}$ *such that*

$$\Pr\left[K \in \mathcal{K}_\lambda \ : \ K \xleftarrow{\$} \mathsf{PrtSmp}\left(1^\lambda, Q(\lambda), \epsilon(\lambda)\right)\right] = 1$$

   for all $\lambda > \lambda_0$*. Here,* λ_0 *may depend on functions* $Q(\lambda)$ *and* $\epsilon(\lambda)$*.*
2. *For* $\lambda > \lambda_0$*, there exists* $\gamma_{\max}(\lambda)$ *and* $\gamma_{\min}(\lambda)$ *that depend on* $Q(\lambda)$ *and* $\epsilon(\lambda)$ *such that for all* $X^{(1)}, \ldots, X^{(Q)}, X^\star \in \mathcal{X}_\lambda$ *with* $X^\star \notin \{X^{(1)}, \ldots, X^{(Q)}\}$*,*

$$\gamma_{\max}(\lambda) \geq \gamma(X^{(1)}, \ldots, X^{(Q)}) \geq \gamma_{\min}(\lambda) \tag{7}$$

   holds where

$$\gamma(X^{(1)}, \ldots, X^{(Q)}) = \Pr\left[\left(\mathsf{F}(K, X^{(j)}) = 1 \quad \forall j \in [Q]\right) \wedge \ \mathsf{F}(K, X^\star) = 0\right]$$

and the function $\tau(\lambda)$ *defined as*

$$\tau(\lambda) := \gamma_{\min}(\lambda) \cdot \epsilon(\lambda) - \frac{\gamma_{\max}(\lambda) - \gamma_{\min}(\lambda)}{2} \tag{8}$$

is noticeable. We note that the probability above is taken over the choice of $K \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q(\lambda), \epsilon(\lambda))$.

We call K *the partitioning key and* $\tau(\lambda)$ *the quality of the partitioning function.*

In the following, we often drop the subscript λ and denote F, $\mathcal{K}$, and $\mathcal{X}$ for the sake of simplicity. We remark that the term $\tau(\lambda)$ above, which may seem very specific, is inherited from [Jag15]. As explained in [Jag15], such a term appears typically in security analyses that follows the approach of Bellare and Ristenpart [BR09] (See also Lemma 1). Looking ahead, the quantity $\tau(\lambda)$ will directly affect the reduction cost of our IBEs and VRFs. The length of (the binary representation of) the partitioning key K will affect the efficiency of the resulting schemes. Therefore, we want the partitioning function F for the largest possible $\tau(\lambda)$ and the shortest possible partitioning key.

There are two main differences from the definition of [Jag15]. Firstly, we consider *any* function F, whereas they only considered a specific function (namely, $\mathsf{F}_{\mathsf{ADH}}$ in Sect. 4.2). Secondly, we explicitly add the condition regarding the domain correctness of the output of PrtSmp (the first condition), which was implicit in [Jag15].

Comparison with Programmable Hash Functions. Our notion of the partitioning function is similar to the programmable hash function [HK08, ZCZ16]. The main difference is that whereas the notion of the programmable hash function is defined on specific algebraic structures such as (bilinear) groups [HK08] and lattices [ZCZ16], our definition is irrelevant to them. Since the security proofs of our IBEs and VRFs have the same information theoretic structure in common, we choose to decouple them from the underlying algebraic structures.

4.2 Construction from Admissible Hash Function

Here, we recap the result of Jager [Jag15] who constructed a specific partitioning function that he calls balanced admissible hash function. The result will be used in the next subsection to construct our first partitioning function. Let $k(\lambda) = \Theta(\lambda)$ and $\ell(\lambda) = \Theta(\lambda)$ be integers and let $\{C_k : \{0,1\}^k \to \{0,1\}^\ell\}_{k \in \mathbb{N}}$ be a family of error correcting codes with minimal distance ℓc for a constant $c \in (0, 1/2)$. Explicit constructions of such codes are given in [SS96, Zém01, Gol08] for instance. Let us define

$$\mathcal{K}_{\mathsf{ADH}} = \{0, 1, \bot\}^\ell \qquad \text{and} \qquad \mathcal{X}_{\mathsf{ADH}} = \{0,1\}^k.$$

We define $\mathsf{F}_{\mathsf{ADH}}$ as

$$\mathsf{F}_{\mathsf{ADH}}(K, X) = \begin{cases} 0, & \text{if } \forall i \in [\ell] : \ C(X)_i = K_i \quad \vee \quad K_i = \bot \\ 1, & \text{otherwise} \end{cases}$$

where $C(X)_i$ and K_i are the i-th significant bit of $C(X)$ and K, respectively. Jager [Jag15] showed the following theorem.

Theorem 1 (Adapted from Theorem 1 in [Jag15]). *There exists an efficient algorithm* $\mathsf{AdmSmp}(1^\lambda, Q, \epsilon)$*, which takes as input* $Q \in \mathbb{N}$ *and* $\epsilon \in (0, 1/2]$ *and outputs* K *with exactly* η' *components not equal to* $\perp$*, where*

$$\eta' := \left\lfloor \frac{\log(2Q + Q/\epsilon)}{-\log(1-c)} \right\rfloor,$$

such that Eqs. (7) and (8) hold with respect to $\mathsf{F} := \mathsf{F}_{\mathsf{ADH}}$, $\mathsf{PrtSmp} := \mathsf{AdmSmp}$*, and* $\tau(\lambda) = 2^{-\eta'-1} \cdot \epsilon$*. In particular,* $\mathsf{F}_{\mathsf{ADH}}$ *is a partitioning function.*

4.3 Our Construction Based on Modified Admissible Hash Function

Here, we propose our first construction of the partitioning function $\mathsf{F}_{\mathsf{MAH}}$, which is obtained by modifying $\mathsf{F}_{\mathsf{ADH}}$ in the previous subsection. The advantage of $\mathsf{F}_{\mathsf{MAH}}$ is that it achieves much shorter partitioning keys compared with $\mathsf{F}_{\mathsf{ADH}}$. In particular, the length is $\omega(\log^2 \lambda)$ in $\mathsf{F}_{\mathsf{MAH}}$, whereas $\Theta(\lambda)$ in $\mathsf{F}_{\mathsf{ADH}}$. We will use the same notation as in Sect. 4.2. Let us introduce an integer $\eta(\lambda) = \omega(\log \lambda)$. $\eta(\lambda)$ can be set arbitrarily as long as it grows faster than $\log \lambda$. (See footnote in Sect. 1.) For our construction, we set

$$\mathcal{K}_{\mathsf{MAH}} = \{\mathsf{T} \subseteq [2\ell] \quad | \quad |\mathsf{T}| < \eta\} \qquad \text{and} \qquad \mathcal{X}_{\mathsf{MAH}} = \{0,1\}^k.$$

We define $\mathsf{F}_{\mathsf{MAH}}$ as

$$\mathsf{F}_{\mathsf{MAH}}(\mathsf{T}, X) = \begin{cases} 0, & \text{if } \mathsf{T} \subseteq \mathsf{S}(X) \\ 1, & \text{otherwise} \end{cases} \qquad \text{where} \quad \mathsf{S}(X) = \{\ 2i - C(X)_i \mid i \in [\ell]\ \}.$$

In the above, $C(X)_i$ is the i-th bit of $C(X) \in \{0,1\}^\ell$. See Fig. 1 in Sect. 2.1 for an illustrative example of S.

