Verifiable Random Functions from Identity-based Key Encapsulation*

Michel Abdalla[1], Dario Catalano[2**], and Dario Fiore[2]

[1] CNRS–LIENS, Ecole Normale Supérieure, Paris, France
michel.abdalla@ens.fr.
[2] Dipartimento di Matematica e Informatica, Università di Catania, Italy
{catalano,fiore}@dmi.unict.it.

Abstract. We propose a methodology to construct verifiable random functions from a class of identity based key encapsulation mechanisms (IB-KEM) that we call VRF suitable. Informally, an IB-KEM is VRF suitable if it provides what we call *unique decapsulation* (i.e. given a ciphertext C produced with respect to an identity ID, all the secret keys corresponding to identity ID', decapsulate to the same value, even if $ID \neq ID'$) and it satisfies an additional property that we call pseudorandom decapsulation. In a nutshell, pseudorandom decapsulation means that if one decapsulate a ciphertext C, produced with respect to an identity ID, using the decryption key corresponding to any other identity ID' the resulting value looks random to a polynomially bounded observer. Interestingly, we show that most known IB-KEMs already achieve pseudorandom decapsulation. Our construction is of interest both from a theoretical and a practical perspective. Indeed, apart from establishing a connection between two seemingly unrelated primitives, our methodology is *direct* in the sense that, in contrast to most previous constructions, it avoids the inefficient Goldreich-Levin hardcore bit transformation.

1 Introduction

Verifiable Random Functions (VRFs for short) were introduced by Micali, Rabin and Vadhan [21]. Informally a VRF is something that behaves like a random function but also allows for efficient verification. More precisely, this means that associated with a secret key sk (the seed), there is a public key pk and a function F such that the following properties are satisfied. First, the function is efficiently computable, given sk, on any input. Second, having only pk and oracle access to the function, the value $F_{pk}(x) = y$ looks random to any polynomially bounded observer who did not query $F_{pk}(x)$ explicitly. Third, a proof $\pi_{pk}(x)$ that $F_{pk}(x) = y$ is efficiently computable knowing sk and efficiently verifiable knowing only pk.

* An extended abstract of this paper appears in the proceedings of EUROCRYPT 2009

** Work partially done while visiting the computer science department at Ecole Normale Supérieure

VRFs turn out to be very useful in a variety of applications essentially because they can be seen as a compact commitment to an exponential number of (pseudo)random bits. To give a few examples, Micali and Reyzin [22] show how to use VRFs to reduce to 3 the number of rounds of resettable zero knowledge proofs in the bare model. Micali and Rivest [23] described a very simple non interactive lottery system used in micropayment schemes, based on VRFs. Jarecki and Shmatikov [17] employed VRFs to build a verifiable transaction escrow scheme that preserves users anonymity while enabling automatic de-escrow. Liskov [18] used VRFs to construct updatable Zero Knowledge databases. In spite of their popularity VRFs are not very well understood objects. In fact, as of today, only four constructions are known, in the standard model [21, 20, 11, 13]. The schemes given in [21, 20] build VRFs in two steps. First they focus on constructing a *Verifiable Unpredictable Function* (VUF). Informally a VUF is a function that is hard to compute but whose produced outputs do not necessarily look random. Next they show how to convert a VUF into a VRF using the Goldreich-Levin [15] theorem to "extract" random bits. Unfortunately, however, the VRF resulting from this transformation is very inefficient and, furthermore, it looses a quite large factor in its exact security reduction. This is because, the transformation involves several steps, all rather inefficient. First one uses the Goldreich Levin theorem [15] to construct a VRF with very small (i.e. slightly super polynomial in the security parameter) input space and output size 1. Next, one iterates the previous step in order to amplify the output size to (roughly) that of the input. Then, using a tree based construction, one iterates the resulting function in order to get a VRF with unrestricted input size and finally one evaluates the so obtained VRF several times in order to get an output size of the required length.

The constructions given in [11, 13], on the other hand, are direct, meaning with this that they manage to construct VRF without having to resort to the Goldreich Levin transform. The VRF presented in [11] is based on a "DDH-like" assumption that the author calls *sum-free decisional Diffie-Hellman* (sf-DDH). This assumption is similar to that one employed by Naor-Reingold [24] to construct PRFs, with the difference that it applies an error correcting code C to the input elements in order to compute the function. The specific properties of the employed encoding allow for producing additional values that can be used as proofs. This construction is more efficient than [21, 20] in the sense that it does not need the expensive Goldreich Levin transform. Still it has some efficiency issues as the size of the produced proofs and keys is linear in the input size. Dodis [11] also adapts this construction to provide a *distributed* VRF, that is a standard VRF which can be computed in a distributed manner.

The scheme proposed by Dodis and Yampolskiy [13], on the other hand, is more attractive, at least from a practical point of view, as it provides a simple implementation of VRFs with short (i.e. constant size) proofs and keys. It is interesting to note that, even though the latter construction is far more efficient than previous work, it builds upon a similar approach. Basically, the construction in [13] works in two steps. First they consider a simple VUF (which is basically Boneh Boyen [3] weakly secure signature scheme) that is secure for

slightly superpolynomially sized input spaces. Next, rather than resorting to the Godreich Levin [15] hardcore bit theorem to convert it into a VRF, they show how to modify the original VUF in order to make it a VRF, under an appropriate decisional assumption.

From the discussion above, it seems clear that, with the possible exception of [11], all known constructions of verifiable random functions, follow similar design criteria. First one builds a suitable VUF and then transforms it into a VRF by either using the Goldreich Levin transform, or via some direct, ad hoc, modifications of the original VUF. The main drawback of this approach is that, once a good enough VUF is found, one has to either be able to make it a VRF directly or accept the fact that the VRF obtained from the Goldreich Levin transform is not going to be a practical one. Thus it seems very natural to ask if there are alternative (and potentially more efficient) ways that allow to construct VRFs directly, without needing to resort to the two steps methodology sketched above.

Our Contribution. In this paper we show how to construct VRF from a class of identity based encryption (IBE) schemes [26] that we call *VRF suitable*. In particular we deal with the related notion of identity-based key encapsulation mechanisms (IB-KEM). Roughly speaking an identity based key encapsulation mechanism is an asymmetric encryption scheme where the public key can be an arbitrary string. Such schemes consists of four algorithms. A Setup algorithm, that generates the system common parameters as well as a master key msk; a key derivation algorithm that uses the master secret key to generate a private key d_{sk} corresponding to an arbitrary public key string ID (the identity); an encapsulation algorithm that creates a ciphertext and a session key using the public key ID and a decapsulation algorithm that recovers the session key from a ciphertext using the corresponding private key.

Informally an IB-KEM is said to be VRF suitable if the following conditions are met. First, the scheme has to provide *unique decapsulation*. This means that, given a ciphertext C produced with respect to some arbitrary identity ID, all the secret keys corresponding to any other identity ID' decapsulate to the same value (i.e. even if $ID' \neq ID$). Second the IB-KEM has to provide what we call *pseudorandom decapsulation*. Very informally, pseudorandom decapsulation means that if C is an encapsulation produced using some identity ID, the "decapsulated" key should look random even if the decapsulation algorithm is executed using the secret key corresponding to any other identity $ID^* \neq ID$. Having a scheme that achieves pseudorandom decapsulation may seem a strong requirement at first. We argue that it is not, as basically all currently known secure (in the standard model) IBE schemes *already* provide pseudorandom decapsulation.

Our result is of interest both from a theoretical and a practical point of view. Indeed, apart from establishing a connection between two seemingly unrelated primitives, our method is direct, in the sense that it allows to build a VRF from a VRF suitable IB-KEM without having to resort to the inefficient Goldreich Levin transform. Moreover, the reduction is tight. This means that, once an efficient VRF suitable IB-KEM is available, this leads to an equally efficient

VRF, with no additional security loss. Furthermore, our constructions immediately allow for efficient distributed VRFs as long as a distributed version of the underlying encryption scheme is available (which is the case for most schemes used in practice).

As a second contribution of this paper, we investigate on the possibility of implementing VRF suitable IB-KEMs. Toward this goal, we first show how to construct a VRF suitable IB-KEM from the Sakai-Kasahara IB-KEM [25]. Interestingly, the resulting VRF turns out to be very similar to the Dodis-Yampolskiy VRF [13], thus showing that the latter construction can actually be seen as a special case of our general methodology. Next, we propose a new implementation of VRF suitable IB-KEM inspired (but more efficient) by Lysyanskaya's VRF [20] (which in turn builds from the Naor Reingold's PRF [24]). The proposed scheme can be proved secure under the assumed intractability, in bilinear groups, of the decisional ℓ-th weak Bilinear Diffie Hellman Inversion problem (decisional ℓ-wBDHI* for short) introduced by Boneh, Boyen and Goh [4]. Interestingly, even though the decisional ℓ-wBDHI* assumption is asymptotic in nature, the ℓ parameter does not need to be too large in order for our security proof to go through. This is because it directly affects only the size of the space of valid identities *but not* the number of adversarial queries allowed in the security reduction[3] (as opposed to most known proofs using asymptotic assumptions). This means that in practice it is enough to assume the decisional ℓ-wBDHI* assumption to hold only for reasonably small values of ℓ (such as $\ell = 160$ or $\ell = 256$).

IBEs and Digital Signatures. Naor pointed out (see [5]) that a fully secure identity based encryption scheme can be transformed into a secure signature scheme as follows. One sets the message space as the set I of valid identities of the IBE. To sign $m \in I$ one executes the key derivation algorithm on input m, and outputs d_{sk} as the signature. A signature on m is verified by encrypting a random M with respect to the identity m, and then by checking that decrypting the resulting ciphertext one gets back M. Thus if one considers an IBE with unique key derivation (i.e. where for each identity one single corresponding decryption key can be computed) the methodology sketched above leads to a secure *unique* digital signature scheme (i.e. a digital signature scheme for which each message admits one single valid signature). Since secure unique signatures are, by definition, verifiable unpredictable functions, at first glance our construction might seem to (somewhat) follow from Naor's remark. We argue that this does not seem to be the case for two reasons. First, our construction does not require the underlying IB-KEM to have unique key derivation, but only to provide unique decapsulation. Clearly the former property implies the latter, but there is no reason to exclude the possibility of constructing a scheme realizing unique decapsulation using a randomized key derivation procedure. Second, a crucial requirement for Naor's transformation to work is that the original IBE is actu-

[3] Here by not affecting the number of adversarial queries we mean that ℓ grows linearly with respect to the identity space but only logarithmically with respect to the number of adversarial queries

ally fully secure. A VRF-suitable IB-KEM, on the other hand, is required to be secure only in a much weaker sense (that we call *weak selective* ID security).

