Verifiable Random Functions from Non-Interactive Witness-Indistinguishable Proofs

Nir Bitansky*

9 January 2017

Abstract

Verifiable random functions (VRFs) are pseudorandom functions where the owner of the seed, in addition to computing the function's value y at any point x, can also generate a non-interactive proof π that y is correct, without compromising pseudorandomness at other points. Being a natural primitive with a wide range of applications, considerable efforts have been directed towards the construction of such VRFs. While these efforts have resulted in a variety of algebraic constructions (from bilinear maps or the RSA problem), the relation between VRFs and other general primitives is still not well understood.

We present new constructions of VRFs from general primitives, the main one being *non-interactive witness-indistinguishable proofs* (NIWIs). This includes:

- A selectively-secure VRF assuming NIWIs and non-interactive commitments. As usual, the VRF can be made adaptively-secure assuming subexponential hardness of the underlying primitives.
- An adaptively-secure VRF assuming (polynomially-hard) NIWIs, non-interactive commitments, and *(single-key) constrained pseudorandom functions* for a restricted class of constraints.

The above primitives can be instantiated under various standard assumptions, which yields corresponding VRF instantiations, under different assumptions than were known so far. One notable example is a non-uniform construction of VRFs from subexponentially-hard trapdoor permutations, or more generally, from *verifiable pseudorandom generators* (the construction can be made uniform under a standard derandomization assumption). This partially answers an open question by Dwork and Naor (FOCS '00).

The construction and its analysis are quite simple. Both draw from ideas commonly used in the context of *indistinguishability obfuscation*.

*MIT, nirbitan@csail.mit.edu. Supported by NSF Grants CNS-1350619 and CNS-1414119, and the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226. Part of this research was done while visiting Tel Aviv University and supported by the Leona M. & Harry B. Helmsley Charitable Trust and Check Point Institute for Information Security.

Contents

1 Introduction

Verifiable random functions (VRFs), introduced by Micali, Rabin, and Vadhan [MRV99], are pseudorandom functions (PRFs) [GGM86] where it is possible to verify that a given output y corresponds to a correct evaluation of the function on any given input x. Such a VRF is associated with a secret key SK and a corresponding public verification key VK. The secret key allows anyone to compute the function $y = \mathsf{VRF.Eval}_{SK}(x)$ at any point x, and also to compute a proof $\pi_{x,y}$ that y was computed correctly. Here, by "computed correctly", we mean that any verification key VK^*, even a maliciously chosen one, uniquely determines the value y of the function at any point x, and accepting proofs only exist for this value y. The pseudorandomness requirement generalizes that of plain PRFs — the value y of the function at any point x should be pseudorandom, even after evaluating the function and obtaining proofs of correctness for an arbitrary polynomial number of points $\{x_i \neq x\}$. The standard definition is *adaptive*, allowing the point x to be chosen at any point, and we can also consider a *selective* definition, where the adversary chooses the challenge x, before getting the verification key VK, and before any evaluation query.

Constructions. VRFs are a natural primitive with a variety of applications (listed for instance in [ACF14]), and considerable effort has been invested in the pursuit of constructions, aiming to diversify and simplify the underlying assumptions [MRV99, Lys02, Dod03, BB04, DY05, ACF14, HW10, BMR10, CRV14, Fuc14, Jag15, HJ16]. Despite the progress made, almost all known constructions are of an algebraic nature, and are based directly either on the (strong) RSA assumption, or on different assumptions related to bilinear (or multilinear) maps. Attempts to construct VRFs from more general assumptions have been limited to constructions from *VRF-suitable identity-based encryption* [ACF14], or from indistinguishability obfuscation (IO) and injective one-way functions [SW14]. In both cases, concrete instantiations are, again, only known based on bilinear or multilinear maps.[1] Alternatively, *weak VRFs*, which are the verifiable analog of *weak PRFs* [NR99], can be constructed from (doubly enhanced) trapdoor permutations [BGRV09].

In terms of barriers, we know that VRFs imply [GO92] non-interactive zero-knowledge proofs (NIZKs) [BSMP91], and accordingly constructing VRFs from symmetric-key primitives like one-way functions, or collision-resistant hashing, seems out of reach for existing techniques. In contrast, NIZKs can be constructed from (doubly enhanced) trapdoor permutations (TDPs) [FLS99, BY96, GR13], and we may hope that so can VRFs. As possible evidence that this is a false hope, Fiore and Schröder show that there is no *black-box* reduction from VRFs to (doubly enhanced) TDPs [FS12].

1.1 This Work

We present new constructions of VRFs from general assumptions, the main one being *non-interactive witness-indistinguishable proofs* (NIWIs), which were introduced by Barak, Ong, and Vadhan [BOV07].

Our most basic result is a selectively-secure construction based on NIWIs, non-interactive commitments, and *puncturable PRFs* [BW13, BGI14, KPTZ13a, SW14] (these are in turn implied by one-way functions and thus also by non-interactive commitments). As usual, adaptive security of the construction can be shown assuming all primitives are subexponentially-secure.

Theorem 1.1 (informal)**.** *Assuming the existence of NIWIs and non-interactive commitments, there exist selectively-secure VRFs. Further assuming subexponential hardness of these primitives, there exist adaptively-secure VRFs.*

[1] The construction based on IO is also limited to either selective security, or reliance on subexponential hardness.

Aiming to avoid subexponential assumptions, our more general construction replaces puncturable PRFs with more general types of *single-key constrained PRF* (CPRFs) [BW13, BGI14, KPTZ13a]] and achieves adaptive security from polynomial assumptions.

Theorem 1.2 (informal)**.** *Assuming the existence of NIWIs, non-interactive commitments, and single-key CPRFs (for some restricted class of constraints), there exist adaptively-secure VRFs.*

Given the reliance on generic primitives, the above theorems already allow (and may further allow in the future) to base VRFs on different assumptions. We now review the (generic and specific) assumptions under which the above primitives are known, and derive corresponding corollaries. (For now, we focus on the implications of the theorems. We recall the definitions of NIWIs and CPRFs later, in the technical overview.)

NIWIs. Dwork and Naor [DN07] gave a non-uniform construction of NIWIs from NIZKs (which as noted before, can be constructed from doubly enhanced TDPs). Barak, Ong, and Vadhan [BOV07] showed that the construction can be made uniform assuming also the existence of a problem solvable in deterministic time $2^{O(n)}$ with non-deterministic circuit complexity $2^{\Omega(n)}$. The latter is a worst-case assumption previously used to derandomize **AM** [MV99], and can be seen as an extension of the assumption that $\mathbf{EXP} \not\subseteq \mathbf{NP}/\mathbf{poly}$ (see further discussion in [BOV07]). Groth, Ostrovsky, and Sahai [GOS12] then constructed NIWIs based on standard assumptions on bilinear maps such as the Decision Linear (DLIN) assumption, the Symmetric External Diffie Hellman (SXDH) assumption, or the Subgroup Decision Assumption. In [BP15], NIWIs are constructed from IO and one-way permutations.

Non-Interactive Commitments. Such commitments are known from any family of injective one-way functions [Blu81]. Naor [Nao91] gave a non-uniform construction from plain one-way functions, which can be made uniform under the same derandomization assumption mentioned above [BOV07].

CPRFs. Theorem 1.2 relies on single-key CPRFs for certain specific classes of constraints (see the technical outline below). It can be instantiated either by the CPRFs of Brakerski and Vaikuntanathan [BV15], based on LWE and 1D-SIS , or from those of Boneh and Zhandry, based on IO [BZ14]. We also give new instantiations under the DDH assumption.[2]

We can now combine the above in different ways to get instantiations of (adaptively-secure) VRFs from different assumptions, several of which were previously unknown. For example:

- A non-uniform construction from subexponential hardness of (doubly enhanced) TDPs. This should be contrasted with the black-box barrier of Fiore and Schróder mentioned above. The barrier does not apply to this construction both due to non-uniformity, and also non-black-box use of some of the underlying primitives, such as the commitments or puncturable PRFs.

- By instantiating these TDPs with a variant of the Rabin construction [GR13], we get a non-uniform construction from subexponential hardness of Factoring. This should be compared with the construction from subexponential hardness of strong RSA [MRV99]. (We can avoid subexponential hardness relying on DDH or LWE and 1D-SIS. We can further make the construction uniform under the above mentioned derandomization assumption.)

- We get constructions from various simple assumptions on bilinear groups, such as the DLIN assumption or the SXDH assumption, and the DDH assumption (in non-bilinear groups). This *does not* improve on the most recent construction of Hofheinz and Jager from DLIN [HJ16], but is a very different construction.

[2] We also give a simpler construction under the stronger d-power DDH assumption (which in turn can be reduced to the subgroup hiding assumption in composite DDH groups [CM14, HKW15]).

- We get a construction from polynomially hard IO and one-way permutations. In comparison, the existing construction mentioned above [SW14] required subexponential hardness for adaptive security.

An Equivalence between Nonuniform VRFs, VPRGs, and NIZKs. Dwork and Naor [DN07] defined a verifiable version of pseudo-random generators (VPRGs) and showed their equivalence to NIZKs. Such VPRGs (or NIZKs) are implied (even by selectively-secure) VRFs. Dwork and Naor raised the question of whether the converse holds: *do VPRGs imply VRF? (analogously to the fact that PRGs imply PRFs)* Our result shows that for non-uniform constructions this is indeed the case — VPRGs imply selectively-secure VRFs (or adaptively-secure if they are subexponentially-hard). For uniform constructions, we only establish this equivalence conditioned on the mentioned derandomization assumption.

1.2 Techniques

We now explain the main ideas behind our constructions.

A Naïve Idea: NIWIs instead of NIZKs. Our starting point is the simple construction of VRFs in the common random string model [MRV99] — to construct a VRF, let the verification key VK be a commitment $\mathsf{C} = \mathsf{Com}(\mathsf{F})$ to a function F drawn at random from a PRF family[GGM86], and store F along with the commitment randomness as the private evaluation key SK. The value of the function at any point x is simply $y = \mathsf{F}(x)$, and the proofs of correctness $\pi_{x,y}$ are simply NIZKs that y is consistent with the commitment C.

This solution works as expected, but requires a common random string. Aiming to get a construction in the plain model, a natural direction is to replace NIZKs with NIWIs, which exist in the plain model and still offer some level of privacy. Concretely, NIWIs guarantee absolute soundness (accepting proofs for false statements simply do not exist), and witness indistinguishability — a proof for a statement with multiple witnesses leaks no information about which witness was used in the proof; namely, proofs that use different witnesses are computationally indistinguishable. It is not hard see, however, that this relaxed privacy guarantee does not allow using NIWIs *as is* in the above solution. Indeed, since F is uniquely determined by the commitment C, a NIWI proof may very well leak it in full, without ever compromising witness indistinguishability.

Indeed, leveraging witness indistinguishability would require a different commitment mechanism that would not *completely* determine the underlying description of the function F. This may appear to conflict with the uniqueness requirement of VRFs that in the naïve construction was guaranteed exactly due to the fact that the commitment fixes the key. However, we observe that there is still have some wiggle room here — the uniqueness requirement of VRFs only requires that the *functionality* $\{\mathsf{F}(x)\}_x$ is fixed (rather than the actual description F of the function). Our solution will take advantage of this fact.

At high level, our first step will be to devise a mechanism such that on one hand, any verification key VK^* would completely determine the underlying function, but on the other hand, will guarantee that verification keys corresponding to two functionally equivalent circuits will remain indistinguishable, even given an arbitrary polynomial number of evaluations with proofs of consistency. This indistinguishability guarantee is reminiscent of the indistinguishability guarantee of IO, and indeed, our second step will rely on common IO techniques, such as *puncturing* [SW14], to leverage the established indistinguishability guarantee and prove the security of the VRF. More details follow.

Step 1: The Commit and Prove Mechanism. The implementation of this step is quite simple and is inspired by a construction of *verifiable IO* by Badrinarayanan et al. [BGJS16]. Instead of computing a single commitment C to F as the verification key VK, we compute and publish three independent commitments $(\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3)$ to the same PRF circuit F. The secret key consists of F and the randomness for the commitments.

Then, to prove the correctness of $y = \mathsf{F}(x)$, we give a NIWI that y is consistent with two out of the three commitments; namely, there exist $1 \leq i < j \leq 3$ such that $\mathsf{C}_i, \mathsf{C}_j$ are commitments to circuits $\mathsf{F}_i, \mathsf{F}_j$, and $y = \mathsf{F}_i(x) = \mathsf{F}_j(x)$. (This will, in fact, be the entire construction, only that we will require certain properties from the PRF family.)