Lemma 3. *The function* $\mathsf{F}_{\mathsf{MAH}}$ *defined above is a partitioning function.*

Proof. To prove the lemma, we define $\mathsf{PrtSmp}_{\mathsf{MAH}}$ as follows. It uses the algorithm AdmSmp from the previous subsection as a subroutine.

$\mathsf{PrtSmp}_{\mathsf{MAH}}(1^\lambda, Q, \epsilon)$: It runs $\mathsf{AdmSmp}(1^\lambda, Q, \epsilon) \to K$ and sets

$$\mathsf{T} = \{2i - K_i \mid i \in [\ell],\ K_i \neq \perp\} \subseteq [2\ell],$$

where K_i is the i-th bit of K. It finally outputs T.

See Fig. 1 in Sect. 2.1 for an illustrative example of T. We first show that $\mathsf{PrtSmp}_{\mathsf{MAH}}$ satisfies the first property of Definition 3. By Theorem 1, $|\mathsf{T}| = \eta' = \lceil \log(2Q + Q/\epsilon) / \log(1-c) \rceil$. To show $\mathsf{T} \in \mathcal{K}_{\mathsf{MAH}}$ for all sufficiently large λ, it suffices to show $\eta'(\lambda) < \eta(\lambda)$ for all sufficiently large λ. This follows since

$$\eta'(\lambda) = \left\lfloor \frac{\log(2Q + Q/\epsilon)}{-\log(1-c)} \right\rfloor = O\left(\log(\mathsf{poly}(\lambda))\right) = O(\log \lambda) \quad \text{and} \quad \eta(\lambda) = \omega(\log \lambda)$$

when $Q(\lambda)$ is polynomially bounded and ϵ is noticeable for constant c. We next prove the second property. This follows from Theorem 1 and by the following observation:

$$\begin{aligned}\mathsf{F}_{\mathsf{ADH}}(K, X) = 0 \quad &\Leftrightarrow C(X)_i = K_i \quad \forall i \in [\ell] \,\text{such that}\, K_i \neq \bot \\ &\Leftrightarrow \mathsf{T} \subseteq \mathsf{S}(X) \\ &\Leftrightarrow \mathsf{F}_{\mathsf{MAH}}(\mathsf{T}, X) = 0.\end{aligned}$$

This completes the proof of Lemma 3.

4.4 Our Construction Based on Affine Functions

Here, we propose our second construction of the partitioning function $\mathsf{F}_{\mathsf{AFF}}$. Compared to $\mathsf{F}_{\mathsf{MAH}}$, the function achieves an even shorter length of $\omega(\log \lambda)$ for the partitioning keys. Let $k(\lambda) = \Theta(\lambda)$ and $\eta(\lambda) = \omega(\log \lambda)$ be integers. For our construction, we set

$$\mathcal{K}_{\mathsf{AFF}} = \{0,1\}^{3\eta}, \qquad \mathcal{X}_{\mathsf{AFF}} = \{0,1\}^{k}$$

$\mathsf{F}_{\mathsf{AFF}}(K, X)$ is defined as

$$\mathsf{F}_{\mathsf{AFF}}(K = (\alpha, \beta, \rho),\ X) = \begin{cases} 0, & \text{if } \rho \neq 0 \quad \wedge \quad \alpha X + \beta \equiv 0 \mod \rho \\ 1, & \text{otherwise} \end{cases},$$

where $\alpha, \beta, \rho \in \{0,1\}^{\eta}$. Here, we slightly abuse the notation and identify a bit-string in $\{0,1\}^{\eta}$ with an integer in $[0, 2^{\eta} - 1]$ by its binary representation. Similarly, a bit-string in $\{0,1\}^{k}$ is identified with an integer in $[0, 2^{k} - 1]$.

Theorem 2. $\mathsf{F}_{\mathsf{AFF}}$ *defined above is a partitioning function.*

5 Our IBE Schemes

In this section, we give a generic construction of an adaptively secure lattice based IBE from a partitioning function. Our generic construction requires the underlying partitioning function to be compatible (in some sense) with the structure of lattices. In the following, we first formalize the requirement by giving the definition of compatibility. Then, we show that $\mathsf{F}_{\mathsf{MAH}}$ and $\mathsf{F}_{\mathsf{AFF}}$ are compatible in this sense. Finally, we show the generic construction of IBE.

5.1 Compatible Algorithms for Partitioning Functions

The following definition gives a sufficient condition for partitioning functions to be useful for constructing adaptively secure IBE schemes.

Definition 4. *We say that the* deterministic *algorithms* $(\mathsf{Encode}, \mathsf{PubEval}, \mathsf{TrapEval})$ *are* δ*-compatible with a function family* $\{\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \{0,1\}\}$ *if they are efficient and satisfy the following properties:*

- $\mathsf{Encode}(K \in \mathcal{K}) \to \kappa \in \{0,1\}^u$
- $\mathsf{PubEval}\left(X \in \mathcal{X}, \{\mathbf{B}_i \in \mathbb{Z}_q^{n\times m}\}_{i\in[u]}\right) \to \mathbf{B}_X \in \mathbb{Z}_q^{n\times m}$
- $\mathsf{TrapEval}\left(K \in \mathcal{K},\ X \in \mathcal{X},\ \mathbf{A} \in \mathbb{Z}_q^{n\times m},\ \{\mathbf{R}_i \in \mathbb{Z}^{m\times m}\}_{i\in[u]}\right) \to \mathbf{R}_X \in \mathbb{Z}^{m\times m}$
  We require that the following holds:

$$\mathsf{PubEval}\left(X, \{\mathbf{A}\mathbf{R}_i + \kappa_i\mathbf{G}\}_{i\in[u]}\right) = \mathbf{A}\mathbf{R}_X + \mathsf{F}(K,X)\cdot\mathbf{G}$$

where $\kappa_i \in \{0,1\}$ is the i-th bit of $\kappa = \mathsf{Encode}(K) \in \{0,1\}^u$. Furthermore, if $\mathbf{R}_i \in \{-1,0,1\}^{m\times m}$ for all $i \in [u]$, we have $\|\mathbf{R}_X\|_\infty \le \delta$.

It is possible to obtain compatible algorithms for any partitioning functions, including ours, by directly leveraging the fully key homomorphic algorithm in [BGG+14]. However, if we apply the algorithm naively, it will end up with super-polynomial δ, which is undesirable. By carefully applying the idea from [GV15], we can provide δ-compatible algorithms for $\mathsf{F}_{\mathsf{MAH}}$ and $\mathsf{F}_{\mathsf{AFF}}$ with polynomial δ. In particular, we have following lemmas.

