Obfuscation of Hyperplane Membership

Ran Canetti[1,⋆], Guy N. Rothblum[2,⋆⋆], and Mayank Varia[3,⋆⋆⋆]

[1] School of Computer Science, Tel Aviv University (canetti@cs.tau.ac.il)
[2] Princeton University (rothblum@princeton.edu)
[3] Massachusetts Institute of Technology (varia@csail.mit.edu)

Abstract. Previous work on program obfuscation gives strong negative results for general-purpose obfuscators, and positive results for obfuscating simple functions such as equality testing (point functions). In this work, we construct an obfuscator for a more complex algebraic functionality: testing for membership in a hyperplane (of constant dimension). We prove the security of the obfuscator under a new strong variant of the Decisional Diffie-Hellman assumption. Finally, we show a cryptographic application of the new obfuscator to digital signatures.

1 Introduction

The problem of program obfuscation has been of long-standing interest to practitioners, and has recently been an active topic of research in theoretical cryptography. The high-level goal of program obfuscation is to compile a computer program in such a way that an adversary cannot learn anything from seeing the program beyond could be learned by running the program and observing its input-output behavior.

Barak *et al.* [1] formalized the notion of obfuscation using simulation-based definitions. Over the past decade, the theory community has found a few positive obfuscation results for specific families of programs. In this paper, we provide an obfuscator for a new family of programs.

Virtual black-box obfuscation. The procedure of “obfuscating” a computer program should garble the program’s code and make it unintelligible. The extent of the garbling is limited by the fact that the program’s functionality should be preserved. As a result, both honest and adversarial users of the obfuscated program can learn some information by observing the program’s input-output functionality, and we do not wish to prevent users from learning information this way. Instead, obfuscation ensures that this is the only way that an adversary can learn information from the obfuscated program.

[⋆] Supported by the Check Point Institute for Information Security, an ISF grant, an EU Marie Curie grant, and an Israel-US BSF grant.

[⋆⋆] Supported by NSF Grants CCF-0635297, CCF-0832797 and by a Computing Innovation Fellowship.

[⋆⋆⋆] Supported by the Department of Defense through the NDSEG Program.

There are several ways to formalize this intuitive notion [2,3,4]. This paper uses the virtual black-box formalization of [1]. A significant obstacle to obtaining positive results with respect to this definition is that the security notion must hold for *all* programs in a given family, and not just a random instance. This is one reason why standard cryptographic tools and analytical techniques (which usually deal with randomly chosen instances) are not always helpful for obfuscation.

Previous results. Several works have disproved the existence of a "general-purpose obfuscator" that can simultaneously obfuscate every program [1,2,5]. In fact, these papers demonstrate specific programs that cannot be obfuscated, and these programs come from a relatively low complexity class. While these negative results are disheartening, they focus on specific (often contrived) functionalities. Obfuscation remains possible for many programs of interest.

Still, very few positive results are known even for specific, simple programs (or boolean circuits). One family of programs for which positive results are known is the family of "point circuits": password-checking programs that accept a single input string and reject all other inputs. This family can be obfuscated under a variety of cryptographic assumptions [6,7,8,9]. Some of these constructions can be generalized in two ways. First, we can obfuscate "multi-point circuits," which accept a polynomially-sized list of input strings, and second, we can obfuscate "point circuits with multi-bit output," which store a hidden output string that is revealed only for a single input value [10]. Other formalizations of program obfuscation [3,4] allow for the obfuscation of cryptographic tasks such as checking proximity to a hidden point [11], vote mixing [12], and re-encryption [4]. The latter two applications use a different security guarantee that only holds over a random choice of the circuit from the family.

Our result. In this paper, we obfuscate programs that perform hyperplane membership testing. Let P be a hyperplane in a vector space, and let H_P be a program that tests whether its input is a point on the hyperplane. An obfuscation of H_P allows a user to determine whether her input point is on the hyperplane, but reveals no additional information such as the distance from her point to the hyperplane or any other points that are on the hyperplane.

More precisely, given a prime p and positive integer d, consider the family of hyperplanes through the origin in the vector space $(\frac{\mathbb{Z}}{p\mathbb{Z}})^d$ over the finite field $\frac{\mathbb{Z}}{p\mathbb{Z}}$. In this setting, a hyperplane can be defined by a vector that is orthogonal to every point in the plane. Let $\boldsymbol{a}$ be a vector in $(\frac{\mathbb{Z}}{p\mathbb{Z}})^d$ and consider the program

$$H_{\boldsymbol{a}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \langle \boldsymbol{a}, \boldsymbol{x} \rangle = 0 \\ 0 & \text{otherwise.} \end{cases}$$

We construct an obfuscator for this family of programs.

This primitive subsumes many of the previously-known results. In the $d = 2$ case, these "hyperplanes" turn out to be equivalent to point circuits, and our specific construction and assumption reduce to those in [6]. Furthermore, the technique of [10] can be applied to our primitive as well, so we can obfuscate circuits that output a hidden message when its input is on the hyperplane.

We also note that hyperplane membership testing has been considered in the context of private predicate encryption schemes by Shen, Shi, and Waters [13], although our results are incomparable to theirs.

Application to digital signatures. As an example of the proposed primitive's usefulness, we demonstrate an application of our obfuscator to leakage-resilient one-time signatures. We emphasize, though, that the main motivation for this work is the new obfuscator, rather than any single application.

The signature scheme is constructed as follows: the secret key is a randomly chosen plane in 3-dimensional space, and the public key is the obfuscated membership program. To sign a message m, find a point on the plane that is related to m. This signature can be verified by running the public obfuscated program.

This signature scheme satisfies a weaker form of the unforgeability game, where the adversary is required to submit a message m to be signed before receiving the public key. Techniques from [14] allows us to transform the weak scheme into an ordinary one-time signature scheme. Additionally, this one-time signature scheme remains unforgeable even when a function of the secret key is leaked whose output length is up to half as long as the secret key. For schemes that do not use general zero-knowledge proofs, this matches the leakage bound of [15] (albeit under much stronger assumptions).

Construction. The construction is as follows. Let G be a group of order p that satisfies a strengthened version of the Decisional Diffie-Hellman assumption, which we describe in more detail below. When the obfuscator is given a hyperplane $H_{\boldsymbol{a}}$ to obfuscate, where $\boldsymbol{a} = (a_1, \ldots, a_d)$, it chooses a random generator $g \overset{U}{\leftarrow} G$ and outputs g^{a_i} for all i. This allows the user to compute whether a given point $\boldsymbol{x} = (x_1, \ldots, x_d)$ is on the hyperplane by computing

$$(g^{a_1})^{x_1} \times \cdots \times (g^{a_d})^{x_d} = g^{\langle \boldsymbol{a}, \boldsymbol{x} \rangle} ,$$

and checking whether this equals G's identity element (i.e. whether $\langle \boldsymbol{a}, \boldsymbol{x} \rangle = 0$).

Our assumption. We are not able to prove the security of our construction based on the standard Decisional Diffie-Hellman assumption, which states that g^{ab} is indistinguishable from uniform, given g, g^a, and g^b for uniformly-chosen exponents a and b. We describe the difficulty with basing our scheme on DDH, as it motivates and clarifies our new assumption.

For our construction, it is crucial that the adversary not be able to compute any polynomial relationships in the exponent (not just quadratic ones). Consider, for instance, whether it is possible to compute g^{a^3} given just g and g^a. What if we wish to compute g^{abc} from g, g^a, g^b, and g^c? Can elements of the form g^{a^3}

or g^{abc} even be distinguished from uniform? No efficient algorithms for running these computations are known (e.g. in groups where DDH is hard), but standard assumptions such as DDH do not seem to rule out the existence of such algorithms. In general, we wish to understand when $g^{p(a,b)}$ is distinguishable from uniform, given a polynomial p and group elements g, g^a, and g^b. Clearly, this is true when p is linear, or closely resembles a line. Our new assumption states that these are the only such polynomials for which $g^{p(a,b)}$ can be distinguished from uniform.