OTHER RELATED WORK. As pointed out above the notion of VRF is related to the notion of unique signatures. Unique signatures were introduced by Goldwasser and Ostrovsky [16] (they called them invariant signatures). The only known constructions of unique signatures in the plain model (i.e. without common parameters or random oracles) are due to Micali, Rabin and Vadhan [21], to Lysyanskaya [20] and to Boneh and Boyen [3]. In the common string model, Goldwasser and Ostrovsky [16] also showed that unique signatures require the same kind of assumptions needed to construct non interactive zero knowledge.

Dodis and Puniya in [12] address the problem of constructing Verifiable Random Permutations from Verifiable Random Functions. They define VRPs as the verifiable analogous of pseudorandom permutations. In particular they point out that the technique of Luby-Rackoff [19] (for constructing PRPs from PRFs) cannot be applied in this case. This is due to the fact that VRP proofs must reveal the VRF outputs and proofs of the intermediate rounds. In their paper they show a construction in which a super-logarithmic number of executions of the Feistel transformation suffices to build a VRP.

More recently Chase and Lysyanskaya [8] introduced the notion of simulatable VRF. Informally a simulatable VRF is a VRF with the additional property that proofs can be simulated, meaning with this that a simulator can fake proofs showing that the value of $F_{sk}(x)$ is y for any y of its choice. Simulatable VRFs can be used to provide a direct transformation from single theorem non interactive zero knowledge to multi theorem NIZK and work in the common reference string model.

2 Preliminaries

Before presenting our results we briefly recall some basic definitions. In what follows we will denote with k a security parameter. The participants to our protocols are modeled as probabilistic Turing machines whose running time is bounded by some polynomial in k. Denote with $\mathbb{N}$ the set of natural numbers and with $\mathbb{R}^+$ the set of positive real numbers. We say that a function $\epsilon : \mathbb{N} \to \mathbb{R}^+$ is negligible if and only if for every polynomial $P(k)$ there exists an $k_0 \in \mathbb{N}$ such that for all $k > k_0$ $\epsilon(k) < 1/P(k)$. If A is a set, then $a \xleftarrow{\$} A$ indicates the process of selecting a at random and uniformly over A (which in particular assumes that A can be sampled efficiently).

VERIFIABLE RANDOM FUNCTIONS Verifiable Random Functions (VRFs for short) were introduced by Micali, Rabin and Vadhan [21]. Intuitively, a VRF is something that behaves like a pseudorandom function, but also allows for a proof of its output correctness. More formally, a VRF is a triplet of algorithms $\mathsf{VRF} = (\mathsf{Gen}, \mathsf{Func}, \mathsf{V})$ providing the following functionalities. The key generation algorithm Gen is a probabilistic algorithm that takes as input the security parameter

and produces a couple of matching public and private keys (vpk, vsk). The deterministic algorithm Func, on input the secret key vsk and the input x to the VRF, computes $(\mathsf{F}_{vsk}(x), \mathsf{Prove}_{vsk}(x))$, where $\mathsf{F}_{vsk}(x)$ is the value of the VRF and $\mathsf{Prove}_{vsk}(x)$ its proof of correctness. The verification algorithm V takes as input (vpk, x, v, π) and outputs a bit indicating whether or not π is a valid proof that $\mathsf{F}_{vsk}(x) = v$.

Let $a : \mathbb{N} \to \mathbb{N} \cup \{*\}$ and $b : \mathbb{N} \to \mathbb{N}$ be functions computable in polynomial time (in k). Moreover we assume that $a(k)$ and $b(k)$ are bounded by a polynomial in k, except if a takes the value $*$ (in this case we simply assume that the VRF can take inputs of arbitrary length). Formally, we say that $\mathsf{VRF} = (\mathsf{Gen}, \mathsf{Func}, \mathsf{V})$ is a VRF of input length $a(k)$ and output length $b(k)$, if the following conditions are met.

Domain Range Correctness For all $x \in \{0,1\}^{a(k)}$ it has to be the case that $\mathsf{F}_{vsk}(x) \in \{0,1\}^{b(k)}$. We require this condition to hold with overwhelming probability (over the choices of (vpk, vsk)).

Provability For all $x \in \{0,1\}^{a(k)}$ if $\mathsf{Prove}_{vsk}(x) = \pi$ and $\mathsf{F}_{vsk}(x) = v$ then $\mathsf{V}(vpk, x, v, \pi) = 1$. We require this condition to hold with overwhelming probability (over the choices of (vpk, vsk) and the coin tosses of V).

Uniqueness No values $(vpk, x, v_1, v_2, \pi_1, \pi_2)$ can satisfy (unless with negligible probability over the coin tosses of V) $\mathsf{V}(vpk, x, v_1, \pi_1) = \mathsf{V}(vpk, x, v_2, \pi_2) = 1$, when $v_1 \neq v_2$.

Pseudorandomness For all probabilistic polynomial time adversaries $A = (A_1, A_2)$ we require that

$$\Pr\left[b' = b \;\middle|\; \begin{array}{l} (vpk, vsk) \xleftarrow{\$} \mathsf{Gen}(1^k);\ (x, \omega) \leftarrow A_1^{\mathsf{Func}(\cdot)}(vpk) \\ b \xleftarrow{\$} \{0,1\};\ v_0 \leftarrow \mathsf{F}_{vsk}(x);\ v_1 \xleftarrow{\$} \{0,1\}^{b(k)} \\ b' \leftarrow A_2^{\mathsf{Func}(\cdot)}(\omega, v_b) \end{array} \right] \leq \frac{1}{2} + \epsilon(k)$$

where the notation $A^{\mathsf{Func}(\cdot)}$ indicates that A has oracle access to the algorithm Func. In order to make this definition sensible, we impose that A cannot query the oracle on input x.

We also introduce a new notion of VRF that we call *selective-VRF*. Informally speaking, a *selective-VRF* is a VRF with a relaxed pseudorandomness property in which the adversary is required to commit ahead of time (i.e. before seeing the public key) to the input value it intends to attack.

More formally we define a *selective-VRF* as a VRF in which the pseudorandomness property is replaced by the following selective variant.

Selective Pseudorandomness For all probabilistic polynomial time adversaries $A = (A_1, A_2)$ we require that

$$\Pr\left[b' = b \;\middle|\; \begin{array}{l} (x, \omega) \leftarrow A_1(); (vpk, vsk) \xleftarrow{\$} \mathsf{Gen}(1^k) \\ b \xleftarrow{\$} \{0,1\}; v_0 \leftarrow \mathsf{F}_{vsk}(x); v_1 \xleftarrow{\$} \{0,1\}^{b(k)} \\ b' \leftarrow A_2^{\mathsf{Func}(\cdot)}(\omega, v_b) \end{array} \right] \leq \frac{1}{2} + \epsilon(k)$$

where the adversary cannot query x to the oracle $\mathsf{Func}(\cdot)$.

ID BASED ENCRYPTION An identity based encryption scheme is a tuple of algorithms $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ providing the following functionality. The trusted authority runs Setup, on input 1^k, to generate a master key pair (mpk, msk). Without loss of generality we assume that the public key mpk specifies a message space $\mathcal{M}$ and a value n (polynomial in the security parameter) indicating the length of each identity. It publishes the master public key mpk and keeps the master secret key msk private. When a user with identity ID wishes to become part of the system, the trusted authority distributor generates a user decryption key $d_{ID} \xleftarrow{\$} \mathsf{KeyDer}(msk, ID)$, and sends this key over a secure and authenticated channel to the user. To send an encrypted message m to the user with identity ID, the sender computes the ciphertext $C \xleftarrow{\$} \mathsf{Enc}(mpk, ID, m)$, which can be decrypted by the user as $m \leftarrow \mathsf{Dec}(d_{ID}, C)$.

Boneh and Franklin [5] formally defined the notion of security for identity based encryption schemes. In particular they defined chosen plaintext security against adaptive chosen identity attack. Intuitively, such a notion, captures the requirement that security should be preserved even when facing an adversary who is allowed to choose the identity it wishes to attack. Later, Canetti, Halevi, and Katz [7] introduced a weaker notion of security in which the adversary is required to commit ahead of time (i.e. before the parameters of the scheme are made public) to the identity it intends to attack. A scheme meeting such a weaker security requirement is said selective ID, chosen plaintext secure IBE (IND-sID-CPA).

Selective identity IBE security is defined as follows: during a preliminary phase the adversary outputs an identity ID^* on which it wants to be challenged. Next, we distinguish the following stages:

Setup The challenger runs the Setup algorithm, gives to the adversary the public parameters of the system, and keeps the master secret key msk for himself.

Phase 1 The adversary is allowed to ask an arbitrary (but polynomially limited) number of key derivation queries. During each of these queries the adversary gives the challenger an identity $ID \neq ID^*$ of its choice and gets back the corresponding private key (the challenger answers such queries by providing to the adversary the output produced by the algorithm KeyDer, when executed on input ID and msk). The queries may be asked adaptively, meaning with this that each query can depend on previously issued ones.

Challenge When Phase 1 is over the adversary outputs two (equal length) messages m_0 and m_1. The challenger picks a random bit b and sets $C \leftarrow \mathsf{Enc}(mpk, ID^*, m_b)$ as the challenge ciphertext. Finally, it sends C to the adversary.

Phase 2 This goes exactly as phase 1.

Guess When phase 2 is over, the adversary outputs a bit b' denoting its guess for the bit b. The adversary wins if $b' = b$

We define the advantage of an adversary A in attacking the encryption scheme as

$$\mathbf{Adv}^{\text{IND-sID-CPA}}(A) = 2\Pr[b = b'] - 1$$

where the probability is taken over the internal coin tosses of the challenger and the adversary.

In this paper we introduce a new notion of security for IBE schemes that we call *weak selective ID security*. More precisely, we define weak selective ID security as the full fledged selective case with the exception that here the challenge identity is chosen by the challenger and given in input to the adversary. Clearly, this notion is weaker with respect to selective ID security as it is easy to see that the latter implies the former.

