# On the (Un)biasability of Existing Verifiable Random Functions

Davide Carnemolla[1], Dario Catalano[1], Valentina Frasca[1], Emanuele Giunta[2]

[1] Dipartimento di Matematica e Informatica, Università di Catania, Italy.
davide.carnemolla@phd.unict.it dario.catalano@unict.it
valentina.frasca@phd.unict.it
[2] ETH Zurich, Zurich, Switzerland.
emanuele.giunta@inf.ethz.ch

**Abstract.** Verifiable Random Functions (VRFs) play a fundamental role in modern blockchain designs because of their applications in leader election protocols. In such contexts, however, the original definition by Micali, Rabin and Vadhan (FOCS 99), falls short at guaranteeing fairness when keys are sampled maliciously.
The elegant notion of *unbiasable* VRF, recently proposed by Giunta and Stewart (Eurocrypt 24), addresses these concerns while remaining simple to state and easy to realize, at least in the random oracle model. Achieving unbiasability in the standard model is a different story, though: all known constructions rely on compilers that invariably reduce the efficiency of the VRF from which one starts.
In this paper, we look at the unbiasability of existing VRFs in the standard model. Our findings are mostly negative; we show that, essentially, all known constructions are not natively unbiasable. We do so by showing classes of attacks that (almost) completely cover the set of existing VRF constructions.
On the positive side, we show that some concrete schemes (and notably the well-known Dodis-Yampolskiy VRF) can be modified to achieve meaningful notions of unbiasability, while retaining their original efficiency.

**Keywords:** Verifiable Random Functions · Unbiasability · Standard Model

## 1 Introduction

Verifiable (pseudo) random functions, introduced by Micali *et. al* in [MRV99], allow the owner of a secret key $\mathsf{sk}$ to compute a pseudorandom random function $F$ on input $x$ along with a proof $\pi$ guaranteeing correctness of the output, and that anyone can verify against a public verification key $\mathsf{vk}$. The main properties a VRF is required to possess are *correctness*, *uniqueness*, and *pseudorandomness*. The first means that honestly generated proofs must be accepted. The second dictates that it is *impossible* to prove the function to have two distinct output values on the same input. Finally, pseudorandomness requires that the output of $F$ on a point for which an adversary has not yet seen a proof is indistinguishable from a truly random value.

Beyond being an intrinsically interesting object to study, VRFs are useful in a variety of applications ranging from lotteries [MR02, LBM20] to round-efficient resettable zero knowledge protocols [MR01]. Over the last few years, renewed interest in this primitive has been sparked by its relevance in Proof of Stake (PoS) blockchains, due to their central role in efficient secret leader election protocols (e.g. [CM19, KRDO17, DGKR18, BASV23]). In this context, however, the three defining properties of VRFs turn out to be insufficient to prove security. The issue is that the original definition from [MRV99] guarantees pseudo-randomness only when the keys are *honestly generated*. For *adversarially generated* ones instead, VRFs offer no guarantee about the output distribution. This is very problematic for VRF-based leader election protocols. There, the approach is informally to have each party generate its own verification key, and then elect as leader the one with the lowest VRF output on some publicly known (random) input. Being able to craft a key whose output distribution is skewed toward lower values leads to an unfair advantage in winning elections, thus collecting higher rewards.

The issue was first noticed in [DGKR18, EKS$^+$21] and finally assessed in [GS24] where a simple and elegant notion of *unbiasability* was proposed[3]. Informally, unbiasability states that it should be unfeasible to come up with VRF keys whose behavior on *random* inputs deviates significantly from that of honestly crafted ones. Along with the definition, two main positive results were presented in [GS24]. First, they observed that the folklore construction of hashing the (salted) output of a verifiable unpredictable function satisfies unbiasability in the random oracle model. Next, they provide a construction in the standard model (that, interestingly, does not need trusted setup). This latter construction is actually a compiler that turns any standard VRF into an unbiasable one. The compiler is non-generic, though, as it relies on DDH. Very recently, Brandt [Bra25] improved on [GS24] by presenting a simpler and more elegant compiler that only relies on generic primitives (such as injective one-way functions and collision-resistant hashing).

Both these compilers, however, while theoretically interesting, add a significant burden to the underlying VRF design. Specifically:

- the compiler proposed in [GS24] requires running an Encode procedure and computing a Pedersen Hash, as well as running two evaluations of the underlying VRF; moreover, the solution proposed by the authors for the Encode procedure requires one evaluation of a PRF and three evaluations of some efficiently computable and injective map;
- the compiler proposed in [Bra25] requires, beyond the evaluation algorithm of the underlying VRF, exactly one invocation each of an injective one-way function, a collision-resistant hash, a hardcore function for injective one-way function[4], and a PRF.

[3] In fact Giunta and Stewart propose three, increasingly more stringent, notions of unbiasability. We will say more about this later on.

[4] Informally, we say that $f$ is a hardcore function for an injective one-way function if its output on input $x$ is computationally indistinguishable from random even if the output of the injective one-way function on the same point is known.

Although the exact cost varies with the specific instantiation, it is evident that these computations introduce overhead compared to performing a single VRF evaluation.

Moreover, for most number-theoretic constructions in the standard model, they break the simple algebraic verification procedure, which is among the advantages of not relying on random oracle/hash functions. Given the importance of unbiasability in VRF applications, and the lack of "simple" standard model construction, we consider the following question:

*Is any of the existing VRF constructions in the standard model natively unbiasable?*

In this paper, we address the question by providing a systematic analysis of existing VRFs. Our findings are prevalently negative; we show that no pairing-based VRF achieves the stronger unbiasability notion in [GS24]. Even worse, most constructions do not achieve even the most basic notion of unbiasability (named *weak unbiasability* in [GS24]). In this direction, we propose several general attacks that completely cover the set of existing VRF constructions in the standard model. Our attacks suggest that unbiasability, while easy to achieve in the random oracle model, does not seem to come for free in the standard model. Moreover, the nice algebraic structure of existing construction appears to be, for the most part, a problem rather than a benefit.

In spite of this, we provide some evidence that the goal of realizing a simple algebraic and unbiasable construction in the standard model might not be impossible to achieve. We show that a large domain variant of the well-known Dodis-Yampolskiy VRF achieves (some form of) unbiasability[5]. The catch here is, unfortunately, that we only manage to prove pseudorandomness for this variant in the generic group model.

In addition, we show that constructions based on *non-interactive witness indistinguishable proofs* (NIWIs), *non-interactive commitments*, and *constrained pseudorandom functions*(constrained-PRF), proposed in [Bit17, GHKW17], can achieve weak unbiasability when instantiated with a specific *injective* constrained-PRF, namely the one from [BKW17].

**Our attacks, more in detail.** As mentioned above, the main result of this paper is to put forward several attacks to the unbiasability of existing VRF constructions in the standard model. In the following, we introduce each of them at a high level. We summarize them and the impact on affected schemes in Figure 1.

*Output Guessing.* Before explaining the attack, let us first recall the notion of *weak unbiasability* [EKS+21, GS24]. Informally, this dictates that an adversary who maliciously produced a VRF key cannot predict the result significantly better than guessing a random point in the co-domain.

[5] The so-called *Unbiasability on independent points* [GS24].

As a warm-up, our first attack shows how to break weak-unbiasability for all those VRFs in which the image (i.e. the set of values with an actual preimage) is both polynomially small, and sparse in the co-domain (i.e. a super-set containing the image). This property is satisfied by several well-known VRFs, such as [MRV99, DY05, BMR10], which are therefore vulnerable to this attack. The attack is almost trivial: the adversary samples $(\mathsf{vk}, \mathsf{sk})$ honestly, and then guesses the output by guessing the preimage. Then, the assumption on the VRF above implies that our adversary guesses correctly with significant probability, whereas guessing from the co-domain only works with negligible probability.

While this first attack is conceptually trivial, we would like to stress that it is *not* just a technicality. Indeed, in the VRFs cited above, the range has to be sparse over a much larger co-domain because the range itself does not allow for efficient membership testing and sampling, thus making it impossible to directly fix the issue by *restricting* the codomain.

*Powers of alpha attack.* Having established this fact, we turn our attention to more interesting cases, namely confuting the unbiasability of VRFs with large domain and range. A common blueprint of our attacks will be to generate adversarially one malformed verification key whose associated function has only polynomially small range. This allows then to predict the output on a random point with non-negligible probability.

Our first attack of this form, which we call *powers of $\alpha$ attack*, applies to Naor-Reingold [NR97] like VRFs (e.g. [Lys02, ACF09, HW10, ACF14]). As all these constructions build on the same strategy of adding a verification mechanism on top of the Naor-Reingold PRF, we showcase first our attack on a simplified and idealized construction. First, our toy VRF chooses a group $\mathbb{G}$ of prime order $p$ and generator $G$, samples $n+1$ random group elements[6] $[a_0], [a_1], \ldots, [a_n]$ (i.e. the Naor-Raingold PRF key in the exponent), and publish them in the verification key. Next, to evaluate the VRF on input $x \in \{0,1\}^n$, it returns

$$Y = \left[a_0 \cdot \prod_{i=1}^{n} a_i^{x_i}\right]$$

along with a proof $\pi$. In concrete constructions, a pairing group $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ would be used instead, and verifiability is typically obtained via a step ladder approach, as originally proposed by Lysyanskaya in [Lys02]: given all the intermediate $[a_0 \prod_{i=1}^{j} a_i^{m_i}]$, for $t \leq n$, one uses the verification key and the bilinear map to check $Y$ was computed correctly.

Our powers of $\alpha$ attack consists in choosing all the $a_i$ $(i > 0)$ above as powers of the a common based $\alpha$, i.e. $a_i = \alpha^i$, where $\alpha$ is a random element of $\mathbb{Z}_p$. In this way, for any input $x$, the resulting output will have an exponent of the form

$$a_0 \cdot \prod_{i=0}^{n} a_i^{x_i} \;=\; a_0 \cdot \alpha^{x_1 + 2x_2 + \ldots + nx_n}$$

And in particular, the output only depends on $x_1 + 2x_2 + \ldots + nx_n \in \{0, \ldots, n^2/2\}$, making the malformed key's range polynomially small as desired.

[6] We adopt the usual notation for scalar multiplications $[a] = a \cdot G$.

At first glance, a natural countermeasure to prevent the attack might consist in modifying the verification procedure of the VRF. For instance, in symmetric pairing groups, one might use the pairing, together with public information, to check whether the evaluation key follows the right distribution or not. Unfortunately, however, we give evidence that such a strategy cannot work in general. In particular, for the case of the VRF by Hohenberger and Waters [HW10], we prove that our powers of $\alpha$ attack can be generalized to use exponents $z_1, ..., z_n$ (as opposed to $1, \ldots, n$) so that no efficient verification procedure extending the original one can detect the attack with high probability. More precisely, our proof relies on an ad-hoc $Q$-type assumption in symmetric pairing groups, which we justify in the generic bilinear group model. What we show then is that any extended (efficient) verification procedure for the VRF rejecting malformed key/proofs either leads to an efficient solver for the assumption, or rejects almost always also honestly generated keys. We also show how our attack generalizes to constructions whose keys take the form of square, invertible matrices, as in [HJ16].

*Related Key Attack.* The class of attacks described so far targets the simpler notion of weak unbiasability of essentially all Naor-Reingold VRFs. However, several algebraic constructions do not rely on Naor-Reingold as their underlying PRF. This is for instance the case of [BMR10, §7] and related constructions. For all these schemes we provide an attack against "full" unbiasability.

Although for a technical discussion on the definition we defer to Section 2.2, at a high level the notion requires no adversary to be able to produce *several* keys $\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ whose output of given set of input biases a given predicate. The key observation, which applies to almost *all* the pairing-based construction (with the exception of [DY05]), is that when two keys are generated as multiples of each-other, the output is usually also offset by a (possibly different) constant factor, meaning the two keys produce related VRF values.

To see this in action, the VRFs in [HJ16, Yam17, Koh19] and [BMR10, §7]can be seen to produce output of the form $e\left(F_{\mathsf{sk}}(x), [u]\right)$ for some private key function $F_{\mathsf{sk}}$ and $[u]$ a (vector of) group element(s). Then, generating two keys respectively with $[u], [\alpha u]$ means that on a point $x$ the two functions generate values $Y_1, Y_2$ such that $Y_1 = \alpha Y_2$, which breaks unbiasability.

*Specialized Attacks.* Finally, although already affected by our related keys attack, we eventually provide specialized attacks against the unbiasability of several constructions. Specifically, this is the case for [GHKW17, Bit17, Ros18, Koh19], for which our power of $\alpha$ methodology is not directly applicable. We defer a more detailed discussion to the respective sections.

*Positive Results.* As a final contribution, we provide promising directions towards the realization of a direct, unbiasable, VRF[7]. As pointed out by Giunta and Stewart, an intermediate notion of unbiasability comes essentially for free

[7] By direct here we mean not relying on existing compilers [GS24, Bra25].

when one starts from a VRF which is also a *certified bijection* over its input space $\mathcal{X}$ and output space $\mathcal{Y}$.

More precisely, Giunta and Stewart [GS24] noticed that bijective VRF satisfy what they called *unbiasability on independent points*. On a high level, this notion guarantees that, for any adversarially generated verification keys, outputs corresponding to independent, uniformly sampled inputs are indistinguishable from truly uniform outputs. This is weaker than full-fledged unbiasability in that the inputs are chosen independently and at random for each verification key. Full unbiasability, on the other hand, prescribes that the same set of random inputs is applied to all verification keys.

Our first observation, in this sense, is that the well-known Dodis-Yampolskiy VRF, when extended to consider the whole $\mathbb{Z}_p$ as the underlying input space, is actually bijective. Unfortunately, however, when setting $\mathcal{X} = \mathbb{Z}_p$ the proof of pseudorandomness from [DY05] does not go throughand we manage to prove pseudorandomness of the extended DY-VRF only in the generic group model[8].

Towards more promising candidates, we look at the constructions in [Bit17, GHKW17], that show how to build a VRF from constrained-PRFs, commitment schemes (CS), and NIWIs. Informally, the idea here is to first commit to a PRF key and later use the NIWI to prove that, on a given point $x$, a value $y$ is the correct output of the PRF corresponding to the committed key. The reason why this construction cannot be (fully) unbiasable is that it relies on (computationally) *hiding* commitments. This means that it is not possible to check efficiently whether two commitments contain the same value or not. A malicious adversary can thus commit twice to the same PRF key, this leads to a public key that, while indistinguishable from an honestly generated one, makes the resulting VRF biasable[9].

The good news is that if one instantiates the above construction with the *puncturable* PRF from [BKW17], the resulting VRF can be proven to be weakly unbiasable assuming sub-exponential hardness of the underlying primitives.

*Open Questions and Future Work.* Although our attacks and positive results provide a clearer picture regarding the unbiasability of currently known constructions and the pitfalls of algebraic techniques in this regards, some problems remain open. Specifically, while we show [BMR10, §7] and their subsequent variants [Kat17, Yam17, Nie21] to not be unbiasable in the strongest sense, we have no indications that weaker unbiasability notions can be broken as well. Furthermore, although we prove [Bit17, GHKW17] to be weakly unbiasable at least for a specific PRF choice, understanding whether the usage of stronger underlying primitives could lead to unbiasability on independent points remains an interesting theoretical question.

[8] However, unbiasability on independent points is not proved through idealized models.

[9] We stress that the attack is against full unbiasability. While in principle this does not prevent the VRF to achieve unbiasability over independent points, we cannot formally prove it achieves this property either.

| Construction | Weak Unb. | Unb. oIP | Unb. | Attack Type | Reference |
|---|---|---|---|---|---|
| [Lys02] | ✗ | – | – | Powers of $\alpha$ | Section 3.2 * |
| [Dod03] | ✗ | – | – | Powers of $\alpha$ | Section 3.2 * |
| [DY05] | ✗ | – | – | Output guess | Section 3.1 * |
| [DP07] | – | ✓ | ✗ | [Bra25, §A.2] | Section B.3 |
| [ACF09] | ✗ | – | – | Output guess | Section 3.1 |
| [BCKL09] | ✗ | – | – | Output guess | Section 3.1 |
| [BMR10, §6] | ✗ | – | – | Output guess | Section 3.1 |
| [BMR10, §7] | ? | ? | ✗ | Related keys | Section 3.3 * |
| [HW10] | ✗ | – | – | Powers of $\alpha$ | Section 3.2 |
| [ACF14] | ✗ | – | – | Powers of $\alpha$ | Section 3.2 * |
| [Jag15] | ✗ | – | – | Powers of $\alpha$ | Section 3.2 * |
| [LLC15] | ✗ | – | – | Powers of $\alpha$ | Section 3.2 * |
| [HJ16] | ✗ | – | – | Powers of $\alpha$ | Section 3.2 * |
| [Bit17] | ✓** | ? | ✗ | Duplicate keys | Section 3.4 * |
| [GHKW17] | ✓** | ? | ✗ | Duplicate keys | Section 3.4 |
| [Kat17] | ? | ? | ✗ | Related keys | Section 3.3 * |
| [Yam17] | ? | ? | ✗ | Related keys | Section 3.3 * |
| [Ros18] | ✗ | – | – | Ad hoc | Section B.2 |
| [Koh19] | ✗ | – | – | Ad hoc | Section B.1 |
| [Nie21] | ? | ? | ✗ | Related keys | Section 3.3 * |

**Fig. 1.** Summary of our results. Security notions in ascending order are: weak unbiasability (**Weak Unb.**), unbiasability on independent points (**Unb. oIP**) and unbiasability (**Unb.**). A dash – denotes results that can be inferred from attacks to weaker notions. A question mark **?** means we have no attack nor proofs for the given notion.
* The attack is readily adapted from the one described in the given section.
** The positive result only holds for specific choices of the primitives in their generic construction.