We also consider the effect of choosing exponents from weak entropy distributions. This setting has been previously considered by Canetti [6], who forms a modified DDH assumption in which g^{ab} is considered to be indistinguishable from uniform, even given g, g^a, and g^b, where a is chosen from the uniform distribution but b is chosen from any distribution of super-logarithmic min-entropy. Our assumption expands upon this idea and considers many exponents that are not only chosen from weak entropy distributions, but which may also be related.

Specifically, given a tuple of group elements $\langle g^{a_1}, g^{a_2}, \ldots, g^{a_d} \rangle$ where the a_i's are chosen from some joint distribution, we ask for which polynomials p is $g^{p(a_1,\ldots,a_d)}$ still indistinguishable from uniform? If the polynomial p looks linear when restricted to the support of the joint distribution, then of course $g^{p(a_1,\ldots,a_d)}$ can be distinguished from uniform. Our new assumption states that indistinguishability holds in all other cases.

This new assumption is stronger than the standard DDH assumption, or even the modified DDH assumption of [6], but we provide evidence of its feasibility by proving that it holds in the generic group model. Furthermore, we believe that resolving the status of this new assumption would be interesting either way. If it holds, then we obtain an obfuscator for a new and interesting family of functions. Showing that the assumption does not hold would shed new light on which computations can be run efficiently in the exponent of DDH groups.

Organization. Section 2 defines virtual black-box obfuscation and the hyperplane membership testing programs. Section 3 describes our assumption in detail and compares it to previous assumptions. Section 4 presents our obfuscator for the family of hyperplanes and proves its security. Section 5 extends our construction to the multi-bit setting. Section 6 presents our one-time signature scheme. Some of the proofs are relegated to the full version of this paper [16].

2 Definitions

2.1 Virtual black-box obfuscation

In [1], [5], and other works, an obfuscator is defined as a compiler that takes a circuit as input and returns another circuit. The output circuit should be a “garbled” version of the input circuit, in the sense that the circuits should have the same functionality, but it should be difficult for an adversary to learn information from the output circuit.

Consider an imaginary world in which people can give others access to oracles at will. In this imaginary world, we can easily perform perfect obfuscation by giving users oracle access to a computer program. The oracle allows them to learn the program's input-output functionality, but any other aspect of the program's behavior is hidden from the users. Unfortunately, in the real world we cannot hand out oracles to other people. Instead, we want obfuscators to be able to replicate the power of oracles in the imaginary world. The formalization of obfuscation provided by Barak *et al.* [1], called the *virtual black-box* property, achieves this goal.

The virtual black-box property considers two different worlds. In the real world, an efficient adversary has access to the obfuscated program code and attempts to learn a one bit predicate about the underlying program. The definition ensures that there exists a simulator in imaginary world that only interacts with an oracle to the program but can still can learn the same predicate that the adversary learns in the real world. Hence, the virtual black-box property ensures that access to the code of an obfuscated program is *no more* useful than access to the oracle.

We only require that the obfuscator operate over a specified family of circuits. Throughout this work, all circuits are assumed to be non-uniform.

Definition 1 (Obfuscation). *Let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a family of polynomial-size circuits, where $\mathcal{C}_n$ denotes all circuits of input length n. A probabilistic polynomial time (PPT) algorithm $\mathcal{O}$ is an* obfuscator *for the family $\mathcal{C}$ if the following three conditions are met.*

1. Approximate functionality: *There exists a negligible function ε such that for every n, every circuit $C \in \mathcal{C}_n$ and every x in the input space of C,*

$$\Pr[\mathcal{O}(C)(x) = C(x)] > 1 - \varepsilon(n) ,$$

where the probability is over the randomness of $\mathcal{O}$. If this probability always equals 1, then we say that $\mathcal{O}$ has exact functionality.

2. Polynomial slowdown: *There exists a polynomial q such that for every n, every circuit $C \in \mathcal{C}_n$, and every possible sequence of coin tosses for $\mathcal{O}$, the circuit $\mathcal{O}(C)$ runs in time at most $q(|C|)$.*
3. Virtual black-box: *For every PPT adversary A and polynomial δ, there exists a PPT simulator S such that for all sufficiently large n, and for all $C \in \mathcal{C}_n$,*

$$|\Pr[A(\mathcal{O}(C)) = 1] - \Pr[S^C(1^n) = 1]| < \frac{1}{\delta(n)} ,$$

where the first probability is taken over the coin tosses of A and $\mathcal{O}$, and the second probability is taken over the coin tosses of S.

2.2 Vector spaces

In this section, we define the vector spaces over which our constructions operate. Let $d \in \mathbb{N}$, p be a prime number, and $\mathbb{F}_p = \frac{\mathbb{Z}}{p\mathbb{Z}}$. Then, $\mathbb{F}_p$ is a field and $\mathbb{F}_p^d$ is a vector space over $\mathbb{F}_p$. We denote a vector in the vector space by $\boldsymbol{x} = (x_1, \ldots, x_d)$, and we have an inner product-style operation given by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{d} x_i y_i$.

Definition 2. *Let $S \subseteq \mathbb{F}_p^d$ be a set.*

1. *Two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_p^d$ are said to be* orthogonal *if their inner product is zero, so $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$. Note that the set of all vectors orthogonal to $\boldsymbol{x}$ forms a $(d-1)$ dimensional hyperplane.*
2. *The* closure *of S, written $\bar{S}$, is the subspace of all linear combinations of vectors in S.*
3. *The* orthogonal complement *of S, written $S^{\perp}$, is the subspace of all vectors that are orthogonal to every vector in S. That is,*

$$S^{\perp} = \{\boldsymbol{x} \in \mathbb{F}_p^d : \langle \boldsymbol{x}, \boldsymbol{s} \rangle = 0 \; \forall \boldsymbol{s} \in S\} \; .$$

We caution that $\mathbb{F}_p^d$ does not satisfy all of the axioms of an inner product space. Nevertheless, the following theorem about inner product spaces, which we need in the proof of our main theorem, does hold over $\mathbb{F}_p^d$.

Theorem 3. *Let $S \subseteq \mathbb{F}_p^d$ be a set. Then, $(S^{\perp})^{\perp} = \bar{S}$.*

Proof (sketch). First, $S^{\perp}$ and $S^{\perp\perp}$ are subspaces of $\mathbb{F}_p^d$ because the conditions imposed on them are linear. Second, $\bar{S} \subseteq S^{\perp\perp}$ because the vectors in $\bar{S}$ are orthogonal to those in $S^{\perp}$, so they are in $S^{\perp\perp}$. Third, $\dim(\bar{S}) = \dim(S^{\perp\perp})$ because both of them are equal to $d - \dim(S^{\perp})$.

Therefore, $\bar{S}$ and $S^{\perp\perp}$ are subspaces of $\mathbb{F}_p^d$ of the same dimension such that one is included in the other, so they are equal. □

We also note that the vector space $\mathbb{F}_p^d$ is a bit redundant for our needs. We wish to identify a hyperplane P with a vector $\boldsymbol{x}$ that is orthogonal to every vector in the hyperplane. However, the vector $\boldsymbol{x}$ is not unique: indeed, for any $c \in \mathbb{F}_p \setminus 0$, the vector $c\boldsymbol{x}$ is also orthogonal to every vector in P, so the normal vector to the hyperplane is only unique up to scalar multiplication. As a result, we note that there are only $d-1$ degrees of freedom when choosing a normal vector, which is why the $d = 2$ case corresponds to point functions. One canonical representation of the normal vector, which we will use when convenient throughout the paper, is to consider all of the vectors in $\mathbb{F}_p^d$ whose first non-zero coordinate equals 1.[4]

3 Our assumption

In this section, we define the main assumption. Then, we relate our assumption to a DDH variant found in [6] and consider the assumption in the generic group model.

Our assumption uses groups of increasing prime order. We use the following definition to encapsulate the order requirement.

[4] In fact, the appropriate ambient space from which to consider the normal vectors is the projective space $\mathbb{P}^{d-1}(\mathbb{F}_p)$, and we are using its embedding in $\mathbb{F}_p^d$ as a concrete instantiation of the projective space.

