On Strong Simulation and Composable Point Obfuscation

Nir Bitansky and Ran Canetti

School of Computer Science, Tel Aviv University
{nirbitan,canetti}@tau.ac.il

Abstract. The Virtual Black Box (VBB) property for program obfuscators provides a strong guarantee: Anything computable by an efficient adversary given the obfuscated program can also be computed by an efficient simulator with only oracle access to the program. However, we know how to achieve this notion only for very restricted classes of programs.

This work studies a simple relaxation of VBB: Allow the simulator unbounded computation time, while still allowing only polynomially many queries to the oracle. We then demonstrate the viability of this relaxed notion, which we call Virtual Grey Box (VGB), in the context of fully composable obfuscators for point programs: It is known that, w.r.t. VBB, if such obfuscators exist then there exist multi-bit point obfuscators (aka "digital lockers") and subsequently also very strong variants of encryption that are resilient to various attacks, such as key leakage and key-dependent-messages. However, no composable VBB-obfuscators for point programs have been shown. We show fully composable *VGB*-obfuscators for point programs under a strong variant of the Decision Diffie Hellman assumption. We show they suffice for the above applications and even for extensions to the public key setting as well as for encryption schemes with resistance to certain related key attacks (RKA).

1 Introduction

Informally, an obfuscator is an algorithm which gets as input a program (e.g. a Turing machine or circuit) and outputs a new program which has the same functionality as the original one, but is otherwise "unintelligible". The rigorous study of obfuscation was initiated in the work of [3], who introduced the concept of *virtual black box* security (VBB in short). This concept requires the obfuscated program to behave like a "black box", in the sense that it should not leak any information about the program except its input-output behavior. More precisely, any efficient adversary with access to an obfuscated program can be simulated by an *efficient simulator* with only oracle access to the program. The same work presented the impossibility of *"universal VBB obfuscation"*, showing a family of programs which can not be VBB obfuscated.

In light of this negative result, subsequent work has included several research directions. One line of work extends the result of [3], ruling out obfuscation in various settings [16, 26]. Another line of work is aimed at constructing obfuscators

T. Rabin (Ed.): CRYPTO 2010, LNCS 6223, pp. 520–537, 2010.

for specific program families, which are not ruled out by the universal impossibility result. Here, if we stick to VBB obfuscators, our knowledge is limited essentially to obfuscating point programs and their extensions [8, 11, 23, 14, 26, 9, 13, 12]. A point program $P_v : \mathbb{D}_n \to \{0,1\}$ holds a value $v \in \mathbb{D}_n$ in its code, and accepts its input x iff $x = v$. We only know how to obfuscate point programs in which the point v is explicitly obtainable from the code. Moreover, the known constructions depend on rather strong hardness assumptions, and somewhat inherently so [26].

A third line of work focuses on relaxations of VBB. In this context, [3] suggested the notion of *indistinguishability obfuscators* (INDO) , according to which obfuscations of two related size programs implementing the same functionality should be indistinguishable to any *efficient* adversary. Another relaxation, called *best possible obfuscation* (BPO) [17], requires that any information which the obfuscation leaks is efficiently learnable from any other program with the same functionality and related size (hence "best possible"). These two notions turn out to be equivalent, when restricted to efficient obfuscators.

Both notions are easier to satisfy than VBB. However, the security guarantee they provide is less clear. Unlike VBB, both seem to lose their meaning for a relatively wide range of program classes which are natural candidates for obfuscation. For instance, these notions become meaningless if we allow the obfuscator to work only when the program is given in some *"canonical"* representation in which case, no two programs have the same functionality. Another relaxation requires the obfuscation to be secure only when the program is sampled from some adequate distribution (rather than requiring security for any program in the family). This was done in the context of *perfect one-way hashing* [11] , *point proximity testing* [14], *re-encryption* [22] and more [1, 21, 18]. However, in some scenarios such a relaxation does not capture the security properties we would expect from an obfuscation.

A natural goal is thus to come up with a notion of secure obfuscation that is both meaningful and achievable. Here there is room to consider notions which might be meaningful only for certain program families but not for all.

1.1 This Work

We study a new relaxation of VBB security notion for obfuscators. The requirement is that an obfuscation leaks no information about the program, rather than what can also be learned by an *all-powerful learner* that witnesses only a limited number of input-output pairs (at his choice).

More formally, any *efficient* adversary with access to an obfuscated program can be simulated by an *all-powerful simulator* with poly many oracle queries to the program (in contrast to poly-time simulation which VBB requires). For lack of better name, we call this notion *virtual grey box* (or VGB in short). The extra power given to the simulator is intended to allow it to "reverse engineer" the adversary while avoiding technical difficulties that might be irrelevant to the overall goal. In certain cases (such as "highly unlearnable" programs), this could be done without losing too much of the meaningfulness of the guarantee.

Relationship with existing notions. VGB obfuscation is clearly weaker than VBB obfuscation. In particular, a VGB obfuscation is allowed to leak information which a VBB obfuscation can not. Formally, we show that VGB is strictly weaker, demonstrating a family of programs which can not be VBB obfuscated but is (trivially) VGB obfuscatable. On the other hand, we show that VGB is stronger than the INDO and BPO notions mentioned above. To do so, we observe that even if we further weaken the VGB security requirement by allowing the simulator an unlimited number of oracle queries, it still implies INDO.

For Turing machine obfuscators, the impossibility result of [3] extends to rule out *"universal VGB obfuscation"*. However, we could not rule out universal VGB *circuit* obfuscators (see more details within regarding this difference). We note that [17] show impossibility of strong universal BPO obfuscation that can handle even circuits that use *random oracle* gates. This impossibility applies to the stronger VGB notion.

A setting where VGB is both meaningful and achievable. Like INDO and BPO, VGB is not strong enough for some desirable obfuscation tasks. For example, its weakness might be revealed whenever the obfuscated program computes some kind of cryptographic functionality; indeed, in such cases an all-powerful simulator, even with limited oracle access to the program has a clear advantage over a bounded simulator. In general, it seems that VGB is mostly meaningful for program classes which are *unlearnable* with only poly many queries *even for learners with unbounded computation time.* We demonstrate concrete obfuscation tasks where VGB obfuscation is both meaningful and achievable (under appropriate hardness assumptions) while VBB is not known to be achievable. We hope that this notion will prove instrumental in other obfuscation settings as well.

The main task we consider is that of *composable obfuscation of point programs.* A point program obfuscator is t-composable if having access to t obfuscated point programs, where the values hidden in the programs are related to each other in arbitrary ways, has the "expected effect". In other words, any adversary that has access to the obfuscated programs can be simulated given only oracle access to the programs. Ideally, t could be any polynomial.

In the context of VBB obfuscation, composable point obfuscators were shown to suffice for obfuscating *multi-bit* point programs (MBPP). A MBPP has two hidden values k, m in its code. It returns m on input k, and $\perp$ on any other input. MBPP obfuscators (MBPO's) were in turn shown to imply strong symmetric encryption schemes that are simultaneously secure against weakly random keys (i.e., keys with any super-log entropy) and key dependent messages (KDM) [10]. However, as natural and fruitful as the composability property may seem, none of the known point program obfuscators were shown to be composable (w.r.t. VBB). In particular, existing MBPO's were only shown to be secure for the restricted case that the message m is independent of the key k [9, 10].

