Adaptive Security of Constrained PRFs

Georg Fuchsbauer[1], Momchil Konstantinov[2], Krzysztof Pietrzak[1⋆], and Vanishree Rao[3]

[1] IST Austria
[2] London School of Geometry and Number Theory, UK
[3] UCLA, USA

Abstract. Constrained pseudorandom functions have recently been introduced independently by Boneh and Waters [Asiacrypt'13], Kiayias et al. [CCS'13], and Boyle et al. [PKC'14]. In a standard pseudorandom function (PRF) a key k is used to evaluate the PRF on all inputs in the domain. Constrained PRFs additionally offer the functionality to delegate "constrained" keys k_S which allow to evaluate the PRF only on a subset S of the domain.

The three above-mentioned papers all show that the classical GGM construction [J.ACM'86] of a PRF from a pseudorandom generator (PRG) directly gives a constrained PRF where one can compute constrained keys to evaluate the PRF on all inputs with a given prefix. This constrained PRF has already found many interesting applications. Unfortunately, the existing security proofs only show selective security (by a reduction to the security of the underlying PRG). To get full security, one has to use complexity leveraging, which loses an exponential factor 2^N in security, where N is the input length.

The first contribution of this paper is a new reduction that only loses a quasipolynomial factor $q^{\log N}$, where q is the number of adversarial queries. For this we develop a novel proof technique which constructs a distinguisher by interleaving simple guessing steps and hybrid arguments a small number of times. This approach might be of interest also in other contexts where currently the only technique to achieve full security is complexity leveraging.

Our second contribution is concerned with another constrained PRF, due to Boneh and Waters, which allows for constrained keys for the more general class of bit-fixing functions. Their security proof also suffers from a 2^N loss. We construct a meta-reduction which shows that any "simple" reduction that proves full security of this construction from a non-interactive hardness assumption must incur an exponential security loss.

Keywords: Constrained PRF, complexity leveraging, full security, meta-reduction.

1 Introduction

PRFs. Pseudorandom functions (PRFs) were introduced by Goldreich, Goldwasser and Micali [GGM86]. A PRF is an efficiently computable keyed function $\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, where $\mathsf{F}(K,.)$, instantiated with a random key $K \xleftarrow{*} \mathcal{K}$, cannot be distinguished from a function randomly chosen from the set of all functions $\mathcal{X} \to \mathcal{Y}$ with non-negligible probability.

Constrained PRFs. Recently, the notion of constrained PRFs (CPRFs) was introduced independently in three papers by Boneh and Waters [BW13], Boyle, Goldwasser and Ivan [BGI14] and Kiayias, Papadopoulos, Triandopoulos and Zacharias [KPTZ13].[4]

A constrained PRF is defined with respect to a set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$ and supports the functionality to "delegate" (short) keys that can only be used to evaluate the function $\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ on inputs specified by a subset $S \in \mathcal{S}$. Concretely, there is a "constrained" keyspace $\mathcal{K}_c$ and additional algorithms $\mathsf{F.constrain}\colon \mathcal{K} \times \mathcal{S} \to \mathcal{K}_c$ and $\mathsf{F.eval}\colon \mathcal{K}_c \times \mathcal{X} \to \mathcal{Y}$, which for all $k \in \mathcal{K}, S \in \mathcal{S}, x \in S$ and $k_S \leftarrow \mathsf{F.constrain}(k,S)$, satisfy $\mathsf{F.eval}(k_S,x) = \mathsf{F}(k,x)$ if $x \in S$ and $\mathsf{F.eval}(k_S,x) = \perp$ otherwise.

⋆ Research supported by ERC starting grant (259668-PSPC)

[4] The name "constrained PRF" is from [BW13]; in [KPTZ13] and [BGI14] these objects are called "delegatable PRFs" and "functional PRFs", respectively. In this paper we follow the naming and notation from [BW13].

The GGM and the Boneh-Waters construction. All three papers [BW13,BGI14,KPTZ13] show that the classical GGM construction [GGM86] of the PRF $\mathsf{GGM}\colon \{0,1\}^\lambda \times \{0,1\}^N \to \{0,1\}^\lambda$ from a length-doubling pseudorandom generator (PRG) $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ directly gives a constrained PRF, where for any key K and input prefix $z \in \{0,1\}^{\leq N}$, one can generate a constrained key K_z that allows to evaluate $\mathsf{GGM}(K,x)$ for any x with prefix z. This simple constrained PRF has found many applications; apart from those discussed in [BW13,BGI14,KPTZ13], it can be used to construct the so-called "punctured PRF", which is a key ingredient in almost all the recent proofs of indistinguishability obfuscation [SW13,BCPR13,HSW14].

Boneh and Waters [BW13] construct a constrained PRF for a much more general set of constraints, where one can delegate keys that fix any subset of bits of the input (not just the prefix, as in GGM). The construction is based on leveled multilinear maps [GGH13] and its security relies on a generalization of the decisional Diffie-Hellman assumption.

Security of constrained PRFs. The security definition for normal PRFs is quite intuitive. One considers two experiments – the "real" experiment and the "random" experiment – in which an adversary A gets access to an oracle $\mathcal{O}(.)$, and finally outputs a bit. In the real experiment, $\mathcal{O}(.)$ implements the PRF $\mathsf{F}(K,.)$ using a random key; in the random experiment, $\mathcal{O}(.)$ implements a random function. We require that every efficient A outputs 1 in both experiments with (almost) the same probability.

Defining the security of constrained PRFs requires a bit more thought. We want to give an adversary access not only to $\mathsf{F}(K,.)$, but also to the constraining function $\mathsf{F.constrain}(K,.)$. But now we cannot expect the values $\mathsf{F}(K,.)$ to look random, as an adversary can always ask for a key $K_S \leftarrow \mathsf{F.constrain}(K,S)$ and then for any $x \in S$ check if $\mathsf{F}(K,x) \stackrel{?}{=} \mathsf{F.eval}(K_S,x)$.

Instead, security is formalized by defining the experiments in two phases. In the first phase, the adversary gets access to the same pair of oracles $\mathsf{F}(K,.), \mathsf{F.constrain}(K,.)$ in both the experiments. The experiments differ only in a second phase, where the adversary chooses some challenge query x^*. In the real experiment the adversary then obtains $\mathsf{F}(K,x^*)$, whereas in the random experiment she gets a random value. Intuitively, when no efficient adversary can distinguish these two games, this means that the outputs of $\mathsf{F}(K,.)$ look random on all points that the adversary cannot compute by herself using the constrained keys she has received so far.

Selective vs. full security. In the above definition, we let the adversary choose the challenge input x^* after getting access to the oracles. This is the notion one typically requires, and is called "full security" or "adaptive security". One can also consider a weaker "selective security" notion, where the adversary must choose x^* before getting access to the oracles.

The reason to consider selective security notions, not only here, but also for other objects like identity-based encryption [BF01,BB04,AFL12] is that it is often much easier to achieve. Although there exists a simple generic technique called "complexity leveraging", which translates any selective security guarantee into a security bound for full security, this technique (which really just consists of guessing the challenge) typically loses an exponential factor (in the length of the challenge) in the quality of the reduction, often making the resulting security guarantee meaningless for practical parameters.

1.1 Our Contributions

[BW13,BGI14,KPTZ13] only show selective security of the GGM constrained PRF, and [BW13] also only give a selective security proof for their bit-fixing constrained PRF. In this paper we investigate the full security of these two constructions. For GGM we achieve a positive result, giving a reduction that only

loses a quasipolynomial factor. For the Boneh-Waters bit-fixing CPRF we give a negative result, showing that for a large class of reductions, an exponential loss is necessary. We elaborate on these results below.

A quasipolynomial reduction for GGM. To prove full security of $\mathsf{GGM}\colon \{0,1\}^\lambda \times \{0,1\}^N \to \{0,1\}^\lambda$, the "standard" proof proceeds in two steps (we give a precise statement in Proposition 3 of this paper).

1. A guessing step (a.k.a. complexity leveraging), which reduces full to selective security. This step loses an exponential factor 2^N in the input length N.
2. Now one applies a hybrid argument which loses a factor $2N$.

Readers not familiar with hybrid arguments can find a simple application of this technique in Appendix A.

The above two steps transform an adversary $\mathsf{A_f}$ that breaks the full security of GGM with advantage ϵ into a new adversary that breaks the security of the underlying pseudorandom generator G (used to construct the GGM function) with advantage $\epsilon/(2N \cdot 2^N)$.

As a consequence, even if one makes a strong exponential hardness assumption on the PRG G, one must use a PRG whose domain is $\Theta(N)$ bits in order to get any meaningful security guarantee.

The reason for the huge security loss is the guessing step, in which one basically guesses the challenge $x^* \in \{0,1\}^N$, which is correct with probability 2^{-N}. To avoid this exponential loss, one must thus avoid guessing the entire x^*. Our new proof also consists of a guessing step followed by a hybrid argument.

1. A guessing step, where (for some ℓ) we guess which of the adversary's queries will be the first one that agrees with x^* in the first ℓ positions.[5] This step loses a factor q, which denotes the number of queries made by the adversary.
2. A hybrid argument which loses a factor 3.

The above two steps only lose a factor $3q$. Unfortunately, after one iteration of this approach we do not get a distinguisher of G right away. Very informally, what we achieve is the following. We start with two games which in some sense are at distance N from each other, and we end up with two games with are at distance $N/2$. We can iterate the above process $n := \log N$ times to end up with games at distance $N/2^n = 1$. Finally, from any distinguisher for games at distance 1 we can get a distinguisher for the PRG G with the same advantage. Thus, starting from an adversary against the full security of GGM with advantage ϵ, we get a distinguisher for the PRG with advantage $\epsilon/(3q)^{\log N}$.

We can combine this approach with the original proof, and this way obtain a quasipolynomial loss of $2q \log q \cdot (3q)^{\log N - \log\log q}$. To give some numerical example, let $N = 2^{10} = 1024$ and $q = 2^{32}$. Then we get a loss of $2q \log q \cdot (3q)^{\log N - \log\log q} = 2 \cdot 2^{32} \cdot 32 \cdot (3 \cdot 2^{32})^{10-5} = 2^{198} \cdot 3^5 \leq 2^{206}$, whereas complexity leveraging loses $2N2^N = 2^{1035}$.

Although our proof is somewhat tailored to the GGM construction, the general "fine-grained" guessing approach outlined above might be useful to improve the bounds for other constructions (like CPRFs, and even IBE schemes) where currently the only proof technique that can be applied is complexity leveraging.

A lower bound for Boneh-Waters and Hofheinz's construction. We then turn our attention to the bit-fixing constrained PRF of Boneh and Waters [BW13]. Also for this construction complexity leveraging—losing an exponential factor—is the only known way to prove full security. We give strong evidence that this is inherent.

Concretely, we prove that every "simple" reduction (which runs the adversary once without rewinding; see Section 5.2) of the full security of this scheme from any decisional (and thus also search) assumption must lose an exponential factor. Our proof is a so-called meta-reduction [BV98,Cor02,FS10,Fis12], showing

[5] This guessing is somewhat reminiscent of a proof technique from [HW09].

that any reduction that breaks the underlying assumption when given access to any adversary that breaks the CPRF, could be used to break the underlying assumption without the help of an adversary.

This impossibility result is similar to existing results, the closest one being a result of Lewko and Waters [LW14] ruling out security proofs without exponential loss for so-called "prefix-encryption" (which satisfies some special properties). Other related results are those of Coron [Cor02] and Hofheinz et al. [HJK12], which show that security reductions for certain signature schemes must lose a factor polynomial in the number of signing queries.

The above impossibility proofs are for public-key objects, where a public key exists that uniquely determines the input/output distribution of the object. This property is crucially used in the proof, wherein one first gets the public key and then runs the reduction, rewinding the reduction multiple times to the point right after the public key has been received.

As we consider a secret-key primitive, the above approach seems to be inapplicable. We overcome this by observing that for the Boneh-Waters CPRF we can initially make some fixed "fingerprint" queries, which then uniquely determine the remaining outputs. We can therefore use the responses to these fingerprint queries instead of a public key as in [LW14].

Hofheinz has (independently and concurrently with us) investigated the adaptive security of bit-fixing constrained PRFs [Hof14]. He gives a new construction of such PRFs which is more sophisticated than the Boneh-Waters construction, but for which he can give a security reduction that only loses a polynomial factor. The main tool that allows Hofheinz to overcome our impossibility result is the use of a random oracle $H(.)$. Very informally, instead of evaluating the PRF on an input X, it is evaluated on $H(X)$ which forces an attacker to make every query X explicit. Unfortunately, this idea does not work directly as it completely destroys the structure of the preimages, and thus makes the construction of short delegatable keys impossible. Hofheinz's construction deals with this problem using several other ideas.

2 Preliminaries

For $a \in \mathbb{N}$, we let $[a] = \{1, 2, \ldots, a\}$ and $[a]_0 = \{0, 1, \ldots, a\}$. With $\{0,1\}^{\leq a} = \bigcup_{i \leq a} \{0,1\}^i$ we denote the set of bitstrings of length at most a, including the empty string $\emptyset$. U_a denotes the random variable with uniform distribution over $\{0,1\}^a$. $X \| Y$ denotes the concatenation of the bitstrings X and Y. For sets $\mathcal{X}, \mathcal{Y}$, we denote with $\mathcal{F}[\mathcal{X}, \mathcal{Y}]$ the set of all functions $\mathcal{X} \to \mathcal{Y}$. $\mathcal{F}[a, b]$ is short for $\mathcal{F}[\{0,1\}^a, \{0,1\}^b]$. For $x \in \{0,1\}^*$, we denote with x_i the i-th bit of x, and with $x[i \ldots j]$ the substring $x_i \| x_{i+1} \| \ldots \| x_j$.