Definition 4. *A function $\rho(n)$ is called a* prime sequence *if for every $n \in \mathbb{N}$, $\rho(n)$ is a prime number in the range $(2^{n-1}, 2^n]$.*

Our assumption is parametrized by $d \in \mathbb{N}$. We abuse notation a bit and denote $\mathbb{F}_\rho^d = \{\mathbb{F}_{\rho(n)}^d\}_{n\in\mathbb{N}}$.

Assumption 5. *Given $d \in \mathbb{N}$, there exists a family of groups $\mathcal{G} = \{G_n\}_{n\in\mathbb{N}}$ (written multiplicatively) such that the following three conditions hold:*

1. *There are efficient algorithms to perform the group operation, to test for equality with the identity element, and to sample uniformly from $\mathcal{G}$.*
2. *The orders of the groups form a prime sequence $\rho(n) = |G_n|$.*
3. *For every PPT adversary A and for all families of distributions $\mathcal{L} = \{\mathcal{L}_n\}_{n\in\mathbb{N}}$ and $\mathcal{R} = \{\mathcal{R}_n\}_{n\in\mathbb{N}}$ over $\mathbb{F}_\rho^d$, there exists a polynomial q such that for all n,*

$$\begin{aligned}|\Pr[\boldsymbol{l} \leftarrow \mathcal{L}_n, g \overset{U}{\leftarrow} G_n : A(g^{l_1}, \ldots, g^{l_d}) = 1] \\ - \Pr[\boldsymbol{r} \leftarrow \mathcal{R}_n, g \overset{U}{\leftarrow} G_n : A(g^{r_1}, \ldots, g^{r_d}) = 1]| \\ \leq q(n) \cdot \max_{\boldsymbol{x}\in\mathbb{F}_{\rho(n)}^d} |\Pr[\boldsymbol{l} \leftarrow \mathcal{L}_n : \langle \boldsymbol{l}, \boldsymbol{x}\rangle = 0] - \Pr[\boldsymbol{r} \leftarrow \mathcal{R}_n : \langle \boldsymbol{r}, \boldsymbol{x}\rangle = 0]| \,. \quad (1)\end{aligned}$$

In words, this assumption states that an adversary can distinguish two distributions of vectors if and only if linear tests can do so as well.

3.1 Discussion

We make several remarks:

1. The right-hand side of (1) depends on ρ but not on any other property of $\mathcal{G}$.
2. Note that the adversary is allowed to distinguish $\mathcal{L}$ and $\mathcal{R}$ better than any single linear test does. For example, the adversary may try many linear tests. The assumption merely states that the left-hand side of (1) is negligible whenever the right-hand side is.
3. For a given adversary A, we denote $A_{\boldsymbol{l}} = \Pr[g \overset{U}{\leftarrow} G_n : A(g^{l_1}, \ldots, g^{l_d}) = 1]$ and $A_{\mathcal{L}} = \Pr[\boldsymbol{l} \leftarrow \mathcal{L}_n : A_{\boldsymbol{l}}]$ for simplicity. We will say that $\mathcal{L}$ and $\mathcal{R}$ are *indistinguishable by linear tests* if

$$\varepsilon(n) = \max_{\boldsymbol{x}\in\mathbb{F}_{\rho(n)}^d} |\Pr[\boldsymbol{l} \leftarrow \mathcal{L} : \langle \boldsymbol{l}, \boldsymbol{x}\rangle = 0] - \Pr[\boldsymbol{r} \leftarrow \mathcal{R} : \langle \boldsymbol{r}, \boldsymbol{x}\rangle = 0]|$$

   is a negligible function of n. Thus, the assumption states that for all $\mathcal{L}$ and $\mathcal{R}$ that are indistinguishable by linear tests, $|A_{\mathcal{L}} - A_{\mathcal{R}}|$ is negligible as well for all PPT adversaries A.
4. This assumption is computationally falsifiable, though perhaps inefficiently. There are two possible obstructions to efficiency. First, the descriptions of $\mathcal{L}$ and $\mathcal{R}$ may be inefficient, although this is not a problem for the distributions constructed in our proof. Second, it may not be efficient to determine which linear test performs the best. An interesting question is whether this computation can be performed efficiently, leading to an efficient falsification procedure.

3.2 On the assumption's hardness

In this section, we categorize the hardness of our assumption. To begin with, we present a DDH-based assumption due to Canetti [6].

Assumption 6. *Let n be a security parameter and let $p = 2q + 1$ be a randomly chosen n-bit safe prime. Consider the group Q of squares in $\mathbb{F}_p^*$. For any well-spread distribution ensemble $\{X_q\}$ where the domain of X_q is $\mathbb{F}_q$, for $g \overset{U}{\leftarrow} Q$, $a \leftarrow X_q$, $b, c \overset{U}{\leftarrow} \mathbb{F}_q$, the ensembles $\langle g, g^a, g^b, g^{ab} \rangle$ and $\langle g, g^a, g^b, g^c \rangle$ are computationally indistinguishable.*

In this assumption, a "well-spread ensemble" means that the min-entropy $H_\infty(X_q)$ is a super-logarithmic function of n.

The following theorem exemplifies the strength of our assumption by relating it to Assumption 6, which is already considered by the cryptographic community to be quite strong.

Theorem 7. *Assumption 6 implies Assumption 5 for dimension 2. For higher dimensions, our assumption may be stronger because Assumption 5 for dimension $d + 1$ implies Assumption 5 for dimension d.*

On the other hand, we provide evidence that our assumption is feasible by showing that it holds in the generic group model.

Theorem 8. *For all $d \in \mathbb{N}$, Assumption 5 for dimension d holds in the generic group model.*

Finally, we note that Assumption 5 for dimension 1 trivially holds, even by groups that do not satisfy DDH. A precise definition of the generic group model and a proof of Theorem 8 can be found in [16]. The rest of this section is devoted to a proof of Theorem 7, which we break into the following two lemmas.

Lemma 9. *Assumption 5 for dimension $d+1$ implies Assumption 5 for dimension d.*

Lemma 10. *Assumption 6 implies Assumption 5 for dimension 2.*

Proof (Lemma 9). Assume that Assumption 5 for dimension d is false, so for every prime sequence ρ and every set of groups $\mathcal{G} = \{G_n\}_{n \in \mathbb{N}}$, there exists a PPT adversary A and two distributions $\mathcal{L}$, $\mathcal{R}$ over vectors in $\mathbb{F}_\rho^d$ that are indistinguishable by linear tests but such that $|A_\mathcal{L} - A_\mathcal{R}|$ is noticeable.

Now construct distributions $\mathcal{L}'$ and $\mathcal{R}'$ over vectors in $\mathbb{F}_\rho^{d+1}$ that sample $\mathcal{L}$ and $\mathcal{R}$, respectively, to obtain the first d components of the vector, and then sample the final component uniformly over $\mathbb{F}_\rho$. We claim that linear tests do not distinguish $\mathcal{L}'$ from $\mathcal{R}'$.

Any linear test $\boldsymbol{x'_n} \in \mathbb{F}_{\rho(n)}^{d+1}$ that has a non-zero final component will not distinguish $\mathcal{L}'_n$ from $\mathcal{R}'_n$ because the final component of these two distributions is uniform, so the inner product will have the uniform distribution in both cases as well. Furthermore, if there exists a sequence of linear tests $\{\boldsymbol{x'_n}\} \in \mathbb{F}_\rho^{d+1}$ that have zero for the final component and distinguish $\mathcal{L}'$ from $\mathcal{R}'$, then the sequence $\{\boldsymbol{x_n}\} \in \mathbb{F}_\rho^d$ formed by deleting the final component from $\boldsymbol{x'_n}$ distinguishes $\mathcal{L}$ and $\mathcal{R}$, contradicting our assumption that $\mathcal{L}$ and $\mathcal{R}$ are indistinguishable by linear tests.

Finally, let A' be the adversary that drops its final component and feeds the rest to A. It is clear that $A'_{\mathcal{L}'} = A_{\mathcal{L}}$ and $A'_{\mathcal{R}'} = A_{\mathcal{R}}$, so $|A'_{\mathcal{L}'} - A'_{\mathcal{R}'}|$ is noticeable. Therefore, Assumption 5 for dimension $d+1$ is false as well. □

Proof (Lemma 10). Suppose that Assumption 6 holds. For every n, the assumption holds for a randomly chosen safe prime p, and thus for every n there exists some safe prime $p_n = 2q_n + 1$ for which it holds. Let G_n be the subgroup of quadratic residues in $\mathbb{F}_{p_n}^*$, and let $\mathcal{G} = \{G_n\}_{n\in\mathbb{N}}$. We claim that Assumption 5 for dimension 2 holds for the family $\mathcal{G}$ and prime sequence $\rho(n) = q_n$.