Lemma 4. *For $u = \eta\cdot\lceil\log(2\ell+1)\rceil$, there are $m^3u(\ell+1)$-compatible algorithms for $\mathsf{F}_{\mathsf{MAH}}$.*

Lemma 5. *For $u = 3\eta$, there are $\mathsf{poly}(n)$-compatible algorithm for $\mathsf{F}_{\mathsf{AFF}}$, where $\mathsf{poly}(n)$ denotes some fixed polynomial in n.*

5.2 Construction

Here, we construct an IBE scheme based on a partitioning function $\mathsf{F} : \mathcal{K}\times\mathcal{X} \to \{0,1\}$ with associating δ-compatible algorithms $(\mathsf{Encode}, \mathsf{PubEval}, \mathsf{TrapEval})$. We assume $\mathcal{X} = \mathcal{ID} = \{0,1\}^k$, where $\mathcal{ID}$ is the identity space of the scheme. If a collision resistant hash $\mathsf{CRH} : \{0,1\}^* \to \{0,1\}^k$ is available, we can use any bit-string as an identity. For simplicity, we let the message space of the scheme be $\{0,1\}$. For the multi-bit variant, we refer to Sect. 5.3. Our scheme can be instantiated with $\mathsf{F}_{\mathsf{MAH}}$ and $\mathsf{F}_{\mathsf{AFF}}$, which would lead to schemes with efficiency and security trade-offs. We compare the resulting schemes with existing schemes in Sect. 7. (See also Table 1 in Sect. 1.)

$\mathsf{Setup}(1^\lambda)$: On input 1^λ, it sets the parameters n, m, q, σ, α, and α' as specified later in this section, where q is a prime number. Then, it picks random matrices $\mathbf{B}_0, \mathbf{B}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ for $i \in [u]$ and a vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$. It also picks $(\mathbf{A}, \mathbf{A}_{\sigma_0}^{-1}) \xleftarrow{\$} \mathsf{TrapGen}(1^n, 1^m, q)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\sigma_0 = \omega(\sqrt{n\log q\log m})$. It finally outputs

$$\mathsf{mpk} = \left(\ \mathbf{A},\ \mathbf{B}_0,\ \{\mathbf{B}_i\}_{i\in[u]},\ \mathbf{u}\ \right) \qquad \text{and} \qquad \mathsf{msk} = \mathbf{A}_{\sigma_0}^{-1}.$$

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{ID})$: Given an identity ID, it first computes

$$\mathsf{PubEval}\left(\mathsf{ID},\ \{\mathbf{B}_i\}_{i\in[u]}\ \right) \to \mathbf{B}_{\mathsf{ID}} \in \mathbb{Z}_q^{n\times m}.$$

It then computes $[\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{\mathsf{ID}}]_\sigma^{-1}$ from $\mathbf{A}_{\sigma_0}^{-1}$ and samples

$$\mathbf{e} \xleftarrow{\$} [\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{\mathsf{ID}}]_\sigma^{-1}(\mathbf{u}).$$

Then, it returns $\mathsf{sk}_{\mathsf{ID}} = \mathbf{e} \in \mathbb{Z}^{2m}$. Note that we have $[\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{\mathsf{ID}}] \cdot \mathbf{e} = \mathbf{u} \bmod q$.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: To encrypt a message $\mathsf{M} \in \{0,1\}$ for an identity ID, it first computes $\mathsf{PubEval}(\mathsf{ID}, \{\mathbf{B}_i\}_{i\in[u]}) \to \mathbf{B}_{\mathsf{ID}}$. It then picks $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $x_0 \xleftarrow{\$} D_{\mathbb{Z},\alpha q}$, $\mathbf{x}_1, \mathbf{x}_2 \xleftarrow{\$} D_{\mathbb{Z}^m,\alpha' q}$ and computes

$$c_0 = \mathbf{s}^\top \mathbf{u} + x_0 + \mathsf{M} \cdot \lceil q/2 \rceil, \qquad \mathbf{c}_1^\top = \mathbf{s}^\top [\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{\mathsf{ID}}] + [\mathbf{x}_1^\top \| \mathbf{x}_2^\top].$$

Finally, it returns the ciphertext $\mathsf{ct} = (c_0, \mathbf{c}_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$.

$\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{ct})$: To decrypt a ciphertext $\mathsf{ct} = (c_0, \mathbf{c}_1)$ using a private key $\mathsf{sk}_{\mathsf{ID}} := \mathbf{e}$, it first computes

$$w = c_0 - \mathbf{c}_1^\top \cdot \mathbf{e} \in \mathbb{Z}_q.$$

Then it returns 1 if $|w - \lceil q/2 \rceil| < \lceil q/4 \rceil$ and 0 otherwise.

We claim that the correctness and security of the scheme can be proven under the following parameter selection. We refer full version to the justification.

$$m = O(n \log q), \qquad q = n^{7/2} \cdot \delta^2 \cdot \omega(\log^{7/2} n), \qquad \sigma = m \cdot \delta \cdot \omega(\sqrt{\log m})$$
$$\alpha q = 3\sqrt{n}, \qquad \alpha' q = 5\sqrt{n} \cdot m \cdot \delta.$$

Here, the parameter δ is determined by the compatible algorithms corresponding to F. The following theorem addresses the security of the scheme.

Theorem 3. *If* $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \{0,1\}$ *is a partitioning function and* $(\mathsf{Encode}, \mathsf{PubEval}, \mathsf{TrapEval})$ *are the corresponding* δ*-compatible algorithms, our scheme achieves adaptively-anonymous security assuming* $\mathsf{dLWE}_{n,m+1,q,\alpha}$.

5.3 Multi-bit Variant

Here, we explain how to extend our scheme to be a multi-bit variant without increasing much the size of the master public keys and ciphertexts following [PVW08, ABB10a, Yam16]. (However, it comes with longer private keys.) To modify the scheme so that it can deal with the message space of length ℓ_M, we replace $\mathbf{u} \in \mathbb{Z}_q^n$ in mpk with $\mathbf{U} \in \mathbb{Z}_q^{n\times\ell_M}$. The component c_0 in the ciphertext is replaced with $\mathbf{c}_0^\top = \mathbf{s}^\top \mathbf{U} + \mathbf{x}_0^\top + \mathsf{M}\lceil q/2 \rceil$, where $\mathbf{x}_0 \xleftarrow{\$} D_{\mathbb{Z}^{\ell_M},\alpha q}$ and $\mathsf{M} \in \{0,1\}^{\ell_M}$ is the message to be encrypted. The private key is replaced to be $\mathbf{E} \in \mathbb{Z}^{m\times\ell_M}$, where $\mathbf{E}$ is chosen as $\mathbf{E} \xleftarrow{\$} [\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{\mathsf{ID}}]_\sigma^{-1}(\mathbf{U})$. We can prove security for the multi-bit variant from $\mathsf{dLWE}_{n,m+\ell_M,q,\alpha}$ by naturally extending the proof of Theorem 3. We note that the same parameters as in Sect. 5.2 will also work for the multi-bit variant. By this change, the sizes of the master public keys,

ciphertexts, and private keys become $\tilde{O}(n^2u + n\ell_M)$, $\tilde{O}(n + \ell_M)$, and $\tilde{O}(n\ell_M)$ from $\tilde{O}(n^2u)$, $\tilde{O}(n)$, and $\tilde{O}(n)$, respectively. The sizes of the master public keys and ciphertexts will be asymptotically the same as long as $\ell_M = \tilde{O}(n)$. To deal with longer messages, we employ a KEM-DEM approach as suggested in [Yam16]. Namely, we encrypt a random ephemeral key of sufficient length and then encrypt the message by using the ephemeral key.