We show that, with respect to VGB obfuscation, composable point obfuscators do exist, under appropriate hardness assumptions. Specifically, we show that the point program obfuscator from [8] is VGB-composable for any polynomial number of instances. This is done under a strong variant of the Decision

Diffie Hellman assumption, which naturally extends the assumption used in [8] to demonstrate that this construction yields a VBB point obfuscator.

We then show that VGB composable point obfuscators suffice for constructing MBPO's which are VGB composable on their own. This yields very strong encryption schemes which are resilient to a variety of attacks. This includes the aforementioned KDM and weak keys resilience as well as resistance to certain *related key attacks* (RKA) [2]. The encryption schemes can also be extended to the public key setting (given an extra re-randomiztion property that the [8] obfuscator has). We remark that the result for KDM encryption should be contrasted with the fact that fully KDM-secure encryption schemes can not be proven secure using *fully black box reductions* to *standard cryptographic game* [19]. Our proof of security does not fit this characterization.

1.2 Our Techniques

Proving composability for point obfuscators encounters several difficulties. We sketch these difficulties as well as the ideas and techniques we use to overcome them. We also exhibit how the VGB relaxation comes to our aid.

Simulation and distributional indistinguishability. Ideally, we might try to require that for fixed sequence of points, the resulting obfuscated point programs would appear to an efficient adversary as a sequence of obfuscated *random* point programs (similarly to the *semantic security* requirement for encryption schemes). This would allow simple simulation, by running the adversary on obfuscations of random hidden values. However, in the context of obfuscation such a requirement is unachievable, since the adversary is able to run the program and verify any guesses it might have; in particular it can have some hardwired values which it can always recognize. Instead, we consider a weaker requirement which we call *Distributional Indistinguishability* (DI in short). We show that: (a) DI is necessary and sufficient for constructing VGB simulators, and (b) It is achievable under appropriate hardness assumptions.

DI is an extension of a notion used in [8] in the context of plain point obfuscators. The requirement refers to a specific type of distributions over tuples of points which we call *coordinatewise well spread* (CWS in short). $\mathcal{X} = \left\{(X_n^{(1)} \dots X_n^{(t)})\right\}$ is a CWS distribution ensemble on $\{\mathbb{D}_n^t\}$ if for any $a \in \mathbb{D}_n$ and $i \in [t]$, $X_i \neq a$ except with negligible probability.

Essentially, $\mathcal{O}$ is a *t-DI obfuscator* if for any CWS distribution, $\mathcal{X}$, over t-tuples of elements in $\mathbb{D}_n$, no *efficient* adversary can distinguish obfuscations of t uniform values from obfuscations of a tuple of values sampled from $\mathcal{X}$. We show:

Theorem 1.1 (informal). *If $\mathcal{O}$ is a t-DI point obfuscator then it is a t-composable VGB point obfuscator. Moreover, if $\mathcal{O}$ is t-DI for any polynomial t, then it is a composable VGB obfuscator for any polynomial number of point programs.*

The main technical difficulty in this work is in proving Theorem 1.1. We sketch the ideas used in the proof. Our starting point is a result of [8] showing that for point obfuscators (i.e. $t = 1$) the notions of DI and VBB obfuscation are equivalent and that DI obfuscation is achievable under certain number theoretic assumptions.

First we ask whether t-DI obfuscators imply t-composable VBB obfuscators for $t > 1$. We show that this is the case as long as $t = O(1)$. However, when $t = \omega(1)$, major (and seemingly inherent) difficulties rise. Specifically, recall that when constructing a simulator, we should deal with the fact that the adversary can run the obfuscated programs and might have some hardwired values which it can always recognize. When the adversary has access only to a single obfuscated point program, [8, 26, 10] show that in fact it cannot do much more than have a polynomial number of such hardwired test elements. We call these the *distinguishing elements.* This allows the simulator to check its oracle only on the polynomially many distinguishing elements.

However, in the case of multiple obfuscated points, this plan does not go through. The main difficulty is *adaptivity.* More specifically, while in the case of a single hidden point there is only a single secret, in the case of composable point obfuscators the adversary might first discover only some of the points, and then use this partial information to make his next choices. Fortunately, we can show that for any partial information already learnt there is a corresponding poly set of distinguishing elements. However, there still remains the question of how to compute these elements ahead of time.

We show that the total number of potentially queried elements is $n^{\Theta(t)}$. Here VGB comes to our aid when $t = \omega(1)$. That is, having limited oracle access to the point programs and sufficient power to compute the distinguishing elements allows performing the required simulation.

We remark that a converse statement is also true. That is, DI is necessary for VGB composable obfuscation (and thus also for the stronger VBB notion).

A t-DI point obfuscator. Finally, we reconsider the point program obfuscator constructed in [8]. Under a strong Decision Diffie Hellman (DDH) assumption, we show that this obfuscator is t-DI for any polynomial t and hence is a t-composable VGB obfuscator. As evidence of plausibility, we show that our assumption holds in the *Generic Group Model* [25], where algorithms are only allowed to perform generic group operations and can not exploit the representation of group elements. We note that there exist well studied group ensembles (e.g. Quadratic Residues modulo a prime, and Elliptic Curves groups) where the best cryptanalytic techniques are in fact generic ones [6].

In addition to the above construction, Theorem 1.1 enables construction of composable point obfuscators, based on other hardness assumptions. One natural candidate is the *decisional learning with errors* assumption (LWE) [24] when considered with *weak* (non uniform) secrets. Indeed, under appropriate parameter settings, LWE with weak secrets can be reduced to LWE with uniform secrets [15]. This implies point obfuscators which are secure as long as the secret point is taken from a distribution with some poly-logarithmic entropy.

Organization. Section 2 is devoted to the definition and discussion of VGB obfuscation and its relations with the VBB notion and previous relaxations. Section 3 shows how to construct composable VGB obfuscators for point programs. Section 4 discusses the nature and plausibility of our hardness assumption. Section 5 demonstrates the applications of composable point obfuscators to multi-bit point programs, to set programs, and to strong encryption schemes. Most proofs and some of the secondary results appear in the full version [4].

2 Definitions

We formalize the notion of *virtual black box obfuscation with strong simulators*, and explore its relation to existing notions. In all following definitions, we consider the task of obfuscating an ensemble $\mathcal{C} = \{\mathcal{C}_n\}$, where each $\mathcal{C}_n$ is a collection of circuits with input length n and poly(n) size.

2.1 VBB, IND and BP Obfuscation

We first recall the *virtual black box* definition and two of its previous relaxations.