**Other Related work.** Over the years, several works extended the basic notion of VRF to encompass additional properties. Here we discuss a few of these variants. Chase and Lysyanskaya [CL07] introduced the notion of Simulatable VRFs (in the CRS model). Here, knowledge of a trapdoor can be used to open the VRF to any given output. The notion of constrained VRFs, introduced by Fuchsbauer in [Fuc14], considers the possibility of producing a constrained key that allows to evaluate the VRF only for inputs belonging to some given set. Ring VRFs consider the setting where a group of users wants to evaluate a VRF anonymously. All of these notions however do not consider the case of maliciously sampled keys. More related to unbiasable VRFs, are the notion of universally composable VRFs [DGKR18, BGK+18, BGQR22], where security against malicious adversaries comes from the usage of the UC framework itself, and distributed VRFs [Dod03] where malicious behavior is prevented by performing key generation in a distributed way. The question of considering VRF with biased keys was also considered in [BGRV09]. In the original paper [MRV99] the authors further pro-

pose a generic transformation from verifiable unpredictable functions (VUFs) to VRF applying techniques from [GL89]. We suspect that such a compiler could provide, at least, weak unbiasability when applied to certain classes of VUFs. However, VRFs resulting from the Goldreich Levin transform are cumbersome and very inefficient. As remarked above, the focus of this paper is on simple (and direct) VRF realizations.

Finally, we remark that other constructions of VRFs beyond the ones studied in this paper exist. Specifically, these are based on other assumptions, including hash function [BDE+22], lattices [EKS+21, ESLR23], group actions [Lai23], and isogenies [Ler25]. The reason we chose not to include them however is that all of them rely on the ROM for the VUF to VRF transformation, and are otherwise only known to yield a VRF through [MRV99]. As proven in [GS24], this usage of the ROM automatically gives unbiasability. Moreover, some of the above VRFs are only few-times secure, or require specific settings (e.g. a public blockchain).

## 2 Preliminaries

### Notation

$\lambda$ denotes the security parameter. A function $\varepsilon(\lambda)$ is negligible if it approaches 0 faster than the inverse of any polynomial in $\lambda$. Given a set $X$, $x \leftarrow^{\$} X$ means $x$ is sampled uniformly in $X$. For a probabilistic Turing Machine $\mathcal{A}$ we write $y \leftarrow^{\$} \mathcal{A}(x)$ to denote that $y$ is the output of $\mathcal{A}$ on input $x$. $\mathbb{Z}_p$ is the field of integers modulo $p$ with $p$ a prime number. For symmetric bilinear maps, we adopt the notation introduced in [EHK+13].

### 2.1 Bilinear Groups

We briefly recall (symmetric) bilinear maps over prime order groups [Mil04]. Given two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order $p$ (with additive notation) and $g$ a generator of $\mathbb{G}$, a pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is any map satisfying the following:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(a \cdot u, b \cdot v) = (ab) \cdot e(u, v)$.
2. Non-degenerate: $e(g, g) \neq 1$.
3. Computable: there is a $\mathsf{PPT}$ algorithm to compute $e(u, v)$ for all $u, v \in \mathbb{G}$.

*Notation.* We adopt the notation introduced in [EHK+13]. Namely, for $a \in \mathbb{Z}_p$, we denote $[a] := a \cdot g \in \mathbb{G}$ and $[a]_T := a \cdot e(g, g)$ for $\mathbb{G}_T$. We further extend this notation to vector and matrices, i.e. given $\mathbf{v} \in \mathbb{Z}_p^n$, then $[\mathbf{v}] = ([v_1], \ldots, [v_n]) \in \mathbb{G}^n$. Finally, given $A \in \mathbb{Z}_p^{n,m}$ and $B \in \mathbb{Z}_p^{m,\ell}$, we denote $e([A], [B]) \in \mathbb{G}_T^{n,\ell}$ the matrix whose $(i, k)$-entry is $\sum_{j=1}^{m} e([A_{i,j}], [B_{j,k}])$.

*Certified Groups.* We further assume all groups considered in this work to be certified [HJ16, §2]. This means that given a description of the pairing groups, it is possible to publicly verify each group to have prime order and the bilinear map to be correct and non-degenerate. Moreover, each group element has an efficiently computable unique representation.

### 2.2 Verifiable Random Functions

Verifiable random functions were introduced by [MRV99]. Here we provide a definition following the syntax of [Lys02, HJ16]. We denote with $\mathcal{X}$ the input space and with $\mathcal{Y}$ the output space (both admitting efficient membership tests and uniform sampling).

**Definition 1 (Verifiable Random Function).** *A Verifiable Random Function is a triplet of PPT algorithms* $(\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Vfy})$ *satisfying:*

1. ***Correctness:*** *For any* $(\mathsf{vk}, \mathsf{sk})$ *in the image of* $\mathsf{VRF.Gen}(1^\lambda)$ *and any* $x \in \mathcal{X}$

$$(y, \pi) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}, x) \implies \mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi) \to 1.$$

   *We say that* VRF *satisfies* $\delta$*-correctness if this happens with probability* $1-\delta$*, for some* $\delta \neq 0$*.*
2. ***Unique Provability:*** *For any* vk *(not necessarily in the range of* VRF.Gen*), input* $x \in \mathcal{X}$*, pair of outputs* $y_0, y_1 \in \mathcal{Y}$ *and proofs* $\pi_0, \pi_1$ *it holds that*

$$\mathsf{VRF.Vfy}(\mathsf{vk}, x, y_0, \pi_0) \to 1,\ \mathsf{VRF.Vfy}(\mathsf{vk}, x, y_1, \pi_1) \to 1 \implies y_0 = y_1.$$

3. ***Pseudorandomness:*** *For any* PPT *adversary* $\mathcal{A}$ *executed in experiment 2 there exists a negligible function* $\varepsilon$ *such that*

$$\mathsf{Adv}^{\mathsf{rnd}}_{\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{rnd}}_{\mathcal{A}}(\lambda) \to 1\right] - \frac{1}{2} \right| \leq \varepsilon(\lambda).$$

$\mathsf{Exp}^{\mathsf{rnd}}_{\mathcal{A}}(\lambda)$

1: Sample $b \leftarrow^{\$} \{0,1\}$
2: $\mathsf{vk}, \mathsf{sk} \leftarrow^{\$} \mathsf{VRF.Gen}(1^\lambda)$ and set $x^* = \perp$
3: $x^* \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{eval}}}(\mathsf{vk})$
4: **if** $x^*$ was previously queried:
5: **return** 0
6: $y_0 \leftarrow^{\$} \mathcal{Y}$
7: $(y_1, \pi) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}, x^*)$
8: $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{eval}}}(\mathsf{vk}, y_b)$
9: **return** $b == b'$

$\mathcal{O}_{\mathsf{eval}}(x)$

1: **if** $x \neq x^*$:
2: $(y, \pi) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}, x)$
3: **return** $(y, \pi)$
4: **else**
5: **return** $\perp$

**Fig. 2.** The pseudorandomness security game with adversary $\mathcal{A}$.

Relative to an oracle $f \in \mathcal{F}$ (e.g. the RO or the GBGM), unique provability is further required to hold for any $f \in \mathcal{F}$, whereas the advantage in the pseudorandomness game is defined as

$$\mathcal{A}^{\mathsf{rnd}}_{\mathcal{A}}(\lambda) := \left| \Pr_{f \leftarrow^{\$} \mathcal{F}} \left[\mathsf{Exp}^{\mathsf{rnd}}_{f, \mathcal{A}^f}(\lambda) \to 1\right] - \frac{1}{2} \right|$$

The focus of this work is on the *unbiasability* of a given VRF, i.e. the inability of an adversary to maliciously compute verification keys with skewed or highly related output distributions. The first game-based notion has been proposed in [EKS+21].

**Definition 2 (Weak Unbiasability [EKS+21]).** *A VRF is weakly unbiasable if for any* PPT *adversary $\mathcal{A}$, there exists a negligible function $\varepsilon$ such that*

$$\mathsf{Adv}^{\mathsf{w\text{-}bias}}_{\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{w\text{-}bias}}_{\mathcal{A}}(\lambda) \to 1\right] \leq \frac{1}{|\mathcal{Y}|} + \varepsilon(\lambda),$$

*where the experiment $\mathsf{Exp}^{\mathsf{w\text{-}bias}}_{\mathcal{A}}$ is defined in Figure 3.*

$\mathsf{Exp}^{\mathsf{w\text{-}bias}}_{\mathcal{A}}(\lambda)$

1 : $(\mathsf{vk}, y^*) \leftarrow^{\$} \mathcal{A}(1^\lambda)$
2 : $x \leftarrow^{\$} \mathcal{X}$
3 : $(\pi, y) \leftarrow^{\$} \mathcal{A}(x)$
4 : **if** $\mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi) \to 1$ and $y = y^*$
5 : **return** 1
6 : **else**
7 : **return** 0

**Fig. 3.** The weak-unbiasability experiment.

While useful in some contexts, the above notion present some limitation. Specifically it does not prevent an adversary to bias just a single output bit, or more generally, a given predicate of the output. Moreover, it does not prevent several verification keys to return correlated outputs. For this reason two stronger notions have been put forward in [GS24]. In order to avoid edge-cases where the adversary attempts to bias the output by simply refusing to evaluate the VRF on certain points, they introduce the following class of probabilistic predicates.

**Definition 3 (Monotone predicate [GS24]).** *Over $(\mathcal{Y} \cup \{\bot\})^n$ we define a partial order*

$$(y_1, \ldots, y_n) \leq (z_1, \ldots, z_n) \iff \forall i (y_i = z_i \vee y_i = \bot).$$

*A probabilistic predicate $p : (\mathcal{Y} \cup \{\bot\})^n \times \{0,1\}^r \to \{0,1\}$ is monotone if, for a uniformly sampled random tape $\rho \leftarrow^{\$} \{0,1\}^r$,*

$$\mathbf{y} \leq \mathbf{z} \implies \Pr\left[p(\mathbf{y}; \rho) = 1\right] \leq \Pr\left[p(\mathbf{z}; \rho) = 1\right].$$

*When clear from context, we omit the random tape $\rho$ and instead assume that $p(\mathbf{x})$ means sampling a tape $\rho \leftarrow^{\$} \{0,1\}^r$ and then computing $p(\mathbf{x}; \rho)$.*

Next, a notion is proposed against an adversary attempting to bias a monotone predicate toward 1. The adversary generates a number of verification keys, obtains a set of inputs and evaluates each VRF key on the respective inputs. It wins if the predicate is true on the resulting outputs with significantly higher probability than it would be on truly random outputs. Note requiring the predicate to be monotone means the advantage decreases when the adversary deliberately chooses not to evaluate on one point.

**Definition 4 (Unbiasability on independent points [GS24]).** *A VRF is unbiasable on independent points if for all polynomially bounded integers $n, m \in \mathbb{N}$ and $\mathsf{PPT}$ adversary $\mathcal{A}$, there exists a negligible function $\varepsilon$ such that*

$$\mathsf{Adv}^{\mathsf{ip\text{-}bias}}_{\mathcal{A}}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathsf{ip\text{-}bias}}_{0,n,m,\mathcal{A}}(\lambda) \to 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{ip\text{-}bias}}_{1,n,m,\mathcal{A}}(\lambda) \to 1\right] \leq \varepsilon(\lambda)$$

*where the experiment $\mathsf{Exp}^{\mathsf{ip\text{-}bias}}_{b,n,m,\mathcal{A}}$ is defined in Figure 4.*

$\mathsf{Exp}^{\mathsf{ip\text{-}bias}}_{b,n,m,\mathcal{A}}(\lambda)$

$(\mathsf{vk}_1, \ldots, \mathsf{vk}_n, p) \leftarrow \mathcal{A}$ such that:
  $\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ are all distinct
  $p$ is a $\mathsf{PPT}$computable monotone prediate
$(x_{1,1}, \ldots, x_{n,m}) \leftarrow^{\$} \mathcal{X}$
$((y_{1,1}, \pi_{1,1}), \ldots, (y_{n,m}, \pi_{n,m})) \leftarrow \mathcal{A}(x_{1,1}, \ldots, x_{n,m})$
**for** all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$ **do**
  **if** $\mathsf{VRF.Vfy}(\mathsf{vk}_i, x_{i,j}, y_{i,j}, \pi_{i,j}) \to 0$
    Set $y_{i,j} \leftarrow \perp$
Sample $(z_{1,1}, \ldots, z_{n,m}) \leftarrow^{\$} \mathcal{Y}^{n,m}$
**if** $b = 0$
  **return** $p(y_{1,1}, \ldots, y_{n,m})$
**else**
  **return** $p(z_{1,1}, \ldots, z_{n,m})$

**Fig. 4.** The unbiasability experiment with independently sampled points for each $\mathsf{vk}$.

Finally, we recall that any bijective VRF is always at least unbiasable on independent points.

**Definition 5 (Verifiable Random Bijection [GS24]).** *A Verifiable Random Bijection is a VRF* $(\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Vfy})$ *whose input space $\mathcal{X}$ and output space $\mathcal{Y}$ have the same cardinality $|\mathcal{X}| = |\mathcal{Y}|$ and furthermore satisfies*

4. ***Injectivity.*** *For any $\mathsf{vk}$ (not necessarily in the range of $\mathsf{VRF.Gen}$), any output $y \in \mathcal{Y}$, pair of inputs $x_0, x_1 \in \mathcal{X}$ and proofs $\pi_0, \pi_1$ it holds that*

$$\mathsf{VRF.Vfy}(\mathsf{vk}, x_0, y, \pi_0) = 1,\ \mathsf{VRF.Vfy}(\mathsf{vk}, x_1, y, \pi_1) = 1 \implies x_0 = x_1$$

**Theorem 1 (VRB $\implies$ unbiasability on independent points [GS24]).** *Any tuple* $(\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Vfy})$ *with* $|\mathcal{X}| = |\mathcal{Y}|$ *satisfying correctness, unique provability and injectivity, then further satisfies unbiasability on independent points.*

## 3 Attacks

In this section we describe several attacks that can be applied against the unbiasability of several standard-model VRF constructions. Other attacks against specific constructions are given in Appendix, Section B.

### 3.1 Output Guess

Many early realizations of VRFs in the literature have been shown to be secure for polynomial-sized input spaces. Among them, some feature a polynomially small *image*, while having a super-polynomially large *output space*. In this section we show such constructions cannot be weakly unbiasable. In the following, given $(\mathsf{sk}, \mathsf{vk})$ valid key pair, $\mathcal{S}_{\mathsf{vk}}$ denotes the VRF's image with public key $\mathsf{vk}$. Formally

$$\mathcal{S}_{\mathsf{vk}} \;=\; \{y \;:\; \exists(x,\pi) : \mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi) = 1\}.$$

**Theorem 2.** *Let* $\mathsf{VRF} = (\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Vfy})$ *be a VRF with domain* $\mathcal{X}$ *and codomain* $\mathcal{Y}$*. If there exists a polynomial* $q(\lambda) = \mathsf{poly}(\lambda)$ *such that* $|\mathcal{S}_{\mathsf{vk}}| \leq q(\lambda)$ *for all valid* $\mathsf{vk}$*, and* $|\mathcal{Y}|^{-1} = \mathsf{negl}(\lambda)$*, then* $\mathsf{VRF}$ *is not weakly unbiasable.*

*Proof.* We will show an explicit $\mathsf{PPT}$ adversary $\mathcal{A}$ against weak unbiasability. The attack is shown in Figure 5 and it works as follows. Initially $\mathcal{A}$ generates honestly a key pair $(\mathsf{sk}, \mathsf{vk})$. Then, it samples a random point $x^* \in \mathcal{X}$, computes the VRF's output $y^*$ and outputs the pair $(\mathsf{vk}, y^*)$. On the second call, $\mathcal{A}$ on input $x$ returns the output of $\mathsf{VRF.Eval}(\mathsf{sk}, x)$. As the VRF's output is honestly computed, the verification algorithm will accept with probability 1 by perfect correctness. Thus the advantage of $\mathcal{A}$ depends only on its ability to guess the output $y$ associated to the uniformly sampled input $x$. Since $y^*, y \in \mathcal{S}_{\mathsf{vk}}$ by definition, they follow the same probability distribution conditioning on $\mathsf{vk} = \mathsf{vk}_0$ for all valid $\mathsf{vk}_0$, and are mutually independent, we have that

$$\Pr[y^* = y] \geq \sum_{\mathsf{vk}_0} \Pr[\mathsf{vk} = \mathsf{vk}_0] \Pr[y^* = y \mid \mathsf{vk} = \mathsf{vk}_0] \geq \sum_{\mathsf{vk}_0} \frac{\Pr[\mathsf{vk} = \mathsf{vk}_0]}{|\mathcal{S}_{\mathsf{vk}_0}|} \geq \frac{1}{q(\lambda)}.$$

In particular, $\mathsf{Adv}^{\mathsf{w\text{-}bias}}_{\mathcal{A}}(\lambda) - 1/|\mathcal{Y}| \geq 1/q(\lambda) - 1/|\mathcal{Y}|$ which is not negligible. □

Consequently, the constructions in the standard model, proposed in [DY05, ACF09, BCKL09] and [BMR10, §6] are not weakly unbiasable.

$\mathcal{A}(1^\lambda)$ // first call

1 : $(\mathsf{sk}, \mathsf{vk}) \leftarrow^{\$} \mathsf{VRF.Gen}(1^\lambda)$
2 : $x^* \leftarrow^{\$} \mathcal{X}$
3 : $(y^*, _) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}, x^*)$
4 : **return** $(\mathsf{vk}, y^*)$

$\mathcal{A}(x)$ // second call

1 : $(y, \pi) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}, x)$
2 : **return** $(y, \pi)$

**Fig. 5.** Weak unbiasability attack for VRFs with $|\mathcal{S}_{\mathsf{vk}}| \leq q(\lambda)$ and $1/|\mathcal{Y}| = \mathsf{negl}(\lambda)$.