Definition 1 (Indistinguishability). *Two distributions X and Y are (ϵ, s)-***indistinguishable***, denoted $X \sim_{(\epsilon,s)} Y$, if no circuit D of size s can distinguish them with advantage greater than ϵ, i.e.,*

$$X \sim_{(\epsilon,s)} Y \iff \forall\, \mathsf{D}, |\mathsf{D}| \leq s \ : \ |\Pr[\mathsf{D}(X) = 1] - \Pr[\mathsf{D}(Y) = 1]| \leq \epsilon \ .$$

$X \sim_\delta Y$ denotes that the statistical distance of X and Y is δ (i.e., $X \sim_{(\delta,\infty)} Y$), and $X \sim Y$ denotes that they have the same distribution.

The following definition measuring how "close" sets (that differ in one element) are, will be useful in defining neighboring hybrids in our hybrid arguments.

Definition 2 (Neighboring sets). *For $k \in \mathbb{N}^+$, sets $\mathcal{S}, \mathcal{S}' \subset \mathbb{N}^0$ are k-***neighboring** *if*

1. *$\mathcal{S} \,\Delta\, \mathcal{S}' := (\mathcal{S} \cup \mathcal{S}') \setminus (\mathcal{S} \cap \mathcal{S}') = \{d\}$ for some $d \in \mathbb{N}^0$, i.e., they differ in exactly one element d.*
2. *$d - k \in \mathcal{S}$.*
3. *$\forall i \in [k-1] \ : \ d - i \notin \mathcal{S}$.*

Definition 3 (PRG). *An efficient function* $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ *is an* (ϵ, s)*-secure (length-doubling)* **pseudorandom generator** *(PRG) if*

$$\mathsf{G}(U_\lambda) \sim_{(\epsilon,s)} U_{2\lambda} \ .$$

Definition 4 (PRF). *A keyed function* $\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *is an* (ϵ, s, q)*-secure* **pseudorandom function** *if for all adversaries* A *of size at most* s *making at most* q *oracle queries*

$$\left|\Pr_{K\leftarrow\mathcal{K}}[\mathsf{A}^{\mathsf{F}(K,.)} \to 1] - \Pr_{f\leftarrow\mathcal{F}[\mathcal{X},\mathcal{Y}]}[\mathsf{A}^{f(.)} \to 1]\right| \leq \epsilon \ .$$

Constrained pseudorandom functions. Following [BW13], we say that a function $\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a constrained PRF for a set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$, if there is a "constrained" keyspace $\mathcal{K}_c$ and algorithms

$$\mathsf{F.constrain}\colon \mathcal{K} \times \mathcal{S} \to \mathcal{K}_c \quad \text{and} \quad \mathsf{F.eval}\colon \mathcal{K}_c \times \mathcal{X} \to \mathcal{Y} \ ,$$

which for all $k \in \mathcal{K}, S \in \mathcal{S}, x \in S$ and $k_S \leftarrow \mathsf{F.constrain}(k, S)$ satisfy $\mathsf{F.eval}(k_S, x) = \begin{cases} \mathsf{F}(k,x) & \text{if } x \in S \\ \perp & \text{otherwise} \end{cases}$

That is, $\mathsf{F.constrain}(k, S)$ outputs a key k_S that allows to evaluate $\mathsf{F}(k, .)$ on all $x \in S$.

Informally, a constrained PRF F is secure, if no efficient adversary can distinguish $\mathsf{F}(k, x^*)$ from random, even given access to $\mathsf{F}(k, .)$ and $\mathsf{F.constrain}(k, .)$ which he can query on all $x \neq x^*$ and $S \in \mathcal{S}$ where $x^* \notin S$, respectively. We will always assume that $\mathcal{S}$ contains all singletons, i.e., $\forall x \in \mathcal{X} \ : \ \{x\} \in \mathcal{S}$; this way we do not have to explicitly give access to $\mathsf{F}(k, .)$ to an adversary, as $\mathsf{F}(k, x)$ can be learned by querying for $k_x \leftarrow \mathsf{F.constrain}(k, \{x\})$ and computing $\mathsf{F.eval}(k_x, x)$.

We distinguish between selective and full security. In the selective security game, the adversary must choose the challenge x^* before querying the oracles. Both games are parametrized by the maximal number q of queries the adversary makes, of which the last query is the challenge query.

$\mathbf{Exp}^{\mathsf{sel}}_{\mathsf{CPRF}}(\mathsf{A}, \mathsf{F}, b, q)$	$\mathbf{Exp}^{\mathsf{full}}_{\mathsf{CPRF}}(\mathsf{A}, \mathsf{F}, b, q)$	**Oracle** $\mathcal{O}(S)$
$K \xleftarrow{*} \mathcal{K}$, $\hat{S} := \emptyset$, $c := 0$	$K \xleftarrow{*} \mathcal{K}$, $\hat{S} := \emptyset$, $c := 0$	$c := c + 1$
$x^* \leftarrow \mathsf{A}$	$\mathsf{A}^{\mathcal{O}(.)}$	if $c = q - 1$ return $\perp$
$\mathsf{A}^{\mathcal{O}(.)}$	$x^* \leftarrow \mathsf{A}$	$\hat{\mathcal{S}} := \hat{\mathcal{S}} \cup S$
$C_0 \xleftarrow{*} \mathcal{Y}$, $C_1 := \mathsf{F}(K, x^*)$	$C_0 \xleftarrow{*} \mathcal{Y}$, $C_1 := \mathsf{F}(K, x^*)$	$k_S \leftarrow \mathsf{F.constrain}(K, S)$
A gets C_b	A gets C_b	return k_S
$\tilde{b} \leftarrow \mathsf{A}$	$\tilde{b} \leftarrow \mathsf{A}$	
if $x^* \in \hat{S}$ return 0	if $x^* \in \hat{S}$ return 0	
return $\tilde{b}$	return $\tilde{b}$	

For $\mathsf{atk} \in \{\mathsf{sel}, \mathsf{full}\}$ we define A's advantage as

$$\mathsf{Adv}^{\mathsf{atk}}_{\mathsf{F}}(\mathsf{A}, q) = 2\left|\Pr_{b\leftarrow\{0,1\}}[\mathbf{Exp}^{\mathsf{atk}}_{\mathsf{CPRF}}(\mathsf{A}, \mathsf{F}, b, q) = b] - \tfrac{1}{2}\right| \tag{1}$$

and denote with

$$\mathsf{Adv}^{\mathsf{atk}}_{\mathsf{F}}(s, q) = \max_{\mathsf{A},|\mathsf{A}|\leq s} \mathsf{Adv}^{\mathsf{atk}}_{\mathsf{F}}(\mathsf{A}, q)$$

the advantage of the best q-query adversary of size at most s.

Definition 5 (Selective and full security of CPRFs). *A constrained PRF* F *is*

- **selectively** (ϵ, s, q)**-secure** *if* $\mathsf{Adv}^{\mathsf{sel}}_{\mathsf{F}}(s, q) \leq \epsilon$ *and*
- **fully** (ϵ, s, q)**-secure** *if* $\mathsf{Adv}^{\mathsf{full}}_{\mathsf{F}}(s, q) \leq \epsilon$.

Remark 1 (CCA1 vs. CCA2 security). In the selective and full security notion, we assume that the challenge query x^* is only made at the very end, when A has no longer access to the oracle (this is reminiscent of CCA1 security). All our positive results hold for stronger notions (reminiscent to CCA2 security) where A still has access to $\mathcal{O}(.)$ after making the challenge query, but may not query on any S where $x^* \in S$.

Remark 2 (Several challenge queries). We only allow the adversary one challenge query. As observed by [BW13], this implies security against any $t > 1$ challenge queries, losing a factor of t in the distinguishing advantage, by a standard hybrid argument.

Using what is sometimes called "complexity leveraging", one can show that selective security implies full security, but the distinguishing advantage drops by a factor of the domain size $|\mathcal{X}|$. The following is proved in Appendix B.1.

Lemma 1 (complexity leveraging). *If the constrained PRF $\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is (ϵ, s, q)-***selectively** *secure, then it is $(\epsilon|\mathcal{X}|, s', q)$-***fully** *secure (where $s' = s - O(\log |\mathcal{X}|)$), i.e.,*

$$\mathsf{Adv}^{\mathsf{full}}_{\mathsf{F}}(s', q) \leq |\mathcal{X}| \cdot \mathsf{Adv}^{\mathsf{sel}}_{\mathsf{F}}(s, q) \ .$$

3 The GGM Construction

The GGM construction, named after its inventors Goldreich, Goldwasser, and Micali [GGM86], is a construction of a keyed function $\mathsf{GGM}^{\mathsf{G}}\colon \{0,1\}^\lambda \times \{0,1\}^* \to \{0,1\}^\lambda$ from any length-doubling PRG $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$, defined as

$$\mathsf{GGM}(K_\emptyset, X) = K_X \quad \text{where} \quad \forall Z \in \{0,1\}^{\leq N-1} \ : \ K_{Z\|0} \| K_{Z\|1} = \mathsf{G}(K_Z) \ . \tag{2}$$

[GGM86] shows that when the inputs are restricted to $\{0,1\}^N$, $\mathsf{GGM}^{\mathsf{G}}(K, .)$ is a secure PRF if G is a secure PRG. Their proof is one of the first applications of the so-called hybrid argument.[6] The proof loses a factor of $q \cdot N$ in distinguishing advantage (where q is the number of queries).

Proposition 1 (GGM is a PRF [GGM86]). *If $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is an $(\epsilon_{\mathsf{G}}, s_{\mathsf{G}})$-secure PRG then (for any N, q) $\mathsf{GGM}^{\mathsf{G}}\colon \{0,1\}^\lambda \times \{0,1\}^N \to \{0,1\}^\lambda$ is an (ϵ, s, q)-secure PRF with*

$$\epsilon = \epsilon_{\mathsf{G}} \cdot q \cdot N \qquad s = s_{\mathsf{G}} - O(q \cdot N \cdot |\mathsf{G}|)$$

We will not give a proof of this proposition here; however it follows from Proposition 2 below, for which we do give a proof sketch. In his book [Gol01] Goldreich presents several generalizations of GGM, including a variant which is secure even if we allow the entire domain $\{0,1\}^*$ as inputs. Here, we'd like to mention that the original GGM construction is a secure "prefix-free" PRF as defined below. The reason for presenting this variant of GGM here is so we can later, in Remark 3, discuss why this variant of GGM does *not* already imply security of GGM as a constrained PRF. Instead of the number q of points queried by the adversary, the security of a prefix-free PRF with domain $\{0,1\}^*$ is parametrized by the sum m of the bitlengths of all queries.

Definition 6 (PF-PRF). *A keyed function $\mathsf{F}\colon \mathcal{K} \times \{0,1\}^* \to \mathcal{Y}$ is an (ϵ, s, m)-secure* **prefix-free pseudorandom function** *(PF-PRF) if for all adversaries A of size at most s making queries of total bitlength at most m, but where no query can be a prefix of another query,*

$$\left| \Pr_{K \xleftarrow{*} \mathcal{K}}[\mathsf{A}^{\mathsf{F}(K,.)} \to 1] - \Pr_{f \leftarrow \mathcal{R}[\{0,1\}^*, \mathcal{Y}]}[\mathsf{A}^{f(.)} \to 1] \right| \leq \epsilon \ .$$

[6] The first application is in the "probabilistic encryption" paper [GM84].

Proposition 2 (GGM is a PF-PRF). *If* $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ *is an* $(\epsilon_\mathsf{G}, s_\mathsf{G})$*-secure PRG then (for any* m*)* $\mathsf{GGM}^\mathsf{G}\colon \{0,1\}^\lambda \times \{0,1\}^* \to \{0,1\}^\lambda$ *is an* (ϵ, s, m)*-secure PF-PRF with*

$$\epsilon = \epsilon_\mathsf{G} \cdot m \qquad s = s_\mathsf{G} - O(m \cdot |\mathsf{G}|)$$

We prove this proposition in Appendix B.2.[7]

3.1 GGM is a Constrained PRF

As observed recently by three different works independently [BW13,BGI14,KPTZ13], the GGM construction can be used as a constrained PRF for the set $\mathcal{S}_\mathsf{pre}$ defined as

$$\mathcal{S}_\mathsf{pre} = \{S_p \ : \ p \in \{0,1\}^{\leq N}\} \quad \text{where} \quad S_p = \{p \| z \ : \ z \in \{0,1\}^{N-|p|}\} \ .$$

Thus, given a key K_p for the set S_p, one can evaluate $\mathsf{GGM}^\mathsf{G}(K, .)$ on all inputs with prefix p. Formally, the constrained PRF with key $K = K_\emptyset$ is defined using (2)

$$\mathsf{GGM}^\mathsf{G}.\mathsf{constrain}(K_\emptyset, p) = \mathsf{GGM}^\mathsf{G}(K_\emptyset, p) = K_p \qquad \mathsf{GGM}^\mathsf{G}.\mathsf{eval}(K_p, x = p\|z) = \mathsf{GGM}^\mathsf{G}(K_p, z) = K_x \ .$$

Remark 3. One might be tempted to think that the fact that GGM is a PF-PRF (Proposition 2), together with the fact that constrained-key derivation is simply the GGM function itself, already implies that it is a secure constrained PRF. Unfortunately, this is not sufficient, as the (selective and full) security notions for constrained PRFs do allow queries that are prefixes of previous queries

The *selective* security of this construction can be proven using a standard hybrid argument, losing only a factor of $2N$ in the distinguishing advantage.

