Obfuscating Simple Functionalities from Knowledge assumptions

Ward Beullens[1] and Hoeteck Wee[2]

[1] imec-COSIC KU Leuven, Belgium
ward.beullens@esat.kuleuven.be
[2] CNRS, ENS and PSL, France
wee@di.ens.fr

Abstract. This paper shows how to obfuscate several simple functionalities from a new Knowledge of OrthogonALity Assumption (KOALA) in cyclic groups which is shown to hold in the Generic Group Model. Specifically, we give simpler and stronger security proofs for obfuscation schemes for point functions, general-output point functions and pattern matching with wildcards. We also revisit the work of Bishop et al. (CRYPTO 2018) on obfuscating the pattern matching with wildcards functionality. We improve upon the construction and the analysis in several ways:

- attacks and stronger guarantees: We show that the construction achieves virtual black-box security for a simulator that runs in time roughly $2^{n/2}$, as well as distributional security for larger classes of distributions. We give attacks that show that our results are tight.
- weaker assumptions: We prove security under KOALA
- better efficiency: We also provide a construction that outputs $n + 1$ instead of $2n$ group elements.

We obtain our results by first obfuscating a simpler "big subset functionality", for which we establish full virtual black-box security; this yields a simpler and more modular analysis for pattern matching. Finally, we extend our distinguishing attacks to a large class of simple linear-in-the-exponent schemes.

1 Introduction

Program obfuscation is a powerful cryptographic primitive where an *obfuscator* $\mathcal{O}$ takes the description of a program as input and outputs an obfuscated program that has the same input-output behavior as the original program while hiding how the program works internally. The first theoretic investigation of obfuscation was made in the work of Barak et al. [1, 14] that defined the Virtual Black Box (VBB) security definition, and showed that this strong definition can not be satisfied for general circuits. This has sparked a line of research, starting from [12], into trying to realize the weaker notion of indistinguishability obfuscation for general circuits. There have been many candidate IO for circuits, but they all rely on non-standard and poorly understood assumptions several of

Security	General Output	self-composable	Assumption	Reference
VBB	✗	✗	Nonstandard DDH	[7]
VBB	✓	✓	ROM	[15]
VBB	✓	✗	Strong OWP	[16]
VBB	✓	✓	Perfect OWF	[8]
VGB	✓	✓	Nonstandard DDH	[4]
VBB	✓	✓	KOALA	This work

Fig. 1. Security for obfuscation of point functions.

which have been broken. In contrast, there is a different line of work to achieve the full VBB obfuscation for more restricted functionalities. Work in this direction has shown that one can VBB obfuscate simple functionalities such as point functions [7, 15, 16, 4, 8] and hyperplane membership testing [9]. Obfuscating the pattern matching with wildcards functionality (also called conjunctions) was shown to be possible from LWE and variants [5, 6, 17, 13]. A pattern is specified by a string ρ in $\{0, 1, \star\}^n$ and matches an input $\mathbf{x} \in \{0, 1\}^n$ if $\rho_i = x_i$ or $\rho_i = \star$ for all $i \in [n]$. Recently at CRYPTO 2018 Bishop et al. presented a simple and efficient method for obfuscating pattern matching with wildcards [3] where the obfuscated pattern compromises of $2n$ elements in a cyclic group, and showed that the construction achieves distributional VBB (DVBB) security for the uniform distribution over patterns containing a fixed number of wildcards up to $0.75n$.

1.1 Our Results

We introduce a knowledge assumption that is weaker than the generic group model. The knowledge assumption is a natural decisional analogue of Damgard's KEA assumption [10], and asserts that given any adversary that distinguishes $g^{\mathbf{Mr}}$ for any $\mathbf{M}$ and a random $\mathbf{r}$ from the uniform distribution, there exists another adversary (sometimes referred to as an "extractor") that outputs a non-trivial vector $\mathbf{z}$ such that $\mathbf{zM} = \mathbf{0}$. We refer to this as the Knowledge of OrthogonALity Assumption (KOALA). The assumption can also be viewed as a natural decisional analogue of the recent algebraic group model [11], which essentially asserts that the only way an adversary can *compute* a new group element is to take a linear combination of previous ones.

To showcase the power of KOALA we give a simple proof for the VBB security of the point function obfuscator of [7]. Moreover, we also give the first proof of the self composability of this obfuscator and we extend the construction to VBB obfuscation of point functions with general output. Prior work on obfuscating point functions is summarized in Fig 1.

We improve on the work of Bishop et al. in a number of directions. First we explain that it is possible to, given an obfuscation of a pattern ρ, learn if the first half of ρ consists of wildcards. Since it is not possible to learn this efficiently through black box access only, this attack shows that the construction is not VBB

Class of Patterns	Distribution	Security	Assumption	Reference
$\{0,1,\star\}^n$, exactly w $\star$'s	uniform, $w \leq 0.75n$	DVBB	generic group	[3]
	uniform, $w \leq n - \omega(\log n)$	DVBB	generic group	[2]
	uniform, $w \leq n - \omega(\log n)$	DVBB	KOALA	Thm. 10
$\{0,1,\star\}^n$	–	$2^{0.5n}$-VBB	KOALA	Thm. 8
	–	**not** $2^{0.499n}$-VBB	–	Thm. 6
	min-entropy $\geq n + \omega(\log n)$	DVBB	KOALA	Thm. 9
	min-entropy $\leq n - \omega(\log n)$	**not** DVBB	–	Lem. 4

Fig. 2. Security for obfuscation of pattern matching with wildcards in cyclic groups [3]. Note that the KOALA knowledge assumption holds in the generic group model. The (independent) work of [2] also proved DVBB results for the class of patterns with exactly w $\star$'s in the high min-entropy setting.

secure. Moreover, the attack shows that there are high entropy distributions for which the scheme is not DVBB secure. On the other hand we prove stronger security claims by proving the scheme to be VBB secure with simulators that run in time roughly $2^{n/2}$. We also give optimal min-entropy bounds such that any distribution that has this amount of min-entropy is automatically DVBB secure. More precisely, we prove that any distribution over $\{0,1,\star\}^n$ with $n + \omega(\log n)$ bits of min-entropy is distributional VBB secure. We give a similar result for distributions with a fixed number of wildcards. Previous works only showed DVBB security for certain specific distributions, namely uniform distributions with a fixed number of wildcards $\leq 3n/4$. These distributions have min-entropy at least $1.06n$ for sufficiently large n and therefore DVBB security for these distributions follows as a special case of our DVBB result for high min-entropy distributions. Another advantage of our security proofs is that they only rely on the KOALA, rather than on the full generic group model.

In our security proof we show that the construction of Bishop et al. is essentially built around an obfuscator for a new *Big Subset*-functionality that could be of independent interest. For input size n, the functions of this functionality are parametrized by a subset $Y \subset [n]$ and a threshold value $0 \leq t \leq n$. The function $f_{Y,n,t} : \mathcal{P}([n]) \to \{0,1\}$ takes a subset $X \subset [n]$ as input and outputs 1 if and only if X is a big enough subset of Y (i.e. $|X| \geq t$). The key result is that the big subset functionality can be obfuscated with VBB security assuming KOALA. The security guarantees for the pattern matching functionality follow from this result by embedding the pattern matching functionality into the big subset functionality.

The scheme of [3] uses only linear operations which are hidden in the exponent of a cryptographic group. We formulate the framework for linear-in-the-exponent obfuscation schemes in the hope of finding more efficient and more secure constructions. On the positive side we find a more efficient construction whose obfuscated programs are represented by $n+1$ group elements rather than $2n$ group elements while having at least the same security as the construction of [3]. On the negative side we prove that our distinguishing attack extends to a wide family of "natural" linear-in-the-exponent obfuscation schemes.

1.2 Technical Overview

We provide a brief overview of our obfuscation construction for the "big subset functionality", which is implicit in [3], and then explain how this relates to obfuscating pattern matching with wildcards.

Obfuscating "big subset". The functionality $f_{Y,n,t}$ is parametrized by (Y, n, t) where $Y \subseteq [n], t \leq n$, and given an input $X \subseteq [n]$,

$$f_{Y,n,t}(X) = 1 \Leftrightarrow |X| \geq t \text{ and } X \subseteq Y\,.$$

The obfuscation of $f_{Y,n,t}$ comprises n group elements $[v_1]_g, \ldots, [v_n]_g$ (we use $[\cdot]_g$ to denote group exponentiation) where

- $\{v_i : i \in Y\}$ are random Shamir shares of 0, that is, the evaluations of a random degree $t-1$ polynomial whose constant term is 0, and
- the remaining v_i's, $i \notin Y$ are chosen uniformly at random.

To evaluate the obfuscated program on input X, we simply return 1 if and only if reconstruction "in the exponent" over the shares corresponding to X returns $[0]_g$.

To prove VBB security, we adopt a "random or learn" strategy similar to that in [7, 9, 16]. Given an adversary $\mathcal{A}$, we try to simulate its view by feeding it n random group elements. Suppose this simulation fails, which means $\mathcal{A}$ distinguishes an obfuscation of $f_{Y,n,t}$ from uniformly random group elements. Then, by our KOALA assumption, then we can "extract" from $\mathcal{A}$ a vector $\mathbf{z}$ from which we can efficiently compute an X such that $f_{Y,n,t}(X) = 1$. In fact, X simply corresponds to the indices of $\mathbf{z}$ that are non-zero; $X \subseteq Y$ follows from the fact that v_i's outside Y are uniformly random, and $|X| \geq t$ follows from the secrecy of Shamir's secret-sharing scheme. Finally, we show that given oracle access to $f_{Y,n,t}$ and an X such that $f_{Y,n,t}(X) = 1$, we can efficiently recover Y, t, upon which we can simulate the view of the adversary perfectly.