Definition 2.1 (*obfuscator* [3]). *A PPT $\mathcal{O}$ is a VBB obfuscator for $\mathcal{C}$, if it satisfies:*

- *(*Functionality*) For any $n \in \mathbb{N}$, $C \in \mathcal{C}_n$, $\mathcal{O}(C)$ is a circuit which computes the same function as C.*
- *(*Polynomial Slowdown*) There is a polynomial q such that for any $n \in \mathbb{N}$, $C \in \mathcal{C}_n$, $|\mathcal{O}(C) \leq q(|C|)$.*
- *(*Virtual Black Box*)[1] For any PPT adversary $\mathcal{A}$ and polynomial p there is a PPT simulator $\mathcal{S}$ such that for all sufficiently large $n \in \mathbb{N}$ and $C \in \mathcal{C}_n$:*

$$\left| \Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(C)) = 1] - \Pr_{\mathcal{S}}[\mathcal{S}^C(1^{|C|}) = 1] \right| \leq 1/p(n)$$

Definition 2.2 (*Indistinguishability Obfuscation* [3]). *$\mathcal{O}$ is said to be an* indistinguishability obfuscator *(INDO in short) for $\mathcal{C}$, if it satisfies the* functionality *and* polynomial slowdown *and for any ensemble of circuit pairs $\mathcal{C}^{(1)} \times \mathcal{C}^{(2)} = \{(C_n^{(1)}, C_n^{(2)}) \in \mathcal{C}_n \times \mathcal{C}_n\}$, where the two circuits in each pair are of the same size and functionality, it holds that: $\mathcal{O}(\mathcal{C}^{(1)}) \approx_c \mathcal{O}(\mathcal{C}^{(2)})$.*

Another relaxation of VBB is *Best Possible Obfuscation* (BPO in short) [17]. Here the requirement is that any information which the obfuscation leaks is efficiently learnable from any other circuit with the same functionality and related size (hence it is "best possible"). The two definitions are equivalent when the obfuscator is required to be a PPT [17].

Before presenting our definition we make the following preliminary observation regarding the nature of the above relaxations. The INDO (BPO) definition is

[1] As noted by [3] the above can be replaced with the equivalent requirement that $\left|\Pr[\mathcal{A}(\mathcal{O}(C) = \pi(C)] - \Pr[\mathcal{S}^C(1^{|C|}) = \pi(C)]\right| \leq \frac{1}{p(n)}$ for any predicate $\pi : \mathcal{C}_n \to \{0, 1\}$. Also the size of the simulator can depend on $p(n)$, namely the required simulation quality.

in fact equivalent to a weak black-box definition, which allows an unbounded simulator with unlimited number of oracle queries (proof in [4]).

Proposition 2.1. *$\mathcal{O}$ is an indistinguishability obfuscator for an ensemble of circuits $\mathcal{C} = \{\mathcal{C}_n\}$ iff for any efficient distinguisher $\mathcal{A}$ and polynomial p, there is a (possibly inefficient) simulator $\mathcal{S}$, such that for all large enough n and $C \in \mathcal{C}_n$:*

$$\left| \Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(C)) = 1] - \Pr_{\mathcal{S}}[\mathcal{S}^{C}(1^{|C|}) = 1] \right| \leq 1/p(n)$$

2.2 VGB Obfuscation

The new definition relaxes the VBB security requirement by allowing the simulator to have more computational power. However, we still restrict the number of oracle queries it is allowed to make. The *functionality* and *polynomial slowdown* requirements should be satisfied as in Definition 2.1. The VBB requirement is replaced by the following. Denote by $C[q]$ an oracle to the circuit (function) C which allows at most q queries.

Definition 2.3 (*Virtual Grey Box - obfuscation with a strong simulator*). *A PPT $\mathcal{O}$ has the VGB property if for any PPT adversary $\mathcal{A}$ and polynomial p there is a (possibly* inefficient*) simulator $\mathcal{S}$ and a polynomial q such that for all sufficiently large $n \in \mathbb{N}$ and any $C \in \mathcal{C}_n$:*

$$\left| \Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(C)) = 1] - \Pr_{\mathcal{S}}[\mathcal{S}^{C[q(n)]}(1^{|C|}) = 1] \right| \leq 1/p(n)$$

Remark 2.1. The definitions above concern obfuscators for circuits. That is, both the input program and the output of the obfuscator are given by circuits. One can naturally adjust these definitions to fit the case of *Turing Machine obfuscators* (both input and output are given by a description of a TM). In this work we shall focus on circuit obfuscators (see[4] for corresponding TM definition).

When Is VGB Meaningful? Like INDO and BPO, VGB obfuscation does not seem strong enough for some desirable obfuscation tasks. Examples include: transforming private key encryption schemes to public ones and constructing homomorphic encryption schemes[2]. Informally, the problem in these scenarios is that the obfuscated program computes some kind of cryptographic functionality, which does not remain secure in the presence of unbounded simulators. In general, it seems that VGB is mostly meaningful for program classes which are *unlearnable* with only poly many queries even for learners with unbounded computation time. For families of programs that are not *efficiently learnable*, but are *learnable* for unbounded algorithms with only polynomially many oracle queries, VGB might not guarantee the required security .

[2] See the section on applications in [3] for more details.

2.3 VGB vs. VBB and INDO

VGB is strictly weaker than VBB. The VGB definition is clearly implied by the VBB definition. We show that in fact it is strictly weaker. That is, we show a family which can not be obfuscated according to the VBB definition but is (trivially) obfuscatable under the weaker VGB definition. To do so, we use a slight variation of the family constructed in the [3] impossibility result.

Proposition 2.2. *Assuming the existence of one-way permutations, there exist a family of programs which is not VBB obfuscatable but is VGB obfuscatable.*

To prove the above we use the notion of TM obfuscation (in contrast to circuit obfuscation used in most of this work). The corresponding definitions and proof are given in in [4].

VGB implies INDO (BPO). The relation between VGB obfuscation and the INDO (BPO) follows from Proposition 2.1. That is, even when VGB is further weakened by allowing the (unbounded) simulator unlimited oracle access, it still implies INDO and (for *efficient* obfuscators) BPO.

2.4 Impossibility Results

We consider the possibility of *"universal VGB obfuscation"*. That is, could there exist a VGB obfuscator for the class of all programs? We observe that for TM's obfuscators the impossibility result of [3] extends and also applies for VGB obfuscation. However, for circuits obfuscators the [3] separation no longer holds. Essentially, the reason for this difference is that the *VBB unobfuscatable circuit family* constructed by [3] include cryptographic functionalities (such as *encryption schemes* and *pseudo random functions*) which fail to remain secure in the presence of unbounded simulators (even with limited oracle access). We could not rule out universal VGB obfuscation in the circuit case.

We note that [17] show impossibility of universal BPO obfuscation for circuits which are allowed to use *random oracle* gates. Their result also applies for the stronger VGB notion; however, the meaning of an impossibility result in such setup is somewhat less clear.

3 Composable Point Obfuscators

In this section we define and construct composable VGB point obfuscators. In next sections we show that such obfuscators suffice for meaningful applications.

3.1 Composition of Obfuscators

One central question in the context of obfuscation is the question of *composition*, which asks when and whether is it secure to obfuscate a sequence of programs by obfuscating each program on its own and combining the obfuscated programs. There are several forms of *composition* one could consider, in this work we consider one specific form, namely *composition by concatenation* [23].

Definition 3.1 (*t-composable obfuscation* [**23**])**.** *A PPT $\mathcal{O}$ is a t-*composable obfuscator *for a circuit ensemble $\mathcal{C} = \{\mathcal{C}_n\}$ if it satisfies the* functionality *and* poly slow-down *as in Definition 2.1 and for any PPT binary adversary $\mathcal{A}$ and polynomial p, there is a simulator $\mathcal{S}$, such that for any sequence of circuits $C^1, \dots, C^t \in \mathcal{C}_n$ (where $t = \mathrm{poly}(n)$) and any sufficiently large n:*

$$\left|\Pr[\mathcal{A}(\mathcal{O}(C^1), \dots, \mathcal{O}(C^t)) = 1] - \Pr[\mathcal{S}^{C^1,\dots,C^t}(1^{|C^1|}, \dots, 1^{|C^t|}) = 1]\right| \leq 1/p(n)$$

Where $C^1, \dots, C^t$ gets as input (x, i) and returns $C^i(x)$.