6 Our VRF Scheme Based on $\mathsf{F}_{\mathsf{MAH}}$

6.1 Construction

Here, we construct a verifiable random function scheme based on the partitioning function $\mathsf{F}_{\mathsf{MAH}}$. We let the input and output space of the scheme be $\mathcal{X} = \{0,1\}^k$ and $\mathcal{Y} = \mathbb{G}_T$, respectively. Let $\eta := \eta(\lambda)$, $\ell := \ell(\lambda)$, $C : \{0,1\}^k \to \{0,1\}^\ell$, and S be as in Sect. 4.3. We also introduce $\ell_1 := \ell_1(\lambda)$ and $\ell_2 = \ell_2(\lambda)$ such that $\ell = \ell_1\ell_2$. These parameters will control the trade-offs between sizes of proofs and verification keys. A typical choice would be $(\ell_1, \ell_2) = (O(\sqrt{\ell}), O(\sqrt{\ell}))$ or $(\ell_1, \ell_2) = (O(\ell), O(1))$.

$\mathsf{Gen}(1^\lambda)$: On input 1^λ, it chooses a group description $\mathit{\Pi} \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$. It chooses random generators $g, h \xleftarrow{\$} \mathbb{G}^*$ and $w_1, \ldots, w_\eta \xleftarrow{\$} \mathbb{Z}_p$. It then outputs

$$\mathsf{vk} = \left(\mathit{\Pi},\ g,\ h,\ \left\{ W_{i,j_1} := g^{w_i^{j_1}} \right\}_{(i,j_1)\in[\eta]\times[\ell_1]} \right) \quad \text{and} \quad \mathsf{sk} = \left(\{w_i\}_{i\in[\eta]} \right).$$

$\mathsf{Eval}(\mathsf{sk}, X)$: Given $X \in \{0,1\}^k$, it first computes $\mathsf{S}(X) = \{s_1, \ldots, s_\ell\} \subset [2\ell]$,

$$\theta = \prod_{(i,j)\in[\eta]\times[\ell]} (w_i + s_j), \qquad \text{and} \qquad \theta_{i,j_2} = \prod_{(i',j')\in\Omega_{i,j_2}} (w_{i'} + s_{j'}) \tag{9}$$

for $(i, j_2) \in [\eta] \times [\ell_2]$, where

$$\Omega_{i,j_2} = \{(i',j') \in [\eta] \times [\ell] \quad | \quad (i' \in [i-1]) \ \vee \ (i' = i \ \wedge \ j' \in [j_2\ell_1])\}.$$

We note that $\theta = \theta_{\eta,\ell_2}$. If $\theta \equiv 0 \mod p$, it outputs $Y = 1_{\mathbb{G}_T}$ and $\pi = (\{\pi_{i,j_2} = 1_{\mathbb{G}}\}_{(i,j_2)\in[\eta]\times[\ell_2]})$[4]. Otherwise, it outputs

$$Y = e(g,h)^{1/\theta} \qquad \text{and} \qquad \pi = \left(\left\{ \pi_{i,j_2} = g^{1/\theta_{i,j_2}} \right\}_{(i,j_2)\in[\eta]\times[\ell_2]} \right).$$

$\mathsf{Verify}(\mathsf{vk}, X, Y, \pi)$: It first checks the validity of vk by the following steps. It outputs 0 if any of the following does not hold:

[4] The event occurs with only negligible probability. This choice of the output is arbitrary and can be replaced with any fixed group elements.

1. vk is of the form $(\mathit{\Pi}, g, h, \{W_{i,j_1}\}_{(i,j_1)\in[\eta]\times[\ell_1]})$.
2. $\mathsf{GrpVfy}(\mathit{\Pi}) \to 1$, $g, h \in \mathbb{G}^*$, and $W_{i,j_1} \in \mathbb{G}$ for all $(i, j_1) \in [\eta] \times [\ell_1]$.
3. $e(W_{i,1}, W_{i,j_1-1}) = e(g, W_{i,j_1})$ for all $(i, j_1) \in [\eta] \times [2, \ell_1]$.

It then checks the validity of Y and π. To do this, it computes $\mathit{\Phi}_{i,j_2} \in \mathbb{G}$ for $(i, j_2) \in [\eta] \times [\ell_2]$ as

$$\mathit{\Phi}_{i,j_2} := g^{\varphi_{j_2,0}} \cdot \prod_{j_1\in[\ell_1]} W_{i,j_1}^{\varphi_{j_2,j_1}}, \tag{10}$$

where $\{\varphi_{j_2,j_1} \in \mathbb{Z}_p\}_{(j_2,j_1)\in[\ell_2]\times[0,\ell_1]}$ are the coefficients of the following polynomial:

$$\prod_{j'\in[(j_2-1)\ell_1+1,j_2\ell_1]} (\mathsf{Z} + s_{j'}) = \varphi_{j_2,0} + \sum_{j_1\in[\ell_1]} \varphi_{j_2,j_1} \mathsf{Z}^{j_1} \in \mathbb{Z}_p[\mathsf{Z}].$$

It outputs 0 if any of the following does not hold:

4. $X \in \{0,1\}^k$, $Y \in \mathbb{G}_T$, π is of the form $\pi = (\{\pi_{i,j_2} \in \mathbb{G}\}_{(i,j_2)\in[\eta]\times[\ell_2]})$.
5. If there exists $(i, j_2) \in [\eta] \times [\ell_2]$ such that $\mathit{\Phi}_{i,j_2} = 1_{\mathbb{G}}$, we have $Y = 1_{\mathbb{G}_T}$ and $\pi_{i,j_2} = 1_{\mathbb{G}}$ for all $(i, j_2) \in [\eta] \times [\ell_2]$.
6. If $\mathit{\Phi}_{i,j_2} \neq 1_{\mathbb{G}}$ for all $(i, j_2) \in [\eta] \times [\ell_2]$, the following equation holds for all $(i, j_2) \in [\eta] \times [\ell_2]$:

$$e(\pi_{i,j_2}, \mathit{\Phi}_{i,j_2}) = e(\pi_{i,j_2-1}, g) \tag{11}$$

   where we define $\pi_{i,0} := \pi_{i-1,\ell_2}$ for $i \geq 2$ and $\pi_{1,0} := g$.
7. $e(\pi_{\eta,\ell_2}, h) = Y$ holds.

If all the above conditions hold, it outputs 1.

The correctness and unique provability of the scheme can be proven by a standard argument. The following theorem addresses the pseudorandomness of the scheme.

Theorem 4. *Our scheme satisfies pseudorandomness assuming* L-DDH *with* $L = (4\ell + 1)\eta + \ell_1$.