We mention here that the actual simulation is a bit more complex, since the KOALA assumption only guarantees extraction with inverse polynomial probability. Therefore, we will need to "extract" multiple $\mathbf{z}$'s and run the above simulation of each of these $\mathbf{z}$; the number of samples we need and thus the running time of the simulator is inverse polynomial in the simulation accuracy. We also note that the same approach also yields a much easier proof for the VBB security of Canetti's point function obfuscator (which outputs just two group elements). Moreover, we can also give a proof for the self-composability of Canetti's obfuscator.

Obfuscating pattern matching with wildcards. To go from obfuscating the "big subset functionality" to obfuscating pattern-matching with wildcards, we observe that there is a simple embedding of $\{0, 1, \star\}^n$ into $(\mathcal{P}([2n]), 2n, n)$ where we replace the i'th symbol with either $2i-1, 2i$ or both. Indeed, this

was the approach (implicitly) taken in [3]. Unfortunately, this embedding also allows an adversary to check whether any subset of $n/2$ positions of a pattern correspond to wildcards, which is the basis for our distinguishing attack. As mentioned earlier in the introduction, we show that

- this construction achieves VBB security with roughly $2^{n/2}$-time simulation. This essentially follows from the fact that we can simulate any query to big subset oracle with $2^{n/2}$ queries to the pattern matching oracle.
- this construction achieves D-VBB security for any distribution over $\{0,1,\star\}^n$ with min-entropy at least $n+\omega(\log n)$. This essentially follows from the fact that any distribution over $(\mathcal{P}([2n]), 2n, n)$ for big subset with min-entropy $n+\omega(\log n)$ is evasive. The latter in turn follows from the fact that any $X \subseteq [2n]$ of size n is an accepting input for at most 2^n patterns in $(\mathcal{P}([2n]), 2n, n)$.
- the construction is not D-VBB secure for some distribution over $\{0,1,\star\}^n$ with min-entropy $n-\omega(\log n)$. In particular, take any $a = \omega(\log n)$ and consider the distribution where the first a positions is uniform over $\{0,1\}^a$, the next a positions are $\star$'s, and the last $n-2a$ positions are uniform over $\{0,\star\}^{n-2a}$. This distribution is evasive, and yet we can distinguish obfuscation of this distribution from that of the uniform distribution over $\{0,1,\star\}^n$.

Prior analysis only considers restricted distributions, namely the uniform distribution over patterns with a fixed number of wildcards; we note that our techniques are fairly general and also provide matching results for these restricted distributions.

In the last section of the paper, we explore the possibility of achieving VBB obfuscation for pattern matching with wildcards via some "natural" generalization of the above constructions. Our results here are mostly negative. Along the way, we also present a compression technique that allows us to reduce the output of the obfuscator from $2n$ to $n+1$ group elements.

Open problems. We conclude with a number of open problems on efficient obfuscation using cyclic groups:

- Construct simple obfuscation schemes for simple functionalities beyond "big subset".
- Prove or disprove: for every $\delta > 0$, there exists an efficient obfuscation scheme for pattern matching with wildcards that is D-VBB for any distribution over $\{0,1,\star\}$ with min-entropy δn (alternatively, VBB secure with $2^{\delta n}$-time simulation).

Roadmap. The rest of the paper consists of the following: In section 2 we state the definitions of VBB, T-VBB and DVBB secure obfuscation schemes. We also prove that T-VBB security implies DVBB security for T-elusive distributions. In section 3 we describe the construction of [3] for obfuscating pattern matching with wildcards. In section 4 we introduce the KOALA knowledge assumption and we prove that it holds in the generic group model. We also showcase the power of

the KOALA by giving a simple proof of the VBB security of the point function obfuscator of Canetti [7], and giving the first proof of its self composability. In section 5 we introduce the big subset functionality, we show that it can be obfuscated with VBB security and we prove that certain distributions of big subset functions are elusive. In section 6 we give our security analysis including a family of attacks and new security proofs. In section 7 we describe and study linear-in-the-exponent obfuscation schemes for pattern matching with wildcards. We find more efficient schemes, but we prove that there are no VBB secure obfuscators in a broad class of constructions that follow this paradigm.

Independent work. We clarify here that an independent work of Bartusek, et al. [2] achieved a subset of our results (in addition to other results not in this work): the overlap are the construction with $n+1$ group elements, as well as distributional VBB for the uniform distribution over patterns with exactly w wildcards for any $w = \omega(\log n)$ in the generic group model.

2 Preliminaries

Notation. Throughout the paper we use $[n]$ to denote the set $\{1, \cdots, n\}$. We write vectors in boldface (e.g. $\mathbf{x}$) and their entries in plain text (e.g. x_1). We also use the implicit representation of group elements: If G is a cyclic group of order p with generator g, then for $a \in \mathbb{Z}_p$ we use $[a]_g$ to denote the group element g^a. If $\mathbf{v} \in \mathbb{Z}_p^n$ is a vector mod p, then $[\mathbf{v}]_g$ denotes the tuple of n group elements $\{g^{v_i}\}_{i\in[n]}$.

2.1 Security Definitions

In this section we define Virtual Black Box [1] (VBB) and Distributional Virtual Black Box [5] (DVBB) security. We also introduce T-VBB security, which is a variant of VBB security where the simulator is allowed to run in super-polynomial time $O(T)$. We prove that T-VBB security implies distributional VBB security for distributions that are T-evasive (even with simulators that make no black box queries).

Let $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ be a sequence of function families where $\mathcal{F}_n$ is a set of functions that takes n bits as input. A PPT algorithm $\mathcal{O}$ is said to be an *Obfuscator* for $\mathcal{F}$ if it takes an input length n (in unary representation) and a function $f \in \mathcal{F}_n$ as input, and outputs an obfuscated program $\mathcal{O}(1^n, f)$ that:

1. preserves functionality: For any n, $f \in \mathcal{F}_n$ and $\mathbf{x} \in \{0,1\}^n$ we have that $\mathcal{O}(1^n, f)(\mathbf{x}) = f(\mathbf{x})$ with a probability that is overwhelming as a parameter of n.
2. has only polynomial slowdown: For any n and $f \in \mathcal{F}_n$ the obfuscated program $\mathcal{O}(1^n, f)$ runs in time that is $poly(n, T(f))$, where $T(f)$ is the run time of f.

To ease notation, we don't explicitly write the input length n as an input to the obfuscator $\mathcal{O}$ in the rest of the paper.

Virtual Black Box security (VBB). If an obfuscator reveals no more information about the function $f \in \mathcal{F}_n$ than what can by learned from black box access the obfuscator is said to be Virtual Black Box (VBB) secure. More formally, we have the following definition

Definition 1 (VBB Security). *An obfuscator $\mathcal{O}$ for the functionality $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is said to be VBB secure if for any PPT Adversary $\mathcal{A}$ and polynomial $p(n)$, there exists a PPT simulator $\mathcal{S}$ that has black box access to a function in $\mathcal{F}$ and an n_0 such that for any $n \geq n_0$ and any $f \in \mathcal{F}_n$*

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

Remark 1. In our definition (and in our definition of T-VBB security below), the simulator $\mathcal{S}$ is allowed to depend on the required simulator accuracy $p(n)$. This is slightly weaker than the original definition of [1].

One can relax the condition that $\mathcal{S}$ runs in polynomial time to obtain a weaker security notion. An obfuscator satisfying this relaxed security notion reveals nothing about the function it obfuscated beyond what can be learned with *a lot* of black box queries.

Definition 2 (T-VBB Security). *An obfuscator $\mathcal{O}$ for the functionality $\mathcal{F}$ is said to be T-VBB secure if for any PPT Adversary $\mathcal{A}$ and any polynomial $p(n)$, there exists a simulator $\mathcal{S}$ that has black box access to a function in $\mathcal{F}$ that runs in time $O(T * \mathrm{poly}(n))$ and an n_0 such that for any $n \geq n_0$ and $f \in \mathcal{F}_n$*

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

Distributional Virtual Black Box security ($\mathcal{D}$-DVBB). A weaker notion of Obfuscator security is that of Distributional VBB security (also called Average-Case VBB). In the distributional setting, there is a sequence of distributions $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ that the function f to be obfuscated is drawn from. If an obfuscator $\mathcal{O}$ reveals nothing about functions randomly drawn from $\mathcal{D}$ beyond what can be learned from black box access, the obfuscator $\mathcal{O}$ is said to be $\mathcal{D}$-DVBB secure. This is captured by the following definition:

Definition 3 ($\mathcal{D}$-DVBB Security). *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a sequence of distributions on $\mathcal{F}$, and $\mathcal{O}$ an obfuscator for the $\mathcal{F}$ functionality. Then $\mathcal{O}$ is said to be $\mathcal{D}$-DVBB secure if for any adversary $\mathcal{A}$ and any sequence of predicates $P = \{P_n : \mathcal{F}_n \to \{0,1\}\}$ there exists a PPT Simulator $\mathcal{S}$ such that*

$$\left| \Pr_{f\leftarrow\mathcal{D}_n,\mathcal{O},\mathcal{A}}[\mathcal{A}(\mathcal{O}(f)) = P_n(f)] - \Pr_{f\leftarrow\mathcal{D}_n,\mathcal{S}}[\mathcal{S}^f(1^n) = P_n(f)] \right| = \mathrm{negl}(n) .$$

The fact that VBB security implies distributional VBB security for any arbitrary distribution is trivial. However, we prove that VBB security also implies

DVBB security with simulators that don't make black-box queries for distributions that are evasive. It is also the case that T-VBB implies DVBB with simulators that make no black-box queries for distributions which are T-evasive.