By the binding of the commitments and the absolute soundness of the NIWI, this already guarantees that any verification key corresponds to at most a single function. The value of this function at any point is just the majority value of the functions underlying the commitments (for a malicious verification key, an absolute majority may not exist, in which case no value will be accepted). At the same time, we can show that verification keys and evaluations corresponding to two different circuits that compute the same function will be indistinguishable. This can be done by a hybrid argument where at any point, NIWI proofs only use as the witness the randomness and underlying plaintext for two of the three commitments. For example, at first, proofs will only use the randomness for C_1 and C_2, allowing to change the third commitment C_3 to a different circuit F'. Then, assuming that F' and F compute the same function, we can rely on the witness-indistinguishability guarantee, and now use instead the randomness for two different commitments, say C_1 and C_3 to compute any NIWI proof. Now we can change C_2 to F', and so on.

In fact, note that F and F' need not be completely functional equivalent, they only need to be equivalent on the points x, for which the adversary gets evaluations $(y, \pi_{x,y})$. In the body, we formalize this commit and proof mechanism by a notion we call *verifiable function commitment*.

Step 2: Proving Pseudorandomness (the Selective Case). The above guarantee suggests that we may be able to prove security if we could replace the committed PRF circuit F, with a circuit F' that is functionally equivalent to F on all of the adversary's evaluation queries x_i, and yet does not leak information on the function's value $\mathsf{F}(x)$ at the challenge point x. *Can we generate such a circuit* F'*?* In the case of a selective adversary (that announces the challenge x before even getting the verification key), we certainly can — via puncturable PRFs. Recall that in such PRFs, we can puncture the PRF circuit F at any point x, so that the new punctured circuit $\mathsf{F}'_{\{x\}}$ retains the functionality of F at any point other than x, whereas the value $\mathsf{F}(x)$ at the punctured point x remains pseudorandom.

Concretely, our security reduction will use any selective adversary against the VRF to break the pseudorandomness at the punctured point x. The reduction will generate a commitment to the punctured $\mathsf{F}'_{\{x\}}$, and use it to compute the answers (y_i, π_{x_i,y_i}), for all the queries $x_i \neq x$. As we have already argued, the adversary could not distinguish between this and the real VRF experiment where the unpunctured F would be used, as the two completely agree on all evaluations points x_i. Accordingly, any successful adversary in the VRF game can be used by the reduction to distinguish $\mathsf{F}(x)$ from a truly random output.

Adaptive Security via Constrained PRFs. As already mentioned, selective security implies adaptive security if we assume subexponential hardness — the reduction basically guesses the challenge, which incurs a $2^{|x|}$ security loss. To obtain adaptive security from polynomial assumptions, we follow a common path in adaptive security proofs, relying on the idea of *partitioning*. Roughly speaking, the idea is that instead of guessing at random the challenge (which is successful with exponentially-small probability), the reduction guesses a partition $(S, X \setminus S)$ of the query space X, aiming that with noticeable (rather than exponentially-small) probability, all evaluation queries x_i will fall outside S, but the challenge x will fall inside S.

In our case, given such a partition scheme, we aim to follow the same approach as above (for the selective case), only that now instead of creating a circuit $\mathsf{F}'_{\{x\}}$ that is punctured at a single point, we would like to create a circuit F'_S that is punctured at the entire set S; namely, it retains the functionality of F on any point in $X \setminus S$, but the value $\mathsf{F}(x)$ is pseudorandom for any $x \in S$. This more general notion is indeed known as constrained PRFs (CPRF). Here we only need *single-key* CPRFs in the sense that security holds in the presence of a single constrained PRF. Also, we do not need constraining for arbitrary sets S, but just for the sets S in the support of the partition scheme we use. We give three examples of such partition schemes, two that align with the common notion of *admissible hash functions* [BB04], and another one based on universal hashing [CW79]. As stated in the previous subsection, corresponding CPRFs exist under different (polynomial) assumptions. Overall, the construction is exactly the same as before only that we instantiate the PRF with a CPRF for constrained sets in the support of one of the above partition schemes.[3]

Fulfilling the above partitioning approach involves certain technical subtleties, most of which are common to typical partitioning proofs. One notorious issue concerns the fact that, while overall noticeable, the probability of successful partition may vary with how the adversary chooses its queries. In particular, it may potentially be the case that conditioned on a successful partition, the adversary's advantage in the VRF game becomes negligible (see more elaborate discussion in [Wat05]). There are several approaches for dealing with this in the literature (the most common one is perhaps the artificial abort technique in [Wat05]). We follow an approach suggested by Jager [Jag15] of requiring that the partition schemes in use are *balanced* in the sense that the probability of partition does not change by much over different choices of queries. See further details in Sections 2.6, 3.3.

1.3 Concurrent and Independent Work

In concurrent and independent work, Goyal, Hohenberger, Koppula, and Waters [GHKW17] present a similar approach for constructing VRFs. The general construction and underlying primitives are essentially the same as ours. There are some differences regarding the instantiations provided for the underlying primitives, presentation, and analysis. We summarize the symmetric difference below.

- **Underlying Primitives.** In terms of CPRF instantiations, apart from the instantiations common to both works, they give an instantiation based on the Phi-Hiding Assumption, and we give an instantiations based on the DDH assumption. They also give new instantiations for commitment schemes based on LWE and LPN, which we do not.

- **Presentation and Abstractions.** In the body, to make the proof a bit more modular, we chose to abstract the commit and prove mechanism in use, which we formalize by a notion of verifiable function commitments. Effectively, the same mechanism is present in both constructions. Also, for the sake of adaptive security, they rely on the standard notion of admissible hash functions, whereas we chose to consider a somewhat more general notion of *partition schemes*, with the aim of giving more flexibility when designing corresponding CPRFs (indeed, this allows us to get our DDH-based instantiation).

- **Analysis.** To prove adaptive security, they use the technique of *artificial aborts* [Wat05], whereas we instead use a slightly stronger notion of partition schemes (or admissible hash functions) that are also balanced [Jag15]. (The balance property does not require any additional assumptions and is quite simple to guarantee.)

[3] In the body, we further allow the partition scheme to involve some *encoding* of the input space X into a more structured input space $\widehat{X}$, and then consider applying the CPRF and partitioning for encoded inputs in the new space $\widehat{X}$. See Definition 2.6 and Section 3 for more details.

Organization

In Section 2, we define the primitives used in this work. In Section 3, we present our main construction and its analysis. In Section 4, we discuss possible instantiations, induced by different choices of partition schemes and corresponding CPRFs.

2 Preliminaries

We rely on the standard computational concepts:

- We follow the standard habit of modeling any efficient adversary strategy as a family of polynomial-size circuits. For an adversary $\mathcal{A}$ corresponding to a family of polynomial-size circuits $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, we often omit the subscript λ, when it is clear from the context.

- We say that a function $f : \mathbb{N} \to \mathbb{R}$ is negligible if for all constants $c > 0$, there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$. We will denote negligible functions by negl.

- If $\mathcal{X}^{(b)} = \{X_\lambda^{(b)}\}_{\lambda\in\mathbb{N}}$ for $b \in \{0,1\}$ are two ensembles of random variables indexed by $\lambda \in \mathbb{N}$, we say that $\mathcal{X}^{(0)}$ and $\mathcal{X}^{(1)}$ are computationally indistinguishable if for all polynomial-size distinguishers $\mathcal{D}$, there exists a negligible function negl such that for all λ,

  $$\left|\Pr[\mathcal{D}(X_\lambda^{(0)}) = 1] - \Pr[\mathcal{D}(X_\lambda^{(1)}) = 1]\right| \leq \mathrm{negl}(\lambda).$$

  We denote this by $\mathcal{X}^{(0)} \approx_c \mathcal{X}^{(1)}$.

2.1 Verifiable Random Functions

We define verifiable random functions (VRFs).

Definition 2.1 (VRF [MRV99])**.** *Let n, m, k be polynomially bounded functions. A verifiable random function* $\mathsf{VRF} = (\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.P}, \mathsf{VRF.V})$ *consists of the following polynomial-time algorithms:*

- *a probabilistic key sampler* $\mathsf{VRF.Gen}(1^\lambda)$ *that given a security parameter 1^λ outputs a secret key SK and public verification key $VK \in \{0,1\}^{k(\lambda)}$,*

- *an evaluator* $\mathsf{VRF.Eval}_{SK}(x)$ *that given the secret key and $x \in \{0,1\}^{n(\lambda)}$ outputs $y \in \{0,1\}^{m(\lambda)}$,*

- *a prover* $\mathsf{VRF.P}_{SK}(x)$ *that given x and the secret key produces a proof π that y is consistent with the verification key VK,*

- *and verifier* $\mathsf{VRF.V}_{VK}(\pi, x, y)$ *that verifies the proof.*

We make the following requirements:

1. *Completeness: For every security parameter $\lambda \in \mathbb{N}$ and input $x \in \{0,1\}^{n(\lambda)}$,*

$$\Pr\left[\mathsf{VRF.V}_{VK}(\pi, x, y) = 1 \;\middle|\; \begin{array}{l} (SK, VK) \leftarrow \mathsf{VRF.Gen}(1^\lambda) \\ y = \mathsf{VRF.Eval}_{SK}(x) \\ \pi \leftarrow \mathsf{VRF.P}_{SK}(x) \end{array}\right] = 1 \enspace .$$

2. *Uniqueness: For every security parameter $\lambda \in \mathbb{N}$, input $x \in \{0,1\}^{n(\lambda)}$, and arbitrary verification key $VK^* \in \{0,1\}^{k(\lambda)}$, there exists at most a single $y \in \{0,1\}^{m(\lambda)}$ for which there exists an accepting proof π. That is,*

$$\textit{if} \quad \mathsf{VRF.V}_{VK^*}(\pi_0, x, y_0) = \mathsf{VRF.V}_{VK^*}(\pi_1, x, y_1) = 1 \quad \textit{then} \quad y_0 = y_1 \ .$$

3. *Adaptive Indistinguishability: for any adversary $\mathcal{A}(1^\lambda)$, consider the following game $\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}$:*

   (a) *The VRF challenger samples $(SK, VK) \leftarrow \mathsf{VRF.Gen}(1^\lambda)$, and sends VK to $\mathcal{A}$.*

   (b) *$\mathcal{A}$ submits to a challenger* evaluation queries *$x_1, \ldots, x_Q$, and gets back from the challenger $(y_1, \pi_1), \ldots, (y_Q, \pi_Q)$, where $y_i = \mathsf{VRF.Eval}_{SK}(x_i)$, $\pi_i \leftarrow \mathsf{VRF.P}(x_i, SK)$.*

   (c) *At any point, including between evaluation queries, $\mathcal{A}$ may submit a challenge input $x_* \in \{0,1\}^{n(\lambda)}$. The challenger then sets $y_*^0 = \mathsf{VRF.Eval}_{SK}(x_*), y_*^1 \leftarrow \{0,1\}^{m(\lambda)}$, samples $b \leftarrow \{0,1\}$, and sends y_*^b to $\mathcal{A}$. (The adversary $\mathcal{A}$ may then make additional evaluation queries.)*

   (d) *At the end, $\mathcal{A}$ outputs a guess b'. The result of the game $\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}(\lambda)$ is 1 if $b' = b$, and 0 otherwise.*

   We say that $\mathcal{A}$ is **admissible** *if in the above game it is always the case that $x_* \notin \{x_i \mid i \in [Q]\}$. We require that any polynomial-size admissible adversary wins the game with negligible advantage:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{vrf}} := \left| \Pr\left[\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \operatorname{negl}(\lambda) \ .$$

   We say that the VRF satisfies Selective Indistinguishability (rather than adaptive) if $\mathcal{A}$ submits the challenge query x_* at the beginning of the game, before getting VK and making any evaluation query.

2.2 Non-Interactive Witness-Indistinguishable Proofs

We define non-interactive witness-indistinguishable proofs (NIWIs).