It is clear that the first two properties of Assumption 5 for dimension 2 hold. Also, using our convention that the first non-zero coordinate of a vector is fixed to be 1, we may assume without loss of generality that every vector in $\mathbb{F}_\rho^2$ has the form $(1, x)$ for $x \in \mathbb{F}_{q_n}$ except for the vector $(0, 1)$, which is easy to test for. Thus, a "vector" is really just a group element. Furthermore, a "linear test" is just an equality check because the vector $(y, -1)$ has an inner product of zero with the vector $(1, x)$ if and only if $y = x$.

Hence, it remains to prove the following: for every PPT adversary A and for all families of distributions $\mathcal{L}$ and $\mathcal{R}$ over $\{\mathbb{F}_{q_n}\}_{n\in\mathbb{N}}$ such that

$$\max_{x\in\mathbb{F}_{q_n}} |\Pr[l \leftarrow \mathcal{L} : l = x] - \Pr[r \leftarrow \mathcal{R} : r = x]|$$

is negligible, the quantity

$$|\Pr[l \leftarrow \mathcal{L}_n, g \overset{U}{\leftarrow} G_n : A(g, g^l) = 1] - \Pr[r \leftarrow \mathcal{R}_n, g \overset{U}{\leftarrow} G_n : A(g, g^r) = 1]|$$

is negligible as well.

First, we prove that the statement holds when $\mathcal{L}$ is well-spread and $\mathcal{R}$ is the uniform distribution. Hence, we wish to show that for all PPT A,

$$|\Pr[l \leftarrow \mathcal{L}_n, g \overset{U}{\leftarrow} G_n : A(g, g^l) = 1] - \Pr[r \overset{U}{\leftarrow} \mathbb{F}_{q_n}, g \overset{U}{\leftarrow} G_n : A(g, g^r) = 1]|$$

is negligible. The proof of this statement closely follows the proofs in [6], so we only sketch the details here. If this statement is not true, then the probability $P_x = A_{(1,x)} = \Pr[A(g, g^x) = 1]$ is noticeably different from the mean value $\bar{P} = A_{\mathcal{R}}$ for super-polynomially many values of x. Without loss of generality, there exist super-polynomially many values a for which P_a is noticeably larger than $\bar{P}$. Let X_{q_n} be the uniform distribution over all such a. Then, the ensembles $\langle g, g^a, g^b, g^c\rangle$ and $\langle g, g^a, g^b, g^{ab}\rangle$ are distinguishable when $a \leftarrow X_{q_n}$ by running A on the final two components of the ensemble. In the first case, A outputs 1 with probability $\bar{P}$, and in the second case, A outputs 1 with noticeably higher probability. This contradicts Assumption 6.

Next, we note that the statement immediately extends to the setting where both $\mathcal{L}$ and $\mathcal{R}$ are well-spread by a simple hybrid argument.

Finally, we consider arbitrarily distributions $\mathcal{L}$ and $\mathcal{R}$ such that

$$\max_{x \in \mathbb{F}_{q_n}} |\Pr[\boldsymbol{l} \leftarrow \mathcal{L} : l = x] - \Pr[r \leftarrow \mathcal{R} : r = x]|$$

is negligible. In words, this equation means that for every x that occurs with noticeable probability in $\mathcal{L}$, it occurs with the same probability in $\mathcal{R}$ as well up to a negligible difference. Thus, the distributions $\mathcal{L}$ and $\mathcal{R}$ can only differ on outcomes that occur with negligible probability. Therefore, it suffices to consider $\mathcal{L}$ and $\mathcal{R}$ that are well-spread, and in this case we showed that for every PPT A,

$$|\Pr[l \leftarrow \mathcal{L}_n, g \overset{U}{\leftarrow} G_n : A(g, g^l) = 1] - \Pr[r \leftarrow \mathcal{R}_n, g \overset{U}{\leftarrow} G_n : A(g, g^r) = 1]|$$

is negligible, so Assumption 5 for dimension 2 holds as desired. □

We note that a literal converse to this lemma does not quite make sense because Assumption 6 is specific to the group of quadratic residues modulo $\mathbb{F}_p^*$ for a safe prime p, whereas Assumption 5 makes the more general claim that there exists some family of groups that satisfy a certain condition (potentially quite different from the groups used in Assumption 6).

4 Construction

In this section, we define the family of programs that we obfuscate, present the obfuscator, and prove its security under Assumption 5.

Let d be an integer and ρ be a prime sequence. Given a vector $\boldsymbol{a} \in \mathbb{F}^d_{\rho(n)}$, let $H_{\boldsymbol{a}}$ be the circuit that has $\boldsymbol{a}$ hardwired, and on input $\boldsymbol{x} \in \mathbb{F}^d_{\rho(n)}$, computes $\langle \boldsymbol{a}, \boldsymbol{x} \rangle$ in the obvious way and accepts if and only if the inner product equals 0. Let $\mathcal{F}^d_\rho = \{H_{\boldsymbol{a}} : n \in \mathbb{N}, \boldsymbol{a} \in \mathbb{F}^d_{\rho(n)}\}$ be the family of all such circuits.

We show how to obfuscate the family $\mathcal{F}^d_\rho$ for any $d \in \mathbb{N}$, prime sequence ρ, and set of groups $\mathcal{G}$ (written multiplicatively) that satisfy Assumption 5 for dimension d. The obfuscator $\mathcal{O}_{\mathcal{G},d}$ operates as follows.

Algorithm 1 Obfuscator $\mathcal{O}_{\mathcal{G},d}$ for the family of hyperplanes $\mathcal{F}^d_\rho$

```
Input: vector $\boldsymbol{a} = (a_1, \ldots, a_d)$ in $\mathbb{F}^d_{\rho(n)}$
  choose a generator $g \overset{U}{\leftarrow} G_n \setminus \{1_{G_n}\}$ uniformly at random
  compute $g_i \leftarrow g^{a_i}$ for $i = 1, \ldots, d$
Output: circuit that has $g_1, \ldots, g_d$ hardwired, and on input a vector $\boldsymbol{x}$, accepts if and
  only if $\prod_{i=1}^d g_i^{x_i} = 1_{G_n}$
```

We stress that the generator g is not made public in addition to the g_i. However, recall that the vector $\boldsymbol{a}$ is only defined uniquely up to scalar multiplication, and that one way to enforce this requirement is to assume that the first non-zero coordinate of $\boldsymbol{a}$ equals 1. With this convention, the generator g is revealed.

This convention makes it clear that in the $d = 2$ case, this construction is the same as the one in [6], and it can be based off the same DDH assumption by Theorem 7. Hence, our construction subsumes the one in [6].

We note that in the work of Shen, Shi and Waters [13] on *private* inner-product predicate encryption schemes, their construction also tests whether an inner product is 0 by running it in the exponent of a group where CDH is hard. Otherwise the settings, constructions, and assumptions are quite different. In particular, a user who wants to check whether a vector $\boldsymbol{x}$ has inner product 0 with a hidden vector $\boldsymbol{v}$ needs to first encrypt $\boldsymbol{v}$ using a secret key (so their predicate encryption scheme does not directly yield an obfuscation).

We now show that $\mathcal{O}_{\mathcal{G},d}$ is an obfuscator, based on Assumption 5.

Theorem 11. *Let $d \in \mathbb{N}$ and $\mathcal{G}$ be a set of groups satisfying Assumption 5. Then, the algorithm $\mathcal{O}_{\mathcal{G},d}$ is an obfuscator for the family $\mathcal{F}_\rho^d$ with exact functionality.*

It is clear that $\mathcal{O}_{\mathcal{G},d}$ satisfies the exact functionality and polynomial slowdown properties required of an obfuscator, so it remains to prove the virtual black-box property. Before doing so, we present a definition that will be useful throughout the proof and an intermediate lemma.