Definition 4 (evasive, T-evasive). *A sequence $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ of distributions on $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is evasive if there is a negligible function $\mu(n)$ such that for all $\mathbf{x} \in \{0,1\}^n$ we have*

$$\Pr_{f\leftarrow\mathcal{D}_n}[f(\mathbf{x}) \neq 0] < \mu(n)\,.$$

A the sequence of distributions is said to be T-evasive if there is a negligible function $\mu(n)$ such that for all $\mathbf{x} \in \{0,1\}^n$ we have

$$\Pr_{f\leftarrow\mathcal{D}_n}[f(\mathbf{x}) \neq 0] < \frac{\mu(n)}{T(n)}\,.$$

Lemma 1 (VBB implies DVBB without black-box queries for evasive distributions). *Suppose $\mathcal{O}$ is a VBB secure (resp. T-VBB secure) obfuscator for the functionality $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ and let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be an evasive (resp. T-evasive) sequence of distributions that can be sampled from efficiently, then $\mathcal{O}$ is $\mathcal{D}$-DVBB secure with a simulator that does not make any black box queries.*

Proof. Let $\mathcal{O}, \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ and $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be as in the statement of the theorem. Let $\mathcal{A}$ be an adversary and P a predicate on $\mathcal{F}$. We define a simulator $\mathcal{S}$ that draws a function f from $\mathcal{D}_n$ and outputs $\mathcal{A}(\mathcal{O}(f))$. It is clear that this simulator makes no black box queries to f. We now prove that $\mathcal{S}$ has negligible simulation error.

Fix any polynomial $p(n)$ and let $\mathcal{S}_{\mathrm{VBB}}$ be a simulator that runs in polynomial time (resp. $O(T * poly(n))$) with a simulation error that is eventually less than $\frac{1}{3p(n)}$. This is guaranteed to exist because of the VBB (resp T-VBB) property of $\mathcal{O}$. Then we have for large enough n that

$$\left|\Pr_{f\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathcal{O}(f)) = P(f)] - \Pr_{f\leftarrow\mathcal{D}_n}[\mathcal{S}^f_{\mathrm{VBB}}(1^n) = P(f)]\right| \leq$$
$$\sum_{f\in\mathcal{F}_n}\Pr[D_n = f]\left|\Pr[\mathcal{A}(\mathcal{O}(f)) = 1] - \Pr[\mathcal{S}^f_{\mathrm{VBB}}(1^n) = 1]\right| \leq \frac{1}{3p(n)}\,. \quad (1)$$

Since $\mathcal{S}_{\mathrm{VBB}}$ makes at most polynomially many (resp. $O(T * poly(n))$) queries to f and since the sequence $\mathcal{D}$ is evasive (resp. T-evasive) we have

$$\left|\Pr_{f\leftarrow\mathcal{D}_n}[\mathcal{S}^f_{\mathrm{VBB}}(1^n) = P(f)] - \Pr_{f\leftarrow\mathcal{D}_n}[\mathcal{S}^0_{\mathrm{VBB}}(1^n) = P(f)]\right| \leq negl(n)\,, \quad (2)$$

and similarly we have

$$\left|\Pr_{f\leftarrow\mathcal{D}_n}[\mathcal{S}^0_{\mathrm{VBB}}(1^n) = P(f)] - \Pr_{f_1,f_2\leftarrow\mathcal{D}_n}[\mathcal{S}^{f_1}_{\mathrm{VBB}}(1^n) = P(f_2)]\right| \leq negl(n)\,. \quad (3)$$

Finally, similar to Eqn. 1 we have for large enough n that

$$\left| \Pr_{f_1,f_2\leftarrow\mathcal{D}_n}[\mathcal{S}_{\mathrm{VBB}}^{f_1}(1^n) = P(f_2)] - \Pr_{f_1,f_2\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathcal{O}(f_1)) = P(f_2)] \right| \leq$$
$$\sum_{f_1\in\mathcal{F}_n} \Pr[\mathcal{D}_n = f_1] \left| \Pr[\mathcal{S}_{\mathrm{VBB}}^{f_1}(1^n) = 1] - \Pr[\mathcal{A}(\mathcal{O}(f_1)) = 1] \right| \leq \frac{1}{3p(n)} \,. \quad (4)$$

Putting these four inequalities together we have for large enough n that

$$\left| \Pr_{f\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathcal{O}(f)) = P(f)] - \Pr_{f_1,f_2\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathcal{O}(f_1)) = P(f_2)] \right| \leq \frac{2}{3p(n)} + negl(n) \,,$$

which shows that the simulator error is eventually lower than $\frac{1}{p(n)}$ for any $p(n)$. □

3 Obfuscation for Pattern Matching with Wildcards

The class of functions for the pattern matching with wildcards functionality it parametrized by length n strings over the alphabet $\{0, 1, \star\}$. For a pattern $\rho = (\rho_i)_{i\in[n]}$ in $\{0, 1, \star\}^n$ we define the pattern matching function f_ρ that takes a binary string $\mathbf{x} = (x_i)_{i\in[n]}$ as input, and outputs whether the string matches the pattern ρ. More precisely we have

$$f_\rho(\mathbf{x}) = \begin{cases} 1 & \text{if for all } i \text{ either } \rho_i = x_i \text{ or } \rho_i = \star \\ 0 & \text{otherwise} \end{cases}$$

A simple and efficient construction. The work of Bishop et al. [3] gives a simple obfuscation scheme for the pattern matching with wildcards functionality. The obfuscation of a pattern ρ consists of $2n$ elements $\{v_{i,j}\}_{(i,j)\in[n]\times 1,2}$ of a cyclic group G of prime order p with generator g. This obfuscation is produced by picking a random degree $n-1$ polynomial $h(x) = a_1 x + \cdots + a_{n-1}x^{n-1}$ with $h(0) = 0$ and defining

$$v_{i,j} = \begin{cases} h(2i - j) & \text{if } \rho_i = \star \text{ or } \rho_i = j \\ r_{i,j} & \text{otherwise} \end{cases} \,,$$

where the $r_{i,j}$ are chosen uniformly at random. The obfuscation $\mathcal{O}(\rho)$ then consists of the $2n$ group elements $[\{v_{i,j}\}_{(i,j)\in[n]\times\{0,1\}}]_g$.

To evaluate the obfuscated program on input $\mathbf{x}$, the evaluator computes the polynomial interpolation coefficients

$$C_a = \prod_{\substack{b\in[n],\\ b\neq a}} \frac{-2b - x_b}{2a + x_a - 2b - x_b} \,,$$

and computes $h_0 = [\sum_{i\in[n]} C_i v_{i,x_i}]_g$. If the pattern ρ accepts $\mathbf{x}$ then all the v_{i,x_i} are of the form $[h(2i-j)]_g$ and the polynomial interpolation will work in the exponent such that $h_0 = [h(0)]_g = [0]_g$. If $h_0 = [0]_g$ the obfuscated program accepts the input $\mathbf{x}$ and otherwise it rejects. If the pattern ρ does not accept $\mathbf{x}$ at least one uniformly random group element enters into h, so that the obfuscated program will only accept a bad input with probability $1 - \frac{1}{p}$.

Prior analysis in [3]. The construction of [3] is proven to be Distributional VBB secure (Def. 3) in the generic group model for uniform distributions of patterns with a fixed number up to $\frac{3n}{4}$ wildcards. More strongly, it is proven that the result of obfuscating a uniformly random pattern in $\{0,1,\star\}^n$ with a fixed number up to $\frac{3n}{4}$ wildcards is indistinguishable from $2n$ uniformly chosen group elements.

4 A New Knowledge Assumption : KOALA

We introduce a new assumption, the Knowledge of OrthogonALity Assumption (KOALA), that is valid in the generic group model and based on which we will prove the security of the Obfuscation scheme. The assumption says that an adversary can only distinguish $[\mathbf{v}]_g$ for vectors $\mathbf{v}$ drawn uniformly at random from a subspace $V \subset \mathbb{Z}_p^n$ from $[\mathbf{u}]_g$ for uniformly random vectors $\mathbf{u} \in \mathbb{Z}_p^n$ if it can also produce a non-zero vector orthogonal to V in the clear.

Definition 5 (KOALA). *A sequence of cyclic groups $\{G_n\}_{n\in\mathbb{N}}$ of order $p_n \in [2^n, 2^{n+1})$ satisfies the knowledge of orthogonality assumption if for every PPT adversary $\mathcal{A}$, there exists a polynomial $s(n)$ and a PPT algorithm $\mathcal{A}'$ that outputs nonzero vectors such that for every subspace $V \subset \mathbb{Z}_p^n$, if $\mathcal{A}$ distinguishes uniform samples of $[V]_g$ from random with advantage*

$$\mathsf{Adv}_{\mathcal{A},V} = \left| \Pr_{\mathbf{v}\leftarrow V}[\mathcal{A}([\mathbf{v}]_g) = 1] - \Pr_{\mathbf{u}\leftarrow\mathbb{Z}_p^n}[\mathcal{A}([\mathbf{u}]_g) = 1] \right| ,$$

then $\mathcal{A}'(1^n)$ is orthogonal to V with probability

$$\Pr[\mathcal{A}'(1^n) \in V^\perp \setminus \{\mathbf{0}\}] \geq \frac{\mathsf{Adv}_{\mathcal{A},V}}{s(n)} .$$

4.1 KOALA is weaker than Generic Group Model

Although KOALA is quite a strong assumption, it is weaker than the generic group model:

Theorem 1 (Generic groups satisfy KOALA). *A sequence of cyclic groups $\{G_{p_n}\}_{n\in\mathbb{N}}$ of order $p_n \in [2^n, 2^{n+1})$ satisfies KOALA in the generic group model.*

Proof. Given an adversary $\mathcal{A}$, we construct an extractor $\mathcal{E}^{\mathcal{A}}$ that satisfies the condition of Def 5. The extractor runs $\mathcal{A}$ on a list of n generic group elements $\mathbf{e} = \{e_i\}_{i\in[n]}$, then by looking at how $\mathcal{A}$ interacts with the group oracle $\mathcal{E}$ records all the vectors $\mathbf{v}$ for which $\mathcal{A}$ has computed $\mathbf{v}\cdot\mathbf{e}$. When $\mathcal{A}$ terminates, $\mathcal{E}$ chooses two distinct vectors that it has collected and outputs their difference.