### 3.2 Powers of $\alpha$ Attack

Naor-Reingold based VRFs are subject to an attack against weak unbiasability we call *powers of* $\alpha$. For the sake of simplicity, we present it for [HW10], which we recall in Appendix A.1, but it can be instantiated analogously against [Lys02, Dod03, Jag15, HJ16].

**Theorem 3.** *The VRF construction in [HW10] is not weakly unbiasable.*

*Proof.* An explicit adversary $\mathcal{A}$ breaking weak unbiasability is shown in Figure 6. First of all, $\mathcal{A}$ samples $\alpha \in \mathbb{Z}_p$. Then, it computes the secret key by sampling $\widetilde{u}, u_0$ and setting $u_i = \alpha^i$ for each $i \in \{1, \ldots, n\}$. The verification key is constructed according to the scheme. Finally, to guess the output, it returns the VRF's evaluation $y^*$ on a randomly chosen point $x^*$. When $\mathcal{A}$ is later called on a randomly chosen $x \in \{0,1\}^n$, it returns $(y, \pi) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}, x)$. Since the output is computed honestly, $\mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi) \to 1$ as it satisfies by construction all pairing equations checked by the verifier. Focusing on $y$, we know that

$$y \;=\; \left[\widetilde{u} \cdot u_0 \cdot \prod\nolimits_{i=1}^{n} \alpha^{i \cdot x_i}\right]_T \;=\; \left[\widetilde{u} \cdot u_0 \cdot \alpha^{\sum_{i=1}^{n} i \cdot x_i}\right]_T$$

In particular it only depends on $\sum_{i=1}^{n} i \cdot x_i$ which lies in $\{0, \ldots, n(n+1)/2\}$. Moreover, $y = y^*$ if and only if $\sum_{i=1}^{n} i \cdot x_i = \sum_{i=1}^{n} i \cdot x_i^*$, which occurs with probability $\geq 1/n^2$. Thus, we have that

$$\mathsf{Adv}^{\mathsf{w\text{-}bias}}_{\mathcal{A}}(\lambda) \;=\; \Pr\left[\mathsf{Exp}^{\mathsf{w\text{-}bias}}_{\mathcal{A}}(\lambda) \to 1\right] \;\geq\; \frac{1}{n^2} \;>\; \frac{1}{|\mathbb{Z}_p|} + \mathsf{negl}(\lambda).$$

It follows that [HW10] is not weakly unbiasable. □

*Remark 1.* The previous attack is readily adapted to [HJ16], even though their underlying PRF mildly deviates from Naor-Reingold. Specifically, in their VRF the setup involves sampling $2n$ invertible matrices $M_{i,b} \in \mathbb{Z}_p^{d,d}$ (for $1 \leq i \leq n$ and $b \in \{0,1\}$) and two vectors $\mathbf{u}, \mathbf{w} \leftarrow^{\$} \mathbb{Z}_p^d$. The output on (hashed) input $x = (x_1, \ldots, x_n)$ is then $\left[\widehat{\mathbf{w}}^\top M_{n,x_n} \cdot \ldots \cdot M_{1,x_1} \mathbf{u}\right]_T$ where $\widehat{\mathbf{w}} = (1/w_1, \ldots, 1/w_d)$. Weak unbiasability is broken setting $M_{i,b} = M^{i+nb}$ for an invertible $M \leftarrow^{\$} \mathbb{Z}_p^{d,d}$.

| $\mathcal{A}(1^\lambda)$ // first call | $\mathcal{A}(x)$ // second call |
|---|---|
| 1 : Sample $\alpha \leftarrow^\$ \mathbb{Z}_p$ | 1 : **return** $\mathsf{VRF.Eval}(\mathsf{sk}, x)$ |
| 2 : $\widetilde{u} \leftarrow^\$ \mathbb{Z}_p$; $u_0 \leftarrow^\$ \mathbb{Z}_p$ | |
| 3 : $\mathsf{sk} \leftarrow (\widetilde{u}, u_0, \alpha^1, \ldots, \alpha^n)$ | |
| 4 : $\mathsf{vk} \leftarrow \left([\widetilde{u}], [u_0], [\alpha^1], \ldots, [\alpha^n]\right)$ | |
| 5 : $x^* \leftarrow^\$ \{0,1\}^n$ | |
| 6 : $(y^*, _) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}, x^*)$ | |
| 7 : **return** $(\mathsf{vk}, y^*)$ | |

**Fig. 6.** Adversary $\mathcal{A}$ against weak unbiasability for [HW10].

### 3.3 Related Keys Attack

In most Naor-Reingold based VRF, e.g. [Lys02, HW10, Jag15, HJ16, Koh19], full unbiasability can also be attacked via a related keys attack. Informally, we will exploit the fact that two keys generated as multiples of each other, when evaluated on the same input generate outputs values that are also proportional (possibly with a different factor). Here we present it against [Koh19], although the attack can be readily adapted to all constructions mentioned above. Before detailing the attack we quickly recall [Koh19] (a more in depth description can be found in the Appendix, Section A.3).

The scheme relies on a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and an admissible hash [BB04] $\mathsf{AHF} : \{0,1\}^L \to \Sigma^n$ for a polynomially small alphabet $\Sigma$. The secret key consists of matrices[10] $M_{i,j,k} \in \mathbb{Z}_p^{d,d}$, for $i \in \{1, \ldots, \ell\}$, $j \in \{1, \ldots, n\}$, $k \in \Sigma$ and two vectors $\mathbf{u} \in \mathbb{Z}_p^d$, $\mathbf{w} \in \mathbb{Z}_p^d$. The verification key is $\mathsf{vk} = \left([M_{i,j,k}], [\mathbf{u}], [\mathbf{w}]\right)$. On input $x \in \{0,1\}^L$ with encoding $\mathsf{AHF}(x) = (x_1, \ldots, x_n) \in \Sigma^n$ the VRF output is $y \in \mathbb{G}$ with

$$\mathbf{v} = \left(\prod\nolimits_{i=1}^{\ell} \sum\nolimits_{j=1}^{n} M_{i,j,x_j}\right) \mathbf{u}, \qquad y = [v_1/w_1] + \ldots + [v_d/w_d] \,.$$

A proof instead consists of all the partial products $[\mathbf{v}_\iota]$ and terms $[\mathbf{z}]$ of $y$, i.e. such that for $\iota \in \{1, \ldots, \ell\}$

$$\mathbf{v}_\iota \;=\; \left(\prod\nolimits_{i=1}^{\iota} \sum\nolimits_{j=1}^{n} M_{i,j,x_j}\right) \mathbf{u}, \qquad z_i = v_{\ell,i}/w_i.$$

Given such a proof for input $x$ the verifier checks that:

1. For all $\iota \in \{1, \ldots, \ell\}$, $[\mathbf{v}_\iota]_T = e\left(\sum_{j=1}^{n} \left[M_{\iota,j,x_j}\right], [\mathbf{v}_{\iota-1}]\right)$, where $\mathbf{v}_0 = \mathbf{u}$.
2. $e([w_i], [z_i]) = [v_{\ell,i}]_T$ for $i \in \{1, \ldots, d\}$
3. $y = [z_1] + \ldots + [z_n]$

[10] When security is reduced to the Decision Linear assumption [BBS04], $d = 3$.

**Theorem 4.** *The VRF construction proposed in [Koh19] is not unbiasable.*

*Proof.* The theorem is proven by showing the existence of an adversary with significant advantage. We consider the case of two verification keys $\mathsf{vk}_1, \mathsf{vk}_2$ evaluated in a single point $x$. The adversary $\mathcal{A}$ is shown in Figure 7 and works as follows.

On the first call, $\mathcal{A}$ honestly generates a key pair $(\mathsf{sk}_1, \mathsf{vk}_1)$ with $\mathsf{sk} = (\mathbf{M}, \mathbf{u}, \mathbf{w})$. Then, it samples $\alpha$ in $\mathbb{Z}_p$ and, for each $i, j, k$, it computes $\mathbf{M}'_{i,j,k} = \alpha \cdot \mathbf{M}_{i,j,k}$. It then sets the second key $\mathsf{vk}_2$ as $([\mathbf{M}'], [\mathbf{u}], [\mathbf{w}])$ and $\mathsf{sk}_2$ as $(\mathbf{M}', \mathbf{u}, \mathbf{w})$. Finally, it chooses a predicate $p(y, y')$ that is true iff $y' = \alpha^\ell \cdot y$ and returns $(\mathsf{vk}_1, \mathsf{vk}_2, p)$. Next, when $\mathcal{A}$ is called on a randomly chosen point $x_1$, it honestly computes $(y_{1,1}, \pi_{1,1})$ and $(y_{2,1}, \pi_{2,1})$ using $\mathsf{sk}_1, \mathsf{sk}_2$ with the algorithm $\mathsf{VRF.Eval}$ and returns them to experiment.

By the perfect correctness of the given scheme, as $(\mathsf{sk}_1, \mathsf{vk}_1)$ is honestly generated, $(y_{1,1}, \pi_{1,1})$ will be accepted by $\mathsf{VRF.Vfy}$ with probability 1. Analogously, as $(\mathbf{M}', \mathbf{u}, \mathbf{w})$ follows the same distribution of an honestly generated key,[11] we also have that $\Pr\left[\mathsf{VRF.Vfy}(\mathsf{vk}_2, x_1, y_{1,2}, \pi_{1,2}) = 0\right]$ is zero.

Next we show the predicate to always be true when provided the VRF's output values. Let $\mathsf{AHF}(x) = (x_1, \ldots, x_n) \in \Sigma^n$ and $\mathbf{v}, \mathbf{v}'$ respectively the intermediate vectors used to compute $y_{1,1}, y_{2,1}$. Then, by construction

$$\mathbf{v}' \;=\; \left(\prod\nolimits_{i=1}^{\ell} \sum\nolimits_{j=1}^{n} M'_{i,j,x_j}\right)\mathbf{u} \;=\; \left(\prod\nolimits_{i=1}^{\ell} \sum\nolimits_{j=1}^{n} \alpha M_{i,j,x_j}\right)\mathbf{u} \;=\; \alpha^\ell \mathbf{v}.$$

In particular $y_{2,1} = [v'_1/w'_1 + \ldots + v'_d/w'_d] = \alpha^\ell \cdot [v'_1/w'_1 + \ldots + v'_d/w'_d] = \alpha^\ell y_{1,1}$. The advantage of $\mathcal{A}$ is therefore, given uniformly distributed $y^*_{1,1}, y^*_{2,1}$ in $\mathbb{G}$.

$$\begin{aligned}\mathsf{Adv}^{\mathsf{bias}}_{\mathcal{A}}(\lambda) &= \Pr\left[\mathsf{Exp}^{\mathsf{bias}}_{0,2,1,\mathcal{A}}(\lambda) \to 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{bias}}_{1,2,1,\mathcal{A}}(\lambda) \to 1\right] \\ &= 1 - \Pr\left[y^*_{2,1} = \alpha^\ell \cdot y^*_{1,1}\right] \;=\; 1 - \frac{1}{p} = 1 - \mathsf{negl}(\lambda).\end{aligned}$$

□

A similar attack can also be applied against the constructions in [BMR10, Kat17, Yam17, Ros18] and [Nie21].

### 3.4 Duplicate Keys Attack

[Bit17] and [GHKW17] proposed two VRF constructions from constrained PRFs and commitment schemes (CSs) and non-interactive witness indistinguishability (NIWI). At a high level both constructions work by first committing to a PRF key, and later using the NIWI to prove that, on a given point $x$, a value $y$ is the correct output of the PRF with the committed key. In order for their proof to work however, the commitment must be hiding, and in particular does not

[11] Crucially, this only holds *not* conditioning on $(\mathsf{sk}_1, \mathsf{vk}_1)$, which suffices to bound the rejection probability.

$\mathcal{A}(1^\lambda)$ // first call

1 : $(\mathsf{sk}_1, \mathsf{vk}_1) \leftarrow^{\$} \mathsf{VRF.Gen}(1^\lambda)$
2 : parse $\mathsf{sk}_1 = (\mathbf{M}, \mathbf{u}, \mathbf{w})$ // $M_{i,j,k} \in \mathbb{Z}_p^{d,d}$, $\mathbf{u}, \mathbf{w} \in \mathbb{Z}_p^d$
3 : sample $\alpha \leftarrow^{\$} \mathbb{Z}_p$
4 : For each $(i, j, k) \in \{1, \ldots, \ell\} \times \{1, \ldots, n\} \times \Sigma$:
5 : $\quad M'_{i,j,k} \leftarrow \alpha \cdot M'_{i,j,k}$
6 : $\mathsf{sk}_2 = (\mathbf{M}', \mathbf{u}, \mathbf{w}), \mathsf{vk}_2 = ([\mathbf{M}'], [\mathbf{u}], [\mathbf{w}])$
7 : set $p(y_{1,1}, y_{2,1})$ the predicate $y_{2,1} == \alpha^\ell \cdot y_{1,1}$
8 : **return** $(\mathsf{vk}_1, \mathsf{vk}_2, p)$

$\mathcal{A}(x_1)$ // second call

1 : $(y_{1,1}, \pi_{1,1}) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}_1, x_1)$
2 : $(y_{2,1}, \pi_{2,1}) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}_2, x_1)$
3 : **return** $((y_{1,1}, \pi_{1,1}), (y_{2,1}, \pi_{2,1}))$

**Fig. 7.** Adversary $\mathcal{A}$ against unbiasability for [Koh19].

allow to efficiently test whether two (perfectly binding) commitments contain the same value or not. We exploit this to break full unbiasability, as we can generate two distinct verification keys defining the same underlying function.

In the following we informally recall [GHKW17], and defer further details on the construction to the Appendix, Section A.2. Let $(\mathsf{CPRF.Gen}, \mathsf{CPRF.Eval}, \mathsf{CPRF.Cons})$ be a constrainable PRF[12], $(\mathsf{C.Com}, \mathsf{C.Vfy})$ a perfectly binding commitment and $(\mathsf{NIWI.P}, \mathsf{NIWI.V})$ a NIWI for the relation

$$\mathcal{R} = \left\{ ((c_1, c_2, c_3, x, y), (i_1, i_2, k_1, k_2, r_1, r_2)) \;\middle|\; \begin{array}{l} \mathsf{C.Vfy}(c_{i_j}, k_j, r_j) = 1 \;\; \forall j \in \{1, 2\} \\ i_1 \neq i_2, \;\; f_{k_1}(x) = f_{k_2}(x) = y \end{array} \right\}.$$

Given the above, the construction works as follows. A verification key consists of three commitments $\mathsf{vk} = (c_1, c_2, c_3)$ to a PRF key $k$, along with their opening information $r_1, r_2, r_3$. The evaluation on a point $x$ is defined as $y = f_k(x)$ while the proof is a NIWI proof about $(c_1, c_2, c_3, x, y)$ being in $\mathcal{L}_\mathcal{R}$ with witness $(1, 2, k, k, r_1, r_2)$. Finally, the verification accepts if and only if the NIWI is accepting. Note that up to minor syntactical changes the general construction by Bitansky is identical, see [Bit17, §1.3].

**Theorem 5.** *For any perfectly binding and computationally hiding commitment* $(\mathsf{C.Com}, \mathsf{C.Vfy})$*, constrainable PRF* $(\mathsf{CPRF.Gen}, \mathsf{CPRF.Eval}, \mathsf{CPRF.Cons})$ *and perfectly correct NIWI* $(\mathsf{NIWI.P}, \mathsf{NIWI.V})$ *for the relation* $\mathcal{R}$*, the VRF construction in [GHKW17] is not unbiasable.*

*Proof.* We consider the unbiasability experiment with two verification keys $\mathsf{vk}_1, \mathsf{vk}_2$ evaluated on the same point $x_1$. Our adversary $\mathcal{A}$ is shown in Figure 8. On the

[12] As customary, we denote $\mathsf{CPRF.Eval}(k, x) = f_k(x)$.

first call, $\mathcal{A}$ generates two keys $(\mathsf{vk}_1, \mathsf{vk}_2)$ obtained committing to *the same* PRF key $k$. It then returns the two keys along with a predicate $p$ for equality testing. Next, when prompted with input $x$, it honestly evaluates both VRF on that input and returns the result.

First, we observe that in our attack it may happen that $\mathsf{vk}_1 = \mathsf{vk}_2$. However $\Pr[\mathsf{vk}_1 = \mathsf{vk}_2] \leq \Pr[c_{1,j} = c_{2,j},\ j \in \{1,2,3\}]$ which is negligible due to the computational hiding property of the commitment. Informally the reduction is done trough an adversary $\mathcal{B}$ that samples $k$, asks for a commitment $c_1^*$ of either $k$ or $k^* \neq k$ and locally compute a commitment $c_2^*$ of $k$. It accepts only if $c_1^* = c_2^*$, which due to perfect binding can only occur when $k$ was committed. Hence $\mathsf{negl}(\lambda) \geq \mathsf{Adv}_{\mathcal{B}}(\lambda) = \Pr[c_{1,j} = c_{2,j}]$ for each $j$.

Next, as each individual $(\mathsf{vk}_i, \mathsf{sk}_i)$ is distributed perfectly as prescribed in [GHKW17], the verifier accepts due to perfect correctness. Finally, due to $k$ being the same in both commitments, by construction we have that $y_{1,1} = f_k(x_1) = y_{2,1}$. Hence, calling $\mathcal{Y}$ the PRF output space[13] and $z_1, z_2 \leftarrow^{\$} \mathcal{Y}$:

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A}}(\lambda) &= \Pr[p(y_{1,1}, y_{2,1}) = 1 \wedge \mathsf{vk}_1 \neq \mathsf{vk}_2] - \Pr[p(z_1, z_2)] \\
&\geq \Pr[y_{1,1} = y_{2,1} \mid \mathsf{vk}_1 \neq \mathsf{vk}_2] \Pr[\mathsf{vk}_1 \neq \mathsf{vk}_2] - \frac{1}{|\mathcal{Y}|} \\
&\geq \Pr[f_k(x_1) = f_k(x_1) \mid \mathsf{vk}_1 \neq \mathsf{vk}_2] (1 - \mathsf{negl}(\lambda)) - \frac{1}{2} \\
&= \frac{1}{2} - \mathsf{negl}(\lambda),
\end{aligned}$$

which is not negligible. Hence $\mathcal{A}$ breaks unbiasability.