Definition 12. *Let $d \in \mathbb{N}$ and p be a prime number. We say that the set $V \subseteq \mathbb{F}_p^d$* distinguishes *two vectors $\boldsymbol{l}, \boldsymbol{r} \in \mathbb{F}_p^d$ if there exists $\boldsymbol{x} \in V$ such that exactly one of the inner products $\langle \boldsymbol{l}, \boldsymbol{x} \rangle$ and $\langle \boldsymbol{r}, \boldsymbol{x} \rangle$ equals 0. Otherwise, we say that $\boldsymbol{l}$* and *$\boldsymbol{r}$* are indistinguishable by *V, which means that for all $\boldsymbol{x} \in V$, $\langle \boldsymbol{l}, \boldsymbol{x} \rangle = 0$ if and only if $\langle \boldsymbol{r}, \boldsymbol{x} \rangle = 0$.*

At a high level, this lemma states that for every adversary A, there exists a set V that can distinguish vectors in $\mathbb{F}_{\rho(n)}^d$ as well as A can.

Lemma 13. *Suppose $(\mathcal{G}, d)$ satisfy Assumption 5. For every PPT adversary A and polynomial ε, there exists a polynomial s (that can depend on A) such that for every $n \in \mathbb{N}$, there exists a set $V \subseteq \mathbb{F}_{\rho(n)}^d$ of size at most $s(n)$, such that for every pair of vectors $\boldsymbol{l}, \boldsymbol{r} \in \mathbb{F}_{\rho(n)}^d$ that are indistinguishable by V, $|A_{\boldsymbol{l}} - A_{\boldsymbol{r}}| < \frac{1}{\varepsilon(n)}$.*

Using standard techniques found in [6] and other papers, we can show that the lemma implies that $\mathcal{O}_{\mathcal{G},d}$ is an obfuscator.

Proof (Theorem 11 from Lemma 13). Let A be an adversary and ε be a polynomial, and we must construct a simulator S such that for every $n \in \mathbb{N}$ and every vector $\boldsymbol{r} \in \mathbb{F}_{\rho(n)}^d$,

$$|\Pr[A(\mathcal{O}_{\mathcal{G},d}(H_{\boldsymbol{r}})) = 1] - \Pr[S^{H_{\boldsymbol{r}}}(1^n) = 1]| < \frac{1}{\varepsilon(n)} \ .$$

By Lemma 13, there exists a polynomial s such that for every $n \in \mathbb{N}$, there exists a set $V \subseteq \mathbb{F}_{\rho(n)}^d$ of size at most $s(n)$ such that the property in the lemma holds. Let $S^{H_{\boldsymbol{r}}}(1^n)$ be the nonuniform circuit that receives V as advice and does the following:

```
for all $\boldsymbol{x} \in V$ do
    query the oracle on input $\boldsymbol{x}$ and record the response
end for
choose a vector $\boldsymbol{l} \in \mathbb{F}^d_{\rho(n)}$ such that $\forall \boldsymbol{x} \in V,\ \langle \boldsymbol{l}, \boldsymbol{x} \rangle = 0$ iff $H(\boldsymbol{x})$ accepts
output $A(\mathcal{O}_{\mathcal{G},d}(H_{\boldsymbol{l}}))$
```

Finally, since $\Pr[A(\mathcal{O}_{\mathcal{G},d}(H_{\boldsymbol{r}})) = 1] = A_{\boldsymbol{r}}$ by definition and $\Pr[S^{H_{\boldsymbol{r}}}(1^n) = 1] = A_{\boldsymbol{l}}$ by construction, Lemma 13 ensures that S satisfies the virtual black-box condition. □

Next, we provide some high-level intuition about why the lemma is true. Suppose there is an adversary A that breaks the obfuscation (and thus the lemma as well). We build a new adversary A^* that runs A many times. Also, we construct two distributions $\mathcal{L}$ and $\mathcal{R}$. These distributions will be uniform over their support, so we can really just think of them as sets. The construction of $\mathcal{L}$ and $\mathcal{R}$ proceeds iteratively, subject to two invariant conditions: first, A^* must be able to distinguish these distributions, and second, no linear test should do so. These constraints together violate Assumption 5.

We achieve the first invariant using the negation of Lemma 13, which continually gives us a pair of vectors $(\boldsymbol{l_i}, \boldsymbol{r_i})$ that A (and thus A^*) can distinguish. We add $\boldsymbol{l_i}$ to the support of $\mathcal{L}$ and $\boldsymbol{r_i}$ to the support of $\mathcal{R}$. The second invariant is achieved by continually monitoring $\mathcal{L}$ and $\mathcal{R}$ as they grow. We "trap" any linear test $\boldsymbol{x}$ once it is able to distinguish d of the pairs $(\boldsymbol{l_i}, \boldsymbol{r_i})$. Once we have identified such a linear test, we ensure that subsequent pairs of vectors that we add to $\mathcal{L}$ and $\mathcal{R}$ are indistinguishable by $\boldsymbol{x}$. Hence, any linear test only distinguishes a constant number of the pairs, so by making the distributions $\mathcal{L}$ and $\mathcal{R}$ well-spread, we ensure that all linear tests succeed with only negligible probability. The only downside to the proof is that the "trapping" procedure requires a simulator whose runtime is exponential in d, so the proof only holds for constant dimension.

The rest of this section is devoted to a formal proof of the lemma, which uses some techniques from the proofs in [6], some novel proof concepts, and some linear algebra. We use the set notation $[k] = \{1, 2, \ldots, k\}$ in this proof.

Proof (Lemma 13). Given $\mathcal{G}$ and d, assume for the sake of contradiction that the obfuscator $\mathcal{O}_{\mathcal{G},d}$ does not satisfy Lemma 13. Hence, there exists an adversary A and polynomial ε such that for all polynomials s, there exist infinitely many $n \in \mathbb{N}$ such that for every set $V \subseteq \mathbb{F}^d_{\rho(n)}$ of size at most $s(n)$, there exist vectors $\boldsymbol{l}, \boldsymbol{r} \in \mathbb{F}^d_{\rho(n)}$ with the property that $\langle \boldsymbol{l}, \boldsymbol{x} \rangle = 0$ if and only if $\langle \boldsymbol{r}, \boldsymbol{x} \rangle = 0$ for all $\boldsymbol{x} \in V$, such that $|A_{\boldsymbol{r}} - A_{\boldsymbol{l}}| \geq \frac{1}{\varepsilon(n)}$.

Because these probabilities are separated by a noticeable amount, an efficient algorithm is able to determine which of $A_{\boldsymbol{l}}$ and $A_{\boldsymbol{r}}$ is larger by taking n samples of each one (using independent randomness for A and the choice of $g \overset{U}{\leftarrow} G_n$ each time) and observing which sample probability is greater. By a Chernoff bound, this algorithm succeeds with overwhelming probability. Thus, from now on we assume without loss of generality that $A_{\boldsymbol{r}} > A_{\boldsymbol{l}}$, which allows us to drop the absolute value.

Given a constant c, apply this statement to the polynomial $s_c(n) = n^c$ and the resulting $n \in \mathbb{N}$ in order to build two large sets $\hat{L}_n^c$ and $\hat{R}_n^c$ iteratively as follows.

```
initialize $V \leftarrow \emptyset$ and $i \leftarrow 1$
while $|V| \leq n^c$ do
    given the set $V$, let $\boldsymbol{l_i}$ and $\boldsymbol{r_i}$ be vectors that violate Lemma 13
    insert $\boldsymbol{l_i} \in \hat{L}_n^c$ and $\boldsymbol{r_i} \in \hat{R}_n^c$
    for all subsets $T \subseteq \hat{L}_n^c \cup \hat{R}_n^c$ of size at most $d - 2$ do
        add to $V$ random bases of $(T \cup \{\boldsymbol{l_i}\})^\perp$ and $(T \cup \{\boldsymbol{r_i}\})^\perp$
    end for
    increment $i \leftarrow i + 1$
end while
```

This algorithm iteratively finds pairs of vectors that the adversary A can distinguish but the set V cannot. Then, it adds many points to V. We now describe in detail how these additional points affect future iterations of the loop.

