Another Step Towards Realizing Random Oracles: Non-Malleable Point Obfuscation

Ilan Komargodski* Eylon Yogev†

Abstract

The random oracle paradigm allows us to analyze the security of protocols and constructions in an idealized model, where all parties have access to a truly random function. This is one of the most popular and well-studied models in cryptography. However, being such a strong idealized model, it is known to be susceptible to various weaknesses when implemented naively in "real-life", as shown by Canetti, Goldreich and Halevi (J. ACM 2004).

As a counter-measure, one could try to identify and implement only one or few of the properties a random oracle possesses that are needed for a specific setting. Such a systematic study was initiated by Canetti (CRYPTO 1997), who showed how to implement the property that the output of the function does not reveal anything regarding the input by constructing a point function obfucator. This property turned out to suffice in many follow-up works and applications.

In this work, we tackle another natural property of random oracles and implement it in the standard model. The property we focus on is *non-malleability*, where it is required that the output on an input cannot be used to generate an output on any related point. We construct a point obfuscator that is both hiding (à la Canetti) *and* is non-malleable for a non-trivial class of mauling functions. Our construction does not use heavy cryptographic machinery (such as zero-knowledge proofs) and is comparable to that of Canetti in terms of time complexity and obfuscation size. The security of our construction relies on variants of the DDH and power-DDH assumptions.

On the technical side, we introduce a new technique for proving security of a construction based on a DDH-like assumption. We call this technique "double-exponentiation" and believe it will be useful in the future.

*Cornell Tech, New York, NY 10044, USA. Email: komargodski@cornell.edu. Supported in part by a Packard Foundation Fellowship and by an AFOSR grant FA9550-15-1-0262. Initial parts of this work were done at the Weizmann Institute of Science, supported in part by a grant from the Israel Science Foundation (no. 950/16) and by a Levzion Fellowship.

†Department of Computer Science and Applied Mathematics, Weizmann Institute of Science Israel, Rehovot 76100, Israel. Email: eylon.yogev@weizmann.ac.il. Supported in part by a grant from the Israel Science Foundation (no. 950/16).

1 Introduction

The Random Oracle model [BR93] is one of the most well studied models in the cryptographic literature. In this model, everyone has access to a single random function. It is usually possible to show clean and simple constructions that are information-theoretically secure in this idealized model. Also, in many cases it allows to prove unconditional lower bounds.

One major question is when (and under what assumptions) can we replace the Random Oracle with a "real life" object. It is known that such a transformation is impossible in the general case, but the counter examples are usually quite contrived [CGH04, GK03, BFM15]. This leaves the possibility that for specific applications of a Random Oracle such a transformation could possibly exist. One of the obstacles in answering the aforementioned question is that it seems hard to formalize and list all the properties such a generic transformation should preserve. In practice, this difficulty is circumvented by replacing the Random Oracle with an ad-hoc "cryptogrpahic hash function" (e.g., MD5, SHA-1, SHA-256) which results with protocols and constructions that have no provable security guarantees, and often tend to be broken [WY05, WYY05, SBK+17].

Motivated by the above, Canetti [Can97] initiated the systematic study of identifying useful properties of a Random Oracle and then realizing them in the standard model. In his work, he focused on one property called "point obfuscation" (or "oracle hashing"). This property ensures that when the Random Oracle is applied on an input, the output value is completely uncorrelated to the input, and at the same time, it is possible to verify whether a given output was generated from a given input. Canetti formally defined this notion and gave a construction of such a primitive in the standard model based on a variant of the decisional Diffie-Hellman assumption (DDH). Since then, other instantiations of this primitive were suggested. Wee [Wee05] gave a construction whose security is based on a strong notion of one-way permutations, Goldwasser et al. [GKPV10] gave a construction based on the Learning With Errors assumption, and more recently Bellare and Stepanovs [BS16] proposed a framework for constructing point obfuscators. The latter result gives a generic construction of point obfuscators based on either (1) indistinguishability obfuscation [BGI+12, GGH+13] and any one-way function, (2) deterministic public-key encryption [BBO07], or (3) UCEs [BHK13].

While hiding the point is a natural and useful goal, there are many setting where this is not enough to replace a Random Oracle. One other natural property we wish to realize in "real life" is that of *non-malleability*: given the value of a Random Oracle on a random point x, it is infeasible to get the value of the Random Oracle at any "related" point (e.g., the point $x + 1$). The work of Canetti and Varia [CV09] identified this property and the goal of realizing it. Their work provided definitions (of non-malleable obfuscation for general circuits, and not only for point functions) and constructions of non-malleable (multi) point obfuscators in the random oracle model.

In this work, we focus on constructing non-malleable point obfuscators in the plain model. Observe that many of the known constructions of point obfuscators *are* malleable. For example, let us recall the construction of Canetti [Can97] which involves a group $\mathcal{G}$ with a generator $g \in \mathcal{G}$. For an input point x and randomness r (interpreted as a random group element) the obfuscation is:

$$O(x; r) = (r, r^x).$$

Indeed, the obfuscation of $x + 1$ can be computed by multiplying r^x by r and outputting the pair (r, r^{x+1}). In other words, the obfuscation of a point is *malleable*. The point obfuscators of Wee [Wee05] and of Goldwasser et al. [GKPV10] admit similar attacks (i.e., they are malleable).[1]

[1]The work of [BS16] is an exception since it gives constructions based on generic primitives so we need non-malleability of the underlying building block. The required notion of non-malleability is usually very strong. Consider, for example, their construction from DPKE, where the point function obfuscation includes a ciphertext and a public-

Thus, we ask whether we can remedy this situation and provide a construction of a secure point obfuscator in the plain model that is provably non-malleable under simple and concrete assumptions. We view this as a necessary and vital step towards understanding the possibility for realizing a Random Oracle in "real life".

1.1 Our Results

We provide a construction of a secure point obfuscator that is non-malleable for a wide class of mauling functions. Our notion of non-malleability is parametrized by a distribution $\mathcal{X}$ over the input domain X and by a class of possible mauling attacks $\mathcal{F} = \{f : X \to X\}$. Roughly speaking, our notion guarantees that for every function $f \in \mathcal{F}$, any polynomial-time adversary, when given the obfuscation of a point $x \leftarrow \mathcal{X}$, cannot generate the obfuscation of the point $f(x)$.[2]

We give a construction of a (public-coin[3]) point obfuscator that is non-malleable for any well-spread distribution $\mathcal{X}$ (i.e., a distribution that has super-logarithmic min-entropy) and the class of mauling functions $\mathcal{F}$ which can be described by univariate polynomials of bounded polynomial degree (in the security parameter). Our construction involves a group $\mathcal{G}$ with a generator $g \in \mathcal{G}$. For an input point x and randomness r (interpreted as a random group element) the obfuscation is:

$$O(x; r) = (r, r^{g^{h(x)}}),$$

where $h(x) = x^4 + x^3 + x^2 + x$. We prove security and non-malleability of the above point obfuscator under variants of the DDH and power-DDH assumptions (see Section 2.2). We also present two ways to support more general mauling functions $\mathcal{F}$ by strengthening the underlying security assumption (yet the construction remains the same). First, we show how to support a larger class of mauling function by assuming (sub-)exponential security of the underlying assumption. Second, we show that our construction is secure against any mauling function f for which one cannot distinguish the triple $(g, g^x, g^{h(f(x))})$ from a triple (g, g^{r_1}, g^{r_2}), where r_1, r_2 are random exponents. We do not have a simple characterization of the functions f for which this assumption holds.