Proof. Let $\mathcal{A}$ be a PPT adversary that breaks pseudorandomness of the scheme. In addition, let $\epsilon = \epsilon(\lambda)$ and $Q = Q(\lambda)$ be its advantage and the upper bound on the number of evaluation queries, respectively. By assumption, $Q(\lambda)$ is polynomially bounded and there exists a noticeable function $\epsilon_0(\lambda)$ such that $\epsilon(\lambda) \geq \epsilon_0(\lambda)$ holds for infinitely many λ. By the property of the partitioning function (Definition 3, Item 1), we have that

$$|\mathsf{T}| < \eta \qquad \text{where} \qquad \mathsf{T} \xleftarrow{\$} \mathsf{PrtSmp}_{\mathsf{MAH}}(1^\lambda, Q, \epsilon_0)$$

holds with probability 1 for all sufficiently large λ. Therefore, in the following, we assume that this condition always holds. We show the security of the scheme via the following sequence of games. In each game, a value $\mathsf{coin}' \in \{0,1\}$ is defined. While it is set $\mathsf{coin}' = \widehat{\mathsf{coin}}$ in the first game, these values might be different in the later games. In the following, we define E_i be the event that $\mathsf{coin}' = \mathsf{coin}$.

Game_0 : This is the real security game. Recall that since the range of the function is $\mathcal{Y} = \mathbb{G}_T$, in the challenge phase, $Y_1^\star \xleftarrow{\$} \mathbb{G}_T$ is returned to $\mathcal{A}$ if $\mathsf{coin} = 1$. At the end of the game, $\mathcal{A}$ outputs a guess $\widehat{\mathsf{coin}}$ for coin. Finally, the challenger sets $\mathsf{coin}' = \widehat{\mathsf{coin}}$. By definition, we have

$$\left|\Pr[\mathsf{E}_0] - \frac{1}{2}\right| = \left|\Pr[\mathsf{coin}' = \mathsf{coin}] - \frac{1}{2}\right| = \left|\Pr[\widehat{\mathsf{coin}} = \mathsf{coin}] - \frac{1}{2}\right| = \epsilon.$$

Game_1 : In this game, we change Game_0 so that the challenger performs the following additional step at the end of the game. First, the challenger runs $\mathsf{PrtSmp}_{\mathsf{MAH}}(1^\lambda, Q, \epsilon_0) \to \mathsf{T} \subseteq [2\ell]$ and checks whether the following condition holds:

$$\mathsf{T} \not\subseteq \mathsf{S}(X^{(1)}) \ \wedge \ \cdots \ \wedge \ \mathsf{T} \not\subseteq \mathsf{S}(X^{(Q)}) \ \wedge \ \mathsf{T} \subseteq \mathsf{S}(X^\star) \tag{12}$$

where $X^\star$ is chosen by $\mathcal{A}$ at the challenge phase, and $X^{(1)}, \ldots, X^{(Q)}$ are inputs to the VRF for which $\mathcal{A}$ has queried the evaluation of the function. If it does not hold, the challenger ignores the output $\widehat{\mathsf{coin}}$ of $\mathcal{A}$, and sets $\mathsf{coin}' \xleftarrow{\$} \{0,1\}$. In this case, we say that the challenger aborts. If condition (12) holds, the challenger sets $\mathsf{coin}' = \widehat{\mathsf{coin}}$. By Lemmas 1 and 3 (See also Definition 3, Item 2),

$$\left|\Pr[\mathsf{E}_1] - \frac{1}{2}\right| \geq \gamma_{\min}\epsilon - \frac{\gamma_{\max} - \gamma_{\min}}{2} \geq \gamma_{\min}\epsilon_0 - \frac{\gamma_{\max} - \gamma_{\min}}{2} = \tau$$

holds for infinitely many λ and a noticeable function $\tau = \tau(\lambda)$. Here, $\gamma_{\min}$, $\gamma_{\max}$, and τ are specified by ϵ_0, Q, and the underlying partitioning function $\mathsf{F}_{\mathsf{MAH}}$.

Game_2 : In this game, we change the way w_i are chosen. At the beginning of the game, the challenger picks $\mathsf{T} \xleftarrow{\$} \mathsf{PrtSmp}_{\mathsf{MAH}}(1^\lambda, Q, \epsilon_0)$ and parses it as $\mathsf{T} = \{t_1, \ldots, t_{\eta'}\} \subset [2\ell]$. Recall that by our assumption, we have $\eta' < \eta$. It then sets $t_i := 0$ for $i \in [\eta' + 1, \eta]$. It then samples $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$, and $\tilde{w}_i \xleftarrow{\$} \mathbb{Z}_p^*$ for $i \in [\eta]$. Then, w_i are defined as

$$w_i = \tilde{w}_i \cdot \alpha - t_i \qquad \text{for } i \in [\eta].$$

The rest of the game is the same as in Game_1. The statistical distance of the distributions of $\{w_i\}_{i\in[\eta]}$ in Game_1 and Game_2 is at most η/p, which is negligible. Therefore, we have $|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| = \mathsf{negl}(\lambda)$.

Before describing the next game, for any $\Omega \subseteq [\eta] \times [\ell]$, $\mathsf{T} \subset [2\ell]$ with $|\mathsf{T}| = \eta' < \eta$, and $X \in \{0,1\}^k$, we define polynomials $\mathsf{P}_{X,\Omega}(\mathsf{Z}), \mathsf{Q}(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ as

$$\mathsf{P}_{X,\Omega}(\mathsf{Z}) = \prod_{(i,j)\in\Omega} (\tilde{w}_i \mathsf{Z} - t_i + s_j)$$

$$\text{and} \qquad \mathsf{Q}(\mathsf{Z}) = \mathsf{Z}^{\eta'-1} \cdot \prod_{(i,j)\in[\eta]\times[-2\ell,2\ell]\setminus\{0\}} (\tilde{w}_i \mathsf{Z} + j)\,,$$

where $\{s_j\}_{j\in[\ell]} = \mathsf{S}(X)$ and $\{t_i\}_{i\in[\eta]}$ are defined as in Game_2 (namely, $\mathsf{T} = \{t_i\}_{i\in[\eta']}$ and $t_i = 0$ for $i > \eta'$). In the special case of $\Omega = [\eta] \times [\ell]$, we denote $\mathsf{P}_X(\mathsf{Z}) := \mathsf{P}_{X,[\eta]\times[\ell]}(\mathsf{Z})$. We state the following lemma, which plays an important roll in our security proof.

Lemma 6. *There exist* $\xi_X \in \mathbb{Z}_p^*$ *and* $\mathsf{R}_X(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ *such that*

$$\frac{\mathsf{Q}(\mathsf{Z})}{\mathsf{P}_X(\mathsf{Z})} = \begin{cases} \dfrac{\xi_X}{\mathsf{Z}} + \mathsf{R}_X(\mathsf{Z}) & \text{if} \quad \mathsf{T} \subseteq \mathsf{S}(X) \\ \mathsf{R}_X(\mathsf{Z}) & \text{if} \quad \mathsf{T} \not\subseteq \mathsf{S}(X) \end{cases}.$$

From the above lemma, we can see that for any $\Omega \subseteq [\eta] \times [\ell]$, it holds that

$$\mathsf{P}_{X,\Omega}(\mathsf{Z}) \mid \mathsf{Q}(\mathsf{Z}) \qquad \text{if} \qquad \mathsf{T} \not\subseteq \mathsf{S}(X),$$

because $\mathsf{P}_{X,\Omega}(\mathsf{Z}) \mid \mathsf{P}_X(\mathsf{Z})$.