$\mathcal{A}(1^\lambda)$ // first call

1 : $k \leftarrow^{\$} \mathsf{CPRF.Gen}(1^\lambda)$
2 : **for** $(i,j) \in \{1,2\} \times \{1,2,3\}$:
3 : $\quad (c_{i,j}, r_{i,j}) \leftarrow^{\$} \mathsf{C.Com}(1^\lambda, k)$
4 : $\mathsf{vk}_i = (c_{i,1}, c_{i,2}, c_{i,3})$ for $i \in \{1,2\}$
5 : **If** $\mathsf{vk}_1 = \mathsf{vk}_2$: **return** $\perp$
6 : Let $p$ be s.t. $p(y_1, y_2) = (y_1 == y_2)$
7 : **return** $(\mathsf{vk}_1, \mathsf{vk}_2, p)$

$\mathcal{A}(x_1)$ // second call

1 : $\mathsf{sk}_i \leftarrow (k, r_{i,1}, r_{i,2}, r_{i,3})$ for $i \in \{1,2\}$
2 : $(y_{1,1}, \pi_{1,1}) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}_1, x_1)$
3 : $(y_{2,1}, \pi_{2,1}) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}_2, x_1)$
4 : **return** $((y_{1,1}, \pi_{1,1}), (y_{2,1}, \pi_{2,1}))$

**Fig. 8.** Duplicate key attack $\mathcal{A}$ against the unbiasability of [GHKW17].

□

[13] A-priori we cannot assume $\mathcal{Y}$ to be exponentially large, but at least $|\mathcal{Y}| \geq 2$.

## 4 Limitations in preventing powers of $\alpha$ attack

Our "powers of $\alpha$" attack (Section 3.2) is carried out setting in the Hohenberger-Waters VRF the verification key as $\mathsf{vk} = ([\widetilde{u}], [u], [\alpha^1], \ldots, [\alpha^n])$, as opposed to a vector of truly random group elements in $\mathbb{G}$. However, such an attack would be easily detectable in practice. Indeed anyone could verify that $e([\alpha], [\alpha]) = e([1], [\alpha^2])$ due to the pairing being symmetric. In particular, there exists a way to modify $\mathsf{VRF.Vfy}$ so that it rejects *malformed keys*, ultimately preventing the attack.

In this section we will show that the above attack can be generalized to use exponents $z_1, \ldots, z_n$ (as opposed to $1, \ldots, n$) so that no efficient verification procedure extending the one defined in [HW10] can detect the attack with overwhelming probability.

To show that this generalization is undetectable, we introduce a (non-standard) $Q$-type non-interactive assumption over symmetric pairing groups. We then prove undetectability by showing that any enhanced verification algorithm would break this new assumption. As is often done in such cases (see, e.g. [Boy08]), we justify this new assumption in the generic group model.

In order to describe in a compact ways the relations among the group elements in the verification key and proof for the Hohenberger-Waters VRF, for each $n \in \mathbb{N}$, $\mathbf{x} \in \{0,1\}^n$ we define $V_{\mathbf{x}} \subseteq \mathbb{Z}_p^n$ the set containing the degrees of all monomials in $u_1, \ldots, u_n$ a verifier can compute through pairings. Formally $V_{\mathsf{x}} =$

$$\{\mathbf{e}_i : i \leq n\} \cup \{\mathbf{e}_i + \mathbf{e}_j : i \leq j\} \cup \{\pi_i(\mathbf{x}) + \mathbf{e}_j : i, j \leq n\} \cup \{\pi_i(\mathbf{x}) + \pi_j(\mathbf{x}) : i \leq j\}$$

with $\pi_i : \mathbb{Z}_p^n \to \mathbb{Z}_p^n$ the function $\pi_i(x_1, \ldots, x_n) = (x_1, \ldots, x_i, 0, \ldots, 0)$. Moreover, for any $\mathbf{z} \in \mathbb{Z}_p^n$ we will define $\varphi_{\mathbf{z},\mathbf{x}} : V_{\mathbf{x}} \to \mathbb{Z}_p$ so that $\varphi_{\mathbf{z},\mathbf{x}}(\mathbf{v}) = \mathbf{v}^\top \mathbf{z}$. With this notation we introduce the following assumption.

**Definition 6.** *Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ symmetric pairing, $n, B$ polynomially bounded in $\lambda$, and $(\mathbf{z}, \mathbf{x}) \leftarrow^{\$} \{1, \ldots, B\}^n \times \{0,1\}^n$ such that $\varphi_{\mathbf{z},\mathbf{x}}$ is injective. Given $h, \widetilde{u}, u_0, \alpha, u_{1,1}, \ldots, u_{1,n} \leftarrow^{\$} \mathbb{Z}_p$, and $u_{0,i} = \alpha^{z_i}$, consider the two distributions $D_0, D_1$ such that $D_b = ([h], [\widetilde{u}], [u_0], ([u_{b,i}])_{i=1}^n, [y_b]_T, [\pi_{b,0}], ([\pi_{b,i}])_{i=1}^n)$ where*

$$y_b = h\widetilde{u}u_0 \prod_{j=1}^n u_{b,j}^{x_j}, \quad \pi_{b,0} = \widetilde{u}u_0 \prod_{j=1}^n u_{b,j}^{x_j}, \quad \pi_{b,i} = \widetilde{u} \prod_{j=1}^i u_{b,j}^{x_i}.$$

*Then the $(B,n)$-Assumption holds if for any $\mathsf{PPT}$ adversary $\mathcal{A}$*

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) \;=\; |\Pr[\mathcal{A}(D_0) \to 1] - \Pr[\mathcal{A}(D_1) \to 1]| \;\leq\; \mathsf{negl}(\lambda).$$

**Proposition 1.** *In the symmetric bilinear GGM of order $p = \Theta(2^\lambda)$, for all $n, B$ polynomially bounded, any $Q$-queries adversary $\mathcal{A}$ breaks the $(B,n)$-Assumption with advantage*

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) \;\leq\; \frac{Q^2(nB+3)}{p} + \mathsf{negl}(\lambda)$$

*Proof.* Let us first rename the group elements in the distribution $D_b$ as

$$(T, \widetilde{X}, X_0, X_1, \ldots, X_n, Y, Z_0, Z_1, \ldots, Z_n)$$

In the polynomial ring $\mathbb{Z}_p$ of $2n+5$ variables, consider the following polynomials, describing the *trivial* relations among the above group elements:

$$\widetilde{q} = Y - TZ_0, \quad q_0 = Z_0 - Z_n X_0, \quad q_1 = Z_1 - \widetilde{X} X_1^{x_1}, \quad q_i = Z_i - Z_{i-1} X_i^{x_i}$$

for $i \in \{2, \ldots, n\}$. Given a generic adversary $\mathcal{A}$ distinguishing the two distributions, we then describe a simulator $\mathcal{S}$. This stores two lists of polynomials $L_0, L_1$ representing respectively group elements in $\mathbb{G}$ and $\mathbb{G}_T$. Initially $L_0 = \{T, \widetilde{X}, X_0, X_1, \ldots, X_n, Z_0, Z_1, \ldots, Z_n\}$ and $L_1 = \{Y\}$. Then, given the quotient ring

$$R = \frac{\mathbb{Z}_p[T, \widetilde{X}, X_0, X_1, \ldots, X_n, Y, Z_0, Z_1, \ldots, Z_n]}{(\widetilde{q}, q_0, q_1, \ldots, q_n)}$$

it lazily maintains two random functions $\xi_0, \xi_1 : R \to \{0,1\}^\ell$.

Next, each time $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{add}}(\xi_b(f_1), \xi_b(f_2))$ for $f_1, f_2 \in L_b$ returns $\xi_b(f_1 + f_2)$ and stores $f_1 + f_2$ in $L_b$ if not present already[14]. Similarly, $\mathcal{O}_{\mathsf{pair}}(\xi_0(f_1), \xi_0(f_2))$ is answered with $\xi_1(f_1 \cdot f_2)$, and $f_1 f_2$ is added to $L_2$ if not present already. In all other cases, it returns $\perp$. Note that up to negligible probability $\leq (2Qp)/2^\ell$, $\mathcal{S}$ returns $\perp$ exactly when the GGM oracles would[15].

Conditioning on the above event not occurring, we observe that $\mathcal{S}$ perfectly simulates the view of $\mathcal{A}$ with input drawn from the distribution $D_1$. To conclude it suffice to prove that the view of $\mathcal{A}$ respectively simulated by $\mathcal{S}$ and with input $D_0$ with real GGM oracles has negligible statistical distance.

The two views are in fact identically distributed as long as $\mathcal{A}$ does not query two $f_i, f_j \in L_b$ such that $f_i \neq f_j$ over $R$ but, for $h, \widetilde{u}, u_0, \alpha \leftarrow^{\$} \mathbb{Z}_p$

$$f(h, \widetilde{u}, u_0, \alpha^{z_1}, \ldots, \alpha^{z_n}, h\widetilde{u}u_0\alpha^{\mathbf{x}^\top \mathbf{z}}, \widetilde{u}u_0\alpha^{\mathbf{x}^\top \mathbf{z}}, \widetilde{u}\alpha^{x_1 z_1}, \ldots, \widetilde{u}\alpha^{\mathbf{x}^\top \mathbf{z}}) = 0.$$

Let $f = f_i - f_j \neq 0$. It can easily be shown by induction on the queries of $\mathcal{A}$ that there exists a representative of $f \in \mathbb{Z}_p[T, \widetilde{X}, X_0, \ldots, X_n, Y, Z_0, \ldots, Z_n]$ of total degree at most 2. Then, using the relations on $R$, we can fix a different representative $f^* \in \mathbb{Z}_p[T, \widetilde{X}, X_0, \ldots, X_n]$ obtained mapping in $f$

$$\begin{aligned} Y &\mapsto T\widetilde{X}X_0X_1 \cdot \ldots \cdot X_n \\ Z_0 &\mapsto \widetilde{X}X_0X_1 \cdot \ldots \cdot X_n \\ Z_i &\mapsto \widetilde{X}X_1 \cdot \ldots \cdot X_i \qquad \forall i \in \{1, \ldots n\} \end{aligned}$$

[14] Note equality in $R$ can be tested efficiently, even though we do not need $\mathcal{S}$ to be efficient as our argument is unconditional.

[15] The bound follows as, calling $S_1 \subseteq \mathbb{G} \cup \mathbb{G}_T$ the observed labels and $S_0$ the unobserved ones involved in a previous query with output $\perp$, the probability of an unobserved label to lie in $\mathbb{G}$ (resp. $\mathbb{G}_T$) conditioned on previous queries is smaller than $(p - |S_1|)/(2^\ell - |S_0|) \leq p/(2^\ell - 2Q) \leq 2p/2^\ell$, where $|S_0| \leq 2Q$ as each query involves at most two labels. A union bound on $Q$ queries gives the claimed inequality.

In particular, each monomial of $f^*$ is of the form $T^{\delta_0}\widetilde{X}^{\delta_1}X_0^{\delta_2}X_1^{d_1}\cdot\ldots\cdot X_n^{d_n}$ with $(d_1,\ldots,d_n)\in V_{\mathbf{x}}$. Note $f\in R$ is non-zero iff $f^*$ is non-zero in $\mathbb{Z}_p[T,\widetilde{X},X_0,\ldots,X_n]$. Finally consider the map $\varphi:\mathbb{Z}_p[T,\widetilde{X},X_0,\ldots,X_n]\to\mathbb{Z}_p[T,\widetilde{X},X_0,W]$ such that $\varphi$ maps $T,\widetilde{X},X_0$ to the homonymous variables in the second ring, and $\varphi(X_i)=W^{z_i}$. We can define

$$g(T,\widetilde{X},X_0,W)\;=\;f^*(T,\widetilde{X},X_0,W^{z_1},\ldots,W^{z_n}).$$

We will now show that $f^*\neq 0$ implies $g\neq 0$. Indeed let $T^{\delta_0}\widetilde{X}^{\delta_1}X_0^{\delta_2}X_1^{d_1}\ldots X_n^{d_n}$ be a monomial with non-zero coefficient. Then it suffices to show that no other monomial of $f^*$ is mapped to

$$\varphi\left(T^{\delta_0}\widetilde{X}^{\delta_1}X_0^{\delta_2}X_1^{d_1}\ldots X_n^{d_n}\right)\;=\;T^{\delta_0}\widetilde{X}^{\delta_1}X_0^{\delta_2}W^{d_1z_1+\ldots d_nz_n}.$$

Toward contradiction, let $T^{\delta'_0}\widetilde{X}^{\delta'_1}X_0^{\delta'_2}X_1^{d'_1}\ldots X_n^{d'_n}$ be a distinct monomial with non-zero coefficient in $f^*$ colliding with the one above. Then $\delta'_j=\delta_j$ for $j\in\{0,1,2\}$. Moreover $\mathbf{d}^\top\mathbf{z}=(\mathbf{d}')^\top\mathbf{z}$ for $\mathbf{d},\mathbf{d}'\in V_{\mathbf{x}}$. As $\varphi_{\mathbf{z},\mathbf{x}}$ is injective over $V_{\mathbf{x}}$ by hypothesis, this implies $\mathbf{d}=\mathbf{d}'$. Thus the two monomials are the same. Hence, $f\neq 0$ implies $g\neq 0$ with $\deg_W(g)\leq 2n\|\mathbf{z}\|_\infty\leq 2nB$ and degree in all other variables no greater than 2. By Schwartz-Zippel Lemma then

$$\begin{aligned}&\Pr\left[f(h,\widetilde{u},u_0,\alpha^{z_1},\ldots,\alpha^{z_n},h\widetilde{u}u_0\alpha^{\mathbf{x}^\top\mathbf{z}},\widetilde{u}u_0\alpha^{\mathbf{x}^\top\mathbf{z}},\widetilde{u}\alpha^{x_1z_1},\ldots,\widetilde{u}\alpha^{\mathbf{x}^\top\mathbf{z}})=0\right]\\ =\;&\Pr\left[g(h,\widetilde{u},u_0,\alpha)=0\right]\;\leq\;\frac{\deg(g)}{p}\;\leq\;\frac{2nB+6}{p}.\end{aligned}$$

We can then conclude with a union bound on all possible polynomial pairs $(f_i,f_j)$ that

$$\mathsf{Adv}_{\mathcal{A}}(\lambda)\;\leq\;\frac{Q^2\cdot(nB+3)}{p}+\mathsf{negl}(\lambda).$$

□

**Theorem 6.** *Let* $(\mathsf{VRF.Gen},\mathsf{VRF.Eval},\mathsf{VRF.Vfy})$ *be the VRF in [HW10]. Then under the assumption in Definition 6, there exists no efficient procedure* $\mathsf{VRF.Vfy}^*$ *such that* $(\mathsf{VRF.Gen},\mathsf{VRF.Eval},\mathsf{VRF.Vfy}^*)$ *is simultaneously:*

1. *$\delta$-correct, for $\delta\leq 1-1/\mathsf{poly}(\lambda)$;*
2. *Weakly unbiasable.*

*Proof.* Assume that $(\mathsf{VRF.Gen},\mathsf{VRF.Eval},\mathsf{VRF.Vfy}^*)$ is a VRF with $\delta$-correctness. We then define the following adversary $(\mathcal{A}_0,\mathcal{A}_1)$ against weak unbiasability. Let $q$ a polynomially small prime so that $4(n^2+n)^2\leq q<8(n^2+n)^2$ (such a prime exists by Bertrand's postulate). $\mathcal{A}_0$ initially generates a symmetric bilinear group of order $p$, samples $\widetilde{u},u_0,\alpha\leftarrow^{\$}\mathbb{Z}_p$ and $z_1,\ldots,z_n\leftarrow^{\$}\{1,\ldots,q\}$ and sets:

$$\mathsf{vk}=([\widetilde{u}],[u_0],[\alpha^{z_1}],\ldots,[\alpha^{z_n}]),\qquad\mathsf{sk}=(\widetilde{u},u_0,\alpha^{z_1},\ldots,\alpha^{z_n}).$$

Then, it samples $m \leftarrow^{\$} \{1, \ldots, nq\}$, computes $y^* = [\widetilde{u} u_0 \alpha^m]_T$ and returns $(\mathsf{vk}, y^*)$, passing $\mathsf{sk}$ as a state to $\mathcal{A}_1$. Next, given $\mathbf{x}$ uniformly sampled in $\{0,1\}^n$ by the challenger, $\mathcal{A}_1(\mathsf{sk}, \mathbf{x})$ simply returns $(y, \pi) \leftarrow^{\$} \mathsf{VRF.Eval}(\mathsf{sk}, \mathbf{x})$.