When $T = \emptyset$ in the for loop, the algorithm adds to V a basis of vectors orthogonal to $\boldsymbol{l_i}$. Since $\boldsymbol{l_i}$ is the only vector (up to scalar multiplication) that is orthogonal to every vector in this basis, it follows that in all future iterations $i' > i$ of the loop, $\boldsymbol{l_{i'}}$ and $\boldsymbol{r_{i'}}$ are linearly independent from $\boldsymbol{l_i}$, because $\boldsymbol{l_{i'}}$ and $\boldsymbol{r_{i'}}$ must be indistinguishable by V. The same is true for $\boldsymbol{r_i}$, so the sets $\hat{L}_n^c$ and $\hat{R}_n^c$ are continually increasing in size.

When T is not equal to the empty set, the additional points added to V ensure that linear tests cannot distinguish $\hat{L}_n^c$ from $\hat{R}_n^c$. Specifically, we claim that for every vector $\boldsymbol{x} \in \mathbb{F}_{\rho(n)}^d$, there are at most d indices such that $\langle \boldsymbol{x}, \boldsymbol{l_i} \rangle = 0$ but $\langle \boldsymbol{x}, \boldsymbol{r_i} \rangle \neq 0$, or vice-versa.

To see this, suppose without loss of generality that there exists a vector $\boldsymbol{x} \in \mathbb{F}_{\rho(n)}^d$ and J indices $i_1 < i_2 < \cdots < i_J$ such that $\langle \boldsymbol{x}, \boldsymbol{l}_{\boldsymbol{i_j}} \rangle = 0$ but $\langle \boldsymbol{x}, \boldsymbol{r}_{\boldsymbol{i_j}} \rangle \neq 0 \; \forall j \in [J]$. We show by induction that the vectors $\boldsymbol{l}_{\boldsymbol{i_1}}, \dots, \boldsymbol{l}_{\boldsymbol{i_J}}$ are linearly independent. As the base case, we showed above that any two vectors from $\hat{L}_n^c \cup \hat{R}_n^c$ are linearly independent. Now, for $j \geq 2$ suppose that $S_j = \{\boldsymbol{l}_{\boldsymbol{i_1}}, \dots, \boldsymbol{l}_{\boldsymbol{i_j}}\}$ contains linearly independent vectors. At iteration i_j of the loop, a basis $\{\boldsymbol{b_1}, \dots, \boldsymbol{b_k}\}$ of the space $S_j^\perp$ is added to V. By definition, the basis vectors are linearly independent. If $\boldsymbol{l}_{\boldsymbol{i_{j+1}}}$ were linearly dependent on S_j, say $\boldsymbol{l}_{\boldsymbol{i_{j+1}}} = \alpha_1 \boldsymbol{l}_{\boldsymbol{i_1}} + \cdots + \alpha_j \boldsymbol{l}_{\boldsymbol{i_j}}$, then

$$\langle \boldsymbol{l}_{\boldsymbol{i_{j+1}}}, \boldsymbol{b_{i'}} \rangle = \langle \alpha_1 \boldsymbol{l}_{\boldsymbol{i_1}} + \cdots + \alpha_j \boldsymbol{l}_{\boldsymbol{i_j}}, \boldsymbol{b_{i'}} \rangle = \alpha_1 \langle \boldsymbol{l}_{\boldsymbol{i_1}}, \boldsymbol{b_{i'}} \rangle + \cdots + \alpha_j \langle \boldsymbol{l}_{\boldsymbol{i_j}}, \boldsymbol{b_{i'}} \rangle = 0$$

for all $i' \in [k]$. Because $\boldsymbol{l}_{\boldsymbol{i_{j+1}}}$ and $\boldsymbol{r}_{\boldsymbol{i_{j+1}}}$ are indistinguishable by V, it follows that $\langle \boldsymbol{r}_{\boldsymbol{i_{j+1}}}, \boldsymbol{b_{i'}} \rangle = 0$ for all $i' \in [k]$ as well, so

$$\boldsymbol{r}_{\boldsymbol{i_{j+1}}} \in \overline{\{\boldsymbol{b_1}, \dots, \boldsymbol{b_k}\}}^{\perp} = (S_j^\perp)^\perp = \overline{S_j}$$

by Theorem 3, which means that $\boldsymbol{r}_{\boldsymbol{i_{j+1}}}$ is linearly dependent on the vectors in S_j so $\langle \boldsymbol{x}, \boldsymbol{r}_{\boldsymbol{i_{j+1}}} \rangle = 0$. This contradicts the assumption that $\boldsymbol{x}$ distinguishes $\boldsymbol{l}_{\boldsymbol{i_{j+1}}}$ from $\boldsymbol{r}_{\boldsymbol{i_{j+1}}}$, so the vectors $\boldsymbol{l}_{\boldsymbol{i_1}}, \dots, \boldsymbol{l}_{\boldsymbol{i_{j+1}}}$ must be linearly independent, which completes the induction. The vectors come from a space with dimension d, so there can only be d linearly independent vectors, so $J \leq d$ as desired.

Next, we find a lower bound on the size of the sets $\hat{L}_n^c$ and $\hat{R}_n^c$. The loop condition is to stop when $|V| > n^c$. On each iteration of the loop, $|\hat{L}_n^c|$ and $|\hat{R}_n^c|$ each increase by 1 and $|V|$ increases by at most

$$2d \times \left[\sum_{k=0}^{d-2} \binom{|\hat{L}_n^c \cup \hat{R}_n^c|}{k}\right] \leq 2d^2 \binom{|\hat{L}_n^c \cup \hat{R}_n^c|}{d-2} \leq \mathrm{O}(|\hat{L}_n^c \cup \hat{R}_n^c|^{d-2}) \ ,$$

which means that the size of V is

$$|V| = \mathrm{O}(2^{d-2}) + \mathrm{O}(4^{d-2}) + \cdots + \mathrm{O}(|\hat{L}_n^c \cup \hat{R}_n^c|^{d-2}) = \mathrm{O}(|\hat{L}_n^c \cup \hat{R}_n^c|^{d-1}) \ .$$

We also know that $|V| \leq n^c$, so it follows that $|\hat{L}_n^c|$ and $|\hat{R}_n^c|$ are $\Omega(n^{c/d})$.

Consider the $2\varepsilon(n)$ intervals $[0, \frac{1}{2\varepsilon(n)}]$, $[\frac{1}{2\varepsilon(n)}, \frac{1}{\varepsilon(n)}]$, ..., $[1 - \frac{1}{2\varepsilon(n)}, 1]$ that partition the unit interval. We say that an interval $[\alpha, \beta]$ "separates" an $\boldsymbol{l_i}$, $\boldsymbol{r_i}$ pair if $A_{\boldsymbol{l_i}} < \alpha$ and $A_{\boldsymbol{r_i}} > \beta$. Since $A_{\boldsymbol{r_i}} - A_{\boldsymbol{l_i}} > \frac{1}{\varepsilon(n)}$, each pair is separated by at least one of the $2\varepsilon(n)$ intervals. Hence, by the pigeonhole principle, there exists one interval that separates a $\frac{1}{2\varepsilon(n)}$ fraction of the pairs. Call this interval $[\alpha_c^*, \beta_c^*]$. Let L_n^c and R_n^c be subsets of $\hat{L}_n^c$ and $\hat{R}_n^c$, respectively, consisting only of the $\boldsymbol{l_i}$, $\boldsymbol{r_i}$ pairs that are separated by $[\alpha_c^*, \beta_c^*]$. Note that $|L_n^c|$ and $|R_n^c|$ are $\Omega(\frac{n^{c/d}}{\varepsilon(n)})$.

