The Price of Verifiability: Lower Bounds for Verifiable Random Functions

Nicholas Brandt, Dennis Hofheinz, Julia Kastner, and Akin Ünal*

Department of Computer Science
ETH Zurich
Zurich, Switzerland
{nicholas.brandt,hofheinz,julia.kastner,akin.uenal}@inf.ethz.ch

Abstract. Verifiable random functions (VRFs) are a useful extension of pseudorandom functions for which it is possible to generate a *proof* that a certain image is indeed the correct function value (relative to a public verification key). Due to their strong soundness requirements on such proofs, VRFs are notoriously hard to construct, and existing constructions suffer either from complex proofs (for function images), or rely on complex and non-standard assumptions.
In this work, we attempt to explain this phenomenon. We first propose a framework that captures a large class of pairing-based VRFs. We proceed to show that in our framework, it is not possible to obtain short proofs *and* a reduction to a simple assumption simultaneously. Since the class of "consecutively verifiable" VRFs we consider contains in particular the VRF of Lysyanskaya and that of Dodis-Yampolskiy, our results explain the large proof size, resp. the complex assumption of these VRFs.

1 Introduction

Verifiable Random Functions. Pseudorandomness, and in particular pseudorandom generators [6, 47] and pseudorandom functions (PRFs, [20]) have proven to be immensely useful and universal cryptographic building blocks. A PRF takes as input a short seed (or key) sk, and an input x, and outputs a function value $\mathbf{y} = \mathsf{prf}_{\mathsf{sk}}(x)$. The distinguishing feature of a PRF is that for a fixed but random sk, oracle access to $\mathsf{prf}_{\mathsf{sk}}(\cdot)$ cannot be distinguished from oracle access to a truly random function. This allows to use prf as a compact drop-in replacement for a truly random function.

In this work, we focus on a special class of PRFs whose image can be *proven* to be correct (relative to a public key vk that fixes prf's behavior). Indeed, a verifiable random function (VRF [36]) vrf is a PRF for which it is possible to generate proofs π (from a given sk and x) that show that a given $\mathbf{y}$ really satisfies $\mathbf{y} = \mathsf{vrf}_{\mathsf{sk}}(x)$. We want such proofs to be sound in a very strong sense: We require that for *any* vk and x, no two $\mathbf{y} \neq \mathbf{y}'$ can both be proven to be $\mathsf{vrf}_{\mathsf{sk}}(x)$.

* Work done while all authors were supported by ERC Project PREP-CRYPTO 724307

This property, dubbed "unique provability", is crucial for most applications of VRFs, and is the main reason why constructing VRFs is difficult. For instance, unique provability cannot be achieved by using non-interactive zero-knowledge proofs on a given PRF. (This would require a trusted common reference string, which we cannot assume in the VRF setting.) We do note, however, that (non-straightforward) solutions with non-interactive witness-indistinguishable (NIWI) proofs are possible [5, 23].

VRFs have a number of interesting applications. These include signatures with very strong verifiability guarantees [22], resettable zero-knowledge proofs [37], lottery systems [38], transaction escrow schemes [27], updatable zero-knowledge databases [33], and e-cash systems [2, 4].

Existing Constructions of VRFs. There are a variety of constructions of VRFs already [1, 5, 8, 15, 16, 23–26, 30–32, 34, 36, 41, 43, 46]. These constructions are diverse in the used techniques and the resulting features: For instance, some constructions (such as Lysyanskaya's VRF [34] and its variants [8, 24–26, 43]) are based on the specific algebraic properties of the Naor-Reingold PRF [39], while others (such as [5, 23]) are based on more generic primitives such as NIWI proofs. However, none of the above VRF constructions achieves all of the following useful features simultaneously:

- its input space is large (i.e., exponential in the security parameter),
- its proofs π are short (i.e., comprise a constant number of group elements),
- its security proof is based on a "simple" (i.e., non-interactive and compact[1]) assumption.

We do note that some of the constructions come close: E.g., Kohl's VRF [30] achieves all of the above properties, except that proofs π comprise $\omega(1)$ group elements. Conversely, the VRF of Dodis and Yampolskiy [16] enjoys very compact proofs, but relies on a complex hardness assumption (with challenges as large as the input space). While there exists work on the difficulty of achieving VRFs (e.g., from trapdoor one-way functions [17], cf. [11], or in a tightly secure way [41]), the proof size and necessary assumptions for VRFs are generally not well-understood.

Our Contribution. In this work, we are concerned with the reason *why* it is difficult, even after a plethora of different approaches and 20 years of research, to construct useful and compact VRFs from standard assumptions. In order to give a meaningful answer, we put forward a framework of VRF restrictions that however covers many existing constructions. We proceed to show lower bounds within this framework.

Specifically, we restrict ourselves to VRFs vrf in the standard model (i.e., that do not use random oracles or generic groups) that are algebraic over a

[1] With a non-interactive and compact assumption, we mean one in which the adversary gets a constant number of group elements as challenge and is then supposed to output a solution (e.g., a decision bit).

group, such that secret keys sk are comprised of exponents, and public keys vk, images $\mathbf{y}$, and proofs π are all comprised of group elements. We do allow pairings, however, such that in particular images may be elements of a target group.

Furthermore, we require that verification (of a proof π for an image $\mathbf{y}$) operates in a specific and "consecutive" way. We give more details on the conditions on verification below in the technical overview. We stress, however, that we believe that this way to verify is natural, and in fact many existing VRFs support consecutive verification, including Lysyanskaya's VRF [34], the VRF of Dodis and Yampolskiy [16], and many more (see Fig. 1). A convenient consequence of this type of consecutive verification is that the function image $\mathbf{y}$ has a specific form: We can deduce that $\mathbf{y} = \mathsf{vrf}_{\mathsf{sk}}(x)$ is of the form $\mathbf{g}^{\sigma_x(\overrightarrow{v})/\rho_x(\overrightarrow{v})}$, where

- $\mathbf{g}$ is a fixed group generator,
- σ_x and ρ_x are multivariate polynomials (that depend in any efficiently computable way on the preimage), and
- $\overrightarrow{v}$ is the vector of discrete logarithms of the verification key vk.

We finally assume a large (i.e., superpolynomial in the security parameter) input space. Again, while this of course severely restricts the VRFs we consider, many previous constructions fall into this class.[2]

For such algebraic VRFs with consecutive verification, we show necessary relations between the size of proofs π and the "size" of the underlying assumption (i.e., the size of the challenge in group elements in a non-interactive hardness assumption). To develop and express these relations, it is useful to consider what we call the *evaluation degree* of the VRF. Formally, this degree is simply the maximum of the degrees of the (multivariate) polynomials σ_x and ρ_x from the image $\mathbf{y} = \mathbf{g}^{\sigma_x(\overrightarrow{v})/\rho_x(\overrightarrow{v})}$ above (and for this exposition, we assume that these degrees do not depend on x).

We show that for any VRF vrf that matches all of our formal requirements,

(a) if the size of π (in group elements) is small, then so is the degree of vrf,
(b) if vrf's degree is small, then vrf cannot be proven secure with a generic reduction to a constant-size non-interactive hardness assumption. (We note that almost all existing cryptographic reductions are generic.)

As an example, our results show that the VRF of Dodis and Yampolskiy cannot be proven secure (at least not generically) from more traditional hardness assumptions. Our results also show that the (comparatively large) proofs in Lysyanskaya's VRF are inherent, at least when relying on standard hardness assumptions. Figure 1 lists more VRFs that fulfill our requirements (and whose proof sizes and/or assumptions can hence be justified with our results).

[2] A prominent verifiable *unpredictable* function (VUF, a weaker form of VRF) that does not fall into this class is the one by Brakerski *et al.* [11]. This VUF takes group elements *as input*, and hence does not quite fit our framework. We will discuss this particular construction in Section 2.1, and argue that this approach is unlikely to yield purely group-based *VRFs*.

While our result (a) is a direct consequence of our requirement on consecutive verifiability, we in fact give two versions of statement (b) that differ in exact requirements and formalization. For instance, one version of (b) even excludes *algebraic* reductions (i.e., is formalized within the algebraic group model [19]) from non-interactive assumptions of any polynomial size, but only applies to VRFs whose verification keys depend on a single variable or from non-interactive computational assumptions that depend on a single variable. This allows to model Dodis and Yampolskiy's VRF, but not Lysyanskaya's. The other version of (b) allows more general verification keys, but only excludes *generic* reductions (i.e., is formalized within the generic group model [35, 40, 45]). In the next section, we give a more technical overview over our results.

Discussion. While the formal requirements for our lower bounds seem restrictive, their preconditions are met by most existing VRFs (see Fig. 1). In that sense, they justify the limitations of existing constructions, resp. proofs. An obvious question is thus: How can one circumvent our lower bounds (in order to construct VRFs with short proofs from standard assumptions)?

First of all, one could of course circumvent our results by not (or at least not completely) working over cyclic groups. However, while there are a few more generic VRF constructions (e.g., [5, 23]) that do not rely on groups, it seems that generic VRF constructions are less well-investigated than constructions based on cyclic groups.

Second, one could try to circumvent the more specific requirements of our lower bounds. In particular, our "consecutive verifiability" requirement seems like a very specific requirement. An "interesting" (as opposed to a purely mechanical) way to circumvent consecutive verifiability would be the following. Recall that consecutive verifiability implies that VRF images consist of rational functions, i.e., are of the form $\mathbf{y} = \mathbf{g}^{\sigma_x(\overrightarrow{v})/\rho_x(\overrightarrow{v})}$. Jumping ahead, we will be interested in small-degree polynomials σ_x, ρ_x. The following VRF candidate does not have this property:

$$\mathsf{vk} = e(\mathbf{g},\mathbf{g})^s, \qquad \mathbf{y} = \mathbf{g}^{\sqrt[3]{s+x}} \qquad \pi = \mathbf{g}^{(\sqrt[3]{s+x})^2}.$$

Verification checks that $e(\mathbf{y},\mathbf{y}) = e(\mathbf{g},\pi)$ and $e(\pi,\mathbf{y}) = \mathsf{vk}\cdot e(\mathbf{g},\mathbf{g})^x$. The security of this VRF candidate seems unclear, but observe that we require $3 \not| (\mathrm{ord}(\mathbf{g})-1)$ both for uniqueness, and to be able to compute $\sqrt[3]{s+x} \bmod \mathrm{ord}(\mathbf{g})$.

More generally, our results do not exclude VRFs in which the image is an active ingredient in intermediate verification computations, and not only considered in a final verification step (that involves previously computed and/or verified proof elements). Of course, for constructions that use, e.g., roots of exponents (like the above candidate), it may be challenging to prove their security from Diffie-Hellman-like assumptions.

1.1 High-level Technical Overview

The Evaluation Degree of a VRF. Our technical results rely on the "evaluation degree" of a VRF vrf as a helpful technical notion that connects vrf's proof sizes

Reference	CV	degree	$\lvert\mathsf{vk}\rvert$	$\lvert\pi\rvert$	assumption	remark
MRV99 [36]	×	—	large	large	RSA	tree-based
Lys02 [34]	✓	λ	2λ	λ	q-type	
Dod03 [15]	✓	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	ad-hoc	
DY05 [16]	✓	1	2	1	q-type	small inputs
ACF09 [1]	✓	$\lambda+2$	$2\lambda+2$	$\lambda+1$	q-type	
BCKL09 [4]	×	1	3	$O(1)$	q-type	small inputs
BGRV09 [11]	×	—	1	1	gap-CDH	weak security
BMR10 [8]	✓	$\lambda+1$	$(\lambda+2)$	λ	q-type	small inputs
HW10 [25]	✓	$\lambda+1$	$\lambda+3$	$\lambda+1$	q-type	
Jag15 [26]	✓	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	q-type	
LLC15 [32]	✓	$\lambda+1$	$2\lambda+1$	1	q-type	multilinear maps
HJ16 [24]	✓	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	DLIN	
Bit17 [5]	×	—	depends	large	depends	generic/NIWI-based
GHKW17 [23]	×	—	depends	large	depends	generic/NIWI-based
Kat17 [28]	✓	$\omega(\log(\lambda)^2)$	$\omega(\sqrt{\lambda}\log(\lambda))$	$\omega(\sqrt{\lambda})$	q-type	
Yam17 [46]	✓	$O(\log(\lambda)^2)$	$O(\lambda\log(\lambda)^2)$	$O(\log(\lambda)^2)$	q-type	
Ros18 [43]	✓	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	DLIN	smaller π than [24]
Koh19 [30]	✓	κ	$\mathrm{poly}(\lambda)$	κ	DLIN	$\kappa \in \omega(1)$ parameter
Nie21 [41]	✓	$O(\lambda)$	$\omega(\log(\lambda))$	$O(\lambda)$	q-type	

Fig. 1. Existing VRF constructions. The "CV" column indicates whether the construction is consecutively verifiable in our sense. "Degree" denotes its evaluation degree (where applicable), and $\lvert\mathsf{vk}\rvert$ and $\lvert\pi\rvert$ denote its verification key size, resp. proof size in group elements. When possible, we have chosen parameters such that the input size is $\{0,1\}^\lambda$. For comparability, we classify assumptions with polynomially many challenge elements as "q-type", and other nonstandard assumptions as "ad-hoc". "Small inputs" (as a remark) means that the VRF only supports polynomially-small input spaces. Theorem 1 applies to [11, 16], Theorem 2 applies to [16], Theorem 3 applies to [8, 11, 16] in the sense that these VUF/VRFs cannot have constant size proofs based on standard assumptions.

and vrf's underlying hardness assumption. Hence, let us first take a closer look at this notion of degree.

First, we recall one of our restrictions on the VRFs we consider. We assume that vk and π consist of group elements, and that verification operates in a "consecutive" way, in the following sense: Assume that verification wants to verify a proof π (which consists of, say, κ group elements $\pi_1, \ldots, \pi_\kappa$) for an alleged image $\pi_{\kappa+1} := \mathbf{y}$ (which is a single group element). Then, we require that verification proceeds in $\kappa+1$ steps, and in the i-th step checks an a priori fixed system of pairing product equations in variables $\pi_1, \ldots, \pi_i$ and vk. We also require that in the equations for the check for π_i, this element only occurs linearly (but not quadratically, i.e., in both arguments of a pairing).