By construction $y = [\widetilde{u} u_0 \alpha^{x_1 z_1 + \ldots + x_n z_n}]_T$. Thus $y = y^*$ if and only if $m = \mathbf{x}^\top \mathbf{z}$. Note $\mathbf{x}^\top \mathbf{z} \in \{1, \ldots, n \cdot q\}$ while $m$ is uniformly distributed over the same set, and independently from $\mathbf{x}$, $\mathbf{z}$. Calling $\mathsf{Accept}$ the event $\mathsf{VRF.Vfy}^*(\mathsf{vk}, \mathbf{x}, y, \pi) = 1$ we can bound the adversary's advantage as

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A}}(\lambda) \;&=\; \Pr\left[y = y^* \wedge \mathsf{Accept}\right] - \frac{1}{p} \;=\; \Pr\left[y = y^* \,|\, \mathsf{Accept}\right] \cdot \Pr\left[\mathsf{Accept}\right] - \frac{1}{p} \\
&\geq\; \frac{1}{nq} \cdot \Pr\left[\mathsf{Accept}\right] - \frac{1}{p} \;\geq\; \frac{1}{8n(n^2+n)^2} \cdot \Pr\left[\mathsf{Accept}\right] - \frac{1}{p}.
\end{aligned}$$

To conclude we have to prove $\Pr[\mathsf{Accept}]$ is significant. Let $\mathsf{Good}$ be the event "$\varphi_{\mathbf{z},\mathbf{x}}$ is injective".

**Claim 1.** $\Pr[\mathsf{Good}] \geq 1/2$.

Conditioning on $\mathsf{Good}$ then $(\mathbf{z}, \mathbf{x})_{|\mathsf{Good}}$ is uniformly distributed over $\{1, \ldots, q\}^n \times \{0,1\}^n$ so that $\varphi_{\mathbf{z},\mathbf{x}}$ is injective. $(\mathsf{vk}, y, \pi)$ follows the distribution $D_0$ in Definition 6. Moreover, calling $(\widetilde{\mathsf{vk}}, \widetilde{\mathsf{sk}}) \leftarrow^{\$} \mathsf{VRF.Gen}(1^\lambda)$, and $(\widetilde{y}, \widetilde{\pi}) \leftarrow^{\$} \mathsf{VRF.Eval}(\widetilde{\mathsf{sk}}, \mathbf{x})$ then $(\widetilde{\mathsf{vk}}, \widetilde{y}, \widetilde{\pi})$ is distributed as $D_1$. Since we assume the two distributions to be computationally indistinguishable we have that

$$\begin{aligned}
\Pr\left[\mathsf{Accept}\right] \;&\geq\; \Pr\left[\mathsf{VRF.Vfy}^*(\mathsf{vk}, \mathbf{x}, \mathbf{y}, \pi) = 1 \,|\, \mathsf{Good}\right] \Pr\left[\mathsf{Good}\right] \\
&\geq\; \frac{1}{2}\left(\Pr\left[\mathsf{VRF.Vfy}^*(\widetilde{\mathsf{vk}}, \mathbf{x}, \widetilde{y}, \widetilde{\pi}) = 1\right] - \mathsf{negl}(\lambda)\right) \\
&\geq\; \frac{1}{2}\left(\frac{1}{\mathsf{poly}(\lambda)} - \mathsf{negl}(\lambda)\right),
\end{aligned}$$

where the second inequality uses Claim 1 and the hypothesis on the group satisfying the $(q,n)$-assumption of Definition 6, while the third comes from $\delta$-correctness with $\delta = 1 - 1/\mathsf{poly}(\lambda)$. We can thus conclude that

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) \;\geq\; \frac{1}{16n(n^2+n)^2} \cdot \frac{1}{\mathsf{poly}(\lambda)} - \mathsf{negl}(\lambda)$$

which is not negligible. □

*Proof.* Before proving the claim we make the following observations. First, $\mathbf{z} \in \{1, \ldots, q\}^n$ and $V_{\mathbf{x}} \subseteq \{0,1,2\}^n$ implies that $0 \leq \mathbf{v}^\top \mathbf{z} \leq 2qn$ for all $\mathbf{v} \in V_{\mathbf{x}}$. Second, $|V_{\mathbf{x}}| \leq 2(n^2+n)$.

$$
\begin{aligned}
\Pr[\neg\mathsf{Good}] &= \Pr\left[\exists \mathbf{v}_1, \mathbf{v}_2 \in V_{\mathbf{x}} \,:\, \mathsf{v}_1 \neq \mathbf{v}_2 \;\wedge\; \left(\mathbf{z}^\top(\mathbf{v}_1 - \mathbf{v}_2) = \mathbf{0} \mod p\right)\right] \\
&= \Pr\left[\exists \mathbf{v}_1, \mathbf{v}_2 \in V_{\mathbf{x}} \,:\, \mathsf{v}_1 \neq \mathbf{v}_2 \;\wedge\; \left(\mathbf{z}^\top(\mathbf{v}_1 - \mathbf{v}_2) = \mathbf{0}\right)\right] \\
&\leq \Pr\left[\exists \mathbf{v}_1, \mathbf{v}_2 \in V_{\mathbf{x}} \,:\, \mathsf{v}_1 \neq \mathbf{v}_2 \;\wedge\; \left(\mathbf{z}^\top(\mathbf{v}_1 - \mathbf{v}_2) = \mathbf{0} \mod q\right)\right] \\
&\leq \sum\nolimits_{\mathbf{v}_1, \mathbf{v}_2 \in V_{\mathbf{x}} : \mathbf{v}_1 \neq \mathbf{v}_2} \Pr\left[\mathbf{z}^\top(\mathbf{v}_1 - \mathbf{v}_2) = \mathbf{0} \mod q\right] \\
&= \sum\nolimits_{\mathbf{v}_1, \mathbf{v}_2 \in V_{\mathbf{x}} : \mathbf{v}_1 \neq \mathbf{v}_2} \frac{1}{q} \\
&\leq \frac{|V_{\mathbf{x}}|^2}{2} \cdot \frac{1}{q} \;\leq\; \frac{1}{2} \cdot 4(n^2+n)^2 \cdot \frac{1}{4(n^2+n)^2} \;=\; \frac{1}{2}.
\end{aligned}
$$

The second equality follows as $\mathbf{z}^\top(\mathbf{v}_1 - \mathbf{v}_2)$ is zero modulo $p$ iff it is zero over the integers, since $|\mathbf{z}^\top(\mathbf{v}_1 - \mathbf{v}_2)| \leq 2qn < p/2$. The first inequality follows as equality over the integers implies it modulo $q$. The second is a union bound. The third equality follows as the function from $\mathbb{Z}_q^n \to \mathbb{Z}_q$ mapping $\mathbf{v} \mapsto \mathbf{z}^\top\mathbf{v}$ with ($\mathbf{z}$ mod $q$) uniform over $\mathbb{Z}_q^n$ is a universal linear hash [CW79]. □

# 5 Positive Results

## 5.1 Weak Unbiasability of [GHKW17]

In Section 3.4 we proved that the construction in [GHKW17] (and by extension the one in [Bit17], being the two identical) does not achieve the strongest unbiasability notion when two or more verification keys are involved. In this section we show that using a puncturable PRF from [BKW17, §4.1], the scheme in [GHKW17, §4.4] can be proven to be weakly unbiasable assuming sub-exponential hardness of the underlying primitives.

We begin recalling the construction in [BKW17, §4.1]. Their only building blocks are two *private*[16] puncturable PRFs $(\mathsf{PPRF.Gen}_i, \mathsf{IPF.Eval}_i, \mathsf{IPF.Punct}_i)$ for $i \in \{0,1\}$ with domain and range $\mathcal{X}_i, \mathcal{Y}_i$ such that $\mathcal{Y}_0 \subseteq \mathcal{X}_1$ and $\mathcal{Y}_1 \subseteq \mathcal{X}_0$. Given this they construct an *invertible* puncturable PRF. In our context we can ignore the inversion procedure, and present it only as an *injective puncturable PRF* as follows:

- $\mathsf{IPF.Gen}(1^\lambda)$: samples keys $k_0 \leftarrow^\$ \mathsf{PPRF.Gen}_0(1^\lambda)$ and $k_1 \leftarrow^\$ \mathsf{PPRF.Gen}_1(1^\lambda)$ and returns $k = (k_0, k_1)$
- $\mathsf{IPF.Eval}(k, x)$: calling $k = (k_0, k_1)$ and $F_{i,K}(\cdot) = \mathsf{IPF.Eval}_i(K, \cdot)$, it returns

$$y \;=\; (F_{0,k_0}(x),\, x \oplus F_{1,k_1}(F_{0,k_0}(x)))\,.$$

- $\mathsf{IPF.Punct}(k, x^*)$: calling $z^* = F_{0,k_0}(x^*)$, it returns $(k_0^*, k_1^*)$ with

$$k_0^* \leftarrow^\$ \mathsf{IPF.Punct}(k_0, x^*), \quad k_1^* \leftarrow^\$ \mathsf{IPF.Punct}(k_1, x^*).$$

[16] Such that the punctured key does not reveal the removed point. Note that actually only the second PRF needs to be privately puncturable.

In [BKW17, Theorem 4.3] it is proven that, if $|\mathcal{X}_0|/|\mathcal{Y}_1| = \mathsf{negl}(\lambda)$ then the puncturable PRF above is *selectively*-secure, a notion implying the one required in Goyal et al's construction (see [GHKW17, Definition 2.7]) Moreover, we will require $\mathcal{X}_0$, the first PRF domain, to be exponentially large, i.e. $|\mathcal{X}_0| = \Omega(2^\lambda)$.

**Theorem 7.** *The construction in [GHKW17, §4.4] with the (sub-exponentially secure) puncturable PRF* $(\mathsf{IPF.Gen}, \mathsf{IPF.Eval}, \mathsf{IPF.Punct})$*, a (sub-exponentially secure) statistically sound NIWI* $(\mathsf{NIWI.P}, \mathsf{NIWI.V})$ *and a (sub-exponentially secure) perfectly binding and computationally hiding commitment scheme* $(\mathsf{C.Com}, \mathsf{C.Vfy})$*, is a weakly unbiasable VRF.*

*Proof.* Correctness, uniqueness and pseudorandomness are implied by [GHKW17, Theorem 4.2] and complexity leveraging. Next, we prove it to be weakly unbiasable.

To any public key $\mathsf{vk} = (c_1, c_2, c_3)$ we associate three functions $f_{\mathsf{vk},j}$ so that if there exists $k$ and $r$ such that $\mathsf{C.Vfy}(c_j, k, r) = 1$ then $f_{\mathsf{vk},j}(x) = \mathsf{IPF.Eval}(k, x)$ for all $x \in \mathcal{X}_0$. Otherwise $f_{\mathsf{vk},j}(x) = \perp$. Note $f_{\mathsf{vk},j}$ is well defined as the commitment is perfectly binding. Moreover, we define $S_{\mathsf{vk}}$ the set of points in which at least two functions agrees on a non-error value and $F_{\mathsf{vk}} : S_{\mathsf{vk}} \to \{0,1\}^*$ the majority voting of $f_{\mathsf{vk},1}, f_{\mathsf{vk},2}, f_{\mathsf{vk},3}$ defined over $S_{\mathsf{vk}}$. Formally

$$\begin{aligned} S_{\mathsf{vk}} &= \{x \in \mathcal{X}_0 \mid \exists i_1, i_2 \,:\, i_1 \neq i_2,\ f_{\mathsf{vk},i_1}(x) = f_{\mathsf{vk},i_2}(x) \neq \perp\} \\ F_{\mathsf{vk}}(x) &= y \ \Leftrightarrow\ \exists i_1, i_2 \,:\, i_1 \neq i_2,\ f_{\mathsf{vk},i_1}(x) = f_{\mathsf{vk},i_2}(x) = y \end{aligned}$$

$F_{\mathsf{vk}}$ is a well defined function over $S_{\mathsf{vk}}$ as any $x \in S_{\mathsf{vk}}$ admits at least two agreeing functions. Next, we claim $F_{\mathsf{vk}}$ is injective. Note that by construction, the IPF is injective for any key, even for malformed ones as we assume the inner PRFs to default to 0 whenever they would otherwise return $\perp$. Hence $f_{\mathsf{vk},j}$ is either injective, or identically $\perp$ over the whole domain. Finally, for any $x_1, x_2 \in S_{\mathsf{vk}}$ we have $F_{\mathsf{vk}}(x_1) = F_{\mathsf{vk}}(x_2) \Rightarrow \exists i_1, i_2, j_1, j_2 \in \{1,2,3\}$ :

$$i_1 \neq i_2,\ j_1 \neq j_2,\ f_{\mathsf{vk},i_1}(x_1) = f_{\mathsf{vk},i_2}(x_1) = f_{\mathsf{vk},j_1}(x_2) = f_{\mathsf{vk},j_2}(x_2) \neq \perp .$$

By the pigeonhole principle, as $\{i_1, i_2\}$ and $\{j_1, j_2\}$ are subsets of $\{1,2,3\}$ we can assume without loss of generality $i_1 = j_1$. So $f_{\mathsf{vk},i_1}(x_1) = f_{\mathsf{vk},i_1}(x_2) \neq \perp$ which implies $x_1 = x_2$. Hence $F_{\mathsf{vk}}$ is injective.

We can now prove weak unbiasability. Let $\mathcal{A}$ be a $\mathsf{PPT}$ adversary such that $\mathcal{A}(1^\lambda) \to (\mathsf{vk}, y^*)$ which we assume to be computed deterministically[17], $x^* \leftarrow^{\$} \mathcal{X}_0$

[17] up to assuming $\mathcal{A}$ to be non-uniform.

be the challenge input and $(y, \pi)$ the final response. We have that

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A}}(\lambda) &= \Pr\left[y = y^*,\ \mathsf{VRF.Vfy}(\mathsf{vk}, x^*, y, \pi) = 1\right] \\
&= \Pr\left[y = y^*, \mathsf{NIWI.V}(c_1, c_2, c_3, x^*, y, \pi) = 1\right] \\
&\leq \Pr\left[y = y^*, y = F_{\mathsf{vk}}(x^*), x^* \in S_{\mathsf{vk}}\right] + \mathsf{negl}(\lambda) \\
&\leq \Pr\left[y^* = F_{\mathsf{vk}}(x^*), x^* \in S_{\mathsf{vk}}\right] + \mathsf{negl}(\lambda) \\
&= \Pr\left[y^* = F_{\mathsf{vk}}(x^*) | x^* \in S_{\mathsf{vk}}\right] \cdot \frac{|S_{\mathsf{vk}}|}{|\mathcal{X}_0|} + \mathsf{negl}(\lambda) \\
&\leq \frac{1}{|F_{\mathsf{vk}}(S_{\mathsf{vk}})|} \cdot \frac{|S_{\mathsf{vk}}|}{|\mathcal{X}_0|} + \mathsf{negl}(\lambda) \ = \ \frac{1}{|\mathcal{X}_0|} + \mathsf{negl}(\lambda)
\end{aligned}$$

where in the first inequality we used the fact that

$$\Pr\left[(x^* \notin S_{\mathsf{vk}} \vee F_{\mathsf{vk}}(x) \neq y),\ \mathsf{NIWI.V}(c_1, c_2, c_3, x^*, y, \pi) = 1\right] \leq \mathsf{negl}(\lambda)$$

due to the statistical soundness of the given NIWI, and in the last, the fact that under the given condition $x^*$ is uniform over $S_{\mathsf{vk}}$ and $F_{\mathsf{vk}}$ is injective. □

### 5.2 Dodis-Yampolskiy construction in the GBGM

In Section 3.1, we proved that a VRF with a polynomial-sized image and a super-polynomial-sized output space cannot achieve weak unbiasability. This also affects the construction proposed in [DY05], as its security is proven only for polynomial-sized input space. We observe that if we instantiate the construction with input space $\mathcal{X} = \mathbb{Z}_p$, we can easily prove that it is a VRB (i.e. it satisfies unbiasability on independent points). However, with this input space, we are unable to prove the pseudorandomness in the standard model.

For this reason, in this section we show that [DY05] with input space $\mathcal{X} = \mathbb{Z}_p$ and output space $\mathcal{Y} = \mathbb{G}_T$ is a VRF in the Generic Bilinear Group Model (GBGM).

**Theorem 8.** *The construction in [DY05] with input space $\mathcal{X} = \mathbb{Z}_p$ and output space $\mathcal{Y} = \mathbb{G}_T$ is a verifiable random function in the generic bilinear group model.*

*Proof.* It is trivial to show that correctness and unique provability properties are satisfied. We thus concentrate on pseudorandomness.

Let $\mathcal{A}$ be an adversary against pseudorandomness. In the following we describe the usual simulator $\mathcal{S}(1^\lambda)$ so that the view of $\mathcal{A}$ interacting with $\mathcal{S}$ will be statistically close to its view when executed in the pseudorandomness experiment respectively with challenge bit $b = 0$ and $b = 1$, implying the two distribution cannot be distinguished.

$\mathcal{S}$ lazily maintains two injective random functions $\xi_0, \xi_1$ taking value over the field or rational functions in two variables $\xi_b : \mathbb{Z}_p(T, Y) \to \{b\} \times \{0,1\}^\ell$, and two sets of rational polynomials $L_0, L_1 \subseteq \mathbb{Z}_p(T, Y)$. Initially $L_0 = T$ and $L_1 = \varnothing$. Then it executes $\mathcal{A}$ on input $\xi_0(T)$ (i.e. a label standing for the verification key $[\alpha]$) and respond to each query from $\mathcal{A}$ as follows:

**Group Addition.** On input two string $s_1, s_2 \in \{b\} \times \{0,1\}^\ell$, if there exists $f_1, f_2 \in L_b$ such that $\xi_b(f_i) = s_i$, it computes $f_3 = f_1 + f_2$, adds it to $L_b$ and returns $\xi_b(f_3)$. Otherwise, it returns $\perp$.

**Bilinear Pairing.** Given two strings $s_1, s_2 \in \{0\} \times \{0,1\}^\ell$ (i.e. in $\mathbb{G}$) for which there exists $f_1, f_2 \in L_0$ such that $\xi_0(f_i) = s_i$, it computes $f_3 = f_1 \cdot f_2$, stores it in $L_1$ and returns $\xi_1(f_3)$. Otherwise, it returns $\perp$.

**VRF Evaluation.** On input $m_i \neq m^*$ (if $m^*$ is already defined), stores $1/(T + m_i)$ in both $L_0$ and $L_1$ and returns $\xi_0(1/(T + m_i))$ and $\xi_1(1/(T + m_i))$, respectively representing the proof and VRF value. Otherwise, it returns $\perp$.