More formally the extractor $\mathcal{E}^{\mathcal{A}}$ works as follows: $\mathcal{E}^{\mathcal{A}}$ simulates a group oracle $\mathcal{G}_2$ that gives randomly encoded access to the group $\mathbb{Z}_p^n$. He does this by maintaining a table $\{(\mathbf{q}_i, h_i)\}_{i\in I} \subset \mathbb{Z}_p^n \times \{0,1\}^n$ mapping vectors of $\mathbb{Z}_p^n$ to random handles that he updates on the fly when new vectors are discovered. Initially he populates the table with random handles for the n unit vectors e_i for $i \in [n]$. Then it runs $\mathcal{A}^{\mathcal{G}_2}$ with the handles of the n unit vectors as input. When $\mathcal{A}$ terminates the extractor picks two distinct vectors $\mathbf{q}_i, \mathbf{q}_j$ from the group oracle table and outputs $\mathbf{q}_i - \mathbf{q}_j$.

We now fix a subspace $V \subset \mathbb{Z}_p^n$ with basis $\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$ and we show that if $\mathcal{A}$ makes Q queries to the group oracle and distinguishes $[V]_g$ from $[\mathbb{Z}_p^n]_g$ with probability

$$\mathsf{Adv}_{\mathcal{A},V} = \left| \Pr_{\mathbf{v}\leftarrow V}[\mathcal{A}([\mathbf{v}]_g) = 1] - \Pr_{\mathbf{u}\leftarrow\mathbb{Z}_p^n}[\mathcal{A}([\mathbf{u}]_g) = 1] \right| ,$$

then the extractor will output a nonzero vector orthogonal to V with probability

$$\Pr[\mathcal{A}'(1^n) \in V^\perp \setminus \{\mathbf{0}\}] \geq \frac{\mathsf{Adv}_{\mathcal{A},V}}{(Q+2n)^2} - \frac{2}{p} ,$$

so the extractor satisfies the requirement in Def. 5. We show this through a sequence of four games.

1. In the first game $\mathcal{A}$ is given access to the group oracle $\mathcal{G}_1$ for G_p, and it is given the encoding of $[\mathbf{u}]_g$, for $\mathbf{u}$ a random vector from $\mathbb{Z}_p^n$ as input.

$$\mathsf{Game}_1 = \mathbf{u} \leftarrow \mathbb{Z}_p^n;\ \mathsf{Return}\ \mathcal{A}^{\mathcal{G}_1}(\mathbf{u}));$$

2. In the second game $\mathcal{A}$ is given a group oracle $\mathcal{G}_2$ for the group $\mathbb{Z}_p^n$. Let $\mathbf{e}_i \in \mathbb{Z}_p^n$ for $i \in [n]$ be the unit vectors of $\mathbb{Z}_p^n$. The input to $\mathcal{A}$ is a random encoding of these n unit vectors.

$$\mathsf{Game}_2 = \mathsf{Return}\ \mathcal{A}^{\mathcal{G}_2}(\{\mathbf{e}_i\}_{i\in[n]});$$

3. In the third game $\mathcal{A}$ is given access to a group oracle $\mathcal{G}_3$ for $\mathbb{Z}_p^k$. Let $\mathbf{e}'_i \in \mathbb{Z}_p^k$ for $i \in [k]$ be the unit vectors of $\mathbb{Z}_p^k$. The input to $\mathcal{A}$ is the encoding of n vectors $\{\mathbf{m}_i\}_{i\in[n]}$, where $\mathbf{m}_i = \sum_{j=1}^k (\mathbf{v}_i)_j \mathbf{e}'_j$.

$$\mathsf{Game}_3 = \mathsf{Return}\ \mathcal{A}^{\mathcal{G}_3}(\{\mathbf{m}_i\}_{i\in[n]});$$

4. In the last game $\mathcal{A}$ is given access to the group oracle $\mathcal{G}_1$ for G_p again, and it is given the encoding of $[\mathbf{v}]_g$, for $\mathbf{v}$ a random vector from V as input.

$$\mathsf{Game}_4 = \mathbf{v} \leftarrow V;\ \mathsf{Return}\ \mathcal{A}^{\mathcal{G}_1}(\mathbf{v}));$$

Game_1 **and** Game_2 **are close.** Consider the map $\phi_1 : \mathbb{Z}_p^n \to G_p : \mathbf{x} \mapsto [\mathbf{x} \cdot \mathbf{u}]_g$, where $\mathbf{u}$ is the vector chosen uniformly from $\mathbb{Z}_p^n$ in the first game. Now consider a group oracle $\mathcal{G}_1 \circ \phi_1$ that maintains a table $\{(\mathbf{q}_i, [\mathbf{q}_i \cdot \mathbf{u}]_g, h_i)\}_{i \in I}$ of vectors that were queried, their images under ϕ_1, and random encodings of the images $\phi_1(\mathbf{q}_i)$. As long as no two queries q_i,q_j map to the same element of G_p this is an honest implementation of the group oracle G_2. Moreover, ϕ_1 maps the inputs to $\mathcal{A}$ in Game_2 to the inputs of $\mathcal{A}$ in Game_1, so unless $\mathcal{A}$ queries $\mathcal{G}_2$ at two vectors that are mapped to the same group element by ϕ_1 the views of $\mathcal{A}$ in Game_1 and Game_2 are identical. After Q group oracle queries the table contains $Q + 2n$ entries. For each pair of distinct vectors $(\mathbf{q}_i, \mathbf{q}_j)$ the probability that $\phi_1(\mathbf{q}_i) = \phi_1(\mathbf{q}_j)$ is $1/p$, so a union bound yields

$$|\Pr[\mathsf{Game}_1() = 1] - \Pr[\mathsf{Game}_2() = 1]| < (Q + 2n)^2/p$$

Game_2 **and** Game_3 **are close unless nonzero vectors orthogonal to** V **are found.** Now, consider the map $\phi_2 : \mathbb{Z}_p^n \to \mathbb{Z}_p^k$, defined on the unit vectors as

$$\phi_2(\mathbf{e}_i) = \sum_{j=1}^{k} (\mathbf{v}_i)_j \mathbf{e}'_j \,,$$

and extended to all of $\mathbb{Z}_p^n$ by linearity. Notice that the vectors orthogonal to V are precisely the vectors in the kernel of ϕ_2 because the i-th component of $\phi_2(\mathbf{u})$ is $\mathbf{u} \cdot \mathbf{v}_i$. Now consider the group oracle $G_3 \circ \phi_2$ that maintains the table $\{(\mathbf{q_i}, \phi_2(\mathbf{q}_i), h_i)\}_{i \in I}$. This is an honest implementation of G_2 as long as it is not queried on two different vectors $\mathbf{q}_i, \mathbf{q}_j$ that are mapped to the same vector by ϕ_2. The connecting map ϕ_2 maps the inputs to $\mathcal{A}$ in Game_2 to the inputs of $\mathcal{A}$ in Game_3. Therefore we have

$$|\Pr[\mathsf{Game}_2() = 1] - \Pr[\mathsf{Game}_3() = 1]| \leq \mathsf{Collision} \,,$$

where $\mathsf{Collision}$ is the probability that two vectors in the table of $\mathcal{G}_3 \circ \phi_2$ have the same image under ϕ_2.

Game_3 **and** Game_4 **are close.** The proof of this transition is very similar to that of the first transition, with the connecting map $\phi_3 : \mathbb{Z}_p^k \to G_p : \mathbf{x} \mapsto [\mathbf{x} \cdot \mathbf{c}]_g$, where $\mathbf{c} \in \mathbb{Z}_p^k$ is the unique vector such that $\mathbf{v} = \sum_{i=1}^{k} c_i \mathbf{v}_i$. The map ϕ_3 sends the input of $\mathcal{A}$ in Game_3 to the input of $\mathcal{A}$ in Game_4, so like in the first transition, the view of $\mathcal{A}$ is identical in Game_3 and Game_4 as long as no two queries to $\mathcal{G}_1 \circ \phi_3$ are mapped to the same group element by ϕ. This happens with probability bounded by $(Q + 2n)^2/p$, so we have

$$|\Pr[\mathsf{Game}_3() = 1] - \Pr[\mathsf{Game}_4() = 1]| < (Q + 2n)^2/p$$

Putting everything together. Combining the previous results with the triangle inequality we get

$$|\Pr[\mathsf{Game}_1() = 1] - \Pr[\mathsf{Game}_4() = 1]| < \mathsf{Collision} + 2(Q + 2n)^2/p \,.$$

Here the left hand side is exactly the distinguishing advantage $\mathsf{Adv}_{\mathcal{A},V}$, so we get

$$\mathsf{Collision} > \mathsf{Adv}_{\mathcal{A},V} - \frac{2(Q+2n)^2}{p}\,.$$

The extractor outputs the difference of two randomly chosen vectors out of the $Q+2n$ vectors in the table of $\mathcal{G}_2$. Therefore, since the kernel of ϕ_2 is exactly the set of vectors orthogonal to V we know that $\mathcal{E}$ outputs a vector in $V^\perp \setminus \{\mathbf{0}\}$ with probability at least $\frac{\mathsf{Collision}}{(Q+2n)^2}$, which finishes the proof. □

4.2 Obfuscating Point Functions from KOALA

To demonstrate the power of KOALA, we prove the VBB security of the simple point function obfuscator of [7]. To obfuscate the function that tests whether an input $x \in \mathbf{Z}_p$ is equal to x_0 the obfuscator simply outputs $[r]_g, [-x_0 r]_g$, where $[r]_g$ is a uniformly random group element. On input $x \in \mathbb{Z}_p$, the evaluator simply computes $[xr - x_0 r]$ and outputs 1 if and only if this is equal to $[0]_g$.