Verification succeeds if all these systems of equations hold. In other words, proof elements (and eventually image $\mathbf{y}$) are verified one at a time, each time checking a quadratic equation in the corresponding exponents of this and all previous elements and vk.

This notion of consecutive verification sounds natural in a pairing setting, and indeed many existing vrf constructions (including the ones from [16, 34]) have a consecutive verification procedure in the above sense. Intuitively, consecutive verification requires that "higher-degree" exponents in proof elements or image must be verified using intermediate group elements with intermediate degrees. Fortunately, as already outlined, consecutive verification also implies that images $\mathbf{y}$ are of the form

$$\mathsf{vrf}_{\mathsf{sk}}(x) = \mathbf{y} = \mathbf{g}^{\sigma_x(\overrightarrow{v})/\rho_x(\overrightarrow{v})}$$

for multivariate polynomials σ_x and ρ_x (which both are efficiently computable from x), and the component-wise discrete logarithm $\overrightarrow{v}$ of vk. Now we say that the *evaluation degree* of vrf (or $\mathbf{y}$) is simply the maximum of the polynomial degrees of σ_x and ρ_x. The evaluation degree of the VRF is then simply the maximal degree over all inputs x.

First Result: Proof Size Bounds Degree for VRFs with Consecutive Verification. Our first result ((a) above, described in more detail in Section 2.1, and in full detail in Section 4) shows that for VRFs vrf with consecutive verification (as above), the size of proofs π imposes a limit on the vrf's evaluation degree. Concretely, we show that the evaluation degree of vrf is at most exponential in the proof size κ. Hence, if its proof size is constant, then so is the evaluation degree of vrf.

This result is not too surprising, since intuitively, each additional proof element only raises the degree of computed exponents (as algebraic fractions in $\overrightarrow{v}$) by a factor of 2. In fact, our proof largely consists in keeping track of expressions of all intermediate proof elements (and finally of $\mathbf{y}$) as expressions in $\overrightarrow{v}$. The main technical work consists in maintaining a suitable canonical form of these (rational) expressions at all times.

Interlude: the Case of Trivial Denominators. If function images are of the form $\mathbf{y} = \mathbf{g}^{\sigma_x(\overrightarrow{v})}$ for a constant-degree (but multivariate) polynomial σ_x, already a very simple linear algebra attack breaks the pseudorandomness of the given VRF. In fact, for sufficiently many preimages x_i, the polynomials σ_{x_i} must eventually become linearly dependent (because the set of their monomials is polynomially small). Hence, it is possible to linearly combine sufficiently many given images to form the image of a fresh preimage. This breaks pseudorandomness, and we detail this attack in the full version[12] for completeness. The case of rational function images $\mathbf{y} = \mathbf{g}^{\sigma_x(\overrightarrow{v})/\rho_x(\overrightarrow{v})}$ (with $\deg(\rho_x) \geq 1$) is hence not only more general (and covers, e.g., the Dodis-Yampolskiy VRF), but also technically much more interesting.

Second Result: Security of Polynomial-Degree VRFs Requires Complex Assumptions (for Univariate Verification Keys and in the Algebraic Group Model). Our second result (first variant of (b) above, described in Section 2.2 more extensively, and in Section 5 in full detail) shows that for any polynomial-degree VRF vrf, we can rule out the existence of an "algebraic black-box" reduction to a class

of non-interactive group-based computational assumptions. Here, an "algebraic black-box" reduction $\mathcal{B}$ fulfills the following requirements:

- It is algebraic (in the sense of [19]): That means that whenever $\mathcal{B}$ outputs a group element $\mathbf{g}^*$, it also outputs (on a special channel) an explanation as to how $\mathbf{g}^*$ is computed from previously seen group elements.
- It uses the VRF adversary $\mathcal{A}$ only in a black-box way (i.e., it gets oracle access to polynomially many instances of $\mathcal{A}$).

Most existing reductions (in particular for VRFs) are simple in the above sense.

A non-interactive (group-based) computational assumption (NICA) states that it is hard for any efficient adversary $\mathcal{B}$ to win the following game: $\mathcal{B}$ gets a challenge (that is a vector of s group elements), and is then supposed to output a solution to that challenge (which is an exponent related to s). The size of such a NICA is simply the length (i.e., number of entries) of s.

We are now ready to state our result a bit more formally: Assume we are given a polynomial-degree VRF vrf with verification key $\mathsf{vk} = \mathbf{g}^v$. Furthermore, assume that vrf enjoys a simple reduction $\mathcal{B}$ to a NICA. Then, we construct a meta-reduction [13] that wins the NICA game without any external help. Our meta-reduction $\mathcal{M}$ interacts with $\mathcal{B}$ (which gets a NICA challenge), and then attempts to take the role of a successful VRF adversary $\mathcal{A}$. In order to do this, $\mathcal{M}$ can query many VRF images $\mathbf{y}_i$, and use the algebraicity of $\mathcal{B}$ to obtain representations of these $\mathbf{y}_i$ in terms of the NICA challenge elements. Hence, eventually $\mathcal{B}$ will find linear dependencies between the queried VRF images by making sufficiently (but still polynomially) many queries. These linear dependencies can then be used to compute the verification key's exponent v. Using v, the meta-reduction can predict *any* challenge image as $\mathbf{g}^{\sigma_x(v)/\rho_x(v)}$. This allows $\mathcal{A}$ to win the VRF security game, and hence $\mathcal{M}$ can use $\mathcal{B}$ to solve the NICA.

This intuition neglects a number of technical obstacles: For instance, the linear dependencies among the algebraic representations of VRF images linearly connect the algebraic fractions $\sigma_{x_i}(v)/\rho_{x_i}(v)$ of the corresponding images. To construct a new image $\mathbf{g}^{\sigma_{x^*}(v)/\rho_{x^*}(v)}$ from these, we need to distinguish the cases when the polynomial fraction $\sigma_{x^*}(X)/\rho_{x^*}(X)$ of the challenge can be expressed as a linear combination of the polynomial fractions $\sigma_{x_i}(X)/\rho_{x_i}(X)$ of the queries, and when this is not the case. In the first case, the corresponding linear dependence immediately allows to compute $\mathbf{g}^{\sigma_{x^*}(v)/\rho_{x^*}(v)}$. Note that this is also possible for an adversary that does *not* get to see the algebraic representations because the linear dependence holds for the fractions, not only for the representations.

In the second case, we have to develop a linear dependence among the algebraic representations (in the NICA challenge elements) of the $\sigma_{x_i}(v)/\rho_{x_i}(v)$. In this case, in fact the linear *independence* of the fractions $\sigma_{x_i}(X)/\rho_{x_i}(X)$ guarantees that these linearly dependent algebraic representations allow to extract the secret v.

In these observations, we crucially use that we deal with univariate polynomials σ_{x_i} and ρ_{x_i} of small degree (which can be represented by short coefficient vectors). In a separate result, we generalize this approach to multivariate σ_{x_i}

and ρ_{x_i} where the underlying assumption only depends on a single variable with polynomial degree.

Third Result: Security of Low-Degree VRFs Requires Complex Assumptions (in the Generic Group Model). Our last result (second variant of (b) above, explained more extensively in Section 2.3, and in full detail in the full version[12] is similar in spirit to our second result, but features different requirements on the considered VRFs and reductions. Specifically, we prove that no generic reduction (i.e., that treats the underlying group as generic in the sense of [45]) that is algebraic black-box (as outlined above) is able to show security of a constant-degree VRF based on any "Uber-assumption" [7, 10] of arbitrary polynomial degree but constant challenge size.

An "Uber-assumption" is a special class of a NICA in which an adversary $\mathcal{B}$ is given a number of group elements $\mathbf{g}^{f_i(\overrightarrow{z})}$, where the f_i are multivariate polynomials specific to the concrete assumption, and $\overrightarrow{z}$ is a vector of secret (and uniformly randomly chosen) exponents. Typically, the task of $\mathcal{B}$ is then to compute a group element not in the linear span of the given group elements (or to distinguish such an element from random). Here, we restrict ourselves to Uber-assumptions in which the degree of the f_i is at most polynomial in the security parameter.

We again give a meta-reduction $\mathcal{M}$ that shows the following: Any simple generic reduction $\mathcal{B}$ that shows the security of a constant-degree VRF under such an Uber-assumption can be transformed into a successful Uber-solver. Again, $\mathcal{M}$ takes the role of a VRF adversary that interacts with $\mathcal{B}$. In the following, we outline our technique for the specific case of the Dodis-Yampolskiy VRF, in which $\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2) = (\mathbf{h}, \mathbf{h}^s)$, $\mathbf{y} = e(\mathbf{h}, \mathbf{h})^{1/(s+x)}$ (for a pairing e), and $\pi = \mathbf{h}^{1/(s+x)}$.

Our meta-reduction $\mathcal{M}$, when interacting with a reduction $\mathcal{B}$ in the role of a VRF adversary $\mathcal{A}$, first of all gets to see vk and an algebraic representation of vk in terms of the NICA challenge. (In this work, we call an algorithm generic iff it is generic in the sense of Shoup's GGM *and* algebraic, cf. Definition 5.) This representation of $\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$ allows $\mathcal{M}$ to write $\mathsf{vk}_i = \mathbf{g}^{g_i(\overrightarrow{z})}$ for polynomials g_i in the Uber-assumption secrets $\overrightarrow{z}$.

Now we distinguish two cases: First, if the polynomial g_2 is a scalar multiple of g_1, (i.e., if $g_2 = s' \cdot g_1$ for a scalar s'), then we have found the VRF secret key $s = s'$. This s can directly be used to break VRF security and allows $\mathcal{M}$ to imitate a successful adversary for $\mathcal{B}$ (which in turn breaks the underlying Uber-assumption). But in case g_2 is not a scalar multiple of g_1, such a simple extraction of s is not possible.

The main technical work in our proof consists in showing that this second case cannot, in fact, occur with non-negligible probability. Essentially, we do so by observing that the representations of VRF proofs $\pi_i = \mathbf{h}^{1/(s+x_i)}$ (i.e., of $(s+x_i)$-th roots of $\mathbf{h} = \mathsf{vk}_1$) imply polynomial factors of g_1. We prove that if g_2 is not a scalar multiple of g_1, then these factors are coprime for different x_i. Hence, querying sufficiently many VRF proofs (for different x_i) yields many non-trivial coprime factors of g_1. Since we assumed that the degree of g_1 is polynomial (since the Uber-assumption polynomials f_i are of polynomial degree), this eventually

yields a contradiction. Hence, g_2 must be a scalar multiple of g_1, and our meta-reduction $\mathcal{M}$ can proceed as described above.

In the full version[12], we also show how to generalize this argument to a broader class of constant-degree VRFs which we call *parameterized rational*.

Omitted Details. All the above explanations have omitted or simplified a few details. For instance, we did not discuss the role of *group parameters* (that fix the concrete group and pairing setting). For VRFs, such group parameters should be *certified* [24] (i.e., reliably defining an actual group), and they can be an additional part of vk or any public parameters. Since generic groups can be viewed as "implicitly trusted", we omit this certification in the generic group model.

Furthermore, we have treated the VRF image always as a target group element. However, since we are in a pairing setting, this image can also (and in fact without loss of generality) be an element of the *source group* of the pairing. (This does not change any of the arguments above.) Finally, we mostly consider verifiable *unpredictable* functions (VUFs), a relaxation of verifiable *random* functions. Since we present lower bounds, this only makes our results stronger.

2 Detailed Technical Overview

2.1 First Result: Connecting the Proof Size with the Evaluation Degree

Consecutively Verifiable VUFs/VRFs. To make the connection between the number of group elements in the proof and the evaluation degree, we first define a class of VUFs/VRFs that have a very straightforward verification algorithm. We assume that the VUFs/VRFs in question operate over a symmetric[3] pairing group with pairing $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$:

- The verification key vk consists of group elements $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{G} \cup \mathbb{G}_T$
- For each input x, the proof consist of group elements $\pi_1, \ldots, \pi_\kappa \in \mathbb{G} \cup \mathbb{G}_T$
- For each input x, the evaluation value is a group element $\mathbf{y} \in \mathbb{G}_T$

Each possible input element x of the VUF/VRF defines a set of pairing equations E_x that can be efficiently derived[4] from the input x. By pairing equations we mean a set of polynomial equations of degree 2 in the input variables. We make the additional restriction that variables that represent elements from the target group may appear only in monomials of degree 1. We require that the pairing equations can be verified *consecutively*, that is, there is an ordering of the group elements in the proof and subsets $E_{i,x}$ of the sets of pairing equations such that the following hold:

[3] We note that our results can easily be transferred to asymmetric pairings, but for simplicity we restrict ourselves to symmetric pairings.

[4] We note that the weak VRF by Brakerski *et al.* [11] does not have this efficiency property, as the inputs are group elements and the pairing equations can only be derived from the discrete logarithm of the inputs.

- in the pairing equation set $E_{i,x}$ for the i-th proof element, only the verification key elements and proof elements up to the i-th occur, i.e., $E_{i,x} \subset \mathbb{Z}_p[V_1, \ldots, V_n, P_1, \ldots P_i]$
- in the pairing equation set $E_{i,x}$ for the i-th proof element, there is at least one equation where the i-th proof element occurs only linearly, i.e., there exist polynomials $a_i \in \mathbb{Z}_p[V_1, \ldots, V_n, P_1, \ldots, P_{i-1}], b_i \in \mathbb{Z}_p[V_1, \ldots, V_n, P_1, \ldots P_i]$ such that $a_i \cdot P_i + b_i = 0$ is an equation that occurs in $E_{i,x}$.