**VRF Challenge.** On input $m^*$, if $m^* \neq m_i$, stores the polynomial $Y$ in $L_1$ only and returns $\xi_1(1/(T + m_i))$, i.e. a representation of the VRF challenge value.

Before diving into the proof, we will need the following Claim, which readily follows by induction on the queries performed by $\mathcal{A}$.

**Claim 2.** *At the end of the execution, given $m_1, \ldots, m_q$, all polynomials in $L_0, L_1$ are of the form $F(T, 1/(T + m_1), \ldots, 1/(T + m_q)) + c \cdot Y$ with $F \in \mathbb{Z}_p[z_0, z_1, \ldots, z_q]$ of degree $\deg(F) \leq 2$ and $c \in \mathbb{Z}_p$.*

As usual, we observe that in the GGM, the probability that a label not obtained from the oracles represents a group element is negligible, even conditioning on a polynomially-bounded number of queries. This is true as we set the model with $\ell > \log p + \lambda$. Hence, in the following we will assume that when executed in the real experiment, all $\mathcal{A}$ queries to unobserved labels are always answered with $\perp$. Moreover, without loss of generality we assume $\alpha \notin \{m_1, \ldots, m_q, m^*\}$. Indeed, any adversary querying an evaluation in $\alpha$ (by lucky guessing) can be simulated by a mildly more inefficient one which first checks whether $\mathsf{vk} = [\alpha]$ and if so returns 1 if the challenge value is $[1/(\alpha + m^*)]_T$.

Conditioning on the above we will show that $\mathcal{S}$ simulates the pseudorandomness challenger when $b = 0$, i.e. when $y$ is a random group element, and when $b = 1$, i.e. when $y$ is the actual VRF output.

*Random Output.* $\mathcal{S}(1^\lambda)$ perfectly simulates the view of $\mathcal{A}$ with when $b = 0$ iff, calling $\alpha$ and $y$ the discrete log respectively of $\mathsf{vk}$ and the challenge output, then for any pair of distinct rational functions $f_i, f_j \in L_1 \cup L_2$ we have that $f_i(\alpha, y) \neq f_j(\alpha, y)$. Indeed in this case, is easy to show by induction that the $i$-th group element obtained through any call by $\mathcal{A}$ (upon conditioning on the first $i - 1$ calls) is equal to a previously queried group element when executed with $\mathcal{S}(1^\lambda)$ if and only if it is so in the pseudorandomness experiment.

We thus bound the probability that $f_i(\alpha, y) = f_j(\alpha, y)$ for $f_i \neq f_j$. Let $f = f_i - f_j \neq 0$. By Claim 2 there exists $F \in \mathbb{Z}_p[z_0, \ldots, z_q]$ of degree 2 and $c \in \mathbb{Z}_p$ such that $f(T, Y) = F(T, 1/(T+m_1), \ldots, 1/(T+m_q)) + c \cdot Y$. In particular $f \neq 0$ iff

$$F^*(T, Y) \coloneqq \prod_{i=1}^{q} (T + m_i)^2 \cdot \left( F\left(T, \frac{1}{T + m_1}, \ldots, \frac{1}{T + m_q}\right) + c \cdot Y \right).$$

is non-zero. Since $F \in \mathbb{Z}_p[T, Y]$ with $\deg(F^*) \leq 2q + 2$ and $F^* \neq 0$, by Schwartz-Zippel we have that for uniformly sampled $y \leftarrow^{\$} \mathbb{Z}_p$ and $\alpha \leftarrow^{\$} \mathbb{Z}_p \setminus \{-m_1 \ldots, -m_q, -m^*\}$[18]

$$\Pr[f(\alpha, y) = 0] \;=\; \Pr[F^*(\alpha, y) = 0] \;\leq\; \frac{\deg(F^*)}{p - (q+1)} \;\leq\; 2 \cdot \frac{2q+2}{p}.$$

A union bound on all pairs of polynomials implies that the probability of a collision occurring is smaller that $q^2(2q+2)/p$, that is negligible.

*Correct VRF Output.* As in the previous case, we have that $\mathcal{S}(1^\lambda)$ perfectly simulates the view of $\mathcal{A}$ with $b = 1$ if and only if, calling $\alpha$ the discrete logarithm of $\mathsf{vk}$, for any distinct pair $f_i, f_j \in L_0 \cup L_1$, then $f_i(\alpha, 1/(\alpha + m^*)) \neq f_j(\alpha, 1/(\alpha + m^*))$.

For any such pair let $f = f_i - f_j \neq 0$. We will show that $f(T, Y)$ is non zero then so is $f(T, 1/(T + m^*))$. Indeed let $F, c$ be the polynomial and constant of Claim 2 for $f$. Then we call $D = \prod_{i=1}^{q}(T + m_i)^2$ and $F^*(T) = D(T) \cdot F(T, 1/(T + m_1), \ldots, 1/(T + m_q))$. Then $D(T) \cdot f(T, Y) = F^*(T) + c \cdot D(T) \cdot Y \in \mathbb{Z}_p[T, Y]$. Toward contradiction assume $f(T, 1/(T + m^*))$ is the zero polynomial. Then $c \neq 0$ or else $F^*(T) = 0$ which implies $f(T, Y) = 0$. Then

$$0 = F^*(T) + \frac{cD(T)}{T + m^*} \quad \Leftrightarrow \quad c \cdot D(T) = -(T + m^*) \cdot F^*(T)$$

which implies that $(T + m^*)$ divides $D(T)$ and in particular $m^* = m_i$ for some $i$, that is a contradiction. Hence $f(T, 1/(T + m^*))$ is non zero. By Schwartz-Zippel, for a uniformly sampled $\alpha \leftarrow^{\$} \mathbb{Z}_p \setminus \{-m_1, \ldots, -m_q, -m^*\}$

$$\begin{aligned}
\Pr[f(\alpha, 1/(\alpha + m^*)) = 0] \;&=\; \Pr[D(\alpha) \cdot f(\alpha, 1/(\alpha + m^*)) = 0] \\
&=\; \Pr\left[F^*(\alpha) + \frac{c \cdot D(\alpha)}{\alpha + m^*} = 0\right] \\
&=\; \Pr[F^*(\alpha)(\alpha + m^*) + c \cdot D(\alpha) = 0] \\
&\leq\; \frac{2q+3}{p - (q+1)} \;\leq\; 2 \cdot \frac{2q+3}{p}.
\end{aligned}$$

A union bound over all possible pairs yields that the collision probability is smaller than $q^2(2q+3)/p$.

*Conclusion.* We thus conclude that the advantage of $\mathcal{A}$ is smaller than

$$\mathsf{Adv}^{\mathsf{rnd}}_{\mathcal{A}}(\lambda) \;\leq\; \frac{q^2(4q+5)}{p} + \mathsf{negl}(\lambda).$$

□

[18] Note that here we inductively assume that no collision happened for previously computed polynomials. Since, up to this point, the view of the experiment and the simulator are the same, the new polynomials are independent from $\alpha, y$ (and hence the values produced by $\mathcal{S}$).

## Acknowledgments

The authors would like to thank Nicholas Brandt for the helpful discussions. This study has been supported by the project "PrepAring cRypTograpHy for privacy-awarE blockchaiN applicatiONs (PARTHENON)" - PRIN 2022 - Finanziato dall'Unione europea - Next Generation EU, Missione 4 Componente 1 - CUP: E53D23007990006. It was also supported in part by the PIAno di inCEntivi per la Ricerca di Ateneo 2024/2026, linea di intervento 1. This work was also partially funded by a research gift from Public Good Crypto.

## References


ACF09. Michel Abdalla, Dario Catalano, and Dario Fiore. Verifiable random functions from identity-based key encapsulation. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 554–571. Springer, Berlin, Heidelberg, April 2009.

ACF14. Michel Abdalla, Dario Catalano, and Dario Fiore. Verifiable random functions: Relations to identity-based key encapsulation and new constructions. *Journal of Cryptology*, 27(3):544–593, July 2014.

BASV23. Jeffrey Burdges, Handan Kılınç Alper, Alistair Stewart, and Sergey Vasilyev. Sassafras and semi-anonymous single leader election. Cryptology ePrint Archive, Paper 2023/031, 2023.

BB04. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Berlin, Heidelberg, August 2004.

BBG05. 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. Springer, Berlin, Heidelberg, May 2005.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Berlin, Heidelberg, August 2004.

BCKL09. Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. Compact e-cash and simulatable VRFs revisited. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009*, volume 5671 of *LNCS*, pages 114–131. Springer, Berlin, Heidelberg, August 2009.

BDE+22. Maxime Buser, Rafael Dowsley, Muhammed F. Esgin, Shabnam Kasra Kermanshahi, Veronika Kuchta, Joseph K. Liu, Raphaël C.-W. Phan, and Zhenfei Zhang. Post-quantum verifiable random function from symmetric primitives in PoS blockchain. In Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen, and Weizhi Meng, editors, *ESORICS 2022, Part I*, volume 13554 of *LNCS*, pages 25–45. Springer, Cham, September 2022.

BGI14. Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, Berlin, Heidelberg, March 2014.

BGK+18. Christian Badertscher, Peter Gazi, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 913–930. ACM Press, October 2018.

BGQR22. Christian Badertscher, Peter Gazi, Iñigo Querejeta-Azurmendi, and Alexander Russell. A composable security treatment of ECVRF and batch verifications. In Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen, and Weizhi Meng, editors, *ESORICS 2022, Part III*, volume 13556 of *LNCS*, pages 22–41. Springer, Cham, September 2022.

BGRV09. Zvika Brakerski, Shafi Goldwasser, Guy N. Rothblum, and Vinod Vaikuntanathan. Weak verifiable random functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 558–576. Springer, Berlin, Heidelberg, March 2009.

Bit17. Nir Bitansky. Verifiable random functions from non-interactive witness-indistinguishable proofs. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 567–594. Springer, Cham, November 2017.

BKW17. Dan Boneh, Sam Kim, and David J. Wu. Constrained keys for invertible pseudorandom functions. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 237–263. Springer, Cham, November 2017.

BMR10. Dan Boneh, Hart William Montgomery, and Ananth Raghunathan. Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 131–140. ACM Press, October 2010.

Boy08. Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56. Springer, Berlin, Heidelberg, September 2008.

Bra25. Nicholas Brandt. Unbiasable verifiable random functions from generic assumptions. Cryptology ePrint Archive, Paper 2025/766, 2025.

BW13. Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, Berlin, Heidelberg, December 2013.

CL07. Melissa Chase and Anna Lysyanskaya. Simulatable VRFs with applications to multi-theorem NIZK. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 303–322. Springer, Berlin, Heidelberg, August 2007.

CM19. Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. *Theoretical Computer Science*, 777:155–183, July 2019.

CW79. J.Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, April 1979.

DGKR18. Bernardo David, Peter Gazi, Aggelos Kiayias, and Alexander Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 66–98. Springer, Cham, April / May 2018.

Dod03. Yevgeniy Dodis. Efficient construction of (distributed) verifiable random functions. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 1–17. Springer, Berlin, Heidelberg, January 2003.

DP07. Yevgeniy Dodis and Prashant Puniya. Feistel networks made public, and applications. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 534–554. Springer, Berlin, Heidelberg, May 2007.

DY05. 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. Springer, Berlin, Heidelberg, January 2005.

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

EKS+21. Muhammed F. Esgin, Veronika Kuchta, Amin Sakzad, Ron Steinfeld, Zhenfei Zhang, Shifeng Sun, and Shumo Chu. Practical post-quantum few-time

verifiable random function with applications to algorand. In Nikita Borisov and Claudia Díaz, editors, *FC 2021, Part II*, volume 12675 of *LNCS*, pages 560–578. Springer, Berlin, Heidelberg, March 2021.

ESLR23. Muhammed F. Esgin, Ron Steinfeld, Dongxi Liu, and Sushmita Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and VRFs. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 484–517. Springer, Cham, August 2023.

FS90. Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, pages 416–426. ACM Press, May 1990.

Fuc14. Georg Fuchsbauer. Constrained verifiable random functions. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 95–114. Springer, Cham, September 2014.

GHKW17. Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 537–566. Springer, Cham, November 2017.

GL89. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.

GS24. Emanuele Giunta and Alistair Stewart. Unbiasable verifiable random functions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part IV*, volume 14654 of *LNCS*, pages 142–167. Springer, Cham, May 2024.

HJ16. Dennis Hofheinz and Tibor Jager. Verifiable random functions from standard assumptions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 336–362. Springer, Berlin, Heidelberg, January 2016.

HW10. Susan Hohenberger and Brent Waters. Constructing verifiable random functions with large input spaces. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 656–672. Springer, Berlin, Heidelberg, May / June 2010.

Jag15. Tibor Jager. Verifiable random functions from weaker assumptions. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 121–143. Springer, Berlin, Heidelberg, March 2015.

Kat17. Shuichi Katsumata. On the untapped potential of encoding predicates by arithmetic circuits and their applications. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 95–125. Springer, Cham, December 2017.

Koh19. Lisa Kohl. Hunting and gathering - verifiable random functions from standard assumptions with short proofs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 408–437. Springer, Cham, April 2019.

KRDO17. Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 357–388. Springer, Cham, August 2017.

Lai23. Yi-Fu Lai. CAPYBARA and TSUBAKI: Verifiable random functions from group actions and isogenies. Cryptology ePrint Archive, Report 2023/182, 2023.

LBM20. Bei Liang, Gustavo Banegas, and Aikaterini Mitrokotsa. Statically aggregate verifiable random functions and application to e-lottery. *Cryptography*, 4(4), 2020.

Ler25. Antonin Leroux. Verifiable random function from the deuring correspondence and higher dimensional isogenies. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 167–194. Springer, 2025.

LLC15. Bei Liang, Hongda Li, and Jinyong Chang. Verifiable random functions from (leveled) multilinear maps. In Michael Reiter and David Naccache, editors, *CANS 15*, LNCS, pages 129–143. Springer, Cham, December 2015.

Lys02. 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. Springer, Berlin, Heidelberg, August 2002.

Mil04. Victor S. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, September 2004.

MR01. Silvio Micali and Leonid Reyzin. Soundness in the public-key model. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 542–565. Springer, Berlin, Heidelberg, August 2001.

MR02. Silvio Micali and Ronald L. Rivest. Micropayments revisited. In Bart Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 149–163. Springer, Berlin, Heidelberg, February 2002.

MRV99. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th FOCS*, pages 120–130. IEEE Computer Society Press, October 1999.

Nie21. David Niehues. Verifiable random functions with optimal tightness. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 61–91. Springer, Cham, May 2021.

NR97. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

Ros18. Razvan Rosie. Adaptive-secure VRFs with shorter keys from static assumptions. In Jan Camenisch and Panos Papadimitratos, editors, *CANS 18*, volume 11124 of *LNCS*, pages 440–459. Springer, Cham, September / October 2018.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Berlin, Heidelberg, May 1997.

Yam17. Shota Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 161–193. Springer, Cham, August 2017.

# A VRF Constructions

Here we recall some VRF constructions on which we show our attacks against unbiasability. For further details we refer to the original papers.

## A.1 [HW10]

Let $\mathcal{X} = \{0,1\}^n$ be the input space of the VRF. The construction works as follows:

- $\mathsf{VRF.Gen}(1^\lambda)$, on input a security parameter $1^\lambda$,
  1. choose a bilinear group $\mathbb{G}$ of prime order $p$;
  2. select random generators $g, h \in \mathbb{G}$;
  3. sample random values $\widetilde{u}, u_0, \ldots, u_n \in \mathbb{Z}_p$;
  4. set $\widetilde{U} = [\widetilde{u}], U_0 = [u_0], \ldots, U_n = [u_n]$;
  5. output $\mathsf{vk} := (\mathbb{G}, p, g, h, \widetilde{U}, U_0, \ldots, U_n), \mathsf{sk} := (\mathbb{G}, p, g, h, \widetilde{u}, u_0, \ldots, u_n)$.
- $\mathsf{VRF.Eval}(\mathsf{sk}, x)$, on input a secret key $\mathsf{sk} = (\mathbb{G}, p, g, h, \widetilde{u}, u_0, \ldots, u_n)$ and a preimage $x \in \{0,1\}^n$,
  1. parse $x$ as $x = x_1 x_2, \ldots, x_n$
  2. compute $y = e(\left[\widetilde{u} u_0 \prod_{j=1}^n u_j^{x_j}\right], h)$;
  3. for each $i \in \{1, \ldots, n\}$, compute $\pi_i = \left[\widetilde{u} \prod_{j=1}^i u_j^{x_j}\right]$;
  4. compute $\pi_0 = [\widetilde{u} u_0 \prod_{j=1}^n u_j^{x_j}]$;
  5. output $y, \pi := (\pi_0, \pi_1, \ldots, \pi_n)$.
- $\mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi)$ on input a verification key $\mathsf{vk} = (\mathbb{G}, p, g, h, \widetilde{U}, U_0, \ldots, U_n)$, a preimage $x \in \{0,1\}^n$, an output $y \in \mathbb{G}_T$ and a proof $\pi = (\pi_0, \pi_1, \ldots, \pi_n)$,
  1. verify that the proof $\pi = (\pi_0, \pi_1, \ldots, \pi_n)$ contains legal encodings of elements in $\mathbb{G}$.
  2. Check that
  $$e(\pi_1, [1]) = \begin{cases} e([1], \widetilde{U}) & \text{if } x_1 = 0; \\ e(U_1, \widetilde{U}) & \text{otherwise.} \end{cases}$$
  3. For $i = 2$ to $n$, check
  $$e(\pi_i, [1]) = \begin{cases} e(\pi_{i-1}, [1]) & \text{if } x_i = 0; \\ e(\pi_{i-1}, U_i) & \text{otherwise.} \end{cases}$$
  4. Finally, check that
  $$e(\pi_0, [1]) = e(\pi_n, U_0) \quad \text{and} \quad e(\pi_0, h) = y.$$
  5. Output 1 if all checks verify, 0 otherwise.