Definition 2.2 (NIWI [BOV07])**.** *A non-interactive witness-indistinguishable proof system* $\mathsf{NIWI} = (\mathsf{NIWI.P}, \mathsf{NIWI.V})$ *for an **NP** relation $\mathcal{R}_{\mathcal{L}}$ consists of two polynomial-time algorithms:*

- *a probabilistic prover $\mathsf{NIWI.P}(x, w, 1^\lambda)$ that given an instance x, witness w, and security parameter 1^λ, produces a proof π,*
- *and a deterministic $\mathsf{NIWI.V}(x, \pi)$ that verifies the proof.*

We make the following requirements:

1. *Completeness: for every $\lambda \in \mathbb{N}, (x, w) \in \mathcal{R}_{\mathcal{L}}$,*

$$\Pr_{\mathsf{NIWI.P}}[\mathsf{NIWI.V}(x, \pi) = 1 : \pi \leftarrow \mathsf{NIWI.P}(x, w, 1^\lambda)] = 1 \ .$$

2. *Soundness: for every $x \notin \mathcal{L}$ and $\pi \in \{0,1\}^*$:*

$$\mathsf{NIWI.V}(x, \pi) \neq 1 \ .$$

3. *Witness Indistinguishability: for any sequence* $\mathcal{I} = \left\{ (\lambda, x, w_0, w_1) : \begin{array}{c} \lambda \in \mathbb{N}, x \in \{0,1\}^{\mathrm{poly}(\lambda)}, \\ w_0, w_1 \in \mathcal{R}_{\mathcal{L}}(x) \end{array} \right\}$*:*

$$\left\{ \pi_0 : \pi_0 \leftarrow \mathsf{NIWI.P}(x, w_0, 1^\lambda) \right\}_{(\lambda,x,w_0,w_1)\in\mathcal{I}} \approx_c \left\{ \pi_1 : \pi_1 \leftarrow \mathsf{NIWI.P}(x, w_1, 1^\lambda) \right\}_{(\lambda,x,w_0,w_1)\in\mathcal{I}} .$$

Barak, Ong, and Vadhan [BOV07] constructed NIWIs based on NIZK and the worst-case assumption that there exists a problem solvable in deterministic time $2^{O(n)}$ with non-deterministic circuit complexity $2^{\Omega(n)}$ (or more generally the existence of hitting set generators that fool non-deterministic distinguishers). Groth, Ostrovsky, and Sahai [GOS12] then constructed NIWIs based on standard assumptions on bilinear maps such as the Decision Linear Assumption, the Symmetric External Diffie Hellman assumption, or the Subgroup Decision Assumption. Bitansky and Paneth [BP15] constructed NIWIs from indistinguishability obfuscation and one-way permutations.[4]

2.3 Non-Interactive Commitments

We define non-interactive commitments.

Definition 2.3 (Non-Interactive Commitment [Blu81])**.** *A non-interactive commitment scheme consists of a polynomial-time commitment algorithm* $\mathsf{Com}(x; r)$ *that given a message* $x \in \{0,1\}^*$ *and randomness* $r \in \{0,1\}^\lambda$ *outputs a commitment* C*. We make the following requirements:*

1. *Perfect Binding: For every security parameter* $\lambda \in \mathbb{N}$*, and string* $\mathsf{C} \in \{0,1\}^*$ *there exists at most a single* $x \in \{0,1\}^*$ *such that* Com *is a commitment to* x*:*

$$\forall \lambda \in \mathbb{N}, r_0, r_1 \in \{0,1\}^\lambda \quad \textit{if} \quad \mathsf{Com}(x_0; r_0) = \mathsf{Com}(x_1; r_1) \quad \textit{then} \quad x_0 = x_1 \ .$$

2. *Computational Hiding: for any sequence* $\mathcal{I} = \left\{ \lambda \in \mathbb{N}, x_0, x_1 \in \{0,1\}^{\mathrm{poly}(\lambda)} \right\}$*:*

$$\left\{ \mathsf{C}_0 : \begin{array}{c} r \leftarrow \{0,1\}^\lambda \\ \mathsf{C}_0 \leftarrow \mathsf{Com}(x_0; r) \end{array} \right\}_{(\lambda,x_0,x_1)\in\mathcal{I}} \approx_c \left\{ \mathsf{C}_1 : \begin{array}{c} r \leftarrow \{0,1\}^\lambda \\ \mathsf{C}_1 \leftarrow \mathsf{Com}(x_1; r) \end{array} \right\}_{(\lambda,x_0,x_1)\in\mathcal{I}} .$$

Non-interactive commitments can be constructed from any injective one-way function (or a certifiable collection thereof) [Blu81]. Barak, Ong, and Vadhan [BOV07] constructed such commitments based on plain one-way functions and the worst-case assumption that there exists a problem solvable in deterministic time $2^{O(n)}$ with non-deterministic circuit complexity $2^{\Omega(n)}$ (or more generally the existence of hitting set generators that fool non-deterministic distinguishers).

2.4 Sets with Efficient Representation

Throughout the paper, we consider collections of sets $\mathcal{S} = \{\mathcal{S}_\lambda\}_{\lambda\in\mathbb{N}}$. Where $\mathcal{S}_\lambda$ consists of sets S over some domain $\{0,1\}^{n(\lambda)}$. We always consider sets that have efficient representation.

Definition 2.4 (Efficient Representation of Sets)**.** $\mathcal{S} = \{\mathcal{S}_\lambda\}_{\lambda\in\mathbb{N}}$ *is a collection of sets with efficient representation if there is a polynomial* poly *such that any set* $S \in \mathcal{S}_\lambda$ *can be represented by a circuit* C_S *of size* $\mathrm{poly}(\lambda)$ *such that* $C_S(s) = 1$ *if* $s \in S$ *and* $C_S(s) = 0$ *otherwise. We further require that given* C_S*, it is possible to efficiently sample some* $s \in S$*.*

[4] In their construction, verification is probabilistic. Using their construction in our context would accordingly give a VRF with probabilistic verification. For simplicity, in this paper, we shall restrict attention to deterministic verification.

It will typically be convenient to abuse notation and identify any set S with its circuit representation C_S. In particular, when an algorithms gets as input a set S that may be super-polynomially large, we mean that it gets as input its efficient representation C_S.

2.5 Constrained Pseudo-Random Functions

We next define constrained pseudo-random functions (CPRFs).

Definition 2.5 (Constrained PRFs [BW13, BGI14, KPTZ13a]). *Let n, m, k be polynomially bounded functions. Let $\mathcal{S} = \left\{\mathcal{S}_\lambda \subseteq 2^{\{0,1\}^{n(\lambda)}}\right\}_{\lambda \in \mathbb{N}}$ be a collection of sets with efficient representation. A constrained PRF* $\mathsf{CPRF} = (\mathsf{CPRF.Gen}, \mathsf{CPRF.Eval}, \mathsf{CPRF.Cons})$ *for $\mathcal{S}$ consists of the following polynomial-time algorithms:*

- *a probabilistic key sampler* $\mathsf{CPRF.Gen}(1^\lambda)$ *that given a security parameter 1^λ outputs a key $K \in \{0,1\}^{k(\lambda)}$,*
- *an evaluator* $\mathsf{CPRF.Eval}_K(x)$ *that given as input the key K and $x \in \{0,1\}^{n(\lambda)}$ outputs $y \in \{0,1\}^{m(\lambda)}$,*
- *and a constraining algorithm that given as input the key K and a set $S \in \mathcal{S}_\lambda$, outputs a constrained key $K_S \in \{0,1\}^{k(\lambda)}$.*

We make the following requirements:

1. *Functionality: For every security parameter $\lambda \in \mathbb{N}$, set $S \in \mathcal{S}_\lambda$, and input $x \in \{0,1\}^{n(\lambda)} \setminus S$,*

$$\Pr\left[\mathsf{CPRF.Eval}_{K_S}(x) = \mathsf{CPRF.Eval}_K(x) \;\middle|\; \begin{array}{l} K \leftarrow \mathsf{CPRF.Gen}(1^\lambda) \\ K_S \leftarrow \mathsf{CPRF.Cons}(K, S) \end{array}\right] = 1 \enspace .$$

2. *(Single-Key) Indistinguishability: for any adversary $\mathcal{B}(1^\lambda)$, consider the following game $\mathcal{G}_{\mathcal{B}}^{\mathsf{cprf}}$:*

   (a) $\mathcal{B}$ submits a constraint S to a CPRF challenger.

   (b) The CPRF challenger samples $K \leftarrow \mathsf{CPRF.Gen}(1^\lambda)$, computes a constrained key $K_S \leftarrow \mathsf{CPRF.Cons}(K, S)$, and sends K_S to $\mathcal{B}$.

   (c) $\mathcal{B}$, given K_S, chooses a challenge input $x_* \in \{0,1\}^{n(\lambda)}$, and sends it to the challenger.

   (d) The challenger sets $\begin{array}{l} y_*^0 = \mathsf{CPRF.Eval}_K(x_*), \\ y_*^1 \leftarrow \{0,1\}^{m(\lambda)} \end{array}$, samples $b \leftarrow \{0,1\}$, and sends y_*^b to $\mathcal{B}$.

   (e) $\mathcal{B}$, given y_*^b, outputs a guess b'. The result of the game $\mathcal{G}_{\mathcal{B}}^{\mathsf{cprf}}(\lambda)$ is 1 if $b' = b$, and 0 otherwise.

   We say that $\mathcal{B}$ is **admissible** *if in the above game it is always the case that $S \in \mathcal{S}_\lambda$ and $x_* \in S$. We require that any polynomial-size admissible adversary wins the game with negligible advantage:*

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{cprf}} := \left|\Pr\left[\mathcal{G}_{\mathcal{B}}^{\mathsf{cprf}}(\lambda) = 1\right] - \frac{1}{2}\right| \leq \mathrm{negl}(\lambda) \enspace .$$

Remark 2.1 (Key Size). In the above definition, constrained keys and unconstrained keys have the same description size k. Furthermore, we have a single evaluation algorithm for both constrained and unconstrained keys. Both of these assumptions are without loss of generality and are just meant to simplify presentation in our construction.

Remark 2.2 (Computational Functionality). We can also consider a relaxed computational functionality requirement [BV15], which essentially says that inputs outside the constrained set S, on which functionality isn't preserved, may exist, but are hard to find. Formally,

1. *Computational Functionality: For any polynomial-size adversary $\mathcal{A}$, any $\lambda \in \mathbb{N}$, and any $S \in \mathcal{S}_\lambda$:*

$$\Pr\left[\begin{array}{c} x \notin S \\ \mathsf{CPRF.Eval}_{K_S}(x) \neq \mathsf{CPRF.Eval}_K(x) \end{array} \middle| \begin{array}{l} K \leftarrow \mathsf{CPRF.Gen}(1^\lambda) \\ K_S \leftarrow \mathsf{CPRF.Cons}(K,S) \\ x \leftarrow \mathcal{A}^{\mathsf{CPRF.Eval}_K(\cdot)}(K_S) \end{array}\right] \leq \operatorname{negl}(\lambda) \enspace .$$

2.6 Partition Schemes

We now define *partition schemes*, which generalize the concept of *admissible hash functions* [BB04] often used in the literature to prove in the context of adaptive security.

Such a scheme for a domain $\{0,1\}^n$ provides a way to efficiently encode any element $x \in \{0,1\}^n$ to an element $\widehat{x} = \mathsf{PAR.Enc}(x)$ in a new domain $\{0,1\}^{\widehat{n}}$. The new domain is associated with a partition sampler $\mathsf{PAR.Gen}$ that samples a partition $(S, \overline{S})$, where $\overline{S} = \{0,1\}^{\widehat{n}} \setminus S$. The main guarantee is that for any set of Q elements $X \subseteq \{0,1\}^n$ and any $x_* \notin X$, with high probability $\widehat{x}_* \in S$ and $\widehat{X} \subseteq \overline{S}$; namely, x_* and X are split by the partition. We shall further require that the scheme is balanced, roughly meaning that the probability that the above occurs does not change much between different choices of (X, x_*). This property was suggested in [Jag15] for admissible hash functions as an alternative to the artificial abort technique in partition-based proofs [Wat05], inspired by [BR09].

Definition 2.6 (Partition Schemes)**.** *Let $n, \widehat{n}$ be polynomially bounded functions, $\tau < 1$ an inverse-polynomial function, and $\mathcal{S} = \left\{\mathcal{S}_\lambda \subseteq 2^{\{0,1\}^{\widehat{n}(\lambda)}}\right\}_{\lambda \in \mathbb{N}}$ a collection of sets with efficient representation. A partition scheme $\mathsf{PAR} = (\mathsf{PAR.Enc}, \mathsf{PAR.Gen})$ parameterized by $(n, \widehat{n}, \tau, \mathcal{S})$ consists of the following polynomial-time algorithms*

- *a deterministic encoder $\mathsf{PAR.Enc}(x)$ that maps any $x \in \{0,1\}^{n(\lambda)}$ to $\widehat{x} \in \{0,1\}^{\widehat{n}(\lambda)}$*
- *a probabilistic sampler $\mathsf{PAR.Gen}(1^\lambda, Q, \delta)$ that given security parameter 1^λ, integer Q, and balance parameter δ, outputs a set $S \in \mathcal{S}_\lambda$, interpreted as a partition $(S, \overline{S})$ of $\{0,1\}^{\widehat{n}(\lambda)}$.*[5]

Fix $\lambda, Q \in \mathbb{N}, \delta < 1$. Let $\mathcal{X}$ be a distribution on pairs (X, x_*) such that $X := (x_1, \ldots, x_Q) \in \{0,1\}^{n(\lambda) \times Q}$ and $x_* \in \{0,1\}^{n(\lambda)} \setminus X$. We define the probability that (X, x_*) are split by the sampled partition:

$$P_{\mathcal{X}}(\lambda, Q, \delta) := \Pr\left[\widehat{x}_* \in S, \widehat{X} \subseteq \overline{S} \middle| \begin{array}{r} (X, x_*) \leftarrow \mathcal{X}, \\ \widehat{x}_* = \mathsf{PAR.Enc}(x_*), \\ \widehat{X} = \{\mathsf{PAR.Enc}(x_i) \mid x_i \in X\}, \\ S \leftarrow \mathsf{PAR.Gen}(1^\lambda, Q, \delta) \end{array}\right] \enspace .$$

For every $\lambda, Q \in \mathbb{N}, \delta < 1$, and any two distributions $\mathcal{X}, \mathcal{X}'$ as above, we require:

1. *Probable Partitioning:*

$$P_{\mathcal{X}}(\lambda, Q, \delta) \geq \tau(\lambda, Q, \delta^{-1}) = \left(\frac{\delta}{Q \cdot \lambda}\right)^{O(1)} \quad ,$$

[5] We note that the set S has efficient representation in terms of λ, and does not grow with Q, δ^{-1}. Indeed, throughout this paper, Q, δ^{-1}, will be arbitrary polynomials in λ that depend on the adversary. In our partition schemes, the representation of sets will only scale with $\min\{\log(Q/\delta), n(\lambda)\}$.