In terms of efficiency, our construction is quite efficient: it involves only two group exponentiation (Canetti's construction requires a single exponentiation), does not rely on any setup assumptions, and does not rely on expensive machinery such as zero-knowledge proofs, which are usually employed to achieve non-malleability. Moreover, it satisfies the same *privacy* guarantees as of Canneti's obfuscator. As such, our point obfuscator can be used in any application where point obfuscators are used. These include encryption schemes [Can97], storing passwords [WG00, CV09], reusable fuzzy extractors [CFP+16], round-efficient zero-knowledge proofs and arguments [BP12], and more.

Applications to non-interactive non-malleable commitments. It is possible to view our obfuscator as a non-interactive non-malleable commitment that is secure when committing to strings that come from a distribution with super-logarithmic entropy. To commit to a string x, compute the obfuscation of x and that would be the commitment. The opening is x itself (and thus for security it has to have entropy). The resulting commitment scheme is computationally hiding by the security of the point obfuscator, and also non-malleable against a large class of mauling functions.

key (of some encryption scheme). To get non-malleability for the point obfuscator we need non-malleability for the DPKE for an adversary that can maul not only the ciphertext but also the public-key.

[2] We also require that the obfuscation that the adversary outputs is verifiable, that is, it looks like an obfuscation of the value $f(x)$ (i.e., it comes from the "same family" of circuits). This prevents trivial attacks that treat the input circuit as a black-box.

[3] An obfuscator is public-coin if the random bits used for the obfuscation are given as part of the output of the obfuscator.

Previously, constructions of non-interactive non-malleable commitments (in the plain model, without any setup assumptions) required an ad-hoc and non-standard primitive called "adaptive injective one-way functions" that has built-in some form of non-malleability [PPV08]. More recent works provide constructions that are secure against uniform adversaries [LPS17] or ensure limited forms of non-malleability ("with respect to opening") [KS17]. These constructions, however, allow to commit on worst-case inputs and handle arbitrary mauling functions.

1.2 Related Work

Non-malleable cryptography. Non-malleability was introduced as a measure to augment and strengthen cryptographic primitives (such as encryption schemes or commitment schemes) in such a way that it does not only guarantee privacy, but also that it is hard to manipulate a given ciphertext (or commitment) of one value into a ciphertext of another.

Non malleability was first defined in the seminal work of Dolev, Dwork, and Naor [DDN03] where they presented a non-malleable public-key encryption scheme, a non-malleable string commitment scheme, a non-malleable zero-knowledge protocol. Since then, there has been a long line of works on non-malleability. See [PSV06, BGR$^+$15, GKS16, GPR16, PR08, LP15, Pas16, LPS17, LPS17, KS17] to name just a few.

A particular type of non-malleable protocols (or primitives) that may a-priori be related to non-malleable point obfuscators are *non-interactive* commitments and encryption schemes. These were the focus of multiple works (see, for example, [DDN03, Sah99, CKOS01, FF11] and some of the references given above). However, these notions do not imply point obfuscators as they do not support public verification on a given input (without revealing the randomness which completely breaks security).

In the context of obfuscation, the only work we are aware of is that of Canetti and Varia [CV09] who gave several incomparable definitions for non-malleable obfuscation. They also gave a construction of a (multi-bit) non-malleable point obfuscator (under each definition), however, their construction is in the Random Oracle model.

A related work to ours is the one of Applebaum, Harnik, and Ishai [AHI11] which studies the security of encryption schemes under related-key attacks (RKA). In this model, the adversary needs to break an encryption scheme by invoking it with several secret-keys which satisfy some known relation. Their RKA-secure scheme is based on the power-DDH assumption and the proof of security resembles some of our techniques.

Obfuscation with high min-entropy. Canetti, Micciancio and Reingold [CMR98] gave a construction of a point obfuscator that satisfies a relaxed notion of security where the input is guaranteed to come from a source with high min-entropy. Their underlying assumption is any collision resistant hash function. There is a significant (qualitative) difference between this notion and the original notion of Canetti [Can97] that we consider in this work. We refer to Wee [Wee05, Section 1.3] for an elaborate discussion.

Boldyreva et al. [BCFW09] showed how to make the point obfuscator of [CMR98] non-malleable using non-interactive zero-knowledge proofs (assuming a common reference string). Following the work of Boldyreva et al., Baecher et al. [BFS11] presented a game-based definition of non-malleability which is very similar to ours (see also [CQZ$^+$16]). However, they did not provide new constructions in the plain model.

1.3 Our Techniques

Our starting point is Canetti's point function construction [Can97], who presented a construction under a variant of the DDH assumption (and no random oracles). Recall that the DDH assumption involves a group ensemble $\mathcal{G} = \{\mathbb{G}_\lambda\}_{\lambda\in\mathbb{N}}$ with a generator g and it asserts that (g^x, g^y, g^{xy}) is computationally indistinguishable from a sequence of random group elements, where x and y are chosen uniformly at random. Canetti's variant is that the foregoing indistinguishability holds even if x has high enough min-entropy (yet y is completely random). For an input point x and using randomness r, viewed as a random group element of $\mathbb{G}_\lambda$, Canetti's construction is:

$$O(x;r) = r, r^x.$$

As we mentioned, it is easy to modify r^x to get r^{x+1}, giving an obfuscation of the point $x+1$. Let us first focus on the goal of modifying the construction such that it is non-malleable against this function: $f(x) = x+1$. Towards this end, we change the construction to be:

$$O(x;r) = r, r^{x^2}.$$

The claim is that under a suitable variant of the power-DDH assumptions this is a non-malleable point obfuscator against the function f. Roughly speaking, we assume that $(g^x, g^{x^2}, g^{x^3}, \ldots)$ is indistinguishable from a sequence of random group elements, where x comes from a distribution with high enough min-entropy. Assume first that the adversary outputs a point obfuscation of $x+1$ under the *same randomness* r as she received. That is, on input r, w, the output is r, w' for an element $w' \in \mathbb{G}$. Later, we show how to handle adversaries that output an obfuscation of $x+1$ under new randomness.

The point obfuscation of $x+1$ under this construction (with the same randomness r) is (r, w), where $w = r^{x^2+2x+1}$. Suppose that there is an adversary $\mathcal{A}$ that given r, r^{x^2} can compute w, then we show how to break the security of our assumption. We are given a challenge (g, g^{z_1}, g^{z_2}), where either $z_i = x^i$ or $z_i = r_i$ and each r_i is chosen at random. Then, we can run the adversary on the input g^s, g^{sz_2}, for a random s to get w. We compute $w' = g^{s(z_2+2z_1+1)}$ and compare it to w. If $w = w'$ we output 1, and otherwise we output a random bit. In the case that $z_i = x^i$, the adversary gets g^s, g^{sx^2} which is exactly the distribution of a point obfuscation of x and thus will output $w = g^{s(x^2+2x+1)} = w'$ with some non-negligible probability. Otherwise, the adversary gets g^{sr_2} for a random r_2 and the probability that she outputs $w' = g^{s(r_2+2r_1+1)}$ is negligible as she has no information regarding r_2 (this is true even for an unbounded adversaries). Overall, we have a non-negligible advantage in distinguishing the two cases.