Proving full security seems much more challenging, and prior to our work it was only known how to achieve full security by complexity leveraging (cf. Lemma 1), which loses an additional exponential factor 2^N in distinguishing advantage, as stated in Proposition 3 below.

Remark 4. In the proof of Proposition 3 and Theorem 1 we will slightly cheat, as in the security game when $b = 0$ (i.e., when the challenge output is random) we not only replace the challenge output K_{x^*}, but also its sibling $K_{x^*[1\ldots N-1]\overline{x}^*_N}$, with a random value. Thus, technically this only proves security for inputs of length $N-1$ (as we can e.g. simply forbid queries $x\|0, x \in \{0,1\}^{N-1}$, in which case it is irrelevant what the sibling is, as it will never be revealed). The proofs without this cheat require one extra hybrid, which requires a somewhat different treatment than all others hybrids and thus would complicate certain proofs and definitions. Hence, we chose to not include it. The bounds stated in Proposition 3 and Theorem 1 are the bounds we get *without* this cheat.

Proposition 3. *If* $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ *is an* $(\epsilon_\mathsf{G}, s_\mathsf{G})$*-secure PRG then (for any* N, q*)* $\mathsf{GGM}^\mathsf{G}\colon \{0,1\}^N \to \{0,1\}^\lambda$ *is a*

1. **selectively** (ϵ, s, q)*-secure constrained PRF for* $\mathcal{S}_\mathsf{pre}$*, where*

$$\epsilon = \epsilon_\mathsf{G} \cdot 2N \qquad s = s_\mathsf{G} - O(q \cdot N \cdot |\mathsf{G}|)$$

[7] Note that if we drop the restriction that queries must be prefix-free, the construction is trivially insecure, as from $y = \mathsf{GGM}^\mathsf{G}(K, x)$ one can compute $y' = \mathsf{GGM}^\mathsf{G}(K, x\|z)$ for any z, thus y' is not pseudorandom given y.

2. **fully** (ϵ, s, q)*-secure constrained PRF for* $\mathcal{S}_{\mathsf{pre}}$*, where*

$$\epsilon = \epsilon_{\mathsf{G}} \cdot 2^N 2N \qquad s = s_{\mathsf{G}} - O(q \cdot N \cdot |\mathsf{G}|)$$

Full security as stated in item *2.* of the proposition follows from selective security (item *1*) by complexity leveraging as explained in Lemma 1. To prove selective security, we consider two games H_0 and H_{2N-1} that correspond to the real and random selective-security games. We then define hybrid games $H_1, \ldots, H_{2N-2}$ by embedding random values along the path $K_\emptyset, K_{x_1^*}, \ldots, K_{x^*[1\ldots N-1]}, K_{x^*}$, so that from any distinguisher for two consecutive games H_i, H_{i+1}, we get a distinguisher for the PRG G with the same advantage. (See Appendix B.3 for proof details.)

This hybrid argument only loses a factor $2N$ in distinguishing advantage, but complexity leveraging loses a huge factor 2^N. In the next section we show how to prove full security avoiding such an exponential loss.

4 Full Security with Quasipolynomial Loss

Theorem 1. *If* $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ *is an* $(\epsilon_{\mathsf{G}}, s_{\mathsf{G}})$*-secure PRG then (for any* N, q*)* $\mathsf{GGM}^{\mathsf{G}}\colon \{0,1\}^N \to \{0,1\}^\lambda$ *is a* **fully** (ϵ, s, q)*-secure constrained PRF for* $\mathcal{S}_{\mathsf{pre}}$*, where*

$$\epsilon = \epsilon_{\mathsf{G}} \cdot (3q)^{\log N} \qquad s = s_{\mathsf{G}} - O(q \cdot N \cdot |\mathsf{G}|)$$

At the end of this section we will sketch how to combine the proof of this theorem with the standard complexity leveraging proof from Proposition 3 to get a better loss of $\epsilon = \epsilon_{\mathsf{G}} \cdot 2q \log q \cdot (3q)^{\log N - \log\log q}$.

Neighboring sets with low weight. Let $N = 2^n$ be a power of 2. Below we construct $3^n + 1$ subsets $\mathcal{S}_{\langle 0\rangle}, \ldots, \mathcal{S}_{\langle 10_n\rangle} \subset \{0, \ldots, N\}$, which we will use in the proof of Theorem 1. It will be convenient to work with ternary numbers, which we will represent as strings of digits from $\{0,1,2\}$ within angular brackets $\langle\ldots\rangle$. We denote repetition of digits as $0_n = 0\ldots0$ (n times). Addition will also be in ternary, e.g., $\langle 202\rangle + \langle 1\rangle = \langle 210\rangle$.

We define the first and last set, with index $3^n = \langle 10_n\rangle$, as

$$\mathcal{S}_{\langle 0\rangle} := \{0\} \quad \text{and} \quad \mathcal{S}_{\langle 10_n\rangle} := \{0, N\} \ . \tag{3}$$

The remaining intermediate sets are defined recursively as follows. For $\ell = 0, \ldots, n$, we define the ℓ-th level of sets to be all the sets of the form $\mathcal{S}_{\langle ?0_{n-\ell}\rangle}$ (i.e., whose index in ternary ends with $(n-\ell)$ zeros). Thus, $\mathcal{S}_{\langle 0\rangle}$ and $\mathcal{S}_{\langle 10_n\rangle}$ are the (only) 0-level sets.

Let $\mathcal{S}_I, \mathcal{S}_{I'}$ be two consecutive ℓ-th level sets, by which we mean that $I' = I + \langle 10_{n-\ell}\rangle$. By construction, these sets will differ in exactly one element $\{d\}$ (i.e., $\mathcal{S}_I \cup \{d\} = \mathcal{S}_{I'}$ or $\mathcal{S}_{I'} \cup \{d\} = \mathcal{S}_I$). Then the two $(\ell+1)$-level sets between the ℓ-level sets $\mathcal{S}_I, \mathcal{S}_{I'}$ are defined as

$$\mathcal{S}_{I+\langle 10_{n-(\ell+1)}\rangle} := \mathcal{S}_I \cup \{d - \tfrac{N}{2^{\ell+1}}\} \quad \text{and} \quad \mathcal{S}_{I'-\langle 10_{n-(\ell+1)}\rangle} := \mathcal{S}_{I'} \cup \{d - \tfrac{N}{2^{\ell+1}}\} \ . \tag{4}$$

A concrete example for $N = 2^n = 2^3 = 8$ is illustrated in Figure 1 (where the blue nodes of H_I correspond to $\mathcal{S}_I$).

An important fact we will use is that consecutive ℓ-th level sets are $N/2^\ell$-neighboring (cf. Definition 2); in particular, consecutive n-th level sets (4 consecutive sets are illustrated at the bottom of Figure 1) are 1-neighboring, i.e.,

$$\forall I \in \{\langle 0\rangle, \ldots, \langle 2_n\rangle\} \ : \ \mathcal{S}_I \,\Delta\, \mathcal{S}_{I+\langle 1\rangle} = \{d\} \ \text{ and } \ d - 1 \in \mathcal{S}_I \ . \tag{5}$$

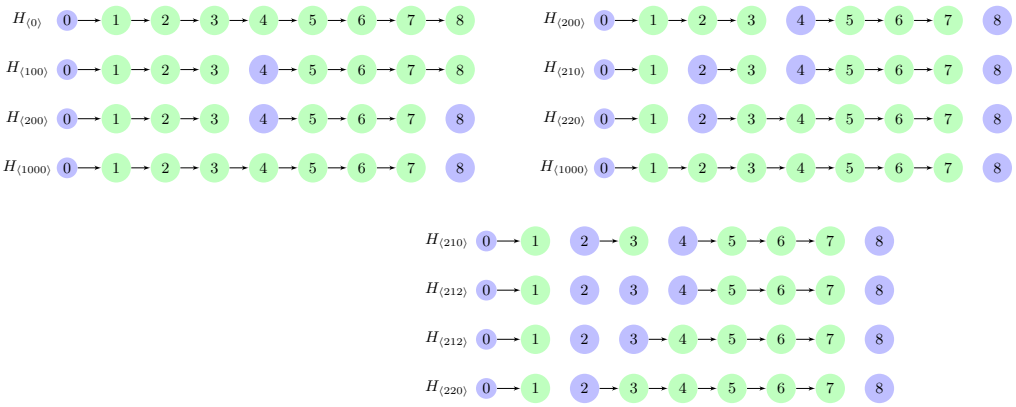

Fig. 1. Concrete example ($n = 3$) illustrating the iterative construction of hybrids in Theorem 1.

Proof of Theorem 1. Below we prove two lemmata (2 and 3) concerning the games defined in Figure 2, from which the theorem follows quite immediately. As the games and the lemmata are rather technical, we first intuitively explain what is going on, going through a concrete example as illustrated in Figure 1.

To prove the theorem, we assume that there exists an adversary $\mathsf{A_f}$ that breaks the *full* security of $\mathsf{GGM}^{\mathsf{G}}$ with some advantage ϵ, and from this, we want to construct a distinguisher for G with advantage at least $\epsilon/(3q)^n$. Like in the proof of Proposition 3, we can think of the two games $\mathsf{A_f}$ distinguishes as the games where we let $\mathsf{A_f}$ query $\mathsf{GGM}^{\mathsf{G}}$, but along the path from the root $K_\emptyset$ down to the challenge K_{x^*} we use either G or random values, as defined by the sets $\mathcal{S}_{\langle 0\rangle} = \{0\}$ and $\mathcal{S}_{\langle 10_n\rangle} = \{0, N\}$, respectively (i.e., in both cases the root $K_\emptyset$ is random, and in one game also the final output K_{x^*} is uniform). We will call these two games $H^\emptyset_{\langle 0\rangle}$ and $H^\emptyset_{\langle 10_n\rangle}$, corresponding to the games defined in Figure 2 (with $\mathcal{P} = \emptyset$ and $I = \langle 0\rangle$ and $\langle 10_n\rangle$, respectively), and as just explained, they satisfy

$$H^\emptyset_{\langle 0\rangle} \sim \mathbf{Exp}^{\mathsf{full}}_{\mathsf{CPRF}}(\mathsf{A_f}, \mathsf{GGM}^{\mathsf{G}}, 0, q) \qquad H^\emptyset_{\langle 10_n\rangle} \sim \mathbf{Exp}^{\mathsf{full}}_{\mathsf{CPRF}}(\mathsf{A_f}, \mathsf{GGM}^{\mathsf{G}}, 1, q)$$

And thus, if $\mathsf{A_f}$ breaks the full security of $\mathsf{GGM}^{\mathsf{G}}$ with advantage ϵ then

$$|\Pr[H^\emptyset_{\langle 0\rangle} = 1] - \Pr[H^\emptyset_{\langle 10_n\rangle} = 1]| \geq \epsilon \ . \tag{6}$$

In the proof of Proposition 3 we were able to "connect" the real and random experiments H_0 and H_{2N-1} via intermediate hybrids $H_1, \ldots, H_{2N-2}$, such that from a distinguisher for any two consecutive hybrids we can build a distinguisher for G with the same advantage.

We did this by using random values (instead of applying G) in some steps along the path from the root $K_\emptyset$ to the challenge K_{x^*}. Here we cannot use the same approach to get hybrids in between $H^\emptyset_{\langle 0\rangle}$ and $H^\emptyset_{\langle 10_n\rangle}$, as these games consider full (and not selective) security where we learn x^* only at the very end, and thus "the path to x^*" is not even defined until the end of the experiment.

We could reduce the problem from the full to the selective setting by guessing x^* at the beginning like in the proof of Lemma 1, but this would lose us a factor of 2^N, which is what we want to avoid.