We further make a more technical requirement that the coefficient a_i of the i-th proof element in the equation where it occurs linearly cannot become zero. Let the proof have κ many elements, then we consider the evaluation value to be the $\kappa + 1$st proof element, i.e., it is the last group element to be "verified" in this way.

This consecutive verification property on the one hand yields an efficient pairing-based verification algorithm (for input x, first efficiently derive the pairing equation sets $E_{i,x}$, then consecutively check them). On the other hand, the linearity requirement actually implies that given the verification key and the previous proof elements, each proof element is uniquely defined. As the evaluation value is the last element to be verified, i.e., the $\kappa + 1$st "proof element", it is therefore also uniquely provable.

We note that this consecutive verification property applies to many known VRFs, see Fig. 1 for a detailed overview.

We briefly sketch how the pairing equations look for the VRF of Dodis & Yampolskiy [16]: Recall that the evaluation key is $\mathsf{sk} = s \in \mathbb{Z}_p$ and the verification key is $\mathsf{vk} = \mathbf{h}^s$ for a publicly known group generator $\mathbf{h}$ of $\mathbb{G}$. Evaluation at value x computes $\mathbf{y} = e(\mathbf{h}, \mathbf{h})^{\frac{1}{s+x}}$ as well as the proof $\pi = \mathbf{h}^{\frac{1}{s+x}}$. We can consecutively verify this as follows: First verify the proof via $E_{1,x} = \{(V + x) \cdot P = 1\}$ where V represents the verification key, and P represents the group element. That is, the verification algorithm checks $e(\mathsf{vk} \cdot \mathbf{h}^x, \pi) = e(\mathbf{h}, \mathbf{h})$. Then, we verify $E_{2,x} = \{P \cdot 1 = Y\}$ where P is as before and Y represents the evaluation value, that is the verification algorithm checks $e(\pi, \mathbf{h}) = \mathbf{y}$.

Remark 1 (Consecutive Verifiability of the VUF of Brakerski et al. *[11]).* As we pointed out above, the weak VUF of Brakerski *et al.* [11], where evaluation works by $\mathsf{Eval}_{\mathsf{vuf}}(\mathsf{sk}, \mathbf{h}) = \mathbf{h}^{\mathsf{sk}}$ for $\mathsf{sk} \in \mathbb{Z}_p$ and $\mathsf{vk} = \mathbf{g}^{\mathsf{sk}}$ and an input $\mathbf{h} \in \mathbb{G}$, and verification accepts if $e(\mathbf{h}, \mathsf{vk}) = e(\mathbf{y}, \mathbf{g})$, is not consecutively verifiable in the sense of this work. In fact, we would need to know the discrete logarithm of the input $\mathbf{h}$ to efficiently compute a pairing equation for it. Therefore, the results of this paper are not applicable to this VUF.

However, while this might seem to limit the class of VUFs we consider in this work, we claim that weak VUFs that have group elements as inputs are – for the pursuit of strong VRFs – not relevant, anyway. In fact, images of the weak VUF of Brakerski *et al.* [11] can easily be predicted for adversially chosen inputs. This observation can be extended to other weak VRF/VUF candidates that operate in a similar algebraic manner, i.e., that take group elements as inputs and interpret them *as group elements* only and use the group operations

and pairing operations on them to compute the output. We show in the full version[12] that these VRFs/VUFs become insecure by as their evaluation degree is at most 2 in the inputs if the discrete logarithms of the input group elements are known to the adversary.

Rational VUFs/VRFs. We want to show that the formerly mentioned class of consecutively verifiable VUFs/VRFs has a particularly straightforward way to describe their evaluation algorithm. To this end, we define rational VUFs. These are VUFs whose evaluation value consists of a (publicly known) group generator raised to a rational function evaluated on the exponents of the verification key. More formally, for each input value x, there are polynomials ρ_x and σ_x such that the output $\mathbf{y}$ evaluated at x is

$$\mathbf{y} = \mathbf{g_T}^{\frac{\sigma_x(v_1,\ldots,v_n)}{\rho_x(v_1,\ldots,v_n)}}$$

where $v_1, \ldots, v_n$ are the exponents of the group elements in the verification key vk. We say that the total degree of the polynomials σ_x and ρ_x is the *evaluation degree* of the VUF/VRF.

From Consecutive Verifiability to Rationality with Bounded Degree. We show, using an inductive argument, that (a) consecutively verifiable pairing based VUFs/VRFs are also rational VUFs/VRFs, and (b) that the evaluation degree is at most exponential in the proof size – this implies that the proof size needs to be at least logarithmic in the evaluation degree for consecutively verifiable VUFs/VRFs. The proof uses induction to show that in fact all proof elements can be expressed through rational functions in the exponent, i.e., there exist σ_{x,π_i} and ρ_{x,π_i}, and that the degree of the i-th proof element is at most 4^i. The base case is easy to see: To obtain σ_{x,π_1} and ρ_{x,π_1} from the first set of pairing equations, we use the pairing equation that contains P_1 as a linear factor. This equation can be expressed as $a \cdot P_1 + b = 0$ where a, b are polynomials (a has degree at most 1 and b degree at most 2). We can therefore express $P_1 = b/-a$.

For the inductive step it is again crucial that the i-th proof element occurs only linearly in at least one pairing equation, as it can then be viewed as a zero of a linear equation and expressed as a rational function of the previous proof elements and the verification key. We replace the previous proof element P_{i-1} by its rational expression $\frac{\sigma_{x,\pi_{i-1}}}{\rho_{x,\pi_{i-1}}}$ in the pairing equation set $E_{i,x}$ to obtain $P_i \cdot a'_i + b'_i = 0$ where the a'_i and b'_i are rational functions in the verification key elements. We then derive the rational expression for $P_i = b'_i/-a'_i = \sigma_{x,\pi_i}/\rho_{x,\pi_i}$ where σ_{x,π_i} and ρ_{x,π_i} are polynomials. It remains to show that the resulting polynomials have the degrees required by our statement which can be done using some simple arguments.

Inductively replacing all proof elements by such rational expressions in the verification key elements yields the result for the last element to be verified – the evaluation value.

2.2 Second Result: Security of Univariate Polynomial-Degree VRFs Requires Complex Assumptions

In current pairing-based constructions of VRFs there seems to be a tradeoff between the size/complexity of the underlying assumption and the size of the proofs. Some constructions, like [16], achieve constant-sized proofs but require a q-type assumption, while others [30] achieve proofs of any superconstant size under a constant-sized assumption. Here, we consider VRF constructions based on non-interactive (group-based) computational assumptions (NICA), i.e., search problems as opposed to a decisional assumptions. These NICAs state that any "efficient" algorithm only has a negligible probability of solving the corresponding computational problem, e.g. finding some "secret" exponent. In particular, we consider NICAs where the challenge elements' exponents only depend on a single variable with polynomial degree. These include for example the q-DLog-assumption and the q-DHI-assumption. There the challenge is $\mathbf{g}, \mathbf{g}^{\alpha}, \mathbf{g}^{\alpha^2}, \ldots, \mathbf{g}^{\alpha^q}$ and the secret exponent is α. We give two meta-reductions [14] (for slightly different settings) that break the resp. underlying assumption if there is an algebraic reduction from the assumption to the unpredictability (resp. pseudorandomness) of the VUF (resp. VRF).

Theorem 1 (Informal Lower Bound for Univariate VUFs). *Let* vuf *be a rational VUF whose verification key exponents depend—with polynomial degree—on a single common variable. Let* NICA *be any NICA of polynomial size. If there exists an algebraic reduction that transforms an adversary for the weak selective unpredictability of* vuf *into a solver for* NICA*, then* NICA *can be solved in polynomial time with some noticable advantage.*

Theorem 2 (Informal Lower Bound for Univariate NICAs). *Let* vrf *be a rational VRF. Let* NICA *be any NICA of polynomial size whose exponents depend—with polynomial degree—on a single common variable (e.g. q-*DLog *or q-*DHI*). If there exists an algebraic reduction that transforms an adversary for the weak selective pseudorandomness into a solver for* NICA*, then* NICA *can be solved in polynomial time with some noticable advantage.*

Remark 2 (Separation between Decisional and Computational Assumptions). As a theoretical sidenote, we observe that on the one hand non-interactive *decisional* assumptions, like q-DDH, suffice for constructing VRFs [46], while on the other hand (univariate) non-interactive *computational* assumptions, like the q-DLog or q-DHI assumption, do not suffice via algebraic reductions. This yields in particular an algebraic separation between the q-DDH and the q-DLog assumption.

Remark 3 (No Algebraic GL Construction). One can transform a VUF (e.g. the VUF of Dodis & Yampolskiy [16] based on the q-DHI assumption) into a VRF via the construction of Goldreich & Levin [21]. While this seems like a contradiction (because it gives a VRF based on the q-DHI assumption), it is actually consistent with our results because the GL hardcore bit is not an algebraic technique[5],

[5] The GL construction uses the bits of the representation of the group elements.

hence the reduction from the q-DHI assumption to the pseudorandomness of the resulting VRF is not an algebraic reduction. By contraposition, our results show that there cannot be an algebraic analogue of the GL construction.

Our Technique. Both meta-reductions share the same core idea. In a nutshell, the meta-reduction—when simulating an adversary towards the reduction—uses the representation vectors[6] of the received group elements to either (a) predict the challenge image, e.g. as a linear combination of received representations, or (b) construct a polynomial function over the exponent field $\mathbb{Z}_p$ which has the NICA's secret exponent as a zero. Thus, in case (a) the meta-reduction could successfully answer the reduction's challenge while in case (b) the meta-reduction can leverage the fact that polynomials over some finite field can be efficiently factorized and solve its own challenge directly using the NICA's secret exponent. In both cases the meta-reduction relies on the facts that the VUF (resp. VRF) has correctness and unique provability, and that the VUF (resp. VRF) is of *rational* form, i.e., $\mathsf{vrf}_{\mathsf{sk}}(x) = \mathbf{g_T}^{\sigma_x(\overrightarrow{v})/\rho_x(\overrightarrow{v})}$ where σ_x, ρ_x are of polynomial degree and $\overrightarrow{v}$ is the vector of verification key exponents. Because the reduction is algebraic, whenever it outputs a group element $\mathbf{y} \in \mathbb{G}_T$ it must also provide a representation $\overrightarrow{z} \in \mathbb{Z}_p^L$ w.r.t. the NICA challenge elements s.t.

$$\mathbf{g_T}^{\sigma_x(\overrightarrow{v})/\rho_x(\overrightarrow{v})} = \mathbf{y} = \mathbf{g_T}^{f_1(s)z_1+\ldots+f_L(s)z_L} \tag{1}$$

$$\iff \sigma_x(\overrightarrow{v}) - (f_1(s)z_1 + \ldots + f_L(s)z_L)\rho_x(\overrightarrow{v}) = 0 \tag{2}$$

where $\mathbf{g}^{(f_1(s),\ldots,f_L(s))} \in \mathbb{G}^L$ is the NICA challenge and $s \xleftarrow{\$} \mathbb{Z}_p$ is the secret exponent. Equation (2) is the basis for both meta-reductions. For Theorem 1 the meta-reduction queries many preimages $x_1, \ldots, x_Q$ and challenge x_0 uniformly at random. We consider two cases (for simple exposition we assume that the verification key only has one group element $\mathbf{g}^v$):
In the first case (a) the rational functions $\sigma_{x_i}(V)/\rho_{x_i}(V)$ are linearly dependent. With this linear dependence the meta-reduction can predict the challenge image by combining the representations of the queried images. [7]
In the second case (b) although the rational functions $\sigma_{x_i}(V)/\rho_{x_i}(V)$ are linearly independent, by a counting argument there must exist a linear dependence $\alpha \in \mathbb{Z}_p^Q$ among the representations of the queried preimages. The meta-reduction computes the polynomial $\psi(V) := \rho_{x_1}(V) \cdots \rho_{x_Q}(V) \cdot \sum_{\ell=1}^{Q} \alpha_\ell \sigma_\ell(V)/\rho_\ell(V)$. Because $\sigma_{x_i}(V)/\rho_{x_i}(V)$ are linearly independent, the polynomial is non-zero yet it contains the vk's exponent v as a zero (due to $\sum_{\ell=1}^{Q} \alpha_\ell \sigma_\ell(v)/\rho_\ell(v) = 0$). Thus the meta-reduction can factor the polynomial ψ to obtain the secret exponent and predict the challenge image as $\mathbf{g_T}^{\sigma_{x_0}(v)/\rho_{x_0}(v)}$.

For Theorem 2 we consider pseudorandomness, hence the meta-reduction obtains a representation for each verification key element and a representation

[6] Recall that we consider algebraic reductions here, so they have to output a vector of representations with each group element.

[7] If all $\sigma_{x_i}(V)/\rho_{x_i}(V)$ are linearly dependent, then with noticable probability the challenge's function $\sigma_{x_0}(V)/\rho_{x_0}(V)$ will be linearly dependent on the other rational functions because all x_i are independent and identitically distributed.

$\overrightarrow{z}^*$ for the challenge image $\mathbf{y}^*$. That is, the meta-reduction knows a function[8] $\xi : \mathbb{Z}_p \to \mathbb{Z}_p^L$ that maps the NICA challenge's secret key to the verification key exponents $\overrightarrow{v} = \xi(s)$. Plugging ξ into Eq. (2) gives

$$\sigma_x(\xi(s)) - (f_1(s)z_1 + \ldots + f_L(s)z_L)\rho_x(\xi(s)) = 0 \ . \tag{3}$$

Now, for any representation $\overrightarrow{z}$ of the real challenge image the univariate polynomial $\psi_{\overrightarrow{z}}(S) \coloneqq \sigma_x(\xi(S)) - (f_1(S)z_1 + \ldots + f_L(S)z_L)\rho_x(\xi(S))$ must vanish on the secret exponent s due to Eq. (3).
If $\psi_{\overrightarrow{z}}(S) \not\equiv 0$ is non-zero for all $\overrightarrow{z}$, then the meta-reduction can factorize $\psi_{\overrightarrow{z}^*}(S)$ and find a list of polynomially many candidates for the NICA's secret exponent. If no candidate matches the NICA's secret exponent, then the challenge image $\mathbf{y}^*$ must be random, otherwise the meta-reduction has trivially found the NICA's secret exponent.
On the other hand, if $\psi_{\overrightarrow{z}}(S) \equiv 0$ is zero for some $\overrightarrow{z}$, then the meta-reduction can efficiently find such a representation $\overrightarrow{z}$. Due to Eq. (3) such a $\overrightarrow{z}$ must correspond to the correct challenge image, hence the meta-reduction can distinguish the given element from random.

