Verifiable Random Functions with Optimal Tightness[*]

David Niehues
Paderborn University, Paderborn, Germany, david.niehues@upb.de

February 26, 2021

Abstract

Verifiable random functions (VRFs), introduced by Micali, Rabin and Vadhan (FOCS'99), are the public-key equivalent of pseudorandom functions. A public verification key and proofs accompanying the output enable all parties to verify the correctness of the output. However, all known standard model VRFs have a reduction loss that is much worse than what one would expect from known optimal constructions of closely related primitives like unique signatures. We show that:

1. Every security proof for a VRF that relies on a non-interactive assumption has to lose a factor of Q, where Q is the number of adversarial queries. To that end, we extend the meta-reduction technique of Bader *et al.* (EUROCRYPT'16) to also cover VRFs.
2. This raises the question: Is this bound optimal? We answer this question in the affirmative by presenting the first VRF with a reduction from the non-interactive qDBDHI assumption to the security of VRF that achieves this optimal loss.

We thus paint a complete picture of the achievability of tight verifiable random functions: We show that a security loss of Q is unavoidable and present the first construction that achieves this bound.

1 Introduction

Verifiable Random Functions (VRFs), introduced by Micali, Rabin and Vadhan in [MRV99], can be thought of as the public key equivalent of pseudorandom functions (PRFs). That is, a secret key sk always comes together with a public verification key vk. The secret key sk allows the evaluation of the verifiable random function $F_{\mathsf{sk}}(X)$ on input X and obtain the pseudorandom output Y. In contrast to pseudorandom functions, however, a verifiable random function also produces a non-interactive proof of correctness π. Together with vk, the proof π allows everyone to verify that Y is the output of $F_{\mathsf{sk}}(X)$. We require two security properties from VRFs: *unique provability* and *pseudorandomness*. Unique provability means that for every verification key vk and every VRF input X, there is a *unique* Y for which a proof π exists such that the verification algorithm accepts. However, note that there might be multiple valid proofs π verifying the correctness of Y with respect to vk and X. Further, we (informally) say that a VRF is pseudorandom if there is no efficient adversary that can distinguish a VRF output without the accompanying proof from a uniformly random element of the range of the VRF. In addition to these properties, Hofheinz and Jager introduced the notion of *VRFs with all desired properties* [HJ16]. Namely, we say that a VRF possesses *all*

[*] This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre On-The-Fly Computing (GZ: SFB 901/3) under the project number 160364472.

desired properties if it fulfills all requirements above, has an exponentially sized domain, is secure even in presence of an adaptive the adversary is proven secure under a non-interactive complexity assumption. In this work, we only consider VRFs that have all desired properties.

Applications of VRFs. VRFs have found a wide range of applications in theory in practice. One of the most notable ones is the recent application of VRFs in *proof of stake* consensus mechanisms, like the ones used in the Algorand Blockchain [GHM$^+$17], the Cardano Blockchain [BGK$^+$18, DGKR18] and the DFINITY Blockchain [AMNR18]. Further applications are in *key transparency systems* like CONIKS [MBB$^+$15], where VRFs prevent the enumeration of all users that have keys in the system. Similarly, VRFs are used in the proposed *DNSSEC extension NSECv5* [VGP$^+$18], where they provably prevent zone enumeration attacks in the authenticated denial of existence mechanism of DNSSEC [GNP$^+$15]. Further classical applications are resettable zero-knowledge proofs [MR01], lottery systems [MR02], verifiable transaction-escrow systems [JS04], updatable zero-knowledge databases [Lis05] and E-Cash [ASM07, BCKL09]. The wide range of applications has led to currently ongoing efforts to standardize VRFs [GRPV20].

Tightness. Following the reductionist approach to security, we relate the difficulty of breaking the security of a cryptographic scheme to the difficulty of solving an underlying hard problem. Let λ be the security parameter and consider a reduction showing that any adversary that breaks the security of a cryptographic scheme in time $t(\lambda)$ with probability $\epsilon(\lambda)$ implies an algorithm that solves the underlying hard problem with probability $\epsilon'(\lambda)$ in time $t'(\lambda)$ with $t'(\lambda) \geq t(\lambda)$ and $\epsilon'(\lambda) \leq \epsilon(\lambda)$. We then say that the reduction *loses* a factor $\ell(\lambda)$ if $t'(\lambda))/\epsilon'(\lambda) \geq \ell(\lambda)t(\lambda)/\epsilon(\lambda)$ for all $\lambda \in \mathbb{N}$. We say that a reduction is *tight* if ℓ is a constant, *i.e.* if the quality of the reduction does not depend on the security parameter.

The loss of a reduction is of particular practical importance when deciding on the key sizes to use for cryptographic schemes. For simplicity, assume that we have a reduction with $\epsilon'(\lambda) = \epsilon(\lambda)$ and $t'(\lambda) = \ell(\lambda)t(\lambda)$ and let $t_{\mathsf{opt}}(\lambda)$ denote the time the fastest algorithm takes to solve an instance of the hardness assumption. Then, if we want to rule out the existence of an adversary that breaks the security of the scheme faster than t_{adv}, we have to choose the security parameter large enough such that $t_{\mathsf{opt}}(\lambda)/\ell(\lambda) \geq t_{\mathsf{adv}}$. Hence, if ℓ is large, then λ has to be rather large in order to guarantee that any adversary that breaks the security of the scheme has runtime at least t_{adv}. However, a large security parameter also implies large keys, which negatively affects the real-world efficiency of the scheme. On the positive side, this means that if we are able to construct a tight reduction, this allows us to use *small key sizes and guarantee security* against all adversaries with runtime at most t_{adv}. This approach to security is also known as concrete security and is more thoroughly discussed in [BR09a].

Impossibility of tight reductions. Unfortunately, we know that tight reductions can not exist for some primitives. Coron presented the first result of this kind in 2002 for unique signatures [Cor02], in which he showed that every security reduction for unique signatures loses at least a factor of $\approx Q$, where Q is the number of adaptive signature queries made by the forger. He achieved this result by introducing the *meta-reduction technique*. That is, one shows that a tight reduction can not exist by proving that any tight reduction would be able to solve the underling hard problem without the help of an adversary. Subsequently, the technique has been successfully used to prove the same lower bound for the loss of security reductions for efficiently re-randomizable signatures by Hofheinz *et al.* [HJK12] and later on to an even wider classes of primitives by Bader *et al.* [BJLS16]. Most recently the Coron's technique has been extended by further works. First, Morgan and Pass extended Coron's technique to also incorporate interactive complexity assumptions and reductions that execute several instances of an adversary in parallel. However, since the result applies to a wider class of reductions and complexity assumptions, the lower bound on the loss is only $\sqrt{Q}$ instead of Q. Then Morgan*et al.* applied the technique to MACs and PRFs [MPS20].

Even though VRFs are closely related to unique signatures, none of the lower bounds on the loss mentioned above applies to VRFs in general because the non-interactive proofs of VRFs do not need to be unique, nor do they need to be re-randomizable. For example, the VRF by Bitansky does not have unique proofs [Bit20]. Hence, in contrast to a remark in [MP18], a VRF does not immediately imply a unique signature, but only a signature with a unique component.

Circumventing tightness lower bounds. Despite all the lower bounds on the loss of reductions to the security of unique signatures, Guo *et al.* showed in [GCS$^+$17] that reductions circumventing the lower bounds are possible by making heavy use of the programmability of a random oracle. However, this technique is only applicable in the random oracle model and can not be adapted in the standard model to the best of our knowledge.

Moreover, the tightness lower bounds have also been circumvented in the standard model by making the signatures non-randomizable [AFLT12, BKKP15, CD96, HJ12, KW03, Sch11]. Kakvi and Kiltz even describe a tightly secure unique signature scheme by using a public key in the reduction that allows for non-unique signatures and is indistinguishable from an honestly generated public key [KK12].

Furthermore, for identity based encryption – a primitive that is closely related to VRFs [ACF14]– Wee and Chen [CW13] describe a scheme that can proven secure with a reduction whose loss depends only on the security parameter and not on the number of queries made by the adversary. In 2016, Boyen and Li then presented the first tightly secure construction in [BL16]. Similar to our approach in this work, they homomorphically evaluate a pseudorandom function in the reduction. However, they use it in order to apply the technique of Katz and Wang to construct tightly secure signatures by making the signatures non-re-randomizable [KW03].

However, the techniques above are not applicable to VRFs. Replacing the verification with a indistinguishable verification key that allows for non-unique signatures is not possible due to the strong uniqueness requirement. Moreover, our meta reduction makes no assumptions about the re-randomizability of the proof of correctness produced by a VRF evaluation. Hence, making the proofs of correct evaluation non-rerandomizable can not allow for tighter reductions. Thus, to the best of our knowledge the only avenues to achieve tighter reductions for VRFs would be either to use the random oracle model, to prove the security from an interactive assumption or to use a reductions that can run several instances of an adversary in parallel. However, for the latter two approaches, it seems unlikely to achieve a loss better than $\sqrt{Q}$ due to the lower bound by Morgan and Pass [MP18].

Our contributions. In this paper, we study the tightness of reductions from non-interactive complexity assumptions to the security of verifiable random functions.

1. We first extend the lower bound for the loss of re-randomizable signatures from Bader *et al.* [BJLS16] to verifiable *unpredictable* functions (VUFs), which differ from VRFs in that the output only has to be unpredictable instead of pseudorandom. Since this is a weaker requirement, the theorem for VUFs also implies the same bound for reductions to the security of VRFs. Concretely, we prove that any reduction from a non-interactive complexity assumption to the unpredictability of a VUF loses a factor of at least Q.

2. We present a VRF and a reduction from the non-interactive q-DBDHI assumption to the adaptive

pseudorandomness of the VRF that achieves this bound. The VRF is based on the VRF by Yamada [Yam17a, Yam17b].

1.1 Notation

We introduce some notation before giving a technical overview of our work. For this, let $a, b, c \in \mathbb{N}$ with $a \leq b \leq c$. We then let $[c] := \{1, \ldots, c\}$. Analogously, we let $[a, c] := \{a, \ldots, c\}$ and $[c \setminus b] := [c] \setminus \{b\}$. Also, for any finite set S, we denote drawing a uniformly random element y from S by $y \overset{\$}{\leftarrow} S$. Further, for a probabilistic algorithm $\mathcal{A}$ that uses k bits of randomness and takes some input x, we write $\mathcal{A}(x; \rho_{\mathcal{A}})$ for the execution of $\mathcal{A}$ on input x with fixed random bits $\rho_{\mathcal{A}} \in \{0, 1\}^k$. Analogously, we write $a \overset{\$}{\leftarrow} \mathcal{A}(x)$ for executing $\mathcal{A}$ on input x with uniformly random bits and assigning the result to a. Finally, we will view the time to execute the security experiment as part of the runtime of an adversary that is executed in the security experiment. We do so as to not worsen the runtime of a reduction by accounting it runtime for simulating the security experiment for the adversary.

1.2 Technical Overview

Before presenting our results, we give a short overview over our techniques below. We first describe how we prove the lower bound for the loss of VRFs and then describe our construction attaining this bound.

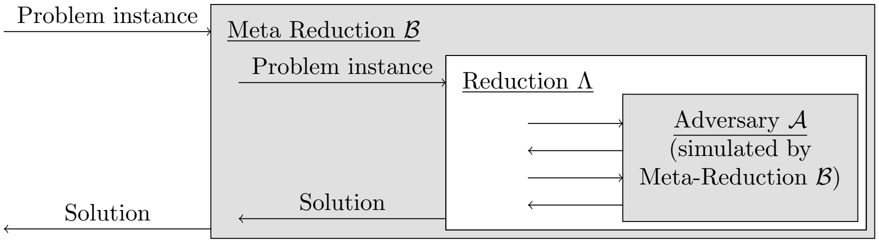

Figure 1: The meta-reduction technique of Coron [Cor02].

Bounding the tightness of VRFs. We first extend the meta-reduction of Bader *et al.* to VRFs and thus show that any reduction from a non-interactive complexity assumption to the security of a VRF necessarily loses a factor of at least Q, where Q is the number of queries made by the adversary. The results by Bader *et al.* do not cover VRFs and VUFs because their theorems only apply to *re-randomizable* signatures/relations[1]. However, VRFs and VUFs do not fall into this class of primitives because their non-interactive proofs are not necessarily re-randomizable. In order to explain how we extend their technique, we shortly revisit Coron's meta-reduction technique depicted in Figure 1. A *meta-reduction* can be thought of as a reduction against a reduction. That is, the meta-reduction $\mathcal{B}$ simulates a hypothetical adversary $\mathcal{A}$ for a reduction Λ. Since the meta-reduction is constructed to have a polynomial runtime and simulates the hypothetical adversary, it is actually the reduction Λ that solves the instance of the hardness assumption. This allows us to show that any reduction with a certain tightness is able to break the underlying hardness assumption without the help of any adversary and therefore contradicts the hardness assumption.

[1] Note that unique signatures are re-randomizable because, given a unique signature for a message, it is trivial to sample from all signatures for that message since there is only that one signature.

In their proof, Bader *et al.* use the re-randomizability/uniqueness of the signatures that Λ produces for $\mathcal{A}$ in order to solve the challenge when simulating $\mathcal{A}$. We extend their technique to VRF/VUFs by showing that it is sufficient if the part of the signature that the adversary has to provide for the challenge, in the case of VUFs the unpredictable value Y, is unique or re-randomizable.

For simplicity, we prove the theorem for VUFs: this automatically implies the same bound for VRFs because every VRF is also a VUF. Following Bader *et al.*, we consider a very weak security model in which the number of queries Q is fixed a priori. Further, the adversary is presented with Q uniformly random and pairwise distinct inputs $X_1, \ldots, X_Q$ and has to choose a challenge X^* from these. For all other inputs, the adversary is then given the VUF output and proof. Finally, the adversary has to output the VUF value for the challenge input and wins if the output is correct. We refer to this very weak security as *weak-selective unpredictability*. We describe a hypothetical adversary that breaks the adaptive pseudorandomness with certainty and then show that our meta-reduction can efficiently simulate this adversary for the reduction. Informally, on input a problem instance for a non-interactive complexity assumption, the meta-reduction Λ behaves as follows.

1. It passes on the problem instance to the reduction and lets it output a verification key vk and Q pairwise different VUF inputs $X_1, \ldots, X_Q$.

2. It then iterates over all $j \in [Q]$ and executes the second part of the reduction as if it chose j as the challenge and lets the reduction produce all pairs of VUF output and proof except for the j'th pair. It then verifies them and saves them if they are correct with respect to vk and the corresponding input.

3. Finally, it chooses $j^* \xleftarrow{\$} [Q]$ and passes on the correct VUF output for X_{j^*} to the reduction. We formally prove in Section 2 that the meta-reduction indeed has learned the correct VUF output for X_{j^*} from the reduction with probability at least $1/Q$.

4. When the reduction then outputs the solution to the underlying problem instance, the meta-reduction outputs this solution as well.

Overall, we can then show that the meta-reduction takes time at most $\mathcal{B} = Q \cdot t_\Lambda + Q(Q+1)t_{\mathsf{Vfy}}$ and has a success probability at least $\epsilon_\Lambda - 1/Q$, where t_Λ and ϵ_Λ are the runtime and the success probability of the reduction and t_{Vfy} is the time it takes to verify a VUF output. Now we can follow that Λ has a loss of at least $\ell = (\epsilon_N + 1/Q)^{-1}$, where ϵ_N is the largest probability any algorithm running in time $t_{\mathcal{B}}$ has in breaking the hardness assumption. Since the hardness assumption implies that ϵ_N is negligibly small, we have that $\ell \approx Q$.