2. Balance:

$$1-\delta \leq \frac{P_{\mathcal{X}}(\lambda,Q,\delta)}{P_{\mathcal{X}'}(\lambda,Q,\delta)} \leq 1+\delta \ .$$

Remark 2.3 (Admissible Hash Functions). Admissible hash functions [BB04] are a special case of partition schemes where the partitions considered are of a specific kind — namely S is always the set of all strings that contain a certain substring (we call these *substring matching* in Section 4). For our construction, we may use other partition schemes as well (we give such an example in Section 4).

We also note that the balance requirement is inspired by the definition in [Jag15] for balanced admissible hash functions. There, the requirements of probable partition and balanced are unified to one requirement. We find that the above formulation captures the balance requirement in a somewhat more intuitive way.

3 The Construction

In this section, we present our VRF construction. For this purpose we first define and construct a primitive that we call verifiable function commitment, which captures the properties required from the commit and prove mechanism described in the introduction. We then use this primitive in conjunction with constrained PRFs to obtain our VRFs.

3.1 A Verifiable Function Commitment

We define verifiable function commitment schemes (VFCs). At high-level such a scheme has a similar syntax to that of a VRF, it allows to commit to a function and then verify its uniquely determined values. Security of such commitments says that commitments to two circuits C_0, C_1 remain indistinguishable, as long as the attacker only sees evaluations (with proofs) on inputs x such that $C_0(x) = C_1(x)$.

Definition 3.1 (Verifiable Function Commitment)**.** *Let n, m, k be polynomially bounded functions. A verifiable function commitment $\mathsf{VFC} = (\mathsf{VFC.Gen}, \mathsf{VFC.P}, \mathsf{VFC.V})$ consists of the following polynomial-time algorithms:*

- *a probabilistic key sampler $\mathsf{VFC.Gen}(1^\lambda, C)$ that given a security parameter 1^λ and a circuit $C : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ outputs a secret key SK and public verification key $VK \in \{0,1\}^{k(\lambda)}$,*
- *a prover $\mathsf{VFC.P}_{SK}(x)$ that given x and the secret key produces a proof π that $y = C(x)$ is consistent with the verification key VK,*
- *and verifier $\mathsf{VFC.V}_{VK}(\pi, x, y)$ that verifies the proof.*

We make the following requirements (the first two are analogous to those of a VRF):

1. *Completeness: For every security parameter $\lambda \in \mathbb{N}$, input $x \in \{0,1\}^{n(\lambda)}$, and circuit C,*

$$\Pr\left[\mathsf{VFC.V}_{VK}(\pi,x,y)=1 \;\middle|\; \begin{array}{l}(SK,VK)\leftarrow \mathsf{VFC.Gen}(1^\lambda,C)\\ y=C(x)\\ \pi\leftarrow \mathsf{VFC.P}_{SK}(x)\end{array}\right]=1 \ .$$

2. *Uniqueness: For every security parameter $\lambda \in \mathbb{N}$, input $x \in \{0,1\}^{n(\lambda)}$, and arbitrary verification key $VK^* \in \{0,1\}^{k(\lambda)}$, there exists at most a single $y \in \{0,1\}^{m(\lambda)}$ for which there exists an accepting proof π. That is,*

$$\textit{if} \quad \mathsf{VFC.V}_{VK^*}(\pi_0, x, y_0) = \mathsf{VFC.V}_{VK^*}(\pi_1, x, y_1) = 1 \quad \textit{then} \quad y_0 = y_1 \ .$$

3. *Indistinguishability: for any adversary $\mathcal{A}(1^\lambda)$, consider the following game $\mathcal{G}_{\mathcal{A}}^{\mathsf{vfc}}$:*

   (a) *$\mathcal{A}$ submits to the challenger two circuits C_0, C_1.*

   (b) *The challenger samples $b \leftarrow \{0,1\}$, $(SK, VK) \leftarrow \mathsf{VFC.Gen}(1^\lambda, C_b)$, and sends VK to $\mathcal{A}$.*

   (c) *$\mathcal{A}$ submits to a challenger* evaluation queries *$x_1, \dots, x_Q$, and gets back from the challenger $\pi_1, \dots, \pi_Q$, where $\pi_i \leftarrow \mathsf{VFC.P}(x_i, SK)$.*

   (d) *At the end, $\mathcal{A}$ outputs a guess b'. The result of the game $\mathcal{G}_{\mathcal{A}}^{\mathsf{vfc}}(\lambda)$ is 1 if $b' = b$, and 0 otherwise.*

   We say that $\mathcal{A}$ is **admissible** *if in the above game the circuits C_0, C_1 map $\{0,1\}^{n(\lambda}$ to $\{0,1\}^{m(\lambda)}$ are of the same size and $C_0(x_i) = C_1(x_i)$ for all $i \in [Q]$. We require that any polynomial-size admissible adversary wins the game with negligible advantage:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{vfc}} := \left| \Pr\left[\mathcal{G}_{\mathcal{A}}^{\mathsf{vfc}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \mathrm{negl}(\lambda) \ .$$

We now show how to construct such a VFC.

Ingredients:

- A non-interactive commitment Com.
- A non-interactive witness-indistinguishable proof system NIWI.

The Construction:

- The key sampler $\mathsf{VRF.Gen}(1^\lambda, C)$:
  - Compute three commitments $\{\mathsf{C}_i := \mathsf{Com}(C; r_i)\}_{i \in [3]}$, using randomness $r_i \leftarrow \{0,1\}^\lambda$.
  - Output the secret key $SK = (C, r_1, r_2)$ and public key $VK = (\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3)$.
- The prover $\mathsf{VRF.P}_{SK}(x)$:
  - Construct the statement $\Psi = \Psi(\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3, x, y)$ asserting that the answer y is consistent with the function value given by the majority of the commitments:

$$\exists((i, r_i, C_i), (j, r_j, C_j)) : \begin{array}{l} 1 \leq i < j \leq 3, \\ \mathsf{C}_i = \mathsf{Com}(C_i; r_i), \mathsf{C}_j = \mathsf{Com}(C_j; r_j), \\ y = C_i(x) = C_j(x) \end{array} .$$

  - Output a NIWI proof $\pi \leftarrow \mathsf{NIWI.P}(\Psi, (1, r_1, C), (2, r_2, C), 1^\lambda)$ for the statement Ψ, using the commitment randomness r_1, r_2 and the circuit C as the witness.
- The verifier $\mathsf{VRF.V}_{VK}(\pi, x, y)$:

- Construct Ψ as above.
- Run the NIWI verifier $\mathsf{NIWI.V}(\pi, \Psi)$ and output the same answer.

Completeness and Uniqueness. The completeness of the scheme follows readily from the completeness of the NIWI system. The uniqueness follows from the perfect binding of the commitment as well as the soundness of the NIWI. Indeed, given the verification key $VK = (\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3)$, binding implies that for each commitment C_i, there exists at most a single circuit C_i such that C_i is a valid commitment to C_i. Thus, also for any input x, each C_i is consistent with at most a single value $y_i = C_i(x)$. By the soundness of the NIWI, any accepted y must be consistent with the majority of value y_1, y_2, y_3.

Indistinguishability. We now prove the security of the scheme.

Proposition 3.1. *For any polynomial-size admissible adversary $\mathcal{A}$, it holds that* $\mathsf{Adv}^{\mathsf{vfc}}_{\mathcal{A}}(\lambda) \le \mathrm{negl}(\lambda)$.

Proof. To prove the claim we examine a sequence of hybrid CPRF games $\left\{\mathcal{G}^{\mathsf{cprf}}_{\alpha}\right\}$, each with a corresponding adversary $\mathcal{A}_\alpha$ and challenger $\mathcal{C}_\alpha$, which slightly augment the adversary and challenger of the previous hybrid. In all games, as in the original VFC game, the result of the game is 1 if and only if the adversary $\mathcal{A}_\alpha$ guesses correctly the challenge bit, i.e. $b' = b$.

Hybrid $\mathcal{G}^{\mathsf{vfc}}_0$: this corresponds to the game $\mathcal{G}^{\mathsf{vfc}}_{\mathcal{A}}$ described above. Namely $\mathcal{A}_0$ is the above described $\mathcal{A}$ and $\mathcal{C}_0$ is the usual VFC challenger.

Hybrid $\mathcal{G}^{\mathsf{vfc}}_1$: in this game, the VFC challenger $\mathcal{C}_1$ generates C_1 as a commitment to C_0 instead of C_b.

We claim that by the hiding of the commitment scheme, $\mathcal{A}_1$ is still admissible and

$$\left|\Pr\left[\mathcal{G}^{\mathsf{vfc}}_1(\lambda) = 1\right] - \Pr\left[\mathcal{G}^{\mathsf{vfc}}_0(\lambda) = 1\right]\right| \le \mathrm{negl}(\lambda) \ .$$

Indeed, all NIWI proofs, and thus the entire experiment, are independent of the randomness r_1 used for the commitment C_1, and only depend on the randomness r_2, r_3 for the commitments $\mathsf{C}_2, \mathsf{C}_3$. Thus, if the adversary becomes inadmissible (makes a query x_i such that $C_0(x_i) \ne C_1(x_i)$), or if there is any noticeable difference between the games, we directly get an efficient distinguisher that can break the hiding of the commitment scheme.

Hybrid $\mathcal{G}^{\mathsf{vfc}}_{2,j}$, $j \in \{0, \dots, Q\}$: in this game, for every $i \le j$, the proof π_i for the statement Ψ_i, computed by $\mathcal{A}_{2,j}$ for the i^{th} evaluation query, uses the witness $((1, C_0, r_1), (3, C_b, r_3))$, whereas for every $i > j$, it is computed using the witness $((2, C_b, r_2), (3, C_b, r_3))$.

First note that by definition,

$$\mathcal{G}^{\mathsf{vfc}}_{2,0}(\lambda) \equiv \mathcal{G}^{\mathsf{vfc}}_1(\lambda) \ .$$

Also, by the witness indistinguishability of the NIWI proof system, each $\mathcal{A}_{2,j}$ is still admissible and

$$\max_{j \in [Q]} \left|\Pr\left[\mathcal{G}^{\mathsf{vfc}}_{2,j-1}(\lambda) = 1\right] - \Pr\left[\mathcal{G}^{\mathsf{vfc}}_{2,j}(\lambda) = 1\right]\right| \le \mathrm{negl}(\lambda) \ .$$

Indeed, for the statement Ψ_j, we know by the fact that adversary $\mathcal{A}_{2,j}$ has the same view as $\mathcal{A}_{2,j-1}$ until the jth query, and $\mathcal{A}_{2,j-1}$ is admissible, that $C_0(x_j) = C_1(x_j)$. Thus, both $((1, C_0, r_1), (3, C_b, r_3))$ and $((2, C_b, r_2), (3, C_b, r_3))$ are valid witnesses for the statement Ψ_j. Therefore, if $\mathcal{A}_{2,j}$ us not admissible, or there is any noticeable difference between the games, we get an efficient distinguisher that can break the witness indistinguishability of the NIWI scheme.

Hybrid $\mathcal{G}_3^{\mathsf{vfc}}$: in this game, $\mathcal{A}_3$ computes C_2 as a commitment to C_0 instead of C_b. By the hiding of the commitment, $\mathcal{A}_3$ is admissible and

$$\left|\Pr\left[\mathcal{G}_{2,Q}^{\mathsf{vfc}}(\lambda)=1\right]-\Pr\left[\mathcal{G}_3^{\mathsf{vfc}}(\lambda)=1\right]\right|\leq \mathrm{negl}(\lambda)\ .$$

This is argued as in the transition from $\mathcal{G}_0^{\mathsf{vfc}}$ to $\mathcal{G}_1^{\mathsf{vfc}}$ where now we rely on the fact that NIWI proofs in $\mathcal{G}_{2,Q}^{\mathsf{vfc}}$ are independent of the randomness r_2 used for C_2, and only depend on the randomness r_1, r_3.