While the above construction is non-malleable against the function $f(x) = x+1$, it is malleable for the function $f(x) = 2x$. Indeed, given r^{x^2} one can simply compute $(r^{x^2})^4 = r^{4x^2} = r^{(2x)^2}$ which is a valid obfuscation of the point $2x$. Our second observation is that we can modify the construction to resist this attack by defining:

$$\mathcal{O}(x;r) = r, r^{x^2+x}.$$

The proof of non-malleability is similar to the proof above; we run the adversary $\mathcal{A}$ on $g^s, g^{s(z_2+z_1)}$ to get w, and compute $w' = g^{s(4z_2+2z_1)}$. If $z_i = x^i$, then the adversary sees exactly the distribution of a point obfuscation of x and thus will output $w = g^{s(4z_2+2z_1)} = w'$ with some non-negligible probability. Otherwise, the adversary gets $g^{s(r_2+r_1)}$ for random r_i's. We bound the probability that $\mathcal{A}$ outputs $w' = g^{s(4r_2+2r_1)}$. This is again an information theoretic argument where we assume that the adversary gets r_2+r_1 and needs to compute $4r_2+2r_1$. The argument follows since the adversary has only information regarding the sum r_2+r_1 which leaves the random variable corresponding to

$4r_2 + 2r_1$ with high min-entropy (given the adversary's view), and thus the probability of outputting $w = w'$ is negligible.

One important thing to notice is that the proof relied on the fact that the adversary only had the sum $r_1 + r_2$ which is a linear combination of (r_1, r_2) with the coefficients $(1, 1)$ but the final goal was to output a different combination with the coefficients $(4, 2)$, which are linearly independent of $(1, 1)$. That is, the key observation is that for $h(x) = x^2 + x$ the polynomial $h(f(x))$ for $f(x) = 2x$ has (non-free) coefficients which are not all the same. Generalizing this argument, we can show that the construction is non-malleable against any linear function $f(x) = ax + b$ for any constants a, b such that the function $h(f(x))$ written as a polynomial over x has at least 2 different (non-free) coefficients. For non-linear functions, a similar proof works but the running time of the security reduction (that is, the loss in the security of our scheme) will be proportional to the degree of $f(x)$.

Given the above observation, we can easily check if our construction is non-malleable for a function f by computing the polynomial $h(f(x))$. It turns our that the above construction is actually *malleable* for a simple function such as $f(x) = 3x + 1$. Indeed, $h(f(x)) = (3x + 1)^2 + (3x + 1) = 9x^2 + 9x + 2$ has the same two non-free coefficients. In order to eliminate more functions f, we need to add more constraints to the set of equations which translates to taking a higher degree of polynomial $h(x)$. That is, we define $h(x) = x^3 + x^2 + x$, and construct the obfuscator:

$$\mathcal{O}(x; r) = r, r^{x^3+x^2+x}.$$

For a function f to be malleable under this construction, it must hold that the polynomial $h(f(x))$ has all *three* non-free coefficients equal. However, there is still single function that satisfies this condition (the function is $f(x) = -x - 2 \cdot 3^{-1}$, where 3^{-1} is the inverse of 3 in the relevant group). As a final step, we modify the construction to be of one degree higher and this does eliminate all possible functions f. Thus, we define the construction:

$$\mathcal{O}(x; r) = r, r^{x^4+x^3+x^2+x}.$$

The double exponentiation. In our exposition above, we assumed that the adversary "uses the same randomness she received". That is, on input r, w she mauls the point and outputs r, w'. Suppose now that the adversary is allowed to output r', w', where r' might be arbitrary. Recall that the issue is that we cannot simulate the power of w' from the challenge under the randomness r' to check consistency (since we do not know the discrete log of r'). Let us elaborate on this in the simple case where the obfuscation is r, r^x (and not the degree 4 polynomial in the exponent; this is just for simplicity). When the malleability adversary gets r, r^x and returns r, w', it is easy to check that $w' = r^{f(x)}$ by recomputing this value since we know the discrete log of r. However, when it return r', w', it is hard to recompute $r'^{f(x)}$ since we do not know the discrete log of r (and only get the value x in the exponent from the challenge).

In other words, we need to be able (in the security proof) to compute the obfuscation of some input that depends on the exponents from the challenge under randomness that comes from the adversary's mauled obfuscation. If we knew either the discrete log of the challenge or the discrete log of the randomness used by the adversary we would be done.

In the description above we actually used this property. Since we assumed that the adversary outputs the same randomness r (that we chose and know the discrete log of), we could use $r = g^s$ to compute the obfuscation of the challenge we received. However, if the adversary outputs randomness r', then not only we no longer know the discrete log of r' (and this is hard to compute), but we also do not have the discrete log of the challenge.

Thus, we need to modify our construction such that we can compute the obfuscation of x given only g^x and while given the public coins r explicitly (without given their discrete log). Towards this

end, we introduce a new technique that we call "double exponentiation". Consider any mapping of the group elements $\mathbb{G}_\lambda \to \mathbb{Z}_q^*$ where q is the order of $\mathbb{G}_\lambda$ (e.g., their binary representation as strings). Then, we define the final version of our construction:

$$\mathcal{O}(x;r) = r, r^{g^{x^4+x^3+x^2+x}}.$$

One can observe that it is possible to compute the obfuscation of x given only $g^{x^4+x^3+x^2+x}$ and given r by a single exponentiation. In addition, the construction is still efficient, consists of just two group elements, and involves only two exponentiations.

A final remark about security. Proving that the resulting construction is still a point obfuscator is not immediate a-priori. Our proof works by a reduction to the security of Canetti's construction via an intermediate notion of security called virtual gray-box obfuscation [BC14]. We refer to Section 4 for more details.

2 Preliminaries

For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. For a randomized function f and an input $x \in \mathcal{X}$, we denote by $y \leftarrow f(x)$ the process of sampling a value y from the distribution $f(x)$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$.

Throughout the paper, we denote by λ the security parameter. A function $\mathsf{neg}\colon \mathbb{N} \to \mathbb{R}^+$ is *negligible* if for every constant $c > 0$ there exists an integer N_c such that $\mathsf{neg}(\lambda) < \lambda^{-c}$ for all $\lambda > N_c$. Two sequences of random variables $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are *computationally indistinguishable* if for any probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that $\left|\Pr[\mathcal{A}(1^\lambda, X_\lambda) = 1] - \Pr[\mathcal{A}(1^\lambda, Y_\lambda) = 1]\right| \leq \mathsf{neg}(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

2.1 Point Obfuscation

For an input $x \in \{0,1\}^n$, the point function $I_x\colon \{0,1\}^n \to \{0,1\}$ outputs 1 on input x and 0 everywhere else. A point obfuscator is a compiler that gets a point x as input and outputs a circuit that has the same functionality as I_x but where x is (supposedly) computationally hidden. Let us recall the definition of security of Canetti [Can97] (called there *oracle simulation*).

Definition 2.1 (Functional equivalence)**.** We say that two circuits C and C' are *functionally equivalent* and denote it by $C \equiv C'$ if they compute the same function (i.e., $\forall x : C(x) = C'(x)$).

Definition 2.2 (Point obfuscation)**.** A point obfuscator $\mathcal{O}$ for a domain $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ of inputs is a probabilistic polynomial-time algorithm that gets as input a point $x \in X_\lambda$, and outputs a circuit C such that

1. **Completeness:** For all $\lambda \in \mathbb{N}$ and all $x \in X_\lambda$, it holds that

$$\Pr[\mathcal{O}(x) \equiv I_x] = 1,$$

where the probabilities are over the internal randomness of $\mathcal{O}$.

2. **Soundness:** For every probabilistic polynomial-time algorithm $\mathcal{A}$, and any polynomial function $p(\cdot)$, there exists a probabilistic polynomial-time simulator $\mathcal{S}$, such that for every $x \in X_\lambda$, any predicate $P\colon X_\lambda \to \{0,1\}$, and all large enough $\lambda \in \mathbb{N}$,

$$\left|\Pr[\mathcal{A}(\mathcal{O}(x)) = P(x)] - \Pr[\mathcal{S}^{I_x}(1^\lambda) = P(x)]\right| \leq \frac{1}{p(\lambda)},$$

where the probabilities are over the internal randomness of $\mathcal{A}$ and $\mathcal{O}$, and $\mathcal{S}$, respectively.