While the meta-reduction above is only applicable to reductions that execute the adversary exactly once, our proof of the lower bound on the loss of VRFs in Section 2, like the one by like Bader *et al.*, also applies to reductions that can sequentially rewind the adversary.

On the difficulty of constructing tightly secure VRFs. As Table 1 shows, known security proofs for VRFs in the standard model are significantly more lossy than the lower bound Q. This raises the question:

Do verifiable random functions with a loss of Q exist?

In consequence, such a VRF would show that a loss of Q is indeed optimal.

We proceed by explaining why all previous constructions have a loss much worse than Q and then give an overview over our approach that achieves the optimal tightness. They all have in common that the reduction makes a guess in the very beginning and then has to abort and output a random bit depending

Schemes	**Security loss**
Hohenberger and Waters [HW10]	$\mathcal{O}(\lambda Q/\epsilon)$
Boneh *et al.* Sec. 7 in [BMR10]	$(Q\lambda)^{\tau(\epsilon)}$
Jager [Jag15]	$\mathcal{O}(Q^{\nu}/\epsilon^{\nu+1})$
Hofheinz and Jager [HJ16]	$\mathcal{O}(\lambda\log(\lambda)Q^{2/c}/\epsilon^{3})$
Yamada Sec 6.1 in [Yam17b]	$\mathcal{O}(Q^{\nu}/\epsilon^{\nu+1})$
Yamada Sec. 6.2 in [Yam17b]	$\mathcal{O}(Q^{\nu}/\epsilon^{\nu+1})$
Yamada App. C in [Yam17a]	$\mathcal{O}(\lambda^{2}Q/\epsilon^{2})$
Katsumata Sec. 5.1 in [Kat17]	$\mathcal{O}(Q^{\nu}/\epsilon^{\nu+1})$
Kastumata Sec. 5.3 in [Kat17]	$\mathcal{O}(Q^{\nu}/\epsilon^{\nu+1})$
Rosie [Ros18]	$\mathcal{O}(\lambda\log(\lambda)Q^{2/c}/\epsilon^{3})$
Kohl [Koh19]	$\mathcal{O}(\lvert\pi\rvert\log(\lambda)Q^{2/\nu}/\epsilon^{3})$
Kohl [Koh19]	$\mathcal{O}(\lvert\pi\rvert\log(\lambda)Q^{2+2/\nu}/\epsilon^{3})$
Jager and Niehues [JN19]	$\mathcal{O}(t^{3}/\epsilon^{2})$
Jager *et al.* [JKN21]	$\mathcal{O}(t^{3}/\epsilon^{2})$
Section 4	$\mathcal{O}(Q)$

Table 1: We compare the loss of previous VRFs with all desired properties. For the variables, let $|\pi|$ denotes the size of the proofs of the VRF and ϵ, t and Q the advantage, runtime and number of queries made by the adversary the reduction is run against. Further, there are three values that depend on the error correcting code used in the construction: the function $\tau(\epsilon) > 1$ and the constants $\nu > 1$ and $c \leq 1/2$. Note that the full version [BMR] of [BMR10] has been updated with the bound stated above.

on the queries and the challenge of the adversary. Let $\mathsf{succ\text{-}red}$ be the event that the reduction solves the underlying hardness assumption and let abort be the event that the reduction aborts and outputs a random bit. For a clear exposition, we assume that the reduction always succeeds when it does not abort and the adversary succeeds. We then have that

$$\begin{aligned}\Pr[\mathsf{succ\text{-}red}] &= \Pr[\mathsf{succ\text{-}red} \wedge \mathsf{abort}] + \Pr[\mathsf{succ\text{-}red} \wedge \neg\mathsf{abort}] \\ &= \frac{1}{2}(1 - \Pr[\neg\mathsf{abort}]) + \Pr[\mathsf{succ\text{-}red} \wedge \neg\mathsf{abort}] \\ &= \frac{1}{2} + \Pr[\mathsf{succ\text{-}red} \wedge \neg\mathsf{abort}] - \frac{\Pr[\neg\mathsf{abort}]}{2}.\end{aligned}$$

This shows that, in contrast to computational security experiments/hardness assumptions, where a lower bound would suffice, we need upper and lower bounds on $\Pr[\mathsf{abort}]$ that are close to each other in order prove the security of a VRF. Waters used the *artificial abort technique* to prove close lower and upper bounds on $\Pr[\neg\mathsf{abort}]$ [Wat05]. That is, the reduction estimates the probability of aborting over all possible choices it can make in the very beginning for the sequence of queries made by the adversary and then aborts with a probability that ensures that the reduction always aborts with almost the same probability. However, the estimation step in the reduction is computationally expensive. Bellare and Ristenpart addressed this issue with a more thorough analysis and by making $\Pr[\neg\mathsf{abort}]$ slightly smaller [BR09b]. Jager then applied Bellare's and Ristenpart's technique to admissible hash functions (AHFs) and introduced *balanced admissible hash functions* [Jag15]. But in conclusion, none of the techniques known so far achieves the optimal loss of Q.

A reduction with optimal tightness. We next answer the question stated above in the affirmative by presenting a VRF with a reduction that only loses a factor of Q. To do so, we have to address the issue raised above: that the success probability for the partitioning argument depends on the sequence of queries made by the adversary. We achieve this by passing every query and the challenge of the adversary trough a pseudorandom function (PRF). Further, we utilize a property of the VRF Yamada introduced in [Yam17a, Appendix C]. This VRF allows the reduction to homomorphically embed an arbitrary NAND circuit of polynomial size and logarithmic depth in the VRF. The idea here is that the reduction can embed an arbitrary NAND-circuit in the VRF such that it can answer all queries by the adversary for which the circuit evaluates to 0 and can extract a solution to the underlying hard problem whenever the circuit evaluates to 1. In particular, the homomorphic evaluation hides selected parts of the circuit inputs, all internal states of the circuit and the output of the circuit from the adversary.

We use these properties to homomorphically evaluate a PRF. Since the adversary does not learn any internal states or outputs of the PRF, we thus have that the outputs of the PRF are distributed as if they were the outputs of a random function. In particular, we then have that the outputs of the PRF are distributed uniformly and independent from each other. We show in Section 3 that it then suffices for the reduction to guess $\lceil \log(Q) \rceil + 1$ bits of the PRF output of the challenge. Then the probability that the following two events both occur is at least $1/8Q$:

1. The PRF output of the challenge matches the guess.

2. The guess does not match the PRF output for any of the adversary's queries.

Further, viewing the PRF outputs as the output of a truly random function, the probability for the reduction to succeeds is independent from the probability of the adversary breaking the security of the VRF. Ultimately, this yields a VRF, which has a loss of Q plus the loss of the PRF.

```
$G^{\mathcal{VRF}}_{(\mathcal{A}_1,\mathcal{A}_2)}(\lambda)$
$(\mathsf{vk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda); \rho_{\mathcal{A}} \xleftarrow{\$} \{0,1\}^\lambda$
$(X^*,\mathsf{st}) \xleftarrow{\$} \mathcal{A}_1^{\mathsf{Eval}(\mathsf{sk},\cdot)}(\mathsf{vk};\rho_{\mathcal{A}})$
$Y_0 := \mathsf{Eval}(\mathsf{sk},X^*)$
$Y_1 \xleftarrow{\$} \mathcal{Y}$
$b \xleftarrow{\$} \{0,1\}$
$b' := \mathcal{A}_2^{\mathsf{Eval}(\mathsf{sk},\cdot)}(Y_b,\mathsf{st})$
return $b == b'$
```

Figure 2: The security experiment specifying pseudorandomness of verifiable random functions.

2 Impossibility of VRFs and VRFs with tight reductions

In this section, we prove that any reduction from a non-interactive complexity assumption to the security of a VUF or VRF unavoidably loses a factor of Q. To do so, we first formally introduce VUFs and VRFs and their accompanying security notions. We then introduce a very weak security notion for VUFs and prove that even for this notion, every reduction form a non-interactive complexity assumption to it necessarily loses a factor of Q.

2.1 Syntax of Verifiable Random Functions (VRFs) and Verifiable Unpredictable Functions (VUFs).

Formally, a VRF or VUF consists of algorithms $(\mathsf{Gen},\mathsf{Eval},\mathsf{Vfy})$ with the following syntax.

- $(\mathsf{vk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ takes as input the security parameter λ and outputs a key pair $(\mathsf{vk},\mathsf{sk})$. We say that sk is the *secret key* and vk is the *verification key*.
- $(Y,\pi) \xleftarrow{\$} \mathsf{Eval}(\mathsf{sk},X)$ takes as input a secret key sk and $X \in \{0,1\}^\lambda$, and outputs a function value $Y \in \mathcal{Y}$, where $\mathcal{Y}$ is a finite set, and a proof π. We write $V_{\mathsf{sk}}(X)$ to denote the function value Y computed by Eval on input (sk,X).
- $\mathsf{Vfy}(\mathsf{vk},X,Y,\pi) \in \{0,1\}$ takes as input a verification key vk, $X \in \{0,1\}^\lambda$, $Y \in \mathcal{Y}$, and proof π, and outputs a bit.

Note that VRFs and VUFs share a common syntax. The only difference is in the achieved security properties. We first define security for VRFs and then describe how the definition has to be adapted for VUFs.

Definition 1. *$\mathcal{VRF} = (\mathsf{Gen},\mathsf{Eval},\mathsf{Vfy})$ is a verifiable random function (VRF) if all of the following hold.*

Correctness. For all $(\mathsf{vk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$, $X \in \{0,1\}^\lambda$ and $(Y,\pi) \xleftarrow{\$} \mathsf{Eval}(\mathsf{sk},X)$ it must holds that $\mathsf{Vfy}(\mathsf{vk}, X,Y,\pi) = 1$. Further, the algorithms Gen, Eval, Vfy are polynomial-time.

Unique provability. For all $\mathsf{vk} \in \{0,1\}^*$ and all $X \in \{0,1\}^\lambda$, there does not *exist* any $Y_0,\pi_0,Y_1,\pi_1 \in \{0,1\}^*$ such that $Y_0 \neq Y_1$ and it holds that $\mathsf{Vfy}(\mathsf{vk},X,Y_0,\pi_0) = \mathsf{Vfy}(\mathsf{vk},X,Y_1,\pi_1) = 1$.

```
$\mathsf{weak\text{-}selective\text{-}Unpredictability}^{Q,\mathcal{VUF}}_{(\mathcal{A}_1,\mathcal{A}_2)}(\lambda)$
$(\mathsf{vk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda);\ \rho_\mathcal{A} \xleftarrow{\$} \{0,1\}^\lambda$
$(X_1,\ldots,X_Q) \xleftarrow{\$} \{0,1\}^\lambda$ s.t. $X_i \neq X_j$ for all $i \neq j$
$(Y_i,\pi_i) \xleftarrow{\$} \mathsf{Eval}(\mathsf{sk},X_i)$
$(j,\mathsf{st}) \xleftarrow{\$} \mathcal{A}_1(\mathsf{vk},(X_i)_{i\in[Q];\rho_\mathcal{A}})$
$Y^* \xleftarrow{\$} \mathcal{A}_2((Y_i,\pi_i,\mathsf{st})_{i\in[Q\setminus j]})$
return $Y^* == Y_j$
```

Figure 3: The security experiment specifying weak selective pseudorandomness.

Pseudorandomness. Consider an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with access (via oracle queries) to $\mathsf{Eval}(\mathsf{sk},\cdot)$ in the pseudorandomness game depicted in Figure 2. Let $\mathcal{Q} = (X_1,\ldots,X_Q)$ be the oracle queries made by $\mathcal{A}_1$ and $\mathcal{A}_2$, then we say that $\mathcal{A}$ is *legitimate* if there is no $\rho_\mathcal{A} \in \{0,1\}^\lambda$ such that there exists $i \in [Q]$ with $X_i = X^*$, where X_i is the i'th query to Eval made by $\mathcal{A}$. We define the advantage of $\mathcal{A}$ in breaking the pseudorandomness of $\mathcal{VRF}$ as

$$\mathsf{Adv}^{\mathcal{VRF}}_{\mathcal{A}}(\lambda) := \left|\Pr\left[G^{\mathcal{VRF}}_{(\mathcal{A}_1,\mathcal{A}_2)}(\lambda) = 1\right] - 1/2\right|.$$

We require the same security properties from VUFs as the properties we require from VRFs in Definition 1, with the exception that we require the weaker property of *unpredictability* instead of pseudorandomness from VUFs. This property can be formalized just like pseudorandomness just that the adversary has to output the correct Y^* instead of distinguishing it from a random element as depicted in Figure 2. We do not give a formal definition since it is very similar to VRFs and we use the notion of weak select unpredictability, which is defined in Section 2.2, in our proof.

2.2 Lower tightness bounds for VUFs

We begin by introducing the very weak security notion of *weak-selective unpredictability*. In this security model, all queries and the challenge are uniformly random and pairwise different. We formally define it as follows.

Definition 2. Let $\mathcal{VUF} = (\mathsf{Gen},\mathsf{Eval},\mathsf{Vfy})$ be a verifiable unpredictable function and let $t : \mathbb{N} \to \mathbb{N}, \epsilon : \mathbb{N} \to [0,1]$. For an adversary $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2)$, we say that $\mathcal{A}$ (t,Q,ϵ)-breaks the weak selective pseudorandomness of $\mathcal{VUF}$ if $\mathcal{A}$ runs in time t and

$$\mathsf{Adv}^{\mathcal{VUF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda) := \Pr\left[\mathsf{weak\text{-}selective\text{-}Unpredictability}^{Q,\mathcal{VUF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda) = 1\right] = \epsilon(\lambda)$$

where $\mathsf{weak\text{-}selective\text{-}Unpredictability}^{Q,\mathcal{VUF}}_{(\mathcal{A}_1,\mathcal{A}_2)}(\lambda)$ is the security experiment depicted in Figure 3.

Note that any verifiable random function fulfilling the requirements of Definition 1 has also weak-selective unpredictability. Hence, ruling out a tight reduction from weak selective unpredictability to a class of hardness assumptions, also rules out tight reductions from pseudorandomness to that class of hardness assumptions. We thus prove a lower bound on the loss of any reduction from any non-interactive complexity

```
$\mathsf{NICA}^N_{\mathcal{A}}(\lambda)$
$(c, w) \xleftarrow{\$} \mathsf{T}(1^\lambda); \rho_{\mathcal{A}} \xleftarrow{\$} \{0,1\}^\lambda$
$s \xleftarrow{\$} \mathcal{A}(c; \rho_{\mathcal{A}})$
return $\mathsf{V}(c, w, s)$
```

Figure 4: The generic security experiment for a non-interactive complexity assumption $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ between the challenger and an adversary $\mathcal{A}$.

assumption to the weak selective unpredictability of a VUF, where the reduction my sequentially repeat the execution of the adversary.

Following [AGO11, BJLS16], we define a non-interactive complexity assumption as a triple $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ of Turing machines (TMs). While the TM T generates a problem instance and V verifies the correctness of a solution, the TM U represents a trivial adversary to compare an actual adversary against. For example, a trivial adversary against the DDH assumption would just output random bit as its guess. We formally define non-interactive complexity assumptions as follows.