Furthermore, there is an algorithm A_c^* that distinguishes L_n^c from R_n^c. It is nonuniformly hardcoded with the value $\mu_c^* = \frac{\alpha_c^* + \beta_c^*}{2}$, and operates as follows.

```
Input: a vector $\boldsymbol{v} \in \mathbb{F}_{\rho(n)}^d$
1: run $A(\mathcal{O}_{\mathcal{G},d}(H_{\boldsymbol{v}}))$ a total of $32n \cdot \varepsilon(n)^2$ times using fresh randomness for $A$
   and $\mathcal{O}$ each time
2: let $\tau$ denote the fraction of iterations that $A$ accepts
Output: "$L_n^c$" if $\tau \leq \mu_c^*$ and "$R_n^c$" otherwise
```

If the input to this algorithm is a vector $\boldsymbol{l} \in L_n^c$, then we know that $A_{\boldsymbol{l}} \leq \alpha_c^*$. By a Chernoff bound, the probability that the empirical acceptance rate τ is greater than $\mu_c^* = \alpha_c^* + \frac{1}{4\varepsilon(n)}$ is at most e^{-n}. The same is true for vectors in R_n^c, so this algorithm succeeds with probability $1 - e^{-n}$. On the other hand, we argued above that linear tests distinguish L_n^c from R_n^c with probability at most $\frac{d}{|L_n^c|} = \mathrm{O}(\frac{\varepsilon(n)}{n^{c/d}})$.

Finally, we construct the distributions $\mathcal{L}$ and $\mathcal{R}$ that break Assumption 5. Recall that the negation of Lemma 13 yields a function $n(c)$ as follows: for every poly s_c the lemma provides some value n of the security parameter where there is a counterexample to the lemma. Furthermore, we note that as $c \to \infty$, the sequence $\{n(c)\}_{c \in \mathbb{N}} \to \infty$ as well. This is due to the fact that if $c > nd$, then the lemma considers sets V of size up to $n^{nd} > \rho(n)^d$, so the entire collection of vectors in $\mathbb{F}_{\rho(n)}^d$ can fit in V and the lemma is obviously true in this case.

We form a sort of inverse to this function as follows: given n, let c_n be the biggest value of c such that the counterexample with c applies to n. Note that c_n is not well-defined for all values of n, but it is defined for infinitely large set of values which we will denote by $N \subseteq \mathbb{N}$. It follows from the above argument that as $n \to \infty$, the sequence $\{c_n\}_{n \in N} \to \infty$ as well. Hence, there exists an infinitely large subset $N' \subseteq N$ such that $\{c_n\}_{n \in N'}$ is monotonically increasing. We form the families of distributions $\mathcal{L}$ and $\mathcal{R}$ such that $\mathcal{L}_n$ and $\mathcal{R}_n$ are uniform over the sets $L_n^{c_n}$ and $R_n^{c_n}$, respectively, for all $n \in N'$. We set $\mathcal{L}_n = \mathcal{R}_n$ arbitrarily for all $n \notin N'$.

Consider the following unified adversary A^* that is nonuniformly hardcoded with the values $\mu_{c_n}^* = \frac{1}{2}(\alpha_{c_n}^* + \beta_{c_n}^*)$ for all $n \in N'$ (and arbitrarily values of $\mu_{c_n}^*$ for $n \notin N'$).

```
Input: a vector $\boldsymbol{v} \in \mathbb{F}_{\rho(n)}^d$
run $A(\mathcal{O}_{\mathcal{G},d}(H_{\boldsymbol{v}}))$ a total of $32n \cdot \varepsilon(n)^2$ times using fresh randomness for $A$ and $\mathcal{O}$ each time
let $\xi$ denote the fraction of iterations that $A$ accepts
Output: "$L_n$" if $\xi \leq \mu_{c_n}^*$ and "$R_n$" otherwise
```

This adversary will succeed at distinguishing $\mathcal{L}$ from $\mathcal{R}$ with overwhelming probability $1 - e^{-n}$ for all $n \in N'$ (and of course the adversary will fail on all $n \notin N'$). On the other hand, any sequence of linear tests only succeeds with probability $\mathrm{O}(\frac{\varepsilon(n)}{n^{c_n/d}})$ which is negligible since $c_n \to \infty$ as $n \to \infty$. Hence, there is no polynomial $q(n)$ that bounds the ratio of success probabilities for the infinitely many $n \in N'$, so Assumption 5 is false as desired. □

5 Obfuscation of hyperplanes with multi-bit output

Given an obfuscator for the family of point functions, the work of [10] shows how to construct an obfuscator for the family of point functions with multi-bit output. This family also accepts a single point, but instead of just having a yes or no output, it returns a hidden message on the correct input value. Such an obfuscator can be used to create a strong symmetric-key encryption scheme that satisfies leakage resilience and circular security [17]. Their construction applies in our case too, so we can obfuscate the family of "hyperplanes with multi-bit output," with the nice property that the message is not revealed when the input is the zero vector (the one vector that is known to be in every hyperplane).

Formally, let $H_{\boldsymbol{a},m}$ be the circuit that has the vector $\boldsymbol{a} \in \mathbb{F}_{\rho(n)}^d$ hardwired, and on input a vector $\boldsymbol{x} \in \mathbb{F}_{\rho(n)}^d$, outputs m if $\langle \boldsymbol{a}, \boldsymbol{x} \rangle = 0$ but $\boldsymbol{x} \neq \boldsymbol{0}$, and outputs $\perp$ otherwise. Let $\mathcal{M}_{\rho,l}^d = \{H_{\boldsymbol{a},m} : \boldsymbol{a} \in \mathbb{F}_{\rho(n)}^d, m \in \{0,1\}^{l(n)}\}$ be the family of all such circuits. In particular, we can think of the hyperplanes family $\mathcal{F}_\rho^d$ as a special case of this family where $l = 0$ (i.e. there is only one possible message).

We show how to obfuscate the family $\mathcal{M}_{\rho,l}^d$ given any $d \in \mathbb{N}$ and obfuscator for hyperplanes $\mathcal{O}_{\mathcal{G},d}$ that is $(l+1)$-composable.

Definition 14 (t-composable obfuscation [10]). *A PPT $\mathcal{O}$ is a t-composable obfuscator for the family $\mathcal{C}$ if functionality and polynomial slowdown hold as before, and the virtual black-box property holds whenever the adversary and simulator are given up to t circuits in $\mathcal{C}$. Formally, for every PPT adversary A and polynomial δ, there exists a PPT simulator S such that for all sufficiently large*

n, and for all $C_1, \dots, C_t \in \mathcal{C}_n$,

$$|\Pr[A(\mathcal{O}(C_1), \dots, \mathcal{O}(C_t)) = 1] - \Pr[S^{C_1, \dots, C_t}(1^n) = 1]| < \frac{1}{\delta(n)} \ ,$$

where the first probability is taken over the coin tosses of A and $\mathcal{O}$, and the second probability is taken over the coin tosses of S.

Unfortunately, we do not know how to prove from Assumption 5 that $\mathcal{O}_{\mathcal{G},d}$ is even 2-composable. All we can show is that the composability of $\mathcal{O}_{\mathcal{G},d}$ is related to the length of messages that we can obfuscate. Let $\tilde{\mathcal{O}}_{\mathcal{G},l,d}$ be an obfuscator for the family $\mathcal{M}^d_{\rho,l}$ that operates as follows.

Algorithm 2 Obfuscator $\tilde{\mathcal{O}}_{\mathcal{G},l,d}$ for the family $\mathcal{M}^d_{\rho,l}$

```
Input: vector $\boldsymbol{a} \in \mathbb{F}^d_{\rho(n)}$
set $C_0 = \mathcal{O}_{\mathcal{G},d}(\boldsymbol{a})$
for $i = 1$ to $l$ do
    if $m_i = 1$ then
        set $C_i = \mathcal{O}_{\mathcal{G},d}(\boldsymbol{a})$
    else
        choose $\boldsymbol{a'} \overset{U}{\leftarrow} \mathbb{F}^d_{\rho(n)}$ and set $C_i = \mathcal{O}_{\mathcal{G},d}(\boldsymbol{a'})$
    end if
end for
Output: circuit that hardwires $C_0, \dots, C_l$ and operates as follows on input $\boldsymbol{x} \in \mathbb{F}^d_{\rho(n)}$:
    output $\perp$ if $C_0(\boldsymbol{x})$ rejects or if $\boldsymbol{x} = \boldsymbol{0}$, otherwise output the string $s$ formed by
    $s_i = C_i(\boldsymbol{x})$ for $i = 1, \dots, l$
```

Theorem 15. *Suppose that the obfuscator $\mathcal{O}_{\mathcal{G},d}$ is $(l+1)$-composable for some $l = \mathrm{poly}(n)$, and let $\rho(n) = |G_n|$. Then, $\tilde{\mathcal{O}}_{\mathcal{G},l,d}$ is an obfuscator for the family of hyperplanes with multi-bit output $\mathcal{M}^d_{\rho,l}$ with approximate functionality.*

The proof of this theorem is similar to the one in [10].