Hybrid $\mathcal{G}_{4,j}^{\mathsf{vfc}}$, $j \in \{0,\dots,Q\}$: in this game, for every $i \leq j$, the proof π_i for the statement Ψ_i, computed in the i^{th} evaluation query, uses the witness $((1,C_0,r_1),(2,C_0,r_2))$, whereas for every $i > j$, it is computed using the witness $((2,C_0,r_2),(3,C_b,r_3))$.

By definition,

$$\mathcal{G}_{4,0}^{\mathsf{vfc}}(\lambda)\equiv\mathcal{G}_3^{\mathsf{vfc}}(\lambda)\ .$$

Also, by the witness indistinguishability of the NIWI proof system, it holds that

$$\max_{j\in[Q]}\left|\Pr\left[\mathcal{G}_{4,j-1}^{\mathsf{vfc}}(\lambda)=1\right]-\Pr\left[\mathcal{G}_{4,j}^{\mathsf{vfc}}(\lambda)=1\right]\right|\leq \mathrm{negl}(\lambda)\ .$$

This is argued as in the transition from $\mathcal{G}_{2,j-1}^{\mathsf{vfc}}$ to $\mathcal{G}_{2,j-1}^{\mathsf{vfc}}$ where now we rely on the fact that both $((1,K,r_1),(2,K,r_2))$ and $((2,K,r_2),(3,K_S,r_3))$ are valid witnesses for the statement Ψ_j.

Hybrid $\mathcal{G}_5^{\mathsf{vfc}}$: in this game, $\mathcal{A}_5$ computes C_3 as a commitment to C_0 instead of C_b. By the hiding of the commitment scheme, $\mathcal{A}_5$ is admissible and

$$\left|\Pr\left[\mathcal{G}_{4,Q}^{\mathsf{vfc}}(\lambda)=1\right]-\Pr\left[\mathcal{G}_5^{\mathsf{vfc}}(\lambda)=1\right]\right|\leq \mathrm{negl}(\lambda)\ .$$

This is argued as in the transition from $\mathcal{G}_0^{\mathsf{vfc}}$ to $\mathcal{G}_1^{\mathsf{vfc}}$ (or $\mathcal{G}_2^{\mathsf{vfc}}$ to $\mathcal{G}_3^{\mathsf{vfc}}$) where now we rely on the fact that NIWI proofs in $\mathcal{G}_{4,Q}^{\mathsf{vfc}}$ are independent of the randomness r_3 used for C_3, and only depend on r_1, r_2.

It is left to note that in $\mathcal{G}_5^{\mathsf{vfc}}$, the view of $\mathcal{A}_5$ is completely independent of the bit b (all the commitments are to C_0), and thus

$$\Pr\left[\mathcal{G}_5^{\mathsf{vfc}}(\lambda)=1\right]=\frac{1}{2}\ .$$

□

3.2 The VRF

We now present the VRF construction based on verifiable function commitments and constrained pseudo-random functions. We first list the ingredients we rely on.

Ingredients:

- A partition scheme PAR parameterized by $(n,\widehat{n},\tau,\mathcal{S})$ for a collection of sets $\mathcal{S}=\{\mathcal{S}_\lambda\}_{\lambda\in\mathbb{N}}$ with efficient representation.
- A constrained pseudo-random function CPRF for the collection $\mathcal{S}$, mapping $\widehat{n}$ bits to m bits. (For simplicity, we assume perfect functionality. We later observe that the same construction works also given computational functionality.)
- A verifiable function commitment scheme VFC for circuits mapping $\widehat{n}$ bits to m bits.

The Construction:

- The key sampler $\mathsf{VRF.Gen}(1^\lambda)$:
  - Sample a CPRF key $K \leftarrow \mathsf{CPRF.Gen}(1^\lambda)$, and consider the circuit $C_K(\cdot) = \mathsf{CPRF.Eval}_K(\cdot)$.
  - Sample VFC keys $(\overline{SK}, \overline{VK}) \leftarrow \mathsf{VFC.Gen}(1^\lambda, C_K)$.
  - Output the secret key $SK = (K, \overline{SK})$ and public key $VK = \overline{VK}$.
- The evaluator $\mathsf{VRF.Eval}_{SK}(x)$:
  - Compute $\widehat{x} = \mathsf{PAR.Enc}(x)$.
  - Output $y := \mathsf{CPRF.Eval}_K(\widehat{x})$.
- The prover $\mathsf{VRF.P}_{SK}(x)$:
  - Output a VFC proof $\pi \leftarrow \mathsf{VFC.P}_{SK}(\widehat{x})$ for the consistency of $y = C_K(\widehat{x})$ with $\overline{VK}$.
- The verifier $\mathsf{VRF.V}_{VK}(\pi, x, y)$:
  - Run the VFC verifier $\mathsf{VFC.V}_{VK}(\pi, x, y)$ and output the same answer.

Completeness and Uniqueness. Completeness and uniqueness follow readily from those of the VFC.

3.3 Security Analysis

We now prove the security of the VRF constructed above. Concretely, given an admissible adversary $\mathcal{A}$ against the VRF, we construct an admissible adversary $\mathcal{B}$ against the underlying constrained PRF. Throughout, we assume that $\mathcal{A}$ makes (w.l.o.g exactly) $Q = Q(\lambda)$ evaluation queries in the VRF game, for some polynomially bounded $Q(\lambda)$, and denote its advantage $\mathsf{Adv}^{\mathsf{vrf}}_{\mathcal{A}}(\lambda)$ by $\delta = \delta(\lambda)$.

The CPRF adversary. Adversary $\mathcal{B}(1^\lambda)$ operates as follows:

1. Initializes a variable result $=$ `succ`.
2. Invokes $\mathsf{PAR.Gen}(1^\lambda, Q, \delta)$ to sample a partition set $S \in \mathcal{S}_\lambda$.
3. Submits S to the CPRF challenger as the constraint, and obtains a constrained key K_S.
4. It now emulates $\mathcal{A}$ in $\mathcal{G}^{\mathsf{vrf}}_{\mathcal{A}}$ as follows:
   (a) Computes the constrained evaluation circuit $C_{K_S}(\cdot) = \mathsf{CPRF.Eval}_{K_S}(\cdot)$, samples corresponding VFC keys $(\overline{SK}, \overline{VK}) \leftarrow \mathsf{VFC.Gen}(1^\lambda, C_{K_S})$, and sends $VK = \overline{VK}$ to $\mathcal{A}$.
   (b) When $\mathcal{A}$ makes an evaluation query $x_i \in \{0,1\}^n$, for $i \in [Q]$,
      i. $\mathcal{B}$ computes the encoding $\widehat{x}_i$ of x_i.
      ii. If $\widehat{x}_i \in S$, sets result $=$ `fail`, and jumps to the last step 4d.
      iii. Otherwise, computes $y_i = C_{K_S}(\widehat{x}_i)$, and a VFC proof $\pi_i \leftarrow \mathsf{VFC.P}_{SK}(\widehat{x}_i)$ that y_i is consistent with $\overline{VK}$. Sends (y_i, π_i) to $\mathcal{A}$.
   (c) When $\mathcal{A}$ makes the challenge query $x_* \in \{0,1\}^n$,
      i. As before, $\mathcal{B}$ computes the encoding $\widehat{x}_*$ of x_*.

ii. If $\widehat{x}_* \notin S$, sets result $=$ `fail`, and jumps to the last step 4d.

iii. Otherwise, submits $\widehat{x}_*$ to the CPRF challenger as the challenge query, obtains y_*^b, and sends it to $\mathcal{A}$ as the VRF challenge.

(d) At the end of the game, if result $=$ `fail`, $\mathcal{B}$ acts as follows:

i. If a challenge query $\widehat{x}_*$ has not yet been submitted to the CPRF challenger (due to a pre-challenge failure in step 4(b)ii or 4(c)ii), samples some $z \in S$ and submits it as the challenge. Disregards the challenger's answer.

ii. Outputs a random guess $b' \leftarrow \{0,1\}$.

If result $=$ `succ`, $\mathcal{B}$ obtains a guess b' from $\mathcal{A}$, and outputs b'.

Note that $\mathcal{B}$ is admissible by construction (it always respects the constraint S). We now show that the advantage of $\mathcal{B}$ in the CPRF game is as large as the advantage δ of $\mathcal{A}$ in the VRF game, up to some loss τ that depends on the partition scheme (the guaranteed partition probability).

Proposition 3.2. $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{cprf}}(\lambda) \geq \tau(\lambda, Q, \delta^{-1}) \cdot \frac{\delta}{2} - \mathrm{negl}(\lambda) \geq \left(\frac{\delta}{\lambda \cdot Q}\right)^{O(1)} \cdot \frac{\delta}{2} - \mathrm{negl}(\lambda)$.

Proof. To prove the claim we examine a sequence of hybrid CPRF games $\left\{\mathcal{G}_\alpha^{\mathsf{cprf}}\right\}$, each with a corresponding adversary $\mathcal{B}_\alpha$ and challenger $\mathcal{C}_\alpha$, which slightly augment the adversary and challenger of the previous hybrid. In all games, as in the original CPRF game, the result of the game is 1 if and only if the adversary $\mathcal{B}_\alpha$ guesses correctly the challenge bit, i.e. $b' = b$.

Hybrid $\mathcal{G}_0^{\mathsf{cprf}}$: this corresponds to the game $\mathcal{G}_{\mathcal{B}}^{\mathsf{cprf}}$ described above. Namely $\mathcal{B}_0$ is the above described $\mathcal{B}$ and $\mathcal{C}_0$ is the usual CPRF challenger.

Hybrid $\mathcal{G}_1^{\mathsf{cprf}}$: in this game, the CPRF challenger $\mathcal{C}_1$ also provides $\mathcal{B}_1$ with the unconstrained key K, and $\mathcal{B}_1$ generates the VFC keys $(\overline{SK}, \overline{VK}) \leftarrow \mathsf{VFC.Gen}(1^\lambda, C_K)$ corresponding to the circuit $C_K(\cdot) = \mathsf{CPRF.Eval}_K(\cdot)$ instead of the constrained circuit C_{K_S}.

We argue that by the indistinguishability of the VFC scheme

$$\left|\Pr\left[\mathcal{G}_1^{\mathsf{cprf}}(\lambda) = 1\right] - \Pr\left[\mathcal{G}_0^{\mathsf{cprf}}(\lambda) = 1\right]\right| \leq \mathrm{negl}(\lambda) \ .$$

Indeed, any noticeable difference between the games, leads to an efficient distinguisher $\mathcal{D}$ that can break the VFC scheme. The distinguisher $\mathcal{D}$ will submit to the VFC challenger the circuits $C_0 = C_{K_S}, C_1 = C_K$, and then will emulate $\mathcal{B}$ only that instead of generating $(\overline{SK}, \overline{VK})$ and the proofs π_i by itself, it will use the verification key $\overline{VK}$ and proofs π_i given by the VFC challenger. First, note that this always induces an admissible VFC adversary. Indeed, $\mathcal{B}$ only answers the queries x_i of $\mathcal{A}$ as long as they are such that $\widehat{x}_i \notin S$, meaning that $C_{K_S}(\widehat{x}_i) = C_K(\widehat{x}_i)$. It is left to note that when the challenge bit is b, the emulated $\mathcal{B}$ acts exactly as $\mathcal{B}_b$ in $\mathcal{G}_b^{\mathsf{cprf}}$.

Hybrid $\mathcal{G}_2^{\mathsf{cprf}}$: in this game, the adversary $\mathcal{B}_2$ and challenger $\mathcal{C}_2$ act differently given evaluation queries x_i, or the challenge query x_*, from the emulated $\mathcal{A}$. $\mathcal{B}_2$ does not check right away whether $\widehat{x}_i$, or $\widehat{x}_*$ are in S. Instead, first all evaluation queries are answered according to the unconstrained circuit C_K, and the challenge is also answered according to this circuit, or a random string, depending on the challenge bit b. Namely, this part exactly emulates the real VRF game $\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}$.

Having finished emulating $\mathcal{A}$ as above, and recording its output guess b', $\mathcal{B}_2$ now checks that for all evaluation queries x_i made $\widehat{x}_i \notin S$ and for the challenge query $\widehat{x}_* \in S$. If this is the case, it outputs the recorded b' (previously output by $\mathcal{A}$) as the guess. Otherwise, it outputs a random guess $b' \leftarrow \{0,1\}$.

We argue that

$$\Pr\left[\mathcal{G}_1^{\mathsf{cprf}}(\lambda) = 1\right] = \Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1\right] \quad .$$

Indeed, consider in either game the event bad that either $\widehat{x}_i \in S$ for some evaluation query by $\mathcal{A}$ or $\widehat{x}_* \notin S$ for the challenge query by $\mathcal{A}$. Then, until the first query that induces bad, the view of $\mathcal{A}$ in the two experiments is distributed exactly the same. This also implies that bad occurs in both experiments with exactly the same probability. Furthermore, if bad does occur, then from that point on, $\mathcal{A}$'s emulation is disregarded and the two experiments again have exactly the same output distribution, a random b'. The required equality follows.