Definition 3. A *non-interactive complexity assumption* $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ consist of three Turing machines. The instance generation machine $(c, w) \xleftarrow{\$} \mathsf{T}(1^\lambda)$ takes the security parameter as input and outputs a problem instance c and a witness w. U is a probabilistic polynomial-time Turing machine, which takes c as input and outputs a candidate solution s. The verification Turing machine V takes as input (c, w) and a candidate solution s. If $\mathsf{V}(c, w, s) = 1$, then we say that s is a correct solution to the challenge c.

Definition 4. Let $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ be a non-interactive complexity assumption and let NICA be the security experiment depicted in Figure 4. For functions $t : \mathbb{N} \to \mathbb{N}, \epsilon : \mathbb{N} \to [0, 1]$ and a probabilistic Turing machine $\mathcal{B}$ running in time $t(\lambda)$, we say that $\mathcal{B}$ (t, ϵ)-breaks N if

$$\left|\Pr\left[\mathsf{NICA}^N_{\mathcal{B}}(\lambda) = 1\right] - \Pr\left[\mathsf{NICA}^N_{\mathsf{U}}(\lambda) = 1\right]\right| \geq \epsilon(\lambda),$$

where the probabilities are taken over the randomness consumed by T and the random choices of ρ_{U} and $\rho_{\mathcal{B}}$ in the security experiments $\mathsf{NICA}^n_{\mathcal{B}}(\lambda)$ and $\mathsf{NICA}^n_{\mathsf{U}}(\lambda)$.

Bader *et al.* prove lower bounds for simple reductions as well as for reductions that can sequentially rewind the adversary [BJLS16]. Since the latter class of reduction include the former class, we directly prove the lower bound on the loss for the larger class of reductions. Following Bader *et al.*, we view a reduction that sequentially rewinds an adversary up to $r \in \mathbb{N}$ times as a $3r + 2$-tuple of Turing machines. That is, one TM that initializes the reduction, one to produce a solution in the end and three for each execution of the adversary. For an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the weak selective unpredictability of a verifiable unpredictable function $\mathcal{VUF}$, we let $r\text{-}\Lambda^{\mathcal{A}}$ be the Turing machine depicted in Figure 5.

Definition 5 (Def. 6 in [BJLS16])**.** For a verifiable unpredictable function $\mathcal{VUF}$, we say that a Turing machine $r\text{-}\Lambda = (\Lambda_1, (\Lambda_{\ell,1}, \Lambda_{\ell,2}, \Lambda_{\ell,3})_{\ell \in [r]}, \Lambda_3)$ is an r-simple $(t_\Lambda, Q, \epsilon_\Lambda, \epsilon_{\mathcal{A}})$-reduction from breaking the non-interactive complexity assumption $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ to breaking the weak selective unpredictability of $\mathcal{VUF}$ if for any TM $\mathcal{A}$ that $(t_{\mathcal{A}}, Q, \epsilon_{\mathcal{A}})$-breaks the weak selective unpredictability of $\mathcal{VUF}$, TM $r\text{-}\Lambda^{\mathcal{A}}$ as defined in Figure 5 $(t_\Lambda + rt_{\mathcal{A}}, \epsilon_{\mathcal{A}})$ breaks N.

```
$r\text{-}\Lambda^{\mathcal{A}}(c, \rho_\Lambda)$
$\mathsf{st}_{\Lambda_{1,1}} \xleftarrow{\$} \Lambda_1(c; \rho_0)$
For $1 \le \ell \le r$ do:
    $(\mathsf{vk}^\ell, (X_i^\ell)_{i \in [Q]}, \rho_{\mathcal{A}}, \mathsf{st}_{\Lambda_{\ell,2}}) \xleftarrow{\$} \Lambda_{\ell,1}(\mathsf{st}_{\Lambda,1})$
    $(j^{*\ell}, \mathsf{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_1(\mathsf{vk}^\ell, (X_i^\ell)_{i \in [Q]}; \rho_{\mathcal{A}})$
    $((Y_i^\ell, \pi_i^\ell)_{i \in [Q \backslash j^{*\ell}]}, \mathsf{st}_{\Lambda_{\ell,3}}) \xleftarrow{\$} \Lambda_{\ell,2}(j^{*\ell}, \mathsf{st}_{\Lambda_{\ell,2}})$
    $Y^\ell_{j^{*\ell}} \xleftarrow{\$} \mathcal{A}_2((Y_i^\ell, \pi_i^\ell)_{i \in [Q \backslash j^{*\ell}]}, \mathsf{st}_{\mathcal{A}})$
    $\mathsf{st}_{\Lambda_{\ell+1,1}} \xleftarrow{\$} \Lambda_{\ell,3}\left(Y^\ell_{j^{*\ell}}, j^{*\ell}, \mathsf{st}_{\Lambda_{\ell,3}}\right)$
$s \xleftarrow{\$} \Lambda_3(\mathsf{st}_{\Lambda r+1,1})$
```

Figure 5: Description of the Turing $r\text{-}\Lambda^{\mathcal{A}}$ machine built from an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the weak selective unpredictability of a verifiable unpredictable function and a reduction $(\Lambda_1, (\Lambda_{\ell,1}, \Lambda_{\ell,2}, \Lambda_{\ell,3})_{\ell \in [r]}, \Lambda_3)$.

Furthermore, we define the loss of a reduction as the factor that $(t_\Lambda(\lambda) + rt_{\mathcal{A}}(\lambda))/\epsilon_\Lambda(\lambda)$ is larger than $t_{\mathcal{A}}(\lambda)/\epsilon_{\mathcal{A}}(\lambda)$. We formalize this in the following definition.

Definition 6. For a verifiable unpredictable function $\mathcal{VUF}$, a non-interactive complexity assumption N, a function $\ell : \mathbb{N} \to \mathbb{N}$ and a reduction Λ, we say that Λ loses ℓ, if there exists an adversary $\mathcal{A}$ that $(t_{\mathcal{A}}, Q, \epsilon_{\mathcal{A}})$ breaks the weak selective unpredictability of $\mathcal{VUF}$ such that $\Lambda^{\mathcal{A}}$ $(t_\Lambda + r \cdot t_{\mathcal{A}}, \epsilon_{\mathcal{A}})$-breaks N, where

$$\frac{t_\Lambda(\lambda) + rt_{\mathcal{A}}(\lambda)}{\epsilon_\Lambda(\lambda)} \ge \ell(\lambda) \cdot \frac{t_{\mathcal{A}}(\lambda)}{\epsilon_{\mathcal{A}}(\lambda)}.$$

After introducing the needed notations and notions, we can now state our theorem regarding the loss of VRFs and VUFs.

Theorem 1. *Let $N = (\mathsf{T}, \mathsf{V}, \mathsf{U})$ be a non-interactive complexity assumption, $Q, r \in \mathsf{poly}(\lambda)$ and let $\mathcal{VUF}$ be a verifiable unpredictable function. Then for any r-simple $(t_\Lambda, Q, \epsilon_\Lambda, 1)$-reduction Λ from breaking N to breaking the weak selective unpredictability of $\mathcal{VUF}$ there exists a TM $\mathcal{B}$ that $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks N, where*

$$\begin{aligned} t_{\mathcal{B}} &\le r \cdot Q \cdot t_{\mathcal{A}} + r \cdot Q \cdot (Q-1) \cdot t_{\mathsf{Vfy}} \\ \epsilon_{\mathcal{B}} &\ge \epsilon_\Lambda - \frac{r}{Q}. \end{aligned}$$

Here, t_{Vfy} is time needed to run the algorithm Vfy *of $\mathcal{VUF}$.*

Note that the theorem also applies to adversaries with $\epsilon_{\mathcal{A}} < 1$, as we discuss after the proof of Theorem 1. However, before proving Theorem 1, we show that it implies that every r-simple reduction Λ from a non-interactive complexity assumption N has at least a loss of $\approx Q$. For $t_N := t_{\mathcal{B}} = r \cdot Q \cdot t_\Lambda + r \cdot Q \cdot (Q-1) \cdot t_{\mathsf{Vfy}}$, let ϵ_N be the largest probability such that there exists an algorithm that (t_N, ϵ_N)-breaks N. We then have that $\epsilon_N \ge \epsilon_{\mathcal{B}}$ and by Theorem 1, we have that $\epsilon_\Lambda \le \epsilon_{\mathcal{B}} + r/Q \le \epsilon_N + r/Q$. We can then conclude that

$$\frac{t_\Lambda + r \cdot t_{\mathcal{A}}}{\epsilon_\Lambda} \ge \frac{r \cdot t_{\mathcal{A}}}{\epsilon_N + r/Q} = (\epsilon_N + r/Q)^{-1} \cdot r \cdot \frac{t_{\mathcal{A}}}{1} = (\epsilon_N + r/Q)^{-1} \cdot r \cdot \frac{t_{\mathcal{A}}}{\epsilon_{\mathcal{A}}}.$$

This means that Λ loses at least a factor of $\ell = r/(\epsilon_N + r/Q)$. Further, if ϵ_N is very small, which it is supposed to be for a good complexity assumption, then $\ell \approx Q$.

PROOF. Our proof is structured like the proofs in [BJLS16, HJK12, LW14] and thus first describes a hypothetical adversary that breaks the weak selective unpredictability of $\mathcal{VUF}$ with certainty and then describes a meta reduction that perfectly and efficiently simulates this adversary towards Λ.

The hypothetical adversary $\mathcal{A}$. The hypothetical adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consists of the following two procedures.

$\mathcal{A}_1(\mathsf{vk}, (X_i)_{i\in[Q]}; \rho_{\mathcal{A}})$ samples $j \xleftarrow{\$} [Q]$ and outputs (j, st) with the state $\mathsf{st} = (\mathsf{vk}, (X_i)_{i\in[Q]}, j)$.

$\mathcal{A}_2((Y_i, \pi_i)_{i\in[Q\setminus j]}, \mathsf{st})$ first parses the state st as $(\mathsf{vk}, (X_i)_{i\in[Q]}, j)$ and checks whether $\mathsf{Vfy}(\mathsf{vk}, X_i, Y_i, \pi_i) = 1$ for all $i \in [Q \setminus j]$. If there is i^* such that $\mathsf{Vfy}(\mathsf{vk}, X_i, Y_i, \pi_i) = 0$, it aborts with result $\perp$. Otherwise it computes $Y^* \in \mathcal{Y}$ such that there exists $\pi \in \{0,1\}^*$ with $\mathsf{Vfy}(\mathsf{vk}, X_j, Y^*, \pi) = 1$. The existence of such a Y^* is guaranteed by the correctness of $\mathcal{VUF}$.

Observe that $\mathcal{A}$ breaks the weak selective unpredictability of $\mathcal{VUF}$ with certainty because a correct VUF produces only valid pairs of outputs and proofs, but $\mathcal{A}_2$ may not be efficiently computable. However, we show that $\mathcal{B}$ can efficiently simulate $\mathcal{A}$ nonetheless.

The meta-reduction $\mathcal{B}$. We now describe the meta-reduction $\mathcal{B}$ that simulates $\mathcal{A}$ r times for the reduction $\Lambda = (\Lambda_1, (\Lambda_{\ell,1}, \Lambda_{\ell,2}, \Lambda_{\ell,3})_{\ell\in[r]}, \Lambda_3)$. $\mathcal{B}$'s goal in this is to break N and is therefore called on input c, where $(c, w) \xleftarrow{\$} \mathsf{T}(1^\lambda)$.

i. $\mathcal{B}$ receives c as input. It samples randomness $\rho_\Lambda \xleftarrow{\$} \{0,1\}^\lambda$ and executes $\mathsf{st}_{\Lambda_{1,1}} = \Lambda_1(c, \rho_\Lambda)$. If Λ_1 does not output $\mathsf{st}_{\Lambda_{1,1}}$, then $\mathcal{B}$ aborts and outputs $\perp$. Since the randomness of Λ_1 is fixed, we view all subroutines of Λ as deterministic. Note that Λ_1 can pass on random coins to the other subroutines via $\mathsf{st}_{\Lambda_{1,1}}$.

ii. Next, $\mathcal{B}$ sequentially simulates $\mathcal{A}$ r times for Λ. That is, for all $1 \le \ell \le r$ it does the following.

   a) Initialize an empty array A^ℓ with Q places, that is $A^\ell[i] = \perp$ for all $i \in [Q]$.

   b) Run $(\mathsf{vk}^\ell, (X_i^\ell)_{i\in[Q]}, \rho_{\mathcal{A}}, \mathsf{st}_{\Lambda_{\ell,2}}) = \Lambda_{\ell,1}(\mathsf{st}_{\Lambda_{\ell,1}})$. If $\Lambda_{\ell,1}$ does not produce such an output, then $\mathcal{B}$ aborts and outputs $\perp$.

   c) Then $\mathcal{B}$ runs $\left((Y_{i,j}^\ell, \pi_{i,j}^\ell)_{i\in[Q\setminus j]}, \mathsf{st}_{\Lambda_{3,\ell}}\right) = \Lambda_{\ell,2}(j, \mathsf{st}_{\Lambda_{\ell,2}})$ for all $j \in [Q]$. If $\Lambda_{\ell,2}$ only produces correct outputs with respect to vk^ℓ, that is if

$$\bigwedge_{i\in[Q\setminus\ell]} \mathsf{Vfy}(\mathsf{vk}^\ell, X_i^\ell, Y_{i,j}^\ell, \pi_{i,j}^\ell) = 1,$$

   then $\mathcal{B}$ sets $A^\ell[i] := Y_{i,j}^\ell$ for all $i \in [Q \setminus j]$.

   d) $\mathcal{B}$ then samples $j^{*\ell} \xleftarrow{\$} [Q]$. It then proceeds in one of the following cases:

      1. If $\Lambda_{\ell,2}(j^{*\ell}, \mathsf{st}_{\Lambda_{\ell,2}})$ produced any invalid pair of output and proof, that is, if there exists $i \in [Q \setminus j^{*\ell}]$ such that it holds that the Vfy rejects, that is $\mathsf{Vfy}(\mathsf{vk}^\ell, X_i^\ell, Y_{i,j^{*\ell}}^\ell, \pi_{i,j^{*\ell}}^\ell) = 0$, then $\mathcal{B}$ aborts and outputs $\perp$.

2. Otherwise, $\mathcal{B}$ sets $Y^* := A^\ell[j^{*\ell}]$.

e) Set $\mathsf{st}_{\Lambda_{\ell+1,1}} := \Lambda_{\ell,3}(Y^*, \mathsf{st}_{\Lambda_{\ell,3}})$

iii. Finally, $\mathcal{B}$ runs $s \xleftarrow{\$} \Lambda_3(\mathsf{st}_{\Lambda_{r+1,1}})$ and outputs s.

Success probability of $\mathcal{B}$. In order to analyze the success probability of $\mathcal{B}$, we compare the simulation of $\mathcal{A}$ by $\mathcal{B}$ with the description of $\mathcal{A}$. Note that $\mathcal{A}_1$ samples j uniformly at random and $\mathcal{A}_2$ aborts if it is given an invalid pair of output and proof. $\mathcal{B}$ also samples $j^{*\ell}$ uniformly at random from $[Q]$ and aborts if $\Lambda_{\ell,2}(j^{*\ell}, \mathsf{st}_{\Lambda_{\ell,2}})$ produced any invalid pair of output and proof, just like $\mathcal{A}$. However, we are only guaranteed that $A^\ell[j^{*\ell}]$ contains the correct output of $\mathcal{VUF}$ for X_i^ℓ if there is $j' \in [Q \backslash j^{*\ell}]$ such that $\Lambda_{\ell,2}(j', \mathsf{st}_{\ell,2})$ outputs only correct pairs of outputs and proofs, *i.e.*, if this is not the case the simulation of $\mathcal{A}$ by $\mathcal{B}$ deviates from $\mathcal{A}$'s behaviour. Below, we formally prove that $\mathcal{B}$ perfectly simulates $\mathcal{A}$ unless the event described above occurs and upper bound the probability that it occurs by r/Q.