### A.2 [GHKW17]

Here we recall the construction proposed in [GHKW17]. This construction requires admissible hash functions [BB04], perfectly binding commitments (with no setup assumptions), constrained pseudorandom functions [BW13, BGI14] and non-interactive witness indistinguishable proofs [FS90]. We recall the syntax of these primitives. We refer to the original papers for the formal definitions.

**Definition 7 (NIWI).** *A pair of* PPT *algorithms* $(\mathsf{NIWI.P}, \mathsf{NIWI.V})$ *is a non-interactive witness indistinguishable (NIWI) for a language* $\mathcal{L} \in \mathbf{NP}$ *with witness relation* $\mathcal{R}$ *if it satisfies the following conditions:*

- *(Perfect Completeness) For all* $(x, w)$ *such that* $\mathcal{R}(x, w) = 1$,

$$\Pr\left[\mathsf{NIWI.V}(x, \pi) = 1 : \pi \leftarrow \mathsf{NIWI.P}(x, w)\right] = 1.$$

- *(Statistical Soundness) For every* $x \notin \mathcal{L}$ *and* $\pi \in \{0,1\}^*$,

$$\Pr\left[\mathsf{NIWI.V}(x, \pi) = 1\right] \leq 2^{-\Omega(|x|)}.$$

- *(Witness Indistinguishability) For any sequence* $\mathcal{I} = \{(x, w_1, w_2) : \mathcal{R}(x, w_1) = 1 \wedge R(x, w_2)\}$

$$\{\pi_1 : \pi_1 \leftarrow \mathsf{NIWI.P}(x, w_1)\}_{(x,w_1,w_2)\in\mathcal{I}} \approx_c \{\pi_2 : \pi_2 \leftarrow \mathsf{NIWI.P}(x, w_2)\}_{(x,w_1,w_2)\in\mathcal{I}}$$

**Definition 8 (Commitment Scheme with no setup assumptions).** *A commitment scheme with message space* $\mathcal{M}$*, randomness space* $\mathcal{R}$ *and commitment space* $\mathcal{C}$ *consists of two polynomial time algorithms* $\mathsf{CS.Commit}, \mathsf{CS.Verify}$ *with the following syntax.*

- $\mathsf{C.Com}(1^\lambda, m; r)$ *on input a security parameter* $1^\lambda$*, a message* $m \in \mathcal{M}$ *and a random coin* $r \in \mathcal{R}$*, outputs a commitment* $c \in \mathcal{C}$*.*
- $\mathsf{C.Vfy}(m, c, o)$ *on input a message* $m \in \mathcal{M}$*, a commitment* $c \in \mathcal{C}$ *and an opening* $o \in \mathcal{R}$*, outputs a bit* $b \in \{0, 1\}$*.*

**Definition 9 (Constrained Pseudorandom Functions).** *A constrained pseudorandom function* CPRF *with domain* $\mathcal{X}$*, range* $\mathcal{Y}$*, key space* $\mathcal{K}$ *and constrained key space* $\mathcal{K}^c$ *for a family of admissible hash compatible constrains* $\mathcal{C}$ *consists of three algorithms* $\mathsf{CPRF.Gen}, \mathsf{CPRF.Cons}, \mathsf{CPRF.Eval}$ *with the following syntax.*

- $\mathsf{CPRF.Gen}(1^\lambda)$ *on input a security parameter* $1^\lambda$*, outputs a PRF key* $k \in \mathcal{K}$*.*
- $\mathsf{CPRF.Cons}(k, z)$ *on input a master PRF key* $k \in \mathcal{K}$ *and a constraint* $z \in \mathcal{X}$*, outputs a constrained key* $k_z \in \mathcal{K}^c$*.*
- $\mathsf{CPRF.Eval}(k, x)$ *on input a PRF key* $k$ *(master or constrained) and a point* $x \in \mathcal{X}$*, outputs a* $y \in \mathcal{Y}$*.*

We also recall the notion of *admissible hash function*, which is used in this construction as well as in other constructions we discuss in this paper.

**Definition 10 (Admissible Hash Function, [Koh19]).** *Let $n$ be polynomial in $\lambda$ and $\Sigma$ an alphabet of size $\sigma := |\Sigma|$ polynomial in $\lambda$. Let*

$$\mathsf{AHF} : \{0,1\}^L \to \Sigma^n,$$

*$Y \in \Sigma^n$ and $I \subseteq \{1,\ldots,n\}$ define the partitioning*

$$\mathcal{Y} := \{x \in \{0,1\}^L | \mathsf{AHF}(x)_j = Y_j \text{ for all } j \in I\} \text{ and } \mathcal{Z} := \{0,1\}^L \setminus \mathcal{Y}.$$

*We say that* AHF *is* Q-admissible *if there exists a PPT algorithm* AHF.Part *that on input $(1^\lambda, Q)$ returns a value $Y \in \Sigma^n$ and a set of indices $I \subseteq \{1,\ldots,n\}$, such that for any $x^{(1)},\ldots,x^{(Q)}$ we have*

$$\Pr[x^* \in \mathcal{Y} \wedge x^{(\nu)} \in \mathcal{Z} \forall \nu \in \{1,\ldots,Q\}] \geq 1/\mathsf{poly}(\lambda),$$

*where the probability is taken over the random coins of* AHF.Part*. We say that* AHF *is an* admissible hash function *(AHF) if* AHF *is $Q$-admissible for all $Q$ that are polynomially bounded in $\lambda$.*

**The construction.** Let $\mathcal{X} = \{0,1\}^n$ be the input space and $\mathcal{Y} = \{0,1\}^m$ the output space of the VRF. The construction works as follows:

- $\mathsf{VRF.Gen}(1^\lambda)$
  1. Sample a CPRF key $k \leftarrow^\$ \mathsf{CPRF.Gen}(1^\lambda)$.
  2. Sample $r_i \leftarrow^\$ \mathcal{R}$ and compute $c_i \leftarrow \mathsf{CS.Commit}(1^\lambda, k; r_i)$ for $i \in \{1,2,3\}$.
  3. Output $(\mathsf{sk} := (k, \{(c_i, r_i)\}_{i=1}^3), \mathsf{vk} := (c_1, c_2, c_3))$.
- $\mathsf{VRF.Eval}(\mathsf{sk}, x)$
  1. Compute $y \leftarrow \mathsf{CPRF.Eval}(k, \mathsf{AHF}(x))$.
  2. Compute a NIWI proof $\pi$ for the statement $c_1, c_2, c_3, x, y \in \mathcal{L}$ using NIWI prover algorithm $\mathsf{NIWI.P}$ with $i = 1, j = 2, k, k, r_1, r_2$ as the witness.
  3. Output $(y, p)$
- $\mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi)$
  1. Run the NIWI verifier algorithm to check $\pi$ as $\mathsf{NIWI.V}((c_1, c_2, c_3, x, y), \pi)$.
  2. Output the bit $b$ returned by $\mathsf{NIWI.V}$.

### A.3 [Koh19]

We now recall the construction presented in [Koh19]. This construction is based on the work of [HJ16] but uses a technique called *hunting and gathering*, combined with the partitioning techniques of [Bit17], in order to have a proof that is shorter than the one in [HJ16]. In particular, the author presents a construction of an adapatively programmable verifiable vector hash function and applies the generic transformation of [HJ16] to obtain a verifiable random function.
We refer to [HJ16] for the notion of *certified bilinear group generator*, i.e. a generator $(\mathsf{BG.Gen}, \mathsf{BG.Vfy})$ which, roughly speaking, outputs the description of two groups $\mathbb{G}$ and $\mathbb{G}_T$ and a pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, as well as a deterministic

polynomial time algorithm $\mathsf{BG.Vfy} = (\mathsf{BG.Vfy}_1, \mathsf{BG.Vfy}_2)$ which can be used to validate the parameters (i.e. check if the string $\mathcal{G}$ it takes as input is of the form it is expected to be) and to check that each element in $\mathbb{G}$ has unique representation, which can be efficiently recognized.
We give below the definition of a *verifiable vector hash function*. We will not write explicitly the requirements of correctness and unique provability, which require, respectively, that if $vk, [\mathbf{v}], \pi$ and $x$ are honestly generated, then the verification algorithm will always accept, and that for any possible verification key $vk$ and for any possible input $x$, there is at most one output which can be proven to be correct.

**Definition 11 (Verifiable Vector Hash Function).** *Let* $\mathsf{BG.Gen}$ *be a bilinear group generator and let* $d \in \mathbb{N}$. *A* verifiable vector hash function (VVHF) *for* $\mathsf{BG.Gen}$ *with domain* $\{0,1\}^L$ *and range* $\mathbb{G}^d$ *is a tuple of PPT algorithms* $\mathsf{VVHF} := (\mathsf{VVHF.Gen}, \mathsf{VVHF.Eval}, \mathsf{VVHF.Vfy})$ *with the following properties:*

- $\mathsf{VVHF.Gen}(\mathcal{G})$ *for* $\mathcal{G} \leftarrow \mathsf{BG.Gen}(1^\lambda)$, *outputs a verification key* $\mathsf{vk}$ *and a secret key* $\mathsf{sk}$.
- $\mathsf{VVHF.Eval}(\mathsf{sk}, x)$ *for a secret key* $\mathsf{sk}$ *and* $x \in \{0,1\}^L$, *outputs a function value* $[\mathbf{v}] \in \mathbb{G}^d$ *and a corresponding proof of correctness* $\pi$.
- $\mathsf{VVHF.Vfy}(\mathsf{vk}, [\mathbf{v}], \pi, x)$ *is a deterministic algorithm returning a bit* $b \in \{0,1\}$.

We can now recall the construction of $\mathsf{VVHF}$ proposed in [Koh19].

**The construction of VVHF.** Let $\mathsf{AHF} : \{0,1\}^L \to \Sigma^n$ together with $\mathsf{AHF.Part}$ be an admissible hash function and $\ell$ be un upper bound on the set $I \subseteq \{1, \ldots, n\}$ output by $\mathsf{AHF.Part}(1^\lambda, Q)$ (for $Q$ polynomial in $\lambda$). Let $\mathsf{BG.Gen}$ be a certified bilinear group generator and let $\mathcal{G} \leftarrow^{\$} \mathsf{BG.Gen}(1^\lambda)$. We define a verifiable vector hash function $\mathsf{VVHF} := (\mathsf{VVHF.Gen}, \mathsf{VVHF.Eval}, \mathsf{VVHF.Vfy})$ as follows:

- $\mathsf{VVHF.Gen}(\mathcal{G})$ on input group parameters $\mathcal{G}$:
  1. Samples matrices $\mathbf{M}_{i,j,k} \leftarrow^{\$} \mathbb{Z}_p^{d \times d}$ for all $i \in \{1, \ldots, \ell\}$, $j \in \{1, \ldots, n\}$ and $k \in \Sigma$ uniformly at random.
  2. Samples a vector $\mathbf{u} \leftarrow^{\$} \mathbb{Z}_p^d \setminus \{0\}$.
  3. Outputs the secret and verification key:

$$\mathsf{sk} := \left((\mathbf{M}_{i,j,k})_{i \in \{1,\ldots,\ell\}, j \in \{i,\ldots,n\}, k \in \Sigma}), \mathbf{u}\right)$$
$$\mathsf{vk} := \left([\mathbf{M}_{i,j,k}]_{i \in \{1,\ldots,\ell\}, j \in \{i,\ldots,n\}, k \in \Sigma}), [\mathbf{u}]\right).$$

- $\mathsf{VVHF.Eval}(\mathsf{sk}, x)$ on input a secret key $\mathsf{sk}$ and a preimage $x \in \{0,1\}^L$:
  1. Computes the admissible hash value $X := \mathsf{AHF}(x) \in \Sigma^n$ of $x$.
  2. For each $\iota \in \{1, \ldots, \ell\}$, computes the vector

$$\mathbf{v}_\iota := \left(\prod_{i=1}^{\iota} \sum_{j=1}^{n} \mathbf{M}_{i,j,X_j}\right)^\top \mathbf{u}.$$

  3. Outputs the image $[\mathbf{v}] := [\mathbf{v}_\ell]$ and the proof $\pi := [\mathbf{v}_1, \ldots, \mathbf{v}_{\ell-1}]$.

– $\mathsf{VVHF.Vfy}(\mathsf{vk}, x, [\mathbf{v}], \pi)$ on input a verification key $\mathsf{vk}$, a preimage $x$ with image $\mathsf{AHF}(x) = X = (X_1, \ldots, X_n) \in \Sigma^n$, an image $[\mathbf{v}] = [\mathbf{v}_\ell]$ and a proof $\pi = [\mathbf{v}_1, \ldots, \mathbf{v}_{\ell-1}]$:
  1. Checks whether for all $\iota \in \{1, \ldots, \ell\}$ and $[\mathbf{v}_0] := [\mathbf{u}]$ it holds
$$e([\mathbf{1}_d], [\mathbf{v}_\iota]) = e\left(\sum\nolimits_{j=1}^{n} [\mathbf{M}_{\iota,j,X_j}]^\top, [\mathbf{v}_{\iota-1}]\right)$$
  and returns 1 if and only if this is the case.

**The generic transformation.** We now recall the generic transformation to obtain a verifiable random function from a verifiable vector hash function with adaptive programmability.

Let $\mathsf{BG.Gen}$ be a certified bilinear group generator. Let $\mathsf{VVHF} := (\mathsf{VVHF.Gen}, \mathsf{VVHF.Eval}, \mathsf{VVHF.Vfy})$ be a vector hash function. We define $\mathsf{VRF} := (\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Vfy})$ as follows:

– $\mathsf{VRF.Gen}(1^\lambda)$ on input the security parameter:
  1. Runs $\mathcal{G} \leftarrow \mathsf{BG.Gen}(1^\lambda)$.
  2. Runs $(\mathsf{sk}', \mathsf{vk}') \leftarrow \mathsf{VVHF.Gen}(\mathcal{G})$.
  3. Chooses a random vector $\mathbf{w} \leftarrow^{\$} (\mathbb{Z}_p^*)^d$.
  4. Defines $\mathsf{sk} := (\mathcal{G}, \mathsf{sk}', \mathbf{w})$ and $\mathsf{vk} := (\mathcal{G}, \mathsf{vk}', [\mathbf{w}])$.
  5. Outputs $(\mathsf{vk}, \mathsf{sk})$.
– $\mathsf{VRF.Eval}(\mathsf{sk}, x)$ on input a secret key $\mathsf{sk} = (\mathcal{G}, \mathsf{sk}', \mathbf{w})$ and a preimage $x \in \{0,1\}^L$:
  1. Runs $([\mathbf{v}], \pi') \leftarrow \mathsf{VVHF.Eval}(\mathsf{sk}', x)$.
  2. Computes the function value $y$ and an additional proof $[z] \in \mathbb{G}^d$ as
$$y := \sum_{i=1}^{d} \left\lfloor \frac{v_i}{w_i} \right\rfloor \text{ and } [\mathbf{z}] := \left[ \left( \frac{v_1}{w_1}, \frac{v_2}{w_2}, \ldots, \frac{v_d}{w_d} \right)^\top \right].$$
  3. Sets $\pi = ([\mathbf{v}], \pi', [\mathbf{z}])$.
  4. Outputs $(y, \pi)$.
– $\mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi)$ on input a verification key $\mathsf{vk}$, a preimage $x$, a VRF output $y$ and its corresponding proof $\pi$, outputs 1 if and only if all of the following properties are verified:
  1. The verification key $\mathsf{vk}$ has the form $\mathsf{vk} = (\mathcal{G}, \mathsf{vk}', [\mathbf{w}])$ such that $[\mathbf{w}] = [w_1, \ldots, w_d]$ and the bilinear group parameters and the group elements contained in $\mathsf{vk}$ are valid, which can be checked by running $\mathsf{BG.Vfy}_1$ and $\mathsf{BG.Vfy}_2$.
  2. The input $x$ is an element in $\{0,1\}^L$.
  3. The proof $\pi$ has the form $\pi = ([\mathbf{v}], \pi', [\mathbf{z}])$ with $\mathsf{VVHF.Vfy}(vk', [\mathbf{v}], \pi', x) = 1$ and both vector $[\mathbf{v}]$ and $[\mathbf{z}]$ contain only validly-encoded group elements, which can be checked by running $\mathsf{BG.Vfy}_2$.
  4. It holds that $[z_i] = [v_i/w_i]$ for all $i \in \{i, \ldots, d\}$ and $y = [\sum_{i=1}^{d} v_i/w_i]$. This can be checked by testing
$$e([z_i], [w_i]) \stackrel{?}{=} e([v_i], [1]) \quad \forall i \in \{1, \ldots, d\} \quad \text{and} \quad y \stackrel{?}{=} [z_1] + \ldots + [z_n].$$

### A.4 [DY05]

[DY05] proposed a VRF construction based on decisional bilinear Diffie-Hellman inversion assumption. In the standard model, the construction is proven secure only for polynomial-sized input space. The construction works as follows:

- $\mathsf{VRF.Gen}(1^\lambda)$, on input a security parameter $1^\lambda$,
  1. choose a bilinear group $\mathbb{G}$ of prime order $p$;
  2. select random generator $g \in \mathbb{G}$;
  3. sample a random $s \in \mathbb{Z}_p^*$;
  4. output $\mathsf{vk} := (\mathbb{G}, p, g, [s]), \mathsf{sk} := (\mathbb{G}, p, g, s)$.
- $\mathsf{VRF.Eval}(\mathsf{sk}, x)$, on input a secret key $\mathsf{sk} = (\mathbb{G}, p, g, s)$ and a preimage $x \in \mathbb{Z}_p^*$,
  1. compute $y = [1/(x+s)]_T$;
  2. compute $\pi = [1/(x+s)]$;
  3. output $(y, \pi)$.
- $\mathsf{VRF.Vfy}(\mathsf{vk}, x, y, \pi)$, on input a verification key $\mathsf{vk} = (\mathbb{G}, p, g, g^s)$, a preimage $x \in \mathbb{Z}_p^*$, an output $y \in \mathbb{G}_T$ and a proof $\pi \in \mathbb{G}$,
  1. check if $e([x] + [s], \pi) = [1]_T$;
  2. check if $y = e([1], \pi)$;
  3. output 1 if both checks succeed, 0 otherwise.