The obfuscation is called *public coin* if it publishes its internal coin tosses as part of its output.

Indistinguishability-based secrity. Another way to formalize the security of a point obfuscator is via an indistinguishability-based security definition (rather than simulation-based). Canetti [Can97] suggested such a definition (termed *distributional indistinguishability* there): the input comes from a distribution $\mathcal{X}_\lambda$ over the input space X_λ and the guarantee is that for any adversary $\mathcal{A}$ that outputs a single bit, the following two distributions are computationally indistinguishable:

$$(x, \mathcal{A}(\mathcal{O}(x;r))) \approx_c (x, \mathcal{A}(\mathcal{O}(y;r))), \tag{2.1}$$

where r is the randomness (chosen uniformly) for the point obfuscator and x and y are chosen independently from $\mathcal{X}_\lambda$.

One of Canetti's results [Can97, Theorem 4] was that the indisinguishability-based definition is *equivalent* to the simulation-based definition given in Equation (2.1) if the indisinguishability-based security holds with respect to all distributions that have super-logarithmic min-entropy (over the message space). Such a distribution is called a *well-spread distribution*:

Definition 2.3 (Well-spread distribution)**.** An ensemble of distributions $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$, where $\mathcal{X}_\lambda$ is over $\{0,1\}^\lambda$, is well-spread if

1. it is efficiently and uniformly samplable – there is a probabilistic polynomial-time algorithm that given 1^λ as input, outputs a sample according to $\mathcal{X}_\lambda$.

2. for all large enough $\lambda \in \mathbb{N}$, it has super-logarithmic min-entropy. Namely,

$$\mathsf{H}_\infty(\mathcal{X}_\lambda) = -\min_{x\in\{0,1\}^\lambda} \log_2 \Pr[X = x] \geq \omega(\log \lambda).$$

Canetti's construction. In [Can97], Canetti provided a construction that satisfies Definition 2.2. In his construction, the domain of inputs X_λ is $\mathbb{Z}_p$ for prime $p \approx 2^\lambda$. Let $\mathcal{G} = \{\mathbb{G}_\lambda\}_{\lambda\in\mathbb{N}}$ be a group ensemble with uniform and efficient representation and operations, where each $\mathbb{G}_\lambda$ is a group of prime order $p \in (2^\lambda, 2^{\lambda+1})$. The *public coin* point obfuscator $\mathcal{O}$ for points in the domain $\mathbb{Z}_p$ is defined as follows: $\mathcal{O}(I_x)$ samples a random generator $r \leftarrow \mathbb{G}_\lambda^*$ and outputs the pair (r, r^x). Evaluation of the obfuscation at point z is done by checking whether $r^x = r^z$.

Canetti proved that this construction satisfies Equation (2.1) for any well-spread distribution under the strong variant of the DDH assumption, that we review below (see Assumption 2.6). Thereby, the result is that under the same assumption his construction satisfies Definition 2.2, as well.

2.2 Hardness Assumptions

The DDH and Power-DDH assumptions. The DDH assumption says that in a suitable group, the triple of elements (g^x, g^y, g^{xy}) is pseudorandom for random x and y. The power-DDH assumption says that the power sequence $(g, g^x, g^{x^2}, \ldots, g^{x^t})$ is pseudorandom, for a random x and a polynomially bounded t. While the power-DDH assumption is less common in the literature, there are many works that explicitly rely on it (see, for example, [GJM02, Gen06, CNS07, AHI11]). To the best of our knowledge, the power-DDH assumption is *incomparable* to the DDH assumption.

Throughout this section, let $\mathcal{G} = \{\mathbb{G}_\lambda\}_{\lambda\in\mathbb{N}}$ be a group ensemble with uniform and efficient representation and operations, where each $\mathbb{G}_\lambda$ is a group of prime order $p \in (2^{\lambda-1}, 2^\lambda)$.

Assumption 2.4 (DDH)**.** *The DDH assumption asserts that for the group $\mathbb{G}_\lambda$ with associated generator g, the ensembles (g^x, g^y, g^{xy}) and (g^x, g^y, g^z) are computationally indistinguishable, where $x, y, z \leftarrow \mathbb{Z}_p^*$.*

Assumption 2.5 (Power-DDH)**.** *The power-DDH assumption asserts that for the group $\mathbb{G}_\lambda$ with associated generator g, for every polynomially bounded function $t(\cdot)$, the ensembles $(g, g^x, g^{x^2} \ldots, g^{x^t})$ and $(g, g^{r_1}, g^{r_2} \ldots, g^{r_t})$ are computationally indistinguishable, where $x, r_1, \ldots, r_t \leftarrow \mathbb{Z}_p^*$.*

We need an even stronger variant of both assumptions. The strong variant that we need, first proposed by Canetti [Can97], roughly, says that DDH is hard not only if x, y and z are chosen uniformly at random, but even if x is chosen from a distribution with enough min-entropy (i.e., a well-spread distribution; see Definition 2.3). Analogously, we define a strong variant of the power-DDH assumption where x is chosen from such a distribution rather than from the uniform one.

Assumption 2.6 (Strong DDH and power-DDH)**.** *The* strong *variant of the DDH (resp. power-DDH) assumption is when the two distributions are computationally indistinguishable even if x is chosen uniformly from a well-spread distribution $\mathcal{X}_\lambda$ (rather than from $\mathbb{Z}_p^*$).*

3 Non-Malleable Point Obfuscation

We define non-malleability of point function obfuscators. Such obfuscators not only hide the obfuscated point, but they also (informally) ensure that an obfuscation of a point x cannot be transformed into an obfuscation of a related (yet different) point.

There are several ways to formalize this notion of security. We focus on a notion of security where the objective of the adversary, given an obfuscation of x, is to come up with a circuit (of prescribed structure) that is a point function on a related point (a similar definition is given in [BFS11]). We discuss the relation to the notions of Canetti and Varia [CV09] below.

Definition 3.1 (Verifier)**.** A PPT algorithm $\mathcal{V}$ for a point obfuscator $\mathcal{O}$ for the ensemble of domains $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ domain is called a *verifier* if for all $\lambda \in \mathbb{N}$ and all $x \in X_\lambda$, it holds that $\Pr[\mathcal{V}(\mathcal{O}(x)) = 1] = 1$, where the probability is taken over the randomness of $\mathcal{V}$ and $\mathcal{O}$.

Notice that there is no guarantee as to what $\mathcal{V}$ is suppose to output when its input is not a valid obfuscation. In particular, a verifier that always outputs 1 is a legal verifier. In many cases, including the obfuscator of Canetti [Can97] and our own, one can define a meaningful verifier.

Definition 3.2 (Non-malleable point function)**.** Let $\mathcal{O}$ be a point obfuscator for an ensemble of domains $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ with an associated verifier $\mathcal{V}$. Let $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}} = \{f\colon X_\lambda \to X_\lambda\}_{\lambda\in\mathbb{N}}$ be an ensemble of families of functions, and let $\{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ be an ensemble of distributions over X.

The point obfuscator $\mathcal{O}$ is a *non-malleable obfuscator* for $\mathcal{F}$ and $\mathcal{X}$ if for any polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\mathsf{neg}(\cdot)$, such that for any $\lambda \in \mathbb{N}$ it holds that:

$$\Pr\left[\mathcal{V}(C)=1,\ f\in\mathcal{F}_\lambda,\ \text{and } I_{f(x)}\equiv C \;\middle|\; \begin{array}{l} x\leftarrow\mathcal{X}_\lambda \\ (C,f)\leftarrow\mathcal{A}(\mathcal{O}(x)) \end{array}\right] \leq \mathsf{neg}(\lambda).$$

That is, the adversary $\mathcal{A}$, given an obfuscation of a point x sampled from $\mathcal{X}_\lambda$, cannot output a function $f \in \mathcal{F}_\lambda$ and a valid-looking obfuscation of the point $f(x)$, except with negligible probability.