Originally [23] naturally refer to VBB obfuscation, i.e. the simulator $\mathcal{S}$ is a polynomially bounded. We consider the definition also for VGB obfuscators, i.e. we allow the simulator to be unbounded with poly many oracle queries.

3.2 Point Obfuscators

Point circuits. For a security parameter $n \in \mathbb{N}$ and a domain $\mathbb{D}_n$, a point circuit $C_x : \mathbb{D}_n \to \{0, 1\}$ returns 1 on input x and 0 on all other inputs. The point circuits we discuss are given in some "canonical" form where the point x is explicit. As the size of the canonical circuits is determined by the parameter n, we simplify our notation by letting the simulator take input 1^n (instead of the circuit size). The natural choice for the domain is $\mathbb{D}_n = \{0, 1\}^n$. However, to avoid confusion when discussing tuples of points in $\mathbb{D}_n^t$, we shall stick to the more general notation. We refer to obfuscators for point circuits as point obfuscators.

Is any point obfuscator composable? Point obfuscators have been constructed, both in the plain model and in the random oracle model. A natural question is whether any VBB secure point obfuscator is also guaranteed to be composable (as in Definition 3.1). [23] conjectured that the answer is negative. To support their conjecture they give a point obfuscator in the *Random Oracle model* which is not even 2-composable. In the standard model, it can be shown that if point obfuscators exist, then there are also point obfuscators which are not $\Omega(n)$-composable [9]. In general, none of the constructions of point obfuscators were known to be composable.

3.3 Distributional Indistinguishability and Composable Point Obfuscation

To overcome the difficulties in achieving composable point obfuscators, we explore in this section an additional property of point obfuscators, called *Distributional Indistinguishability (or DI in short)*[3]. We will show that this additional property is necessary for composable obfuscation even under the weaker VGB notion. More importantly, we will show that in fact it suffices for VGB obfuscation. The definition we present generalizes the DI definition presented in [8].

[3] DI should not be confused with *Indistinguishability Obfuscators* of [3] which were presented in Definition 2.2

Definition 3.2 (*Coordinatewise Well Spread*). *Let $\mathcal{X} = \{X_n\}$ be an ensemble, where each X_n is a distribution on $\mathbb{D}_n^{t(n)}$ for a domain ensemble $\{\mathbb{D}_n\}$. We say that $\mathcal{X}$ is CWS if: $\max_{a \in \mathbb{D}_n} \Pr_{\bar{x} \leftarrow X_n}[\exists i \in [t] : x_i = a] = n^{-\omega(1)}$.*

That is any element has only a negligible chance of being picked within a vector sampled from the distribution. Equivalently, in a CWS ensemble the distributions $X_n^{(i)}$ all have super-log *min-entropy*, i.e. $\min_{i \in [t]} H_\infty(X_n^{(i)}) = \omega(\log n)$.

Definition 3.3 (*Distributional Indistinguishability*). *$\mathcal{O}$ is t-DI if for any CWS distribution ensemble, $\mathcal{X} = \{X_n = \langle X_n^{(1)}, \ldots, X_n^{(t)} \rangle\}$, it holds that:*

$$\mathcal{O}(C_{\mathcal{X}^{(1)}}), \ldots, \mathcal{O}(C_{\mathcal{X}^{(t)}}) \approx_c \mathcal{O}(C_{\mathcal{U}^{(1)}}), \ldots, \mathcal{O}(C_{\mathcal{U}^{(t)}})$$

Where each $\mathcal{O}(C_{\mathcal{X}^{(i)}})$ is an ensemble of distributions on point obfuscations, and the hidden point is drawn from $\mathcal{X}^{(i)}$ and $\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(t)}$ are ensembles of independent uniform distributions over $\{\mathbb{D}_n\}$.

We note that for the case $t = 1$ Definition 3.3 is equivalent to the DI definition in [8], where it is shown that for $t = 1$, DI and VBB are in fact equivalent. The proof there does not follow through for larger t. Nevertheless, we show:

Theorem 3.1. *Any t-DI point obfuscator is a t-composable VGB obfuscator. Moreover, for $t = O(1)$ it is VBB composable. Conversely, any t-composable VGB point obfuscator is t-DI.*

The proof of the second part of Theorem 3.1 (namely the necessity of DI for composable VGB obfuscation) is rather simple and brought in [4]. We focus on the first part of the theorem that is more involved. A high-level discussion of the proof, techniques and main ideas is given in the introduction. The proof itself is divided to several lemmas. We start with preliminary notations.

Notations. Given a vector of t points $\bar{x} = \langle x_1, \ldots, x_t \rangle$ we abuse notation and denote by $C_{\bar{x}}$ the vector of point circuits $\langle C_{x_1}, \ldots, C_{x_t} \rangle$. We also denote by $\mathcal{O}(C_{\bar{x}})$ the composition $\mathcal{O}(C_{x_1}), \ldots, \mathcal{O}(C_{x_t})$. Speaking of vectors, we shall often be interested in the (unordered) set of their elements. Whenever we use set operators such as $\in, \cap, \cup$ on vectors, it should be interpreted as operating on the corresponding sets. For integers $s \leq t$ we denote by $\binom{[t]}{s}$ the family of subsets of $[t]$ of size s. For vectors $\bar{x}, \bar{z}$ of dimensions s and $t - s$, and a set of indices $I \subseteq [t]$ of size $|I| = s$, we denote by $\textsc{cmb}_I(\bar{x}, \bar{z})$ the t-vector with the elements of $\bar{x}$ in coordinates I and those of $\bar{z}$ in coordinates $[t] - I$ (the mapping is according to ascending order of indices)[4].

As explained in the introduction, we show that for any partial information the adversary learns, there is a relevant polynomial set of *distinguishing elements.* The first lemma deals with the case that no partial information is learnt, and can be viewed as a generalization of a similar claim in [8] to multiple points.

[4] For example $\textsc{cmb}_{\{2,5\}}((a, b), (c, d, e)) = (c, a, d, e, b)$.