The Advantage in $\mathcal{G}_2^{\mathsf{cprf}}$. To conclude the proof, we show that

$$\left|\Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1\right] - \frac{1}{2}\right| \geq \tau(\lambda, Q, \delta^{-1}) \cdot \frac{\delta}{2} \quad .$$

Let us denote by win the event that in $\mathcal{G}_2^{\mathsf{cprf}}$ the adversary $\mathcal{A}$ emulated in the first part correctly guesses the challenge bit b. We continue to denote by bad the event that either $\widehat{x}_i \in S$ for some evaluation query by $\mathcal{A}$ or $\widehat{x}_* \notin S$ for the challenge query by $\mathcal{A}$.

Then, we have that

$$\begin{aligned}
&\Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1\right] = \\
&\Pr\left[\mathsf{bad}\right] \cdot \Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1 \;\middle|\; \mathsf{bad}\right] + \Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1 \wedge \overline{\mathsf{bad}}\right] = \\
&\left(1 - \Pr\left[\overline{\mathsf{bad}}\right]\right) \cdot \frac{1}{2} + \Pr\left[\mathsf{win}\right] \cdot \Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1 \wedge \overline{\mathsf{bad}} \;\middle|\; \mathsf{win}\right] + \Pr\left[\overline{\mathsf{win}}\right] \cdot \Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1 \wedge \overline{\mathsf{bad}} \;\middle|\; \overline{\mathsf{win}}\right] = \\
&\left(1 - \Pr\left[\overline{\mathsf{bad}}\right]\right) \cdot \frac{1}{2} + \Pr\left[\mathsf{win}\right] \cdot \Pr\left[\overline{\mathsf{bad}} \;\middle|\; \mathsf{win}\right] \cdot \Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda) = 1 \;\middle|\; \mathsf{win} \wedge \overline{\mathsf{bad}}\right] + \Pr\left[\overline{\mathsf{win}}\right] \cdot 0 = \\
&\left(1 - \Pr\left[\overline{\mathsf{bad}}\right]\right) \cdot \frac{1}{2} + \Pr\left[\mathsf{win}\right] \cdot \Pr\left[\overline{\mathsf{bad}} \;\middle|\; \mathsf{win}\right] \cdot 1 = \\
&\frac{1}{2} + \Pr\left[\overline{\mathsf{bad}} \;\middle|\; \mathsf{win}\right] \left(\Pr\left[\mathsf{win}\right] - \frac{1}{2} \cdot \frac{\Pr\left[\overline{\mathsf{bad}}\right]}{\Pr\left[\overline{\mathsf{bad}} \;\middle|\; \mathsf{win}\right]}\right) \quad .
\end{aligned}$$

We next note that by the probable partition and balance properties of the underlying partition schemes:

$$\begin{gathered}
\Pr\left[\overline{\mathsf{bad}} \;\middle|\; \mathsf{win}\right] \geq \tau(Q, \lambda, \delta^{-1}) \ , \\
\frac{\Pr\left[\overline{\mathsf{bad}}\right]}{\Pr\left[\overline{\mathsf{bad}} \;\middle|\; \mathsf{win}\right]} \in [1 - \delta, 1 + \delta] \ .
\end{gathered}$$

Indeed, $\overline{\mathsf{bad}}$ is exactly the event of successful partition where $(X = \{x_1, \dots, x_q\}, x_*)$ are sampled according to $\mathcal{A}$'s queries in the VRF game. $\overline{\mathsf{bad}}|\mathsf{win}$ is the event of successful partition when (X, x_*) are sampled from a different distribution — the one induced by $\mathcal{A}$ in the VRF game, but conditioned on $\mathcal{A}$ winning.

In addition, since the view of the emulated $\mathcal{A}$ in $\mathcal{G}_2^{\mathsf{cprf}}$ is identical to its view in $\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}$, it holds that

$$\Pr\left[\mathsf{win}\right] = \Pr\left[\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}(\lambda) = 1\right] \quad .$$

It now follows that

$$
\begin{aligned}
&\left|\Pr\left[\mathcal{G}_2^{\mathsf{cprf}}(\lambda)=1\right]-\frac{1}{2}\right| = \\
&\Pr\left[\overline{\mathsf{bad}} \mid \mathsf{win}\right]\cdot\left|\Pr\left[\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}(\lambda)=1\right]-\frac{1}{2}\cdot\frac{\Pr\left[\overline{\mathsf{bad}}\right]}{\Pr\left[\overline{\mathsf{bad}} \mid \mathsf{win}\right]}\right| \geq \\
&\tau(\lambda,Q,\delta^{-1})\cdot\left(\left|\Pr\left[\mathcal{G}_{\mathcal{A}}^{\mathsf{vrf}}(\lambda)=1\right]-\frac{1}{2}\right|-\frac{1}{2}\cdot\left|\frac{\Pr\left[\overline{\mathsf{bad}}\right]}{\Pr\left[\overline{\mathsf{bad}} \mid \mathsf{win}\right]}-1\right|\right) \geq \\
&\tau(\lambda,Q,\delta^{-1})\cdot\left(\delta-\frac{\delta}{2}\right)=\tau(\lambda,Q,\delta^{-1})\cdot\frac{\delta}{2}\ .
\end{aligned}
$$

□

Extending the Proof for Constrained PRFs with Computational Functionality. We observe that the proof extends when relying on CPRFs with the relaxed notion of computational functionality (Remark 2.2). We first note that the place where we rely on the functionality of the CPRF is in the transition between $\mathcal{G}_0^{\mathsf{cprf}}$ to $\mathcal{G}_1^{\mathsf{cprf}}$. There, to argue that both C_K and C_{K_S} agree on any $\mathcal{A}$'s query x_i (thus making the VCF attacker admissible), we rely on the fact that for $x_i \notin S$, the two functions agree. In the case that (perfect) functionality of the CPRF, the required agreement is guaranteed.

To show that the analysis extends to the case of computational functionality, we will argue that in the above transition, the VCF distinguisher $\mathcal{D}$ considered still does not *violate functionality* — namely, it does not output any evaluation query $x_i \notin S$ such that $\mathsf{CPRF.Eval}_{K_S}(x_i) \neq \mathsf{CPRF.Eval}_K(x_i)$ — except with negligible probability. Concretely, if it outputs with non-negligible probability $x_i \notin S$ that violates functionality, we can construct from it an adversary that breaks the computational functionality of the CPRF.

First, we argue that if the VCF attacker $\mathcal{D}$ violates functionality with non-negligible probability when the VCF challenge bit b is chosen at random, then it also does so when we restrict $b = 0$; that is, when the VFC keys always correspond to $C_0 = C_{K_S}$. Indeed, until the point that $\mathcal{D}$ outputs x_i that violates functionality, the case that $b = 0$ and $b = 1$ are indistinguishable by the VFC guarantee; furthermore, the event that x_i violates functionality is efficiently testable.

We now observe that in the restricted VFC experiment where $b = 0$, can be perfectly emulated given only the constrained key K_S and oracle access to $\mathsf{CPRF.Eval}_K$ (needed to compute the answer to the challenge query). Thus, we can use $\mathcal{D}$ to break the computational functionality of the CPRF.

4 Instantiations

In this section, we discuss possible instantiations for the underlying partition scheme and constrained PRF. We consider both adaptive security and selective security. For adaptive security, we consider instantiations based on various polynomial assumptions (such as LWE and 1D-SIS, DDH, or IO), or instantiations based on sub-exponential one-way functions. For selective security, we can rely on polynomial one-way functions. (The assumptions mentioned above are those required for appropriate CPRFs. For the CPRFs themselves, we still need NIWIs and non-interactive commitments).

4.1 Adaptive Security from Polynomial Assumptions

To obtain adaptive security from polynomial assumptions, we next describe three partition schemes for three different collections of partition sets $\mathcal{S}$. We then exhibit the existence of CPRFs for these collections based on different assumptions.

4.1.1 Partition Schemes

We give three examples of partition schemes. The first is a code-based scheme that aligns with the common notion of (balanced) admissible hash functions from the literature. The second is a variant of the first to large alphabets (which will be useful later on for simplifying the assumptions behind CPRFs). The third is a simple scheme based on universal hashing [CW79].

Substring Matching over Binary Alphabet. We first describe an existing partition scheme considered first in [Lys02] for the collection substring matching sets, which aligns with the notion of admissible hash functions. The scheme was also shown to be balanced in [Jag15]. Given that our definition is slightly different than that in [Jag15], and for the sake of completeness, we describe the scheme and its analysis.

- The partition scheme's encoding function $\mathsf{PAR.Enc}(x)$ is any binary error correcting code with constant distance $c < 1$. Each element $x \in \{0,1\}^n$ is encoded by an element $\widehat{x} \in \{0,1\}^{\widehat{n}}$.
- The collection of sets $\mathcal{S}_\lambda$ that partitions $\{0,1\}^{\widehat{n(\lambda)}}$ consists of sets S_s parameterized by a string $s \in \{0,1,\star\}^{\widehat{n(\lambda)}}$ containing wildcard symbols $\star$. For an element $z \in \{0,1\}^{\widehat{n(\lambda)}}$, we say that $z \in S_s$ if every non-wildcard bit of s agrees with z; namely, if $s_i \neq \star$, then $s_i = z_i$. We call such a set S_s a *substring matching set*.
- The partition sampler $\mathsf{PAR.Gen}(1^\lambda, Q, \delta)$ works as follows:
  - Let $d := \log(2Q/\delta)/\log(\frac{1}{1-c})$.
  - Sample a random set of d indices $D \leftarrow \binom{[\widehat{n}]}{d}$.
  - For $i \in D$ sample $s_i \leftarrow \{0,1\}$ at random. For $i \notin D$ set $s_i = \star$.
  - Output S_s.

We will now prove probable partition and balance.

For $(X = (x_1, \dots, x_Q), x_*)$, and consistently with Definition 2.6, define:

$$P_{X,x_*}(\lambda, Q, \delta) := \Pr\left[\widehat{x}_* \in S, \widehat{X} \subseteq \overline{S} \;\middle|\; \begin{array}{r} \widehat{x}_* = \mathsf{PAR.Enc}(x_*), \\ \widehat{X} = \{\widehat{x}_i \mid x_i \in X\}, \\ S \leftarrow \mathsf{PAR.Gen}(1^\lambda, Q, \delta) \end{array}\right] .$$

Further define

$$\overline{P} = \max_{(X,x_*):x^* \notin X} P_{X,x_*}(\lambda, Q, \delta), \qquad \underline{P} = \min_{(X,x_*):x^* \notin X} P_{X,x_*}(\lambda, Q, \delta) .$$

First, note that for any fixed $(X = \{x_1, \dots, x_Q\}, x_*)$ and any $x_i \in X$, it holds that

$$\Pr_D[\widehat{x}_i|D = \widehat{x}_*|D] = \prod_{i \in [d]} \left(1 - \frac{cn + i - 1}{n}\right) \leq (1-c)^d .$$

Also, for any fixed D,

$$\Pr_{s|D \leftarrow \{0,1\}^d}[s|D = \widehat{x}_*|D] = 2^{-d} \enspace .$$

Combining the first fact, a union bound over all $x_i \in X$, and the second fact, we have

$$\underline{P} \geq 2^{-d}(1 - Q(1-c)^d) = 2^{-d}(1 - \delta/2) \geq (\delta/Q)^{O(1)} \enspace .$$

Thus, probable partitioning holds with $\tau(\lambda, Q, \delta^{-1}) = (\delta/Q)^{O(1)}$.

Furthermore, we know that

$$\overline{P} \leq \max_{x_*, D} \Pr_{s|D}[s|D = \widehat{x}_*|D] = 2^{-d} \enspace .$$

This in turn implies that

$$1 - \delta \leq 1 - \delta/2 \leq \underline{P}/\overline{P} \leq \overline{P}/\underline{P} \leq \frac{1}{1-\delta/2} \leq 1 + \delta \enspace .$$

Since for every two distributions $\mathcal{X}, \mathcal{X}'$ on pairs (X, x_*) it holds that

$$\underline{P}/\overline{P} \leq \frac{P_{\mathcal{X}}(\lambda, Q, \delta)}{P_{\mathcal{X}'}(\lambda, Q, \delta)} \leq \overline{P}/\underline{P} \enspace ,$$

the balance property follows.

Substring Matching over Polynomial Alphabet. We now describe a variant of the above that on one hand will have a polynomial alphabet, but on the other will allow dealing with d-symbol substrings for a *constant* d, which will be useful in the construction of corresponding CPRFs. We shall restrict attention to a relatively simple setting of parameters, which will be enough for our purpose. (Conceivably, setting the parameters more carefully may lead to more efficient constructions.)