Theorem 2 (Obfuscating point functions from KOALA). *The point function obfuscator from [7] using a sequence of groups $\{G_n\}_{n\in\mathbb{N}}$ that satisfies KOALA is VBB secure.*

Proof. Given an adversary $\mathcal{A}$ and required simulator accuracy of $\frac{1}{p(n)}$, let $\mathcal{A}'$ and $s(n)$ be the PPT algorithm and polynomial that are guaranteed to exists because of KOALA. We construct a simulator $\mathcal{S}$ that on input $[\mathbf{v}]_g = ([v_1]_g, [v_2]_g)$ calls $\mathcal{A}'(1^n)$ to get output $\mathbf{o} = (o_1, o_2)$, if $o_2 = 0$ then $\mathcal{S}$ discards $\mathbf{o}$ and otherwise it makes a black box query to the point function on input $\frac{o_1}{o_2}$. The simulator $\mathcal{S}$ repeats this a total of $s(n)p(n)$ times. Then there are two cases:

- **A All of the black box queries return 0.** In this case $\mathcal{S}$ picks a uniformly random vector $\mathbf{u} \in \mathbb{Z}_p^2$ and outputs $\mathcal{A}([\mathbf{u}]_g)$.
- **B A black box query with input x_0 returns 1.** In this case $\mathcal{S}$ outputs $\mathcal{A}(\mathcal{O}(x_0))$.

In case of event **B** the simulation of $\mathcal{S}$ is perfect, so the simulation error of $\mathcal{S}$ is

$$\Pr[\mathbf{A}]\cdot\left|\Pr_{r\leftarrow\mathbf{Z}_p}[\mathcal{A}([r]_g,[xr]_g)=1] - \Pr_{r_1,r_2\leftarrow\mathbb{Z}_p}[\mathcal{A}([r_1]_g,[r_2]_g)=1]\right| = \Pr[\mathbf{A}]\mathsf{Adv}_{\mathcal{A},\langle(1,-x)\rangle}\,.$$

Event **A** only occurs if none of the outputs of $\mathcal{A}'$ are orthogonal to $\langle(1,-x)\rangle$, so using KOALA we get that the simulation error is bounded by

$$\Pr[\mathcal{A}(1^n)^\perp\cdot(1,-x)\neq 0]^{s(n)p(n)}\mathsf{Adv}_{\mathcal{A},\langle(1,-x)\rangle} \leq \left(1-\frac{\mathsf{Adv}_{\mathcal{A},\langle(1,-x)\rangle}}{s(n)}\right)^{s(n)p(n)}\cdot\mathsf{Adv}_{\mathcal{A},\langle(1,-x)\rangle}\,.$$

Using $1-x \leq e^{-x}$ and $e^{-x} \leq \frac{1}{x}$ for $x > 0$ this means that the simulation error is bounded by

$$e^{-\mathsf{Adv}_{\mathcal{A},\langle(1,-x)\rangle}p(n)}\cdot\mathsf{Adv}_{\mathcal{A},\langle(1,-x)\rangle} \leq \frac{1}{p(n)}\,,$$

as required. □

Definition 6 (Array of functions). *Let $f_1, \cdots, f_k : D \to R$ be a sequence of k functions on the same domain D, then we define a new function $[\![f_1, \cdots, f_k]\!] : [k] \times D \to R$ by*

$$[\![f_1, \cdots, f_k]\!](i, x) = f_i(x).$$

Definition 7 (VBB Self composability). *A VBB secure obfuscator $\mathcal{O}$ for a function family $\mathcal{F}$ is said to be VBB self composable if $\mathcal{O}' : (f_1, \cdots, f_k) \in \mathcal{F}^* \to (\mathcal{O}(f_1), \cdots, \mathcal{O}(f_k))$ is a VBB secure obfuscator for the function family*

$$\left\{ [\![f_1, \cdots, f_k]\!] \,|(f_1, \cdots, f_k) \in \mathcal{F}^k \right\}$$

Remark 2. This definition is stronger than the one of [15] because it works simultaneously for all (polynomially bounded) k, rather than a fixed value of k.

Theorem 3. *The point function obfuscator from [7] using a sequence of groups $\{G_n\}_{n\in\mathbb{N}}$ that satisfies KOALA is VBB self composable.*

Proof. Let $\mathcal{A}$ be an adversary and let $p(n)$ be a polynomial such that $\frac{1}{p(n)}$ is the desired simulator accuracy. We then construct a simulator $\mathcal{S}$ that works in two phases. On input $(\mathcal{O}(\mathbf{x}_1), \cdots, \mathcal{O}(\mathbf{x}_k))$ the simulator $\mathcal{S}$ starts with a learning phase $\mathcal{S}$ in which it tries to recover as many of the $\mathbf{x}_i$ as possible. Then in the simulation phase it outputs $\mathcal{A}(u_1, \cdots, u_k)$, where

$$u_i = \begin{cases} \mathcal{O}(x_i) & \text{if } \mathcal{S} \text{ has learned } \mathbf{x}_i \\ [\mathbf{r}_i]_g & \text{if } \mathcal{S} \text{ has not learned } \mathbf{x}_i \end{cases},$$

where the $\mathbf{r}_i$ are uniformly random vectors in $\mathbb{Z}_p^2$.

Learning phase: The learning phase starts with a empty set $L = \{\}$ of learned $\mathbf{x}_i$'s. Let $\mathcal{A}_1'$ and $s(n)$ be the PPT algorithm and the polynomial given by the KOALA assumption. Then, like in the proof of Theorem 2, we call $\mathcal{A}_0()$ a total of $ks(n)p(n)$ times to get an output $\mathbf{o} = ((o_{1,1}, o_{1,2}), \cdots, (o_{k,1}, o_{k,2}))$. For all $i \in [k]$, if $o_{i,2} \neq 0$, then $\mathcal{S}$ queries the black box oracle for f_i at input $\frac{o_{i,1}}{o_{i,2}}$. If all the queries return False, the learning phase ends and $\mathcal{S}$ moves on to the Simulation phase. Conversely, if the query $f_i(\mathbf{x}_i)$ returns True, then $(i, \mathbf{x}_i)$ is added to L.

If the learning phase has not ended in the first iteration (i.e. if $\mathbf{x}_i$ are discovered), then we construct a new adversary $\mathcal{A}_2$ that accepts $k - |L|$ obfuscated programs as input. The adversary $\mathcal{A}_2$ computes an obfuscation $\mathcal{O}(\mathbf{x}_i)$ for each $\mathbf{x}_i$ that it has learned, and plugs it into the slots of $\mathcal{A} = \mathcal{A}_1$. The $k - |L|$ inputs are plugged into the remaining slots and then $\mathcal{A}_2$ calls $\mathcal{A}$ with these inputs and returns the output of $\mathcal{A}$. The KOALA guarantees there exist a PPT algorithm $\mathcal{A}_2'$ and polynomial $s_2(n)$. Then $\mathcal{S}$ calls $\mathcal{A}_2'()$ a total of $ks_2(n)p(n)$ times to get outputs $\mathbf{o} = ((o_{1,1}, o_{1,2}), \cdots, (o_{k,1}, o_{k,2}))$. Again, if $o_{i,2} \neq 0$, then $\mathcal{S}$ queries the black box oracle for f_i at input $\frac{o_{i,1}}{o_{i,2}}$. If all the queries return False, the learning phase ends and $\mathcal{S}$ moves on to the Simulation phase. Conversely, if the query $f_i(\mathbf{x}_i)$ returns True, then $(i, \mathbf{x}_i)$ is added to L.

This process continues with $\mathcal{A}_i + 1$ the algorithm that calculates $\mathcal{O}(\mathbf{x}_i)$ for the newly discovered $\mathbf{x_i}$ and plugging it into $\mathcal{A}_i$. After at most k iterations no new inputs are learned and the Learning phase terminates.

Simulation phase: After the Learning phase the simulator $\mathcal{S}$ computes $u_i = \mathcal{O}(\mathbf{x}_i)$ for all $(i, \mathbf{x}_i)$ in L. Then it fixes the remaining u_i to $[\mathbf{r}_i]_g$ for $\mathbf{r}_i$ random vectors in $\mathbb{Z}_p^2$ and outputs $\mathbf{A}(\mathbf{u}_1, \cdots, \mathbf{u}_k)$. In other words, $\mathcal{S}$ calls the last iteration $\mathcal{A}_i$ of the adversary constructed in the Learning phase on uniformly random input, and return the result.

Now we analyze the simulation error of this simulator $\mathcal{S}$. Let $I = \{i | \exists \mathbf{x}_i s.t. (i, \mathbf{x}_i) \in L\}$ be the set of indices of the $\mathbf{x}_i$ that are learned at the end of the learning phase. Now, for $X \subset [k]$ Let $u_{X,i}$ be the distribution defined as

$$u_{X,i} = \begin{cases} \mathcal{O}(\mathbf{x}_i) & \text{if } i \text{ in X} \\ [U(\mathbb{Z}_p^2)]_g & \text{else} \end{cases}$$

Then we have that the output of $\mathcal{S}$ is equal to $\mathcal{A}(u_{I,1}, \cdots, u_{I,k})$, so the simulation error of $\mathcal{S}$ is bounded by

$$\left| \Pr[\mathcal{A}(\mathcal{O}(f_{\mathbf{x}_1}), \cdots, \mathcal{O}(f_{\mathbf{x}_k})) = 1] - \Pr[\mathcal{S}^{[\![f_{\mathbf{x}_1}, \cdots, f_{\mathbf{x}_k}]\!]}](1^{kn}) = 1 \right| \leq$$

$$\sum_{X \subset [k]} \Pr[X = I] \left| \Pr[\mathcal{A}(\mathcal{O}(f_{\mathbf{x}_1}), \cdots, \mathcal{O}(f_{\mathbf{x}_k})) = 1] - \Pr[\mathcal{A}(u_{X,1}, \cdots, u_{X,k}) = 1] \right| =$$

$$\sum_{X \subset [k]} \Pr[X = I] \cdot \mathsf{Adv}_{\mathcal{A},X} \,,$$

where $\mathsf{Adv}_{\mathcal{A},X}$ denotes the advantage of $\mathcal{A}$ for distinguishing $\mathcal{O}(f_{\mathbf{x}_1}), \cdots, \mathcal{O}(f_{\mathbf{x}_k})$ from $u_{X,1}, \cdots, u_{X,k}$.