Let $\mathsf{st}_{\Lambda_{\ell,2}}$ be the unique state computed by $\Lambda_{\ell,1}$ and let $j^{*\ell} \in [Q]$ be the unique index that $\Lambda_{\ell,3}$ is executed with. Note that these values are well defined in both $\mathsf{NICA}_N^{\Lambda^{\mathcal{A}}}(\lambda)$ and $\mathsf{NICA}_N^{\mathcal{B}}(\lambda)$. Now, define the event $\mathsf{all\text{-}valid}(\mathsf{st}_{\Lambda_{\ell,2}}, j)$ as the event that $\Lambda_{\ell,2}$ outputs only valid pairs of outputs and proofs. That is

$$\mathsf{all\text{-}valid}(\mathsf{st}_{\Lambda_{\ell,2}}, j) = \begin{cases} 1 & \text{if } \mathsf{Vfy}(\mathsf{vk}^\ell, X_i^\ell, Y_{i,j}^\ell, \pi_{i,j}^\ell) = 1 \text{ for all } i \in [Q \setminus j] \\ 0 & \text{otherwise,} \end{cases}$$

where $(Y_{i,j}^\ell, \pi_{i,j}^\ell)_{i \in [Q \backslash j]} = \Lambda_{\ell,2}(\mathsf{st}_{\Lambda_{\ell,2}}, j)$. Recalling the case in which $\mathcal{B}$'s simulation deviates the hypothetical adversary $\mathcal{A}$, we define the event $\mathsf{bad}(\ell) := \mathsf{all\text{-}valid}(\mathsf{st}_{\Lambda_{\ell,2}}, j^{*\ell}) \bigwedge_{j \in [Q \backslash j^{*\ell}]} \neg\mathsf{all\text{-}valid}(\mathsf{st}_{\Lambda_{\ell,2}}, j)$, that is the event that $\Lambda_{\ell,2}$ only returned only valid pairs of outputs and proofs for $j = j^{*\ell}$ in the ℓ'th simulation of $\mathcal{A}$. Further, we let $\mathsf{bad} := \bigvee_{\ell \in [r]} \mathsf{bad}(\ell)$ be the event that $\mathsf{bad}(\ell)$ occurs for any $\ell \in [r]$.

Next, let $\mathsf{S}(\mathcal{F})$ denote the event that $\mathsf{NICA}_N^{\mathcal{F}}(\lambda) = 1$ for some adversary $\mathcal{F}$ against the non-interactive complexity assumption N. Then we observe the following:

$$\begin{aligned} & \Pr\left[\mathsf{S}(r\text{-}\Lambda^{\mathcal{A}})\right] - \Pr\left[\mathsf{S}(\mathcal{B})\right] \\ = & \Pr\left[\mathsf{S}(r\text{-}\Lambda^{\mathcal{A}}) \wedge \mathsf{bad}\right] + \Pr\left[\mathsf{S}(r\text{-}\Lambda^{\mathcal{A}}) \wedge \neg\mathsf{bad}\right] - \Pr\left[\mathsf{S}(\mathcal{B}) \wedge \mathsf{bad}\right] - \Pr\left[\mathsf{S}(\mathcal{B}) \wedge \neg\mathsf{bad}\right] \\ \leq & \Pr\left[\mathsf{S}(r\text{-}\Lambda^{\mathcal{A}}) \wedge \neg\mathsf{bad}\right] - \Pr\left[\mathsf{S}(\mathcal{B}) \wedge \neg\mathsf{bad}\right] + \Pr\left[\mathsf{bad}\right] \end{aligned}$$

Therefore, we proceed by showing two things:

1. $\Pr\left[\mathsf{S}(r\text{-}\Lambda^{\mathcal{A}}) \wedge \neg\mathsf{bad}\right] = \Pr\left[\mathsf{S}(\mathcal{B}) \wedge \neg\mathsf{bad}\right]$
2. $\Pr\left[\mathsf{bad}\right] \leq r/Q$

In order to prove the first statement, we consider two cases in which $\mathcal{A}$ outputs either $\perp$ or the correct output of $\mathcal{VUF}$ for input X_j^ℓ under verification key vk^ℓ. These are the two cases that $\mathcal{B}$ distinguishes in step ii. d).

1. In the first case $\Lambda_{\ell,2}(j^{*\ell}, \mathsf{st}_{\Lambda_{\ell,2}})$ outputs $(Y_{i,j^{*\ell}}^\ell, \pi_{i,j^{*\ell}}^\ell)_{i \in [Q \backslash j^{*\ell}]}$ such that there is $i \in [Q \setminus j^{*\ell}]$ with $\mathsf{Vfy}(\mathsf{vk}^\ell, X_i^\ell, Y_{i,j^{*\ell}}^\ell, \pi_{i,j^{*\ell}}^\ell) = 0$. Note that in this case, $\mathcal{A}_2$ aborts and outputs $\perp$. $\mathcal{B}$ also aborts and outputs $\perp$ in step ii. d) in the first case.

2. In the second case no such $i \in [Q \setminus j^{*\ell}]$ exists for the output of $\Lambda_{\ell,2}(j^{*\ell}, \mathsf{st}_{\Lambda_{\ell,2}})$. Hence, we have $\mathsf{all\text{-}valid}(\mathsf{st}_{\Lambda_{\ell,2}}, j^{*\ell}) = 1$. Furthermore, since we assumed that bad does not happen, we have that there is also $j \in [Q \setminus j^{*\ell}]$ with $\mathsf{all\text{-}valid}(\mathsf{st}_{\Lambda_{\ell,2}}, j) = 1$ and therefore $A^\ell[j^{*\ell}]$ contains the correct $\mathcal{VUF}$ output, which $\mathcal{B}$ passes on to $\Lambda_{\ell,3}$. Since $\mathcal{A}$ also outputs the correct $\mathcal{VUF}$ value in this case, the two outputs are distributed identically.

We therefore have $\Pr\left[\mathsf{S}(r\text{-}\Lambda^{\mathcal{A}}) \wedge \neg\mathsf{bad}\right] = \Pr\left[\mathsf{S}(\mathcal{B}) \wedge \neg\mathsf{bad}\right]$.

Next, we show that $\Pr[\mathsf{bad}] \leq r/Q$. For this, consider a fixed $\ell \in [r]$ and observe that $\mathsf{bad}(\ell)$ can occur only if there is a unique index $j \in [Q]$ such that $\mathsf{all\text{-}valid}(\mathsf{st}_{\ell,2}, j) = 1$. Hence, the probability that $\mathcal{B}$ draws $j^{*\ell} = j$ in step ii. d) in the ℓ'th round is $1/Q$. We therefore have that $\Pr[\mathsf{bad}(\ell)] = 1/Q$ and it follows by the union bound that $\Pr[\mathsf{bad}] \leq r/Q$. Summing up, we have shown that.

$$\Pr\left[\mathsf{S}(r\text{-}\Lambda^{\mathcal{A}})\right] - \Pr\left[\mathsf{S}(\mathcal{B})\right] \leq \Pr[\mathsf{bad}] \leq r/Q \iff \epsilon_\Lambda \leq \epsilon_{\mathcal{B}} - r/Q$$

It is now only left to compute the running time of $\mathcal{B}$. For this, note that $\mathcal{B}$ executes the algorithms $\Lambda_{\ell,2}$ Q times for each $\ell \in [r]$ and other algorithms of Λ only once. Furthermore, $\mathcal{B}$ executes Vfy $r \cdot Q \cdot (Q-1)$ times. Overall, we therefore conclude that

$$t_{\mathcal{B}} \leq r \cdot Q \cdot t_\Lambda + r \cdot Q \cdot (Q-1) \cdot t_{\mathsf{Vfy}},$$

where t_{Vfy} is the time it takes to execute Vfy. This concludes the proof. □

Non-perfect adversaries. We only considered adversaries that always break the weak selective unpredictability of the VUF in the theorem above. However, the hypothetical adversary $\mathcal{A}$ and the meta-reduction can also simulate adversaries with arbitrary $\epsilon_{\mathcal{A}} \in [0,1]$ by just aborting with probability $1 - \epsilon_{\mathcal{A}}$ in the simulation of $\mathcal{A}$.

3 A reduction strategy with optimal tightness

Now that we showed that every reduction from a non-interactive complexity assumption to the pseudorandomness or unpredictability of a VRF or VUF loses at least a factor of Q, we present a VRF together with a reduction, which attains this bound up to a small constant factor. We achieve this by describing a *partitioning proof strategy*. In these types of proofs, the reduction partitions the input space of the VRF in a controlled set and an uncontrolled set and embeds this partitioning into the verification key. The reduction is then able to answer evaluation queries for inputs in the controlled set and can extract a solution to the underlying complexity assumption if the challenge is in the uncontrolled set. This type of proof has also been used in most of the previous VRFs that do not rely on the random oracle heuristic, for example [Koh19, Jag15, Yam17b, Kat17]. In this section, we describe how the reduction chooses this partition. We discuss the embedding of the partitioning in the VRF in Section 4.

Optimal partitioning. In order to make a partitioning argument with optimal tightness for VRFs, we need to decouple the probability that the partitioning succeeds from the queries and the challenge, which are chosen by the adversary. We achieve this by passing every input of the adversary through a pseudorandom function. This ensures that the outputs are distributed independently and uniformly at random for pairwise different inputs. We formally define a PRF as follows.

Definition 7. For functions $t, m, n : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0, 1]$, we say that a function $\mathsf{PRF} : \{0,1\}^{m(\lambda)} \times \{0,1\}^{\lambda} \to \{0,1\}^{n(\lambda)}$ is an (t, ϵ)-secure *Pseudorandom Function* if it holds for every algorithm $\mathcal{D}$ running in time $t(\lambda)$ that

$$\left| \Pr_{\mathsf{K}^{\mathsf{PRF}} \xleftarrow{\$} \{0,1\}^m} \left[\mathcal{D}^{\mathsf{PRF}(\mathsf{K}^{\mathsf{PRF}},\cdot)}(1^\lambda) = 1 \right] - \Pr_{F \xleftarrow{\$} \mathcal{F}_{\lambda,n(\lambda)}} \left[\mathcal{D}^{F(\cdot)} = 1 \right] \right| \leq \epsilon(\lambda),$$

where $\mathcal{F}_{\lambda,n(\lambda)} = \{F : \{0,1\}^\lambda \to \{0,1\}^{n(\lambda)}\}$ is the set of all functions from $\{0,1\}^\lambda$ to $\{0,1\}^{n(\lambda)}$.

For a clear exposition, assume that all queries by the adversary and the challenge are passed through a truly random function. We later on replace this truly random function with a PRF. If the PRF is secure, then this does only make a negligible difference in the success probability.

We use the outputs X' of the truly function for partitioning in the following way. The reduction draws η uniformly random bits $\mathsf{K}^{\mathsf{part}}$ for some carefully chosen $\eta \in [n(\lambda)]$. It then defines the uncontrolled set, *i.e.*, the set of inputs for which the reduction can extract a solution but not answer evaluation queries, as the set of all inputs whose PRF output match $\mathsf{K}^{\mathsf{part}}$ on the first η bits. We formalize this partitioning as the following function F.

Definition 8. For $X' \in \{0,1\}^{n(\lambda)}$ and $\mathsf{K}^{\mathsf{part}} \in \{0,1\}^{\eta}$, we define

$$\mathsf{F}(X', \mathsf{K}^{\mathsf{part}}) := \begin{cases} 1 & \text{if } X'_{|\eta} = \mathsf{K}^{\mathsf{part}} \\ 0 & \text{otherwise,} \end{cases}$$

where $X'_{|\eta}$ denotes the first η bits of X'.

Such a function F has been used in many previous partitioning arguments, *e.g.* [Jag15, HJ16, Yam17b, Kat17, DKN$^+$20], but has its origin in [BB04b, Sec. 4.1] as *biased binary pseudorandom function*.

Let $\mathsf{TRF} \xleftarrow{\$} \mathcal{F}_{\lambda,n(\lambda)}$ be a truly random function and let $X_1, \ldots, X_Q, X^* \in \{0,1\}^\lambda$ be arbitrary with $X_i \neq X_j$ and $X_i \neq X^*$ for all $i \neq j$. We then let $X'_i := \mathsf{TRF}(X_i)$ and $X^{*'} := \mathsf{TRF}(X^*)$. Observe that we then have that all X'_i and $X^{*'}$ are independent and uniformly random in $\{0,1\}^{n(\lambda)}$. We show in the following Lemma that for $\eta = \lceil \log(Q) \rceil + 1$ and $\mathsf{K}^{\mathsf{part}} \xleftarrow{\$} \{0,1\}^\eta$, where Q is the number of evaluation queries made by the adversary, we have that $\mathsf{F}(X'_i, \mathsf{K}^{\mathsf{part}}) = 0$ for all $i \in [Q]$ and $\mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1$ with probability at least $1/(8Q)$. That means, the partitioning argument has optimal tightness for VRFs up to a small constant factor. We later on show that since a pseudorandom function is indistinguishable from a truly random function, we can efficiently apply this in our construction.