Lemma 3.1. *Assume $\mathcal{O}$ is t-DI, then for any binary PPT $\mathcal{A}$ and $p = \mathrm{poly}(n)$ there is a poly-size family $\mathcal{L} = \{L_n \subseteq \mathbb{D}_n\}$ such that any vector $\bar{x} \in \mathbb{D}_n^t$ which does not intersect L_n (i.e. $\bar{x} \subseteq \mathbb{D}_n \setminus L_n$) satisfies:*

$$\left| \Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(C_{\bar{x}})) = 1] - \Pr_{\mathcal{A},\mathcal{O},\bar{u} \overset{U}{\leftarrow} \mathbb{D}_n^t}[\mathcal{A}(\mathcal{O}(C_{\bar{u}})) = 1] \right| \leq 1/p(n) \tag{1}$$

Proof. Consider a binary PPT $\mathcal{A}$ and a polynomial p. We describe the corresponding family $\mathcal{L}$. Let X_n be the set of all vectors which do not satisfy Equation (1). That is, $X_n = X_n^+ \cup X_n^-$, where:

$$X_n^+ = \left\{\bar{x} \in \mathbb{D}_n^t : \Pr[\mathcal{A}(\mathcal{O}(C_{\bar{x}})) = 1] - \Pr[\mathcal{A}(\mathcal{O}(C_{\bar{u}})) = 1] \geq 1/p(n)\right\}$$
$$X_n^- = \left\{\bar{x} \in \mathbb{D}_n^t : \Pr[\mathcal{A}(\mathcal{O}(C_{\bar{u}})) = 1] - \Pr[\mathcal{A}(\mathcal{O}(C_{\bar{x}})) = 1] \geq 1/p(n)\right\}$$

First reduce X_n^+ to a subset of vectors $Y_n^+ \subseteq X_n^+$ in which any element $a \in \mathbb{D}_n$ which appears in some vector $\bar{x} \in X_n^+$ appears in exactly one vector $\bar{y} \in Y_n^+$. Similarly reduce X_n^- to Y_n^-. Let $Y_n = Y_n^+ \cup Y_n^-$ and define $L_n = \bigcup_{\bar{y} \in Y_n} \bar{y} = \{a \in \mathbb{D}_n : \exists \bar{y} \in Y_n, a \in \bar{y}\}$. By the construction of L_n, any $\bar{x} \subseteq \mathbb{D}_n \setminus L_n$ satisfies Equation (1). It remains to show that $|L_n| = \mathrm{poly}(n)$. As $|L_n| \leq t|Y_n|$, it suffices to show that $|Y_n| = \mathrm{poly}(n)$. Assume towards contradiction that the latter does not hold. We shall construct a CWS distribution ensemble $\mathcal{Z} = \{Z_n\}$ over $\mathbb{D}_n^t$, such that $\mathcal{A}$ distinguishes $\mathcal{O}(C_{\mathcal{Z}})$ from $\mathcal{O}(C_{\mathcal{U}(\mathcal{D}^t)})$ with advantage $1/p$ contradicting the DI property. By the assumption on the size of $|L_n|$ there exist a function $\ell(n) = n^{\omega(1)}$ such that for infinitely many n's either $|Y_n^+| \geq \ell(n)$ or $|Y_n^-| \geq \ell(n)$. We assume WLOG the first case holds (the proof is similar for the second). For any $n \in \mathbb{N}$ such that $|L_n| \geq \ell(n)$, set Z_n to be uniform on the set Y_n^+. For other n let Z_n be uniform on some arbitrary set of size $\ell(n)$ in which any element appears in at most one vector (we can take $\ell = o(|\mathbb{D}_n|)$ to assure such a choice is possible). The resulting ensemble $\mathcal{Z}$ is CWS since any single vector is drawn with probability at most $1/\ell$, and any single element appears in at most one vector. Moreover, for any n such that $Z_n \triangleq U(Y_n^+)$, it holds that:

$$\Pr_{\bar{z} \leftarrow Z_n}[\mathcal{A}(\mathcal{O}(C_{\bar{z}})) = 1] - \Pr_{\bar{u} \leftarrow U(\mathbb{D}_n^t)}[\mathcal{A}(\mathcal{O}(C_{\bar{u}})) = 1] \geq$$
$$\min_{\bar{y} \in Y_n^+} \Pr[\mathcal{A}(\mathcal{O}(C_{\bar{y}})) = 1] - \Pr_{\bar{u} \leftarrow U(\mathbb{D}_n^t)}[\mathcal{A}(\mathcal{O}(C_{\bar{u}})) = 1] \geq 1/p(n) \qquad \square$$

While in [8] the above lemma suffices for constructing a simulator, in our setup it does not, since it does not cover the possibility that the adversary successfully learns only some of the points. The next lemma shows that for any partial information learnt by the adversary there is still a corresponding polynomial *distinguishing set.*

Lemma 3.2. *Assume $\mathcal{O}$ is t-DI. Let $s = s(n)$ be any length function such that $s \leq t$ and let $\mathcal{T} = \left\{(\bar{x}_n, I_n) \in \mathbb{D}_n^s \times \binom{[t]}{s}\right\}_{n \in \mathbb{N}}$ be a family of vectors and index* *sets*[5]*. Then for any binary PPT $\mathcal{A}$ and $p = \mathrm{poly}(n)$ there exists a poly-size family $\mathcal{L}^{\mathcal{T}} = \{L_n\}$ such that for any $\bar{y} \in \mathbb{D}_n^{t-s}$ that does not intersect L_n:*

$$|\Pr[\mathcal{A}(\textsc{cmb}_{I_n}(\mathcal{O}(C_{\bar{x}_n}), \mathcal{O}(C_{\bar{y}})) = 1] - \Pr[\mathcal{A}(\textsc{cmb}_{I_n}(\mathcal{O}(C_{\bar{x}_n}), \mathcal{O}(C_{\bar{u}}))) = 1]| \leq \frac{1}{p(n)}$$

Where $\bar{u} \overset{U}{\leftarrow} \mathbb{D}_n^{t-s}$ and the probabilities are over the coins of $\mathcal{A}, \mathcal{O}$ and $\bar{u}$.

To prove the lemma, we shall need the following intuitively correct claim.

Claim. If $\mathcal{O}$ is t-DI then it is also s-DI for any $s \leq t$. (proof in [4]).

Proof (of Lemma 3.2). Consider the function $r = t - s$, then by Claim 3.3 $\mathcal{O}$ is r-DI. Consider an adversary $\mathcal{A}'$ (for r-compositions) which has $\mathcal{T}$ hardwired, and on input $\bar{w}$ (here $\bar{w} = \mathcal{O}(C_{\bar{y}})$ for some $y_1 \ldots y_r$), runs $\mathcal{A}$ on the valid obfuscation $\textsc{cmb}_{I_n}(\mathcal{O}(C_{\bar{x}_n}), \mathcal{O}(C_{\bar{y}}))$. By Lemma 3.2 this $\mathcal{A}'$ has a family $\mathcal{L}^{\mathcal{T}}$ which satisfies the required property with respect to the original adversary $\mathcal{A}$. □

The next lemma shows there is a uniform polynomial bound on the size of all *distinguishing sets* (corresponding to any partial information), and hence there exists a *distinguishing function family*, which given any partial information outputs a poly-size set of all *distinguishing elements* (with respect to this information).

Lemma 3.3. *Let $\mathcal{O}$ be a t-DI obfuscator. Then for any binary PPT $\mathcal{A}$ and $p = \mathrm{poly}(n)$, there exists a family of functions $\mathcal{F} = \{F_n\}$ and a $q = \mathrm{poly}(n)$ such that $F_n : \bigcup_{s \leq t} \left(\mathbb{D}_n^s \times \binom{[t]}{s}\right) \longrightarrow \bigcup_{s \leq q} \binom{\mathbb{D}_n}{s}$ and satisfies for any $(\bar{x}, I) \in \mathbb{D}_n^{|I|} \times \binom{[t]}{|I|}$ and any $\bar{y} \in \mathbb{D}_n^{t-|I|}$ which does not intersect the set $F_n(\bar{x}, I)$:*

$$|\Pr[\mathcal{A}(\textsc{cmb}_{I_n}(\mathcal{O}(C_{\bar{x}}), \mathcal{O}(C_{\bar{y}}))) = 1] - \Pr[\mathcal{A}(\textsc{cmb}_{I_n}(\mathcal{O}(C_{\bar{x}}), \mathcal{O}(C_{\bar{u}}))) = 1]| \leq \frac{1}{p(n)}$$

Where $\bar{u} \overset{U}{\leftarrow} \mathbb{D}_n^{t-|I|}$ and the probabilities are over the coins of $\mathcal{A}, \mathcal{O}$ and $\bar{u}$.