Instead of guessing the entire x^*, we will guess something easier: During the experiment $H_{\langle 0\rangle}$, we have to compute at most q children $K_{z\|0}\|K_{z\|1} = \mathsf{G}(K_z)$ of nodes at level $N/2 - 1$, i.e., $z \in \{0,1\}^{N/2-1}$. One of these K_z satisfies $z = x^*[1 \ldots N/2 - 1]$, and thus lies on the path from the root $K_\emptyset$ to the challenge K_{x^*} (potentially this happens at the very last query $x_q = x^*$). We randomly guess $q_{N/2} \xleftarrow{*} [q]$ for which invocation of G this will be the case. Note that we have to wait until $\mathsf{A_f}$ makes its last query $x_q = x^*$

Experiment $H_I^{\mathcal{P}}$

// $I \in \{\langle 0\rangle, \ldots, \langle 10_n\rangle\}$
// $\mathcal{P} = \{p_1, \ldots, p_t\} \subseteq \{1, \ldots, N-1\}$
// $\mathcal{S}_I \subseteq \mathcal{P} \cup \{0, N\}$, $\mathcal{S}_I$ as in eq.(15).
$\forall x \in \{0,1\}^{\leq N} \ : \ K_x := \bot$
$K_\emptyset \xleftarrow{*} \{0,1\}^\lambda$
// initialize counters
$\forall j = 1 \ldots N-1 \ : \ c_j = 0$
// make a random guess for each
// element in $\mathcal{P} = \{p_1, \ldots, p_t\}$
$\forall j \in [t] \ : \ q_{p_j} \xleftarrow{*} [q]$
// below $\mathsf{A_f}$ can make exactly q distinct
// oracle queries $x_1, \ldots, x_q$. The last
// (challenge) query $x_q = x^*$ must be in $\{0,1\}^N$.
$\mathsf{A_f}^{\mathcal{O}(.)}$
$\tilde{b} \leftarrow \mathsf{A_f}$
// only if guesses $q_{p_1}, \ldots, q_{p_t}$ were
// correct return $\tilde{b}$, otherwise return 0
if $\forall p \in \mathcal{P} \ : \ x^*[1 \ldots p-1] = z_{p-1}$ then
return $\tilde{b}$
else return 0 fi

$\mathcal{O}(x = x[1 \ldots \ell])$
// return K_x if it is already defined
if $K_x \neq \bot$ then return K_x fi
// get parent of K_x recursively
$K_{x[1\ldots\ell-1]} := \mathcal{O}(x[1 \ldots \ell-1])$
// increase counter for level $\ell - 1$
$c_{\ell-1} = c_{\ell-1} + 1$
// compute K_x and its sibling using G, unless its parent
// $K_{x[1\ldots\ell-1]}$ is a node which we guessed will be on the
// path from $K_\emptyset$ and K_{x^*} and as $\ell \in \mathcal{P}$ we must use a
// random value at this level OR this is the challenge
// query $x_q = x^*$ and $N \in I$, which means the answer
// to the challenge is random
if ($\ell \in \mathcal{P}$ and $c_{\ell-1} = q_{\ell-1}$) OR ($x = x_q$ and $N \in I$)
$K_{x[1\ldots\ell-1]\|0} \| K_{x[1\ldots\ell-1]\|1} \xleftarrow{*} U_{2\lambda}$
// store this node to later check if guess
// was correct
$z_{\ell-1} = x[1 \ldots \ell-1]$
elseif $x = x_q$ and $N \in I$

else
$K_{x[1\ldots\ell-1]\|0} \| K_{x[1\ldots\ell-1]\|1} := \mathsf{G}(K_{x[1\ldots\ell-1]})$
fi
return K_x

Fig. 2. Definition of the hybrid games from the proof of Theorem 1. The sets $\mathcal{S}_I$ are as in Equations (3) and (4). The hybrid $H_I^{\mathcal{P}}$ is defined like the full security game of a q-query adversary $\mathsf{A_f}$ against the constrained PRF $\mathsf{GGM}^{\mathsf{G}}$, but where we "guess", for any value in $p \in \mathcal{P}$, at which point in the experiment the node at depth p on the path from the root $K_\emptyset$ to the challenge K_{x^*} is computed (concretely, the guess is that it's the c_{p-1}th time we compute the children of an $p-1$ level node, we define the p level node $K_{x^*[1\ldots\ell]}$ on the path). At a subset of these points, namely $\mathcal{S}_I$, we embed random values. The final output is 0 unless all guesses were correct, in which case we forward $\mathsf{A_f}$'s output.

before we know if our guess was correct. If the guess was wrong, we output 0; otherwise we output $\mathsf{A_f}$'s output. The experiment just described corresponds to the hybrid $H_{\langle 0\rangle}^{\{N/2\}}$ as defined in Figure 2.

The games $H_{\langle 0\rangle}^{\{N/2\}}$ and $H_{\langle 10_n\rangle}^{\{N/2\}}$ behave *exactly* like $H_{\langle 0\rangle}^{\emptyset}$ and $H_{\langle 10_n\rangle}^{\emptyset}$, except for the final output, which in the former two hybrids is set to 0 with probability $1 - 1/q$, and left unchanged otherwise (namely, if our random guess $q_{N/2} \xleftarrow{*} [q]$ turns out to be correct, which we know after getting x^*). This implies

$$\Pr[H_{\langle 0\rangle}^{\{N/2\}} = 1] = \Pr[H_{\langle 0\rangle}^{\emptyset} = 1] \cdot \tfrac{1}{q} \quad \text{and} \quad \Pr[H_{\langle 10_n\rangle}^{\{N/2\}} = 1] = \Pr[H_{\langle 10_n\rangle}^{\emptyset} = 1] \cdot \tfrac{1}{q} \ ,$$

and with (6)

$$|\Pr[H_{\langle 0\rangle}^{\{N/2\}} = 1] - \Pr[H_{\langle 10_n\rangle}^{\{N/2\}} = 1]| \geq \epsilon/q \ . \tag{7}$$

What did we gain? We paid a factor q in the advantage for aborting when our guess $q_{N/2}$ was wrong. What we gained is that now we can assume that we know $x^*[1 \ldots N/2]$, i.e., the node halfway in between the root and the challenge (as, if the guess is wrong, we output 0 anyway).

We use this fact to define two new hybrids $H_{\langle 10_{n-1}\rangle}^{\{N/2\}}, H_{\langle 20_{n-1}\rangle}^{\{N/2\}}$ which are defined like $H_{\langle 0\rangle}^{\{N/2\}}, H_{\langle 10_n\rangle}^{\{N/2\}}$, respectively, but where the children of $K_{x^*[1\ldots N/2-1]}$ are uniformly random instead of being computed by applying G to $K_{x^*[1\ldots N/2-1]}$.

In Figure 1 (top left) we illustrate the path from $K_\emptyset$ to K_{x^*} in the hybrids $H^{\{4\}}_{\langle 0\rangle}, H^{\{4\}}_{\langle 100\rangle}, H^{\{4\}}_{\langle 200\rangle}, H^{\{4\}}_{\langle 1000\rangle}$ assuming the guessing was correct (a node with label i corresponds to $K_{x^*[1\ldots i]}$, blue nodes are sampled at random, and green ones by applying G to the parent).

By (7) we can distinguish the first from the last hybrid with advantage ϵ/q, and thus there are two consecutive hybrids in the sequence $H^{\{N/2\}}_{\langle 0\rangle}, H^{\{N/2\}}_{\langle 10_{n-1}\rangle}, H^{\{N/2\}}_{\langle 20_{n-1}\rangle}, H^{\{N/2\}}_{\langle 10_n\rangle}$ that we can distinguish with advantage at least $\epsilon/(3q)$. For concreteness, let us fix parameters $N = 8 = 2^3 = 2^n$ as in Figure 1 and assume that this is the case for the last two hybrids in the sequence, i.e.,

$$|\Pr[H^{\{4\}}_{\langle 200\rangle} = 1] - \Pr[H^{\{4\}}_{\langle 1000\rangle} = 1]| \geq \epsilon/(3q) \ . \tag{8}$$

The central observation here is that the above guessing step (losing a factor of q) followed by a hybrid argument (losing a factor of 3) transformed a distinguishing advantage ϵ for two hybrids $H^{\emptyset}_{\langle 0\rangle}, H^{\emptyset}_{\langle 1000\rangle}$ which have random values embedded along the path from $K_\emptyset$ to K_{x^*} on positions defined by N-neighboring sets (cf. Definition 2) $\mathcal{S}_{\langle 0\rangle}, \mathcal{S}_{\langle 1000\rangle}$, into a distinguishing advantage of $\epsilon/(3q)$ for two hybrids that correspond to $N/2$-neighboring sets, e.g. $\mathcal{S}_{\langle 200\rangle}$ and $\mathcal{S}_{\langle 1000\rangle}$.

We can now iterate this approach, in each iteration losing a factor $3q$ in distinguishing advantage, but getting hybrids that correspond to sets of half the neighboring distance. After $n = \log N$ iterations we end up with hybrids that correspond to 1-neighboring sets, and can be distinguished with advantage $\epsilon/(3q)^n$. We will make this formal in Lemma 3 below. From any distinguisher for (hybrids corresponding to) two 1-neighboring sets we get a distinguisher for G with the same advantage, as formally stated in Lemma 2 below. Let's continue illustrating the approach using the hybrids illustrated in Figure 1.

Recall that we assumed that we can distinguish $H^{\{4\}}_{\langle 200\rangle}$ and $H^{\{4\}}_{\langle 1000\rangle}$ as stated in eq.(8). We need to embed hybrids corresponding to the sets $\mathcal{S}_{\langle 210\rangle}, \mathcal{S}_{\langle 220\rangle}$ in between. Since $\mathcal{S}_{\langle 200\rangle} \mathbin{\Delta} \mathcal{S}_{\langle 1000\rangle} = \{4\}$, by eq.(4) for $\ell = 1$, we construct $\mathcal{S}_{\langle 200\rangle + \langle 10\rangle} = \mathcal{S}_{\langle 200\rangle} \cup \{4 - \frac{8}{2^2} = 2\}$ and $\mathcal{S}_{\langle 1000\rangle - \langle 10\rangle} = \mathcal{S}_{\langle 1000\rangle} \cup \{2\}$ (see top right of Figure 1). We add this new element $\{2\}$ to the "guessing set" $\{4\}$, at the price of losing a factor q in distinguishing advantage compared to eq.(8):

$$|\Pr[H^{\{2,4\}}_{\langle 200\rangle} = 1] - \Pr[H^{\{2,4\}}_{\langle 1000\rangle} = 1]| \geq \epsilon/(3q^2) \ . \tag{9}$$

We can now consider the sequence of hybrids $H^{\{2,4\}}_{\langle 200\rangle}, H^{\{2,4\}}_{\langle 210\rangle}, H^{\{2,4\}}_{\langle 220\rangle}, H^{\{2,4\}}_{\langle 1000\rangle}$. Two consecutive hybrids can be distinguished with advantage $\epsilon/(3^2q^2)$; assume this is the case for the middle two.

$$|\Pr[H^{\{2,4\}}_{\langle 210\rangle} = 1] - \Pr[H^{\{2,4\}}_{\langle 220\rangle} = 1]| \geq \epsilon/(3^2q^2) \ . \tag{10}$$

Now $\mathcal{S}_{\langle 210\rangle} \mathbin{\Delta} \mathcal{S}_{\langle 220\rangle} = \{4\}$, and $4 - 8/2^3 = 3$, so we add $\{3\}$ to the guessing set losing another factor q:

$$|\Pr[H^{\{2,3,4\}}_{\langle 210\rangle} = 1] - \Pr[H^{\{2,3,4\}}_{\langle 220\rangle} = 1]| \geq \epsilon/(3^2q^3) \ , \tag{11}$$

and can now consider the games $H^{\{2,3,4\}}_{\langle 210\rangle}, H^{\{2,3,4\}}_{\langle 211\rangle}, H^{\{2,3,4\}}_{\langle 212\rangle}, H^{\{2,3,4\}}_{\langle 220\rangle}$ as shown at the bottom in Figure 1. Two consecutive hybrids in this sequence can be distinguished with advantage at least $1/3$ of the advantage we had for the first and last hybrid in this sequence, let's assume this is the case for the last two, then:

$$|\Pr[H^{\{2,3,4\}}_{\langle 212\rangle} = 1] - \Pr[H^{\{2,3,4\}}_{\langle 220\rangle} = 1]| \geq \epsilon/(3^3q^3) \ . \tag{12}$$

We have shown the existence of two games $H^{\mathcal{P}}_I$ and $H^{\mathcal{P}}_{I+\langle 1\rangle}$ (what $\mathcal{P}$ and I are exactly is irrelevant for the rest of the argument) that can be distinguished with advantage $\epsilon/(3q)^n$. By the following lemma (proven in Appendix B.4), this implies that we can break the security of G with the same advantage.

Lemma 2. *For any $I \in \{\langle 0\rangle, \ldots, \langle 2_n\rangle\}, \mathcal{P} \subset \{1, \ldots, N-1\}$ where $\mathcal{S}_I \cup \mathcal{S}_{I+\langle 1\rangle} \subseteq \mathcal{P} \cup \{0, N\}$ (so the games $H^{\mathcal{P}}_{I+\langle 1\rangle}, H^{\mathcal{P}}_I$ are defined) the following holds. If*

$$|\Pr[H^{\mathcal{P}}_I = 1] - \Pr[H^{\mathcal{P}}_{I+\langle 1\rangle} = 1]| = \delta$$

then G is not a (δ, s)-secure PRG for $s = |\mathsf{A_f}| - O(q \cdot N \cdot |\mathsf{G}|)$.

Lemma 3. *For $\ell \in \{0, \ldots, n-1\}$, any consecutive ℓ-level sets $\mathcal{S}_I, \mathcal{S}_{I'}$ (i.e., $I, I' \in \{\langle 0\rangle, \ldots, \langle 10_n\rangle\}$ are of the form $\langle ?0_{n-\ell}\rangle$ and $I' = I + \langle 10_{n-\ell}\rangle$) and any $\mathcal{P}$ for which the hybrids $H^{\mathcal{P}}_I, H^{\mathcal{P}}_{I'}$ are defined (which is the case if $\mathcal{S}_I \cup \mathcal{S}_{I'} \subseteq \mathcal{P} \cup \{0, N\}$), the following holds. If*

$$|\Pr[H^{\mathcal{P}}_I = 1] - \Pr[H^{\mathcal{P}}_{I'} = 1]| = \delta \tag{13}$$

then for some consecutive $(\ell+1)$-level sets J and $J' = J + \langle 10_{n-\ell-1}\rangle$ and some $\mathcal{P}'$

$$|\Pr[H^{\mathcal{P}'}_J = 1] - \Pr[H^{\mathcal{P}'}_{J'} = 1]| = \delta/(3q) \ .$$

The proof of Lemma 3 is in Appendix B.5. The theorem now follows from Lemmata 2 and 3 as follows. Assume a q-query adversary $\mathsf{A_f}$ breaks the full security of $\mathsf{GGM}^{\mathsf{G}}$ with advantage ϵ, which, as explained in the paragraph before eq.(6), means that we can distinguish the two 0-level hybrids $H^{\emptyset}_{\langle 0\rangle}$ and $H^{\emptyset}_{\langle 10_n\rangle}$ with advantage ϵ. Applying Lemma 3 n times, we get that there exist consecutive n-level hybrids $H^{\mathcal{P}}_I, H^{\mathcal{P}}_{I+\langle 1\rangle}$ that can be distinguished with advantage $\epsilon/(3q)^n$, which by Lemma 2 implies that we can break the security of G with the same advantage $\epsilon/(3q)^n$. This concludes the proof of Theorem 1.