IDENTITY BASED KEY ENCAPSULATION An identity-based key encapsulation mechanism (IB-KEM) scheme allows a sender and a receiver to agree on a random session key K in such a way that the sender can create K from public parameters and receiver identity and the receiver can recover K using his secret key. This notion, in the context of identity-based encryption, was first formalized by Bentahar et al. [1].

An IB-KEM scheme is defined by four algorithms:

- $\mathsf{Setup}(1^k)$ is a probabilistic algorithm that takes in input a security parameter k and outputs a master public key mpk and a master secret key msk.
- $\mathsf{KeyDer}(msk, ID)$ The key derivation algorithm uses the master secret key to compute a secret key sk_{ID} for identity ID.
- $\mathsf{Encap}(mpk, ID)$ The encapsulation algorithm computes a random session key K and a corresponding ciphertext C encrypted under the identity ID.
- $\mathsf{Decap}(C, sk_{ID})$ allows the possessor of a secret key sk_{ID} to decapsulate C to get back a session key K. We denote by $\mathcal{K}$ the session key space.

For correctness it is required that $\forall k \in \mathbb{N}, ID \in \mathcal{ID}, (C, K) \xleftarrow{\$} \mathsf{Encap}(mpk, ID)$ the following probability holds for all possible $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$:

$$Pr[\mathsf{Decap}(C, \mathsf{KeyDer}(msk, ID)) = K] = 1$$

Here we define the notion of *weak selective ID security* for IB-KEM schemes. Let $\mathcal{IBKEM}$ be a IBE scheme with key encapsulation mechanism. Then $\mathcal{IBKEM}$ is *weakly selective ID secure against adaptively-chosen plaintext attacks* (wsIB-KEM-CPA) if there exists no polynomially bounded adversary $\mathcal{A}$ with non negligible advantage against the Challenger in the following game:

Setup In this phase the challenger selects a challenge identity ID^* (according to an arbitrary distribution) and runs $(mpk, msk) \leftarrow \mathsf{Setup}(1^k)$. Then it computes $(C^*, K^*) = \mathsf{Encap}(mpk, ID^*)$ and flips a binary coin $b \xleftarrow{\$} \{0, 1\}$. Then it sets $\bar{K} = K^*$ if $b = 0$, otherwise it picks a random key $\bar{K} \xleftarrow{\$} \mathcal{K}$. Finally it runs $\mathcal{A}$ on input $(mpk, ID^*, C^*, \bar{K})$ and keeps msk for itself.

Key derivation queries The adversary is allowed to ask key derivation queries for an arbitrary (but polynomial) number of adaptively chosen identities different from ID^*.

Guess In the end of this game $\mathcal{A}$ outputs b' as its guess for b.

The adversary wins if $b' = b$. We formally define the advantage of $\mathcal{A}$ against $\mathcal{IBKEM}$ in the above game as

$$\mathbf{Adv}^{\text{wsIB-KEM-CPA}}_{\mathcal{IBKEM},\mathcal{A}}(k) = \left| Pr[b = b'] - \frac{1}{2} \right|$$

where the probability is taken over the coin tosses of the challenger and the adversary.

VRF-suitable IB-KEMs. Our VRF construction relies on a special class of identity based key encapsulation mechanisms that we call *VRF suitable*. In particular, a VRF suitable IB-KEM is defined by the following algorithms

- $\mathsf{Setup}(1^k)$ is a probabilistic algorithm that takes in input a security parameter k and outputs a master public key mpk and a master secret key msk.
- $\mathsf{KeyDer}(msk, ID)$ The key derivation algorithm uses the master secret key to compute a secret key sk_{ID} for identity ID and some auxiliary information aux_{ID} needed to correctly encapsulate and decapsulate the key.
- $\mathsf{Encap}(mpk, ID, aux_{ID})$ The encapsulation algorithm computes a random session key K, using (mpk, ID, aux_{ID}). Moreover it uses (mpk, ID) to computes a ciphertext C encrypted under the identity ID. Notice that aux_{ID} is required to compute K but not to compute C.
- $\mathsf{Decap}(C, sk_{ID}, aux_{ID})$ allows the possessor of sk_{ID} and aux_{ID} to decapsulate C to get back a session key K. We denote by $\mathcal{K}$ the session key space.

Remark 1. Notice that the description above differs from the one given for basic IB-KEM in that here we require the encapsulation and decapsulation mechanism to use some auxiliary information aux_{ID}, produced by KeyDer, to work correctly. Clearly if one sets $aux_{ID} = \perp$ one goes back to the original description. Thus the new paradigm is slightly more general as it allows to consider encapsulation mechanism where everybody can compute the ciphertext but only those knowing the aux_{ID} information can compute the key. Notice however that aux_{ID} does not allow, by itself, to decapsulate. In some sense, this auxiliary information should be seen as a value that completes the public key (rather than something that completes the secret key) [4]. Even though such a syntax may look totally meaningless in the standard public key scenario, it turns out to be extremely useful (see below) in our context.

Moreover, the IB-KEM has to satisfy the following properties:

1. **Unique decapsulation**. Let ID_0 be any valid identity and C a ciphertext encrypted under ID_0. We require that no valid identity ID can satisfy (unless with negligible probability) $\mathsf{Decap}(C, sk'_{ID}, aux_{ID}{}') \neq \mathsf{Decap}(C, sk''_{ID}, aux_{ID}{}'')$, where $(sk'_{ID}, aux_{ID}{}') \leftarrow \mathsf{KeyDer}(msk, ID)$, $(sk''_{ID}, aux_{ID}{}'') \leftarrow \mathsf{KeyDer}(msk, ID)$

[4] In fact this auxiliary information is not required to be kept secret in our constructions since the adversary can in principle obtain its value for any identity of its choice including the challenge identity (see definition of pseudorandom decapsulation).

2. **Pseudorandom decapsulation** Let C be an encapsulation produced using identity ID_0, we require the session key to look random even if the decapsulation algorithm is executed using the secret key corresponding to any other $\overline{ID}$. More formally, we define the following experiment, for a polynomially bounded adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

Experiment $\mathbf{Exp}^{\text{IB-KEM-RDECAP}}_{\mathcal{IBKEM},\mathcal{A}}(k)$

$(mpk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$
Choose $ID_0 \in \mathcal{ID}$ (according to any arbitrary distribution)
$C^* \xleftarrow{\$} \mathsf{Encap}(mpk, ID_0)$
$(\overline{ID}, st) \xleftarrow{\$} \mathcal{A}_1^{\mathsf{KeyDer}(\cdot)}(mpk, C^*, ID_0)$
$(aux_{\overline{ID}}, sk_{\overline{ID}}) \xleftarrow{\$} \mathsf{KeyDer}(msk, \overline{ID})$
$b \xleftarrow{\$} \{0,1\}$; $K_0 \xleftarrow{\$} \mathsf{Decap}(C^*, sk_{\overline{ID}}, aux_{\overline{ID}})$; $K_1 \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}_2^{\mathsf{KeyDer}(\cdot)}(st, K_b, aux_{\overline{ID}})$
If $b' = b$ then return 1, else return 0

With $\mathcal{A}^{\mathsf{KeyDer}(\cdot)}$ we denote that an algorithm $\mathcal{A}$ has oracle access to the key derivation algorithm. Let $\mathcal{ID}$ denote identity space, i.e. the space from which the adversary (and everybody else) is allowed to choose the identities. In the experiment $\mathbf{Exp}^{\text{IB-KEM-RDECAP}}_{\mathcal{IBKEM},\mathcal{A}}$ we need the following restrictions:

- the identity $\overline{ID}$ output by $\mathcal{A}_1$ should not be asked before;
- $\mathcal{A}_2$ is not allowed to query the oracle on $\overline{ID}$.

We define the advantage of $\mathcal{A}$ in the IB-KEM-RDECAP experiment as

$$\mathbf{Adv}^{\text{IB-KEM-RDECAP}}_{\mathcal{IBKEM},\mathcal{A}}(k) = \left| Pr\left[\mathbf{Exp}^{\text{IB-KEM-RDECAP}}_{\mathcal{IBKEM},\mathcal{A}}(k) = 1\right] - \frac{1}{2}\right|.$$

$\mathcal{IBKEM}$ has pseudorandom decapsulation if for any polynomially bounded adversary $\mathcal{A}$ the advantage $\mathbf{Adv}^{\text{IB-KEM-RDECAP}}_{\mathcal{IBKEM},\mathcal{A}}(k)$ is a negligible function in k.

Remark 2. Notice that the definition above essentially rules out all those schemes where the decapsulation algorithm, when invoked with wrong identity keys, returns $\perp$ or any other error message. In other words, a necessary condition for an IBE to be VRF suitable is that its decapsulation procedure always outputs some random looking key, even when invoked on "wrong" identities.

Remark 3. Requiring that an IB-KEM provides pseudorandom decapsulation might seem a very strong requirement at first. We argue that it is not, at least if the known constructions of IB-KEMs are considered. Indeed, all currently known schemes which are IND-CPA secure (but not IND-CCA secure) in the standard model *already* have this property. In appendix B.1 we prove that the IB-KEM derived from Waters scheme [27] provides pseudorandom decapsulation, while in section 4.1 we prove that the same holds true for the IB-KEM by Sakai and Kasahara [25]. It is easy to generalize such proofs to the case of Boneh Boyen (BB1) [2] and to the case of Boneh Boyen (BB2) [2] and Gentry [14], respectively.

3 The Construction

In this section we show our construction of Verifiable Random Functions from a VRF-suitable IB-KEM $\mathcal{IBKEM} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Encap}, \mathsf{Decap})$. Let $\mathcal{ID}$ be the identity space, $\mathcal{K}$ the session key space and $\mathcal{SK}$ the secret key space. Then we construct $\mathsf{VRF} = (\mathsf{Gen}, \mathsf{Func}, \mathsf{V})$ which models a function from input space $\mathcal{ID}$ to output space $\mathcal{K}$.

$\mathsf{Gen}(1^k)$ runs $(mpk, msk) \leftarrow \mathsf{Setup}(1^k)$, chooses an arbitrary identity $ID_0 \in \mathcal{ID}$ and computes $C_0 \leftarrow \mathsf{Encap}(mpk, ID_0)$. Then it sets $vpk = (mpk, C_0)$ and $vsk = msk$.