The verifier $\mathcal{V}$. We require that an attacker outputs an obfuscation with a prescribed structure so that it passes the verifier $\mathcal{V}$. Without such a requirement, there is a trivial attack for the adversary: use the given circuit $\widehat{C}_w$ to create a new circuit that gets x, computes $f^{-1}(x)$ and then applies $\widehat{C}_w$ on this value. The result is a circuit that accepts the point $f(w)$.

In general, it might be hard to come up with a verifier $\mathcal{V}$ that tests whether a given circuit is legal, but here we are interested in the case where this can be done efficiently. In our case, it will be very easy to define $\mathcal{V}$ since a "valid-looking" obfuscation will consist of all pairs of group elements (in some given group).

Adaptivity of f. We stress that our definition is adaptive with respect to the family $\mathcal{F}_\lambda$. That is, the adversary first gets to see the obfuscation $\mathcal{O}(x)$ of the point x and then may chose the function it wishes to maul to. This definition is stronger than a static version in which the function f is fixed and known in advance (before the adversary sees the challenge).

3.1 Relation to Canetti-Varia

The work of Canetti and Varia [CV09] presented a systematic study of non-malleable obfuscation both specifically for point functions and also for general functionalities. They gave two definitions for non-malleability, called *functional non-malleability* and *verifiable non-malleability*.

The verifiable non-malleability definition is more related to ours since there they also require that there is a verifier $\mathcal{V}$ that gets an alleged obfucated circuit and checks whether it is a legitimate output of the obfuscator. Recall that the obfuscator of Canetti (as well as ours) has this property: An obfuscation can be verified by simply checking whether the obfuscation consists of two group elements in the desired group.

The verifiable non-malleability notion of Canetti and Varia asserts that, roughly, whatever mauling attack one can apply on an obfuscation, there exists a simulator that has only oracle access to the input circuit and outputs a "similarly mauled" obfuscation. To prevent trivial attacks (that treat the input circuit as a black-box), they allow the simulator to output a circuit that has oracle gates to its own oracle (namely, to the input circuit). The verifiability ensures that the output of the adversary (and the simulator) have a "legal" structure. The precise definition is subtle and it captures a wide range of mauling attacks in a meaningful way. We refer to [CV09] for their elaborate discussions on the matter. We provide their formal definition, restricted to point functions next.

Definition 3.3 (Verifiable non-malleable point obfuscation [CV09])**.** Let $\mathcal{O}$ be a point obfuscator for a domain $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ with an associated verifier $\mathcal{V}$. For every PPT adversary $\mathcal{A}$ and every polynomial $p(\cdot)$, there exists a PPT simulator $\mathcal{S}$ such that for all sufficiently large $\lambda \in \mathbb{N}$, for any input $x \in X_\lambda$ and any polynomial-time computable relation $E\colon X_\lambda \times X_\lambda \to \{0,1\}$ (that may depend on x), it holds that

$$\begin{aligned}&\Pr\left[C\neq\mathcal{O}(x),\ \mathcal{V}(C)=1 \text{ and } (\exists y\in X_\lambda \text{ s.t. } I_y\equiv C \text{ and } E(x,y)=1) \mid C\leftarrow\mathcal{A}(\mathcal{O}(x))\right] - \\ &\Pr\left[\mathcal{V}(C)=1 \text{ and } \left(\exists y\in X_\lambda \text{ s.t. } I_y\equiv C^{I_x} \text{ and } E(x,y)=1\right) \mid C\leftarrow\mathcal{S}^{I_x}(1^\lambda)\right] \leq \frac{1}{p(\lambda)}.\end{aligned}$$

We observe that our definition is related to the above definition albeit with the following modifications. First, the input for our obfuscator is sampled from a well-spread distribution, rather than being worst-case. Second, the non-malleablility in our definition is parametrized with a family of functions, whereas the above definition requires non-malleability for all possible relations. The modified definition is given next.

Definition 3.4. Let $\mathcal{O}$ be a point obfuscator for a domain $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ with an associated verifier $\mathcal{V}$. Let $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}} = \{f\colon X_\lambda \to X_\lambda\}_{\lambda\in\mathbb{N}}$ be an ensemble of families of functions, and let $\{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ be an ensemble of distributions over X. For every PPT adversary $\mathcal{A}$ and every polynomial $p(\cdot)$, there exists a PPT simulator $\mathcal{S}$ such that for all sufficiently large $\lambda \in \mathbb{N}$, for any function $f \in \mathcal{F}_\lambda$, it holds that

$$\Pr_{x\leftarrow\mathcal{X}_\lambda}\left[C \neq \mathcal{O}(x),\, \mathcal{V}(C) = 1 \text{ and } I_{f(x)} \equiv C \mid C \leftarrow \mathcal{A}(\mathcal{O}(x))\right] -$$
$$\Pr_{x\leftarrow\mathcal{X}_\lambda}\left[\mathcal{V}(C) = 1 \text{ and } I_{f(x)} \equiv C^{I_x} \mid C \leftarrow \mathcal{S}^{I_x}(1^\lambda)\right] \leq \frac{1}{p(\lambda)}.$$

Definition 3.4 is a special case of Definition 3.3 since it has restrictions on the input to the obfuscator and the set of relations it supports. In the next claim, we show that our notion of non-malleability from Definition 3.2 implies the notion from Definition 3.4.

Claim 3.5. *A point obfuscator satisfying Definition 3.2 with respect to an ensemble of families of functions $\mathcal{F}$ and an ensemble of distributions $\mathcal{X}$ also satisfies Definition 3.4 with respect to $\mathcal{F}$ and $\mathcal{X}$.*

Proof. Let $\mathcal{O}$ be an obfuscator that satisfies Definition 3.2 with respect to the function in $\mathcal{F}$ and the distribution $\mathcal{X}$. Thus, for any $f \in \mathcal{F}$, there is no PPT adversary that can generate a valid-looking circuit C such that $I_{f(x)} \equiv C$ for $x \leftarrow \mathcal{X}$, except with negligible probability. Namely,

$$\Pr_{x\leftarrow\mathcal{X}}\left[C \neq \mathcal{O}(x),\, \mathcal{V}(C) = 1 \text{ and } I_{f(x)} \equiv C \mid C \leftarrow \mathcal{A}(\mathcal{O}(x))\right] \leq \mathsf{neg}(\lambda).$$

Hence, a simulator that does nothing (say, outputs $\perp$) will satisfy security requirement of Definition 3.4. ∎

A discussion. Our definition is thus, morally, equivalent to the strong definition of [CV09], albeit with the assumption that the input comes from a well-spread distribution and the mauling is restricted to functions rather than relations. Getting a construction in the plain model that resolves these two issues is left as an open problem.

Lastly, observe that in the above proof, the simulator is in fact independent of the adversary $\mathcal{A}$ and independent of the distinguishability gap (the polynomial $p(\cdot)$). Thus, we actually get one simulator for all adversaries and the computational distance between the output of the adversary and the output of the simulator is negligible.

4 Our Obfuscator

Let $\lambda \in \mathbb{N}$ be the security parameter and let $X_\lambda = \mathbb{Z}_{2^\lambda}$ be the domain. Let $\mathcal{F}_{\mathsf{poly}} = \{f\colon X_\lambda \to X_\lambda\}_{\lambda\in\mathbb{N}}$ be the ensemble of classes of all functions that can be computed by polynomials of degree $\mathsf{poly}(\lambda)$, except the constant functions and the identity function.