To reduce the loss to $2q \log q \cdot (3q)^{n - \log\log q}$ as stated below Theorem 1, we use the same proof as above, but stop after $n - \log\log q$ (instead of n) iterations. At this point, we have lost a factor $(3q)^{n-\log\log q}$, and have constructed games that are $(\log q)$-neighboring. We can now use a proof along the lines of the proof of Proposition 3, and guess the entire remaining path of length $\log q$ at once. This step loses a factor $q 2\log q$ (a factor $q = 2^{\log q}$ to guess the path, and another $2\log q$ as we have a number of hybrids which is twice the length of the path).

5 Impossibility Result for Prefix-Fixing Boneh-Waters PRF

In this section we show that we cannot hope to prove full security without an exponential loss for another constrained PRF, namely the one due to Boneh and Waters [BW13].

5.1 The Boneh-Waters Constrained PRF

Leveled κ-linear maps. The Boneh-Waters constrained PRF [BW13] is based on leveled multilinear maps [GGH13,CLT13], of which they use the following abstraction:

We assume a *group generator* $\mathcal{G}$ that takes as input a security parameter 1^λ and the number of levels $\kappa \in \mathbb{N}$ and outputs a sequence of groups $(\mathbb{G}_1, \ldots, \mathbb{G}_\kappa)$, each of prime order $p > 2^\lambda$, generated by g_i, respectively, such that there exists a set of bilinear maps $\{e_{i,j} \colon \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j} \,|\, i, j \geq 1; i + j \leq \kappa\}$ with

$$\forall a, b \in \mathbb{Z}_p : e_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab} \ .$$

(For simplicity we will omit the indices of e.) Security of the PRF is based on the following assumption:

The *κ-multilinear decisional Diffie-Hellman assumption* states that given the output of $\mathcal{G}(1^\lambda, \kappa)$ and $(g_1, g_1^{c_1}, \ldots, g_1^{c_{\kappa+1}})$ for random $(c_1, \ldots, c_{\kappa+1}) \xleftarrow{*} \mathbb{Z}_p^{\kappa+1}$, it is hard to distinguish $(g_\kappa)^{\prod_{j\in[\kappa+1]} c_j}$ from a random element in $\mathbb{G}_\kappa$ with better than negligible advantage in λ.

The Boneh-Waters bit-fixing PRF. Boneh and Waters [BW13] define a PRF with domain $\mathcal{X} = \{0,1\}^N$ and range $\mathcal{Y} = \mathbb{G}_\kappa$, where $\kappa = N+1$. The sets $S \subseteq \mathcal{X}$ for which constrained keys can be derived are subsets of $\mathcal{X}$ where certain bits are fixed; a set S is described by a vector $v \in \{0,1,?\}^N$ (where '?' acts as a wildcard) as $S_v := \{x \in \{0,1\}^N \mid \forall i \in [N] : (v_i = ?) \vee (x_i = v_i)\}$.

The PRF is set up for domain $\mathcal{X} = \{0,1\}^N$ by running $\mathcal{G}(1^\lambda, N+1)$ to generate a sequence of groups $(\mathbb{G}_1, \ldots, \mathbb{G}_{N+1})$. We let g denote the generator of $\mathbb{G}_1$. Secret keys are random elements

$$k = (\alpha, d_{1,0}, d_{1,1}, \ldots, d_{N,0}, d_{N,1}) \in \mathbb{Z}_p^{2N+1} =: \mathcal{K}$$

and the PRF is defined as

$$\mathsf{F} \colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y} \qquad (k, x) \mapsto (g_{N+1})^{\alpha \prod_{i \in [N]} d_{i,x_i}}$$

$\mathsf{F.constrain}(k, v)$**:** On input a key $k = (\alpha, \{d_{i,b}\}_{i \in [N],\, b \in \{0,1\}})$ and $v \in \{0,1,?\}^N$ describing the constrained set, output the key $\big(v, K, \{D_{i,b}\}_{i \in [N] \setminus V,\, b \in \{0,1\}}\big)$, with $V := \{i \in [N] \mid v_i \neq ?\}$ the set of fixed indices and

$$K := (g_{|V|+1})^{\alpha \prod_{i \in V} d_{i,v_i}} \qquad D_{i,b} := g^{d_{i,b}} \quad \text{for } i \in [N] \setminus V,\ b \in \{0,1\}$$

$\mathsf{F.eval}(k_v, x)$**:** Let $k_v = (v, K, \{D_{i,b}\}_{i \in [N] \setminus V,\, b \in \{0,1\}})$, where $V = \{i \in [N] \mid v_i \neq ?\}$.

- If for some $i \in V$: $x_i \neq v_i$ then return $\perp$ (as x is not in S_v).
- If $|V| = N$ then output K (as $S_v = \{v\}$ and $K = \mathsf{F}(k, v)$).
- Else, compute $T := (g_{N-|V|})^{\prod_{i \in [N] \setminus V} d_{i,x_i}}$ via repeated application of the bilinear maps to the elements $D_{i,x_i} = g^{d_{i,x_i}}$ for $i \in [N] \setminus V$ and output $e(T, K) = (g_{N+1})^{\alpha \prod_{i \in [N]} d_{i,x_i}} = \mathsf{F}(k, x)$.

In [BW13] it is shown how to use an adversary breaking the constrained PRF for N-bit inputs with advantage $\epsilon(\lambda)$ to break the $(N+1)$-multilinear decisional Diffie-Hellman assumption with advantage $\frac{1}{2^N} \cdot \epsilon(\lambda)$. In the next section we show that this is optimal in the sense that every simple reduction from a decisional problem must lose a factor which is exponential in the input length N.

5.2 Adaptive Security of the Boneh-Waters CPRF

Lewko and Waters [LW14], following earlier work [Cor02,HJK12], show that it is hard to prove full security for hierarchical identity-based encryption (HIBE) [HL02] schemes if one can check whether secret keys and ciphertexts are correctly formed w.r.t. the public parameters. In particular, they show that a simple black-box reduction (that is, one that just runs the attacker once without rewinding; see below) from a decisional assumption must lose a factor that is exponential in the depth of the hierarchy. We adapt their proof technique to show that a proof of full security of the Boneh-Waters PRF with constrained keys for prefix-fixing must lose a factor that is exponential in the length of the PRF inputs.

The proof idea in [LW14] is the following: Assume that there exists a reduction which breaks a challenge with some probability δ after interacting with an adversary that breaks the security notion with some probability ϵ. Define a concrete adversary A, which, after receiving the public parameters, guesses a random identity *id* at the lowest level of the hierarchy and then queries the keys for all identities except *id*, checking whether they are consistent with the parameters. By rewinding the reduction to the point after it output the parameters and simulating A again, choosing a fresh random identity *id*′, one can now simulate a successful adversary by using a key for *id*′ from the first run. It is crucial that the keys can be verified w.r.t. the parameters. The reduction can thus be used to break the challenge by simulating the adversary. We formally define decisional problems and simple reductions, following [LW14].

Definition 7. *A non-interactive* **decisional problem** $\Pi = (C, \mathcal{D})$ *is described by a set of challenges* C *and a distribution* $\mathcal{D}$ *on* C*. Each* $c \in C$ *is associated with a bit* $b(c)$*, the solution for challenge* c*. An algorithm* A (ϵ, t)*-solves* Π *if* A *runs in time at most* t *and*

$$\Pr_{c \xleftarrow{\mathcal{D}} C}\left[b(c) \leftarrow \mathsf{A}(c)\right] \geq \tfrac{1}{2} + \epsilon \ .$$

Definition 8. *An algorithm* $\mathcal{R}$ *is a* **simple** $(t, \epsilon, q, \delta, t')$**-reduction** *from a decisional problem* Π *to breaking unpredictability of a CPRF if, when given black-box access to any adversary* A *that* (t, ϵ, q)*-breaks unpredictability,* $\mathcal{R}$ (δ, t')*-solves* Π *after simulating the unpredictability game once for* A*.*

We show that every simple reduction from a decisional problem to unpredictability for the Boneh-Waters CPRF must lose at least a factor exponential in N, where unpredictability is defined as follows:

Definition 9. *Consider the following experiment for a constrained PRF* $(\mathsf{F}, \mathsf{F.constrain}, \mathsf{F.eval})$*:*

- *The challenger chooses* $k \xleftarrow{*} \mathcal{K}$*;*
- A *can query* $\mathsf{F.constrain}$ *for sets* $\mathcal{S}_i$*;*
- A *wins if it outputs* $(x, \mathsf{F}(k, x))$ *with* $x \in \mathcal{X}$ *and* $x \notin \mathcal{S}_i$ *for all queried* $\mathcal{S}_i$*.*

The CPRF is (ϵ, t, q)**-unpredictable** *if no* A *running in time at most* t *making at most* q *queries can win the above game with probability greater than* ϵ*.*

Since for superpolynomial-size domains $\mathcal{X}$, unpredictability follows from pseudorandomness (without any security loss), our impossibility result holds a forteriori for pseudorandomness. In particular, this precludes security proofs for the Boneh-Waters CPRF using the technique from Section 4.

Instead of checking validity of keys computed by the reduction w.r.t. the public parameters, as in [LW14], we show that after two concrete constrained-key queries, the secret key k used by the reduction is basically fixed (the two received keys are thus a "fingerprint" of the secret key). Moreover, correctness of any other key can be verified w.r.t. to this fingerprint thanks to the multilinear map. We define an adversary A that we can simulate by rewinding the reduction: After making the fingerprint queries, A chooses a random value $x \in \mathcal{X}$ and queries keys which allow it to evaluate all other domain points, checking every key is consistent with the fingerprint. By rewinding the reduction to the point after receiving the fingerprint and choosing a different $x' \neq x$, we can break security by using one of the keys obtained in a previous run to evaluate the function at x'.

Theorem 2. *Let* $\Pi(\lambda)$ *be a decisional problem such that no algorithm running in time* $t = poly(\lambda)$ *has an advantage non-negligible in* λ*. Let* $\mathcal{R}$ *be a simple* $(t, \epsilon, q, \delta, t')$ *reduction from* Π *to unpredictability of the Boneh-Waters prefix-constrained PRF with domain* $\{0,1\}^N$*, with both* t, t' *polynomial in* λ*, and* $q \geq N - 1$*. Then* δ *vanishes exponentially as a function of* N *(up to terms that are negligible in* λ*).*

The proof can be found in Appendix C.

References

[AFL12] Michel Abdalla, Dario Fiore, and Vadim Lyubashevsky. From selective to full security: Semi-generic transformations in the standard model. In *Public Key Cryptography*, pages 316–333, 2012.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, May 2004.

[BCPR13] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. Indistinguishability obfuscation vs. auxiliary-input extractable functions: One must fall. Cryptology ePrint Archive, Report 2013/641, 2013. http://eprint.iacr.org/.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, August 2001.

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

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 59–71. Springer, May / June 1998.

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

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 476–493. Springer, August 2013.

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

[Fis12] Marc Fischlin. Black-box reductions and separations in cryptography (invited talk). In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *AFRICACRYPT 12*, volume 7374 of *LNCS*, pages 413–422. Springer, July 2012.

[FS10] Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 197–215. Springer, May 2010.

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, May 2013.

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

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

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

[HL02] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 466–481. Springer, April / May 2002.

[Hof14] Dennis Hofheinz. Fully secure constrained pseudorandom functions using random oracles, 2014.

[HSW14] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 201–220. Springer, May 2014.

[HW09] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 654–670. Springer, August 2009.

[KPTZ13] 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*, pages 669–684. ACM Press, November 2013.

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

[SW13] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. Cryptology ePrint Archive, Report 2013/454, 2013. http://eprint.iacr.org/.

A Hybrid Proofs

In this section we show a simple application of the hybrid technique to prove security (i.e., the pseudorandomness of the output) of the "stream-cipher" we get when iterating a pseudorandom generator. The simple proofs in this section already exemplify some of the techniques that we'll use in the proof of Proposition 3 and Theorem 1.

Given a function $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$, we define the function $\mathsf{SC}^{\mathsf{G}}\colon \mathbb{N} \times \{0,1\}^\lambda \to \{0,1\}^*$ as

$$\mathsf{SC}^{\mathsf{G}}(N, K_0) = (X_1, \ldots, X_N), \text{ where for } i \geq 1 \ : \ (K_i, X_i) \leftarrow \mathsf{G}(K_{i-1}) \ .$$

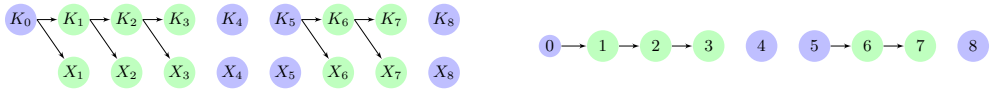

Fig. 3. The left picture shows the evaluation of $\mathsf{SC}^{\mathsf{G}}_{\{0,4,5,8\}}(8)$. The output is $X_1, \ldots, X_8$. The arrows indicate the evaluation of G, e.g., $(K_1, X_1) \leftarrow \mathsf{G}(K_0)$. The blue values are sampled uniformly at random. The right picture illustrates the corresponding compact representation we will use.