$\mathsf{Func}_{vsk}(x)$ computes $\pi_x = (sk_x, aux_x) = \mathsf{KeyDer}(msk, x)$ and $y = \mathsf{Decap}(C_0, \pi_x)$. It returns (y, π_x) where y is the output of the function and π_x is the proof.

$\mathsf{V}(vpk, x, y, \pi_x)$ first checks if π_x is a valid proof for x in the following way. It computes $(C, K) = \mathsf{Encap}(mpk, x, aux_x)$ and checks if $K = \mathsf{Decap}(C, \pi_x)$. Then it checks the validity of y by testing if $\mathsf{Decap}(C_0, \pi_x) = y$. If both the tests are true, then the algorithm returns 1, otherwise it returns 0.

3.1 Security Proof

Now we prove that the proposed construction actually realizes a secure VRF.

Theorem 1. *Assume $\mathcal{IBKEM}$ is a VRF Suitable IB-KEM scheme, as described in section 2 then the construction given above is a verifiable random function.*

Proof. According to the definition given in section 2, we prove that $\mathsf{VRF} = (\mathsf{Gen}, \mathsf{Func}, \mathsf{V})$ is a verifiable random function by showing that it satisfies all the properties. Domain range correctness and provability trivially follow from the IB-KEM scheme correctness. Since $\mathcal{IBKEM}$ has unique decapsulation the uniqueness property is satisfied for construction of VRF. To prove the residual pseudorandomness we assume there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that breaks the residual pseudorandomness of VRF with non-negligible probability $\frac{1}{2} + \epsilon(k)$. Then we can build an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ which has non-negligible advantage $\epsilon(k)$ in the IB-KEM-RDECAP game.

$\mathcal{B}_1$ receives in input from its Challenger the public key mpk and a ciphertext C_0^*. It sets $vpk = (mpk, C_0^*)$ and runs $\mathcal{A}_1(vpk)$. The adversary $\mathcal{A}$ is allowed to make queries to the function oracle $\mathsf{Func}(\cdot)$. $\mathcal{B}$ simulates this oracle in the following way. Given input $x \in \mathcal{ID}$, it queries the key derivation oracle on x. It obtains (sk_x, aux_x) and returns $(\mathsf{Decap}(C_0^*, sk_x), sk_x, aux_x)$ to the adversary. When $\mathcal{A}_1$ outputs an element $\bar{x}$, $\mathcal{B}_1$ gives the same element to its challenger. Thus the challenger produces K^*, which is either the decapsulation of C_0^* with $(sk_{\bar{x}}, aux_{\bar{x}})$ or a random element of $\mathcal{K}$, and gives it to $\mathcal{B}_2$. Then $\mathcal{B}_2$ runs $b' \leftarrow \mathcal{A}_2(st, K^*)$ and outputs b' to the Challenger.

Since the simulation is perfect, if $\mathcal{A}$ outputs $b' = b$ with probability $\frac{1}{2} + \epsilon(k)$, then $\mathcal{B}$'s advantage is exactly $\epsilon(k)$. □

Notice that, when describing the notion of VRF suitable IB-KEM, we did not expect the underlying scheme to meet any additional security requirement. With the following theorem we show that, indeed, a necessary condition, in order for an IB-KEM to be VRF suitable, is that it is secure only in a weak selective sense.

Theorem 2. *Let $\mathcal{IBKEM}$ be a VRF Suitable IB-KEM, then it is also a weakly selective secure IB-KEM (in the sense of the definition given in section 2).*

Proof. Assume, for the sake of contradiction that there exists an adversary A that breaks the weak selective security of the given (VRF suitable) IB-KEM, we show how to use this adversary to construct another adversary B that refutes the pseudorandom decapsulation of the scheme. B starts by receiving (mpk, ID_0, C^*) from his own challenging oracle. Next, B outputs $\overline{ID} = ID_0$ to the oracle, and receives back K_b (which is either the right decapsulation key corresponding to C^* or a random one) and aux_{ID_0}. Now B runs A on input $(mpk, C^*, K_b, ID_0, aux_{ID_0})$. Whenever A asks for a key derivation query, B uses its own oracle to answer the query, in the obvious way. Finally, when A outputs a bit b', B outputs b'. It is easy to see that the advantage of B in breaking the pseudorandom decapsulation is exactly the advantage of A in breaking the weak selective security of the scheme.

4 VRF suitable IB-KEMs

In this section we describe our constructions of Verifiable Random functions from VRF suitable encryption schemes. In particular, in light of the results presented in section 3, we focus on constructing VRF suitable IB-KEM schemes.

We start by describing, in section 4.1, a VRF from the Sakai-Kasahara [25] IB-KEM. Interestingly, the proposed VRF is basically the same as the VRF proposed by Dodis and Yampolskiy [13], thus showing that their construction can be seen as a special case of our general paradigm.

Next we present, in section 4.2, a new construction of VRF suitable IB-KEM from an assumption related to the ℓ-Bilinear Diffie Hellman Inversion assumption (see [2]), that is known as the decisional ℓ-weak Bilinear Diffie Hellman Inversion assumption (decisional ℓ-wBDHI*, following the acronym used in [4]). The decisional ℓ-wBDHI* was introduced by Boneh, Boyen and Goh in [4] and it (informally) states that given $g^b, g^c, g^{b^2}, ..., g^{b^\ell}$, the quantity $e(g,g)^{b^{\ell+1}c}$ should remain indistinguishable from random to any polynomially bounded adversary. The assumption is related to the ℓ bilinear Diffie Hellman Inversion assumption (ℓ-BDHI), in the sense that the former is known to hold in all those groups where the latter holds, but the converse is not known to be true. Interestingly, in order for our construction to work, the ℓ parameter does not need to be too large. This is because it only limits to 2^ℓ the size of the space of valid identities but it does not affect in any other way the number of adversarial queries allowed in the security proof (as in most known proofs using q-type assumptions). Said in a different way, ℓ is required to grow only in a logarithmic way (rather than

linearly) with respect to the number of adversarial queries allowed. This means that it is enough to assume that the ℓ-wBDHI* assumption holds only for rather small values of ℓ (i.e. $\ell = 160$ or $\ell = 256$).

As a final note we mention that, in principle, one could construct a VRF from Boneh Franklin's IBE. Indeed, we prove in appendix B.2, that the KEM version of the scheme is actually a VRF suitable IB-KEM, under the decisional Bilinear Diffie Hellman assumption. This construction, however, is of very limited interest, since the proof holds in the random oracle model.

4.1 Sakai-Kasahara VRF

We briefly recall the KEM version of the Sakai-Kasahara IBE scheme (SK for short) [25]. This scheme relies on the q-decisional Bilinear Diffie-Hellmann Inversion assumption (DBDHI for short), introduced by Boneh and Boyen in [2]. Informally, the DBDHI assumption in bilinear group G of prime order p states that, for every PPT algorithm $\mathcal{A}$ and for a parameter q, $\mathcal{A}$ has negligible probability into distinguishing $e(g,g)^{1/x} \in G_T$ from a random one after seeing $(g, g^x, g^{(x^2)}, \cdots, g^{(x^q)})$. If we suppose that $\mathcal{G}(1^k)$ is a bilinear group generator which takes in input a security parameter k, then (asymptotically) the DBDHI assumption holds for $\mathcal{G}$ if $\mathcal{A}$'s probability of success is negligible in k, for any q polynomial in k.

- $\mathsf{Setup}(1^k)$ runs $\mathcal{G}(1^k)$ to obtain the description of the groups G, G_T and of a bilinear map $e : G \times G \to G_T$. The description of G contains a generator $g \in G$. Then the algorithm picks a random $s \xleftarrow{\$} \mathbb{Z}_p$ and sets $h = g^s$, $mpk = (g, h)$, $msk = s$.
- $\mathsf{KeyDer}(msk, ID)$ We assume $ID \in \mathbb{Z}_p$. The key derivation algorithm constructs the secret key $sk_{ID} = g^{\frac{1}{s+ID}}$.
- $\mathsf{Encap}(mpk, ID)$ The encapsulation algorithm picks a random $t \xleftarrow{\$} \mathbb{Z}_p^\star$ and computes a random session key $K = e(g,g)^t$ and a corresponding ciphertext $C = (g^s g^{ID})^t$.
- $\mathsf{Decap}(C, sk_{ID})$ the decapsulation algorithm uses the secret key sk_{ID} to compute a session key K from a ciphertext C as follows: $K = e(C, sk_{ID})$.

First notice that, assuming $aux_{ID} = \perp \ \forall ID$, the above description fits our syntax of VRF suitable IB-KEMs. Now we prove that the Sakai-Kasahara IB-KEM scheme can be used to construct a VRF (i.e. that it actually provides unique decapsulation and pseudorandom decapsulation). In particular, the resulting VRF can only support superpolynomially-sized (in the security parameter) input space. Notice that all known constructions of VRF made the same assumption.

Theorem 3. *Assuming that the DBDHI assumption holds in a bilinear group G, then the Sakai-Kasahara IB-KEM [25] is a VRF-suitable IB-KEM.*

Proof. We prove the theorem by showing that the IB-KEM scheme presented above has unique decapsulation and satisfies the pseudorandom decapsulation property.

First, let us focus on the key derivation algorithm. If we fix a specific (mpk, msk) pair, it is easy to see that we can obtain only one key for each identity (i.e. no random choices are possible).

Now we prove that SK satisfies pseudorandom decapsulation under the DBDHI assumption. Let $\mathcal{ID} = \{ID_0, \cdots, ID_{q-1}\} \subseteq \mathbb{Z}_p$ be the sets of all possible identities (i.e. the first q elements of $\mathbb{Z}_p$). For sake of contradiction suppose there exists an adversary $\mathcal{A}$ that has non-negligible advantage $\epsilon(k)$ into breaking the pseudorandom decapsulation of SK IB-KEM. Then we are able to build a simulator $\mathcal{B}$ which is able to break the DBDHI assumption with non-negligible advantage $\epsilon(k)/q$.