Game_3 Recall that in the previous game, the challenger aborts at the end of the game, if condition (12) is not satisfied. In this game, we change the game so that the challenger aborts as soon as the abort condition becomes true. Since this is only a conceptual change, we have $\Pr[\mathsf{E}_2] = \Pr[\mathsf{E}_3]$.

Game_4 In this game, we change the way g is sampled. Namely, Game_4 challenger first picks α and $\tilde{w}_i$ as specified in Game_2. It further picks $\hat{g} \xleftarrow{\$} \mathbb{G}^*$. Then, it computes (coefficients of) $\mathsf{Q}(\mathsf{Z})$ and sets

$$g := \hat{g}^{\mathsf{Q}(\alpha)}, \qquad W_{i,j_1} = g^{w_i^{j_1}} = \hat{g}^{\mathsf{Q}(\alpha)\cdot(\tilde{w}_i\alpha - t_i)^{j_1}} \qquad \text{for} \qquad (i, j_1) \in [\eta] \times [\ell_1].$$

It aborts and outputs a random bit if $g = 1_{\mathbb{G}} \Leftrightarrow \mathsf{Q}(\alpha) \equiv 0 \mod p$. It can be seen that the distribution of g and W_{i,j_1} is unchanged, unless $\mathsf{Q}(\alpha) \equiv 0 \mod p$. Since $\mathsf{Q}(\mathsf{Z})$ is a non-zero polynomial with degree $(4\eta\ell + \eta' - 1)$ and α is chosen uniformly at random from $\mathbb{Z}_p^*$, it follows from the Schwartz-Zippel lemma that this happens with probability at most $(4\eta\ell + \eta' - 1)/(p-1) = \mathsf{negl}(\lambda)$. We therefore have $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| = \mathsf{negl}(\lambda)$.

Game_5 In this game, we change the way the evaluation queries are answered. By the change introduced in Game_4, we assume $\mathsf{Q}(\alpha) \not\equiv 0 \mod p$ in the following. When $\mathcal{A}$ makes a query for an input X, the challenger first checks whether $\mathsf{T} \subseteq \mathsf{S}(X)$ and aborts if so (as specified in Game_3). Otherwise, it computes $\mathsf{R}_{X,\Omega_{i,j_2}}(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ such that $\mathsf{Q}(\mathsf{Z}) = \mathsf{P}_{X,\Omega_{i,j_2}}(\mathsf{Z}) \cdot \mathsf{R}_{X,\Omega_{i,j_2}}(\mathsf{Z})$ for $(i, j_2) \in [\eta] \times [\ell_2]$. Note that such polynomials exist by Lemma 6. Then, it returns

$$Y = e\left(\hat{g}^{\mathsf{R}_{X,\Omega_{\eta,\ell_2}}(\alpha)}, h\right), \qquad \pi = \left(\left\{\pi_{i,j_2} = \hat{g}^{\mathsf{R}_{X,\Omega_{i,j_2}}(\alpha)}\right\}_{(i,j_2)\in[\eta]\times[\ell_2]}\right)$$

to $\mathcal{A}$. We claim that this is only a conceptual change. To see this, we first observe that

$$\mathsf{P}_{X,\Omega_{i,j_2}}(\alpha) = \prod_{(i',j')\in\Omega_{i,j_2}} (\tilde{w}_{i'}\alpha - t_{i'} + s_{j'})$$

$$= \prod_{(i',j')\in\Omega_{i,j_2}} (w_{i'} + s_{j'}) = \theta_{i,j_2}. \tag{13}$$

We have $\theta_{i,j_2} \not\equiv 0 \mod p$, since otherwise we have $\mathsf{Q}(\alpha) \equiv \mathsf{P}_{X,\Omega_{i,j_2}}(\alpha) \cdot \mathsf{R}_{X,\Omega_{i,j_2}}(\alpha) \equiv \theta_{i,j_2} \cdot \mathsf{R}_{X,\Omega_{i,j_2}}(\alpha) \equiv 0 \mod p$, which is a contradiction. Thus, we have

$$\hat{g}^{\mathsf{R}_{X,\Omega_{i,j_2}}(\alpha)} = \hat{g}^{\mathsf{Q}(\alpha)/\mathsf{P}_{X,\Omega_{i,j_2}}(\alpha)} = g^{1/\mathsf{P}_{X,\Omega_{i,j_2}}(\alpha)} = g^{1/\theta_{i,j_2}}.$$

This indicates that the simulation by the challenger is perfect. Since the view of $\mathcal{A}$ is unchanged, we have $\Pr[\mathsf{E}_4] = \Pr[\mathsf{E}_5]$.

Game_6 : In this game, we change the way the challenge value $Y_0^\star = \mathsf{Eval}(\mathsf{sk}, X^\star)$ is created when $\mathsf{coin} = 0$. If $\mathsf{coin} = 0$, to generate $Y_0^\star$, it first computes $\xi_{X^\star} \in \mathbb{Z}_p^*$ and $\mathsf{R}_{X^\star}(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ such that $\mathsf{Q}(\mathsf{Z})/\mathsf{P}_{X^\star}(\mathsf{Z}) = \xi_{X^\star}/\mathsf{Z} + \mathsf{R}_{X^\star}(\mathsf{Z})$. Note that such $\xi_{X^\star}$ and $\mathsf{R}_{X^\star}(\mathsf{Z})$ exist by Lemma 6 whenever $\mathsf{T} \subseteq \mathsf{S}(X^\star)$. It then sets

$$Y_0^\star = \left(e\left(\hat{g}, h\right)^{1/\alpha}\right)^{\xi_{X^\star}} \cdot e\left(\hat{g}^{\mathsf{R}_{X^\star}(\alpha)}, h\right)$$

and returns it to $\mathcal{A}$. We claim that this is only a conceptual change. This can be seen by observing that

$$\begin{aligned} e\left(\hat{g}^{1/\alpha}, h\right)^{\xi_{X^\star}} \cdot e\left(\hat{g}^{\mathsf{R}_{X^\star}(\alpha)}, h\right) &= e\left(\hat{g}^{\xi_{X^\star}/\alpha + \mathsf{R}_{X^\star}(\alpha)}, h\right) \\ &= e\left(\hat{g}^{\mathsf{Q}(\alpha)/\mathsf{P}_{X^\star}(\alpha)}, h\right) = e\left(g, h\right)^{1/\mathsf{P}_{X^\star}(\alpha)} \end{aligned}$$

and $\mathsf{P}_{X^\star}(\alpha) = \theta_{\eta,\ell_2}$, where the latter follows from Eq. (13). Since the view of $\mathcal{A}$ is unchanged, we therefore conclude that $\Pr[\mathsf{E}_5] = \Pr[\mathsf{E}_6]$.

Game_7 In this game, we change the challenge value to be a random element in $\mathbb{G}_T$ regardless of whether $\mathsf{coin} = 0$ or $\mathsf{coin} = 1$. Namely, Game_7 challenger sets $Y_0^\star \xleftarrow{\$} \mathbb{G}_T$. In this game, the value coin is independent from the view of $\mathcal{A}$. Therefore, $\Pr[\mathsf{E}_7] = 1/2$.

We claim that $|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_7]|$ is negligible assuming L-DDH with $L = (4\ell+1)\eta+\ell_1$. To show this, we construct an adversary $\mathcal{B}$ against the problem using $\mathcal{A}$, which is described as follows.