Remark 3.1. The function F_n is defined for any "partial information". In particular the set of indices I is allowed to be the empty set corresponding to no partial information as in Lemma 3.1.

Proof. For any $(\bar{x}, I) \in \mathbb{D}_n^{|I|} \times \binom{[t]}{|I|}$, let $F_n(\bar{x}, I) \subseteq \mathbb{D}_n$ be the minimal set which satisfies the above condition (note that such a set always exists as $\mathbb{D}_n$ trivially satisfies the requirement). We show that, there exists a $q = \mathrm{poly}(n)$, such that $|F_n| \leq q(n)$ (i.e. q is a uniform bound on all images). Let $(\bar{x}_n^*, I_n^*)$ be the pair which maximizes $F_n(\bar{x}, I)$, i.e. $|F_n(\bar{x}_n^*, I_n^*)| = \max_{I \subseteq [t], \bar{x} \in \mathbb{D}_n^{|I|}} |F_n(\bar{x}, I)|$. By Lemma 3.2 there exists a $q = \mathrm{poly}(n)$ for which $|F_n(\bar{x}_n^*, I_n^*)| \leq q(n)$ (just by considering the family $\{(\bar{x}_n^*, I_n^*)\}_{n \in \mathbb{N}}$). The result follows. □

[5] Any pair $(\bar{x}, I)$ should be thought of as partial information on a tuple of size t with the elements of $\bar{x}$ in the indices I.

To complete the proof of the theorem, we construct a simulator using the family of *distinguishing functions* $\mathcal{F}$. However, as it might not be computable by a poly-size simulator, the result holds only for strong simulators as in the VGB definition.

Proof (Any t-DI point obfuscator is also VGB t-composable (sketch)). Let $\mathcal{A}$ be a binary PPT adversary and p a polynomial. Let $\mathcal{F}$ be the corresponding family of functions given by Lemma 3.3 and let q be the polynomial bound on the images of $\mathcal{F}$. We construct an unbounded simulator $\mathcal{S}$ which performs at most $q \cdot t$ oracle queries. Given oracle access to a tuple of circuits $C_{\bar{x}} = C_{x_1}, \ldots, C_{x_t}$, for some $\bar{x} \in \mathbb{D}_n^t$. $\mathcal{S}$ first runs F_n (on the empty set), retrieves a set $L^{(0)}$ of all *distinguishing elements* with respect to no partial information, and queries its oracle on all the elements in $L^{(0)}$. In case it did not reveal any elements (i.e. $\bar{x} \cap L^{(0)} = \emptyset$), it chooses a uniform vector $\bar{u} \overset{U}{\leftarrow} \mathbb{D}_n^t$, computes obfuscations of the points in $\bar{u}$ and runs $\mathcal{A}$ on their composition. Otherwise, it revealed some elements given by a pair $(\bar{z}^{(0)}, I^{(0)})$. It then computes $L^{(1)} = F_n(\bar{z}^{(0)}, I^{(0)})$, and as in the first step, queries all the values in $L^{(1)}$. In case it did not reveal any new values, it chooses a uniform vector $\bar{u} \overset{U}{\leftarrow} \mathbb{D}_n^{t-|I^{(0)}|}$ and runs $\mathcal{A}$ on an obfuscation $\textsc{cmb}_{I^{(0)}}(\mathcal{O}(C_{\bar{z}^{(0)}}), \mathcal{O}(C_{\bar{u}})))$. Otherwise it has updated partial information given by a pair $(\bar{z}^{(1)}, I^{(1)})$. It continues on in this manner. If at any point it revealed all the points in $\bar{x}$ it just runs $\mathcal{A}$ on a random composed obfuscation of the points in $\bar{x}$ performing a perfect simulation. Otherwise, it stops after at most t iterations, performing a simulation of $1/p$ accuracy. This completes the main part of the proof of Theorem 3.1. □

A more careful analysis shows that we can somewhat "compress" the distinguishing function $\mathcal{F}$ to a set of distinguishing elements. This yields the following.

Proposition 3.1. *If $\mathcal{O}$ is a t-DI obfuscator, then any binary adversary given a sequence of t obfuscations can be simulated by a simulator of size $n^{O(t)}$ and polynomially many queries. In particular, for $t = O(1)$ this yields a polynomially bounded simulator (VBB). (proof in [4])*

On the possibility of bounded simulation (VBB). We note that our result does not rule out the possibility of bounded simulation for any $t = \text{poly}(n)$. It might be that there always exists a function family $\mathcal{F}$ such as the one required in Theorem 3.1 which is also efficiently computable, or even a "compressed" poly set of distinguishing elements as in Proposition 3.1. Alternatively, there might be other techniques which allow efficient simulation. In this context, we show an example of an adversary whose distinguishing function can not be compressed to a poly set. We also show that if bounded simulation exists then so does an efficiently computable function family $\mathcal{F}$ (i.e. simulation can be proven using the same technique we use above). The details are given in [4].

Remark 3.2. We note that the applications in Section 5.2 can be shown to hold using the DI obfuscation definition, the equivalence given by Theorem 3.1 allows considering a "simulation" definition that holds for *any input,* and provides a security guarantee even with keys (hidden points) from an arbitrary distribution.

3.4 A Composable Point Obfuscator

After establishing the proper framework in the previous, this section is devoted to a concrete construction for composable VGB point obfuscators. We consider the point obfuscator constructed in [8] and analyze its security under composition.

Construction 3.2 (*The* r, r^x *Point Obfuscator* [8]). *Let $\mathcal{G} = \{\mathbb{G}_n\}_{n\in\mathbb{N}}$ be a group ensemble, where each $\mathbb{G}_n$ is a group of prime order $p_n \in (2^{n-1}, 2^n)$. We define an obfuscator, $\mathcal{O}$, for points in the domain $\mathbb{Z}^*_{p_n}$ as follows: $C_x \overset{\mathcal{O}}{\longmapsto} \mathcal{C}(r, r^x)$ Where $r \overset{U}{\leftarrow} \mathbb{G}^*_n$ is a random generator of $\mathbb{G}_n$, and $\mathcal{C}(r, r^x)$ is a circuit which on input z, checks whether $r^x = r^z$.*

In [8] Construction 3.2 is shown to be secure under a strong variant of the Decision Diffie-Hellman assumption. We now present our assumption which is a generalization of the [8] assumption to tuples of points.