2.3 Third Result: Security of Low-Degree VRFs Requires Complex Assumptions

As explained before, Theorem 1 states that there is no algebraic reduction that transforms an adversary for the unpredictability of a rational VUF with polynomial evaluation degree to a solver for a hard polynomial size assumption. However, this result has the caveat that the VUF in question needs to have univariate verification keys, i.e., the verification key needs to be fully determined by one secret variable.

In the remaining part of this work, we will circumvent this problem and show lower bounds for another class of VUFs – the class of *rational parametrized VUFs* (see the full version[12]) – which imposes no restrictions on the verification keys of its VUFs. This class contains the candidates of Dodis & Yampolskiy [16] and of Belenkiy *et al.* [4] and all other DY-inspired candidates.

However, this result comes at a cost: It only shows the impossibility of *generic* reductions that transform adversaries for the unpredictability of parametrized VUFs into solvers of *extremely small* – yet superconstant – Uber-assumptions.

Informally, our result states the following:

Theorem 3 (Informal Lower Bound for Rational Parametrized VUFs). *Let* vuf *be a parametrized rational VUF of constant evaluation degree, i.e., it is rational and the numerators and denominators for evaluation depend polynomially on the input* $x \in \mathbb{Z}_p$. *Let* NICA *be an Uber-assumption of size* $\sqrt{\log \log \mathsf{poly}(\lambda)}$.

Then, there is no generic reduction that transforms an adversary for the weak selective unpredictability of vuf *to a* NICA *solver.*

[8] For simplicity assume that all f_i and hence ξ are polynomials.

We want to emphasize the significance of Theorem 3 for the pursuit of pairing-based VRFs with proofs of constant size. Theorem 3 shows that the security of each VUF in the style of [16] with constant proofs cannot be generically based on a constant-size Uber-assumption.

Now, we want to explain some details that appear in the statement of Theorem 3 before we jump to a proof:

Uber-Assumptions. We demand that NICA is an Uber-assumption [10], i.e., its challenges consist of group elements $\mathbf{g}, \mathbf{g}^{f_1(\overrightarrow{z})}, \ldots, \mathbf{g}^{f_{q_1}(\overrightarrow{z})}, \mathbf{g_T}^{g_1(\overrightarrow{z})}, \ldots, \mathbf{g_T}^{g_{q_2}(\overrightarrow{z})}$ where $\overrightarrow{z} \xleftarrow{\$} \mathbb{Z}_p^t$ has been sampled secretly and uniformly at random by the challenger and $f_1, \ldots, f_{q_1}, g_1, \ldots, g_{q_2} \in \mathbb{Z}_p[Z_1, \ldots, Z_t]$ are publicly known polynomials.

Parametrized Rational VUFs. It is required that vuf is parametrized rational of constant evaluation degree. Formally, this means there are constant-degree polynomials $\sigma, \rho \in \mathbb{Z}_p[V_1, \ldots, V_n, X]$ s.t. we have for each input $x \in \mathbb{Z}_p$ and each verification key vk and corresponding secret key sk

$$\mathsf{Eval}_{\mathsf{vuf}}(\mathsf{sk}, x) = \mathbf{g_T}^{\frac{\sigma(x, \overrightarrow{v})}{\rho(x, \overrightarrow{v})}}$$

where $\overrightarrow{v}$ denotes the vector of exponents of the group elements of vk.

We are now able to sketch a proof for Theorem 3:

Sketch of Proof, Part 1. Assume that Theorem 3 is false for some parametrized VUF vuf and let $\mathcal{R}$ be a reduction that solves instances of some Uber-assumption NICA when given access to an adversary for the unpredictability of vuf. To show a contradiction we construct a meta-reduction $\mathcal{M}$ that takes the role of a successful adversary in the weak selective unpredictability game with $\mathcal{R}$.

$\mathcal{R}$ is given a challenge $\mathbf{g}, \mathbf{g}^{f_1(\overrightarrow{z})}, \ldots, \mathbf{g}^{f_{q_1}(\overrightarrow{z})}, \mathbf{g_T}^{g_1(\overrightarrow{z})}, \ldots, \mathbf{g_T}^{g_{q_2}(\overrightarrow{z})}$ by the NICA challenger and has to compute some solution from this tuple of group elements while having oracle access to $\mathcal{M}$. Since $\mathcal{R}$ is a generic algorithm, we can apply a hybrid step and change the groups $\mathbb{G}, \mathbb{G}_T$ which encode elements of $\mathbb{Z}_p$ to groups $\mathbb{G}^Z, \mathbb{G}_T^Z$ that encode polynomials of $\mathbb{Z}_p[Z_1, \ldots, Z_t]$ without $\mathcal{R}$ noticing the internal change of groups. Additionally, the NICA challenger will now give the group elements $\mathbf{g}, \mathbf{g}^{f_1(\overrightarrow{Z})}, \ldots, \mathbf{g}^{f_{q_1}(\overrightarrow{Z})}, \mathbf{g_T}^{g_1(\overrightarrow{Z})}, \ldots, \mathbf{g_T}^{g_{q_2}(\overrightarrow{Z})}$ as challenge to $\mathcal{R}$. Further, because of the genericness of $\mathcal{R}$, the exponent of each target group element it outputs must be a polynomial of the form

$$\alpha + \sum_{i=1}^{q_1} \beta_i \cdot f_i(\overrightarrow{Z}) + \sum_{i,j=1} \gamma_{i,j} \cdot f_i(\overrightarrow{Z}) \cdot f_j(\overrightarrow{Z}) + \sum_{i=1}^{q_2} \delta_i \cdot g_i(\overrightarrow{Z}) \tag{4}$$

for scalars $\alpha, \beta_i, \gamma_{i,j}, \delta_i \in \mathbb{Z}_p$. Let W denote the vector space of all polynomials that can be expressed in the above way, i.e., $W = \mathrm{span}_{\mathbb{Z}_p}\{1, (f_i)_i, (f_i \cdot f_j)_{i,j}, (g_i)_i\} \subset \mathbb{Z}_p[Z]$. The space W contains the exponents of all target group elements that can be constructed by generic group operations and pairings from

the elements of the NICA challenge. In particular, the exponent of each group element outputted by $\mathcal{R}$ must lie in W.

Now, when $\mathcal{R}$ accesses $\mathcal{M}$ it sends a verification key vk, random inputs $x_0, \ldots, x_Q$, image values $\mathbf{y}_1, \ldots, \mathbf{y}_Q$ and proofs $\pi_1, \ldots, \pi_Q$ to $\mathcal{M}$. To win the unpredictability game, $\mathcal{M}$ needs to return the evaluation $\mathbf{y}_0$ of vuf at x_0 to $\mathcal{R}$. As stated above, the exponents of each group element of vk and of the image values $\mathbf{y}_1, \ldots, \mathbf{y}_Q$ must lie in W. Let $v_1(\overrightarrow{Z}), \ldots, v_n(\overrightarrow{Z}), y_1(\overrightarrow{Z}), \ldots, y_Q(\overrightarrow{Z}) \in W$ be exponents of these group elements. Since $\mathcal{R}$ is generic, $\mathcal{M}$ can extract those polynomials from $\mathcal{R}$ while playing the unpredictability game with $\mathcal{R}$ (we assume in this work that genericness always implies algebraicity, cf. Definition 5). With the help of $\pi_1, \ldots, \pi_Q$ the meta-reduction $\mathcal{M}$ can ensure that for each $i \in [Q]$ the equation

$$\frac{\sigma(x_i, v_1(\overrightarrow{Z}), \ldots, v_n(\overrightarrow{Z}))}{\rho(x_i, v_1(\overrightarrow{Z}), \ldots, v_n(\overrightarrow{Z}))} = y_i(\overrightarrow{Z}) \tag{5}$$

holds.

Sketch of Proof, Part 2. In the first part of the proof, we showed that the fractions $\frac{\sigma(x_i, \overrightarrow{v}(\overrightarrow{Z}))}{\rho(x_i, \overrightarrow{v}(\overrightarrow{Z}))}$, $i \in [Q]$, are not only polynomials, but additionally lie in W. This is the point where we can spring our mathematical trap: we can show if all fractions $\frac{\sigma(x_1, \overrightarrow{v}(\overrightarrow{Z}))}{\rho(x_1, \overrightarrow{v}(\overrightarrow{Z}))}, \ldots, \frac{\sigma(x_Q, \overrightarrow{v}(\overrightarrow{Z}))}{\rho(x_Q, \overrightarrow{v}(\overrightarrow{Z}))}$ lie in W for a large enough number Q then, in fact, the fraction $\frac{\sigma(x, \overrightarrow{v}(\overrightarrow{Z}))}{\rho(x, \overrightarrow{v}(\overrightarrow{Z}))}$ must be an element of W for *each* $x \in \mathbb{Z}_p$. In particular, the exponent $\frac{\sigma(x_0, \overrightarrow{v}(\overrightarrow{Z}))}{\rho(x_0, \overrightarrow{v}(\overrightarrow{Z}))}$ of $\mathbf{y}_0$ must be of this form and therefore $\mathcal{M}$ can compute the element $\mathbf{y}_0 = \mathbf{g_T}^{\frac{\sigma(x_0, \overrightarrow{v}(\overrightarrow{Z}))}{\rho(x_0, \overrightarrow{v}(\overrightarrow{Z}))}}$ from the group elements of the NICA challenge on its own. Ergo, $\mathcal{M}$ can successfully answer the queries of $\mathcal{R}$ for a large enough number of queries Q which gives rise to a generic PPT NICA solver. A contradiction to the hardness of NICA!

2.4 Organization of this Work

In Section 3, we introduce notations and preliminaries. In Section 4, we define consecutive verifiable and rational VUFs and show our first result: a consecutive verifiable VUF is rational and its evaluation degree is exponentially bounded by the size of its proofs. In Section 5, we show our second result: Theorem 1 and Theorem 2, which state that the security of rational VUFs cannot be based by an algebraic reduction on the hardness of a NICA, if either the verification key of the VUF or the NICA is univariate. Finally, in Section 6, we introduce the notion of parametrized rational VUFs and Uber-assumptions, state the formal version of Theorem 3 and give a very high-level idea of its proof.

3 Preliminaries

3.1 Notation

We denote the security parameter by λ. We denote vectors by $\overrightarrow{x}$ and group elements by $\mathbf{g}$. For a matrix M we denote by $m_{i,j}$ the entry in the i-th row and the j-th column. For a finite set X we denote by $x \xleftarrow{\$} X$ that x is sampled uniformly at random from X.

For a probabilistic algorithm Alg we denote by $y \xleftarrow{\$} \mathsf{Alg}(x)$ that y is computed by Alg on input x with a uniform random tape. Set further $\mathsf{poly}(\lambda) := \{f : \mathbb{N} \to \mathbb{N} \mid \exists a, b \in \mathbb{N}, \forall n \in \mathbb{N} : \ f(n) \leq a + n^b\}$ and $\mathsf{negl}(\lambda) := \{\varepsilon : \mathbb{N} \to \mathbb{R} \mid \forall c \in \mathbb{N} : \lim_{n\to\infty} n^c \cdot \varepsilon(n) = 0\}$. For any $n \in \mathbb{N}$ we set $[n] := \{1, \ldots, n\}$. We call an algorithm PPT iff it is probabilistic, and its time complexity lies in $\mathsf{poly}(\lambda)$.

3.2 Mathematical Foundations

Definition 1 (Rational Functions). *For a prime p we define the field of **rational functions** over $\mathbb{Z}_p$ in variables $X_1, \ldots, X_n$ by*

$$\mathbb{Z}_p(X_1, \ldots, X_n) := \left\{ \frac{\sigma(X_1, \ldots, X_n)}{\rho(X_1, \ldots, X_n)} \middle| \sigma, \rho \in \mathbb{Z}_p[X_1, \ldots, X_n],\ \rho \neq 0 \right\}.$$

*Given a rational function $f \in \mathbb{Z}_p(X_1, \ldots, X_n)$, the **degree** of f is defined as*

$$\deg(f) := \min\{\max(\deg(\sigma), \deg(\rho)) \mid \sigma, \rho \in \mathbb{Z}_p[X_1, \ldots, X_n], \rho \neq 0, \rho \cdot f = \sigma\}$$

where $\deg(\sigma), \deg(\rho)$ denote the total degrees of the polynomials σ, ρ.