$\mathcal{B}$ receives in input a tuple $(g, g^x, g^{(x^2)}, \cdots, g^{(x^q)}, Z) \in G^{q+1} \times G_T$ and must output 0 if it believes that $Z = e(g,g)^{1/x}$, or 1 otherwise. $\mathcal{B}$ chooses $ID_0 = 0 \in \mathbb{Z}_p$ and tries to guess the challenge identity by picking a random $ID_k \xleftarrow{\$} \mathcal{ID}$. Let $s = x - ID_k$. Using the binomial theorem it computes $(g, g^s, g^{(s^2)}, \cdots, g^{(s^q)})$. Then $\mathcal{B}$ defines the polynomial $f(z) = \prod_{i=0, i\neq k}^{q-1}(z + ID_i) = \sum_{i=0}^{q-1} z^i\beta_i$. It computes $g' = \prod_{i=0}^{q-1} g^{s^i\beta_i} = g^{f(s)}$ and $h' = \prod_{i=1}^{q-1} g^{s^i\beta_{i-1}} = g^{sf(s)} = (g')^s$. It picks a random $t \xleftarrow{\$} \mathbb{Z}_p$ and sets $C_0 = (g')^t$.We observe that C_0 is a valid ciphertext under identity ID_0 and randomness t/s.Then $\mathcal{B}$ gives $mpk = (g', h')$ and C_0 to the adversary.

When $\mathcal{A}$ asks for the private key of an identity ID_j, if $ID_j = ID_k$ the simulator fails and outputs a random bit b'. Otherwise it is able to compute the secret key in the following way. First it defines the polynomial $f_j(z) = \frac{f(z)}{z+ID_j} = \prod_{i=0, i\neq j,k}^{q-1}(z+ID_i) = \sum_{i=0}^{q-2} z^i\delta_i$. Then it computes $sk_{ID_j} = (g')^{\frac{1}{s+ID_j}} = g^{\frac{f(s)}{s+ID_j}} = g^{f_j(s)} = \prod_{i=0}^{q-2} g^{s^i\delta_i}$ and returns it to $\mathcal{A}$.

In the challenge phase the adversary outputs an identity $\overline{ID}$. If $\overline{ID} \neq ID_k$ $\mathcal{B}$ fails and outputs a random bit b'. Otherwise it can compute a session key $\bar{K}$ as follows. Let $f'(z) = \frac{f(z)}{z+\overline{ID}} - \frac{\gamma}{z+\overline{ID}} = \sum_{i=0}^{q-2} z^i\gamma_i$, where $\gamma \neq 0$ is the remainder of the division of $f(z)$ by $z + \overline{ID}$. First $\mathcal{B}$ computes

$$Z_0 = \left(\prod_{i=0}^{q-1}\prod_{j=0}^{q-2} e(g^{s^i}, g^{s^j})^{\beta_i\gamma_j}\right)\left(\prod_{m=0}^{q-2} e(g, g^{s^m})^{\gamma\gamma_m}\right) = e(g,g)^{\frac{f(s)^2-\gamma^2}{x}}.$$

$\mathcal{B}$ sets $\bar{K} = \left(Z_0 \cdot Z^{\gamma^2}\right)^t$ and gives it to the adversary. $\mathcal{A}$ receives $\bar{K}$ and outputs its guess b'. $\mathcal{B}$ outputs the same b' as its guess for Z. If $Z = e(g,g)^{1/x}$ then $\mathcal{B}$ obtains a session key of the correct form: $\bar{K} = e(g', g')^{\frac{t}{s+\overline{ID}}}$. Otherwise if Z is a random element of G_T, then also $\bar{K}$ will be random.

In conclusion, if we consider the cases in which the simulator fails, its advantage in this experiment is $\epsilon(k)/q$. □

Similarity with the Dodis-Yampolskiy VRF Now we show that the Dodis-Yampolskiy VRF [13] (that we briefly recall in appendix C) can be seen as a special instantiation of the construction given above. Indeed, theorem 3 leads to the following VRF.

$\texttt{Gen}(1^k)$ Runs $\mathcal{G}(1^k)$ to obtain the description of the groups G, G_T and of a bilinear map $e : G \times G \rightarrow G_T$. The description of G contains a generator $g \in G$. Then the algorithm picks random $s, t \xleftarrow{\$} \mathbb{Z}_p$ and sets $h = g^s$, $C_0 = h^t$, $vpk = (g, h, C_0)$, $vsk = s$.

$\texttt{Func}_{vsk}(x)$ Let $\texttt{Func}_{vsk}(x)$=$(F_{vsk}(x), \pi_{vsk}(x))$. One sets $\texttt{Func}_{vsk}(x)$= $e(C_0, sk_x) = e(g, g)^{(st)/(s+x)}$ as the VRF output and $\pi_{vsk}(x) = \mathsf{KeyDer}(x) = g^{1/(s+x)}$ as the proof of correctness.

$\texttt{V}(vpk, x, y, \pi_x)$ To verify whether y was computed correctly, one starts by running the Encap algorithm on input (vpk, x). Encap chooses $\omega \xleftarrow{\$} \mathbb{Z}_p$ and then computes $K \leftarrow e(g, g)^{\omega}$ and $C = (hg^x)^{\omega}$. Then one checks that $K = \mathsf{Decap}(C, \pi_x) = e((g^x \cdot h)^{\omega}, \pi_x)$ and $y = \mathsf{Decap}(C_0, \pi_x) = e(h^t, \pi_x)$.

Thus by setting $t = s^{-1} \bmod p$ and $\omega = 1$, the construction above can be optimized to get exactly the Dodis Yampolskiy VRF.

4.2 The new construction

In this section we propose a new construction of VRF suitable IB-KEM from the (conjectured) computational intractability of the decisional weak ℓ-Bilinear Diffie-Hellman Inversion problem (see below for a formal description). The new scheme is inspired from Lysyanskaya [20] VRF in that the validity of each new auxiliary information aux_{ID} (required to compute the session key) is verified by exploiting the DDH-CDH separation in bilinear groups. The new scheme however is more efficient as it leads to a VRF directly (i.e. rather than having to construct a unique signature scheme first) and does not require error correcting codes.

Decisional weak ℓ-Bilinear Diffie Hellman Inversion Problem [4] The decisional ℓ-wBDHI* problem in G is defined as follows. Let G be a bilinear group of prime order p and g a generator of G. Given $g^b, g^c, g^{b^2}, ..., g^{b^\ell}$, we say that an algorithm $\mathcal{A}$ has advantage ϵ in solving decisional ℓ-wBDHI* in G if

$$\Pr[\mathcal{A}(g^c, g^b, g^{b^2}, ..., g^{b^\ell}, e(g, g)^{b^{\ell+1}c}) = 1] - \Pr[\mathcal{A}(g^c, g^b, g^{b^2}, ..., g^{b^\ell}, e(g, g)^z) = 1] \geq \epsilon$$

where the probability is over the random choices of $b, c, z \in \mathbb{Z}_p^*$

We say that the decisional ℓ-wBDHI* assumption holds in G if no polynomially bounded adversary has advantage better than negligible in solving the decisional ℓ-wBDHI* problem in G.

Remark 4. Cheon showed in [9] an attack against the Strong Diffie-Hellman Assumption and its related problems (among which the DBDHI used to prove

the security of the Dodis-Yampolskiy VRF). This attack reduces the security of a factor $\sqrt{q}$ and applies to the ℓ-wBDHI* as well. However, as it is stated at the beginning of this section, in our construction it is enough to assume that the ℓ-wBDHI* assumption holds only for rather small values of ℓ (i.e. $\ell = 160$ or $\ell = 256$). Thus in our case the security loss is not significant as in Dodis-Yampolskiy's.

The proposed scheme follows

$\mathsf{Setup}(1^k)$ runs $\mathcal{G}(1^k)$ to obtain the description of the groups G, G_T and of a bilinear map $e : G \times G \to G_T$. The description of G contains a generator $g \in G$. Let $\{0,1\}^\ell$ be the space of valid identities. Then the algorithm picks (at random) $a, \alpha_1, \beta_1, \ldots, \alpha_\ell, \beta_\ell \xleftarrow{\$} \mathbb{Z}_p$, sets $g_1 = g^a$, and for $i = 1, \ldots, \ell$ sets $g_{0i} = g^{\beta_i}$ and $g_{1i} = g^{\alpha_i}$. The public parameters are

$$mpk = \left(g, g_1, \{g_{ij}\}_{i=0,1;j=1..\ell}\right)$$

The master secret key is $msk = (a, \{\alpha_i, \beta_i\}_{i=1,..,\ell})$

$\mathsf{KeyDer}(msk, ID)$ We assume $ID = ID_1 \cdots ID_\ell$ where each $ID_i \in \{0,1\}$. The key derivation algorithm constructs the secret key sk_{ID} and the auxiliary information aux_{ID} as follows. Let $h_0 = g$, for $i = 1$ to ℓ one computes

$$h_i = (h_{i-1})^{\alpha_i^{ID_i} \beta_i^{(1-ID_i)}}$$

and sets $aux_{ID} = (h_1, \ldots, h_\ell)$ and $sk_{ID} = h_\ell^a$

$\mathsf{Encap}(mpk, ID, aux_{ID})$ Let $aux_{ID} = (h_1, \ldots, h_\ell)$ computed as above, the encapsulation algorithm picks a random $t \xleftarrow{\$} \mathbb{Z}_p^\star$ and computes a random session key $K = e(g_1, h_\ell)^t$ and a corresponding ciphertext $C = g^t$.

$\mathsf{Decap}(C, sk_{ID}, aux_{ID})$ the decapsulation algorithm uses the secret key sk_{ID} and the auxiliary information aux_{ID} to compute a session key K from a ciphertext C. This is done as follows. First, in order to guarantee the unique decryption property, a check on the validity of the auxiliary information has to be performed. This is done as follows, let $h_0 = g$, for $i = 1, \ldots, \ell$

$$\begin{aligned} \textbf{if } ID_i = 1 \textbf{ check } & e(g, h_i) \stackrel{?}{=} e(g_{1i}, h_{i-1}) \\ \textbf{else} \quad \textbf{check } & e(g, h_i) \stackrel{?}{=} e(g_{0i}, h_{i-1}) \end{aligned}$$

If any of the above checks fails output reject. Second, the key K is computed as $K = e(C, sk_{ID}) = e(g_1, h_\ell)^t$ Notice that, the validity of sk_{ID} can be verified by first encrypting some random message m with respect to the public key (g, g_1, h_ℓ) and then by checking if one can decrypt it correctly using sk_{ID}.