Assumption 3.3 (*t-Strong Vector Decision Diffie Hellman*). *Let $t = \text{poly}(n)$. There exist group ensemble $\mathcal{G} = \{\mathbb{G}_n : |\mathbb{G}_n| = p_n \text{ is prime}\}$ with efficient representation and operations, such that for any CWS distribution ensemble $\mathcal{X} = \{X_n\}$ over vectors in $(\mathbb{Z}^*_{p_n})^t$ the following holds:*

$$\left\{ \begin{matrix} g_1, g_1^{a_1} \\ \vdots \\ g_t, g_t^{a_t} \end{matrix} : \begin{matrix} \bar{g} \overset{U}{\leftarrow} (\mathbb{G}^*_n)^t \\ \bar{a} \overset{X_n}{\leftarrow} (\mathbb{Z}^*_{p_n})^t \end{matrix} \right\}_{n\in\mathbb{N}} \approx_c \left\{ \begin{matrix} g_1, g_1^{u_1} \\ \vdots \\ g_t, g_t^{u_t} \end{matrix} : \begin{matrix} \bar{g} \overset{U}{\leftarrow} (\mathbb{G}^*_n)^t \\ \bar{u} \overset{U}{\leftarrow} (\mathbb{Z}^*_{p_n})^t \end{matrix} \right\}_{n\in\mathbb{N}}$$

We observe that Assumption 3.3 implies that the r, r^x point obfuscator is t-DI with respect to the corresponding group ensemble $\mathcal{G}$, given by the construction. Hence, Theorem 3.1 yields:

Theorem 3.4. *Under Assumption 3.3, the r, r^x point obfuscator is a t-composable VGB point obfuscator (w.r.t the group ensemble $\mathcal{G}$ given by the assumption). Assuming the existence of a "universal" group ensemble which satisfy Assumption 3.3 for any $t = \text{poly}(n)$ implies fully composable VGB obfuscators (i.e. t-composable for any $t = \text{poly}(n)$).*

4 On the Assumption

In this section we discuss Assumption 3.3 and its relation to previous Decision Diffie Hellman variants. We also show that it holds in the *Generic Group Model.*

Relation to Previous DDH Assumptions. We start by presenting another strong variant of DDH for tuples of points, which is in a sense a natural generalization to the standard and strong DDH assumptions [6, 8].

Assumption 4.1 (*t-Strong Vector Decision Diffie Hellman II*). *Let $t = \text{poly}(n)$. There exist group ensemble $\mathcal{G} = \{\mathbb{G}_n : |\mathbb{G}_n| = p_n \text{ is prime}\}$ with* *efficient representation and operations, such that for any CWS distribution ensemble $\mathcal{X} = \{X_n\}$ over vectors in $(\mathbb{Z}^*_{p_n})^t$ the following holds:*

$$\left\{\begin{array}{ccc} g_1, g_1^{a_1}, g_1^{b_1}, g_1^{c_1} & & \bar{g} \xleftarrow{U} (\mathbb{G}^*_n)^t \\ \vdots & : & \bar{a} \xleftarrow{X_n} (\mathbb{Z}^*_{p_n})^t \\ g_t, g_t^{a_t}, g_t^{b_t}, g_t^{c_t} & & \bar{b}, \bar{c} \xleftarrow{U} (\mathbb{Z}^*_{p_n})^t \end{array}\right\}_{n \in \mathbb{N}} \approx_c \left\{\begin{array}{ccc} g_1, g_1^{a_1}, g_1^{b_1}, g_1^{a_1 b_1} & & \bar{g} \xleftarrow{U} (\mathbb{G}^*_n)^t \\ \vdots & : & \bar{a} \xleftarrow{X_n} (\mathbb{Z}^*_{p_n})^t \\ g_t, g_t^{a_t}, g_t^{b_t}, g_t^{a_t b_t} & & \bar{b} \xleftarrow{U} (\mathbb{Z}^*_{p_n})^t \end{array}\right\}_{n \in \mathbb{N}}$$

Restricting the assumption to $t = 1$ results in the strong DDH (SDDH) assumption in [8]. If in addition we restrict $\mathcal{X}$ to be the uniform distribution ensemble, we get the standard DDH assumption. Assumption 4.1 appears as a more familiar and natural generalization of SDDH and DDH than Assumption 3.3 does. However, 3.3 is somewhat simpler and is clearly weaker (the distributions induced by the last two elements of each foursome in 4.1 are identical to those in 3.3). It turns out that the assumptions are in fact equivalent (proof in [4]).

A natural question is whether assumptions 3.3 and 4.1 for $t = 1$ imply the corresponding assumptions for general polynomial t (or even just larger constant t). For the case that the distribution ensemble $\mathcal{X}$ is the uniform distribution this is true (corresponds to showing DDH for any poly number of foursomes from DDH for a single foursome by an hybrid argument). However, when allowing any CWS distribution, such an argument fails to work for two main reasons: (a) dependence among coordinates. (b) the distribution ensemble might not even be efficiently samplable. In general we do not know whether SDDH implies SVDDH.

SVDDH Holds in the Generic Group Model. We show that Assumption 3.3 holds for any $t = \text{poly}(n)$ in the *generic group model* [25] where algorithms can not exploit the representation of the group elements, other than the fact that each element has a unique representation (formal model description and proof in [4]).

5 Applications

In this section, we show how composable VGB point obfuscators, can be used to construct composable VGB obfuscators for MBPC's. Then we discuss how these can be used to obtain strong encryption schemes that are simultaneously resilient to *key dependent messages* (KDM), *leakage* and *related key attacks* (RKA).

5.1 Obfuscation of Point Circuits with Multi-bit Output

A *multibit point circuit* (or MBPC in short), $C_{x \to y} : \mathbb{D}_n \to \{0,1\}^m$, returns y on input x and $\perp$ on all other inputs (once again we assume $C_{x \to y}$ is given in some *canonical* form where x, y are explicit). MBPC obfuscators were constructed by [9] assuming the existence of a composable VBB point obfuscators. However, as explained earlier no known obfuscator has been shown to be composable. We show that applying the [9] construction to composable VGB point obfuscators results in a strong VBB MBPC obfuscator which is also VGB composable. We remark that existing MBPO's were only shown to be secure for the restricted case that the message m is independent of the key k [9, 10]. Moreover, they were not shown to be composable. Both properties are essential for the encryption schemes discussed in the next subsection, in order to get resilience to *key-dependent-messages and related key attacks.*

Construction 5.1 (*Multibit-bit Output Point Obfuscator* [9]). *Let $\mathcal{O}$ be a point obfuscator. Define a PPT $\mathcal{O}^{(m)}$ for point circuits with m-bit output as follows. For a point $x \in \mathbb{D}_n$ and output $y = y_1 y_2 \ldots y_m \in \{0,1\}^m$, choose a random $s \in \mathbb{D}_n - \{x\}$ and define $\bar{a} = \langle a_0, a_1, \ldots, a_m \rangle$ as follows. $a_0 = x$, and for any $i \in [m]$ $a_i = x$ if $y_i = 1$ and $a_i = s$ otherwise. The output of the obfuscator is: $\mathcal{O}^{(m)}(C_{x \to y}) = \mathcal{C}(\mathcal{O}(C_{a_0}), \ldots, \mathcal{O}(C_{a_m}))$. Where $\mathcal{C}$ is a circuit which performs as follows. On input z, it first checks whether $a_0 = z$ (using the first point circuit). If it does not, it returns $\perp$. Otherwise, it finds all other coordinates such that $a_i = z$ and outputs $y_1 \ldots y_m$, where $y_i = 1$ if $a_i = z$ and 0 otherwise.*

Proposition 5.1. *if $\mathcal{O}$ is an $(m+1)$-composable VGB point obfuscator then $\mathcal{O}^{(m)}$ (given by Construction 5.1) is a VBB obfuscator for m-bit point circuits. Moreover, for any decomposition $m + 1 = t \times (m' + 1)$ $\mathcal{O}^{(m')}$ is a VGB t-composable MBPC obfuscator (proof in [4]).*

5.2 Strong Encryption Schemes

As noted in [9], obfuscation of MBPC's implies a very strong type of symmetric encryption (which they call a *digital locker).* This usage was further explored lately by [10] who showed tight relations between MBPC (VBB) obfuscation and the notions of *weak key encryption* and *key dependent messages encryption.* Informally, they show that the existence of MBPC VBB obfuscators imply the existence of strong symmetric encryption schemes which are secure for key dependent messages even with weak random keys. We extend their results by showing that using composable VGB MBPC obfuscators (as the ones described above), similar implications still hold, even for the scenario of multiple messages and keys which are correlated (KDM, RKA). We note that the implications of composable MBPC obfuscation to RKA encryption was not discussed prior to this work. We start by presenting the basic natural transformation between MBPC obfuscators and symmetric encryption schemes.