We recall the following helpful lemma:

Lemma 1 (Schwartz-Zippel-Lemma, [44]). *Let $f \in \mathbb{Z}_p[X_1, \ldots, X_n]$ be a non-zero polynomial over $\mathbb{Z}_p$. Denote by $\deg(f)$ the total degree of f. Then*

$$\Pr_{r_1,\ldots,r_n \xleftarrow{\$} \mathbb{Z}_p}[f(r_1, \ldots, r_n) = 0] \leq \frac{\deg(f)}{p}.$$

3.3 Cryptographic Groups

Definition 2 (Bilinear Group Generator, [24]). *A **bilinear group generator** is a probabilistic polynomial-time algorithm GrpGen that takes as input a security parameter λ (in unary) and outputs $\Pi = (p, \mathsf{pp}_{\mathbb{G}}, \mathsf{pp}_{\mathbb{G}_T}, \circ, \circ_{\mathsf{T}}, e, \phi(1)) \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$ such that the following requirements are satisfied.*

1. *The parameter p is prime and $\log(p) \in \Omega(\lambda)$.*
2. *$\mathbb{G}$ and $\mathbb{G}_T$ as described by $\mathsf{pp}_{\mathbb{G}}$ and $\mathsf{pp}_{\mathbb{G}_T}$ are subsets of $\{0,1\}^*$, defined by algorithmic descriptions of maps $\phi : \mathbb{Z}_p \to \mathbb{G}$ and $\phi_{\mathsf{T}} : \mathbb{Z}_p \to \mathbb{G}_T$.*
3. *$\circ$ and $\circ_{\mathsf{T}}$ are algorithmic descriptions of efficiently computable (in λ) maps $\circ : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ and $\circ_{\mathsf{T}} : \mathbb{G}_T \times \mathbb{G}_T \to \mathbb{G}_T$, such that*

   (a) $(\mathbb{G}, \circ)$ *and* $(\mathbb{G}_T, \circ_{\mathsf{T}})$ *form abstract groups and*
   (b) ϕ *is a group isomorphism from* $(\mathbb{Z}_p, +)$ *to* $(\mathbb{G}, \circ)$ *and*
   (c) ϕ_{T} *is a group isomorphism from* $(\mathbb{Z}_p, +)$ *to* $(\mathbb{G}_T, \circ_{\mathsf{T}})$.
4. e *is an algorithmic description of an efficiently computable (in* λ*) bilinear map* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. *We require that* e *is non-degenerate, i.e.,* $x \neq 0 \implies e(\phi(x), \phi(x)) \neq \phi_{\mathsf{T}}(0)$.

Remark 4. For simplicity, we only consider symmetric pairings. However, while our upcoming formulation of "consecutive verifiability" is easier to state with symmetric pairings, our results do not depend on symmetry of the pairing.

Definition 3 (Certified Generator, [24]). *We say a bilinear group generator* GrpGen *is* ***certified****, if there exists a deterministic polynomial-time algorithm* GrpVfy *with the following properties:*

Parameter Validation. *Given a string* Π *(which may not necessarily be generated by* GrpGen*), algorithm* $\mathsf{GrpVfy}(\Pi)$ *outputs* 1 *if and only if* Π *has the form* $\Pi = (p, \mathsf{pp}_{\mathbb{G}}, \mathsf{pp}_{\mathbb{G}_T}, \circ, \circ_{\mathsf{T}}, e, \phi(1))$ *and all requirements from Definition 2 are satisfied.*

Recognition and Unique Representation of Elements of $\mathbb{G}$ ***(***$\mathbb{G}_T$***).*** *Furthermore, we require that each element in* $\mathbb{G}$ *(*$\mathbb{G}_T$*) has a unique representation, which can be efficiently recognized. That is, on input two strings* Π *and* s, $\mathsf{GrpVfy}(\Pi, s)$ *outputs* 1 *if and only if* $\mathsf{GrpVfy}(\Pi) = 1$ *and it holds that* $s = \phi(x)$ *(*$s = \phi_{\mathsf{T}}(x)$*) for some* $x \in \mathbb{Z}_p$. *Here* $\phi : \mathbb{Z}_p \to \mathbb{G}$ *(*$\phi_{\mathsf{T}} \colon \mathbb{Z}_p \to \mathbb{G}_T$*) denotes the fixed group isomorphism contained in* Π *to specify the representation of elements of* $\mathbb{G}$ *(of* $\mathbb{G}_T$*) (see Definition 2).*

We recall the definition of algebraic algorithms which was first used by [9, 42] in the context of meta-reductions. Our definition of algebraic algorithms is closer to that of [3, 19].

Definition 4 (Algebraic Algorithms [3, 19]). *Let* $\mathsf{pp}_{\mathcal{G}} = (p, \mathsf{pp}_{\mathbb{G}}, \mathsf{pp}_{\mathbb{G}_T}, \circ_{\mathbb{G}}, \circ_{\mathbb{G}_T}, e, \phi_{\mathbb{G}}, \phi_{\mathbb{G}_T})$ *be as in Definition 2. Let* $\mathcal{A}$ *be an algorithm that receives as input source group elements* $\mathbf{g}_1, \ldots, \mathbf{g}_s \in \mathbb{G}$, *target group elements* $\mathbf{h}_1, \ldots, \mathbf{h}_t \in \mathbb{G}_T$ *and some non-group-element input* x.

We say that $\mathcal{A}$ *is* ***algebraic*** *if, whenever* $\mathcal{A}$ *outputs a group element* $\mathbf{y}$, *it also outputs one of the following representations: If* $\mathbf{y} \in \mathbb{G}$, *a vector*

$$\overrightarrow{z} \in \mathbb{Z}_p^s \quad s.t. \quad \mathbf{y} = \prod_{i=1}^{s} \mathbf{g}_i^{z_i}$$

and if $\mathbf{y} \in \mathbb{G}_T$, *a vector and a matrix*

$$\overrightarrow{z} \in \mathbb{Z}_p^t, M = (m_{ij})_{i,j=1}^{s} \in \mathbb{Z}_p^{s \times s} \quad s.t. \quad \mathbf{y} = \prod_{i=1}^{t} \mathbf{h}_i^{z_i} \cdot \left(\prod_{i,j=1}^{s} e(\mathbf{g}_i, \mathbf{g}_j)^{m_{ij}} \right)$$

Definition 5 (The Generic Group Model [40, 45]). *An algorithm interacting with a group (or pairing group) is called* ***generic*** *if it is algebraic in the sense of Definition 4* and *it suffices that the algorithm accesses the group only through an oracle. More concretely, all group elements* $\mathbf{g}_i$ *that the algorithm receives as input are represented by random strings* $\sigma(\mathbf{g}_i)$*, called* ***handles****, and whenever the algorithm wants to compute the product* $\mathbf{g}_i \cdot \mathbf{g}_j$ *resp. the exponentiation* $\mathbf{g}^x$*, it passes* $(\sigma(\mathbf{g}_i), \sigma(\mathbf{g}_j))$ *resp.* $(\sigma(\mathbf{g}_i), x)$ *to the corresponding group operation oracle, and the oracle returns* $\sigma(\mathbf{g}_i \cdot \mathbf{g}_j)$ *resp.* $\sigma(\mathbf{g}_i^x)$*. In a pairing setting the algorithm is given access to a second such group oracle for the target group, as well as a pairing oracle that takes as input two handles* $\sigma(\mathbf{g}_i), \sigma(\mathbf{g}_j)$ *and outputs* $\sigma(e(\mathbf{g}_i, \mathbf{g}_j))$ *if both elements* $\mathbf{g}_i, \mathbf{g}_j$ *are elements of the source group.*

Remark 5. It has been shown recently – despite popular belief – that an algorithm that only interacts with a group by oracles in Shoup's GGM does not need to be algebraic [29, 48]. To circumvent this problem, we require in the definition of generic algorithms explicitly that a generic algorithm is algebraic.

Remark 6. It is not clear how to adapt the notion of a certified group generator (Definition 3) to generic groups. Indeed, in the generic group model, there are no group descriptions as in Definition 2, and instead all algorithms have access to a group via group operation oracles. However, these oracles can be viewed as "implicitly trusted", in the sense that the properties from Definition 2 are always guaranteed. Hence, we will not consider certified (bilinear) group generators in the context of generic groups.

Definition 6 (Non-Interactive Computational Assumptions, NICAs [18]). *A* ***non-interactive computational assumption*** NICA *is defined by the following two oracles available to the adversary:*

Setup *Generates a challenge* $c \xleftarrow{\$} \mathcal{D}(1^\lambda)$ *from a challenge distribution* $\mathcal{D}(1^\lambda)$ *parameterized over the security parameter* λ*. Saves an internal state* st*.*

Finalize *On input of a candidate solution* s *and the internal state* st*, outputs either* 1 *(indicating that* s *is a correct solution) or* 0 *(indicating that* s *is not a correct solution).*

We say that an adversary $\mathcal{A}$ (t, ϵ)-***breaks*** *the assumption if the adversary outputs a correct solution with probability at least* $\epsilon(\lambda)$ *in time at most* $t(\lambda)$*. We further say the assumption is* (t, ϵ)-***hard*** *if there exists no adversary* $\mathcal{A}$ *that* (t, ϵ)*-breaks the assumption. If* NICA *is* $(t, \frac{1}{r})$*-hard for all* $t, r \in \mathsf{poly}(\lambda)$*,* $r > 0$*, we call* NICA ***hard****.*

For a NICA in a group where the challenge consists of m *group elements, we call* m *the* ***size*** *of the NICA. If* m *is linear in a parameter* q*, we call* NICA *a* q***-type assumption****. If* m *is constant we call* NICA *a* ***constant-size assumption****.*

Definition 7 (Univariate Polynomial-Degree Assumptions). *Let* $p = p(\lambda)$ *be a superpolynomial group order. Let* $l_1, l_2, d_{\mathsf{NICA}} \in \mathsf{poly}(\lambda)$*, let* $r_1, \ldots, r_{l_1}$*,* $t_1, \ldots, t_{l_2} \in \mathbb{Z}_p[S]$ *be non-zero polynomials of degree at most* d_{NICA}*. We say*

NICA *is a univariate polynomial-degree assumption, iff it is an* $(l_1 + l_2)$*-type NICA according to Definition 6 and if its challenge distribution*[9] *is* $\mathcal{D}(1^\lambda) \to c = (\Pi, \mathbf{g}^{r_1(s)}, \ldots, \mathbf{g}^{r_{l_1}(s)}, \mathbf{g}^{1/t_1(s)}, \ldots, \mathbf{g}^{1/t_{l_2}(s)})$ *where* $s \xleftarrow{\$} \mathbb{Z}_p$ *is the secret exponent and* $\Pi = (p, \mathsf{pp}_{\mathbb{G}}, \mathsf{pp}_{\mathbb{G}_T}, \circ, \circ_{\mathsf{T}}, e, \phi(1)) \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$ *is a certified group description.*

Definition 8 (DLog-Hard Assumptions). *Let* $l_1, l_2, d_{\mathsf{NICA}} \in \mathsf{poly}(\lambda)$*, let* $r_1, \ldots, r_{l_1}, t_1, \ldots, t_{l_2} \in \mathbb{Z}_p[S]$ *be non-zero polynomials of degree at most* d_{NICA}*. We say* NICA *is a DLog-hard assumption, iff it is an* $(l_1 + l_2)$*-type assumption according to Definition 7 and if no polynomial-time algorithm has noticable probability of solving the corresponding DLog problem, i.e., outputting the secret exponent* $s \in \mathbb{Z}_p$*.*

Remark 7. In particular the *computational* q-DHI assumption (Diffie-Hellman inversion assumption) is a univariate polynomial-degree assumption for $q \in \mathsf{poly}(\lambda)$. The decisional variant is *not* univariate because of the last challenge element.

3.4 Verifiable Unpredictable Functions

Definition 9 (Verifiable Unpredictable Functions, VUFs [36]). *Let* $\mathsf{vuf} = (\mathsf{Gen}_{\mathsf{vuf}}, \mathsf{Eval}_{\mathsf{vuf}}, \mathsf{Verify}_{\mathsf{vuf}})$ *be a tuple of algorithms of the following form:*

- $\mathsf{Gen}_{\mathsf{vuf}}(1^\lambda)$ *outputs a secret key* sk *and a verification key* vk*.*
- $\mathsf{Eval}_{\mathsf{vuf}}(\mathsf{sk}, x)$ *on input a secret key* sk *and* $x \in \mathcal{X} = (\mathcal{X}_\lambda)_\lambda$ *outputs an image* $y \in \mathcal{Y} = (\mathcal{Y}_\lambda)_\lambda$ *and a proof* π*. We assume that the input space* $\mathcal{X}_\lambda$ *has a superpolynomial cardinality in the security parameter* λ*.*
- $\mathsf{Verify}_{\mathsf{vuf}}(\mathsf{vk}, x, y, \pi)$ *on input a verification key* vk*, a preimage* x*, an image* y *and a proof* π *outputs a bit* $b \in \{0, 1\}$*.*

We say that vuf *is a* (t, Q, ϵ)-***verifiable unpredictable function*** *(VUF) if the following holds:*

Statistical Correctness*. There exists a negligible function* $\mu \in \mathsf{negl}(\lambda)$ *s.t. for all* $\lambda \in \mathbb{N}$ *and for all inputs* $x \in \mathcal{X}_\lambda$ *it holds that*

$$\Pr_{(\mathsf{sk},\mathsf{vk}) \xleftarrow{\$} \mathsf{Gen}_{\mathsf{vuf}}(1^\lambda)} [\mathsf{Verify}_{\mathsf{vuf}}(\mathsf{vk}, x, y, \pi) = 1 \mid (y, \pi) \leftarrow \mathsf{Eval}_{\mathsf{vuf}}(\mathsf{sk}, x)] \geq 1 - \mu(\lambda) \ .$$

Unique Provability*. For all* $\lambda \in \mathbb{N}$ *and all possible* vk *(not necessarily generated by* $\mathsf{Gen}_{\mathsf{vuf}}$*), all* $x \in \mathcal{X}_\lambda$*, all* $y_1, y_2 \in \mathcal{Y}_\lambda$ *and all possible proofs* π_1, π_2 *it holds that*

$$\mathsf{Verify}_{\mathsf{vuf}}(\mathsf{vk}, x, y_1, \pi_1) = 1 \land \mathsf{Verify}_{\mathsf{vuf}}(\mathsf{vk}, x, y_2, \pi_2) = 1 \implies y_1 = y_2$$

[9] For exposition, we assume all group element to be in the source group. Our technique applies as well for assumptions with target group elements.