For $\mathcal{S} \subset \mathbb{N}^0 = \{0, 1, \ldots\}$, we denote with $\mathsf{SC}^{\mathsf{G}}_{\mathcal{S}}(N)$ the random variable that has the output distribution of $\mathsf{SC}^{\mathsf{G}}(N, K_0)$ instantiated with a random key K_0, but where for every $i \in \mathcal{S}$, the output in the i-th round is replaced with a uniformly random value, i.e.,

$$\text{for } N \in \mathbb{N}, \mathcal{S} \subset \mathbb{N}^0 : \quad \mathsf{SC}^{\mathsf{G}}_{\mathcal{S}}(N) \to (X_1, \ldots, X_N) \tag{14}$$

$$\text{where} \quad K_0 \xleftarrow{*} \{0,1\}^\lambda \quad \text{and for } i \geq 1 \quad \begin{cases} (K_i, X_i) \leftarrow \mathsf{G}(K_{i-1}) & \text{if } i \notin \mathcal{S} \\ (K_i, X_i) \xleftarrow{*} \{0,1\}^{2\lambda} & \text{otherwise} \end{cases}$$

It will be convenient to require that 0 is always contained in $\mathcal{S}$ (which makes sense as K_0 is always random). In Figure 3 we illustrate the evaluation of $\mathsf{SC}^{\mathsf{G}}_{\{0,4,5,8\}}(8)$. Note that

$$\mathsf{SC}^{\mathsf{G}}_{\{0\}}(N) \sim \mathsf{SC}^{\mathsf{G}}(N, U_\lambda) \quad \text{and} \quad \mathsf{SC}^{\mathsf{G}}_{\{0,\ldots,N\}}(N) \sim U_{N\lambda} \ .$$

Definition 10. $\mathsf{SC}^{\mathsf{G}} \colon \mathbb{N} \times \{0,1\}^\lambda \to \{0,1\}^*$ *is* (N, ϵ', s')**-pseudorandom** *if no circuit of size* s' *can distinguish with advantage greater than* ϵ' *the first* N *blocks of output of* SC^{G} *from random when instantiated with a random key, i.e.,*

$$\mathsf{SC}^{\mathsf{G}}(N, U_\lambda) \sim_{(\epsilon', s')} U_{N\lambda}$$

We say that SC^{G} *is* (N, ϵ', s') **next-block pseudorandom** *if, for any* $N' \leq N$*, no circuit of size* s' *can distinguish the* N'*-th output block from random given the first* $N' - 1$ *blocks, i.e.,*

$$\mathsf{SC}^{\mathsf{G}}(N', U_\lambda) \sim_{(\epsilon', s')} \mathsf{SC}^{\mathsf{G}}_{\{0,N'\}}(N') \sim \mathsf{SC}^{\mathsf{G}}(N' - 1, U_\lambda) \| U_\lambda \ .$$

For 1-neighboring we simply say neighboring. To prove the (next-block) pseudorandomness of SC^{G} we will use a hybrid argument. By the following lemma, two neighboring hybrids (by which we mean $\mathsf{SC}^{\mathsf{G}}_{\mathcal{S}}(N), \mathsf{SC}^{\mathsf{G}}_{\mathcal{S}'}(N)$ for neighboring $\mathcal{S}, \mathcal{S}'$) are indistinguishable if G is pseudorandom.

Lemma 4. *For any* $N \in \mathbb{N}^+$ *and two neighboring sets* $\mathcal{S} \subset \mathcal{S}' \subseteq [N]$

$$\mathsf{G}(U_\lambda) \sim_{(\epsilon, s)} U_{2\lambda} \quad \Rightarrow \quad \mathsf{SC}^{\mathsf{G}}_{\mathcal{S}}(N) \sim_{(\epsilon, s')} \mathsf{SC}^{\mathsf{G}}_{\mathcal{S}'}(N)$$

where $s' \approx s - N|\mathsf{G}|$.

Proof. We assume w.l.o.g. that $|\mathcal{S}'| > |\mathcal{S}|$. Given a PRG challenge $C \in \{0,1\}^{2\lambda}$, we can sample a variable X s.t.

$$X \sim \mathsf{SC}^{\mathsf{G}}_{\mathcal{S}}(N) \quad \text{if} \quad C \sim \mathsf{G}(U_\lambda) \quad \text{but} \quad X \sim \mathsf{SC}^{\mathsf{G}}_{\mathcal{S}'}(N) \quad \text{if} \quad C \sim U_{2\lambda}$$

as follows. Let $d \in \mathcal{S}'$ be the (unique) element not in $\mathcal{S}$. Now sample $\mathsf{SC}^{\mathsf{G}}_{\mathcal{S}}(N)$ as in (14), except that in the d-th step we use $(K_d, X_d) := C$. (Note that (K_{d-1}, X_{d-1}) is random as per 2. in Definition 2.) Thus, from any distinguisher for $\mathsf{SC}^{\mathsf{G}}_{\mathcal{S}}(N)$ and $\mathsf{SC}^{\mathsf{G}}_{\mathcal{S}'}(N)$, we get a distinguisher for the PRG G with the same advantage. □

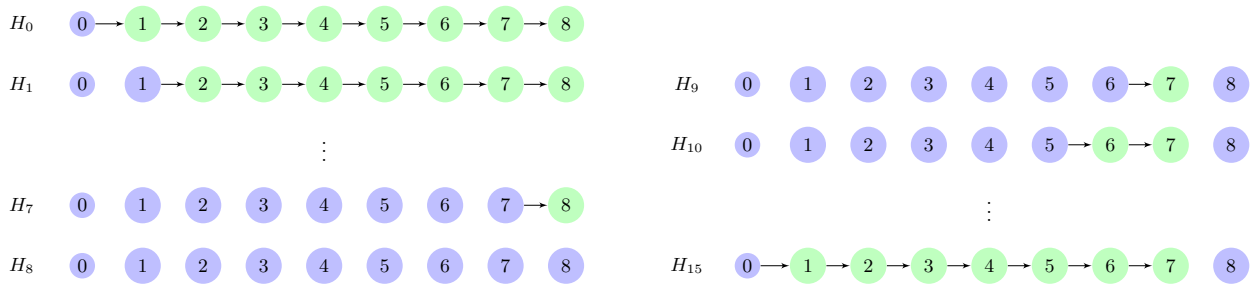

Fig. 4. The hybrids $H_0, \ldots, H_N$ (for $N = 8$) are as defined in the proof of Proposition 5. The proof of Proposition 6 additionally uses the hybrids $H_{N+1}, \ldots, H_{2N-1}$.

We will also use the triangle inequality for indistinguishability

Proposition 4. *Consider any random variables $H_0, H_1, \ldots, H_N$ then*

$$H_0 \not\approx_{(\epsilon,s)} H_N \Rightarrow \exists i \in [N] \ : \ H_{i-1} \not\approx_{(\frac{\epsilon}{N},s)} H_i$$

Proposition 5. *If $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is an (ϵ, s)-secure PRG, then SC^{G} is (N, ϵ', s') pseudorandom with*

$$\epsilon' = \epsilon \cdot N \qquad s' \approx s - N|\mathsf{G}|$$

Proof. Consider the hybrids $H_0, \ldots, H_N$ where $H_i = \mathsf{SC}^{\mathsf{G}}_{[i]_0}(N)$ (as illustrated in Figure 4 for $N = 8$; the hybrids $H_9, \ldots, H_{15}$ in the figure are not needed in this proof). Assume for contradiction that $\mathsf{SC}^{\mathsf{G}}_{\{0\}}(N)$ is not $(\epsilon N, s')$ indistinguishable from $\mathsf{SC}^{\mathsf{G}}_{[N]_0}(N)$, i.e.,

$$H_0 \not\approx_{(\epsilon N, s')} H_N \ .$$

Then by Proposition 4, for some $i \in [N]$

$$H_{i-1} \not\approx_{(\epsilon,s')} H_i \ .$$

Since $[i-1]_0$ and $[i]_0$ are neighboring sets, applying Lemma 4, we get,

$$\mathsf{G}(U_\lambda) \not\approx_{(\epsilon, s' + N|\mathsf{G}|)} U_{2\lambda} \ ,$$

contradicting (ϵ, s)-security of G. □

Proposition 6. *If $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is an (ϵ, s)-secure PRG, then SC^{G} is (N, ϵ', s')* **next-block** *pseudorandom with*

$$\epsilon' = \epsilon \cdot \tfrac{1}{2N-1} \qquad s' \approx s - (2N-1)|\mathsf{G}|$$

We omit the proof as it is almost identical to the proof of Proposition 5, except that we use hybrids H_0 to H_{2N-1} (not just H_N) as illustrated in Figure 4.

Here each hybrid H_i corresponds to the variable $\mathsf{SC}^{\mathsf{G}}_{\mathcal{S}_i}(N)$, for $\mathcal{S}_0, \ldots, \mathcal{S}_{2N-1} \subseteq [N]$ where $\mathcal{S}_0 = \{0\}, \mathcal{S}_{2N-1} = \{0, N\}$ and for any i, the sets $\mathcal{S}_{i-1}$ and $\mathcal{S}_i$ are neighboring. Concretely,

$$\begin{aligned} \mathcal{S}_0 &= \{0\} \\ \mathcal{S}_i &= \{0, 1, \ldots, i\} && \text{for } i \in \{1, \ldots, N\} \\ \mathcal{S}_i &= \{0, 1, \ldots, 2N-i-1, N\} && \text{for } i \in \{N+1, \ldots, 2N-2\} \\ \mathcal{S}_{2N-1} &= \{0, N\} \end{aligned} \tag{15}$$

B Omitted Proofs

B.1 Proof of Lemma 1

From any adversary $\mathsf{A_f}$ against the full security of F we can construct an adversary $\mathsf{A_s}$ (of basically the same size) against the selective security of F losing a factor of $|\mathcal{X}|$ in the advantage, i.e.,

$$\mathsf{Adv}^{\mathsf{sel}}_{\mathsf{F}}(\mathsf{A_s}, q) = \tfrac{1}{|\mathcal{X}|} \cdot \mathsf{Adv}^{\mathsf{full}}_{\mathsf{F}}(\mathsf{A_f}, q) \tag{16}$$

as follows. $\mathsf{A_s}$ initially simply outputs a random $x' \xleftarrow{*} \mathcal{X}$ in the selective security game. It then runs $\mathsf{A}_{\mathsf{f}}^{\mathcal{O}(.)}$, which outputs some x^*. If $x^* = x'$ then $\mathsf{A_s}$ uses $\mathsf{A_f}$ for the rest of the experiment, i.e., it forwards C_b to $\mathsf{A_f}$, and then returns the bit $\tilde{b}$ that $\mathsf{A_f}$ outputs. If $x^* \neq x'$ then $\mathsf{A_s}$ answers with $\tilde{b} = 0$. Thus, $\mathsf{A_s}$ outputs 0 with probability $1 - \frac{1}{|\mathcal{X}|}$, and whatever $\mathsf{A_f}$ outputs otherwise. Let ϵ (possibly negative) be such that

$$\Pr_{b \leftarrow \{0,1\}}\left[\mathbf{Exp}^{\mathsf{full}}_{\mathsf{CPRF}}(\mathsf{A_f}, \mathsf{F}, b, q) = b\right] = \tfrac{1}{2} + \epsilon \ . \tag{17}$$

Then

$$\begin{aligned}
&\Pr_{b \leftarrow \{0,1\}}\left[\mathbf{Exp}^{\mathsf{sel}}_{\mathsf{CPRF}}(\mathsf{A_s}, \mathsf{F}, b, q) = b\right] \\
&\quad = \Pr_{b \leftarrow \{0,1\}}\left[\mathbf{Exp}^{\mathsf{sel}}_{\mathsf{CPRF}}(\mathsf{A_s}, \mathsf{F}, b, q) = b \mid x^* = x'\right]\Pr[x^* = x'] + \\
&\qquad\qquad \Pr_{b \leftarrow \{0,1\}}\left[\mathbf{Exp}^{\mathsf{sel}}_{\mathsf{CPRF}}(\mathsf{A_s}, \mathsf{F}, b, q) = b \mid x^* \neq x'\right]\Pr[x^* \neq x'] \\
&\quad = \Pr_{b \leftarrow \{0,1\}}[\mathbf{Exp}^{\mathsf{full}}_{\mathsf{CPRF}}(\mathsf{A_f}, \mathsf{F}, b, q) = b] \cdot \tfrac{1}{|\mathcal{X}|} + \tfrac{1}{2} \cdot \left(1 - \tfrac{1}{|\mathcal{X}|}\right) \\
&\quad = \left(\tfrac{1}{2} + \epsilon\right)\tfrac{1}{|\mathcal{X}|} + \tfrac{1}{2} \cdot \left(1 - \tfrac{1}{|\mathcal{X}|}\right) \ = \ \tfrac{1}{2} + \tfrac{\epsilon}{|\mathcal{X}|}
\end{aligned}$$

By (1) this means $2|\epsilon|\frac{1}{|\mathcal{X}|} = \mathsf{Adv}^{\mathsf{sel}}_{\mathsf{F}}(\mathsf{A_s}, q)$; on the other hand (1) and (17) give $2|\epsilon| = \mathsf{Adv}^{\mathsf{full}}_{\mathsf{F}}(\mathsf{A_f}, q)$, which proves (16).

B.2 Proof of Proposition 2

We consider two hybrid games H_0 and H_m, which will correspond to the experiments

$$H_0 \sim \mathsf{A}^{\mathsf{F}(K,.)} \quad \text{and} \quad H_m \sim \mathsf{A}^{f(.)} \quad \text{where} \quad K \xleftarrow{*} \mathcal{K}, f \xleftarrow{*} \mathcal{R}[\mathcal{X}, \mathcal{Y}] \ .$$

For the ease of describing the higher hybrids, we describe H_0 as follows. In H_0 we begin by initially defining $K_x = \perp$ for all $x \in \{0,1\}^*$ and then sampling $K_\emptyset \xleftarrow{*} \mathcal{K}$. We then invoke A, who makes queries $x_1, x_2, \ldots$ (of total length at most m), where we answer each query $x[1 \ldots \ell]$ with K_x which is defined as follows: determine the largest ℓ' s.t. $K_{x[1\ldots\ell']} \neq \perp$, and then for $j = \ell' + 1$ to ℓ recursively define $K_{x[1\ldots j-1\|0]} \| K_{x[1\ldots j-1\|1]} := \mathsf{G}(K_{x[1\ldots j-1]})$. The final output of H_0 is whatever A finally outputs. We just emulated $\mathsf{F}(K,.)$ for A, and thus $H_0 \sim \mathsf{A}^{\mathsf{F}(K,.)}$.