Lemma 1. *Let $Q = Q(\lambda)$ be a polynomial, let $\eta = \eta(\lambda) := \lceil \log(Q) \rceil + 1$ and let $X'_1, \ldots, X'_Q, X^{*'}$ be as above. For $\mathsf{K}^{\mathsf{part}} \xleftarrow{\$} \{0,1\}^\eta$, we then have that*

$$\Pr\left[\mathsf{F}(X'_i, \mathsf{K}^{\mathsf{part}}) = 0 \textit{ for all } 0 \leq i \leq Q \textit{ and } \mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1 \right] \geq 1/(8Q).$$

PROOF. We start by lower bound the probability from the lemma as follows.

$$\begin{aligned}
&\Pr\left[\mathsf{F}(X_i', \mathsf{K}^{\mathsf{part}}) = 0 \text{ for all } 0 \le i \le Q \text{ and } \mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1\right] \\
=&\Pr\left[\mathsf{F}(X_i', \mathsf{K}^{\mathsf{part}}) = 0 \text{ for all } 0 \le i \le Q \mid \mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1\right] \Pr\left[\mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1\right] \\
=&\left(\prod_{i=1}^{Q} \Pr\left[\mathsf{F}(X_i', \mathsf{K}^{\mathsf{part}}) \mid \mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1\right]\right) \Pr\left[\mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1\right] && (1) \\
=&\left(1 - \left(\frac{1}{2}\right)^{\eta}\right)^{Q} \Pr\left[\mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1\right] \\
\ge&\left(1 - \left(\frac{1}{2}\right)^{\eta} Q\right) \Pr\left[\mathsf{F}_{\mathsf{K}}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 1\right] && (2) \\
=&\left(1 - \left(\frac{1}{2}\right)^{\eta} Q\right) \left(\frac{1}{2}\right)^{\eta}
\end{aligned}$$

Observe that Equation (1) holds because all X_i' and $X^{*'}$ are stochastically independent and that Equation (2) follows from Bernoulli's inequality. Next, notice that since $\eta = \lceil \log(Q) \rceil + 1$ we have that $\left(\frac{1}{2}\right)^{\eta} \ge \left(\frac{1}{2}\right)^{\log(Q)+2} = \frac{1}{4Q}$ and $-\left(\frac{1}{2}\right)^{\eta} \ge -\left(\frac{1}{2}\right)^{\log(Q)+1} = -\frac{1}{2Q}$. We can therefore conclude the proof as follows.

$$\begin{aligned}
&\Pr\left[\mathsf{F}(X_i, \mathsf{K}^{\mathsf{part}}) = 0 \text{ for all } 0 \le i \le Q \text{ and } \mathsf{F}(X^{*}, \mathsf{K}^{\mathsf{part}}) = 1\right] \\
\ge&\left(1 - \left(\frac{1}{2}\right)^{\eta} Q\right) \left(\frac{1}{2}\right)^{\eta} \ge \left(1 - \frac{1}{2Q} Q\right) \frac{1}{4Q} = \frac{1}{2}\frac{1}{4Q} = \frac{1}{8Q}
\end{aligned}$$

□

Note that Lemma 1 only holds if all X_i' and $X^{*'}$ are distributed independently and uniformly at random in $\{0,1\}^n$, *e.g.*, if $X_i' = \mathsf{TRF}(X_i)$ for all $i \in [Q]$ and $X^{*'} = \mathsf{TRF}(X^*)$. Observe that we stated our argument for a truly random function instead of a PRF and our construction in Section 4 uses a PRF. We therefore define the function G, which uses a pseudorandom function instead of a truly random function.

Definition 9. For $X \in \{0,1\}^{\lambda}$, $\mathsf{K}^{\mathsf{PRF}} \in \{0,1\}^m$ and $\mathsf{K}^{\mathsf{part}} \in \{0,1\}^{\eta}$, we define

$$\mathsf{G}(X, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) := \mathsf{F}(\mathsf{PRF}(\mathsf{K}^{\mathsf{PRF}}, X), \mathsf{K}^{\mathsf{part}}).$$

Intuitively, Lemma 1 also applies to G and adversarially chosen X_i and X^* because the outputs of the pseudorandom function are indistinguishable from the outputs of a truly random function. Hence, any adversary that is able to efficiently make queries to the PRF such that the probability in Lemma 1 differs significantly from the probability for a truly random function would also be able to distinguish the pseudorandom function from a truly random function. We show that this also holds formally as part of the security proof of the pseudorandomness of VRF in Section 4.2.

4 Verifiable Random Functions with optimal tightness

In order to embed the partitioning argument we described in Section 3 into a VRF, we use the verifiable random function that Yamada describes in [Yam17a, Appendix C]. This is the full version of [Yam17b].

This VRF is well-suited for our purposes, because it enables us to embed the homomorphic evaluation of arbitrary NAND-circuits in the reduction such that the reduction can answer all queries for inputs on which the circuit evaluates to zero and can extract a solution to the underlying complexity assumption for all inputs for which the circuit evaluates to 1. At the same time, the embedding of the circuit hides some of the input bits, all internal states and the output of the circuit from the adversary. We use this property to embed the homomorphic evaluation of G from Definition 9. We first describe bilinear group generators, which we require in the VRF construction and then describe how we model NAND circuits. Finally, we describe the VRF.

Bilinear group generators. We shortly introduce (certified) bilinear group generators, which were originally described in [HJ16]. These allow us to define complexity assumptions relative to the way the bilinear group is chosen end ensure that every group element has a unique encoding, which is required for the unique provability of our construction.

Definition 10. A *Bilinear Group Generator* is a probabilistic polynomial-time algorithm GrpGen that takes as input a security parameter λ (in unary) and outputs $\Pi = (p, \mathbb{G}, \mathbb{G}_T, \circ, \circ_T, e, \phi(1)) \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$ such that the following requirements are satisfied.

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

Definition 11. We say that group generator GrpGen is certified, if there exist deterministic polynomial-time (in the security parameter) algorithms GrpVfy and $\mathsf{GrpElemVfy}$ with the following properties.

Parameter Validation. Given the security parameter (in unary) and a string Π, which is not necessarily generated by GrpGen, algorithm $\mathsf{GrpVfy}(1^\lambda, \Pi)$ outputs 1 *if and only if* Π has the form
$$\Pi = (p, \mathbb{G}, \mathbb{G}_T, \circ, \circ_T, e, \phi(1))$$
and all requirements from Definition 10 are satisfied.

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

NAND circuits. Before describing our construction, we require a formal definition of NAND circuits. The type of circuits we consider take two types of inputs: public inputs and secret inputs. For the function G, which we want to embed in the VRF, we can think of the public input as a VRF input $X \in \{0,1\}^{\lambda}$ and of the secret input as the PRF key $\mathsf{K}^{\mathsf{PRF}}$ and the partitioning key $\mathsf{K}^{\mathsf{part}}$. Like Yamada, we roughly follow the notation of [BHR12] when describing NAND circuits. That is, we assign an index to each input bit and to each gate, beginning with the public input bits, continuing with the secret inputs bits and finally indexing the gates. Formally, if there are $k \in \mathbb{N}$ inputs of which $k_{\mathsf{pub}} \in [k]$ are public input bits and $k_{\mathsf{sec}} = k - k_{\mathsf{pub}}$ are secret input bits, then we set $\mathcal{P} := [k_{\mathsf{pub}}]$ and $\mathsf{S} := [k_{\mathsf{pub}} + 1, k_{\mathsf{pub}} + k_{\mathsf{sec}}]$ as the respective index sets for the public and secret input bits.

For a NAND circuit $C : \{0,1\}^{|\mathcal{P}|+|\mathcal{S}|} \to \{0,1\}$ with c many gates and $|\mathcal{P}| + |\mathcal{S}|$ many input bits, we assign an index $j \in \mathcal{C} := [|\mathcal{P}| + |\mathcal{S}| + 1, |\mathcal{P}| + |\mathcal{S}| + c]$ to each gate. Further, we formalize the wiring of the circuit with the functions $\mathsf{in}_1, \mathsf{in}_2 : \mathcal{C} \to \mathcal{P} \cup \mathcal{S} \cup \mathcal{C}$ that represent the input wires of a gate. We require that for all $j \in \mathcal{C}$ it holds that $\mathsf{in}_1(j) < j$ and $\mathsf{in}_2(j) < j$. This condition ensures that the circuit does not contain any circles.

Since we only consider circuits with a single output bit, we assume without loss of generality that the output of the gate with index $|\mathcal{P}| + |\mathcal{S}| + |\mathcal{C}|$ outputs the overall output of the circuit. Furthermore, we define the depth of a gate j as the maximal distance from any input gate to j. Consequentially, we define the depth of a circuit C as the depth of the gate with index $|\mathcal{P}| + |\mathcal{S}| + |\mathcal{C}|$.

Evaluating a circuit. For a circuit C in the notation above with public inputs $\mathbf{p} = (p_j)_{j \in \mathcal{P}}$, secret inputs $\mathbf{s} = (s_j)_{j \in \mathcal{S}}$, gates with indexes in $\mathcal{C}$ and the wiring encoded by $\mathsf{in}^1, \mathsf{in}^2 : \mathcal{C} \to \mathcal{P} \cup \mathcal{S} \cup \mathcal{C}$, we define the function $\mathsf{value} : \mathcal{P} \cup \mathcal{S} \cup \mathcal{C} \to \{0,1\}$ as follows. For all $j \in \mathcal{P}$ we set $\mathsf{value}(j) := p_j$ and for all $j \in \mathcal{S}$ as $\mathsf{value}(j) := s_j$. Further, for all $j \in \mathcal{C}$, we set $\mathsf{value}(j) := \mathsf{value}(\mathsf{in}^1(j))\,\mathsf{NAND}\,\mathsf{value}(\mathsf{in}^2(j))$. In order to evaluate a circuit on input $\mathbf{p} \in \{0,1\}^{|\mathcal{P}|}$ and $\mathbf{s} \in \{0,1\}^{|\mathcal{S}|}$, we compute $\mathsf{value}(|\mathcal{P}| + |\mathcal{S}| + |\mathcal{C}|)$ since the gate with index $|\mathcal{P}| + |\mathcal{S}| + |\mathcal{C}|$ outputs the overall output of C. Note that the evaluation of the circuit is well defined because we have that for all $j \in \mathcal{C}$ it holds that $\mathsf{in}^1(j) < j$ and $\mathsf{in}^2(j) < j$.

Representing G as a circuit. For our construction, we need to represent G from Definition 9 as a NAND-circuit. However, given the plain definition of G, the number of input bits of the circuit depends on $\eta(\lambda)$, which in turn depends on the number Q of Eval queries made by the adversary. We address this by adapting the encoding of $\mathsf{K}^{\mathsf{part}}$. Namely, we let $\mathsf{PrtSmp}(1^{\lambda}, Q(\lambda))$ be the algorithm that samples $\mathsf{K}^{\mathsf{match}} \xleftarrow{\$} \{0,1\}^{n(\lambda)}$, computes $\eta := \lceil \log(Q(\lambda)) \rceil + 1$ sets $\mathsf{K}^{\mathsf{fixing}} = 1^{\eta} || 0^{n(\lambda) - \eta(\lambda)}$ and outputs $\mathsf{K}^{\mathsf{part}} = (\mathsf{K}^{\mathsf{match}}, \mathsf{K}^{\mathsf{fixing}}) \in (\{0,1\}^{n(\lambda)})^2$. We then adapt the function $\mathsf{F}(X', \mathsf{K}^{\mathsf{part}})$ to compare X and $\mathsf{K}^{\mathsf{match}}$ on all positions where $\mathsf{K}^{\mathsf{fixing}}$ is 1 and output 1 if they match on all such positions and 0 otherwise. These adaptations do not change the output of F or G but ensure that the NAND-circuit representing G only depends on λ and not on Q. Note that it would be possible to encode $\mathsf{K}^{\mathsf{fixing}}$ more efficiently, but we use this encoding for simplicity.

Construction. We assume that the NAND-circuits for the function G for different security parameters are publicly known and we denote the circuit for G with security parameter λ by $C_{\mathsf{G},\lambda}$. For our construction, we have that $\mathcal{P} = [\lambda]$, since the public input of G is $X \in \{0,1\}^{\lambda}$. Furthermore, we set $\mathcal{S}^{\mathsf{PRF}} := [|\mathcal{P}| + 1, |\mathcal{P}| + m(\lambda)]$ for the indexes of the bits of $\mathsf{K}^{\mathsf{PRF}} \in \{0,1\}^{m(\lambda)}$, $\mathcal{S}^{\mathsf{part}} := [|\mathcal{P}| + |\mathcal{S}^{\mathsf{PRF}}| + 1, |\mathcal{P}| + |\mathcal{S}^{\mathsf{PRF}}| + 2n(\lambda)]$ for the indexes of $\mathsf{K}^{\mathsf{match}} \in \{0,1\}^{2n(\lambda)}$, and $\mathcal{S} := \mathcal{S}^{\mathsf{PRF}} \cup \mathcal{S}^{\mathsf{part}}$. Finally, we assume that the function $\mathsf{in}^1_{\lambda}, \mathsf{in}^2_{\lambda} : \mathcal{C} \to \mathcal{P} \cup \mathcal{S} \cup \mathcal{C}$ encode the wiring of $C_{\mathsf{G},\lambda}$ and that $|\mathcal{P}| + |\mathcal{S}| + |\mathcal{C}|$ is the index of the output gate. For simplicity, we set $\mathsf{out} := |\mathcal{P}| + |\mathcal{S}| + |\mathcal{C}|$.

$\mathsf{Gen}(1^\lambda)$ first generates a group description $\Pi \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$ and samples uniformly random group generators $g, h \xleftarrow{\$} \mathbb{G} \setminus \{0\}$, $w_0 \xleftarrow{\$} \mathbb{Z}_p^*$ and $w_j \xleftarrow{\$} \mathbb{Z}_p$ for all $j \in \mathcal{S}$. It then sets $W_0 := g^{w_0}$, $W_j := g^{w_j}$ for all $j \in \mathcal{S}$ and outputs

$$\mathsf{vk} := \left(\Pi, g, h, W_0, (W_j)_{j \in \mathcal{S}}\right) \qquad \text{and} \qquad \mathsf{sk} := \left(w_0, (w_j)_{j \in \mathcal{S}}\right).$$

$\mathsf{Eval}(\mathsf{sk}, X)$ parses $X \in \{0,1\}^\lambda$ as $(X_1, \ldots, X_\lambda)$ and sets

$$\theta_j := \begin{cases} X_j & \text{if } j \in \mathcal{P} \\ w_j & \text{if } j \in \mathcal{S} \end{cases}$$

for all $j \in \mathcal{P} \cup \mathcal{S}$. For all $j \in \mathcal{C}$, it sets

$$\theta_j := 1 - \theta_{\mathsf{in}^1_\lambda(j)} \theta_{\mathsf{in}^2_\lambda(j)}.$$

It then sets $\pi_0 := g^{\theta_{\mathsf{out}}/w_0}$ and $\pi_j := g^{\theta_j}$ for all $j \in \mathcal{C}$ and outputs

$$Y := e(g,h)^{\theta_{\mathsf{out}}/w_0} \qquad \text{and} \qquad \pi := (\pi_0, (\pi_j)_{j \in \mathcal{C}}).$$

$\mathsf{Vfy}(\mathsf{vk}, X, Y, \pi)$ verifies that vk has the form $(\Pi, g, h, W_0, (W_j)_{j \in \mathcal{S}})$ and that π has the form $(\pi_0, (\pi_j)_{j \in \mathcal{C}})$. It then verifies the group description by running $\mathsf{GrpVfy}(1^\lambda, \Pi)$ and then verifies all group elements in vk, π and Y by running $\mathsf{GrpElemVfy}(1^\lambda, \Pi, s)$ for all $s \in \{g, h, Y, \pi_0, \pi_{|\mathcal{P}|+|\mathcal{S}|+1}, \ldots, \pi_{|\mathcal{P}|+|\mathcal{S}|+|\mathcal{C}|}\}$. Vfy outputs 0 if any of the checks fails. Next, the algorithm verifies the correctness of Y in respect to vk, X and π by setting $\pi_j := g^{X_j}$ for all $j \in \mathcal{P}$ and $\pi_j := W_j$ for all $i \in \mathcal{S}$ and performing the following steps.

1. It checks whether $e(g, \pi_j) = e(g,g) \left(e(\pi_{\mathsf{in}^1_\lambda(j)}, \pi_{\mathsf{in}^2_\lambda(j)}) \right)^{-1}$ for all $j \in \mathcal{C}$.
2. It checks whether $e(\pi_0, W_0) = e(\pi_{\mathsf{out}}, g)$.
3. It checks whether $e(\pi_0, h) = Y$.

If any of the checks above fail, then Vfy outputs 0. Otherwise, it outputs 1.