Weak Q-Selective Unpredictability [11]. *For any adversary $\mathcal{A}$ running in time at most $t(\lambda)$, we have*

$$\left|\Pr\left[\mathcal{A}(\mathsf{vk},\overrightarrow{x},\overrightarrow{\mathbf{y}},\overrightarrow{\pi})=\mathbf{y}_0 \;\middle|\; \begin{array}{l}\overrightarrow{x}=(x_0,\ldots,x_Q)\xleftarrow{\$}\mathcal{X}_\lambda^{Q+1}\\(\mathsf{sk},\mathsf{vk})\xleftarrow{\$}\mathsf{Gen}_{\mathsf{vrf}}(1^\lambda)\\(\mathbf{y}_i,\pi_i)\leftarrow\mathsf{Eval}_{\mathsf{vrf}}(\mathsf{sk},x_i)\\\overrightarrow{\mathbf{y}}=(\mathbf{y}_1,\ldots,\mathbf{y}_Q)\\\overrightarrow{\pi}=(\pi_1,\ldots,\pi_Q)\end{array}\right]-\frac{1}{|\mathcal{Y}_\lambda|}\right|\leq\epsilon(\lambda)\ .$$

Remark 8. Our notion of weak selective unpredicability is even weaker than the eponymous notion used by Niehues [41] with a loss of $1/Q$ by guessing the adversary's challenge index and reordering the preimages. However, our notion has the advantage that it is a non-interactive game, in particular, no state has to be transmitted between parts of the adversary $(\mathcal{A}_1,\mathcal{A}_2)$ as in [41].

Remark 9. We note that we do not require *perfect* correctness as for some of the VUFs we consider in this work this property does not hold perfectly (e.g. in the case where $\mathsf{Eval}_{\mathsf{vuf}}(\mathsf{sk},x)$ is undefined for a small number of $x\in\mathcal{X}$ for some secret key sk).

Remark 10. We consider pairing-based VUFs where $y\in(\mathbb{G}\cup\mathbb{G}_T)$ and $\pi\in(\mathbb{G}\cup\mathbb{G}_T)^*$. W.l.o.g. we assume that a VUF's image is an element of the target group, i.e., $\mathcal{Y}=\mathbb{G}_T$. Otherwise, we can modify the VUF by appending the original (source group) image $\mathbf{y}_\mathsf{S}\in\mathbb{G}$ to the proof elements, and set the new image as $\mathbf{y}_\mathsf{T}\coloneqq e(\mathbf{g}_\mathsf{S},\mathbf{y}_\mathsf{S})$ where $\mathbf{g}_\mathsf{S}$ is a designated generator of the source group in the verification key. Obviously, the unpredictability of the former VUF can be reduced to the unpredictability of latter, without any loss.

Definition 10 (Verifiable Random Functions, VRFs [36]). *Let* $\mathsf{vrf}=(\mathsf{Gen}_{\mathsf{vrf}},\mathsf{Eval}_{\mathsf{vrf}},\mathsf{Verify}_{\mathsf{vrf}})$ *be a VUF according to Definition 9. We say that* vrf *is a* (t,ϵ)-***verifiable random function*** *(VRF) if the following*[10] *holds:*

Weak Q-Selective Pseudorandomness. *For any adversary $\mathcal{A}$ running in time at most $t(\lambda)$, we have*

$$\left|\Pr\left[\mathcal{A}(\mathsf{vk},\overrightarrow{x},\overrightarrow{\mathbf{y}}^b,\overrightarrow{\pi})=b \;\middle|\; \begin{array}{l}\overrightarrow{x}=(x_0,\ldots,x_Q)\xleftarrow{\$}\mathcal{X}_\lambda^{Q+1}\\(\mathsf{sk},\mathsf{vk})\xleftarrow{\$}\mathsf{Gen}_{\mathsf{vrf}}(1^\lambda)\\(\mathbf{y}_i,\pi_i)\leftarrow\mathsf{Eval}_{\mathsf{vrf}}(\mathsf{sk},x_i)\\\mathbf{y}'_0\leftarrow\mathbb{G}_T\\\overrightarrow{\mathbf{y}}^0=(\mathbf{y}_0,\mathbf{y}_1,\ldots,\mathbf{y}_Q)\\\overrightarrow{\mathbf{y}}^1=(\mathbf{y}'_0,\mathbf{y}_1,\ldots,\mathbf{y}_Q)\\\overrightarrow{\pi}=(\pi_1,\ldots,\pi_Q)\\b\leftarrow\{0,1\}\end{array}\right]-\frac{1}{2}\right|\leq\epsilon(\lambda)\ .$$

[10] To keep the definitions minimal, we choose to only present the 0-selective pseudorandomness property since it is the security notion considered in our results.

3.5 Reductions

Definition 11. *For a VUF* vuf *and a NICA* NICA*, we say a Turing machine* $\mathcal{B}$ *is a* $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}}, r, Q, \epsilon_{\mathcal{A}})$-***reduction*** *from breaking* NICA *to breaking the weak selective unpredictability of* vuf*, if for any* $\mathcal{A}$ *that* $(t_{\mathcal{A}}, Q, \epsilon_{\mathcal{A}})$*-breaks the weak selective unpredicability of* vuf*, the TM* $\mathcal{B}^{\mathcal{A}}$ $(t_{\mathcal{B}} + rt_{\mathcal{A}}, \epsilon_{\mathcal{B}})$*-breaks* NICA *making at most* r *oracle queries*[11] *to* $\mathcal{A}$*.*

4 Proof Size

4.1 Classes of VUFs over Pairing-Friendly Groups

In the following, we introduce the class of VUFs that we want to discuss. Informally speaking, we consider VUFs whose verification algorithm only verifies group membership and pairing equations over the proof, evaluation value, and verification key. We further require that the verification algorithm is *consecutive*, i.e., it first verifies the first element of the proof, then the second, then the third, and so on and at the end of its execution it verifies that the evaluation value is correct. This class of VUFs covers many existing VUFs, we refer to Fig. 1 for an overview of which VUFs are consecutively verifiable.

In this section, we want to show that the evaluation function of VUFs that have such a natural verification algorithm can be expressed as a target group element where the exponent is a rational function in the discrete logarithms of the verification key element and that, informally speaking, the degree of the rational function can be bounded as exponential in the size of the proof. We begin by giving a formal definition of what we consider a set of pairing equations.

Definition 12 (Pairing Equations). *Let* $E \subset \mathbb{Z}_p[X_1, \ldots, X_m]$*. We call* E *a set of* ***pairing equations*** *for a pairing group* $\mathcal{G}$ *with public parameters* $\Pi = (p, \mathsf{pp}_{\mathbb{G}}, \mathsf{pp}_{\mathbb{G}_T}, \circ, \circ_{\mathsf{T}}, e, \phi(1)) \xleftarrow{\$} \mathsf{GrpGen}(1^{\lambda})$ *over variables* $\overrightarrow{X} = X_1, \ldots, X_m$ *with target indicator*[12] *set* $T \subset \{1, \ldots, m\}$ *if the following hold:*

1. $\max_{f \in E}(\deg f) \leq 2$*,*
2. *for all* $i \in T$ *and* $f \in E$ *it holds that if* X_i *appears in a monomial* m *of* f*, then* $m = c \cdot X_i$ *for some* $c \in \mathbb{Z}_p$*.*

We describe the evaluation *of a finite set of pairing equations* E *on input* $\mathbf{x}_1, \ldots, \mathbf{x}_m$ *as follows:*

- *We check that the input is a set of group elements* $(\mathbf{x}_1, \ldots \mathbf{x}_m)$*, i.e.,* $\mathbf{x}_i \in \mathbb{G}$ *or* $\mathbf{x}_i \in \mathbb{G}_T$ *for all* i*, and output* $\perp$ *if otherwise.*
- *For each* $i \in [m]$*, we check if* $i \in T \iff \mathbf{x}_i \in \mathbb{G}_T$ *and output* $\perp$ *if otherwise.*

[11] Because our weak selective unpredictability is a non-interactive game, there are no concurrency issues.

[12] This set indicates which verification key elements are in the target group. Hence, their exponents should only occur linearly, while source group exponents can occur quadratically.

- *For* $f = \left(\sum_{m \in M_f} m\right) \in E$ *where* M_f *is the set of monomials of* f*, we compute* $\mathbf{f}(\overrightarrow{\mathbf{x}}) \coloneqq \prod_{m \in M_f} \mathbf{m}(\overrightarrow{\mathbf{x}})$ *where* $\mathbf{m}(\overrightarrow{\mathbf{x}})$ *are computed as follows:*
  - *if* $m = c \cdot X_i \cdot X_j$ *for some* $i, j \notin T$ *and* $c \in \mathbb{Z}_p$*, compute* $\mathbf{m}(\overrightarrow{\mathbf{x}}) \coloneqq e(\mathbf{x}_i, \mathbf{x}_j)^c$*,*
  - *if* $m = c \cdot X_i$ *for some* $i \notin T$ *and some* $c \in \mathbb{Z}_p$ *and if* $\mathbf{x}_i \in \mathbb{G}$*, compute* $\mathbf{m}(\overrightarrow{\mathbf{x}}) \coloneqq e(\mathbf{x}_i, \mathbf{g})^c$ *where* $\mathbf{g} = \phi(1)$ *is the fixed generator of* $\mathbb{G}$ *as given in the group parameters* Π*. If* $i \in T$ *and* $\mathbf{x}_i \in \mathbb{G}_T$ *compute* $\mathbf{m}(\overrightarrow{\mathbf{x}}) \coloneqq \mathbf{x}_i^c$*,*
  - *if* $m = c$ *for* $c \in \mathbb{Z}_p$*, compute* $\mathbf{m}(\overrightarrow{\mathbf{x}}) \coloneqq e(\mathbf{g}, \mathbf{g})^c$*.*
- *We denote by* $E(\overrightarrow{\mathbf{x}})$ *the function that outputs* 1 *if for all* $f \in E$ *it holds that* $\mathbf{f}(\overrightarrow{\mathbf{x}}) = e(\mathbf{g}, \mathbf{g})^0$ *(if* $E = \emptyset$ *this always holds) and otherwise outputs* 0*.*

In the following we describe our class of VUFs that have a consecutive verification algorithm.

Definition 13 (Consecutively Verifiable Pairing-Based VUFs). *We say a VUF* $\mathsf{vuf} = (\mathsf{Gen}_{\mathsf{vuf}}, \mathsf{Eval}_{\mathsf{vuf}}, \mathsf{Verify}_{\mathsf{vuf}})$ *with input space* $\mathcal{X}$ *is a* ***consecutively verifiable pairing-based VUF*** *if the following hold:*

1. $\mathsf{Gen}_{\mathsf{vuf}}$ *takes as input* 1^λ*. It samples group parameters* $\Pi = (p, \mathsf{pp}_{\mathbb{G}}, \mathsf{pp}_{\mathbb{G}_T}, \circ, \circ_{\mathsf{T}}, e, \mathbf{g} \coloneqq \phi(1)) \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$ *and outputs a verification key* $\mathsf{vk} = (\Pi, \overrightarrow{\mathbf{v}})$ *such that* $\overrightarrow{\mathbf{v}}$ *consists of elements of* $\mathbb{G}$ *and* $\mathbb{G}_T$ *(plus a secret key* sk *for which we make no further constraints).*
2. *All function values* $\mathbf{y}$ *consist of values in* $\mathbb{G}_T$*.*
3. *All proofs consist of* κ *values in* $\mathbb{G} \cup \mathbb{G}_T$*.*
4. *For all* $x \in \mathcal{X}$ *and all* $i \in [\kappa + 1]$*, there exists a set* $E_{i,x}$ *of pairing equations that can be efficiently derived from* x *and the description of* vuf*. We require that* $E_{i,x} \subset \mathbb{Z}_p[V_1, \ldots, V_n, P_1, \ldots, P_i]$ *such that there is at least one polynomial of the form* $a_{i,x} \cdot P_i + b_{i,x} \in E_{i,x}$ *where* $a_{i,x}, b_{i,x} \in \mathbb{Z}_p[V_1, \ldots, V_n, P_1, \ldots, P_{i-1}]$*. (We note that since the set* $E_{i,x}$ *consists of pairing equations it holds that* $a_{i,x}$ *has degree at most* 1 *and* $b_{i,x}$ *has degree at most* 2*.)*
5. *We require that* $\mathsf{Verify}_{\mathsf{vuf}}$ *on input* $(\mathsf{vk} = (\Pi, \overrightarrow{\mathbf{v}}), x, \mathbf{y} =: \pi_{\kappa+1}, \overrightarrow{\pi})$ *outputs* 1 *if and only if the following hold:* $\mathsf{GrpVfy}(\Pi) = 1$*, all* $\mathbf{v}_i$*, for* $i \in [n]$*, and all* π_i*, for* $i \in [\kappa + 1]$*, are valid group elements w.r.t.* Π*, and for all* $i \in [\kappa + 1]$ *we have* $E_{i,x}(\overrightarrow{\mathbf{v}}, \pi_1, \ldots, \pi_i) = 1$*.*
6. *We further require that the ideal* $(E_{1,x}, \ldots, E_{\kappa+1,x}, a_{1,x} \cdot \ldots \cdot a_{\kappa+1,x})$ *(which is generated by the elements of* $E_{1,x}$*, ...,* $E_{\kappa+1,x}$ *and the polynomial* $a_{1,x} \cdot \ldots \cdot a_{\kappa+1,x}$*) contains the constant polynomial 1 (i.e.,* $(E_{1,x}, \ldots, E_{\kappa+1,x}, a_{1,x} \cdot \ldots \cdot a_{\kappa+1,x}) = \mathbb{Z}_p[V_1, \ldots, V_k, P_1, \ldots, P_{\kappa+1}]$*).*

Requirement 4 will be useful in Lemma 2, as it basically means there needs to be at least one equation that contains the current proof element as a linear factor only. This yields in particular that the proof element in question is not a (non-unique) square root of other elements. The last requirement on a consecutively verifiable pairing-based VUF might seem odd, however, as we will see later, it makes sure that there is no tuple $(\mathsf{vk}, x, \mathbf{y}, \pi)$ s.t. any of the a_i can evaluate to zero on the exponents of $(\mathsf{vk}, x, \mathbf{y}, \pi)$.