- Let $\Sigma \supseteq \{0,1\}$ be an alphabet of size $\sigma = O(n^2)$. The partition scheme's encoding function $\mathsf{PAR.Enc}(x)$ is an efficient error correcting code mapping Σ^n to $\Sigma^m \cong \{0,1\}^{\widehat{n}}$ with distance $1 - \frac{1}{n}$. Each element $x \in \{0,1\}^n$ is encoded by an element $\widehat{x} \in \{0,1\}^{\widehat{n}}$. For example, we can take the Reed-Solomon code consisting of degree n polynomials over a field $\mathbb{F}_{2^k}$ of size $O(n^2)$ (so $\widehat{n} = m \times k$).
- The collection of sets $\mathcal{S}_\lambda$ that partitions $\Sigma^m \cong \{0,1\}^{\widehat{n}}$ consists of sets S_s parameterized by a string $(s \in \Sigma \cup \{\star\})^m$ containing wildcard symbols $\star$. For an element $z \in \Sigma^m$, we say that $z \in S_s$ if every non-wildcard symbol of s agrees with z; namely, if $s_i \neq \star$, then $s_i = z_i$. Again, we call such a set S_s a *substring matching set*.
- The partition sampler $\mathsf{PAR.Gen}(1^\lambda, Q, \delta)$ works as follows:
  - Let $d := \log(2Q/\delta)/\log(n)$. In particular, in our setting where both Q/δ and n are polynomial in λ, $d = O(1)$.
  - Sample a random set of d indices $D \leftarrow \binom{[m]}{d}$.
  - For $i \in D$ sample $s_i \leftarrow \Sigma$ at random. For $i \notin D$ set $s_i = \star$.
  - Output S_s.

We will now prove probable partition and balance.

As before, for $X = (x_1, \dots, x_Q), x_*$, we consider the partition probability P_{X,x_*}, and the maximal and minimal (over all $X, x_* \notin X$) partition probabilities $\overline{P}, \underline{P}$.

First, note that for any fixed $(X = \{x_1, \dots, x_Q\}, x_*)$ and any $x_i \in X$, it holds that

$$\Pr_D [\widehat{x}_i|D = \widehat{x}_*|D] = \prod_{i \in [d]} \left(1 - \frac{(1 - \frac{1}{n})m + i - 1}{m}\right) \leq n^{-d} \ .$$

Also, for any fixed D,

$$\Pr_{s|D \leftarrow \Sigma^d} [s|D = \widehat{x}_*|D] = \sigma^{-d} \ .$$

Combining the first fact, a union bound over all $x_i \in X$, and the second fact, we have

$$\underline{P} \geq \sigma^{-d}(1 - Q \cdot n^{-d}) = \sigma^{-d}(1 - \delta/2) = \Omega(n^{-2d}) \cdot (1 - \delta/2) \geq (\delta/Q)^{O(1)} \ .$$

Thus, probable partitioning holds with $\tau(\lambda, Q, \delta^{-1}) = (\delta/Q)^{O(1)}$.

Furthermore, we know that

$$\overline{P} \leq \max_{x_*, D} \Pr_{s|D} [s|D = \widehat{x}_*|D] = \sigma^{-d} \ .$$

As for the previous partition scheme, we have

$$1 - \delta \leq \underline{P}/\overline{P} \leq \overline{P}/\underline{P} \leq 1 + \delta \ ,$$

and the balance property follows.

Universal Hashing. We now describe a simple partition scheme based on universal hashing.

- The partition scheme's encoding function $\mathsf{PAR.Enc}(x)$ is the identity, namely $\widehat{x} = x$.
- Let $\mathcal{H}_{\lambda,T} = \left\{h : \{0,1\}^{n(\lambda)} \to [T]\right\}$ be family of universal hash functions. The collection of sets $\mathcal{S}_\lambda$ that partitions $\{0,1\}^{n(\lambda)}$ consists of sets $S_{T,h,i}$ parameterized by hash function $h \in \mathcal{H}_{\lambda,T}$ and integer (or bin) $i \subseteq [T]$. For an element $z \in \{0,1\}^{n(\lambda)}$, we say that $z \in S_{T,h,i}$ if $h(z) = i$. We call such a set $S_{T,h,i}$ a *universal hash set*.
- The partition sampler $\mathsf{PAR.Gen}(1^\lambda, Q, \delta)$ works as follows:
  - Let $T := 2Q/\delta$.
  - Sample a random hash $h \leftarrow \mathcal{H}_{\lambda,T}$ and bin $i \leftarrow [T]$.
  - Output $S_{T,h,i}$.

We will now prove probable partition and balance.

As before, for $X = (x_1, \dots, x_Q), x_*$, we consider the partition probability P_{X,x_*}, and the maximal and minimal (over all $X, x_* \notin X$) partition probabilities $\overline{P}, \underline{P}$.

First, note that by universality, for any fixed $(X = \{x_1, \dots, x_Q\}, x_*)$, it holds that

$$\Pr_h [\exists x_i \in X : h(x_i) = h(x_*)] \leq \sum_{i \in [Q]} \Pr_h [h(x_i) = h(x_*)] \leq Q \cdot T^{-1} \leq \delta/2 \ .$$

Also, for any fixed h,

$$\Pr_i [h(x_*) = i] = T^{-1} = \frac{\delta}{2Q} \ .$$

Thus, we have

$$\underline{P} \geq \frac{\delta}{2Q}(1 - \delta/2) \geq \delta/4Q \ ,$$

and probable partitioning holds with $\tau(\lambda, Q, \delta^{-1}) = \delta/4Q$.

Furthermore, we know that

$$\overline{P} \leq \max_{x_*, h} \Pr_i [h(x_*) = i] = \frac{\delta}{2Q} \ .$$

As for the previous partition schemes, we have

$$1 - \delta \leq \underline{P}/\overline{P} \leq \overline{P}/\underline{P} \leq 1 + \delta \ ,$$

and the balance property follows.

4.1.2 Constrained PRFs

We now discuss possible CPRF instantiations for the above collections.

Existing Constructions. We start by noting that CPRFs for all set collections with efficient representation, with computational functionality, are known based on the standard lattice assumptions — LWE and 1D-SIS [BV15]. We also note that such CPRFs with perfect correctness are known from indistinguishability obfuscation (IO) [BZ14].

In particular, we can rely on the above CPRFs with either one of the partition schemes presented above.

A Construction for Substring Matching Sets over Binary Alphabet. We now give a construction that can be used together with the first partition scheme for substring matching sets over binary alphabet. The construction is based on the d-power DDH assumption (for logarithmic d), which in turn can be reduced to the subgroup hiding assumption in composite DDH groups [CM14, HKW15]. Later on, we will show how to reduce the assumption to plain DDH, by generalizing this construction.

Assumption 4.1 (d-Power DDH)**.** *There exists a polynomial-time sampler $\mathcal{G}(1^\lambda)$ that outputs a group $\mathbb{G}$ and $g \in \mathbb{G}$, such that for any polynomial-size adversary $\mathcal{A}$, and any $d(\lambda) = O(\log \lambda)$,*

$$\mathsf{Adv}^{\mathsf{dpdh}}_{\mathcal{A}}(\lambda) := \left| \Pr\left[\mathcal{A}(\mathbb{G}, g, g^{\alpha}, \dots, g^{\alpha^{d-1}}, g^{\gamma_b}) = b \;\middle|\; \begin{array}{c} (\mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda) \\ \alpha, \beta \leftarrow \mathbb{Z}^*_{|\mathbb{G}|} \\ \gamma_0 = \alpha^d, \gamma_1 \leftarrow \beta \\ b \leftarrow \{0,1\} \end{array} \right] - \frac{1}{2} \right| \leq \mathrm{negl}(\lambda) \ .$$

We next describe the construction, which is inspired by the Naor-Reingold PRF [NR04] and a construction of adaptive puncturable PRFs from [HKW15] from indistinguishability obfuscation and d-Power DDH. The security notion considered in that work is stronger than the one considered in this work (Definition 2.5), where the constraining set is chosen ahead of time and not adaptively. In particular, it will not require indistinguishability obfuscation and will handle the collection of constraints $\mathcal{S}$ considered in this section.

For domain $\{0,1\}^{\widehat{n}}$, the function is defined as follows:

- Each (non-punctured) key K consists of $\widehat{n}$ pairs $\left(k_{i,b} \leftarrow \mathbb{Z}^*_{|\mathbb{G}|}\right)_{i \in [\widehat{n}], b \in \{0,1\}}$, as well as $(\mathbb{G}, g)$.

- The value of the function is given by $\mathsf{CPRF.Eval}_K(x) = g^{\prod_{i \in [\widehat{n}]} k_{i,x_i}}$.
- The constraining algorithm $\mathsf{CPRF.Cons}(K, s)$, given a key K and a string $s \in \{0, 1, \star\}^{\widehat{n}}$, with d non-wildcards at positions $D \subseteq [\widehat{n}]$, works as follows:
  - Samples $\alpha \leftarrow \mathbb{Z}^*_{\mathbb{G}}$.
  - Outputs a punctured key K_{S_s} consisting of $(s, \mathbb{G}, g, g^\alpha, \dots, g^{\alpha^{d-1}})$ and a new set $\left(k'_{i,b}\right)_{i,b}$, where

$$k'_{i,b} = \begin{cases} \alpha^{-1} \cdot k_{i,b} & i \in D, b = s_i \\ k_{i,b} & \text{otherwise} \end{cases} .$$

- To evaluate the function on $x \in \{0, 1\}^{\widehat{n}} \setminus S_s$ using the punctured key K_{S_s}:
  - Let d' be the number of indices $i \in D$ such that $x_i = s_i$ (note that $d' < d$ since $x \notin S_s$).
  - Output $\left(g^{\alpha^{d'}}\right)^{\prod_{i \in [\widehat{n}]} k'_{i,x_i}}$.

Functionality. By definition,

$$\mathsf{CPRF.Eval}_{K_{S_s}}(x) = \left(g^{\alpha^{d'}}\right)^{\prod_{i \in [\widehat{n}]} k'_{i,x_i}} = \left(g^{\alpha^{d'}}\right)^{\alpha^{-d'} \prod_{i \in [\widehat{n}]} k_{i,x_i}} = g^{\prod_{i \in [\widehat{n}]} k_{i,x_i}} = \mathsf{CPRF.Eval}_K(x) \ .$$

Indistinguishability. We now prove the indistinguishability property of the constructed CPRF. Given an (admissible) adversary $\mathcal{B}$ that breaks the indistinguishability of the CPRF, we construct and adversary $\mathcal{A}$ that breaks the d-Power DDH assumption with the same advantage.

The breaker $\mathcal{A}$. Given $(\mathbb{G}, g, g^\alpha, \dots, g^{\alpha^{d-1}}, g^{\gamma_b})$, the adversary $\mathcal{A}$ emulates $\mathcal{B}$ as follows:

1. When $\mathcal{B}$ submits $s \in \{0, 1, \star\}^{\widehat{n}}$ to the CPRF challenger, where s has d non-wildcard entries on an index set $D \subseteq [\widehat{n}]$, $\mathcal{A}$ samples $\left(k'_{i,b} \leftarrow \mathbb{Z}^*_{|G|}\right)_{i,b}$. It then sends $K_{S_s} := \left(s, \mathbb{G}, g, g^\alpha, \dots, g^{\alpha^{d-1}}, \left(k'_{i,b}\right)_{i,b}\right)$ to $\mathcal{B}$.
2. Then $\mathcal{B}$ gives $x \in S_s$ as the challenge query, $\mathcal{A}$ returns $g^{\gamma_b \prod_{i \in \widehat{n}} k'_{i,x_i}}$.
3. When $\mathcal{B}$ outputs a guess b', $\mathcal{A}$ outputs the same guess.

We observe that the view of the emulated $\mathcal{B}$ is identical to its view in the CPRF game, where the induced non-punctured key is given by

$$k_{i,b} = \begin{cases} \alpha \cdot k'_{i,b} & i \in D, b = s_i \\ k_{i,b} & \text{otherwise} \end{cases} .$$

When $\gamma_b = \alpha^d$, this corresponds to the case that the CPRF value is returned, and when $\gamma_b \leftarrow \mathbb{Z}^*_{|G|}$ is random, this corresponds to the case that a random element $g^\beta, \beta \leftarrow \mathbb{Z}^*_{|G|}$ is returned.[6]

It follows that

$$\mathsf{Adv}^{\mathsf{dpdh}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{cfprf}}_{\mathcal{B}}(\lambda) \ .$$

[6] The above distribution is not necessarily random over strings. In any natural instantiation of the group, e.g. as a prime order group for a large prime, or a composite group of smooth order, g^β is also random in the group $\mathbb{G}$. In any case, and as usual, if one insists, on outputting a random string, we can further apply a randomness extractor (see for example, [NR04]).

A Construction for Substring Matching Sets over Polynomial Alphabet. We now give a construction that can be used together with the second partition scheme for substring matching sets over polynomial alphabet. The construction is based on the Generalized Decision Diffie Hellman Assumption (GDDH), which follows from DDH [NR04].