Construction 5.2 (MBPCO to Symmetric Encryption). *Let $\mathcal{O}$ be an MBPC obfuscator, define (probabilistic) encryption and decryption algorithms: $E_k^{\mathcal{O}}(m) \triangleq \mathcal{O}(C_{k \to m})$ and $D_k^{\mathcal{O}}(C) = C(k)$, where C is interpreted as an MBPC and k is a key taken from a domain of keys $\mathbb{D}_n$ (key sampling is addressed below).*

There are several definitions regarding KDM, RKA and leakage [5, 20, 7, 2]. We use a variant of the definition in [10] extended to the setup of multiple related keys. In this definition, t keys are generated from a distribution $\mathcal{X} = \{X_n\}$ on key vectors in $\mathbb{D}_n^t$ and the adversary witnesses t encryptions of predetermined functions of the keys. Any message might depend on any key, and the keys themselves might also be dependent, according to the joint distribution X_n. The definition considers the case where the distributions X_n are not necessarily uniform but only have certain entropy guarantee.

Definition 5.1 (encryption with multi keys-messages dependence). *An encryption scheme (E, D) is (m, t)-MKM secure if for any CWS distribution ensemble $\mathcal{X} = \{X_n\}$ on key vectors in $\mathbb{D}_n^t$, any PPT $\mathcal{A}$, and (predetermined) functions $f_1, \ldots, f_t : \mathbb{D}_n^t \to \{0,1\}^m$ and all large enough n, the following difference is negligible.*

$$\left| \Pr_{\substack{\bar{k} \leftarrow X_n \\ E, \mathcal{A}}} [\mathcal{A}(E_{k_1}(f_1(\bar{k})), \ldots, E_{k_t}(f_t(\bar{k}))) = 1] - \Pr_{\substack{\bar{k} \overset{U}{\leftarrow} \mathbb{D}_n^t \\ E, \mathcal{A}}} [\mathcal{A}(E_{k_1}(\bar{0}), \ldots, E_{k_t}(\bar{0})) = 1] \right|$$

Where $m(n), t(n)$ are polynomially bounded length functions and $\bar{0} = 0^m$.

Theorem 5.3. *Let $\mathcal{O}$ be a t-composable VGB obfuscator for m-bit point circuits, then the encryption scheme $(E^{\mathcal{O}}, D^{\mathcal{O}})$ is (m, t)-MKM secure (proof in [4]).*

Extension to asymmetric encryption. In case the underlying point obfuscator used in Constructions 5.1,5.2 can be re-randomized, we can in fact get a CPA-secure public key encryption scheme[6] with essentially the same strong properties described above. In particular, one can consider a CPA adaptive definition instead of the one given above. We note that the point obfuscator given by Construction 3.2 is indeed re-randomizable as required.

Other extensions and remarks. We note that the RKA resilience described above does not deal in general with adversaries which adaptively choose the key dependence. However, considering the instantiation of the scheme with the Construction 3.2 obfuscator, one gets RKA security for the family of affine functions of the key even against adaptive adversaries (this follows simply because the construction allows affine homomorphisms of the key). Another remark is that the KDM resilience the scheme is also restricted to a non-adaptive model in which the adversary has to choose in advance the functions of the key which it is interested in, this can be equivalently formulated as an adaptive definition where the family of correlation functions is polynomially bounded, nevertheless this is a meaningful setting which captures common KDM resilience such as the classical *circular dependence.*

References

[1] Adida, B., Wikström, D.: How to shuffle in public. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 555–574. Springer, Heidelberg (2007)

[2] Applebaum, B., Harnik, D., Ishai, Y.: Semantic security under related-key attacks and applications (2010) (manusript)

[3] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)

[4] Bitansky, N., Canetti, R.: On strong simulation and composable point obfuscation (2010), `http://eprint.iacr.org`

[6] Given a secret key $k \in \mathbb{D}_n$ the public key is an obfuscation $\mathcal{O}(C_k)$ and encryption is done as in constructions 5.1,5.2 using the re-randomization properties.

[5] Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer, Heidelberg (2003)

[6] Boneh, D.: The decision diffie-hellman problem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 48–63. Springer, Heidelberg (1998)

[7] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision diffie-hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)

[8] Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997)

[9] Canetti, R., Dakdouk, R.R.: Obfuscating point functions with multibit output. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer, Heidelberg (2008)

[10] Canetti, R., Kalai, Y.T., Varia, M., Wichs, D.: On symmetric encryption and point obfuscation. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 52–71. Springer, Heidelberg (2010)

[11] Canetti, R., Micciancio, D., Reingold, O.: Perfectly one-way probabilistic hash functions (preliminary version). In: STOC, pp. 131–140 (1998)

[12] Canetti, R., Rothblum, G.N., Varia, M.: Obfuscation of hyperplane membership. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 72–89. Springer, Heidelberg (2010)

[13] Canetti, R., Varia, M.: Non-malleable obfuscation. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 73–90. Springer, Heidelberg (2009)

[14] Dodis, Y., Smith, A.: Correcting errors without leaking partial information. In: STOC, pp. 654–663 (2005)

[15] Goldwasser, S., Kalai, Y., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: ICS (2010)

[16] Goldwasser, S., Kalai, Y.T.: On the impossibility of obfuscation with auxiliary input. In: FOCS, pp. 553–562 (2005)

[17] Goldwasser, S., Rothblum, G.N.: On best-possible obfuscation. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 194–213. Springer, Heidelberg (2007)

[18] Hada, S.: Secure obfuscation for encrypted signatures. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 92–112. Springer, Heidelberg (2010)

[19] Haitner, I., Holenstein, T.: On the (im)possibility of key dependent encryption. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 202–219. Springer, Heidelberg (2009)

[20] Halevi, S., Krawczyk, H.: Security under key-dependent inputs. In: ACM Conference on Computer and Communications Security, pp. 466–475 (2007)

[21] Hofheinz, D., Malone-Lee, J., Stam, M.: Obfuscation for cryptographic purposes. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 214–232. Springer, Heidelberg (2007)

[22] Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely obfuscating re-encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 233–252. Springer, Heidelberg (2007)

[23] Lynn, B., Prabhakaran, M., Sahai, A.: Positive results and techniques for obfuscation. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 20–39. Springer, Heidelberg (2004)

[24] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93 (2005)

[25] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)

[26] Wee, H.: On obfuscating point functions. In: STOC, pp. 523–532 (2005)