Remark 11 (On VRFs with multiple output group elements.). We restrict our framework to VRFs with a single group element in the output. For VRFs with δ elements in the output, we propose the following adaption of the definition of consecutive verifiability: For each output element, we add a formal variable $P_{i_1}, \ldots, P_{i_\delta}$ to the polynomial ring. For consecutivity, we require a partial ordering of all $\kappa + \delta$ variables P_i, where the last element is required to be an output value. We further require that the conditions of Definition 13 hold w.r.t. the partial ordering. Such a consecutively verifiable multi-output VRF implies a consecutively verifiable single-output VRF that uses the last output element as its output and puts all other elements into the proof.

As our results apply to VRFs with a single output element, they also apply to VRFs that are obtained from multi-output VRFs through the transformation described above with the proof size adapted accordingly.

We now define the class of VUFs that evaluate a rational function in the exponent. We will show later that a VUF that fulfills Definition 13 and where the number of group elements in the proof is in $O(\log(\lambda))$ also fulfills Definition 14.

Definition 14 (Rational VUFs). *Let $d, n \in \mathsf{poly}(\lambda)$. We say that a VUF $\mathsf{vuf} = (\mathsf{Gen}_{\mathsf{vuf}}, \mathsf{Eval}_{\mathsf{vuf}}, \mathsf{Verify}_{\mathsf{vuf}})$ is **rational** of **evaluation degree** d with $n = n_{\mathsf{S}} + n_{\mathsf{T}}$ verification key elements, if the verification key is of the form $\mathsf{vk} = (\Pi, \overrightarrow{\mathbf{v}})$ where $\Pi \coloneqq (p, \mathsf{pp}_{\mathbb{G}}, \mathsf{pp}_{\mathbb{G}_T}, \circ, \circ_{\mathsf{T}}, e, \mathbf{g} = \phi(1)) \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$ is a certified group description according to Definition 3, and $\overrightarrow{\mathbf{v}} \coloneqq (\mathbf{g}^{v_{\mathsf{S},1}}, \ldots, \mathbf{g}^{v_{\mathsf{S},n_{\mathsf{S}}}}, e(\mathbf{g},\mathbf{g})^{v_{\mathsf{T},1}}, \ldots, e(\mathbf{g},\mathbf{g})^{v_{\mathsf{T},n_{\mathsf{T}}}}) \in \mathbb{G}^{n_{\mathsf{S}}} \times \mathbb{G}_T^{n_{\mathsf{T}}}$.*

Further, we require for a rational VUF of evaluation degree d that for each $x \in \mathcal{X}$ there are coprime polynomials $\sigma_x, \rho_x \in \mathbb{Z}_p[V_1, \ldots, V_n]$ of total degree at most d s.t. we have for all vk, all π and all $\mathbf{y} \in \mathbb{G}_T$

$$\mathsf{Verify}_{\mathsf{vuf}}(\mathsf{vk}, x, \mathbf{y}, \pi) = 1 \implies \rho_x(v_1, \ldots, v_n) \neq 0 \text{ and } \mathbf{y} = e(\mathbf{g},\mathbf{g})^{\frac{\sigma_x(v_1,\ldots,v_n)}{\rho_x(v_1,\ldots,v_n)}} \quad (6)$$

where $(v_1, \ldots, v_n) = (v_{\mathsf{S},1}, \ldots, v_{\mathsf{S},n_{\mathsf{S}}}, v_{\mathsf{T},1}, \ldots, v_{\mathsf{T},n_{\mathsf{T}}})$ are the exponents of vk.

We require that – given x and a description of vuf – one can efficiently compute descriptions of σ_x and ρ_x, e.g. as coefficient vectors.

Definition 15 (Rational Univariate VUFs). *Let $d, n, d_f \in \mathsf{poly}(\lambda)$ and let $f_1, \ldots, f_n : \mathbb{Z}_p \to \mathbb{Z}_p$ be n efficiently computable polynomials of degree at most d_f. Let $\mathsf{vuf} = (\mathsf{Gen}_{\mathsf{vuf}}, \mathsf{Eval}_{\mathsf{vuf}}, \mathsf{Verify}_{\mathsf{vuf}})$ be a rational VUF **evaluation degree** d with $n = n_{\mathsf{S}} + n_{\mathsf{T}}$ verification key elements as in Definition 14. We say vuf is a rational **univariate** VUF of **internal degree** d_f relative to $f_1, \ldots, f_n$, iff for all vk, all $x \in \mathcal{X}$, all π and all $\mathbf{y} \in \mathbb{G}_T$ a successful verification $\mathsf{Verify}_{\mathsf{vuf}}(\mathsf{vk}, x, \mathbf{y}, \pi) = 1$ implies the existence of an "effective secret key" s, i.e.,*

$$\exists s \in \mathbb{Z}_p \text{ s.t. } \overrightarrow{\mathbf{v}} = (\mathbf{g}^{f_1(s)}, \ldots, \mathbf{g}^{f_{n_{\mathsf{S}}}(s)}, e(\mathbf{g},\mathbf{g})^{f_{n_{\mathsf{S}}+1}(s)}, \ldots, e(\mathbf{g},\mathbf{g})^{f_n(s)}) \ , \quad (7)$$

thus $\mathbf{y} = e(\mathbf{g},\mathbf{g})^{\frac{\sigma_x(f_1(s),\ldots,f_n(s))}{\rho_x(f_1(s),\ldots,f_n(s))}} = \mathbf{g}^{\widetilde{\sigma}_x(s)/\widetilde{\rho}_x(s)}$ where σ_x and ρ_x are defined in Definition 14, and $\widetilde{\sigma}_x(s) = \sigma_x(f_1(s), \ldots, f_n(s))$ and $\widetilde{\rho}_x(s) = \rho_x(f_1(s), \ldots, f_n(s))$. Note that $\deg(\widetilde{\sigma}_x), \deg(\widetilde{\rho}_x) \leq d \cdot d_f$.

Remark 12. In particular, the popular VRF of Dodis & Yampolskiy [16] is a rational univariate VUF with $n = d = d_f = 1$ (if extended by a certified group description).

4.2 From Consecutively Verifiable Pairing-Based VUFs to Rational VUFs

We now turn to proving that the evaluation outputs of consecutively verifiable pairing-based VUFs can be expressed through rational functions in the exponents.

Lemma 2. *Let* $\mathsf{vuf} = (\mathsf{Gen}_{\mathsf{vuf}}, \mathsf{Eval}_{\mathsf{vuf}}, \mathsf{Verify}_{\mathsf{vuf}})$ *be a pairing-based consecutively verifiable VUF with proofs of size* κ *and a verification key of size* n*.*

Then, vuf *is a rational VUF of evaluation degree at most* $4^{\kappa+1}$ *over* n *variables.*

We refer the reader to the full version [12] for the proof.

5 Algebraic Attacks on Rational VUFs

In this section we prove that the unpredictability of rational univariate VUFs cannot be based algebraically on some non-interactive computational assumptions. To this end, for any algebraic reduction from the NICA to the unpredictability of the VUF, we give a meta-reduction that internally runs the reduction and supplies it with an adversary for the unpredictability of the VUF. This meta-reduction finds a non-zero, low-degree, univariate *target polynomial* that contains the reduction's *effective* secret key as a root. Because the target polynomial has low (polynomial) degree and is non-zero, the meta-reduction can simply factor it and test each of its polynomially many roots against the reduction's verification key. Using the previously obtained secret key the meta-reduction can predict the reduction's challenge image.

Theorem 1. *Let* p *be a superpolynomial group order. Let* NICA *be a non-interactive computational assumption of size* $q \in \mathsf{poly}(\lambda)$*. Let* $n, d, d_f \in \mathsf{poly}(\lambda)$ *and let* $f_1, \ldots, f_n \in \mathbb{Z}_p[S]$ *be some polynomials of degree at most* d_f*. Let* vuf *be a rational univariate VUF of evaluation degree* d *and internal degree* d_f *over* n *variables relative to the polynomials* $f_1, \ldots, f_n$*.*

If there exists an algebraic $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}}, r, Q, 1/(Q+1))$*-reduction* $\mathcal{B}$ *from* NICA *to the weak* Q*-selective unpredictability of* vuf *s.t.* $Q \geq q^2 + 1$ *and* $r \in \mathsf{poly}(\lambda)$*, then there exists an adversary* $\mathcal{M}$ *that* $(t_{\mathcal{M}}, \epsilon_{\mathcal{M}})$*-breaks* NICA *with* $\epsilon_{\mathcal{M}} \geq \epsilon_{\mathcal{B}} - 2^{-\lambda}$ *and* $t_{\mathcal{M}} \leq t_{\mathcal{B}} + \mathsf{poly}(\lambda)$*.*

We refer the reader to the full version [12] for the proof.

Remark 13. Indeed, Theorem 1 can be applied if the input space $\mathcal{X}$ is only of polynomial size for a suitable definition of weak selective unpredictiability. Here, one has to make sure that the challenge preimage is not contained in the Q many query preimages, otherwise the adversary could predict trivially.

Corollary 1. *If the reduction in Theorem 1 is efficient, then* NICA *is efficiently solvable. In other words,* $t_{\mathcal{B}}/\epsilon_{\mathcal{B}} \in \mathsf{poly}(\lambda) \implies t_{\mathcal{M}}/\epsilon_{\mathcal{M}} \in \mathsf{poly}(\lambda)$.

We move on to our next result.

Theorem 2. *Let* $p = p(\lambda)$ *be a superpolynomial group order. Let* NICA *be some univariate DLog-hard assumption according to Definition 7 with* $l_1, l_2, d_{\mathsf{NICA}} \in \mathsf{poly}(\lambda)$*, and polynomials* $r_1, \ldots, r_{l_1}, t_1, \ldots, t_{l_2} \in \mathbb{Z}_p[S]$ *of degree at most* d_{NICA}*. Let* $n, d, r \in \mathsf{poly}(\lambda)$*. Let* vrf *be a rational VRF of evaluation degree* d *with* n *verification key elements s.t.* $\forall x \in \mathcal{X} : \sigma_x(\overrightarrow{V}) = V_1$.[13]

If there exists an algebraic $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}}, r, 0, 1)$*-reduction* $\mathcal{B}$ *(that forwards its group description as part of the verification key) from* NICA *to the* 0*-selective pseudorandomness of* vrf*, then there exists an adversary* $\mathcal{M}$ *that* $(t_{\mathcal{M}}, \epsilon_{\mathcal{M}})$*-breaks* NICA *with* $\epsilon_{\mathcal{M}} \geq \epsilon_{\mathcal{B}} - 2^{-\lambda}$ *and* $t_{\mathcal{M}} \leq t_{\mathcal{B}} + \mathsf{poly}(l_2, d_{\mathsf{NICA}}, d, \log p, r) = t_{\mathcal{B}} + \mathsf{poly}(\lambda)$.

We refer the reader to the full version [12] for the proof.

6 Generic Attacks on Parametrized Rational VUFs

Finally, we show the impossibility of algebraic and generic black-box reductions of the hardness of Uber-assumptions to the security of parametrized rational VUFs. Rational VUFs can be seen as a strong generalization of the VUFs of Dodis & Yampolskiy [16].

Definition 16. *A VUF* $\mathsf{vuf} = (\mathsf{Gen}_{\mathsf{vuf}}, \mathsf{Eval}_{\mathsf{vuf}}, \mathsf{Verify}_{\mathsf{vuf}})$ *is called* ***parametrized rational*** *of* ***evaluation degree*** $d_{\mathsf{vuf}} = d_{\mathsf{vuf}}(\lambda)$*, if there are polynomials* $\sigma, \rho \in \mathbb{Z}_p[V_{S,1}, \ldots, V_{S,n_1}, V_{T,1}, \ldots, V_{T,n_2}, X]$ *of total degree* d_{vuf} *s.t. the following things hold:*

1. *The set of possible inputs of* vuf *is* $\mathcal{X} = \mathbb{Z}_p$.
2. *For each generator* $\mathbf{h} \in \mathbb{G}$ *and each tuple* $(\mathsf{vk}, x, \mathbf{y}, \pi)$ *accepted by* $\mathsf{Verify}_{\mathsf{vuf}}$ *we have*

$$\rho(\overrightarrow{v_S}, \overrightarrow{v_T}, x) \neq 0 \quad \text{and} \quad \mathbf{y} = \mathbf{g}_T^{\sigma(\overrightarrow{v_S}, \overrightarrow{v_T}, x)/\rho(\overrightarrow{v_S}, \overrightarrow{v_T}, x)}.$$

where $\overrightarrow{v_S}$ *resp.* $\overrightarrow{v_T}$ *denote the exponents of the elements* $\mathsf{vk}_{S,1}, \ldots, \mathsf{vk}_{S,n_1}$ *resp.* $\mathsf{vk}_{T,1}, \ldots, \mathsf{vk}_{T,n_2}$ *relative to the basis* $\mathbf{h}$ *resp.* $e(\mathbf{h}, \mathbf{h})$.

We will now introduce our notion of Uber-assumptions, which is a generalization of the notion of Boyen [10].