The probability $\Pr[X = I]$ is equal to $\Pr[\text{reach } X] \Pr[\text{stay at } X | \text{ reach } X]$, where $\Pr[\text{reach } X]$ is the probability that $\mathcal{S}$ reaches a state where the indices of the learned $\mathbf{x}_i$ is exactly X, and $\Pr[\text{stay at } X | \text{ reach } X]$ is the probability that $\mathcal{S}$ does not leave this state, given that this state is reached. So, $\Pr[\text{stay at } X | \text{ reach } X]$ is bounded by the probability that none of vectors outputted by A_i' is nonzero and orthogonal to the $2(k - |X|)$ dimensional space of obfuscations of the $k - |X|$ point functions that are not learned. According the KOALA this implies

$$\Pr[\text{ stay at } X | \text{ reach } X] \leq \left(1 - \frac{\mathsf{Adv}_{\mathcal{A},X}}{s_i(n)}\right)^{k s_i(n) p(n)} \leq \frac{1}{k \mathsf{Adv}_{\mathcal{A},X} p(n)} \,.$$

Plugging this in to the upper bound for the simulator error shows that it is bounded by

$$\sum_{X \subset [k]} \Pr[\text{reach } X] \cdot \frac{1}{k p(n)} \,.$$

Now, since there are at most k iterations in the learning phase (each new iteration increases $|L|$, and $|L| \leq k$) we know that $\sum_{X \subset [k]} \Pr[\text{reach } X]$ is bounded by k, so the simulation error of $\mathcal{S}$ is bounded by $\frac{1}{p(n)}$, as required. □

Definition 8 (multi-bit output point functions). *Point functions with multi-bit output are parametrized by two bitstrings* $\mathbf{a} \in \{0,1\}^n$ *and* $\mathbf{b} \in \{0,1\}^l$*. The function* $f_{\mathbf{x},\mathbf{y}}$ *is defined as*

$$f_{\mathbf{a},\mathbf{b}}(\mathbf{x}) = \begin{cases} \mathbf{b} & \text{if } \mathbf{x} = \mathbf{a} \\ \perp & \text{else} \end{cases}$$

Theorem 4 (Obfuscating multi-bit output point functions). *Suppose* $\mathcal{O}$ *is a VBB self-composable obfuscator for point functions, then there exists a VB self-composable obfuscator* $\mathcal{O}'$ *for point functions with multi-bit output*

Proof. On input $(\mathbf{a}, \mathbf{b}) \in \{0,1\}^n \times \{0,1\}^l$ the obfuscator $\mathcal{O}'$ simply computes and outputs l obfuscated programs $\mathcal{O}(\mathbf{a}||b_1), \cdots, \mathcal{O}(\mathbf{a}||b_l)$. To evaluate $\mathcal{O}'(\mathbf{a}, \mathbf{b})$ at input $\mathbf{x}$, one simply evaluates all the obfuscations at $\mathbf{x}||0$ and $\mathbf{x}||1$. If for some of the obfuscations neither $\mathbf{x}||0$ nor $\mathbf{x}||1$ is accepted, then the evaluator returns $\perp$, otherwise the evaluator returns $\mathbf{y}$ defined as

$$y_i = \begin{cases} 0 & \text{if } i\text{-th obfuscated program accepts } \mathbf{x}||0 \\ 1 & \text{if } i\text{-th obfuscated program accepts } \mathbf{x}||1 \end{cases}$$

Correctness and poly-time slowdown of this obfuscator $\mathcal{O}'$ follows immediately from the correctness and poly-time slowdown of $\mathcal{O}$.

Now we show that the construction is VBB secure for compositions of k multi-bit output point functions. Let $\mathcal{A}$ be an adversary and let $\frac{1}{p(n)}$ be the desired simulator accuracy. Then the VBB self-composability property of $\mathcal{O}$ immediately implies there is a PPT simulator $\mathcal{S}$ with the desired simulator accuracy that makes black box queries to the $k \times l$ point functions $f_{\mathbf{a_1}||b_{11}}, \cdots, f_{\mathbf{a_1}||b_{1l}}, \cdots, f_{\mathbf{a_k}||b_{k1}}, \cdots, f_{\mathbf{a_k}||b_{kl}}$. We can answer these queries because we have black box access to $f_{\mathbf{a}_1||\mathbf{b}_1}, \cdots, f_{\mathbf{a}_k||\mathbf{b}_k}$. To answer a query to $f_{\mathbf{a}_i||b_{ij}}$ with input $\mathbf{x}, b$ we first query the black box oracle for $f_{\mathbf{a}_i||\mathbf{b}_i}(\mathbf{x})$. If this returns $\perp$, we answer the query with False, otherwise if $f_{\mathbf{a}_i||\mathbf{b}_i}(\mathbf{x}) = \mathbf{b}_i \in \{0,1\}^l$, then we answer the query with $b_{ij} = b$. □

5 Obfuscating Big Subset Functionality

The obfuscator for pattern matching with wildcards of [3] contains an obfuscator for a different functionality, we call this other functionality the *big subset* functionality. We show that there is an embedding of the pattern matching with wildcards functionality into the big subset functionality and hence, that any obfuscator for the big subset functionality can be transformed generically into an obfuscator for pattern matching with wildcards. This transformation preserves VBB security at the cost of a slowdown of the simulator by a factor $2^{n/2}$. The transformation also preserves distributional VBB security with simulators that make no black box queries without slowing down the simulator. Since the obfuscator of [3] is an instantiation of this transformation this will ultimately allow us to prove its VBB security with super-polynomial simulator and Distributional VBB security for a wide variety of distributions.

Definition 9 (Big Subset Functionality). *For each $n \in \mathbb{N}$, we define the class of functions parametrized by (Y, n, t), where Y is a subset of $[n]$ and t is a threshold value with $0 \leq t \leq n$. We define $f_{Y,n,t} : P([n]) \to \{0,1\}$ that on input a subset X outputs*

$$f_{Y,n,t}(X) = \begin{cases} 1 & \text{if } |X| \geq t \text{ and } X \subset Y \\ 0 & \text{otherwise} \end{cases} .$$

5.1 VBB Secure Obfuscation of Big Subset Functionality

The following construction is implicit in [3]: To obfuscate the function $f_{Y,n,t}$ the obfuscator picks a random degree $t-1$ polynomial $h(x) = a_1 x + \cdots + a_{t-1}x^{t-1}$ with coefficients in $\mathbb{Z}_p$ such that $h(0) = 0$. Then it outputs n group elements $[\mathbf{v}]_g$ defined as

$$v_i = \begin{cases} h(i) \text{ if } i \in Y \\ r_i \text{ otherwise} \end{cases} ,$$

where the $r_i \in \mathbb{Z}_p$ are chosen uniformly at random. To evaluate the function at input $X \subset [n]$ we use polynomial interpolation in the exponent to check if the points $\{(i, o_i) \,|\, i \in X\}$ lie on a degree $|X|-1$ polynomial h_x with $h_x(0) = 0$.

We now prove that under KOALA this construction is a VBB secure obfuscator.

Theorem 5 ($\mathcal{O}$ is VBB secure). *Let $\mathcal{O}$ be the obfuscator for the big subset functionality defined above, using a family of cyclic groups that satisfies KOALA. Then $\mathcal{O}$ is VBB secure.*

Proof. Let $\mathcal{A}$ be an adversary and $p(n)$ the polynomial such that $\frac{1}{p(n)}$ is the desired simulator accuracy. Then we construct a simulator $\mathcal{S}$ that runs in time $O(p(n) * poly(n))$ such that for any (Y, n, t) with sufficiently large n the simulation error

$$\left| \Pr_{\mathcal{O},\mathcal{A}}[\mathcal{A}(\mathcal{O}(Y,n,t)) = 1] - \Pr_{\mathcal{S}}[\mathcal{S}^{f_{Y,n,t}}(1^n) = 1] \right| \leq \frac{1}{p(n)} .$$

The simulator $\mathcal{S}$ is constructed as follows: According to KOALA, there exists a PPT algorithm $\mathcal{A}'$ that samples vectors in $\mathbb{Z}_p^n$ that are likely to be orthogonal to any subspace V such that $\mathcal{A}$ can distinguish $[\mathbf{v}]_g \leftarrow [V]_g$ from $[\mathbf{u}]_g \leftarrow [\mathbb{Z}]_g$. Now $\mathcal{S}$ repeatedly calls $\mathbf{x} \leftarrow \mathcal{A}'$ and queries the $f_{Y,n,t}$ oracle on $\mathrm{Sup}(\mathbf{x})$ for a total of $R(n)$ times (for R some polynomial to be determined later). Now there are two possibilities:

A. All of the $f_{Y,n,t}$ queries return 0. In this case $\mathcal{S}$ just picks a uniformly random vector $\mathbf{u} \in \mathbb{Z}_p^n$ and outputs $\mathcal{A}([\mathbf{u}]_g)$.

B. One of the queries $f_{Y,n,t}(X)$ returns 1. In this case $\mathcal{S}$ makes $n - |X|$ additional queries to $f_{Y,n,t}$ on the inputs $X \cup \{i\}$ for $i \notin X$ in order to learn the set Y. Once $\mathcal{S}$ knows Y it queries $f_{Y,n,t}$ on subsets of Y of increasing size until it gets an accept in order to learn the threshold value t. Then $\mathcal{S}$ outputs $\mathcal{A}(\mathcal{O}(Y,n,t))$.