Now we prove the following result

Theorem 4. *Suppose the decisional ℓ-wBDHI* assumption holds in G, then the scheme given above is a secure VRF suitable IB-KEM scheme.*

Proof. Let $\mathcal{ID} = \{0,1\}^\ell$ the identity space. First notice that the scheme fits the syntax of VRF suitable IB-KEMs. We prove the theorem by showing that the scheme has the unique decapsulation property and meets the pseudorandom decapsulation requirement.

Unique Decapsulation We prove this by showing that for a given identity ID the corresponding h_ℓ is uniquely determined as

$$h_\ell = g^{\prod_{i=1}^{\ell} \alpha_i^{ID_i} \beta_i^{1-ID_i}}$$

The proof is by induction on i. First notice that it must be the case $h_1 = g^{\alpha_1^{ID_1} \beta_1^{1-ID_1}}$, as otherwise the check $e(g, h_1) \stackrel{?}{=} e(g_{ID1}, h_0) = e(g^{\alpha_1^{ID_1} \beta_1^{1-ID_1}}, g)$ would fail. Now assume that the statement holds true for any index $j-1 < \ell$, i.e. that $h_{j-1} = g^{\prod_{i=1}^{j-1} \alpha_i^{ID_i} \beta_i^{1-ID_i}}$. We prove that the same holds for j.

$$h_j = h_{j-1}^{\alpha_j^{ID_j} \beta_j^{1-ID_j}} = \left(g^{\prod_{i=1}^{j-1} \alpha_i^{ID_i} \beta_i^{1-ID_i}}\right)^{\alpha_j^{ID_j} \beta_j^{1-ID_j}} = g^{\prod_{i=1}^{j} \alpha_i^{ID_i} \beta_i^{1-ID_i}}$$

Pseudorandom Decapsulation Assume that there is an an adversary $\mathcal{A}$ that breaks the pseudorandom decapsulation of the proposed scheme with advantage ϵ, we show how to build an adversary $\mathcal{B}$ that solves the decisional ℓ-wBDHI* problem with advantage $\epsilon/2^\ell$ and runs in time comparable to that needed by $\mathcal{A}$. $\mathcal{B}$ starts by receiving, from some challenging oracle, the values $(C = g^c, B_1 = g^b, B_2 = g^{b^2}, \ldots B_\ell = g^{b^\ell}$ and a value T that can be either of the form $e(g,g)^{b^{\ell+1}c}$ or of the form $e(g,g)^z$, for random $z \in \mathbb{Z}_p^*$, depending on some random (and hidden) bit d that $\mathcal{B}$ is supposed to guess. First, notice that in the proposed scheme the ciphertext C is independent of specific identities, thus $\mathcal{B}$ can produce it without having to commit to any ID_0. $\mathcal{B}$ chooses $\overline{ID}$ at random as its guess for the challenge identity. Then it sets $g_1 = B_1^a$, for random $a \in \mathbb{Z}_p^*$, chooses at random $\alpha_i, \beta_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, for $i = 1, \ldots, \ell$, and computes for $i = 1, \ldots, \ell$

$$g_{0i} = \begin{cases} B_1^{\beta_i} & if \ \ \overline{ID}_i = 0 \\ g^{\beta_i} & if \ \ \overline{ID}_i = 1 \end{cases} \qquad g_{1i} = \begin{cases} g^{\alpha_i} & if \ \ \overline{ID}_i = 0 \\ B_1^{\alpha_i} & if \ \ \overline{ID}_i = 1 \end{cases}$$

Notice that the public parameters $mpk = \left(g, g_1, \{g_{ij}\}_{i=0,1;j=1..\ell}\right)$ are distributed exactly as those produced by the setup algorithm. The master secret key is implicitly set to $msk = (ab, \{\alpha_i b^{\overline{ID}_i}, \beta_i b^{1-\overline{ID}_i}\}_{i=1,..,\ell})$. Next, $\mathcal{B}$ computes C^* as follows $C^* \leftarrow C = g^c$. Thus, C^* is also correctly distributed. Now $\mathcal{B}$ runs $\mathcal{A}$ on input (mpk, C^*, ID_0), for some randomly chosen identity ID_0. Notice that, from the received inputs, $\mathcal{A}$ gets no information at all about the $\overline{ID}$ chosen by $\mathcal{B}$, thus such a choice will be identical to the challenge identity with probability $1/2^\ell$.

Now we show how $\mathcal{B}$ can answer key derivation queries for identities $ID \neq \overline{ID}$. Since $ID \neq \overline{ID}$ there exists (at least) an index j such that $ID_j \neq \overline{ID}_j$. For such index we have that either $g_{0j} = g^{\beta_j}$ (if $ID_j = 0$) or $g_{1j} = g^{\alpha_j}$ (otherwise). This

means that the h_ℓ corresponding to identity ID will contain the (unknown) b with exponent $\ell - 1$, at most. Let $n < \ell$ denote the number of positions i such that $ID_i = \overline{ID}_i$. $\mathcal{B}$ computes the h_i as follows.

$$h_1 = \begin{cases} g^{\alpha_1^{ID_1}\beta_1^{1-ID_1}} & if \;\; ID_1 \neq \overline{ID}_1 \\ B_1^{\alpha_1^{ID_1}\beta_1^{1-ID_1}} & if \;\; ID_1 = \overline{ID}_1 \end{cases}$$

$$h_2 = \begin{cases} h_1^{\alpha_2^{ID_2}\beta_2^{1-ID_2}} & if \;\; ID_2 \neq \overline{ID}_2 \\ B_1^{\alpha_2^{ID_2}\beta_2^{1-ID_2}} & if \;\; ID_2 = \overline{ID}_2 \wedge ID_1 \neq \overline{ID}_1 \\ B_2^{\alpha_2^{ID_2}\beta_2^{1-ID_2}} & if \;\; ID_2 = \overline{ID}_2 \wedge ID_1 = \overline{ID}_1 \end{cases} \quad \cdots$$

Finally, letting $\omega_{\mathcal{ID}} = \prod_{i=1}^{\ell} \alpha_i^{ID_i}\beta_i^{1-ID_i}$, h_ℓ is computed as $B_n^{\omega_{ID}}$.

The sk_{ID} is set to $B_{n+1}^{a\omega_{ID}}$. Recall that, since $n < \ell$, $\mathcal{B}$ can do this operation using the values received by the challenger. It is easy to check that both the $aux_{ID} = (h_1, \ldots, h_\ell)$ and sk_{ID} are distributed as in the real key derivation algorithm.

Once $\mathcal{A}$ is done with its first phase of key derivation queries it outputs a challenge identity ID^*. If $ID^* \neq \overline{ID}$, $\mathcal{B}$ outputs a random bit and aborts. Otherwise it constructs $K_{\overline{ID}}$ as $T^{a\omega_{\overline{ID}}}$, where $\omega_{\overline{ID}} = \prod_{i=1}^{\ell} \alpha_i^{\overline{ID}_i}\beta_i^{1-\overline{ID}_i}$ and $aux_{\overline{ID}}$ is computed as before. This time however h_ℓ is set to $B_\ell^{\omega_{\overline{ID}}}$, thus $\mathcal{B}$ will not be able to explicitly compute $sk_{\overline{ID}}$. However this is not a problem as $\mathcal{B}$ is not required to do so. Finally $\mathcal{B}$ hands $(K_{\overline{ID}}, sk_{\overline{ID}})$ to $\mathcal{A}$. $\mathcal{A}$ replies back with a guess d' ($d' = 0$ means real, $d' = 1$ means random). $\mathcal{B}$ outputs d' as well. Additional key derivation queries are dealt with as in the first phase. This completes the description of the simulator.

Now notice that if $T = e(g, g)^{b^{\ell+1}c}$, $K_{\overline{ID}}$ is a valid key for the identity $\overline{ID}$. This is because, $K_{\overline{ID}} = e(g_1, h_{\overline{ID}})^c$, where $h_{\overline{ID}}$ is the h_ℓ corresponding to identity $\overline{ID}$. Thus, $h_{\overline{ID}} = g^{b^\ell \omega_{\overline{ID}}}$

$$K_{\overline{ID}} = e(g_1, h_{\overline{ID}})^c = e(g^{ab}, g^{b^\ell \omega_{\overline{ID}}})^c = T^{a\omega_{\overline{ID}}}$$

If on the other hand T is a random value so is $K_{\overline{ID}}$. Thus, by standard calculations one gets that, if $\mathcal{A}$ has advantage ϵ in breaking the pseudorandom decapsulation property of the scheme, $\mathcal{B}$ breaks the decisional ℓ-wBDHI* with advantage $\epsilon/2^\ell$. □

Remark 5. It is interesting to note that if one is interested only into a selective-VRF, then the above construction leads directly to a scheme with large input space. This does not hold for the Dodis-Yampolskiy VRF because in its security proof the simulator has to guess all the queries that the adversary is going to ask even in the weaker selective case.

5 Conclusions

In this paper we introduced a new methodology to construct verifiable random functions (VRF) from a class of identity based key encapsulation schemes that

we call VRF suitable. We showed the applicability of our methods by providing two concrete realizations of the new primitive. The first one leads to a VRF that is very similar to the Dodis-Yampolskiy construction, while the second leads to a new construction. A natural question left open by this research is to find new (potentially better) instantiations of the primitive, possibly ones supporting exponentially large (in the security parameter) identity spaces and provably secure under non interactive assumptions. This would solve the long standing open problem of realizing a secure VRF with unbounded input size.

Acknowledgements

We thank Gregory Neven for collaborating with us at an early stage of this research. We also thank Eike Kiltz and Jonathan Katz for helpful discussions. This work was supported in part by the European Commission through the IST Program under Contract ICT-2007-216646 ECRYPT II and in part by the French National Research Agency through the PACE project.