Instantiation. In order to instantiate the $\mathcal{VRF}$, we need that G can be represented by a circuit of polynomial size and logarithmic depth. While this is certainly possible for the comparison of the PRF output with $\mathsf{K}^{\mathsf{match}}$, we also require a PRF that can be computed by such a NAND circuit. The Naor-Reingold PRF is an example of such a PRF that is also provably secure under the DDH assumption [NR97]. However, we can further optimize the efficiency by using the adaptation of the Naor-Reingold PRF in [JKP18, Section 5.1]. This PRF has secret keys of size $\omega(\log(\lambda))$. Further, we can change the encoding of $\mathsf{K}^{\mathsf{match}}$ and $\mathsf{K}^{\mathsf{fixing}}$ to also consist of only $\omega(\log(\lambda))$ many bits. This would bring the size of the public verification key down to $\omega(\log(\lambda))$, would however only hold for λ large enough. We can further optimize the size of the proofs by applying the technique of [IKOS08], which allows to reduce the circuit size of every PRF to $\mathcal{O}(\lambda)$ at the cost of reducing the output length to $\lambda^{1/c}$ for some constant $c > 0$ that depends on the PRF. However the smaller output length is no issue, since $\lambda^{1/c}$ is larger than $\lceil \log(Q(\lambda)) \rceil + 1 = \mathcal{O}(\log(\lambda))$ for large enough λ, because Q is polynomial in λ. This technique therefore reduces the size of proofs to $\mathcal{O}(\lambda)$.

4.1 Correctness and Unique Provability of the VRF

The proofs for correctness and unique provability closely follow the respective proofs by Yamada [Yam17a]. We therefore only present them here for completeness. Before proving the pseudorandomness of the VRF, we shortly discuss the instantiation with concrete PRFs and the effect on the efficiency.

Correctness. We prove the correctness of $\mathcal{VRF}$ by considering an arbitrary input $X \in \{0,1\}^\lambda$. Let $(\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ and $(Y, \pi) := \mathsf{Eval}(\mathsf{sk}, X)$, then the algorithm $\mathsf{Vfy}(\mathsf{vk}, X, Y, \pi)$ first verifies the structure of vk and π. This verification succeeds because vk and π are generated in this specific format by Gen and Eval. The same applies to the verification of Π and the encoding of the group elements by GrpVfy and $\mathsf{GrpElemVfy}$. Further, the first check succeeds because Eval computes π_j for all $j \in \mathcal{C}$ such that

$$\begin{aligned} e(g, \pi_j) = e(g, g^{\theta_j}) = e(g, g^{1-\theta_{\mathsf{in}^1_\lambda(j)}\theta_{\mathsf{in}^2_\lambda(j)}}) &= e(g,g) e\left(g, g^{\theta_{\mathsf{in}^1_\lambda(j)}\theta_{\mathsf{in}^2_\lambda(j)}}\right)^{-1} \\ = e(g,g) e\left(g^{\theta_{\mathsf{in}^1_\lambda(j)}}, g^{\theta_{\mathsf{in}^2_\lambda(j)}}\right)^{-1} &= e(g,g) e\left(\pi_{\mathsf{in}^1_\lambda(j)}, \pi_{\mathsf{in}^2_\lambda(j)}\right)^{-1}. \end{aligned}$$

Further, the second check succeeds because Eval and Gen compute $\pi_0, \pi_{\mathsf{out}}$ and W_0 such that $e(\pi_0, W_0) = e(g^{\theta_{\mathsf{out}}/w_0}, g^{w_0}) = e(g^{\theta_{\mathsf{out}}}, g) = e(\pi_{\mathsf{out}}, g)$. Finally, we have that

$$e(\pi_0, h) = e(g^{\theta_{\mathsf{out}}/w_0}, h) = e(g, h)^{\theta_{\mathsf{out}}/w_0} = Y.$$

Therefore, Vfy outputs 1, which proves the correctness of $\mathcal{VRF}$.

Unique Provability. In order to show that $\mathcal{VRF}$ has unique provability, we have to show that for every $\mathsf{vk} \in \{0,1\}^*$ and $X \in \{0,1\}^\lambda$ there does not exist $Y^0, \pi^0, Y^1, \pi^1 \in \{0,1\}^*$ with $Y^0 \neq Y^1$ such that $\mathsf{Vfy}(\mathsf{vk}, X, Y^0, \pi^0) = \mathsf{Vfy}(\mathsf{vk}, X, Y^1, \pi^1) = 1$.

We do so by assuming that there are $\mathsf{vk}, Y^0, \pi^0, Y^1, \pi^1 \in \{0,1\}^*$ such that $\mathsf{Vfy}(\mathsf{vk}, X, Y^0, \pi^0) = \mathsf{Vfy}(\mathsf{vk}, X, Y^1, \pi^1) = 1$ and then conclude that $Y^0 = Y^1$ has to hold by going through the checks of the verification algorithm. Vfy first checks whether vk and π both have the correct format and that supposed group elements in vk, π and Y are actual group elements with a unique encoding. Since we assumed that $\mathsf{Vfy}(\mathsf{vk}, X, Y^0, \pi^0) = \mathsf{Vfy}(\mathsf{vk}, X, Y^1, \pi^1) = 1$, we from now on assume that $\mathsf{vk}, Y^0, \pi^0, Y^1, \pi^1$ fulfill these conditions.

Next, observe that it follows from the group structure of $\mathbb{G}$ that $\pi_j = g^{X_j}$ is uniquely defined for all $j \in \mathcal{P}$ and that $\log_g(\pi_j) = w_j$ uniquely defines $w_j \in \mathbb{Z}_p$ for all $j \in \mathcal{S}$. $\mathsf{Vfy}(\mathsf{vk}, X, Y_0, \pi_0) = \mathsf{Vfy}(\mathsf{vk}, X, Y_1, \pi_1) = 1$. Then, the first check of Vfy inductively specifies a unique $\pi_j \in \mathbb{G}$ such that $e(g, \pi_j) = e(g,g)\left(e(\pi_{\mathsf{in}^1_\lambda(j)}, \pi_{\mathsf{in}^2_\lambda(j)})\right)^{-1}$ holds for all $j \in \mathcal{C}$. This implies that the values π_j are identical in π^0 and π^1. The second check of Vfy then uniquely specifies π_0, because W_0 and π_{out} are uniquely specified. Hence, π_0 has to be identical in π^0 and π^1. Finally, the last check of Vfy uniquely specifies Y because π_0 is already unique. Therefore, $Y^0 = Y^1$ has to hold, which proves the unique provability of $\mathcal{VRF}$.

4.2 Proof of pseudorandomness

The security of our VRF is based on the decisional q-bilinear Diffie-Hellman inversion assumption that we formally introduce below.

Definition 12 (Definition 4 in [BB04a]). For a bilinear group generator GrpGen, an algorithm $\mathcal{B}$ and $q \in \mathbb{N}$, let $G_{\mathcal{B}}^{q\text{-}\mathsf{DBDHI}}(\lambda)$ be the following game. The challenger runs $\Pi \xleftarrow{\$} \mathsf{GrpGen}(1^\lambda)$, samples $g, h \xleftarrow{\$} \mathbb{G}$, $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and $b \xleftarrow{\$} \{0, 1\}$. Then it defines $T_0 := e(g, h)^{1/\alpha}$ and $T_1 \xleftarrow{\$} \mathbb{G}_T$. Finally, it runs the bit b' to $b' \xleftarrow{\$} \mathcal{B}(\Pi, g, h, g^\alpha, \ldots, g^{\alpha^q}, T_b)$, and outputs 1 if $b = b'$, and 0 otherwise. We denote with

$$\mathsf{Adv}_{\mathcal{B}}^{q\text{-}\mathsf{DBDHI}}(\lambda) := \left|\Pr\left[G_{\mathcal{B}}^{q\text{-}\mathsf{DBDHI}}(\lambda) = 1\right] - 1/2\right|$$

the *advantage* of $\mathcal{B}$ in breaking the q-DBDHI-assumption for groups generated by GrpGen, where the probability is taken over the randomness of the challenger and $\mathcal{B}$. For functions $t : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0, 1]$, we say that $\mathcal{B}$ (t, ϵ)-breaks the q-DBDHI assumption relative to GrpGen, if $\mathsf{Adv}_{\mathcal{B}}^{q\text{-}\mathsf{DBDHI}}(\lambda) = \epsilon(\lambda)$ and $\mathcal{B}$ runs in time $t(\lambda)$.

Note that the assumption falls in the category of non-interactive complexity assumptions from Definition 3. Based on this assumption, we can formulate the theorem for the pseudorandomness of our VRF.

Theorem 2. *Let $\mathcal{VRF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Vfy})$ be the verifiable random function above, then for every legitimate adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that $(t_{\mathcal{A}}, \epsilon_{\mathcal{A}})$ breaks the pseudorandomness of $\mathcal{VRF}$ and makes $Q(\lambda)$ queries to* Eval *for some polynomial $Q : \mathbb{N} \to \mathbb{N}$, there exists an algorithm $\mathcal{B}$ that $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$-breaks the q-*DBDHI *assumption relative to* GrpGen *used in $\mathcal{VRF}$ with*

$$t_{\mathcal{B}}(\lambda) = t_{\mathcal{A}}(\lambda), \qquad \epsilon_{\mathcal{B}}(\lambda) \geq \frac{\epsilon_{\mathcal{A}}(\lambda)}{8Q(\lambda)} - \epsilon_{\mathsf{PRF}}(\lambda) - \mathsf{negl}(\lambda) \qquad \textit{and} \qquad q := 2^d,$$

where d is the depth of the circuit for G, *ϵ_{PRF} is the largest advantage any algorithm with runtime $t_{\mathcal{A}}(\lambda)$ that makes $Q(\lambda)$ queries to its oracle has in breaking the security of the* PRF *used in $\mathcal{VRF}$ and* $\mathsf{negl}(\lambda)$ *is a negligible function. In particular: $\mathcal{VRF}$ achieves the optimal tightness, since $\epsilon_{\mathsf{PRF}}(\lambda)$ is negligible if the construction is instantiated with a PRF with a security reduction loss of at most $Q(\lambda)$.*

Remark 1. Note that the requirement of a loss of at most Q for the PRF is fulfilled by *e.g.* the Naor-Reingold PRF [NR97] or the PRFs by Jager*et al.* [JKP18].

PROOF. Since Eval is deterministic, $\mathcal{A}$ can not learn anything by making the same query to Eval twice. We therefor assume without loss of generality that $\mathcal{A}$ makes only pairwise distinct queries to Eval. Further, we set $Q := Q(\lambda), n := n(\lambda), m := m(\lambda)$ and $\epsilon_{\mathcal{A}} := \epsilon_{\mathcal{A}}(\lambda)$ in order to simplify notation.

We prove Theorem 2 with a sequence of games argument [Sho04]. We denote the event that Game i outputs 1 by E_i. The first part of the proof will focus on our technique of using a PRF for partitioning. The second part of the proof follows the proof by Yamada [Yam17a, Theorem 6] and we provide it mostly for completeness.

Game 0. This is the original security experiment from Definition 1 and we therefore have that

$$\left|\Pr[E_0] - \frac{1}{2}\right| = \epsilon_{\mathcal{A}}$$

holds by definition.

Game 1. In this game, the challenger first runs the game as before. But, before outputting a result, it samples $X_i' \xleftarrow{\$} \{0,1\}^n$ uniformly and independently at random for each query $X_i \in \{0,1\}^\lambda$ to Eval by $\mathcal{A}$ and $X^{*'} \xleftarrow{\$} \{0,1\}^n$ for the challenge $X^* \in \{0,1\}^\lambda$. Observe that this perfectly emulates the process of evaluating a truly random function on the queries and the challenge because we assumed without loss generality that all queries and the challenge are pairwise distinct. Further, it sets $\eta := \lceil \log Q \rceil + 1$ and samples $\mathsf{K}^{\mathsf{part}} \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q)$. It then aborts and outputs a random bit if $\mathsf{F}(X_i', \mathsf{K}^{\mathsf{part}}) = 1$ for any $i \in [Q]$ or if $\mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 0$. We denote the occurrence of any of the two abort conditions by the event bad. We next show that

$$|\Pr[E_1] - \Pr[E_0]| = \epsilon_{\mathcal{A}}(1 - \Pr[\mathsf{bad}]) \leq \epsilon_{\mathcal{A}} \left(1 - \frac{1}{8Q}\right).$$

We use later that $\Pr[\neg\mathsf{bad}] \geq 1/(8Q)$, which follows from Lemma 1 and will in the end yield the loss stated in Theorem 2. We have the following.

$$\begin{aligned}
|\Pr[E_1] - \Pr[E_0]| &= |\Pr[E_1 \mid \mathsf{bad}] \Pr[\mathsf{bad}] + \Pr[E_1 \mid \neg\mathsf{bad}] \Pr[\neg\mathsf{bad}] - \Pr[E_0]| \\
&= \left| \frac{1}{2}(1 - \Pr[\neg\mathsf{bad}]) + \Pr[E_1 \mid \neg\mathsf{bad}] \Pr[\neg\mathsf{bad}] - \Pr[E_0] \right| \\
&= \left| \frac{1}{2} + \Pr[\neg\mathsf{bad}] \left(\Pr[E_1 \mid \neg\mathsf{bad}] - \frac{1}{2} \right) - \Pr[E_0] \right| \\
&= \left| \frac{1}{2} + \Pr[\neg\mathsf{bad}] \left(\Pr[E_0] - \frac{1}{2} \right) - \Pr[E_0] \right| \qquad (3) \\
&= \left| \Pr[\neg\mathsf{bad}] \left(\Pr[E_0] - \frac{1}{2} \right) - \left(\Pr[E_0] - \frac{1}{2} \right) \right| \\
&= \left| \left(\Pr[E_0] - \frac{1}{2} \right) (\Pr[\neg\mathsf{bad}] - 1) \right| \\
&= \left| \Pr[E_0] - \frac{1}{2} \right| \cdot |\Pr[\neg\mathsf{bad}] - 1| \\
&= \epsilon_{\mathcal{A}} \cdot (1 - \Pr[\neg\mathsf{bad}])
\end{aligned}$$

Note that Equation (3) holds because $\Pr[E_1 \mid \neg\mathsf{bad}] = \Pr[E_0 \mid \neg\mathsf{bad}]$ and the event $\neg\mathsf{bad}$ is independent from E_0. The independence holds because $X^{*'}$ and all X_i' are drawn at random. Note that it is this independence together with the independence between the different X_i' and X^* that allows us to achieve the optimal tightness in contrast to the other approaches discussed in the introduction.

Further, by Lemma 1, we have that $\Pr[\neg\mathsf{bad}] \geq 1/(8Q)$ holds and therefore

$$|E_1 - E_0| = \epsilon_{\mathcal{A}}(1 - \Pr[\neg\mathsf{bad}]) \leq \epsilon_{\mathcal{A}} \left(1 - \frac{1}{8Q}\right).$$

Game 2. In this game, the challenger only changes the way it computes $X^{*'}$ and X_i' for all $i \in [Q]$. The challenger samples $\mathsf{K}^{\mathsf{PRF}} \xleftarrow{\$} \{0,1\}^m$ and aborts and outputs a random bit if $\mathsf{G}(X_i, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) = 1$ or if $\mathsf{G}(X^*, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) = 0$. The only difference to Game 1 is that G sets $X^{*'} := \mathsf{PRF}(\mathsf{K}^{\mathsf{PRF}}, X^*)$ and $X_i' := \mathsf{PRF}(\mathsf{K}^{\mathsf{PRF}}, X_i)$ instead of drawing them uniformly at random.