$\mathcal{B}$ is given the problem instance $(\Pi, \hat{g}, h, \{\hat{g}^{\alpha^i}\}_{i\in[L]}, \Psi)$ of L-DDH where $\Psi = e(\hat{g}, h)^{1/\alpha}$ or $\Psi \xleftarrow{\$} \mathbb{G}_T$. At any point in the game, $\mathcal{B}$ aborts and sets $\mathsf{coin}' \xleftarrow{\$} \{0,1\}$ if condition (12) is not satisfied. It first sets g and W_{i,j_1} as in Game_4 and returns $\mathsf{vk} = (\Pi, g, h, \{W_{i,j_1}\}_{(i,j_1)\in[\eta]\times[\ell_1]})$ to $\mathcal{A}$. These terms can be efficiently computable from the problem instance because $\log_{\hat{g}} g$ and $\log_{\hat{g}} W_{i,j_1}$ can be written as polynomials in α with degree at most $\eta' - 1 + 4\eta\ell + \ell_1 < L$ and the coefficients of the polynomials can be efficiently computable. When $\mathcal{A}$ makes an evaluation query on input X, it computes (Y, π) as in Game_5 and returns it to $\mathcal{A}$. Again, these terms can be efficiently computable from the problem instance, because the degree of $\mathsf{R}_{X,\Omega_{i,j_2}}(\alpha)$ is at most L and coefficients of them can be efficiently computable. When $\mathcal{A}$ makes the challenge query on

input $X^\star$, $\mathcal{B}$ first picks $\mathsf{coin} \xleftarrow{\$} \{0,1\}$ and returns $Y^\star \xleftarrow{\$} \mathbb{G}$ if $\mathsf{coin} = 1$. Otherwise, it returns

$$Y^\star = \Psi^{\xi_{X^\star}} \cdot e\left(\hat{g}^{\mathsf{R}_{X^\star}(\alpha)}, h\right)$$

to $\mathcal{A}$. Note that $\hat{g}^{\mathsf{R}_{X^\star}(\alpha)}$ can be efficiently computed from the problem instance because the degree of $\mathsf{R}_{X^\star}(\mathsf{Z})$ is at most L. At the end of the game, coin' is defined. Finally, $\mathcal{B}$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.
It can easily be seen that the view of $\mathcal{A}$ corresponds to that of Game_6 if $\Psi = e(\hat{g}, h)^{1/\alpha}$ and Game_7 if $\Psi \xleftarrow{\$} \mathbb{G}_T$. It is clear that the advantage of $\mathcal{B}$ is $|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_7]|$. Assuming L-DDH, we have $|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_7]| = \mathsf{negl}(\lambda)$.

Analysis. From the above, we have

$$\left|\Pr[\mathsf{E}_7] - \frac{1}{2}\right| = \left|\Pr[\mathsf{E}_1] - \frac{1}{2} + \sum_{i=1}^{6} \Pr[\mathsf{E}_{i+1}] - \Pr[\mathsf{E}_i]\right|$$

$$\geq \left|\Pr[\mathsf{E}_1] - \frac{1}{2}\right| - \sum_{i=1}^{6} |\Pr[\mathsf{E}_{i+1}] - \Pr[\mathsf{E}_i]| \geq \tau(\lambda) - \mathsf{negl}(\lambda).$$

for infinitely many λ. Since $\Pr[\mathsf{E}_7] = 1/2$, this implies $\tau(\lambda) \leq \mathsf{negl}(\lambda)$ for infinitely many λ, which is a contradiction. This completes the proof of Theorem 4.

6.2 A Variant with Short Verification Keys

Here, we introduce a variant of our scheme in Sect. 6.1. In the variant, we remove $\{W_{i,j_1} = g^{w_i^{j_1}}\}_{(i,j_1)\in[\eta]\times[2,\ell_1]}$ from vk. Instead, we add these components to π. We do not change the verification algorithm and other parts of the scheme. It is straightforward to see that the correctness and pseudorandomness of the scheme can still be proven. To prove the unique provability, we observe that the only possible strategy to break is to include invalid $\{W_{i,j_1}\}_{(i,j_1)\in[\eta]\times[2,\ell_1]}$ in the proof. This is because if these values are correct, the unique provability of the original scheme immediately implies that of the modified scheme. However, this strategy does not work since the invalid values will be detected at Step 3 of the verification algorithm using $\{W_{i,1} = g^{w_i}\}_{i\in[\eta]}$ in vk. The advantage of the variant is that the size of vk is small. In particular, vk only consists of $\eta + 2$ group elements in this variant, whereas $\eta\ell_1 + 2$ group elements were required in the scheme in Sect. 6.1. Of course, this change increases the size of the proofs π. The number of group elements will become $\eta(\ell_1 + \ell_2 - 1)$ from $\eta\ell_2$ by this modification. To minimize the size of the proofs we choose $\ell_1 = \ell_2 = \sqrt{\ell}$.

7 Comparisons

Here, we compare our proposed schemes with previous schemes.

New Lattice IBE Schemes. In Sect. 5.2, we showed how to construct an IBE scheme from a partitioning function with associating compatible algorithms. We have two ways of instantiating the scheme.

- By using the partitioning function $\mathsf{F}_{\mathsf{MAH}}$ in Sect. 4.3 and the corresponding compatible algorithms, where the latter is given by Lemma 4, we obtain our first IBE scheme. The master public key of the scheme only consists of $\omega(\log^2 \lambda)$ matrices.
- By using the partitioning function $\mathsf{F}_{\mathsf{AFF}}$ in Sect. 4.4 and the corresponding compatible algorithms, where the latter is given by Lemma 5, we obtain our second IBE scheme. The master public key of the scheme is even shorter: It only consists of $\omega(\log \lambda)$ matrices.

Both our schemes achieve the best asymptotic space efficiency (namely, the sizes of the master public keys, ciphertexts, and private keys) among existing IBE schemes that are adaptively secure against *unbounded collusion without sub-exponential security assumptions*. In Table 1 in Sect. 1, we compare our schemes with previous schemes. Note that the scheme by Zhang et al. [ZCZ16] achieves shorter master public key size than ours, but only achieves Q-bounded security. This restriction cannot be removed by just making Q super-polynomial, since the encryption algorithm of the scheme runs in time proportional to Q.

Finally, we note that there are two drawbacks that are common in our schemes. The first drawback is that the encryption algorithm is heavy. Our first scheme requires $\tilde{O}(\lambda)$ times of matrix multiplications for the encryption algorithm. Our second scheme requires even heavier computation. It first computes the description of the "division in $\mathbf{NC}^1$ circuit" [BCH86] and then invokes Barrington's theorem [Bar89] to convert it into a branching program. The second drawback is that we have to rely on the LWE assumption with large (but polynomial) approximation factors to prove the security.

New VRF Schemes. Following [HJ16], we say that a VRF scheme has "all the desired properties" if it has exponential-sized input space and a proof of adaptive security under a non-interactive complexity assumption. Here, we compare our schemes proposed in this paper with previous schemes that satisfy all the desired properties.