The intuition to why this simulator works is that either $\mathcal{A}$ can distinguish $\mathcal{O}(Y,n,t)$ from randomness, in which case we can show that $\mathbf{B}$ occurs with overwhelming probability, or $\mathcal{A}$ cannot distinguish $\mathcal{O}(Y,n,t)$ from $[\mathbf{u}]_g$ in which case the event $\mathbf{A}$ can happen with non-negligible probability, but this is not a problem because then $\mathcal{S}$ outputs $\mathcal{A}([\mathbf{u}]_g)$ which is close enough to $\mathcal{A}(\mathcal{O}(Y,n,t))$.

Let $s(n)$ be the polynomial from the KOALA assumption (Def. 5) such that for any subspace $V \subset \mathbb{Z}_p^n$

$$\Pr[\mathcal{A}'(1^n) \in V^\perp \setminus \{\mathbf{0}\}] \geq \frac{\mathsf{Adv}_{\mathcal{A},V}}{s(n)} .$$

The remainder of the proof shows that the simulation error of $\mathcal{S}$ is bounded by $\frac{s(n)}{R(n)}$, so by taking $R(n) = s(n)p(n)$, we get that the simulation error of $\mathcal{S}$ is less than $\frac{1}{p(n)}$, as required.

Let $V_{Y,n,t}$ be the set of exponent vectors of possible obfuscations of $f_{Y,n,t}$. This is a vector space that can be written as $V_{Y,n,t} = C + E$, where C is

$$C = \left\{ \{h(i)\}_{i\in[n]} \,|h \text{ a degree } t-1 \text{ polynomial with } h(0) = 0 \right\} ,$$

and E is the subspace with basis $\{\mathbf{e}_i \,|\, i \notin Y\}$. C is the column space of the (almost Vandermonde) n-by-$(t-1)$ matrix

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 2 & 2^2 & \cdots & n^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ n & n^2 & \cdots & n^{t-1} \end{pmatrix}$$

Any $(t-1)$-by-$(t-1)$ submatrix of this matrix is invertible, which means that elements in $C^\perp$ are either 0 or have more than $(t-1)$ nonzero entries. So for any $\mathbf{x} \in (V_{Y,n,t}^\perp \setminus \{\mathbf{0}\}) \subset (C^\perp \setminus \{0\})$ we have $|\operatorname{Sup}(\mathbf{x})| \geq t$. Also, $\mathbf{x} \in E^\perp$, which implies that $\operatorname{Sup}(\mathbf{x}) \subset Y$. Therefore, $x \in V_{Y,n,t}^\perp \setminus \{\mathbf{0}\}$ implies $f_{Y,n,t}(\operatorname{Sup}(\mathbf{x})) = 1$.

So the event $\mathbf{A}$ that the support of none of the vectors sampled by $\mathcal{A}'$ is accepted by the $f_{Y,n,t}$ oracle is less probable than the event that none of the vectors sampled by $\mathcal{A}'$ is orthogonal to $V_{Y,n,t}$. Because of KOALA this means

$$\Pr[\mathbf{A}] \leq \left(1 - \Pr[\mathcal{A}'(1^n) \in V^\perp \setminus \{\mathbf{0}\}]\right)^{R(n)} \leq \left(1 - \frac{\mathsf{Adv}_{\mathcal{A},V}}{s(n)}\right)^{R(n)} . \qquad (5)$$

The simulator returns the output $\mathcal{A}$ on random input or on input $\mathcal{O}(Y,n,t)$ in case of event $\mathbf{A}$ or event $\mathbf{B}$ respectively, so

$$\Pr[\mathcal{S}^{f_{Y,n,t}}(1^n) = 1] = \Pr[\mathbf{A}] \cdot \Pr_{\mathbf{u}\leftarrow\mathbb{Z}_p^n}[\mathcal{A}([\mathbf{u}]_g) = 1] + \Pr[\mathbf{B}] \cdot \Pr[\mathcal{A}(\mathcal{O}(Y,n,t)) = 1] ,$$

so the simulation error of $\mathcal{S}$ is equal to

$$\Pr[\mathbf{A}] \cdot \left| \Pr[\mathcal{A}(\mathcal{O}(V,n,t)) = 1] - \Pr_{\mathbf{u}\leftarrow\mathbb{Z}_p^n}[\mathcal{A}([\mathbf{u}]_g) = 1] \right| = \Pr[\mathbf{A}] \cdot \mathsf{Adv}_{\mathcal{A},V_{Y,n,t}} .$$

Combining this with Eqn. 5 says that the simulation error of $\mathcal{S}$ is at most

$$\left(1 - \frac{\mathsf{Adv}_{\mathcal{A},V_{Y,n,t}}}{s(n)}\right)^{R(n)} \cdot \mathsf{Adv}_{\mathcal{A},V_{Y,n,t}} \leq \exp\left[\frac{\mathsf{Adv}_{\mathcal{A},V_{Y,n,t}} R(n)}{s(n)}\right] \mathsf{Adv}_{\mathcal{A},V_{Y,n,t}} \leq \frac{s(n)}{R(n)}\,,$$

where for the first inequality we use $1 - x \leq \exp(-x)$, and for the second inequality we use $\exp(-x) \leq \frac{1}{x}$ for $x > 0$. □

5.2 Evasive Distributions

We describe several evasive distributions for the big subset functionality, which will come in handy later for analyzing pattern matching with wildcards.

Lemma 2 (Evasive distributions for big subset). *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a sequence of distributions and $t_0(n), t_1(n)$ functions with $0 \leq t_0(n) \leq t_1(n) \leq n$. Then we have*

1. *If $\mathcal{D}_n$ outputs (Y, n, t) with $t \geq t_0$, and the min-entropy of $\mathcal{D}_n$ is $n - t_o(n) + \omega(\log n)$, then $\mathcal{D}$ is evasive.*
2. *If $\mathcal{D}_n$ outputs $(Y, n, t_0(n))$ with $|Y| = t_1(n)$, and the min-entropy of $\mathcal{D}_n$ is $\log(\binom{n-t_0(n)}{t_1(n)-t_0(n)}) + \omega(\log n)$, then $\mathcal{D}$ is $\mathcal{D}$ evasive.*

Proof. Suppose $\mathcal{D}$ and t_0 satisfy the assumptions of 1 and let m_n be the min-entropy of $\mathcal{D}_n$. Take any n and $X \subset [n]$. Now we prove that

$$\Pr_{(Y,n,t)\leftarrow\mathcal{D}_n}[f_{Y,n,t}(X) = 1] \leq (n - t_0(n))2^{n-t_0(n)-m_n}\,.$$

If $|X| \leq t_0(n)$, then clearly this probability is zero, because we have $|X| \leq t$ with probability 1. So suppose $|X| \geq t_0(n)$. Then there are at most $2^{n-t_0(n)}$ values of Y such that $X \subset Y$ and at most $(n - t_0(n))$ values of t such that $|X| \leq t$. This makes a total of $(n-t_0(n))2^{n-t_0(n)}$ triples (Y, n, t) such that $f_{Y,n,t}(X) = 1$. Since each of these triples occurs with probability at most 2^{-m_n} the inequality above follows.

This shows that if the min-entropy of $\mathcal{D}_n$ is $n - t_o(n) + \omega(\log n)$, then

$$\Pr_{(Y,n,t)\leftarrow\mathcal{D}_n}[f_{Y,n,t}(X) = 1] \leq (n - t_0(n))2^{-\omega(\log n)}\,,$$

which is a negligible function of n, so $\mathcal{D}$ is evasive.

The argument to prove 2 is very similar, the only difference being that $t = t_o(n)$ and $|Y| = t_1(n)$ reduces the number of triples (Y, n, t) such that $f_{Y,n,t}(X) = 1$. If $|X| < t_0(n)$, then there are no accepting triples. If $|X| \geq t_0$ the number of Y of size $t_1(n)$ such that $X \subset Y$ is $\binom{n-t_0(n)}{t_1(n)-t_0(n)}$, so

$$\Pr_{(Y,n,t)\leftarrow\mathcal{D}_n}[f_{Y,n,t}(X) = 1] \leq \binom{n - t_0(n)}{t_1(n) - t_0(n)} 2^{-m_n}\,.$$

The rest of the argument is the same as in the proof of 1. □

6 Obfuscating Pattern Matching with Wildcards, Revisited

In this section, we further investigate the security of the obfuscation scheme of [3]. On the negative side we introduce an attack that allows an adversary to learn if the first half of the pattern consists of wildcards. This proves the scheme is not VBB secure, and even that the scheme is not DVBB secure for some high entropy distributions. On the positive side however, we prove that the scheme is VBB secure if we allow for a super-polynomial simulator.

We also show that any distribution of patterns that has at least $n+\omega(\log n)$ bits of min-entropy is automatically secure. We give similar bounds for distributions that output patterns with a fixed number of wildcards. Our attacks match these min-entropy bounds and hence they show that the bounds are nearly optimal. The bounds immediately prove that the scheme is DVBB secure for uniform patterns and uniform patterns with a fixed number of wildcards up to $n-\omega(\log n)$. This is stronger than the result of [3] that only proves DVBB security for uniform distributions of up to $\frac{3n}{4}$ wildcards. Having up to $n-\omega(\log n)$ wildcards is optimal, because for $n-O(\log n)$ wildcards a pattern can be recovered through black box queries in polynomial time and VBB security is trivial. Indeed, if there are only $O(\log n)$ non wildcards, then after polynomially many black box queries at random inputs we will get an accepting input. Once an accepting input $\mathbf{x} \in \{0,1\}^n$ is found we can learn the entire pattern with n additional black box queries on the n inputs that differ from $\mathbf{x}$ at exactly one position.

6.1 The Construction of [3] is not VBB Secure

By looking at an obfuscation of a pattern ρ it is possible to check whether the first half consists of wildcards. This is done by simply doing polynomial interpolation in the exponent in the values $v_{i,j}$ for $(i,j) \in [\lceil n/2 \rceil] \times \{0,1\}$. Determining whether the first half of a pattern consists of wildcards is not efficiently possible with only black box access, so this attack breaks VBB security. Moreover, this breaks DVBB security for high entropy distributions.