Informally, every algorithm distinguishing Game 2 from Game 1 with advantage ϵ implies a distinguisher for PRF with advantage ϵ. We describe a distinguisher $\mathcal{B}_{\mathsf{PRF}}$ for PRF that is based on Game 2 and

Game 1 and achieves exactly this: $\mathcal{B}_{\mathsf{PRF}}(\lambda)$ with access to either a $\mathsf{PRF}(\mathsf{K}^{\mathsf{PRF}}, \cdot)$ or a truly random function $F \xleftarrow{\$} \mathcal{F}_{\lambda,n(\lambda)}$ as oracle first runs $(\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ and uses sk to answer all queries and the challenge by $\mathcal{A}$. After $\mathcal{A}$ submits its guess b', $\mathcal{B}_{\mathsf{PRF}}$ queries its oracle on X_i and by that obtains X_i' for all $i \in [Q]$. Analogously, it queries its oracle on X^* and by that obtains $X^{*'}$. It then samples $\mathsf{K}^{\mathsf{part}} \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q)$ and aborts and outputs a random bit if $\mathsf{F}(X^{*'}, \mathsf{K}^{\mathsf{part}}) = 0$ or $\mathsf{F}(X_i', \mathsf{K}^{\mathsf{part}}) = 1$ for some $i \in [Q]$. Otherwise, $\mathcal{B}_{\mathsf{PRF}}$ outputs 1 if $\mathcal{A}$'s guess is correct and 0 otherwise.

Note that $\mathcal{B}$ has exactly the same runtime as $\mathcal{A}$ and that the probability that it outputs 1 is identical to $\Pr[E_2]$ if its oracle is the pseudorandom function. Analogously, if its oracle is a truly random function, then its output is 1 with probability $\Pr[E_1]$. We therefore have

$$|\Pr[E_2] - \Pr[E_1]| \;=\; \left| \Pr_{\mathsf{K}^{\mathsf{PRF}} \xleftarrow{\$} \{0,1\}^m} \left[\mathcal{B}_{\mathsf{PRF}}^{\mathsf{PRF}(\mathsf{K}^{\mathsf{PRF}},\cdot)}(1^\lambda) = 1\right] - \Pr_{F \xleftarrow{\$} \mathcal{F}_{\lambda,n(\lambda)}} \left[\mathcal{B}_{\mathsf{PRF}}^{F(\cdot)} = 1\right] \right| \;\leq\; \epsilon_{\mathsf{PRF}}.$$

Game 3. In this game, the challenger samples $\mathsf{K}^{\mathsf{PRF}} \xleftarrow{\$} \{0,1\}^m$ and $\mathsf{K}^{\mathsf{part}} \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q)$ in the very beginning and aborts and outputs a random bit as soon as $\mathcal{A}$ makes an Eval query X_i with $\mathsf{G}(X_i, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) = 1$ or if it holds for $\mathcal{A}$'s challenge X^* that $\mathsf{G}(X^*, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) = 0$. Since this is just a conceptual change, we have that

$$\Pr[E_3] = \Pr[E_2].$$

From here on, the proof mostly follows the proof by Yamada [Yam17a, Appendix C] and we present it here for completeness.

Game 4. In this game, we change the way the w_j are chosen. That is, the challenger samples the partitioning key $\mathsf{K}^{\mathsf{part}} \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q)$ with $\mathsf{K}^{\mathsf{part}} \in \{0,1\}^{|\mathcal{S}^{\mathsf{part}}|}$ and $\mathsf{K}^{\mathsf{PRF}} \xleftarrow{\$} \{0,1\}^{|\mathcal{S}^{\mathsf{PRF}}|}$. For all $j \in \mathcal{S}$ it sets $s_j := \mathsf{K}^{\mathsf{PRF}}_{j-|\mathcal{P}|}$ for all $j \in \mathcal{S}^{\mathsf{PRF}}$ and $s_j := \mathsf{K}^{\mathsf{part}}_{j-|\mathcal{P}|-|\mathcal{S}^{\mathsf{PRF}}|}$ for all $j \in \mathcal{S}^{\mathsf{part}}$. The challenger then samples $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$, and $\tilde{w}_j \xleftarrow{\$} \mathbb{Z}_p^*$ for all $j \in \mathcal{S}$. It then sets

$$w_0 := \tilde{w}_0 \alpha \qquad \text{and} \qquad w_j := \tilde{w}_j \cdot \alpha + s_j \qquad \text{for all } j \in \mathcal{S}.$$

Note that the $\tilde{w}_j$ are drawn from $\mathbb{Z}_p^*$ and not from $\mathbb{Z}_p$ like the w_j in the previous game. This slightly changes the distributions of the w_j. However, the overall statistical distance is at most $|\mathcal{S}|/p$, which is negligible because $p = \Omega(2^\lambda)$ by Definition 10. We therefore have that

$$|E_4 - E_3| = \mathsf{negl}(\lambda).$$

Before proceeding to the next game, we introduce additional notation. That is, for all $X \in \{0,1\}^\lambda$ and all $j \in \mathcal{P} \cup \mathcal{S} \cup \mathcal{C}$, we let

$$\mathsf{P}_{X,j}(\mathsf{Z}) := \begin{cases} X_j & \text{if } j \in \mathcal{P}, \\ \tilde{w}_i \mathsf{Z} + s_j & \text{if } j \in \mathcal{S} \text{ and} \\ 1 - \mathsf{P}_{X,\mathsf{in}^1_\lambda(j)}(\mathsf{Z}) \mathsf{P}_{X,\mathsf{in}^2_\lambda(j)}(\mathsf{Z}) & \text{if } j \in \mathcal{C}. \end{cases}$$

Note that by the definition of w_j form Game 3, we have that $\mathsf{P}_{X,j}(\alpha) = \theta_j$. In order to proceed to the next game, we require the following lemma by Yamada.

Lemma 2 (Lemma 16 in [Yam17a]). *There exists* $\mathsf{R}_X(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ *with* $\mathsf{deg}(\mathsf{R}(\mathsf{Z})) \leq \mathsf{deg}(\mathsf{P}_{X,\mathsf{out}}(\mathsf{Z})) \leq 2^d$*, where d is the depth of the circuit for the function* G*, and*

$$\mathsf{P}_{X,\mathsf{out}}(\mathsf{Z}) = \mathsf{G}(X, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) + \mathsf{Z} \cdot \mathsf{R}_X(\mathsf{Z}).$$

We provide the proof of Lemma 2 in Appendix A for completeness.

Game 5. With Lemma 2 at our hands, we change how the challenger answers $\mathcal{A}$'s queries to Eval in this game. As in the previous game, the challenger aborts and outputs a random bit if $\mathsf{G}(X_i, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) = 1$ for any query X_i by $\mathcal{A}$. Otherwise, the challenger computes and outputs

$$Y := e\left(g^{\mathsf{R}_X(\alpha)/\tilde{w}_0}, h\right), \qquad \pi := \left(\pi_0 = g^{\mathsf{R}_X(\alpha)/\tilde{w}_0}, \left(\pi_j := g^{\mathsf{P}_{X,j}(\alpha)}\right)_{j \in \mathcal{C}}\right).$$

Observe that Y and π are distributed exactly as in Game 4. This holds for all π_j because $\mathsf{P}_{X,j}(\mathsf{Z})$ is defined exactly as P_j in the definition of Eval above, just with w_j defined as in Game 4. Further, it holds for π_0 and Y because

$$\frac{\mathsf{R}_X(\alpha)}{\tilde{w}_0} = \frac{\alpha \cdot \mathsf{R}_X(\alpha)}{\alpha \cdot \tilde{w}_0} = \frac{\mathsf{G}(X, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) + \alpha \cdot \mathsf{R}_X(\alpha)}{\alpha \cdot \tilde{w}_0} = \frac{\mathsf{P}_{X,\mathsf{out}}(\alpha)}{w_0},$$

where the last equality follows from Lemma 2. We therefore have that

$$\Pr[E_5] = \Pr[E_4].$$

Game 6. In this game, we change how the challenger answers to $\mathcal{A}$'s challenge X^*. As in the previous game, the challenger aborts and outputs a random bit if $\mathsf{G}(X^*, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) = 0$. Otherwise, the challenger computes $\mathsf{R}_{X^*}(\alpha)$ and sets

$$\begin{aligned} Y_0 &:= \left(e(g,h)^{1/\alpha} \cdot e\left(g^{\mathsf{R}_{X^*}(\alpha)}, h\right)\right)^{1/\tilde{w}_0} = e\left(g^{(1+\alpha \mathsf{R}_{X^*}(\alpha))/(\tilde{w}_0 \alpha)}, h\right) \\ &= e\left(g^{(\mathsf{G}(X^*, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) + \alpha \mathsf{R}_{X^*}(\alpha))/(\tilde{w}_0 \alpha)}, h\right) = e\left(g^{\mathsf{P}_{X^*,\mathsf{out}}(\alpha)/w_0}, h\right) \end{aligned}$$

Then, the challenger samples a uniformly random bit b and $Y_1 \xleftarrow{\$} \mathbb{G}_T$ and outputs Y_b to $\mathcal{A}$. Again, observe that $\mathsf{P}_{X^*,\mathsf{out}}(\alpha)$ is, relative to w_j as defined in Game 4, distributed exactly as θ_{out} in the definition of Eval . We therefore have that

$$\Pr[E_6] = \Pr[E_5].$$

We now claim that there is an algorithm $\mathcal{B}$ that runs in time $t_{\mathcal{A}}$ and solves the q-DBDHI problem probability $\Pr[E_6]$.

Lemma 3. *Let $d \in \mathbb{N}$ be the depth of the $C_{\mathsf{G},\lambda}$, then there is an algorithm $\mathcal{B}$ with run time $t_{\mathcal{B}} \approx t_{\mathcal{A}}$ that on input a q-*DBDHI *instance with $q = 2^d$ perfectly simulates Game 6 such that* $\Pr\left[G_{\mathcal{B}}^{q\text{-}\mathsf{DBDHI}}(\lambda) = 1\right] = \Pr[E_6]$.

As not to interrupt the proof of Theorem 2, we postpone the proof of Lemma 3 and first conclude the proof of Theorem 2. By Lemma 3 and the (in)equalities we derived above we have that

$$\begin{aligned}\epsilon_{\mathcal{A}} &= \left|\Pr\left[E_0\right] - \frac{1}{2}\right| \leq \left|\Pr\left[E_0\right] - \Pr\left[E_1\right]\right| + \left|\Pr\left[E_1\right] - \frac{1}{2}\right| \\ &\leq \epsilon_{\mathcal{A}}\left(1 - \frac{1}{8Q}\right) + \left|\Pr\left[E_1\right] - \frac{1}{2}\right| \\ &\leq \epsilon_{\mathcal{A}}\left(1 - \frac{1}{8Q}\right) + \epsilon_{\mathsf{PRF}} + \left|\Pr\left[E_2\right] - \frac{1}{2}\right| \\ &= \epsilon_{\mathcal{A}}\left(1 - \frac{1}{8Q}\right) + \epsilon_{\mathsf{PRF}} + \left|\Pr\left[E_3\right] - \frac{1}{2}\right| \\ &\leq \epsilon_{\mathcal{A}}\left(1 - \frac{1}{8Q}\right) + \epsilon_{\mathsf{PRF}} + \mathsf{negl}(\lambda) + \left|\Pr\left[E_4\right] - \frac{1}{2}\right| \\ &= \epsilon_{\mathcal{A}}\left(1 - \frac{1}{8Q}\right) + \epsilon_{\mathsf{PRF}} + \mathsf{negl}(\lambda) + \left|\Pr\left[E_6\right] - \frac{1}{2}\right| \\ &= \epsilon_{\mathcal{A}}\left(1 - \frac{1}{8Q}\right) + \epsilon_{\mathsf{PRF}} + \mathsf{negl}(\lambda) + \epsilon_{\mathcal{B}}\end{aligned}$$

Rearranging the terms, we have that

$$\epsilon_{\mathcal{B}} \geq \frac{\epsilon_{\mathcal{A}}}{8Q} - \epsilon_{\mathsf{PRF}} - \mathsf{negl}(\lambda).$$

This concludes the proof of Theorem 2. □

In order to finalize the proof, we only need to prove Lemma 3, which we do next.

Proof of Lemma 3. On input $(\Pi, g, h, g^{\alpha}, \ldots, g^{\alpha^q}, T)$, where T is, both with probability $1/2$, either $e(g,h)^{1/\alpha}$ or a random element in $\mathbb{G}_T$, the algorithm $\mathcal{B}$ samples $\tilde{w}_0 \xleftarrow{\$} \mathbb{Z}_p^*$ and $\tilde{w}_j \xleftarrow{\$} \mathbb{Z}_p^*$ for all $i \in \mathcal{S}$. It further samples $\mathsf{K}^{\mathsf{part}} \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q)$ and $\mathsf{K}^{\mathsf{PRF}} \xleftarrow{\$} \{0,1\}^m$. For all $j \in \mathcal{S}$ it then sets

$$W_j := \begin{cases} (g^{\alpha})^{\tilde{w}_j}\, g^{\mathsf{K}^{\mathsf{PRF}}_{j-|\mathcal{P}|}} & \text{if } j \in \mathcal{S}^{\mathsf{PRF}} \text{ and} \\ (g^{\alpha})^{\tilde{w}_j}\, g^{\mathsf{K}^{\mathsf{part}}_{j-|\mathcal{P}|-|\mathcal{S}^{\mathsf{PRF}}|}} & \text{if } j \in \mathcal{S}^{\mathsf{part}}. \end{cases}$$

Further, $\mathcal{B}$ sets $W_0 := (g^{\alpha})^{\tilde{w}_0}$. It then gives $\mathsf{vk} := (\Pi, g, h, W_0, (W_j)_{j\in\mathcal{S}})$ to $\mathcal{A}$. Whenever $\mathcal{A}$ makes a query X_i to Eval , then $\mathcal{B}$ computes the coefficients of the polynomials $\mathsf{P}_{X_i,j}(\mathsf{Z})$ for all $j \in \mathcal{C}$. Note that by Lemma 2, we have that the coefficient for degree zero of $\mathsf{P}_{X_i,\mathsf{out}}(\mathsf{Z})$ is identical to $\mathsf{G}(X_i, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}})$. Hence, if the coefficient of degree zero is 1 for any query X_i, then $\mathcal{B}$ aborts and outputs a random bit just as the challenger in Game 6. Otherwise, $\mathcal{B}$ computes Y and π as

$$Y := e\left(g^{\mathsf{R}_X(\alpha)/\tilde{w}_0}, h\right), \qquad \pi := \left(\pi_0 = g^{\mathsf{R}_X(\alpha)/\tilde{w}_0}, \left(\pi_j := g^{\mathsf{P}_{X_i,j}(\alpha)}\right)_{j\in\mathcal{C}}\right).$$

Note that $\mathcal{B}$ can compute these values because all $\mathsf{P}_{X_i,j}(\mathsf{Z})$ and $\mathsf{R}_{X_i}(\mathsf{Z})$ have degree at most $2^d \leq q$ and therefore all group elements g^{x^i} for $i \leq 2^d$ are part of the q-DBDHI instance.