- In Sect. 6.1, we proposed new VRF scheme based on $\mathsf{F}_{\mathsf{MAH}}$. The scheme is parametrized by the parameters ℓ_1 and ℓ_2. By setting $\ell_1 = \ell$ and $\ell_2 = 1$, we obtain a new VRF scheme with very short proofs. They only consist of $\omega(\log \lambda)$ group elements.
- In Sect. 6.2, we proposed a variant of the above scheme. The verification keys consist of $\omega(\log \lambda)$ group elements and proofs consist of $\omega(\sqrt{\lambda} \log \lambda)$ group elements.
- In the full version (Appendix C), we proposed a new VRF scheme based on $\mathsf{F}_{\mathsf{AFF}}$. The verification key of the scheme only consists of $\omega(\log \lambda)$ group elements. However, the proof size of the scheme is large.

We refer to Table 2 in Sect. 1 for the overview. From the table, it can be seen that all previous VRF schemes that satisfy all the desired properties [ACF14, BMR10, HW10, Jag15, HJ16] require $O(\lambda)$ group elements for *both* of verification keys and proofs. Our first scheme above significantly improves the size of proofs.

Our second scheme improves both of the sizes of the verification keys and proofs. Compared to our second scheme, only advantage of our third scheme is that the reduction cost is better. Still, we think that our third scheme is also of interest because the construction is quite different from previous schemes.

Acknowledgement. The author is grateful to the members of the study group "Shin-Akarui-Angou-Benkyokai" and anonymous reviewers of Eurocrypt 2017 and Crypto 2017 for insightful comments. In particular, the author would like to thank Shuichi Katsumata for precious comments on the presentation and for pointing out that our initial construction can be slightly simplified. The author also specially thanks Goichiro Hanaoka for helpful comments. This work was supported by JSPS KAKENHI Grant Number 16K16068 and JST CREST Grant No. JPMJCR1688.

References

[ACF09] Abdalla, M., Catalano, D., Fiore, D.: Verifiable random functions from identity-based key encapsulation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 554–571. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_32

[ACF14] Abdalla, M., Catalano, D., Fiore, D.: Verifiable random functions: relations to identity-based key encapsulation and new constructions. J. Cryptology **27**(3), 544–593 (2014)

[ABB10a] Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_28

[ABB10b] Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14623-7_6

[AFL16] Apon, D., Fan, X., Liu, F.: Compact identity based encryption from LWE. In: IACR Cryptology ePrint Archive, 2016:125 (2016)

[BGJS17] Badrinarayanan, S., Goyal, V., Jain, A., Sahai, A.: A note on VRFs from verifiable functional encryption. In: IACR Cryptology ePrint Archive, 2017: 051 (2017)

[Bar89] Mix Barrington, D.A.: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. J. Comput. Syst. Sci. **38**(1), 150–164 (1989)

[BCH86] Beame, P., Cook, S.A., Hoover, H.J.: Log depth circuits for division and related problems. SIAM J. Comput. **15**(4), 994–1003 (1986)

[BR09] Bellare, M., Ristenpart, T.: Simulation without the artificial abort: simplified proof and improved concrete security for Waters' IBE scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_24

[Bit17] Bitansky, N.: Verifiable random functions from non-interactive witness-indistinguishable proofs. In: IACR Cryptology ePrint Archive, 2017: 18 (2017)

[BB04a] Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24676-3_14

[BB04b] Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28628-8_27

[BF01] Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_13

[BGG+14] Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). doi:10.1007/978-3-642-55220-5_30

[BGH07] Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, pp. 647–657 (2007)

[BMR10] Boneh, D., Montgomery, H.W., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: ACM-CCS, pp. 131–140 (2010)

[Boy10] Boyen, X.: Lattice mixing and vanishing trapdoors: a framework for fully secure short signatures and more. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 499–517. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13013-7_29

[BL16] Boyen, X., Li, Q.: Towards tightly secure lattice short signature and Id-based encryption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 404–434. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53890-6_14

[BLP+13] Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC, pp. 575–584 (2013)

[BV16] Brakerski, Z., Vaikuntanathan, V.: Circuit-ABE from LWE: unbounded attributes and semi-adaptive security. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 363–384. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53015-3_13

[CHKP10] Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_27

[CLL+12] Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 122–140. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36334-4_8

[Coc01] Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001). doi:10.1007/3-540-45325-3_32

[Dod03] Dodis, Y.: Efficient construction of (distributed) verifiable random functions. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 1–17. Springer, Heidelberg (2003). doi:10.1007/3-540-36288-6_1

[DY05] Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 416–431. Springer, Heidelberg (2005). doi:10.1007/978-3-540-30580-4_28

[FHPS13] Freire, E.S.V., Hofheinz, D., Paterson, K.G., Striecks, C.: Programmable hash functions in the multilinear setting. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 513–530. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40041-4_28

[Gen06] Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006). doi:10.1007/11761679_27

[GPV08] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

[Gol08] Goldreich, O.: Computational Complexity: A Conceptual Perspective, 1st edn. Cambridge University Press, New York (2008)

[GV15] Gorbunov, S., Vinayagamurthy, D.: Riding on asymmetry: efficient ABE for branching programs. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 550–574. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48797-6_23

[GKV10] Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 395–412. Springer, Heidelberg (2010). doi:10.1007/978-3-642-17373-8_23

[GHKW17] Goyal, R., Hohenberger, S., Koppula, V., Waters, B.: A generic approach to constructing and proving verifiable random functions. In: IACR Cryptology ePrint Archive, 2017:021 (2017)

[HJ16] Hofheinz, D., Jager, T.: Verifiable random functions from standard assumptions. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 336–362. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49096-9_14

[HK08] Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_2

[HW10] Hohenberger, S., Waters, B.: Constructing verifiable random functions with large input spaces. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 656–672. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_33

[Jag15] Jager, T.: Verifiable random functions from weaker assumptions. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 121–143. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46497-7_5

[JR13] Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (2013). doi:10.1007/978-3-642-42033-7_1

[KY16] Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: more compact IBEs from ideal lattices and bilinear maps. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 682–712. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53890-6_23

[LOS+10] Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_4

[LW10] Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010). doi:10.1007/978-3-642-11799-2_27

[Lys02] Lysyanskaya, A.: Unique signatures and verifiable random functions from the DH-DDH separation. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 597–612. Springer, Heidelberg (2002). doi:10.1007/3-540-45708-9_38

[MRV99] Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: FOCS, pp. 191–201 (1999)

[MP12] Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29011-4_41

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC, pp. 333–342 (2009)

[PVW08] Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_31

[Reg05] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93. ACM Press (2005)

[SOK00] Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairings. In: SCIS (2000). (in Japanese)

[Sha85] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). doi:10.1007/3-540-39568-7_5

[SS96] Sipser, M., Spielman, D.A.: Expander codes. IEEE Trans. Inf. Theory **42**(6), 1710–1722 (1996)

[Wat05] Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). doi:10.1007/11426639_7

[Wat09] Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03356-8_36

[Yam16] Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 32–62. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49896-5_2

[Yam17] Yamada, S.: Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In: IACR Cryptology ePrint Archive, 2017: 096 (2017). (Full version of this paper)

[Zém01] Zémor, G.: On expander codes. IEEE Trans. Inf. Theory **47**(2), 835–837 (2001)

[ZCZ16] Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: short signatures and IBEs with small key sizes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 303–332. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53015-3_11