Definition 17 (Computational Uber-Assumptions). *We call a non-interactive computational assumption* NICA *an* ***Uber-assumption*** *if there is a polynomial bound* $t = t(\lambda)$ *and a set of sparse polynomials* $f_{A_1}, \ldots, f_{A_{q_1}}, f_{B_1}, \ldots, f_{B_{q_2}} \in \mathbb{Z}_p[Z_1, \ldots, Z_t]$ *that can be computed efficiently s.t. the distributions of challenge*

[13] Essentially, the first verification key element $\mathbf{h} := \mathbf{v}_1$ is the new generator relative to which the VRF is evaluated.

samples of NICA *is identical to the output of the following algorithm:*

1. **draw** *a generator* $\mathbf{g}$ *of* $\mathbb{G}$
2. **draw** $(z_1, \ldots, z_t) \xleftarrow{\$} \mathbb{Z}_p^t$
3. **set** $a_1 := f_{A_1}(z_1, \ldots, z_t), \ldots, a_{q_1} := f_{A_{q_1}}(z_1, \ldots, z_t)$
4. **set** $b_1 := f_{B_1}(z_1, \ldots, z_t), \ldots, b_{q_2} := f_{B_{q_2}}(z_1, \ldots, z_t)$
5. **return** $(\mathbf{g}, \mathbf{g}^{a_1}, \ldots, \mathbf{g}^{a_{q_1}}, e(\mathbf{g}, \mathbf{g})^{b_1}, \ldots, e(\mathbf{g}, \mathbf{g})^{b_{q_2}})$

Let $d_{\mathsf{NICA}} = \max\{\deg f_{A_1}, \ldots, \deg f_{A_{q_1}}, \deg f_{B_1}, \ldots, \deg f_{B_{q_2}}\}$. *We call* d_{NICA} *the* ***degree*** *of* NICA *and* $q = 1 + q_1 + q_2$ *the* ***size*** *of* NICA.

We can now state the formal version of Theorem 3.

Theorem 3. *Let* vuf *be a parametrized rational VUF of evaluation degree* $d_{\mathsf{vuf}} \in O(1)$. *Let* NICA *be an Uber-assumption of degree* $d_{\mathsf{NICA}} \in \mathsf{poly}(\lambda)$ *and of size* $q \leq \sqrt{\log\log(w)}$ *for some* $w \in \mathsf{poly}(\lambda)$.

If NICA *is hard and* $Q > 2 \cdot (1 + \log\log w) \cdot w^{2\log(d_{\mathsf{vuf}}+1)}$, *then there is no generic reduction that can transform an adversary for the weak* Q*-selective unpredictability of* vuf *to a* NICA *solver.*

A full and exhaustive proof of Theorem 3 is given in the full version of this paper [12, Section 6].

In a nutshell, the idea of the proof is to see that, since the reduction is algebraic and generic, the algebraic explanations of each group element give a ring morphism that maps representations of group elements to polynomials in the variables $Z_1, \ldots, Z_t$ of the Uber-Assumption NICA. For each $x \in \mathbb{Z}_p$ queried by the adversary, this ring morphism must be chosen in such a way by the reduction s.t. a system $\mathcal{S}_x$ of polynomial equalities is fulfilled. Since vuf is parametrized of constant degree, we have that $\mathcal{S}_x$ depends itself polynomially on x. Therefore, if $\mathcal{S}_x$ is satisfiable for too many $x \in \mathbb{Z}_p$ it must be satisfiable for each $x \in \mathbb{Z}_p$ and a solution for S_{x_0} can be computed by the meta-reduction by mere linear algebra. Therefore, the meta-reduction can predict the image to the challenge query x_0 on its own if it can ask for too many queries.

References

1. Abdalla, M., Catalano, D. & Fiore, D. *Verifiable Random Functions from Identity-Based Key Encapsulation* in *EUROCRYPT 2009* (ed Joux, A.) **5479** (Springer, Heidelberg, Apr. 2009), 554–571.
2. Au, M. H., Susilo, W. & Mu, Y. *Practical Compact E-Cash* in *ACISP 07* (eds Pieprzyk, J., Ghodosi, H. & Dawson, E.) **4586** (Springer, Heidelberg, July 2007), 431–445.
3. Bauer, B., Fuchsbauer, G. & Loss, J. *A Classification of Computational Assumptions in the Algebraic Group Model* in *CRYPTO 2020, Part II* (eds Micciancio, D. & Ristenpart, T.) **12171** (Springer, Heidelberg, Aug. 2020), 121–151.
4. Belenkiy, M., Chase, M., Kohlweiss, M. & Lysyanskaya, A. *Compact E-Cash and Simulatable VRFs Revisited* in *PAIRING 2009* (eds Shacham, H. & Waters, B.) **5671** (Springer, Heidelberg, Aug. 2009), 114–131.

5. Bitansky, N. *Verifiable Random Functions from Non-interactive Witness-Indistinguishable Proofs* in *TCC 2017, Part II* (eds Kalai, Y. & Reyzin, L.) **10678** (Springer, Heidelberg, Nov. 2017), 567–594.
6. Blum, M. & Micali, S. *How to Generate Cryptographically Strong Sequences of Pseudo Random Bits* in *23rd FOCS* (IEEE Computer Society Press, Nov. 1982), 112–117.
7. Boneh, D., Boyen, X. & Goh, E.-J. *Hierarchical Identity Based Encryption with Constant Size Ciphertext* in *EUROCRYPT 2005* (ed Cramer, R.) **3494** (Springer, Heidelberg, May 2005), 440–456.
8. Boneh, D., Montgomery, H. W. & Raghunathan, A. *Algebraic pseudorandom functions with improved efficiency from the augmented cascade* in *ACM CCS 2010* (eds Al-Shaer, E., Keromytis, A. D. & Shmatikov, V.) (ACM Press, Oct. 2010), 131–140.
9. Boneh, D. & Venkatesan, R. *Breaking RSA May Not Be Equivalent to Factoring* in *EUROCRYPT'98* (ed Nyberg, K.) **1403** (Springer, Heidelberg, 1998), 59–71.
10. Boyen, X. *The Uber-Assumption Family (Invited Talk)* in *PAIRING 2008* (eds Galbraith, S. D. & Paterson, K. G.) **5209** (Springer, Heidelberg, Sept. 2008), 39–56.
11. Brakerski, Z., Goldwasser, S., Rothblum, G. N. & Vaikuntanathan, V. *Weak Verifiable Random Functions* in *TCC 2009* (ed Reingold, O.) **5444** (Springer, Heidelberg, Mar. 2009), 558–576.
12. Brandt, N., Hofheinz, D., Kastner, J. & Ünal, A. *The Price of Verifiability: Lower Bounds for Verifiable Random Functions* Cryptology ePrint Archive, Paper 2022/762. `https://eprint.iacr.org/2022/762`. 2022.
13. Coron, J.-S. *On the Exact Security of Full Domain Hash* in *CRYPTO 2000* (ed Bellare, M.) **1880** (Springer, Heidelberg, Aug. 2000), 229–235.
14. Coron, J.-S. *Optimal Security Proofs for PSS and Other Signature Schemes* in *EUROCRYPT 2002* (ed Knudsen, L. R.) **2332** (Springer, Heidelberg, 2002), 272–287.
15. Dodis, Y. *Efficient Construction of (Distributed) Verifiable Random Functions* in *PKC 2003* (ed Desmedt, Y.) **2567** (Springer, Heidelberg, Jan. 2003), 1–17.
16. Dodis, Y. & Yampolskiy, A. *A Verifiable Random Function with Short Proofs and Keys* in *PKC 2005* (ed Vaudenay, S.) **3386** (Springer, Heidelberg, Jan. 2005), 416–431.
17. Fiore, D. & Schröder, D. *Uniqueness Is a Different Story: Impossibility of Verifiable Random Functions from Trapdoor Permutations* in *TCC 2012* (ed Cramer, R.) **7194** (Springer, Heidelberg, Mar. 2012), 636–653.
18. Fleischhacker, N., Jager, T. & Schröder, D. On Tight Security Proofs for Schnorr Signatures. *Journal of Cryptology* **32,** 566–599 (Apr. 2019).
19. Fuchsbauer, G., Kiltz, E. & Loss, J. *The Algebraic Group Model and its Applications* in *CRYPTO 2018, Part II* (eds Shacham, H. & Boldyreva, A.) **10992** (Springer, Heidelberg, Aug. 2018), 33–62.

20. Goldreich, O., Goldwasser, S. & Micali, S. *How to Construct Random Functions (Extended Abstract)* in *25th FOCS* (IEEE Computer Society Press, Oct. 1984), 464–479.
21. Goldreich, O. & Levin, L. A. *A Hard-Core Predicate for all One-Way Functions* in *21st ACM STOC* (ACM Press, May 1989), 25–32.
22. Goldwasser, S. & Ostrovsky, R. *Invariant Signatures and Non-Interactive Zero-Knowledge Proofs are Equivalent (Extended Abstract)* in *CRYPTO'92* (ed Brickell, E. F.) **740** (Springer, Heidelberg, Aug. 1993), 228–245.
23. Goyal, R., Hohenberger, S., Koppula, V. & Waters, B. *A Generic Approach to Constructing and Proving Verifiable Random Functions* in *TCC 2017, Part II* (eds Kalai, Y. & Reyzin, L.) **10678** (Springer, Heidelberg, Nov. 2017), 537–566.
24. Hofheinz, D. & Jager, T. *Verifiable Random Functions from Standard Assumptions* in *TCC 2016-A, Part I* (eds Kushilevitz, E. & Malkin, T.) **9562** (Springer, Heidelberg, Jan. 2016), 336–362.
25. Hohenberger, S. & Waters, B. *Constructing Verifiable Random Functions with Large Input Spaces* in *EUROCRYPT 2010* (ed Gilbert, H.) **6110** (Springer, Heidelberg, 2010), 656–672.
26. Jager, T. *Verifiable Random Functions from Weaker Assumptions* in *TCC 2015, Part II* (eds Dodis, Y. & Nielsen, J. B.) **9015** (Springer, Heidelberg, Mar. 2015), 121–143.
27. Jarecki, S. & Shmatikov, V. *Handcuffing Big Brother: an Abuse-Resilient Transaction Escrow Scheme* in *EUROCRYPT 2004* (eds Cachin, C. & Camenisch, J.) **3027** (Springer, Heidelberg, May 2004), 590–608.
28. Katsumata, S. *On the Untapped Potential of Encoding Predicates by Arithmetic Circuits and Their Applications* in *ASIACRYPT 2017, Part III* (eds Takagi, T. & Peyrin, T.) **10626** (Springer, Heidelberg, Dec. 2017), 95–125.
29. Katz, J., Zhang, C. & Zhou, H.-S. *An Analysis of the Algebraic Group Model* Cryptology ePrint Archive, Report 2022/210. `https://eprint.iacr.org/2022/210`. 2022.
30. Kohl, L. *Hunting and Gathering - Verifiable Random Functions from Standard Assumptions with Short Proofs* in *PKC 2019, Part II* (eds Lin, D. & Sako, K.) **11443** (Springer, Heidelberg, Apr. 2019), 408–437.
31. Kurosawa, K., Nojima, R. & Phong, L. T. *Relation between Verifiable Random Functions and Convertible Undeniable Signatures, and New Constructions* in *ACISP 12* (eds Susilo, W., Mu, Y. & Seberry, J.) **7372** (Springer, Heidelberg, July 2012), 235–246.
32. Liang, B., Li, H. & Chang, J. *Verifiable Random Functions from (Leveled) Multilinear Maps* in *CANS 15* (eds Reiter, M. & Naccache, D.) (Springer, Heidelberg, Dec. 2015), 129–143.
33. Liskov, M. *Updatable Zero-Knowledge Databases* in *ASIACRYPT 2005* (ed Roy, B. K.) **3788** (Springer, Heidelberg, Dec. 2005), 174–198.
34. Lysyanskaya, A. *Unique Signatures and Verifiable Random Functions from the DH-DDH Separation* in *CRYPTO 2002* (ed Yung, M.) **2442** (Springer, Heidelberg, Aug. 2002), 597–612.

35. Maurer, U. M. *Abstract Models of Computation in Cryptography (Invited Paper)* in *10th IMA International Conference on Cryptography and Coding* (ed Smart, N. P.) **3796** (Springer, Heidelberg, Dec. 2005), 1–12.
36. Micali, S., Rabin, M. O. & Vadhan, S. P. *Verifiable Random Functions* in *40th FOCS* (IEEE Computer Society Press, Oct. 1999), 120–130.
37. Micali, S. & Reyzin, L. *Soundness in the Public-Key Model* in *CRYPTO 2001* (ed Kilian, J.) **2139** (Springer, Heidelberg, Aug. 2001), 542–565.
38. Micali, S. & Rivest, R. L. *Micropayments Revisited* in *CT-RSA 2002* (ed Preneel, B.) **2271** (Springer, Heidelberg, Feb. 2002), 149–163.
39. Naor, M. & Reingold, O. *Number-theoretic Constructions of Efficient Pseudo-random Functions* in *38th FOCS* (IEEE Computer Society Press, Oct. 1997), 458–467.
40. Nechaev, V. I. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes* **55,** 165–172 (1994).
41. Niehues, D. *Verifiable Random Functions with Optimal Tightness* in *PKC 2021, Part II* (ed Garay, J.) **12711** (Springer, Heidelberg, May 2021), 61–91.
42. Paillier, P. & Vergnaud, D. *Discrete-Log-Based Signatures May Not Be Equivalent to Discrete Log* in *ASIACRYPT 2005* (ed Roy, B. K.) **3788** (Springer, Heidelberg, Dec. 2005), 1–20.
43. Rosie, R. *Adaptive-Secure VRFs with Shorter Keys from Static Assumptions* in *CANS 18* (eds Camenisch, J. & Papadimitratos, P.) **11124** (Springer, Heidelberg, 2018), 440–459.
44. Schwartz, J. T. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* **27,** 701–717 (Oct. 1980).
45. Shoup, V. *Lower Bounds for Discrete Logarithms and Related Problems* in *EUROCRYPT'97* (ed Fumy, W.) **1233** (Springer, Heidelberg, May 1997), 256–266.
46. Yamada, S. *Asymptotically Compact Adaptively Secure Lattice IBEs and Verifiable Random Functions via Generalized Partitioning Techniques* in *CRYPTO 2017, Part III* (eds Katz, J. & Shacham, H.) **10403** (Springer, Heidelberg, Aug. 2017), 161–193.
47. Yao, A. C.-C. *Theory and Applications of Trapdoor Functions (Extended Abstract)* in *23rd FOCS* (IEEE Computer Society Press, Nov. 1982), 80–91.
48. Zhandry, M. *To Label, or Not To Label (in Generic Groups)* Cryptology ePrint Archive, Report 2022/226. `https://eprint.iacr.org/2022/226`. 2022.