Let $\mathcal{G} = \{\mathbb{G}_\lambda\}_{\lambda\in\mathbb{N}}$ be a group ensemble with uniform and efficient representation and operations, where each $\mathbb{G}_\lambda$ is a group of prime order $q \in (2^{\lambda-1}, 2^\lambda)$. We assume that for every $\lambda \in \mathbb{N}$ there

is a canonical and efficient mapping between the elements of $\mathbb{G}_\lambda$ and the domain X_λ. Let g be the generator of the group $\mathbb{G}_{5\lambda}$. Our obfuscator gets as input an element $x \in X_\lambda$ and randomness $r \in \mathbb{G}_{5\lambda}$ and computes:

$$\mathcal{O}(x;r) = \left(r, r^{g^{x^4+x^3+x^2+x}}\right).$$

The verifier $\mathcal{V}$ for a valid-looking obfuscation is the natural one: it checks whether the obfuscation consists of merely two group elements in $\mathbb{G}_{5\lambda}$. In the next two theorems we show that our obfuscator is both secure and non-malleable. The first part is based on the strong DDH assumption (Assumptions 2.4 and 2.6) and the second is based on (Assumptions 2.5 and 2.6). Thus, overall, our obfuscator is both secure and non-malleable under the assumption that there is a group where the strong DDH and strong power-DDH assumptions hold.

Theorem 4.1. *Under the strong DDH assumption (Assumptions 2.4 and 2.6), the obfuscator $\mathcal{O}$ above is a point obfuscator according to Definition 2.2.*

Theorem 4.2. *Let $\mathcal{X}_\lambda$ be any well-spread distribution over X_λ. Under the strong power-DDH assumption (Assumptions 2.5 and 2.6), the obfuscator $\mathcal{O}$ above is non-malleable according to Definition 3.2 for the family of functions $\mathcal{F}_{\mathsf{poly}}$ and the distribution $\mathcal{X}_\lambda$.*

The proofs of these theorems appear in the following two subsections.

4.1 Proof of Theorem 4.1

For completeness, we first notice that for any $x \in X_\lambda$ it holds that $x^4 + x^3 + x^2 + x \leq 2^{5\lambda}$ and thus for any distinct $x, y \in X_\lambda$ it holds that $y^4 + y^3 + y^2 + y \neq x^4 + x^3 + x^2 + x$. Therefore, we get that for every $x \in X_\lambda$ it holds that $\mathcal{O}(x) \equiv I_x$, as required.

To prove soundness, we reduce to the security of our construction to the security of the r, r^x construction of Canetti [Can97]. We prove the following general claim regarding point function obfuscators.

Claim 4.3. *Let $f\colon X_\lambda \to X'_\lambda$ be an injective polynomial-time computable function, and let $\mathcal{O}$ be a secure point obfuscator. Then, $\mathcal{O}'(x) = \mathcal{O}(f(x))$ is also a secure point obfuscator.*

Proof. We prove that for any probabilistic polynomial-time algorithm $\mathcal{A}$, there is a probabilistic polynomial-time simulator $\mathcal{S}$ and a negligible function $\mathsf{neg}(\cdot)$, such that for all $x \in X_\lambda$ and all $\lambda \in \mathbb{N}$,

$$\left|\Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}'(x)) = 1] - \Pr_{\mathcal{S}}[\mathcal{S}^{I_x}(1^\lambda)) = 1]\right| \leq \mathsf{neg}(\lambda),$$

where the probabilities are over the internal randomness of $\mathcal{A}, \mathcal{O}$ and $\mathcal{S}$.

Let $\mathcal{A}$ be such an adversary and let $\mathcal{S}$ be the corresponding simulator whose existence is guaranteed by the fact that $\mathcal{O}$ is a secure point obfuscator. It holds that for every $x \in X_\lambda$:

$$\left|\Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(x)) = 1] - \Pr_{\mathcal{S}}[\mathcal{S}^{I_x}(1^\lambda)) = 1]\right| \leq \mathsf{neg}(\lambda),$$

As a first step, we construct a simulator $\mathcal{S}'$ that is inefficient yet makes only a polynomial-number of queries to its oracle (we will get rid of this assumption later using a known transformation). We define a simulator $\mathcal{S}'$ (with oracle access to I_x) that works by simulating $\mathcal{S}$ as follows. When $\mathcal{S}$ performs a query y to its oracles, then $\mathcal{S}'$ finds x' such that $f(x') = y$. If no such x' exists, then $\mathcal{S}'$

replies with 0. Otherwise, if $\mathcal{S}'$ found such an x', then it performs the query to its oracle with x' and answers with the reply of the oracle. Since f is injective, we have that $f(x) = y$ if and only if $x' = x$. Thus, it holds that

$$\Pr_{\mathcal{S}}[\mathcal{S}^{I_{f(x)}}(1^\lambda)) = 1] = \Pr_{\mathcal{S}'}[\mathcal{S}'^{I_x}(1^\lambda)) = 1].$$

Thus, we get that

$$\left| \Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}'(x)) = 1] - \Pr_{\mathcal{S}'}[\mathcal{S}'^{I_x}(1^\lambda)) = 1] \right| \leq \mathsf{neg}(\lambda).$$

We are left to take care of the fact that the simulator is inefficient. For this we use a result of Bitansky and Canetti [BC14] who showed that this can be solved generically. Let us elaborate.

Bitansky and Canetti called obfuscators in which the simulation is inefficient yet the number of queries is bounded by a polynomial as *gray-box obfuscation*. This is in contrast to virtual-black box obfuscation where the simulator is required to be both efficient in its running time and the number of queries and indistinguishability obfuscation [BGI+12, GGH+13], which can be phrased as a simulation-based definition where the simulator is unbounded in both running time and number of queries (see [BC14, Proposition 3.1]). One of the main results of Bitansky and Canetti was that for point functions, the virtual-black box and virtual-gray box notions are equivalent: a simulator that runs in unbounded time yet makes a polynomial number of queries can be turned into one that runs in polynomial-time and makes a polynomial number of queries.[4]

Using their result for our construction we obtain a simulator that works in polynomial-time and makes a polynomial number of queries to its oracle. This completes the claim. ■

We finish the proof by applying the claim with $f(x) = g^{x^4+x^3+x^2+x}$, noticing that this function is injective and efficiently computable.

4.2 Proof of Theorem 4.2

Assume that there exists an adversary $\mathcal{A}$, and a distribution $\mathcal{X}_\lambda$ such that given an obfuscation of a point $x \leftarrow \mathcal{X}_\lambda$, the adversary $\mathcal{A}$ outputs a function $f \in \mathcal{F}_{\mathsf{poly}}$ and a valid-looking obfuscation (i.e., an obfuscation that passes the verification of $\mathcal{V}$) of $f(x)$ with probability at least $\varepsilon > 0$. Denote by $t = t(\lambda)$ the degree of f (written as a polynomial over X_λ). We show how to construct an adversary $\mathcal{A}'$ that breaks the strong power-DDH assumption for the power sequence of length $T = 4t$.

Suppose we are given $(g^{z_0}, g^{z_1}, \ldots, g^{z_T})$, where $z_0 = 1$ and either $\forall i \in [T] : z_i = x^i$ for a random $x \leftarrow X_\lambda$ or $\forall i \in [T] : z_i = r_i$ for random $r_1, \ldots, r_t \leftarrow X_\lambda$. Our goal is to show that $\mathcal{A}'$ can distinguish between the two cases. The algorithm $\mathcal{A}'$, on input $(g^{z_0}, \ldots, g^{z_T})$, first samples a random generator $r \leftarrow \mathbb{G}$ and computes $g^{z_1+z_2+z_3+z_4}$. Then, it runs $\mathcal{A}$ on the input pair $(r, r^{g^{z_1+z_2+z_3+z_4}})$ to get a function f and an output pair $(r_{\mathcal{A}}, w_{\mathcal{A}})$. We assume that we are given the coefficients of the polynomial that represents the function f, as otherwise we can learns these coefficients by interpolation of random evaluations of f (according to the distribution of the inputs $\mathcal{X}_\lambda$).