Proof. The approximate functionality and polynomial slowdown of $\tilde{\mathcal{O}}_{\mathcal{G},l,d}$ are clear from the construction and the corresponding properties of $\mathcal{O}_{\mathcal{G},d}$. For $i = 1$ to l, let $\boldsymbol{a_i}$ be the vector such that $\boldsymbol{a_i} = \boldsymbol{a}$ if $m_i = 1$ or $\boldsymbol{a_i}$ is uniformly chosen otherwise. By the $(l+1)$-composable virtual black-box property, we know that there exists a simulator S such that the output of

$$A(\tilde{\mathcal{O}}_{\mathcal{G},l,d}(\boldsymbol{a})) = A(\mathcal{O}_{\mathcal{G},d}(\boldsymbol{a_1}), \dots, \mathcal{O}_{\mathcal{G},d}(\boldsymbol{a_d}))$$

can be simulated by $S^{H_{\boldsymbol{a_1}}, \dots, H_{\boldsymbol{a_d}}}$. Furthermore, the oracles $H_{\boldsymbol{a_1}}, \dots, H_{\boldsymbol{a_d}}$ can be simulated by the oracle $H_{\boldsymbol{a},m}$ up to a negligible simulation error in the following manner: if $H_{\boldsymbol{a},m}(\boldsymbol{x}) = \perp$, then we say that $H_{\boldsymbol{a_i}}(\boldsymbol{x}) = 0$ for all i. Otherwise $H_{\boldsymbol{a},m}(\boldsymbol{x}) = m$, in which case we say that $H_{\boldsymbol{a_i}}(\boldsymbol{x}) = m_i$. Hence, the simulator $T^{H_{\boldsymbol{a},m}}$ that runs $S^{H_{\boldsymbol{a_1}}, \dots, H_{\boldsymbol{a_d}}}$ and emulates the oracle queries in this manner satisfies the virtual black-box property for $\tilde{\mathcal{O}}_{\mathcal{G},l,d}$. □

6 One-time signature schemes

We can use an obfuscator for the family of planes in three-dimensional space to form a one-time signature scheme. Informally, the secret and public keys are a hidden plane and an obfuscation of the plane membership testing program, respectively. A signature of a message is a point on the hyperplane that is related to the message, and the verification procedure runs the obfuscated hyperplane testing circuit to verify signatures.

More formally, let ρ be a prime sequence, and $\mathcal{O}$ be an obfuscator for the family of hyperplanes over $\mathbb{F}_\rho^3$ (such as the obfuscator $\mathcal{O}_{\mathcal{G},d}$ constructed in Section 4). Consider the following three algorithms.

KeyGen(1^n): Choose field elements $sk_1, sk_2, c \overset{U}{\leftarrow} \mathbb{F}_{\rho(n)} \setminus 0$. Form the vector $\boldsymbol{sk} = (sk_1, sk_2, 1)$ in $\mathbb{F}_{\rho(n)}^3$ and the obfuscated plane $P = \mathcal{O}(\boldsymbol{sk})$. The secret key is (sk_1, sk_2), and the public key is (P, c).

Sign($m \in \mathbb{F}_{\rho(n)}$): Let σ_2 be the unique field element such that the inner product $\langle \boldsymbol{sk}, (cm, \sigma_2, 1)\rangle = 0$. The signature is (cm, σ_2).

Verify($m, (\sigma_1, \sigma_2)$): Accept if and only if $\sigma_1 = cm$ and $P(\sigma_1, \sigma_2, 1)$ accepts.

This signature scheme is unforgeable in a weak sense, described in [14] and other works, in which the forger must choose the message on which she requests a signature before being shown the public key. The techniques in [14] allow us to transform this scheme into one that is existentially unforgeable under chosen message attacks (the standard security notion for signature schemes). The transformation requires a chameleon hash function whose seed can be chosen with public coins. It is known how to construct such a hash function under the DDH assumption [18].

Furthermore, our one-time signature scheme is resilient to any leakage function whose output length is less than half as long as the secret key.

Theorem 16. *Let ρ be a prime sequence and $\mathcal{O}$ be an obfuscator for the family of hyperplanes over the vector space $\mathbb{F}_\rho^3$. Then, the above algorithm leads to an existentially unforgeable one-time signature scheme that is resilient to any leakage function whose output length is bounded by*

$$l(n) = n - \omega(\log(n)) \ .$$

In particular, leakage of $l(n) = \gamma n$ bits for any $\gamma < 1$ is permitted.

This theorem is proved in [16]. We note that the leakage bound in the theorem is tight. Consider the following leakage function that has a message m hardcoded: use the secret key to form a signature associated to m, and output σ_2. This leakage function has n bits of output, and permits a forgery of the message m by the signature (cm, σ_2).

The secret key consists of two elements of $\mathbb{F}_{\rho(n)}$, so it is $2n$ bits long. Thus, our signature scheme permits leakage of up to half of the length of the secret key. This matches the leakage bound attained in [15] for schemes that do not use general non-interactive zero-knowledge proofs (albeit under a much stronger assumptions).

Acknowledgments. The authors would like to thank the anonymous referees for their helpful feedback.

References

1. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im)possibility of obfuscating programs. In: CRYPTO. Volume 2139 of Lecture Notes in Computer Science., Springer (2001) 1–18
2. Goldwasser, S., Rothblum, G.N.: On best-possible obfuscation. In: TCC. Volume 4392 of Lecture Notes in Computer Science., Springer (2007) 194–213
3. Hofheinz, D., Malone-Lee, J., Stam, M.: Obfuscation for cryptographic purposes. In: TCC. Volume 4392 of Lecture Notes in Computer Science., Springer (2007) 214–232
4. Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely obfuscating re-encryption. In: TCC. Volume 4392 of Lecture Notes in Computer Science., Springer (2007) 233–252
5. Goldwasser, S., Kalai, Y.T.: On the impossibility of obfuscation with auxiliary input. In: FOCS, IEEE Computer Society (2005) 553–562
6. Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: CRYPTO. Volume 1294 of Lecture Notes in Computer Science., Springer (1997) 455–469
7. Canetti, R., Micciancio, D., Reingold, O.: Perfectly one-way probabilistic hash functions. In: Proceedings of the 30th ACM Symposium on Theory of Computing. (1998) 131–140
8. Lynn, B., Prabhakaran, M., Sahai, A.: Positive results and techniques for obfuscation. In: EUROCRYPT. Volume 3027 of Lecture Notes in Computer Science., Springer (2004) 20–39
9. Wee, H.: On obfuscating point functions. In: Proceedings of the 37th ACM Symposium on Theory of Computing. (2005) 523–532
10. Canetti, R., Dakdouk, R.R.: Obfuscating point functions with multibit output. In: EUROCRYPT. Volume 4965 of Lecture Notes in Computer Science., Springer (2008) 489–508
11. Dodis, Y., Smith, A.: Correcting errors without leaking partial information. In: STOC. (2005) 654–663
12. Adida, B., Wikström, D.: How to shuffle in public. In: TCC. Volume 4392 of Lecture Notes in Computer Science., Springer (2007) 555–574
13. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: TCC. (2009) 457–473
14. Hohenberger, S., Waters, B.: Short and stateless signatures from the rsa assumption. In: CRYPTO. Volume 5677 of Lecture Notes in Computer Science., Springer (2009) 654–670
15. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: ASIACRYPT. (2009)
16. Canetti, R., Rothblum, G., Varia, M.: Obfuscation of hyperplane membership. Cryptology ePrint Archive (2009) http://eprint.iacr.org/.
17. Canetti, R., Kalai, Y., Varia, M., Wichs, D.: On symmetric encryption and point obfuscation. In: TCC. Lecture Notes in Computer Science, Springer (2010)
18. Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS, The Internet Society (2000)