## B Other Attacks

### B.1 Specialized Attack for Kohl's VRF

In Section 3.3 we proved that [Koh19] is subject to a related key attack breaking general unbiasability with 2 or more keys. In this section we actually prove that in the specific case of Kohl's construction, a stronger attack against weak unbiasability can be mounted. The core idea, inspired by the elegant proof of pseudorandomness provided in [Koh19], will be to choose *structured* matrices $M_{i,j,k}$. In particular, we will generated them so that the only ones actually affecting the VRF output will be those with $j = n$. As a consequence, the VRF output will depend only on the last symbol of $\mathsf{AHF}(x)$ thus making the VRF's image polynomially small.

**Theorem 9.** *The VRF construction in [Koh19] is not weakly unbiasable.*

*Proof.* We proceed by providing an explicit attack $\mathcal{A}$ against the weak unbiasability. Initially $\mathcal{A}$ samples a secret vector $\mathbf{t} \leftarrow^\$ \mathbb{Z}_p^d$, defining a 1-dimensional space $T = \mathbb{Z}_p\mathbf{t} \subseteq \mathbb{Z}_p^d$. It then creates matrices $M_{i,j,k} \in \mathbb{Z}_p^{d,d}$, and vectors $\mathbf{u}, \mathbf{w} \in \mathbb{Z}_p^d$ as follows:

1. For $j < n$, then $M_{i,j,k} \leftarrow^\$ \mathbb{Z}_p^{d,d}$ such that $\mathrm{Im}\,(M_{i,j,k}) = T$;
2. For $j = n$, them $M_{i,j,k} \leftarrow^\$ \mathbb{Z}_p^{d,d}$ such that $M_{i,j,k} \cdot \mathbf{t} \in T$;
3. $\mathbf{u} \leftarrow^\$ \mathbb{Z}_p^d \setminus T$;
4. $\widehat{\mathbf{w}} \leftarrow^\$ (\mathbb{Z}_p \setminus \{0\})^d$ such that $\mathbf{t}^\top \widehat{\mathbf{w}} = 0$;

5. $\mathbf{w} = (1/\widehat{w}_1, \ldots, 1/\widehat{w}_d) \in \mathbb{Z}_p^d$

Note all these distributions are efficiently sampleable. In particular, the first is obtained by setting $M_{i,j,k} = \mathbf{t} \otimes \mathbf{a}$ for a randomly sampled $\mathbf{a} \leftarrow^{\$} \mathbb{Z}_p^d$. For the second one it suffices to fix an orthogonal base $(\mathbf{b}_1, \ldots, \mathbf{b}_{d-1}) = B \in \mathbb{Z}_p^{d,d-1}$ of $T^\perp$ (the space orthogonal to $T$), sample a random matrix $M' \leftarrow^{\$} \mathbb{Z}_p^{d-1,d-1}$ defining a map from $T^\perp$ to $T^\perp$ and a random eigen-value $\eta$ so that $\mathbf{t}$ is mapped to $\eta\mathbf{t}$. $M_{i,n,k}$ is then set to $BM'B^\top + \eta \cdot \mathbf{t} \otimes \mathbf{t}$.

Finally, given this elements, let $\mathsf{AHF} : \{0,1\}^L \to \Sigma^n$. In order to guess the final output $\mathcal{A}$ samples $\sigma \leftarrow^{\$} \Sigma$ and returns

$$\mathsf{vk} \;=\; \left([M_{i,j,k}], [\mathbf{u}], [\widehat{\mathbf{w}}]\right), \qquad Y^* \;=\; \left[\widehat{\mathbf{w}}^\top \left(\prod\nolimits_{i=1}^{\ell} M_{i,n,\sigma}\right)\mathbf{u}\right].$$

Finally, on input $x$ with encoding $\mathsf{AHF}(x) = (x_1, \ldots, x_n) \in \Sigma^n$, it computes $Y$ and its proof as in [Koh19]:

$$Y \;=\; \left[\widehat{\mathbf{w}}^\top \left(\prod\nolimits_{i=1}^{\ell} \sum\nolimits_{j=1}^{n} M_{i,j,x_j}\right)\mathbf{u}\right] \quad [\mathbf{v}_\iota] \;=\; \left[\left(\prod\nolimits_{i=1}^{\ell} \sum\nolimits_{j=1}^{n} M_{i,j,x_j}\right)\mathbf{u}\right]$$
$$[\mathbf{z}] \;=\; \left[\widehat{\mathbf{w}} * \left(\prod\nolimits_{i=1}^{\ell} \sum\nolimits_{j=1}^{n} M_{i,j,x_j}\right)\mathbf{u}\right] \qquad \pi = ([\mathbf{z}], [\mathbf{v}_1], \ldots, [\mathbf{v}_n])$$

where we denote with $* : \mathbb{Z}_p^d \times \mathbb{Z}_p^d \to \mathbb{Z}_p^d$ the entry-wise multiplication.

To conclude we study the advantage of $\mathcal{A}$. First of all, by construction, all the pairing equation tested by $\mathsf{VRF.Vfy}$ on the output $y$ and its proof are satisfied, hence $\mathsf{VRF.Vfy}$ accepts with probability 1. Moreover, we have that, calling $Y = [y]$ and $F$ the set of functions from $\{1, \ldots, \ell\}$ to $\{1, \ldots, n\}$, and $f^*$ the function mapping all values to $n$:

$$\begin{aligned} y \;&=\; \widehat{\mathbf{w}}^\top \left(\prod\nolimits_{i=1}^{\ell} \sum\nolimits_{j=1}^{n} M_{i,j,x_j}\right)\mathbf{u} \;=\; \widehat{\mathbf{w}}^\top \left(\sum\nolimits_{f\in F} \prod\nolimits_{i=1}^{\ell} M_{i,f(i),x_{f(i)}}\right)\mathbf{u} \\ &=\; \widehat{\mathbf{w}}^\top \left(\prod\nolimits_{i=1}^{\ell} M_{i,n,x_n}\right)\mathbf{u} + \widehat{\mathbf{w}}^\top \left(\sum\nolimits_{f\in F\setminus\{f^*\}} \prod\nolimits_{i=1}^{\ell} M_{i,f(i),x_{f(i)}}\right)\mathbf{u} \\ &=\; \widehat{\mathbf{w}}^\top \left(\prod\nolimits_{i=1}^{\ell} M_{i,n,x_n}\right)\mathbf{u}. \end{aligned}$$

where the last equality follows since each term of the summation contains as a factor a matrix $M_{i,j,k}$ with $j < k$ meaning that its image is contained in $T$. Since all matrices preserves $T$, then their products maps $\mathbf{u}$ to a multiple of $\mathbf{t}$, which by construction is orthogonal to $\widehat{\mathbf{w}}$. In particular $Y = Y^*$ if and only if $x_n = \sigma$. Since $\sigma$ is sampled uniformly and independently from secret/public keys and input point, we have that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{w\text{-}bias}}(\lambda) - \frac{1}{p} \;=\; \Pr[Y = Y^*] - \frac{1}{p} \;=\; \Pr[x_n = \sigma] - \frac{1}{p} \;=\; \frac{1}{|\Sigma|} - \frac{1}{p}$$

which is not negligible, as $|\Sigma|$ is polynomially bounded. □

### B.2 Specialized Attack for Rosie's VRF

Finally, we provide a specialized attack for the construction in [Ros18]. We briefly recall it here. Given a symmetric bilinear group $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and an admissible hash $\mathsf{AHF} : \{0,1\}^L \to \{0,1\}^n$, the secret key consists of matrices $M_i, P_{i,0}, P_{i,1} \in \mathbb{Z}_p^{d,d}$ for $i < n$ and $M_{n,0}, M_{n,1} \in \mathbb{Z}_p^{d,d}$, as well as two vectors $\mathbf{w}, \mathbf{u}$ so that:

1. $M_1, \ldots, M_{n-1}, M_{n,0}, M_{n,1}$ are sampled uniformly from $\mathbb{Z}_p^{d,d}$;
2. $P_{i,0} = (\mathbf{t}_i, \mathbf{0}, \ldots, \mathbf{0}) \in \mathbb{Z}_p^{d,d}$ for a uniformly sampled $\mathbf{t}_i \leftarrow^{\$} \mathbb{Z}_p^d$;
3. $P_{i,1} = (\mathbf{0}, T_i, \mathbf{e}_d) \in \mathbb{Z}_p^{d,d}$ for a uniformly sampled $T_i \leftarrow^{\$} \mathbb{Z}_p^{d,d-2}$;
4. $\mathbf{u}, \mathbf{w} \leftarrow^{\$} \mathbb{Z}_p^d$.

As done in Section B.1, we will denote $\widehat{\mathbf{w}} = (1/w_1, \ldots, 1/w_n)$. Moreover to simplify notation we will call $\mathbf{M} = (M_1, \ldots, M_{n-1}, M_{n,0}, M_{n,1})$, $\mathbf{P}_0 = (P_{1,0}, \ldots, P_{n,0})$ and $\mathbf{P}_1 = (P_{1,1}, \ldots, P_{n,1})$. The verification key consists then of $\mathsf{vk} = ([\mathbf{M}], [\mathbf{P}_0], [\mathbf{P}_1], [\mathbf{u}], [\mathbf{w}])$.

On input $x$ with hash $\mathsf{AHF}(x) = (x_1, \ldots, x_n)$, the VRF output is defined as[19]

$$y = \left[ \widehat{\mathbf{w}}^\top M_{n,x_n} \left( \prod_{i=1}^{n-1} (M_i - P_{i,x_i}) \right) \mathbf{u} \right].$$

The proof instead consist of $([\mathbf{v}_1], \ldots, [\mathbf{v}_n], [\mathbf{z}])$ where $\mathbf{z} = \widehat{\mathbf{w}} * \mathbf{v}_n$ and

$$\mathbf{v}_n \;=\; M_{n,x_n} \left( \prod_{i=1}^{n-1} (M_i - P_{i,x_i}) \right) \mathbf{u}, \qquad \mathbf{v}_\iota \;=\; \left( \prod_{i=1}^{\iota} (M_i - P_{i,x_i}) \right) \mathbf{u}$$

for $\iota \in \{1, \ldots, n-1\}$. Finally, given input, output and proof, the verification procedure tests membership in $\mathbb{G}$ for all elements expected to be group elements, and accepts if the following pairing equations are satisfied:

1. $[\mathbf{v}_n]_T \;=\; e\left([M_{n,x_n}], [\mathbf{v}_{n-1}]\right)$;
2. $[\mathbf{v}_i]_T \;=\; e\left([M_i - P_{i,x_i}], [\mathbf{v}_{i-1}]\right)$ for all $i \in \{1, \ldots, n-1\}$ and $\mathbf{v}_0 = \mathbf{u}$
3. $[v_{\ell,i}]_T = e\left([w_i], [z_i]\right)$ for all $i \in \{1, \ldots, d\}$
4. $y = [z_1] + \ldots + [z_n]$

**Theorem 10.** *The VRF in [Ros18] is not weakly unbiasable.*

*Proof.* We provide an explicit adversary against weak unbiasability. Although the VRF structure is reminiscent of [Koh19], using the same strategy appears harder due to extra structure of $P_0, P_1$. For this reason we propose a variant of the above drawing ideas from the power of $\alpha$ attack. Crucially we will exploit the fact that the admissible hash alphabet is $\{0,1\}$.

$\mathcal{A}$ initially samples $\mathbf{u} \leftarrow^{\$} \mathbb{Z}_p^d$ and two matrices $S, T \leftarrow^{\$} \mathbb{Z}_p^{d,d}$ such that $\mathbf{t}_d = \mathbf{e}_d$ (where $\mathbf{t}_d$ is the $d$-th column of $T$) and $\mathbf{u}$ is an eigenvector of both. We can sample

[19] Assuming the product to be from right to left, i.e. $\prod_{i=1}^{n} M_i = M_n \cdot \ldots \cdot M_1$.

such a matrix $T$ by sampling $\mathbf{a}, \widehat{\mathbf{u}} \leftarrow^{\$} \mathbb{Z}_p^d$ such that $\widehat{\mathbf{u}}^\top \mathbf{u} = \widehat{\mathbf{u}}^\top \mathbf{e}_d = 0$ and then setting $T = I + \mathbf{a} \otimes \widehat{\mathbf{u}}$. The matrix $S$ can be sampled similarly. Given the above, define $T = P_0 - P_1$ (this uniquely identifies $P_0$ whose only non-zero column is the first, and $P_1$ whose first and last columns are respectively $\mathbf{0}$ and $\mathbf{e}_d$), and $M = S + P_0$. Then both $M - P_0 = S$ and $M - P_1 = S + T$ have $\mathbf{u}$ as an eigenvector. There exists then $\eta_0, \eta_1$ eigenvalues such that $(M - P_b)\mathbf{u} = \eta_b \cdot \mathbf{u}$.

Next, $\mathcal{A}$ proceeds with the key generation sampling $\alpha_i \leftarrow^{\$} \mathbb{Z}_p$ and setting $M_i = \alpha_i M$, $P_{i,0} = \alpha_i P_0$, $P_{i,1} = \alpha_i P_1$, while $M_{n,0}, M_{n,1}$ are sampled uniformly in $\mathbb{Z}_p^{d,d}$ and $\mathbf{w} \leftarrow^{\$} \mathbb{Z}_p^d$. Calling $\mathbf{M} = (M_1, \ldots, M_{n-1}, M_{n,0}, M_{n,1})$, $\mathbf{P}_0 = (P_{1,0}, \ldots, P_{n,0})$ and $\mathbf{P}_1 = (P_{1,1}, \ldots, P_{n,1})$, it finally computes the verification key

$$\mathsf{vk} \;=\; ([\mathbf{M}], [\mathbf{P}_0], [\mathbf{P}_1], [\mathbf{u}], [\mathbf{w}]).$$

Regarding its output guess, it samples $\beta \leftarrow^{\$} \{0,1\}$, $\mu_0 \leftarrow^{\$} \{0, \ldots, n-1\}$, computes $\alpha = \alpha_1 \cdot \ldots \cdot \alpha_{n-1}$, $\mu_1 = n - 1 - \mu_0$ and sets

$$y^* \;=\; \left[(\alpha \cdot \eta_0^{\mu_0} \cdot \eta_1^{\mu_1}) \cdot \widehat{\mathbf{w}}^\top M_{n,\beta} \mathbf{u}\right].$$

Finally, on input $x$ of hash $\mathsf{AHF}(x) = (x_1, \ldots, x_n) \in \{0,1\}^n$, the function computes output and proof $(y, \pi)$ as in the original scheme. In particular, the proof is accepting as by construction all the pairing equations are satisfied.

Finally we bound the advantage of $\mathcal{A}$. By construction the discrete logarithm of $y$ is (calling $s_0, s_1$ the number of 0 and 1 in $(x_1, \ldots, x_n)$)

$$\begin{aligned}
\widehat{\mathbf{w}}^\top M_{i,x_n} \left(\prod_{i=1}^{n-1} (M_i - P_{i,x_i})\right) \mathbf{u} \;&=\; \alpha \cdot \widehat{\mathbf{w}}^\top M_{i,x_n} \left(\prod_{i=1}^{n-1} (M - P_{x_i})\right) \mathbf{u} \\
&=\; \alpha \cdot \widehat{\mathbf{w}}^\top M_{i,x_i} \left(\prod_{i=1}^{n-1} \eta_{x_i}\right) \mathbf{u} \\
&=\; (\alpha \cdot \eta_0^{s_0} \cdot \eta_1^{s_1}) \cdot \widehat{\mathbf{w}}^\top M_{i,x_i} \mathbf{u}.
\end{aligned}$$

In particular $y = y^*$ if and only if $\beta = x_n$ and $\mu_0 = s_0$ (note $\mu_0 = s_0 \Rightarrow \mu_1 = s_1$). Hence

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{w\text{-}bias}}(\lambda) \;=\; \Pr[y = y^*] \;=\; \Pr[\beta = x_n,\ \mu_0 = s_0] \;=\; \frac{1}{2(n-1)}.$$

Thus, $\mathcal{A}$ breaks weak unbiasability as its advantage is significantly greater than $1/|\mathbb{Z}_p| = 1/p$. □

### B.3 Invertible VRF

[Bra25] proposed an attack against the *so-called* invertible VRFs. This attack can be applied against the construction of [DP07]. Since this is a useful result to make our taxonomy complete, we recall it here.

**Definition 12 (Invertible VRF [Bra25]).** *Let* $\mathsf{VRF} = (\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Vfy})$ *be a VRF. We say that* VRF *is invertible if there exists an algorithm* $\mathsf{VRF.Inv}$ *such that for each key pair* $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{VRF.Gen}(1^\lambda)$ *and each preimage* $x \in \mathcal{X}$ *it holds that* $\mathsf{VRF.Inv}(\mathsf{sk}, y) = x$ *where* $(y, \pi) := \mathsf{VRF.Eval}(\mathsf{sk}, x)$.

*Remark 2.* Note that the notion of (perfect) invertability requires the VRF to be injective.

**Theorem 11.** *[Bra25] Let* $\mathsf{VRF} = (\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Vfy})$ *be an invertible VRF, then* $\mathsf{VRF}$ *cannot be unbiasable.*