References

1. Kamel Bentahar, Pooya Farshim, John Malone-Lee, and Nigel P. Smart. Generic constructions of identity-based and certificateless KEMs. *Journal of Cryptology*, 21(2):178–199, April 2008.
2. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.
3. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.
4. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany.
5. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.
6. Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryptionwithout pairings. In *48th FOCS*, pages 647–657, Providence, USA, 2007. IEEE Computer Society Press.
7. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany.
8. Melissa Chase and Anna Lysyanskaya. Simulatable VRFs with applications to multi-theorem NIZK. In Alfred Menezes, editor, *CRYPTO 2007*, LNCS, pages 303–322, Santa Barbara, CA, USA, August 2007. Springer-Verlag, Berlin, Germany.

9. Jung Hee Cheon. Security analysis of the strong Diffie-Hellman problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11, St. Petersburg, Russia, May 28 – June 1, 2006. Springer-Verlag, Berlin, Germany.
10. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *Cryptography and Coding, 8th IMA International Conference*, volume 2260 of *LNCS*, pages 360–363, Cirencester, UK, December 17–19, 2001. Springer-Verlag, Berlin, Germany.
11. Yevgeniy Dodis. Efficient construction of (distributed) verifiable random functions. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 1–17, Miami, USA, January 6–8, 2003. Springer-Verlag, Berlin, Germany.
12. Yevgeniy Dodis and Prashant Puniya. Verifiable random permutations. Cryptology ePrint Archive, Report 2006/078, 2006. `http://eprint.iacr.org/`.
13. Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *PKC 2005*, volume 3386 of *LNCS*, pages 416–431, Les Diablerets, Switzerland, January 23–26, 2005. Springer-Verlag, Berlin, Germany.
14. Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464, St. Petersburg, Russia, May 28 – June 1, 2006. Springer-Verlag, Berlin, Germany.
15. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32, Seattle, Washington, USA, May 15–17, 1989. ACM Press.
16. Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 228–245, Santa Barbara, CA, USA, August 16–20, 1993. Springer-Verlag, Berlin, Germany.
17. Stanislaw Jarecki and Vitaly Shmatikov. Handcuffing big brother: an abuse-resilient transaction escrow scheme. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 590–608, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.
18. Moses Liskov. Updatable zero-knowledge databases. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 174–198, Chennai, India, December 4–8, 2005. Springer-Verlag, Berlin, Germany.
19. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988.
20. Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 597–612, Santa Barbara, CA, USA, August 18–22, 2002. Springer-Verlag, Berlin, Germany.
21. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th FOCS*, pages 120–130, New York, New York, USA, October 17–19, 1999. IEEE Computer Society Press.
22. Silvio Micali and Leonid Reyzin. Soundness in the public-key model. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 542–565, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.
23. Silvio Micali and Ronald L. Rivest. Micropayments revisited. In Bart Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 149–163, San Jose, CA, USA, February 18–22, 2002. Springer-Verlag, Berlin, Germany.
24. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.

25. Ryuichi Sakai and Masao Kasahara. Id based cryptosystems with pairing on elliptic curve. In *2003 Symposium on Cryptography and Information Security – SCIS'2003*, Hamamatsu, Japan, 2003. `http://eprint.iacr.org/2003/054`.
26. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1985. Springer-Verlag, Berlin, Germany.
27. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany.

A Decisional Bilinear Diffie-Hellman assumption

The Bilinear Diffie-Hellman assumption (BDH for short) was first introduced by Boneh and Franklin in [5]. Here we present the decisional version (DBDH) that it was used in a lot of other works (e.g. [27]).

The Decisional Bilinear Diffie-Hellman problem is defined as follows. Let G_1, G_2 be two groups of prime order p with a bilinear map $e : G_1 \times G_1 \leftarrow G_2$ and let g be a generator of G_1. The Challenger chooses random $a, b, c, z \xleftarrow{\$} \mathbb{Z}_p$ and flips a binary coin $b \xleftarrow{\$} \{0, 1\}$. If $b = 0$ it sets $(g, g_1 = g^a, g_2 = g^b, g_3 = g^c, Z = e(g, g)^{abc})$. Otherwise if $b = 1$ it sets $(g, g_1 = g^a, g_2 = g^b, g_3 = g^c, Z = e(g, g)^z)$. The adversary, given in input such tuple, must output a guess b' for b. We define the advantage of an adversary $\mathcal{A}$ in solving the DBDH problem as

$$\mathbf{Adv}(\mathcal{A}) = \left| Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 0] - Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^z) = 0] \right|.$$

Definition 1. *The Decisional Bilinear Diffie-Hellman assumption holds in bilinear groups G_1, G_2 if no polynomially bounded adversary $\mathcal{A}$ obtains non-negligible advantage in the game above.*

B Examples of schemes with Pseudorandom decapsulation

B.1 Waters IBE

In this section we prove that the IBE scheme given by Waters in [27] satisfies the pseudorandom decapsulation property claimed in section 2. More precisely we focus on the KEM version of the scheme ($\mathcal{WKEM}$) and we prove that it has pseudorandom decapsulation assuming the Decisional Bilinear Diffie-Hellman assumption (DBDH for short) and that the scheme is secure against adaptively-chosen plaintext attacks (IB-KEM-CPA).

First we describe Waters IB-KEM scheme ($\mathcal{WKEM}$) [27]. Let G, G_1 be two groups of the same order p equipped with a bilinear map $e : G \times G \to G_1$. We assume that identities are strings of length n.

Setup The setup algorithm picks a random generator $g \xleftarrow{\$} G$ and a random $\alpha \xleftarrow{\$} \mathbb{Z}_p$. It sets $g_1 = g^\alpha$. Then it picks random elements $g_2, u', u_1, \ldots, u_n \xleftarrow{\$} G$. The master public key is $mpk = (g, g_1, g_2, u', \{u_i\}_{i=1}^n)$, while the master secret key is $msk = g_2^\alpha$.

$\mathsf{KeyDer}_{msk}(ID)$ The key derivation algorithm takes in input an identity $ID \in \{0,1\}^n$ and computes its related secret key d_{ID}. It picks a random $r \xleftarrow{\$} \mathbb{Z}_p$ and sets

$$d_{ID} = (d_1, d_2) = \left(g_2^{\alpha} (u' \prod_{i=1}^{n} u_i^{ID_i})^r, g^r \right).$$

$\mathsf{Encap}_{mpk}(ID)$ The encapsulation algorithm takes in input an identity ID and produces a session key K together with a ciphertext C. It picks a random $t \xleftarrow{\$} \mathbb{Z}_p$. Then it sets $C = (C_1, C_2) = (g^t, (u' \prod_{i=1}^{n} u_i^{ID_i})^t)$ and $K = e(g_1, g_2)^t$.

$\mathsf{Decap}_{msk}(C, ID)$ The decapsulation algorithm takes in input a ciphertext C and an identity ID. It uses msk to obtain $d_{ID} = \mathsf{KeyDer}(msk, ID)$ and then it computes $K = \frac{e(d_1, C_1)}{e(d_2, C_2)}$.

Now we prove that such scheme has pseudorandom decapsulation. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary for the pseudorandom decapsulation of $\mathcal{WKEM}$ with advantage $\mathbf{Adv}^{\text{IB-KEM-RDECAP}}_{\mathcal{A}, \mathcal{WKEM}}(k) = \epsilon(k)$. We show how to construct a simulator $\mathcal{B}$ that with advantage at least $\epsilon(k)/2$ breaks either the security of $\mathcal{WKEM}$ or the DBDH assumption (given in appendix A).

Let $ID_0 = 0^n$ be the zero identity chosen by the simulator and let $\overline{ID}$ be the challenge identity outputted by the adversary. We distinguish two cases:

1. $\overline{ID} = ID_0$;
2. $\overline{ID} \neq ID_0$.

Thus $\mathcal{A}$ will output a challenge identity either of type 1 or type 2 with probability at least $1/2$. In the first case the simulator will be able to break the security of the scheme $\mathcal{WKEM}$. In case 2 we can build a simulator that breaks the DBDH assumption.

First, $\mathcal{B}$ flips a binary coin $\beta \xleftarrow{\$} \{0,1\}$. If $\beta = 0$ $\mathcal{B}$ guesses that $\mathcal{A}$ will output a challenge identity of type 1. Otherwise, if $\beta = 1$ it guesses that $\mathcal{A}$ will output $\overline{ID} \neq ID_0$.

$\mathcal{B}$ runs a different simulation for each case. It is easy to verify that the two simulations are perfectly indistinguishable from $\mathcal{A}$'s perspective.

Simulation 0 If $\beta = 0$ $\mathcal{B}$ acts as an adversary for the IB-KEM-CPA security of $\mathcal{WKEM}$. $\mathcal{B}$ receives mpk from its challenger and returns ID_0 as challenge identity. Then it gets back a ciphertext C_0 and a session key K^* and it runs $\mathcal{A}_1$ on input (mpk, C_0). The adversary issues key derivation queries until it outputs a challenge identity $\overline{ID}$. $\mathcal{B}$ answers such queries by using its key derivation oracle. If $\mathcal{A}$ asks for the private key of identity ID_0 or it outputs $\overline{ID} \neq ID_0$ then the simulator fails and outputs a random bit. Otherwise if $\mathcal{B}$ does not fail (and thus $\overline{ID} = ID_0$), it runs $b \leftarrow \mathcal{A}_2(K^*)$ and outputs b. Notice that in this case breaking the pseudorandom decapsulation is equivalent to breaking the IB-KEM-CPA security.

Let fail denote the event that the simulator fails. Then the advantage of $\mathcal{B}$ against the security of $\mathcal{WKEM}$ is

$$\mathbf{Adv}^{\text{IB-KEM-CPA}}_{\mathcal{B}, \mathcal{WKEM}}(k) = \mathbf{Adv}^{\text{IB-KEM-RDECAP}}_{\mathcal{A}, \mathcal{WKEM}}(k) Pr[\overline{\mathsf{fail}}].$$

SIMULATION 1 In this case $\mathcal{B}$ acts as an adversary for the Decisional Bilinear Diffie-Hellmann assumption (DBDH). $\mathcal{B}$ receives in input a DBDH tuple (g, g^a, g^b, g^c, Z) from its challenger.

Then $\mathcal{B}$ constructs the public key for $\mathcal{WKEM}$ as follows. It chooses random $\alpha, w, w_1, \ldots, w_n \xleftarrow{\$} \mathbb{Z}_p$. It chooses $g_2 \in G$. Then it sets $u' = g^w, u_i = (g^b)^{w_i}, g_1 = g^\alpha$. The public key is $mpk = (g, g_1, g_2, u', u_1, \ldots, u_n)$. The master secret key is $msk = \alpha$. $\mathcal{B}$ also computes the ciphertext $C_0 = (g^c, g^{cu})$. Note that it is a correctly distributed ciphertext for the identity ID_0 using randomness c.