When $\mathcal{A}$ submits its challenge X^*, $\mathcal{B}$ computes the coefficients of $\mathsf{R}_{X^*}(\mathsf{Z})$ and $\mathsf{P}_{X^*,j}(\mathsf{Z})$ for all $j \in \mathcal{C}$ as above. As the challenger in Game 6, $\mathcal{B}$ aborts and outputs a random bit if the coefficient of degree zero of $\mathsf{P}_{X^*,\mathsf{out}}(\mathsf{Z})$ is not 1, *i.e.*, if $\mathsf{G}(X^*, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}}) = 0$. Otherwise, $\mathcal{B}$ sets

$$Y^* := \left(T \cdot e\left(g^{\mathsf{R}_{X^*}(\alpha)}, h\right)\right)^{1/\tilde{w}_0}$$

and passes it on to $\mathcal{A}$. In order to conclude the proof, observe that both π and Y are distributed exactly as in Game 6 for all queries $\mathcal{A}$ makes to Eval . Further, if $T = e(g, h)^{1/\alpha}$, which is the case with probability $1/2$, then Y^* is distributed exactly as if $b = 0$ in $G_{\mathcal{A}}^{\mathcal{VRF}}(\lambda)$. Analogously, if T is a uniformly random element from $\mathbb{G}_T$, then Y^* is also a uniformly random element in $\mathbb{G}_T$, *i.e.*, as if $b = 1$ in $G_{\mathcal{A}}^{\mathcal{VRF}}(\lambda)$. We therefore conclude that $\Pr\left[G_{\mathcal{B}}^{q\text{-}\mathsf{DBDHI}}(\lambda) = 1\right] = \Pr\left[E_6\right]$. Furthermore, observe that $t_{\mathcal{A}} \approx t_{\mathcal{B}}$ because $t_{\mathcal{A}}$ already includes the runtime of the security experiment and $\mathcal{B}$ does nothing more than executing the security experiment for $\mathcal{B}$ with the sole difference that it has to compute the coefficients of the polynomials $\mathsf{P}_{X,j}(\mathsf{Z})$ and $\mathsf{R}_X(\mathsf{Z})$. However, these few additional operations do not make a significant difference in the overall runtime of $\mathcal{B}$.

5 Conclusion

We have settled the question: What is the optimal tightness an adaptively secure VRF can achieve? We did so by showing that every reduction from a non-interactive complexity assumption that can sequentially rewind the adversary a constant number of times necessarily loses a factor of $\approx Q$. Further, we constructed the first VRF with a reduction that has this optimal tightness. The takeaway message is that the optimal loss for adaptively secure VRFs is Q and that it is possible to construct VRFs that attain this bound.

Our main technical contributions are:

1. The extension of the lower bound for the loss of reductions by Bader *et al.* [BJLS16] to VRFs and VUFs in Section 2.
2. Further, we presented a new partitioning strategy that achieves this optimal tightness even in the context of decisional security notions and complexity assumptions.
3. Finally, we show that this partitioning strategy can be applied in Yamada's VRF and thus yields a VRF in the standard model with optimal tightness. This also shows that the lower bound on the loss of reductions from a non-interactive complexity assumption to the security of a VRF that we present is optimal.

However, there are still some open questions. The technique of Bader *et al.*, and therefore also our results, only applies to non-interactive complexity assumptions and reductions that sequentially rewind adversaries. While this result covers already a large class of assumptions and reductions, it does not cover interactive assumptions and reductions that can run several instances of the adversary in parallel. Morgan and Pass show a lower bound of $\sqrt{Q}$ for the loss of reductions to the unforgeability of unique signatures from interactive assumptions [MP18]. It seems plausible that their technique could be extended to also cover VRFs and VUFs.

Another open question is whether there are VRFs with an optimally tight reduction that have key and proof sizes comparable to constructions with non-optimal tightness (see *e.g.* [Koh19] or [Kat17] for recent comparisons). Furthermore, the q-DBDHI assumption with a polynomial q is not a standard assumption and gets stronger with q [Che10]. It would therefore be preferable to construct an efficient VRF with optimal tightness from a standard assumption, like the VRFs in [HJ16, Koh19, Ros18].

Acknowledgments. I would like to thank Yuval Ishai for the helpful discussion. Further, I would like to thank my advisor Tibor Jager for his support and helpful feedback.

References

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

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 572–590. Springer, Heidelberg, April 2012.

[AGO11] Masayuki Abe, Jens Groth, and Miyako Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 628–646. Springer, Heidelberg, December 2011.

[AMNR18] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, and Ling Ren. Dfinity consensus, explored. Cryptology ePrint Archive, Report 2018/1153, 2018. `https://eprint.iacr.org/2018/1153`.

[ASM07] Man Ho Au, Willy Susilo, and Yi Mu. Practical compact e-cash. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *ACISP 07*, volume 4586 of *LNCS*, pages 431–445. Springer, Heidelberg, July 2007.

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, Heidelberg, May 2004.

[BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Heidelberg, August 2004.

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

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

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 2012*, pages 784–796. ACM Press, October 2012.

[Bit20] Nir Bitansky. Verifiable random functions from non-interactive witness-indistinguishable proofs. *Journal of Cryptology*, 33(2):459–493, April 2020.

[BJLS16] Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 273–304. Springer, Heidelberg, May 2016.

[BKKP15] Olivier Blazy, Saqib A. Kakvi, Eike Kiltz, and Jiaxin Pan. Tightly-secure signatures from chameleon hash functions. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 256–279. Springer, Heidelberg, March / April 2015.

[BL16] Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and id-based encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 404–434. Springer, Heidelberg, December 2016.

[BMR] Dan Boneh, Hart William Montgomery, and Ananth Raghunathan. Algebraic pseudorandom functions with improved efficiency from the augmented cascade. `https://crypto.stanford.edu/~dabo/pubs/papers/algebprf.pdf`. Accessed: 2020-11-12.

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

[BR09a] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for waters' IBE scheme. Cryptology ePrint Archive, Report 2009/084, 2009. `http://eprint.iacr.org/2009/084`.

[BR09b] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, Heidelberg, April 2009.

[CD96] Ronald Cramer and Ivan Damgård. New generation of secure and practical RSA-based signatures. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 173–185. Springer, Heidelberg, August 1996.

[Che10] Jung Hee Cheon. Discrete logarithm problems with auxiliary inputs. *Journal of Cryptology*, 23(3):457–476, July 2010.

[Cor02] Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 272–287. Springer, Heidelberg, April / May 2002.

[CW13] Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, Heidelberg, August 2013.

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

[DKN$^+$20] Alex Davidson, Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Adaptively secure constrained pseudorandom functions in the standard model. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 559–589. Springer, Heidelberg, August 2020.

[GCS$^+$17] Fuchun Guo, Rongmao Chen, Willy Susilo, Jianchang Lai, Guomin Yang, and Yi Mu. Optimal security reductions for unique signatures: Bypassing impossibilities with a counterexample. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 517–547. Springer, Heidelberg, August 2017.

[GHM$^+$17] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, pages 51–68. ACM, 2017.

[GNP$^+$15] Sharon Goldberg, Moni Naor, Dimitrios Papadopoulos, Leonid Reyzin, Sachin Vasant, and Asaf Ziv. NSEC5: Provably preventing DNSSEC zone enumeration. In *NDSS 2015*. The Internet Society, February 2015.

[GRPV20] Sharon Goldberg, Leonid Reyzin, Dimitrios Papadopoulos, and Jan Včelák. Verifiable Random Functions (VRFs). Internet-Draft draft-irtf-cfrg-vrf-07, Internet Engineering Task Force, June 2020. Work in Progress.

[HJ12] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Heidelberg, August 2012.

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

[HJK12] Dennis Hofheinz, Tibor Jager, and Edward Knapp. Waters signatures with optimal security reduction. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 66–83. Springer, Heidelberg, May 2012.

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

[IKOS08] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 433–442. ACM Press, May 2008.

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

[JKN21] Tibor Jager, Rafael Kurek, and David Niehues. Efficient adaptively-secure ib-kems and vrfs via near-collision resistance. Cryptology ePrint Archive, Report 2021/160, 2021. `https://eprint.iacr.org/2021/160`.

[JKP18] Tibor Jager, Rafael Kurek, and Jiaxin Pan. Simple and more efficient PRFs with tight security from LWE and matrix-DDH. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 490–518. Springer, Heidelberg, December 2018.

[JN19] Tibor Jager and David Niehues. On the real-world instantiability of admissible hash functions and efficient verifiable random functions. In Kenneth G. Paterson and Douglas Stebila, editors, *SAC 2019*, volume 11959 of *LNCS*, pages 303–332. Springer, Heidelberg, August 2019.

[JS04] Stanislaw Jarecki and Vitaly Shmatikov. Handcuffing big brother: an abuse-resilient transaction escrow scheme. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 590–608. Springer, Heidelberg, May 2004.

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

[KK12] Saqib A. Kakvi and Eike Kiltz. Optimal security proofs for full domain hash, revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 537–553. Springer, Heidelberg, April 2012.

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

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM CCS 2003*, pages 155–164. ACM Press, October 2003.

[Lis05] Moses Liskov. Updatable zero-knowledge databases. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 174–198. Springer, Heidelberg, December 2005.

[LW14] Allison B. Lewko and Brent Waters. Why proving HIBE systems secure is difficult. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 58–76. Springer, Heidelberg, May 2014.

[MBB$^+$15] Marcela S. Melara, Aaron Blankstein, Joseph Bonneau, Edward W. Felten, and Michael J. Freedman. CONIKS: Bringing key transparency to end users. In Jaeyeon Jung and Thorsten Holz, editors, *USENIX Security 2015*, pages 383–398. USENIX Association, August 2015.

[MP18] Andrew Morgan and Rafael Pass. On the security loss of unique signatures. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 507–536. Springer, Heidelberg, November 2018.

[MPS20] Andrew Morgan, Rafael Pass, and Elaine Shi. On the adaptive security of MACs and PRFs. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 724–753. Springer, Heidelberg, December 2020.

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

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

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

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

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

[Sch11] Sven Schäge. Tight proofs for signature schemes without random oracles. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 189–206. Springer, Heidelberg, May 2011.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. `http://eprint.iacr.org/2004/332`.

[VGP+18] Jan Včelák, Sharon Goldberg, Dimitrios Papadopoulos, Shumon Huque, and David C. Lawrence. NSEC5, DNSSEC Authenticated Denial of Existence. Internet-Draft draft-vcelak-nsec5-08, Internet Engineering Task Force, December 2018. Work in Progress.

[Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Heidelberg, May 2005.

[Yam17a] Shota Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. Cryptology ePrint Archive, Report 2017/096, 2017. `http://eprint.iacr.org/2017/096`.

[Yam17b] Shota Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 161–193. Springer, Heidelberg, August 2017.

A Proof of Lemma 2

Our proof closely follows Yamada's proof in [Yam17a, Appendix C]. For all $j \in \mathcal{P} \cup \mathcal{S} \cup \mathcal{C}$ let

$$b_j := \begin{cases} X_j & \text{if } j \in \mathcal{P} \\ \mathsf{K}^{\mathsf{PRF}}_{j-|\mathcal{P}|} & \text{if } j \in \mathcal{S}^{\mathsf{PRF}} \\ \mathsf{K}^{\mathsf{part}}_{j-|\mathcal{P}|-|\mathcal{S}^{\mathsf{PRF}}|} & \text{if } j \in \mathcal{S}^{\mathsf{part}} \\ 1 - b_{\mathsf{in}^1_\lambda(j)} \cdot b_{\mathsf{in}^2_\lambda(j)} & \text{if } j \in \mathcal{C}. \end{cases}$$

Note that for two bits $a, b \in \{0,1\}$, it holds that $1 - ab = a \,\mathsf{NAND}\, b$. We therefore have for all $j \in \mathcal{C}$ that b_j is the output of gate j and in particular, that $b_{\mathsf{out}} = \mathsf{G}(X, \mathsf{K}^{\mathsf{PRF}}, \mathsf{K}^{\mathsf{part}})$. We now claim that for all $j \in \mathcal{C}$ there exist $\mathsf{R}_{X,j}(\mathsf{Z}) \in \mathbb{Z}_p[\mathsf{Z}]$ such that

$$\mathsf{P}_{X,j}(\mathsf{Z}) = b_j + \mathsf{Z} \cdot \mathsf{R}_{X,j}(\mathsf{Z}).$$

Furthermore, it holds for all $j \in \mathcal{P} \cup \mathcal{S} \cup \mathcal{C}$, that if $d_j \in \mathbb{N}$ is the depth of j, then $\mathsf{deg}(\mathsf{P}_{X,j}(\mathsf{Z})) \leq 2^{d_j}$. We prove this by induction. For all $j \in \mathcal{P} \cup \mathcal{S}$ this holds by the definition of $\mathsf{P}_{X,j}(\mathsf{Z})$. For all $j \in \mathcal{C}$, let $j_1 := \mathsf{in}^1_\lambda(j)$ and $j_2 := \mathsf{in}^2_\lambda(j)$. Note that by our requirements for in^1_λ and in^2_λ we have that $j_1 < j$ and $j_2 < j$, which allows us to prove the statement by induction. We then prove our claim as follows.

$$\begin{aligned}
\mathsf{P}_{X,j}(\mathsf{Z}) &= 1 - \mathsf{P}_{X,j_1}(\mathsf{Z})\mathsf{P}_{X,j_2}(\mathsf{Z}) \\
&= 1 - (b_{j_1} + \mathsf{Z} \cdot \mathsf{R}_{X,j_1}(\mathsf{Z}))(b_{j_2} + \mathsf{Z} \cdot \mathsf{R}_{X,j_2}(\mathsf{Z})) \qquad (4) \\
&= 1 - b_{j_1} b_{j_2} + \mathsf{Z} \cdot \underbrace{(-b_{j_1}\mathsf{R}_{X,j_2}(\mathsf{Z}) - b_{j_2}\mathsf{R}_{X,j_1}(\mathsf{Z}) + \mathsf{Z}\mathsf{R}_{X,j_1}(\mathsf{Z})\mathsf{R}_{X,j_2}(\mathsf{Z}))}_{:=\mathsf{R}_{X,j}(\mathsf{Z})} \\
&= b_j + \mathsf{Z} \cdot \mathsf{R}_{X,j}(\mathsf{Z})
\end{aligned}$$

Note that Equation (4) holds because we have by induction that $\mathsf{P}_{X,j_1}(\mathsf{Z}) = b_{j_1} + \mathsf{Z} \cdot \mathsf{R}_{X,j_1}(\mathsf{Z})$ and $\mathsf{P}_{X,j_2}(\mathsf{Z}) = b_{j_2} + \mathsf{Z} \cdot \mathsf{R}_{X,j_2}(\mathsf{Z})$ holds.

Moreover, notice that for d_j, the depth of the gate with index j, it holds that $d_j = 1 + \max\{d_{j_1}, d_{j_2}\}$, where d_{j_1} and d_{j_2} are the depths of the gates with index j_1 and j_2 respectively. We then have that

$$\begin{aligned}
\mathsf{deg}(\mathsf{P}_{X,j}(\mathsf{Z})) = \mathsf{deg}(1 - \mathsf{P}_{X,j_1}(\mathsf{Z})\mathsf{P}_{X,j_2}(\mathsf{Z})) &= \mathsf{deg}(\mathsf{P}_{X,j_1}(\mathsf{Z})) + \mathsf{deg}(\mathsf{P}_{X,j_2}(\mathsf{Z})) \\
= 2^{d_{j_1}} + 2^{d_{j_2}} &\leq 2 \cdot 2^{\max\{d_{j_1}, d_{j_2}\}} = 2^{d_j}
\end{aligned}$$

This completes the proof of Lemma 2.