Now for any $i \geq 0$, we define the experiment H_i to be the same as the experiment H_0, except that we replace the outputs of the first i invocations of G with uniformly random values. We have $H_m \sim \mathsf{A}^{f(.)}$, as all the outputs A gets in H_m are uniformly random, exactly like the outputs of $f(.)$.

It follows that if A can distinguish $\mathsf{F}(K,.)$ from a random function (i.e., distinguish H_0 from H_m) with advantage ϵ, there are two hybrids H_i, H_{i+1} s.t.

$$|\Pr[1 \leftarrow H_i] - \Pr[1 \leftarrow H_{i+1}]| \geq \tfrac{\epsilon}{m}$$

Using this, we can distinguish the output $\mathsf{G}(U_\lambda)$ from a random $U_{2\lambda}$ with the same advantage: given a challenge C, simulate the experiment H_i up to the $(i+1)$-th invocation of G, and replace its output with C. If $C = U_{2\lambda}$, this emulates experiment H_{i+1}, and if $C = \mathsf{G}(U_\lambda)$, this emulates the experiment H_i.

Experiment H_i
$\forall x \in \{0,1\}^{\leq N} \ : \ K_x := \bot$
$K_\emptyset \xleftarrow{*} \{0,1\}^\lambda$
$x^* \leftarrow \mathsf{A_s}$
// below, $\mathsf{A_s}$ can make max. q queries
// and last query must be x^*
$\mathsf{A_s}^{\mathcal{O}(.)}$
$\tilde{b} \leftarrow \mathsf{A_s}$
return $\tilde{b}$

$\mathcal{O}(x = x[1 \dots \ell])$
// return K_x if it is already defined
if $K_x \neq \bot$ then return K_x fi
// get parent of K_x recursively
$K_{x[1\dots\ell-1]} := \mathcal{O}(x[1 \dots \ell - 1])$
// compute K_x and its sibling using G, unless $x[1 \dots \ell - 1]$ is a prefix
// of x^* and $|x| \in \mathcal{S}_i$, in this case use random values.
if $x[1 \dots \ell - 1] = x^*[1 \dots \ell - 1]$ and $\ell \in \mathcal{S}_i$ then
$K_{x[1\dots\ell-1]\|0} \| K_{x[1\dots\ell-1]\|1} \xleftarrow{*} U_{2\lambda}$
else
$K_{x[1\dots\ell-1]\|0} \| K_{x[1\dots\ell-1]\|1} := \mathsf{G}(K_{x[1\dots\ell-1]})$
if
return K_x

Fig. 5. Definition of the hybrid games $H_0, \dots, H_{2N-1}$ from the proof of Proposition 3, where $\mathcal{S}_i$ are as in eq.(15).

B.3 Proof of Proposition 3

Full security (as stated in Item 2.) follows from selective security (as stated in Item 1.) by complexity leveraging as in Lemma 1.

To prove selective security, we will use hybrid games $H_0, \dots, H_{2N-1}$ as formally defined in Figure 5. By inspection we see that the first and last hybrids defined in Figure 5 are exactly the real and random selective security game, i.e.,

$$H_0 \sim \mathbf{Exp}^{\mathsf{sel}}_{\mathsf{CPRF}}(\mathsf{A_s}, \mathsf{GGM}^{\mathsf{G}}, 0, q) \quad \text{and} \quad H_{2N-1} \sim \mathbf{Exp}^{\mathsf{sel}}_{\mathsf{CPRF}}(\mathsf{A_s}, \mathsf{GGM}^{\mathsf{G}}, 1, q)$$

The other hybrids correspond to games where we sometimes use uniformly random values instead the output of G on the path from the root $K_\emptyset$ to the challenge output K_{x^*}. More precisely, in H_i we use random values in the j-th step along the path computing the output for x^* for all $j \in \mathcal{S}_i$, with $\mathcal{S}_i$ as in eq.(15) and illustrated in Figure 4.

From any distinguisher for two neighboring hybrids H_i, H_{i+1} we get a distinguisher for G with the same advantage as follows. $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ differ by exactly one element d (and $d-1 \in \mathcal{S}_i, d-1 \in \mathcal{S}_{i+1}$). Given a PRG challenge C, simulate the experiment H_i up to the point where $\mathcal{O}(.)$ (as in Fig.5) is queried for the first time on a query x where $x[1 \dots d-1] = x^*[1 \dots d-1]$. At this point, we embed the challenge $K_{x[1\dots d-1]\|0} \| K_{x[1\dots d-1]\|1} := C$. Depending on whether $C \xleftarrow{*} \mathsf{G}(U_\lambda)$ or $C \xleftarrow{*} U_{2\lambda}$, this will simulate either the experiment H_i or H_{i+1} (when $i < N$) and H_{i+1} or H_i (when $i \geq N$), respectively. Thus, we can break the security of G with the same advantage we have in distinguishing H_i from H_{i+1}. As there are $2N$ hybrids, we loose a factor $2N-1$; the reason we stated a loss of $2N$ in the proposition is explained in Remark 4.

B.4 Proof of Lemma 2

By eq.(5), $\mathcal{S}_I$ and $\mathcal{S}_{I+\langle 1 \rangle}$ differ by exactly one element $d \in \{1, \dots, N-1\}$. Assume that $d \in \mathcal{S}_{I+\langle 1 \rangle}$ (the case where $d \in \mathcal{S}_I$ is symmetric).

Given a PRG challenge $C \in \{0,1\}^{2\lambda}$, we simulate the game $H^{\mathcal{P}}_{I+\langle 1 \rangle}$, but at the point where in $\mathcal{O}(.)$, the if clause "if $\ell - 1 \in \mathcal{P}$ and $c_{\ell-1} = q_{\ell-1}$" evaluates to true for $\ell = d$, we set $K_{x[1\dots\ell-1]\|0} \| K_{x[1\dots\ell-1]\|1} := C$ (instead of assigning it a random value). If the challenge was sampled as $C \xleftarrow{*} U_{2\lambda}$ then we still simulated the game $H^{\mathcal{P}}_{I+\langle 1 \rangle}$. But if $C \xleftarrow{*} \mathsf{G}(U_\lambda)$, we simulated $H^{\mathcal{P}}_I$. Thus, from any distinguisher for $H^{\mathcal{P}}_I$ and $H^{\mathcal{P}}_{I+\langle 1 \rangle}$, we get a distinguisher for the PRG G with the same advantage.

B.5 Proof of Lemma 3

Let $\mathcal{P}' = \mathcal{P} \cup \{a\}$, where a is the element additionally contained in the $(\ell+1)$-level sets $\mathcal{S}_{I+\langle 10_{n-\ell-1}\rangle}$, $\mathcal{S}_{I+\langle 20_{n-\ell-1}\rangle}$ between $\mathcal{S}_I$ and $\mathcal{S}_{I'}$ (i.e., the element $d - N/2^{\ell+1}$ in eq.(4)). Adding an element to the guessing set $\mathcal{P}$, simply decreases the probability of outputting 1 by a factor of q, thus with eq.(13):

$$|\Pr[H_I^{\mathcal{P}'} = 1] - \Pr[H_{I'}^{\mathcal{P}'} = 1]| = |\Pr[H_I^{\mathcal{P}} = 1] \cdot \tfrac{1}{q} - \Pr[H_{I'}^{\mathcal{P}} = 1] \cdot \tfrac{1}{q}| = \delta/q \tag{18}$$

Now we can consider the sequence of consecutive $(\ell+1)$-level sets $H_I^{\mathcal{P}'}, H_{I+\langle 10_{n-\ell-1}\rangle}^{\mathcal{P}'}, H_{I+\langle 20_{n-\ell-1}\rangle}^{\mathcal{P}'}, H_{I'}^{\mathcal{P}'}$. By eq.(18) and a standard hybrid argument, two of these can be distinguished with advantage $\delta/(3q)$ as required.

C Proof of Theorem 2

Our proof follows the one from [LW14], with the main difference that we check consistency of the oracle replies w.r.t. the answers of the fingerprint queries, whereas Lewko and Waters check consistency w.r.t. the public parameters.

Without loss of generality, we assume that the adversary makes exactly $q = N - 1$ queries. We first construct an attacker A that $(t, \epsilon, N-1)$-breaks unpredictability of the prefix-constrained PRF for any given ϵ in some time t not necessarily polynomial in λ. We then show how this attacker can be simulated in polynomial time.

An inefficient attacker. We start with constructing a hypothetical attacker A, which wins the unpredictability game with probability ϵ for any given ϵ. A first makes two queries for constrained keys which will serve as a "fingerprint" of the secret key used by the challenger. A then picks a random value x which cannot be evaluated with the obtained keys. Next, it queries keys with which it can evaluate the PRF at all points in the domain except x and its sibling (i.e., $(x_1, \ldots, x_{N-1}, \overline{x}_N)$, where we let $\overline{x}_i := 1 - x_i$). It checks whether the received keys are consistent with the fingerprints and if so, it computes the PRF value at x under the secret key defined by the fingerprint (a step which may not be efficient); otherwise it aborts.

Phase 1 (Fingerprinting): A starts by making two constrained-key queries for $v_1 := (0, ?, \ldots, ?)$ and $v_2 := (1, 0, ?, \ldots, ?)$. Upon receiving the respective keys

$$(K_1, \{D_{i,b}\}_{i\in[2,N],b\in\{0,1\}}) \in \mathbb{G}_2 \times \mathbb{G}^{2(N-1)} \quad \text{and} \quad (K_2, \{D'_{i,b}\}_{i\in[3,N],b\in\{0,1\}}) \in \mathbb{G}_3 \times \mathbb{G}^{2(N-2)} \ ,$$

A aborts if $D_{i,b} \neq D'_{i,b}$ for any $i \in [3, N], b \in \{0, 1\}$. A also aborts if $D_{2,0} = 1_{\mathbb{G}}$ (as in this case, K_2 does not uniquely fix the value $\alpha \cdot d_{1,1}$ of the challenger's secret key; see below). Otherwise, A stores the values $(K_1, K_2, \{D_{i,b}\}_{i,b})$.

Phase 2 (All-but-x): Next, A picks a random value $x' \in \{0,1\}^{N-2}$, defines $x = 11\|x'$ and makes the following queries:

$$v_i = (1, 1, x_3, \ldots, x_{i-1}, \overline{x}_i, ?, \ldots, ?) \qquad \text{for } 3 \leq i \leq N-1 \tag{19}$$

Note that the keys for v_1, v_2 and $\{v_i\}_{i\in[3,N-1]}$ let A evaluate the PRF at any point different from x and its sibling.

Let the i-th answer be $k_i = (K_i, \{D_{j,b}^{(i)}\}_{j\in[i+1,N],b\in\{0,1\}}) \in \mathbb{G}_{i+1} \times \mathbb{G}^{2(N-i)}$. A makes the following checks; if any of them fail, A aborts:

Check 1: For all $i \in [3, N-1]$, $j \in [i+1, N], b \in \{0,1\}$: $D^{(i)}_{j,b} \stackrel{?}{=} D_{j,b}$ (with $D_{j,b}$ obtained in Phase 1).

Check 2: For all $i \in [3, N-1]$: $e(K_i, D_{2,0}) \stackrel{?}{=} e(K_2, D_{2,1}, \ldots, D_{i-1,x_{i-1}}, D_{i,\overline{x}_i})$.

These checks ensure that the keys are consistent with the fingerprint keys, in particular:

Lemma 5. *For all $i \in [2, N-1], b \in \{0,1\}$, let $d_{i,b} \in \mathbb{Z}_p$ be such that $D_{i,b} = g^{d_{i,b}}$ and let $d'_{1,0}$ and $d'_{1,1}$ be such that $K_1 = g_2^{d'_{1,0}}$ and $K_2 = g_3^{d'_{1,1} \cdot d_{2,0}}$. Since $d_{2,0} \neq 0$ (otherwise* A *aborted in Phase 1), these values are uniquely defined. Moreover, for all $i \in [3, N-1]$ for which Check 2 holds, we have*

$$K_i = g_{i+1}^{d'_{1,1} \cdot d_{2,1} \cdot d_{3,x_3} \cdots d_{i-1,x_{i-1}} \cdot d_{i,\overline{x}_i}} \quad .$$

Proof. Let γ_i be such that $K_i = g_{i+1}^{\gamma_i}$ for $i \in [3, N-1]$. The check ensures that

$$g_{i+2}^{d'_{1,1} \cdot d_{2,0} \cdot d_{2,1} \cdots d_{i-1,x_{i-1}} \cdot d_{i,\overline{x}_i}} = e(K_2, D_{2,1}, \ldots, D_{i-1,x_{i-1}}, D_{i,\overline{x}_i}) = e(K_i, D_{2,0}) = g_{i+2}^{\gamma_i \cdot d_{2,0}} \quad ,$$

which, since $d_{2,0} \neq 0$, yields $\gamma = d'_{1,1} \cdot d_{2,1} \cdots d_{i-1,x_{i-1}}, d_{i,\overline{x}_i}$ and proves the claim. □

Phase 3 (Solve challenge): If all received keys passed the checks then A uses the values $d'_{1,1}, d_{2,1}, d_{3,x_3}$, $\ldots, d_{N,x_N}$, defined by the received keys as in Lemma 5 to compute the following (which we show is the PRF value at x):

$$y = g_{N+1}^{d'_{1,1} \cdot d_{2,1} \cdot \prod_{i=3,N} d_{i,x_i}}$$

(this step may not be efficient). A then flips a biased coin which yields $\beta = 1$ with probability $\xi := \epsilon \cdot (1 - \frac{1}{p})^{-1}$. If $\beta = 1$ then A outputs y; otherwise it aborts.