Let $h(x) = x^4 + x^3 + x^2 + x$ and let us write the polynomial $h(f(x))$ as a polynomial of degree at most $4t$ with coefficients b_i:

$$h(f(x)) = (f(x))^4 + (f(x))^3 + (f(x))^2 + f(x) = \sum_{i=0}^{4t} b_i x^i.$$

The algorithm $\mathcal{A}'(g^{z_0}, g^{z_1}, \ldots, g^{z_T})$**:**

1. Choose a random generator $r \leftarrow \mathbb{G}$ and compute $g^{z_1+z_2+z_3+z_4}$.
2. $(f, r_{\mathcal{A}}, w_{\mathcal{A}}) \leftarrow \mathcal{A}(r, r^{g^{z_1+z_2+z_3+z_4}})$.
3. Compute the coefficients b_i for $i \in [T]$ of $h(f(x))$.
4. Compute $w_{\mathsf{real}} = r_{\mathcal{A}}^{g^{\sum_{i=0}^{T} b_i z_i}}$.
5. If $w_{\mathsf{real}} = w_{\mathcal{A}}$, then output 1. Otherwise, output 0.

Figure 1: The adversary $\mathcal{A}'$ that breaks the power-DDH assumption.

Using these values, it computes $u = g^{\sum_{i=0}^{T} b_i z_i}$ and $w_{\mathsf{real}} = r_{\mathcal{A}}^{u}$. Finally, the adversary $\mathcal{A}'$ outputs 1 if and only if $w_{\mathsf{real}} = w_{\mathcal{A}}$. The precise description of $\mathcal{A}'$ is given in Figure 1.

We argue that $\mathcal{A}'$ successfully breaks the power-DDH assumption.

The real case. Observe that if $z_i = x^i$ for each $i \in [T]$, then the distribution that $\mathcal{A}$ sees is exactly the distribution $(r, r^{g^{x^4+x^3+x^2+x}})$ and thus with probability at least ε, the adversary $\mathcal{A}$ will maul the point obfuscation of x to a point obfuscation of $f(x)$. That is,

$$w_{\mathcal{A}} = r_{\mathcal{A}}^{g^{h(f(x))}} = r_{\mathcal{A}}^{g^{\sum_{i=0}^{T} b_i x^i}} = r_{\mathcal{A}}^{g^{\sum_{i=0}^{T} b_i z_i}} = w_{\mathsf{real}}.$$

Thus, $\mathcal{A}'$ will output 1 with probability at least ε.

The random case. Suppose that $z_i = r_i$ is random for each $i \in [T]$. We show that the probability that $w_{\mathsf{real}} = w_{\mathcal{A}}$ is negligible (in λ). This is an information theoretic claim that holds against unbounded adversaries. The adversary $\mathcal{A}$ holds r and $r^{g^{r_1+r_2+r_3+r_4}}$ and let us even assume that she knows $s = r_1 + r_2 + r_3 + r_4$. In order for $\mathcal{A}'$ to succeed, she needs to be able to compute $s' = \sum_{i=0}^{T} b_i r_i$ (recall that $\mathcal{A}'$ is unbounded). We show that the min-entropy of this value s' given all the information of the adversary is high and therefore it cannot guess it with noticeable probability. Denote by $\mathsf{view}(\mathcal{A})$ a random variables that correspond to the view of $\mathcal{A}$ and denote by S' a random variable that corresponds to the value of s'.

We first show that if the degree of f (denoted above by t) is at least 2, then the min-entropy of S' is at least λ. This means that $\mathcal{A}'$ will be able to guess it with only negligible probability.

Claim 4.4. *If $t \geq 2$, then $\mathsf{H}_\infty(S' \mid \mathsf{view}(\mathcal{A})) \geq \lambda$.*

Proof. If the degree of f is at least 2, then the degree of $h(f(\cdot))$ is at least 5 and thus there exist $i > 4$ such that $b_i \neq 0$. In this case, since r_i is uniform in $\mathcal{X}_\lambda$, then the random variable s' has min-entropy λ given the view of $\mathcal{A}$. ∎

The case where f is a linear function (i.e., a degree 1 polynomial) is slightly harder to handle and here we use properties of the exact choice of our degree 4 polynomial. Let f be written as $f(x) = ax + b$ for some fixed $a, b \in X_\lambda$. We expand the polynomial $h(f(x))$ and rewrite it by

[4] See [BCKP14] for more general families of functions where a similar equivalence holds.

grouping terms:

$$\begin{aligned} h(f(x)) =& (ax+b)^4 + (ax+b)^3 + (ax+b)^2 + (ax+b) \\ =& a^4x^4 + (4a^3b + a^3)x^3 + (6a^2b^2 + 3a^2b + a^2)x^2 + \\ & (4ab^3 + 3ab^2 + 2ab + a)x + b^4 + b^3 + b^2 + b. \end{aligned}$$

We show that the coefficients of $h(f(\cdot))$ cannot be all identical.

Claim 4.5. *The coefficients of h are not all identical.*

Proof. If they were identical, then

$$a^4 = 4a^3b + a^3 = 6a^2b^2 + 3a^2b + a^2 = 4ab^3 + 3ab^2 + 2ab + a.$$

Solving this set of equations gives that the only solutions are $a = 0, b = *$ (i.e., b is arbitrary) and $a = 1, b = 0$ (i.e., the identity function). However, these are illegal according to our definition of $\mathcal{F}_{\mathsf{poly}}$: this class contains neither constant functions nor the identity function. ■

Using the fact that the coefficients are not all identical, we claim that the min-entropy of S' is at least λ even given the view of $\mathcal{A}$. Thus, again, the probability of guessing correctly the value is negligible.

Claim 4.6. *Let $R_1, R_2, R_3, R_4 \leftarrow X_\lambda$ be random variable whose distribution is uniform from $\mathcal{X}_\lambda$, and let their sum be $S = R_1 + R_2 + R_3 + R_4 \in X_{6\lambda}$. Let $b_1, b_2, b_3, b_4 \in X_\lambda$ be arbitrary constants such that at least two of them are different. Let $S' = b_1R_1 + b_2R_2 + b_3R_3 + b_4R_4$. Then, $\mathsf{H}_\infty(S' \mid S) \geq \lambda$.*

Proof. We lower bound the min entropy by computing $\Pr[S' = s' \mid S = s]$ for each $s, s' \in X_\lambda$. This probability is exactly the fraction of possible r_1, r_2, r_3, r_4 such that $r_1 + r_2 + r_3 + r_4 = s$ and $b_1r_1 + b_2r_2 + b_3r_3 + b_4r_4 = s'$. Writing this in matrix form we have

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ b_1 & b_2 & b_3 & b_4 \end{bmatrix}}_{A} \cdot \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} s \\ s' \end{bmatrix}.$$

Denote by Q the size of the support of $\mathcal{X}_\lambda$ and notice that $Q \geq 2^\lambda$. Since $\mathcal{X}_\lambda$ is well-spread, its min-entropy is super logarithmic in λ and thus the support size is super polynomial in λ. Since not all the b_i's are equal, we have that A's rank is 2, and thus the solution dimension is 2 for each $s' \in X_\lambda$ and the number of possible solutions is Q^2 out of the total Q^4 possibilities. Altogether, we get that for every $s' \in X_\lambda$, it holds that $\Pr[S' = s' \mid S = s] = Q^2/Q^4 \leq 1/Q < 1/2^\lambda$. Thus, the min-entropy is at least λ. ■

Combining the above, we get that overall, the probability of distinguishing is:

$$\left|\Pr[\mathcal{A}'(g^{x^1}, \ldots, g^{x^T}) = 1] - \Pr[\mathcal{A}'(g^{r_1}, \ldots, g^{r_T}) = 1]\right| \geq \varepsilon - \mathsf{neg}(\lambda)$$

which contradicts the security of the power-DDH assumption.