$\mathcal{B}$ runs $\mathcal{A}_1$ on input (mpk, C_0) and it answers key derivation queries using α until the adversary outputs the challenge identity $\overline{ID}$. If $\overline{ID} = ID_0$ then $\mathcal{B}$ aborts and outputs a random $b \in \{0,1\}$. Otherwise let $x = \sum_{i=1}^{n} w_i \overline{ID}_i$ and let $d_{\overline{ID}} = (d_1, d_2)$ be the secret key for the identity $\overline{ID}$. More precisely we have $d_1 = g_2^\alpha g^{ua} \prod_{i=1}^{n} g^{abw_i \overline{ID}_i}$ and $d_2 = g^a$. Note that $d_{\overline{ID}}$ is a correctly distributed secret key for the identity $\overline{ID}$ with $r = a$ even if we are not able to compute d_1. Then $\mathcal{B}$ computes the session key $\bar{K} = e(g_2^\alpha g^{au}, g^c) Z^x / e(g^a, g^{cu})$ and gives it to $\mathcal{A}_2$. In the end of the game $\mathcal{A}_2$ will output a bit b as its guess for $\bar{K}$. Then $\mathcal{B}$ will output the same b as its guess for Z.

We note that if $Z = e(g,g)^{abc}$, $\bar{K}$ has the right form

$$\bar{K} = \frac{e(g_2^\alpha g^{au}, g^c) e(g^{abx}, g^c)}{e(d_2, C_{0,2})} = \frac{e(g_2^\alpha g^{au} g^{abx}, g^c)}{e(d_2, C_{0,2})} = \frac{e(d_1, C_{0,1})}{e(d_2, C_{0,2})}.$$

Otherwise if Z is random, then so is $\bar{K}$. Let fail denote the event that the simulator fails. If $\mathcal{B}$ does not fail the simulation is perfect, thus its advantage against DBDH is $\mathbf{Adv}_{\mathcal{B}}^{DBDH}(k) = \mathbf{Adv}_{\mathcal{A}, \mathcal{WKEM}}^{\text{IB-KEM-RDECAP}}(k) Pr[\overline{\mathsf{fail}}]$.

In conclusion, since the two simulations are indistinguishable $\mathcal{A}$ outputs a challenge identity either of type 1 or type 2 with probability at least $1/2$, then we have that in either one of the two simulations $\mathcal{B}$ does not fail with probability at least $1/2$. Thus $\mathcal{B}$ breaks either the security of $\mathcal{WKEM}$ or DBDH with advantage at least $\epsilon(k)/2$.

B.2 Boneh-Franklin IBE

Let us consider the KEM version of the Boneh-Franklin Identity Based Encryption Scheme [5] ($\mathcal{BFKEM}$). We show the construction of this scheme. We point out that it derives from the scheme that Boneh and Franklin call $\mathsf{BasicIdent}$.

Let G_1 be a bilinear group of order p with a bilinear map $e : G_1 \times G_1 \to G_2$. Let $H_1 : \{0,1\}^* \to G_1^*$ be a hash function.

Setup It picks a random generator $P \in G_1$ and a random $s \xleftarrow{\$} \mathbb{Z}_p$. It sets $P_{pub} = P^s$. The master public key is $mpk = (P, P_{pub}, H_1)$. The master secret is $msk = s$.

$\mathsf{KeyDer}_{msk}(ID)$ The key derivation algorithm takes in input an identity $ID \in \{0,1\}^*$ and produces its related secret key d_{ID}. First it computes $Q_{ID} = H_1(ID)$ and then it sets $d_{ID} = Q_{ID}^s$.

$\mathsf{Encap}_{mpk}(ID)$ The encapsulation algorithm takes in input an identity $ID \in \{0,1\}^*$ and returns a session key K together with a ciphertext C. The algorithm picks a random $r \xleftarrow{\$} \mathbb{Z}_p^*$ and computes $Q_{ID} = H_1(ID)$. It sets $K = e(Q_{ID}, P_{pub})^r, C = P^r$.

$\mathsf{Decap}_{msk}(C, ID)$ The decapsulation algorithm takes in input a ciphertext C and an identity ID. First it uses msk to obtain $d_{ID} = \mathsf{KeyDer}(msk, ID)$ and then it returns $K = e(d_{ID}, C)$.

In this section we show that such scheme satisfies the pseudorandom decapsulation property claimed in section 2 assuming that the Decisional Bilinear Diffie-Hellmann assumption (DBDH) is hard. The proof is inspired to that one given in [5] (Lemma 4.2).

More formally, let k be the security parameter, $\mathcal{A}$ be an adversary for the pseudorandom decapsulation (IB-KEM-RDECAP) of $\mathcal{BFKEM}$ with advantage $\mathbf{Adv}^{\text{IB-KEM-RDECAP}}_{\mathcal{A},\mathcal{BFKEM}}(k) = \epsilon(k)$. Let $H_1 : \{0,1\}^* \to G_1^*$ be a random oracle. Thus, assuming q_E to be an upper bound to the number of key derivations asked to the oracle, we show how to construct a simulator $\mathcal{B}$ that uses $\mathcal{A}$ to solve DBDH with advantage at least $\epsilon(k)/e(1+q_E)$.

The simulator receives from its challenger a DBDH tuple $(P, P_1, P_2, P_3, Z) = (P, P^a, P^b, P^c, Z)$ where $P \in G_1$ is a random generator and a, b, c are taken at random in $\mathbb{Z}_p$. The challenger flips a binary coin $\nu \in \{0,1\}$. If $\nu = 0$ it sets $Z = e(P,P)^{abc}$, otherwise it picks a random $Z \xleftarrow{\$} G_2$. $\mathcal{B}$ must output 0 if it believes that $Z = e(P,P)^{abc}$, 1 otherwise.

First, $\mathcal{B}$ constructs the public key for the $\mathcal{BFKEM}$ scheme by setting $mpk = (p, G_1, G_2, P, P_{pub} = P_1, H_1, H_2)$. It also sets $C_0 = P_3$ and gives (mpk, C_0) to $\mathcal{A}_1$.[5] The simulator controls the random oracle H_1 as follows. It maintains a list H_1^{list} of tuples $(ID_i, Q_i, \beta_i, coin_i)$. When the adversary queries H_1 on input ID, $\mathcal{B}$ checks if $ID \in H_1^{list}$. If this is the case and $(ID, Q, \beta, coin)$ is the correlated tuple, then it outputs $H_1(ID) = Q$. Otherwise $\mathcal{B}$ flips a random $coin \in \{0,1\}$ such that $Pr[coin = 0] = \delta$. It picks a random $\beta \xleftarrow{\$} \mathbb{Z}_p^*$. If $coin = 0$ it sets $Q = P^\beta$, otherwise it sets $Q = P_2^\beta$.

The adversary is allowed to make key derivation queries to the oracle $\mathsf{KeyDer}(\cdot)$. $\mathcal{B}$ answers such queries in the following way. On input ID_i it queries $H_1(ID_i)$. Let $(ID_i, Q_i, \beta_i, coin_i)$ the tuple in H_1^{list}. If $coin_i = 1$ the simulator fails and outputs a random guess $\nu' \in \{0,1\}$. Otherwise $\mathcal{B}$ sets $d_{ID} = P_1^{\beta_i}$. In the end of this phase $\mathcal{A}_1$ outputs a challenge identity $\overline{ID}$. $\mathcal{B}$ runs $H_1(\overline{ID})$ to obtain $(\overline{ID}, \overline{Q}, \overline{\beta}, \overline{coin})$. If $\overline{coin} = 0$ $\mathcal{B}$ fails and output a random guess. Otherwise it sets $\overline{K} = Z^{\overline{\beta}}$ and gives $\overline{K}$ to $\mathcal{A}_2$. When the adversary outputs its guess ν' for $\overline{K}$ $\mathcal{B}$ outputs the same value to its challenger.

We observe that if $Z = e(P,P)^{abc}$ then $\overline{K} = e(P,P)^{abc\overline{\beta}} = e(d_{\overline{ID}}, C_0)$ is a correct session key obtained by decapsulating C_0 with identity $\overline{ID}$. Otherwise if Z is random, also $\overline{K}$ will be random. Thus if $\mathcal{B}$ does not abort the simulation is

[5] We observe that in this case it is not necessary to explicitly choose an identity ID_0.

perfect. In general we have:

$$\mathbf{Adv}_{\mathcal{B}}^{DBDH}(k) = \mathbf{Adv}_{\mathcal{A},\mathcal{BFKEM}}^{\text{IB-KEM-RDECAP}}(k) Pr[\overline{abort}].$$

If q_E is the number of key derivation queries issued to the oracle, then the probability that $\mathcal{B}$ does not abort is $\delta^{q_E}(1-\delta)$. This value is maximized with $\delta_{opt} = 1 - \frac{1}{q_E+1}$ for which we obtain $Pr[\overline{abort}] \geq \frac{1}{e(q_E+1)}$.

C The VRF by Dodis and Yampolskiy

In this section we describe the VRF by Dodis and Yampolskiy [13].

- $\texttt{Gen}(1^k)$ Runs $\mathcal{G}(1^k)$ to obtain the description of the groups G, G_T and of a bilinear map $e : G \times G \to G_T$. The description of G contains a generator $g \in G$. Then the algorithm picks a random $s \xleftarrow{\$} \mathbb{Z}_p$ and sets $h = g^s$, $vpk = (g, h)$, $vsk = s$.
- $\texttt{Func}_{vsk}(x)$ Let $\texttt{Func}_{vsk}(x)$=$(F_{vsk}(x), \pi_{vsk}(x))$. One sets $\texttt{Func}_{vsk}(x)$=$e(g,g)^{1/(s+x)}$ as the VRF output and $\pi_{vsk}(x) = g^{1/(s+x)}$ as the proof of correctness.
- $\texttt{V}(vpk, x, y, \pi_x)$ To verify if y was computed correctly, one checks that $e(g^x \cdot h, \pi_x) = e(g,g)$ and $y = e(g, \pi_x)$.