We show that A $(t, \epsilon, N-1)$-breaks unpredictability: A makes $N-1$ key queries. The value y is the PRF value of x: To see this, let $k^* = (\alpha^*, \{d^*_{i,b}\}_{i \in [N], b \in \{0,1\}})$ be the secret key chosen by A's challenger. If k^* is used to answer the fingerprint queries then with $d'_{1,0}, d'_{1,1}, d_{2,0}, d_{2,1}, \ldots$ defined as in Lemma 5 we have

- For all $i \in [2, N], b \in \{0,1\} : d_{i,b} = d^*_{i,b}$;
- and $d'_{1,0} = \alpha^* \cdot d^*_{1,0}$, $d'_{1,1} = \alpha^* \cdot d^*_{1,1}$.

This implies that $y = g_{N+1}^{\alpha^* d^*_{1,1} \cdot d^*_{2,1} \cdot \prod_{i=3,N} d^*_{i,x_i}} = F(k^*, x)$. Note also that constructing the function value this way (i.e., using the values defined by the first two replies) for any $11 \| z \neq x$, except for $z = (x_1, \ldots x_{N-1}, \overline{x}_N)$, leads to the same value as evaluating under the key k_j with $j = \min\{i | z_i \neq x_i\}$.

Finally, (with c, ξ as above)

$$\begin{aligned}
\Pr[F(k^*, x) \leftarrow \mathsf{A}] &= \Pr[F(k^*, x) \leftarrow \mathsf{A} \,|\, c = 1 \wedge d_{2,0} \neq 0] \xi (1 - \tfrac{1}{p}) \\
&\qquad + \Pr[F(k^*, x) \leftarrow \mathsf{A} \,|\, c = 0 \vee d_{2,0} = 0](1 - \xi(1 - \tfrac{1}{p})) \\
&= 1 \cdot \xi(1 - \tfrac{1}{p}) + 0 \cdot (1 - \xi(1 - \tfrac{1}{p})) = \epsilon
\end{aligned}$$

Breaking the assumption using $\mathcal{R}$. Since the only random choices A makes are choosing x in Phase 2 and flipping the biased coin β at the end, we can assume that A draws its coins from a set $Z \times F$, where $Z = \{0,1\}^{N-2}$ is the set of possible strings x', and F are the coins used to choose the value β.

As we consider simple reductions, $\mathcal{R}$ runs A once in a straight-line fashion. We use $\mathcal{R}$ to create an algorithm B which solves Π. B does so by running the reduction $\mathcal{R}$ on a challenge and simulating the adversary A constructed above, but running itself in polynomial time. B starts by passing the received challenge

instance $c \in C$ to $\mathcal{R}$ and simulates Phase 1 of attacker A and stores the received values $(K_1, K_2, \{D_{i,b}\})$ (if A did not abort).

Next, B runs A's interaction with the reduction in the second phase τ times. (We will fix τ later so that it is polynomial in λ.) In each run B chooses fresh random coins for $\mathcal{R}$ and A. Thus, in the i-th interaction, B picks an independent random value $x^{(i)}$, makes the queries $v_j^{(i)}$ defined by $x^{(i)}$, as in (19), and performs the consistency checks. If all checks pass, B stores the received values $(K_3^{(i)}, \ldots, K_{N-1}^{(i)})$; otherwise Run i is labeled an "aborting run". If all τ runs were aborting runs then B terminates and outputs a random guess.

If there was at least one non-aborting run, B chooses a random $z' \in \{0,1\}^{N-2}$, defines $z = 11\|z'$ and if for any $x^{(i)}$, we have $(z_1, \ldots, z_{N-1}) = (x_1, \ldots, x_{N-1})$ then B stops and outputs a random guess. Next, B makes key queries for the values $v_3, \ldots v_{N-1}$ derived from z, as in (19). It checks consistency of the received keys and outputs a random guess if a check fails.

B picks a run i which was not aborting and lets $j \in [3, N-1]$ be the lowest index such that $x_j^{(i)} \neq z_j$. (This must exist, as otherwise, B would have aborted.) Since $v_j^{(i)}$ is a prefix of z, B can use the key received when querying $v_j^{(i)}$ to compute the PRF value y at z.

As we have argued above, the value computed this way is perfectly consistent with the information about the secret key fixed by the replies in Phase 1, meaning that B computes the same value as A would. B flips a biased coin β and with probability $\xi := \epsilon \cdot (1 - \frac{1}{p})^{-1}$ outputs y.

Analyzing B's success probability. Recall that C is the set of possible challenges for Π and that A's coins are drawn from $Z \times F$. Let $R = R_1 \times R_2$ be the set of possible random coins chosen by $\mathcal{R}$, where R_1 are the coins used up to the answering of A's fingerprint queries. Thus (c, r_1) determines the values $(K_1, K_2, \{D_{i,b}\})$, and (r_2, z, f) are the coins that are freshly chosen from $R_2 \times Z \times F$ in every rewind run.

We define W as the set of all tuples (c, r_1, r_2, z, f) such that when $\mathcal{R}$ is run with (r_1, r_2) and A is run with (z, f) on the challenge c then A does not abort and $\mathcal{R}$ solves the challenge c. We partition W into two sets according to a probability threshold ρ (which we will fix later) of a run not aborting when fixing c and r_1 and choosing the other coins freshly. Let T be the set of all (c, r_1, r_2, z, f) that lead to a run aborted by A; define

$$U := \left\{(c, r_1, r_2, z, f) \in W \;\middle|\; \Pr_{r_2', z', f'}\left[(c, r_1, r_2', z', f') \notin T\right] \geq \rho\right\} \quad \text{and} \quad V := W \setminus U \ .$$

We first show the following lemma:

Lemma 6. $\Pr[V] \leq \rho$

Proof. First note that $W \subseteq \overline{T}$ (since for coins to be in W, A must not abort). This implies that

$$\begin{aligned} V &= \left\{(c, r_1, r_2, z, f) \in W \;\middle|\; \Pr_{r_2', z', f'}\left[(c, r_1, r_2', z', f') \notin T\right] < \rho\right\} \\ &\subseteq \left\{(c, r_1, r_2, z, f) \in W \;\middle|\; \Pr_{r_2', z', f'}\left[(c, r_1, r_2', z', f') \in W\right] < \rho\right\} \ . \end{aligned}$$

The lemma now follows because the probability of the latter set is strictly lower than ρ, since we have the following:

For any sets X, Y and $W \subseteq X \times Y$, the set $Z := \{(x, y) \in W \mid \Pr_{y' \leftarrow Y}[(x, y') \in W] < \rho\}$ has $\Pr[Z] < \rho$: Let $X_1 = \{x \in X \mid \Pr_{y' \leftarrow Y}[(x, y') \in W] < \rho\}$ and $X_2 = X_1 \setminus X$. Then

$$\Pr[(x, y) \in Z] = \sum_{x \in X_1} \Pr_{x' \leftarrow X}[x' = x] \underbrace{\Pr_{y' \leftarrow Y}[(x, y') \in Z]}_{< \rho \ \text{(since } x \in X_1\text{)}} + \sum_{x \in X_2} \Pr_{x' \leftarrow X}[x' = x] \underbrace{\Pr_{y' \leftarrow Y}[(x, y') \in Z]}_{= 0 \ \text{(since } x \in X_2\text{)}}$$

□

Next we define S as the set of all (c, r_1, r_2, z, f) so that $\mathcal{R}$ solves the challenge.

Lemma 7. *If Π is computationally hard then* $\Pr[T] \cdot \left| \Pr[S|T] - \frac{1}{2} \right|$ *is negligible in λ.*

Proof. Consider an adversary B' which runs $\mathcal{R}$ and simulates A up to Phase 3. If A has not aborted until then, B' outputs a random guess. If A aborted, it outputs whatever $\mathcal{R}$ outputs. B' runs in polynomial time and solves the challenge with probability $\frac{1}{2}(1 - \Pr[T]) + \Pr[S|T] \Pr[T] = \frac{1}{2} + \Pr[T](\Pr[S|T] - \frac{1}{2})$.

Consider B'', which behaves like B' except that it outputs the complement, i.e., in case A aborts, B'' outputs 0 if $\mathcal{R}$ outputs 1 and vice versa. B success probability is $\frac{1}{2}(1 - \Pr[T]) + (1 - \Pr[S|T]) \Pr[T] = \frac{1}{2} + \Pr[T](\frac{1}{2} - \Pr[S|T])$. Together this yields that $\Pr[T] \cdot |\Pr[S|T] - \frac{1}{2}|$ must be negligible. □

The probability that $\mathcal{R}$ solves the challenge when running A once is the probability of solving it when A does not abort plus the probability of solving it when A aborts:

$$\Pr[W] + \Pr[T] \Pr[S|T] = \tfrac{1}{2} + \delta \ .$$

We have $\Pr[W] = \Pr[U] + \Pr[V] < \Pr[U] + \rho$ (by Lemma 6), which together with Lemma 7 yields

$$\tfrac{1}{2} + \delta < \Pr[U] + \rho + \tfrac{1}{2} \Pr[T] + \mathrm{negl}(\lambda) \ . \tag{20}$$

Let $X^{(i)} \times F^{(i)}$ denote the set of coins for A and $R_2^{(i)}$ denote the set of coins used by $\mathcal{R}$ during the i-th run of $\mathcal{R}$ after answering the first two queries. Define T_i to be the set of those coins $(c, r_1, r_2^{(i)}, x^{(i)}, f^{(i)})$ that lead to an aborting run. Let E_i be the event that $z_j = x_j^{(i)}$ for all $j \in [3, N-1]$.

Consider a set of coins $(c, r_1, \{r_2^{(i)}, x^{(i)}, f^{(i)}\}_{i=1}^{\tau}, r_2, z, f)$ used by B during the overall computation, including the rewinds. Note that B aborts the computation if and only if

$$\forall i \in [1, \tau] : (c, r_1, r_2^{(i)}, x^{(i)}, f^{(i)}) \in T_i \quad \text{or} \quad \exists i \in [1, \tau] \, \forall j \in [3, N-1] : x_j^{(i)} = z_j \ ,$$

which corresponds to the coins being in the set $\bigcap_{i=1}^{\tau} T_i \cup \bigcup_{i=1}^{\tau} E_i$. On the other hand, if $(c, r_1, r_2, z, f) \in U \subseteq W$ and if B does not abort then it solves the challenge. Thus B wins with probability at least

$$\frac{1}{2} \Pr[T] + \sum_{(c, r_1, r_2, z, f) \in U} \Pr[(c, r_1, r_2, z, f)] \cdot \Big(1 - \Pr\Big[\bigcap_{i=1}^{\tau} T_i \cup \bigcup_{i=1}^{\tau} E_i \,\Big|\, (c, r_1, r_2, z, f) \Big] \Big) \ . \tag{21}$$

By the union bound, we have

$$\Pr\big[\textstyle\bigcup_{i=1}^{\tau} E_i \,\big|\, (c, r_1, r_2, z, f) \big] \leq \tau 2^{-(N-3)} \ . \tag{22}$$

Since for fixed (c, r_1), the events T_i are independent, we have

$$\Pr\big[\textstyle\bigcap_{i=1}^{\tau} T_i \,\big|\, (c, r_1, r_2, z, f) \big] = \prod_{i=1}^{\tau} \Pr[T_i \,|\, (c, r_1, r_2, z, f)] \leq (1 - \rho)^{\tau} \ , \tag{23}$$

where the last inequality follows from (c, r_1, r_2, z, f) being in U. By the union bound and from equations (22) and (23), we have that (21) is greater than

$$\begin{aligned} \tfrac{1}{2} \Pr[T] + \Pr[U]\big(1 - \tau 2^{-(N-3)} - (1 - \rho)^{\tau}\big) &\geq \tfrac{1}{2} \Pr[T] + \Pr[U] - 8\tau 2^{-N} - (1 - \rho)^{\tau} \\ &\geq \tfrac{1}{2} + \delta - \rho - 8\tau 2^{-N} - (1 - \rho)^{\tau} - \mathrm{negl}(\lambda) \ , \end{aligned}$$

where the last inequality follows from (20).

Setting $\rho = \frac{\delta}{4}$, and $\tau = \frac{\lambda}{\delta}$ (which is polynomial in λ), the last term equals

$$\frac{1}{2} + \frac{3}{4}\delta - 8\frac{\lambda}{\delta}2^{-N} - [(1 - \frac{\delta}{4})^{\frac{1}{\delta}}]^{\lambda} - \mathrm{negl}(\lambda) \;=\; \frac{1}{2} + \frac{3}{4}\delta - 8\frac{\lambda}{\delta}2^{-N} - \mathrm{negl}(\lambda) \;, \tag{24}$$

since $(1 - \frac{\delta}{4})^{\frac{1}{\delta}} < 1$ for all $\delta \in [0, 1]$. We have showed that B's probability of solving Π is at least (24), which by the assumption that Π is computationally hard means that $\frac{3}{4}\delta - 8\frac{\lambda}{\delta}2^{-N}$ must be negligible in λ, and therefore δ must be exponentially small as a function of N.