4.3 Supporting More Functions

In our construction above, we have shown how to get a point function obfuscator that is non-malleable against any function that can be written as a univariate polynomial of a polynomial degree. The reason that there is a bound on the degree of the polynomial is that the security reduction runs in time that is proportional to the degree. In particular, to be resilient against a function f of degree t we had to construct $g^{h(f(x))}$ in the reduction given the sequence $\{g^{x^i}\}_{i=0}^{4t}$ (recall that $h(x) = x^4 + x^3 + x^2 + x$).

Exponential security. Suppose that the min-entropy of the inputs is k. Thus, the support-size of the distribution is at most 2^k and hence any function can be written as a polynomial of degree at most 2^k. That is, we can assume without loss of generality that the mauling function is described by a degree $t \leq 2^k$ polynomial. Thus, if we assume an exponential version of the strong power-DDH assumption, where the adversary's running time and advantage are bounded by $2^{O(k)}$ and $2^{-\Omega(k)}$, respectively, we can support functions of exponential degree (in k).

Uber assumption. Instead of building the polynomial $h(f(x))$ in the proof monomial by monomial in order to break the power-DDH assumption, we can, alternatively, modify our assumption to get a more direct security proof without the large security loss. Concretely, instead of having the reduction computing $g^{h(f(x))}$ given $\{g^{z_i}\}_{i=0}^{4t}$, where t is the degree f, we assume an "uber" power-DDH assumption that is parametrized by a class of functions $\mathcal{F} = \{f \colon \mathbb{Z}_p \to \mathbb{Z}_p\}$ (and thus can thought of as a collection of assumptions, one per $f \in \mathcal{F}$). The assumption says that for any $f \in \mathcal{F}$, the following two distributions are computationally-indistinguishable:

$$(g, g^x, g^{h(f(x))}) \approx_c (g, g^x, g^s),$$

where $x \leftarrow \mathcal{X}$ and $s \leftarrow \mathbb{Z}_p^*$ is chosen at random. Having such an assumption for a class of mauling functions $\mathcal{F}$, implies that our construction is non-malleable for the same class $\mathcal{F}$.

Acknowledgments

We thank the anonymous reviewers of EUROCRYPT 2018 for their elaborate and useful comments. We are grateful to Ran Canetti for multiple useful suggestions and feedback about this work. Thanks to Nir Bitansky, Abhishek Jain, and Omer Paneth for multiple discussions. Lastly, we thank Moni Naor for his encouragement, support and advice.

References

[AHI11] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. In *Innovations in Computer Science - ICS*, pages 45–60, 2011.

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology - CRYPTO*, pages 535–552, 2007.

[BC14] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. *J. Cryptology*, 27(2):317–357, 2014.

[BCFW09] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In *Advances in Cryptology - ASIACRYPT*, pages 524–541, 2009.

[BCKP14] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. In *Advances in Cryptology - CRYPTO*, pages 108–125, 2014.

[BFM15] Christina Brzuska, Pooya Farshim, and Arno Mittelbach. Random-oracle uninstantiability from indistinguishability obfuscation. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 428–455, 2015.

[BFS11] Paul Baecher, Marc Fischlin, and Dominique Schröder. Expedient non-malleability notions for hash functions. In *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, pages 268–283, 2011.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6, 2012. Preliminary version appeared in CRYPTO 2001.

[BGR+15] Hai Brenner, Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. Fast non-malleable commitments. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS*, pages 1048–1057. ACM, 2015.

[BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via uces. In *Advances in Cryptology - CRYPTO*, pages 398–415, 2013.

[BP12] Nir Bitansky and Omer Paneth. Point obfuscation and 3-round zero-knowledge. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC*, pages 190–208, 2012.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security, CCS*, pages 62–73, 1993.

[BS16] Mihir Bellare and Igors Stepanovs. Point-function obfuscation: A framework and generic constructions. In *Theory of Cryptography - 13th International Conference, TCC*, pages 565–594, 2016.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 1997.

[CFP+16] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam D. Smith. Reusable fuzzy extractors for low-entropy distributions. In *Advances in Cryptology - EUROCRYPT*, pages 117–146, 2016.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[CKOS01] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam D. Smith. Efficient and non-interactive non-malleable commitment. In *Advances in Cryptology - EUROCRYPT*, pages 40–59, 2001.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, STOC*, pages 131–140. ACM, 1998.

[CNS07] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In *Advances in Cryptology - EUROCRYPT*, pages 573–590, 2007.

[CQZ+16] Yu Chen, Baodong Qin, Jiang Zhang, Yi Deng, and Sherman S. M. Chow. Non-malleable functions and their applications. In *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part II*, pages 386–416, 2016.

[CV09] Ran Canetti and Mayank Varia. Non-malleable obfuscation. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 73–90. Springer, 2009.

[DDN03] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Review*, 45(4):727–784, 2003.

[FF11] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. *J. Cryptology*, 24(1):203–244, 2011.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT*, pages 445–464, 2006.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 40–49, 2013.

[GJM02] Philippe Golle, Stanislaw Jarecki, and Ilya Mironov. Cryptographic primitives enforcing communication and storage complexity. In *Financial Cryptography, 6th International Conference, FC*, volume 2357, pages 120–135. Springer, 2002.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *FOCS*, pages 102–113, 2003.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *Innovations in Computer Science - ICS 2010*, pages 230–240. Tsinghua University Press, 2010.

[GKS16] Vipul Goyal, Dakshita Khurana, and Amit Sahai. Breaking the three round barrier for non-malleable commitments. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*, pages 21–30. IEEE Computer Society, 2016.

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1128–1141. ACM, 2016.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 564–575. IEEE Computer Society, 2017.

[LP15] Huijia Lin and Rafael Pass. Constant-round nonmalleable commitments from any one-way function. *J. ACM*, 62(1):5:1–5:30, 2015.

[LPS17] Huijia Lin, Rafael Pass, and Pratik Soni. Two-round concurrent non-malleable commitment from time-lock puzzles. *IACR Cryptology ePrint Archive*, 2017:273, 2017.

[Pas16] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. *Computational Complexity*, 25(3):607–666, 2016.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In *Advances in Cryptology - CRYPTO*, pages 57–74, 2008.

[PR08] Rafael Pass and Alon Rosen. Concurrent nonmalleable commitments. *SIAM J. Comput.*, 37(6):1891–1925, 2008.

[PSV06] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In *Advances in Cryptology - CRYPTO*, pages 271–289, 2006.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science, FOCS*, pages 543–553, 1999.

[SBK+17] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. *IACR Cryptology ePrint Archive*, 2017:190, 2017.

[Wee05] Hoeteck Wee. On obfuscating point functions. In *STOC*, pages 523–532. ACM, 2005.

[WG00] David A. Wagner and Ian Goldberg. Proofs of security for the unix password hashing algorithm. In *Advances in Cryptology - ASIACRYPT*, pages 560–572, 2000.

[WY05] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In *Advances in Cryptology - EUROCRYPT*, pages 19–35, 2005.

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In *Advances in Cryptology - CRYPTO*, pages 17–36, 2005.