Assumption 4.2 (GDDH). *There exists a polynomial-time sampler $\mathcal{G}(1^\lambda)$ that outputs a group $\mathbb{G}$ and $g \in \mathbb{G}$, such that for any polynomial-size adversary $\mathcal{A}$, and any $d = O(1)$,*[7]

$$\mathsf{Adv}^{\mathsf{gddh}}_{\mathcal{A}}(\lambda) := \left| \Pr\left[\mathcal{A}(\mathbb{G}, \left(g^{\prod_{i\in S}\alpha_i} \,\middle|\, S \subsetneq [d] \right), g^{\gamma_b}) = b \,\middle|\, \begin{array}{l} (\mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda) \\ \alpha_1, \dots, \alpha_d, \beta \leftarrow \mathbb{Z}^*_{|\mathbb{G}|} \\ \gamma_0 = \prod_{i\in[d]} \alpha_i, \gamma_1 = \beta \\ b \leftarrow \{0,1\} \end{array} \right] - \frac{1}{2} \right| \leq \mathrm{negl}(\lambda) \enspace .$$

We next describe the construction, which is a carefully augmented variant of the above construction. At first, it might be tempting to use the previous CPRF construction with the binary substring matching partition, as before only that instead of using the same pad α, we would use independent pads $\alpha_1, \dots, \alpha_d$ for each one of the coordinates. The problem with this approach is the fact that the constrained key will need to include all the elements $\left(g^{\prod_{i\in S}\alpha_i} \,\middle|\, S \subsetneq [d] \right)$. In the previous construction, we used the first partition scheme (over binary alphabet) where $d \approx \log Q/\delta$. Thus, the size of the above set is Q/δ and is too large. (It is a polynomial in λ, but one that depends on the adversary's number of queries and advantage, which are not apriori bounded. Before, this was not an issue as we only considered the set of all powers of the same element α.)

To circumvent the above we use the second partition scheme presented over a polynomial alphabet that has a constant d. This require a natural augmentation of the construction, which we present now.

For domain $\{0,1\}^{\widehat{n}} \cong \Sigma^m$, where Σ is of size $\sigma = O(n^2)$, the function is defined as follows:

- Each (non-punctured) key K consists of an $m \times \sigma$ matrix $\left(k_{i,j} \leftarrow \mathbb{Z}^*_{|\mathbb{G}|} \right)_{i\in[m], j\in\Sigma}$, as well as $\mathbb{G}, g$.
- The value of the function on $x \in \Sigma^m$ is given by $\mathsf{CPRF.Eval}_K(x) = g^{\prod_{i\in[m]} k_{i,x_i}}$.
- The constraining algorithm $\mathsf{CPRF.Cons}(K, s)$, given a key K and a string $s \in (\Sigma \cup \{\star\})^m$, with d non-wildcards at positions $\{i_1, \dots, i_d\} = D \subseteq [m]$, works as follows:
  - Samples $\alpha_{i_1}, \dots, \alpha_{i_d} \leftarrow \mathbb{Z}^*_{\mathbb{G}}$.
  - Outputs a punctured key K_{S_s} consisting of $s, \mathbb{G}$, $\left(g^{\prod_{\ell\in S}\alpha_{i_\ell}} \,\middle|\, S \subsetneq [d] \right)$, and a new set $\left(k'_{i,j} \right)_{i,j}$, where
  $$k'_{i,j} = \begin{cases} \alpha_i^{-1} \cdot k_{i,j} & i \in D, j = s_i \\ k_{i,j} & \text{otherwise} \end{cases} \enspace .$$
- To evaluate the function on $x \in \Sigma^m \setminus S_s$ using the punctured key K_{S_s}:
  - Let $D' \subseteq D$ be the subset of indices such that $x_i = s_i$ (note that $D' \neq D$ since $x \notin S_s$).
  - Output $\left(g^{\prod_{\ell\in D'}\alpha_{i_\ell}} \right)^{\prod_{i\in[m]} k'_{i,x_i}}$.

[7] This is a weaker variant of the usual GDDH assumption where d may be polynomial (and the elements are given by an oracle). This weaker variant will be sufficient for us.

Functionality. By definition,

$$\mathsf{CPRF.Eval}_{K_{S_s}}(x) = \left(g^{\prod_{\ell \in D'} \alpha_{i_\ell}}\right)^{\prod_{i\in[m]} k'_{i,x_i}} = \left(g^{\prod_{\ell \in D'} \alpha_{i_\ell}}\right)^{\frac{\prod_{i\in[m]} k_{i,x_i}}{\prod_{\ell \in D'} \alpha_{i_\ell}}} = g^{\prod_{i\in[m]} k_{i,x_i}} = \mathsf{CPRF.Eval}_K(x) \enspace .$$

Indistinguishability. We now prove the indistinguishability property of the constructed CPRF. The proof is similar to the proof of the previous construction. Given an (admissible) adversary $\mathcal{B}$ that breaks the indistinguishability of the CPRF, we construct and adversary $\mathcal{A}$ that breaks the GDDH assumption with the same advantage.

The breaker $\mathcal{A}$. Given $(\mathbb{G}, \left(g^{\prod_{\ell \in S} \alpha_{i_\ell}} \,\middle|\, S \subsetneq [d]\right), g^{\gamma_b})$, the adversary $\mathcal{A}$ emulates $\mathcal{B}$ as follows:

1. When $\mathcal{B}$ submits $s \in (\Sigma \cup \{\star\})^m$ to the CPRF challenger, where s has d non-wildcard entries on an index set $D \subseteq [m]$, $\mathcal{A}$ samples $\left(k'_{i,j} \leftarrow \mathbb{Z}^*_{|G|}\right)_{i,j}$. It then sends $K_{S_s} := \left(s, \mathbb{G}, \left(g^{\prod_{\ell \in S} \alpha_{i_\ell}} \,\middle|\, S \subsetneq [d]\right), \left(k'_{i,j}\right)_{i,j}\right)$ to $\mathcal{B}$.
2. Then $\mathcal{B}$ gives $x \in S_s$ as the challenge query, $\mathcal{A}$ returns $g^{\gamma_b \prod_{i\in[m]} k'_{i,x_i}}$.
3. When $\mathcal{B}$ outputs a guess b', $\mathcal{A}$ outputs the same guess.

We observe that the view of the emulated $\mathcal{B}$ is identical to its view in the CPRF game, where the induced non-punctured key is given by

$$k_{i,j} = \begin{cases} \alpha \cdot k'_{i,j} & i \in D, j = s_i \\ k_{i,j} & \text{otherwise} \end{cases} \enspace .$$

When $\gamma_b = \prod_{\ell \in D} \alpha_{i_\ell}$, this corresponds to the case that the CPRF value is returned, and when $\gamma_b \leftarrow \mathbb{Z}^*_{|G|}$ is random, this corresponds to the case that a random element $g^\beta, \beta \leftarrow \mathbb{Z}^*_{|G|}$ is returned.[8]

It follows that

$$\mathsf{Adv}^{\mathsf{gddh}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{cfprf}}_{\mathcal{B}}(\lambda) \enspace .$$

Remark 4.1 (Verifiable Unpredictable Function from Factoring). We note that [BBR99] show that a computational (rather than decisional) version of GDH holds assuming it is hard to factor Blum integers. In this version, the value $g^{\prod_{\ell \in D} \alpha_{i_\ell}}$ is only unpredictable. Following a similar construction as above, this can be shown to lead to a construction of a Verifiable Unpredictable Function [MRV99]. We omit the details.

4.2 Selective Security (or Adaptive Security from Subexponential Assumptions)

We now discuss how to obtain selective security based on plain puncturable PRFs, instead of the more general CPRFs considered above. As usual, this also gives an adaptively-secure constructions assuming subexponential hardness.

Puncturable PRFs are a special case of constrained PRFs where the collection of sets $\mathcal{S}$ includes singletons $S_x = \{x\}$; namely, every constrained key $K_{\{x\}}$ allows computing the PRF everywhere, but at the point x. As shown in [BGI14, BW13, KPTZ13b, KPTZ13a], the GGM [GGM86] PRF yield puncturable PRFs. In particular, (subexponential) puncturable PRFs can be constructed from (subexponential) one-way functions.

Recall that in the case of selective security (see Definition 2.5), the VRF adversary announces the challenge query x_* ahead of time, before obtaining the verification key, or performing any evaluation query.

[8] The same footnote 6 applies.

In this case, we can avoid using partition schemes, and replace use puncturable PRFs as our CPRFs. Alternatively, we can think of a trivial partition scheme for the collection of singletons where the encoding is the identity, and the partition sampler also gets the challenges x_* as input, and outputs it as the partition, corresponding to the case that successful partition occurs with probability $\tau = 1$. The same analysis as in Section 3.3 now applies.

By taking all the underlying primitives to be subexponentially hard (say 2^{λ^ε}-hard), the scheme can be shown to be adaptively secure (when setting the underlying security parameter to $n^{1/\varepsilon}$). This follows by a standard reduction (see for example [ACF14]).

4.3 Room for Improvement

Currently, to achieve adaptive security, we rely either on subexponentially-hard OWFs, or (polynomially-hard) LWE, d-power DDH, or IO (in addition to NIWIs and non-interactive commitments). A natural direction is to try and improve this to other polynomial assumptions — ideally polynomial one-way functions. One way to do this is to construct constrained PRFs for one of the two set collections considered here, namely, substring matching or universal hash sets. Alternatively, one can try to come up with partitioning schemes for other set collections $\mathcal{S}$ together with corresponding constrained PRFs.

References

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

[BB04] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 443–459, 2004.

[BBR99] Eli Biham, Dan Boneh, and Omer Reingold. Breaking generalized diffie-hellmann modulo a composite is no easier than factoring. *Inf. Process. Lett.*, 70(2):83–87, 1999.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 501–519, Buenos Aires, Argentina, March 26–28, 2014. Springer, Heidelberg, Germany.

[BGJS16] Saikrishna Badrinarayanan, Vipul Goyal, Aayush Jain, and Amit Sahai. Verifiable functional encryption. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, pages 557–587, 2016.

[BGRV09] Zvika Brakerski, Shafi Goldwasser, Guy N. Rothblum, and Vinod Vaikuntanathan. Weak verifiable random functions. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, pages 558–576, 2009.

[Blu81] Manuel Blum. Coin flipping by telephone. In *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981.*, pages 11–15, 1981.

[BMR10] Dan Boneh, Hart William Montgomery, and Ananth Raghunathan. Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 131–140, 2010.

[BOV07] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

[BP15] Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 401–427, 2015.

[BR09] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for waters' IBE scheme. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 407–424, 2009.

[BSMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM J. Comput.*, 20(6):1084–1118, 1991.

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic prfs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 1–30, 2015.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 280–300, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany.

[BY96] Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *J. Cryptology*, 9(3):149–166, 1996.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 480–499, 2014.

[CM14] Melissa Chase and Sarah Meiklejohn. Déjà Q: using dual systems to revisit q-type assumptions. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 622–639, 2014.

[CRV14] Nishanth Chandran, Srinivasan Raghuraman, and Dhinakaran Vinayagamurthy. Constrained pseudorandom functions: Verifiable and delegatable. *IACR Cryptology ePrint Archive*, 2014:522, 2014.

[CW79] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[DN07] Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.

[Dod03] Yevgeniy Dodis. Efficient construction of (distributed) verifiable random functions. In *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings*, pages 1–17, 2003.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *Public Key Cryptography - PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005, Proceedings*, pages 416–431, 2005.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999.

[FS12] Dario Fiore and Dominique Schröder. Uniqueness is a different story: Impossibility of verifiable random functions from trapdoor permutations. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 636–653, 2012.

[Fuc14] Georg Fuchsbauer. Constrained verifiable random functions. In *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, pages 95–114, 2014.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GHKW17] Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. *IACR Cryptology ePrint Archive*, 2017:21, 2017.

[GO92] Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, pages 228–245, 1992.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11, 2012.

[GR13] Oded Goldreich and Ron D. Rothblum. Enhancements of trapdoor permutations. *J. Cryptology*, 26(3):484–512, 2013.

[HJ16] Dennis Hofheinz and Tibor Jager. Verifiable random functions from standard assumptions. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 336–362, 2016.

[HKW15] Susan Hohenberger, Venkata Koppula, and Brent Waters. Adaptively secure puncturable pseudorandom functions in the standard model. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*, pages 79–102, 2015.

[HW10] Susan Hohenberger and Brent Waters. Constructing verifiable random functions with large input spaces. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 656–672, 2010.

[Jag15] Tibor Jager. Verifiable random functions from weaker assumptions. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 121–143, 2015.

[KPTZ13a] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security*, pages 669–684, Berlin, Germany, November 4–8, 2013. ACM Press.

[KPTZ13b] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *CCS*, pages 669–684, 2013.

[Lys02] Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 597–612, 2002.

[MRV99] Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 120–130, 1999.

[MV99] Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 71–80, 1999.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[NR99] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, USA, May 31 – June 3, 2014. ACM Press.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pages 114–127, 2005.