Let $[\mathbf{v}]_g = [\{v_{i,j}\}_{(i,j)\in[n]\times\{0,1\}}]_g$ be the obfuscation of a pattern ρ. To simplify the notation we assume that n is even. The $[v_{i,j}]_g$ are of the form $[p(2i-j)]_g$ for all $(i,j) \in [n/2] \times \{0,1\}$ if and only if the first half of the pattern ρ consist of wildcards. Therefore we can compute the polynomial interpolation coefficients

$$C_{i,j} = \prod_{\substack{(a,b)\in[n/2]\times\{0,1\},\\ (a,b)\neq(i,j)}} \frac{-2a+b}{2i-j-2a+b}$$

and then $h = [\sum_{(i,j)\in[n/2]\times\{0,1\}} C_{i,j} v_{i,j}]_g$ will be equal to $[p(0)]_g = [0]_g$ if the first half of ρ consist of wildcards. If the first half does not consist of wildcards, then a random group element enters in the calculation of h and then $h \neq [0]_g$ with overwhelming probability $1 - 1/p$.

Lemma 3 (A very evasive insecure distribution). *There exists a sequence of distributions $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ that is $2^{n/2}n^{-\omega(1)}$-evasive such that the obfuscation scheme of [3] is not $\mathcal{D}$-DVBB secure.*

Proof. Let $\mathcal{D}_n$ be the distribution that tosses a fair coin and on tails outputs a uniformly random pattern without wildcards and on heads outputs a uniformly random pattern with wildcards in the first half but no wildcards in the second half. Clearly for any $\mathbf{x}$ the probability

$$\Pr_{\rho\leftarrow D_n}[f_\rho(\mathbf{x})=1] < 2^{-n/2},$$

so this sequence of distributions is $2^{n/2}n^{-\omega(1)}$-evasive. Let $\mathcal{A}$ the adversary that executes the attack of the previous paragraph and outputs 1 if $h=[0]_g$ and 0 otherwise. Let P be the predicate of the first half of a pattern being wildcards and let $\mathcal{S}$ be a PPT simulator. Since our distribution is evasive we have

$$\left|\Pr_{\rho\leftarrow\mathcal{D}_n,\mathcal{S}}[S^{f_\rho}(1^n)=P(\rho)] - \Pr_{\rho\leftarrow\mathcal{D}_n,\mathcal{S}}[S^0(1^n)=P(\rho)]\right| = negl(n).$$

Now we can bound the simulation error of $\mathcal{S}$:

$$\left|\Pr_{\rho\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathcal{O}(\rho))=P(\rho)] - \Pr_{\rho\leftarrow\mathcal{D}_n,\mathcal{S}}[S^{f_\rho}(1^n)=P(\rho)]\right| \leq$$

$$\left|\Pr_{\rho\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathcal{O}(\rho))=P(\rho)] - \Pr_{\rho\leftarrow\mathcal{D}_n,\mathcal{S}}[S^0(1^n)=P(\rho)]\right| - negl(n) =$$

$$\left|\left(1-\frac{1}{2p}\right)-\frac{1}{2}\right| - negl(n).$$

which is clearly not negligible. This proves the obfuscation scheme is not distributional VBB secure for this scheme. □

Theorem 6 ($\mathcal{O}$ is not $2^{0.5n}n^{-\omega(1)}$-VBB secure.). *Let $\mathcal{O}$ be the obfuscation scheme for pattern matching with wildcards from [3], then $\mathcal{O}$ is not $2^{0.5n}n^{-\omega(1)}$-VBB secure.*

Proof. This follows immediately from Lemma 1 combined with Lemma 3. □

The distribution of Lemma 3 has $n/2+1$ bits of min-entropy, but the attack can be generalized to showcase distributions that are not DVBB with even more min-entropy. If a pattern has wildcards in the first $a \leq n/2$ positions, 0 or $\star$ in the next $n-2a$ positions and 0,1 or $\star$ in the last a positions, then an attacker can do polynomial interpolation on the $(n-a)+a$ values $\{[v_{i,0}]_g\}_{i\in[n-a]}\cup\{[v_{i,1}]_g\}_{i\in[a]}$ to detect this. If we pick $a=\omega(\log n)$ and we sample from these patterns uniformly we get an evasive distribution. So, similar to the proof of Lemma 3 this leads to an insecure distribution.

Lemma 4 (Insecure distribution with high min-entropy). *There exists a sequence of distributions $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ with $n-2a+\log(3)a+1$ bits of min-entropy such that the obfuscation scheme of [3] is not $\mathcal{D}$-DVBB secure if $a=\omega(\log n)$.*

We showed that the construction is not VBB secure because by looking at $\mathcal{O}(\rho)$ it is possible to learn something about ρ in polynomial time that would take $O(2^{n/2})$ black box queries to learn otherwise. Later, we will prove that this is essentially the best attack (assuming KOALA). Specifically, we prove that anything that can be learned from an obfuscation of f can also be learned from roughly $2^{n/2}$ black box queries to f_ρ (see Theorem 8).

6.2 Pattern Matching from Big Subset

Next, we show how to derive an obfuscation scheme for pattern matching starting with that for big subset.

Theorem 7 (Pattern matching with wildcards obfuscator from big subset obfuscator). *For an obfuscation scheme $\mathcal{O}$ for the big subset functionality, there exists an obfuscator $\mathcal{O}'$ for the pattern matching with wildcards functionality such that:*

1. *If $\mathcal{O}$ is T-VBB secure with simulators making Q black box queries, then $\mathcal{O}'$ is $(T + Q2^{n/2})$-VBB secure.*
2. *If $\mathcal{O}$ is T-VBB secure with simulators making Q black box queries, then $\mathcal{O}'$ is $(T + Q(2^w + n))$-VBB secure for pattern matching with up to w wildcards.*
3. *For a sequence of distributions $\{\mathcal{D}'_n\}_{n\in\mathbb{N}}$ of length n patterns, let $\mathcal{D}_n = (Y_{\mathcal{D}'_n}, 2n, n)$, where for pattern ρ, the subset Y_ρ is defined as*

$$2i - j \in Y_\rho \Leftrightarrow \rho_i = \star \text{ or } \rho_i = j\,.$$

Then, if $\mathcal{O}$ is $\mathcal{D}$-DVBB secure with simulators that don't make black box queries, then $\mathcal{O}'$ is $\mathcal{D}'$-DVBB secure with simulators that don't make black box queries.

Proof. The obfuscator $\mathcal{O}'$ works as follows:

- To obfuscate a pattern $\rho \in \{0, 1, \star\}^n$ the obfuscator $\mathcal{O}'$ simply outputs $\mathcal{O}(Y_\rho, n, 2n)$.
- To evaluate the Obfuscated program at input $\mathbf{x} \in \{0,1\}^n$, one simply outputs $\mathcal{O}(Y_\rho, 2n, n)(X_\mathbf{x})$, where

$$X_\mathbf{x} = \{2i - j \mid (i, j) \in [n] \times \{0, 1\} \text{ s.t. } x_i = j\}\,.$$

To prove 1, assume that $\mathcal{A}$ is an adversary against the $\mathcal{O}'$ obfuscator, and that $\frac{1}{p(n)}$ is the desired simulator accuracy. We can use $\mathcal{A}$ as an adversary to $\mathcal{O}$, so if $\mathcal{O}$ is T-VBB secure there exists a simulator $\mathcal{S}$, running in time $O(T * poly(n))$, such that for sufficiently large n we have

$$\left|\Pr_{\mathcal{O},\mathcal{A}}[\mathcal{A}(\mathcal{O}(Y_\rho, 2n, n)) = 1] - \Pr_{\mathcal{S}}[\mathcal{S}^{f_{Y_\rho,2n,2}}(1^n) = 1]\right| \leq \frac{1}{p(n)}\,.$$

So $\mathcal{S}$ is almost a good simulator to prove T-VBB security for $\mathcal{O}'$, the only problem is that $\mathcal{S}$ makes black box queries to $f_{Y_\rho,2n,n}$ instead of to f_ρ. To solve this

problem it suffices to prove that one can answer queries to $f_{Y_\rho,2n,n}$ using at most $O(2^{n/2})$ queries to f_ρ.

If $f_{Y_\rho,2n,n}$ is queried on input X with $|X| < n$ we can return 0 without making any queries to f_ρ. We define

$$\begin{aligned}\mathsf{Wildcards} &= \{i \,|\, 2i \in X \text{ and } 2i-1 \in X\}\,,\\ \mathsf{Zeros} &= \{i \,|\, 2i \in X \text{ and } 2i-1 \notin X\} \text{ and}\\ \mathsf{Ones} &= \{i \,|\, 2i \notin X \text{ and } 2i-1 \in X\}\,.\end{aligned}$$

Since $2|\mathsf{Wildcards}| + |\mathsf{Ones}| + |\mathsf{Zeros}| = |X| \geq n$ we have $|\mathsf{Wildcards}| + |\mathsf{Ones}| + |\mathsf{Zeros}| \geq n/2$. This means there are at most $2^{n/2}$ inputs $\mathbf{x}$ that are zero at the indices of $\mathsf{Wildcards} \cup \mathsf{Zeros}$ and one at the indices in Ones. We query f_ρ at each of these inputs. If each of these queries returns 0 we know that $X \not\subset Y_\rho$, so we return 0. If one of the queries returns a 1 we can do n additional black box queries to f_ρ to recover the entire pattern ρ and we output 1 only if $X \subset Y_\rho$.

This shows that there is a simulator $\mathcal{S}'$ for $\mathcal{O}'$ with negligible simulation error that runs in time $O(T * poly(n) + Q2^{n/2})$, which proves 1. For 2 we observe that if ρ has at most w wildcards, then $|\mathsf{Wildcards}| \leq w$ which implies $|\mathsf{Wildcards}| + |\mathsf{Ones}| + |\mathsf{Zeros}| \geq n - w$. Therefore we can answer each query to $f_{Y_{\rho,2n,n}}$ in time $O(2^w + n)$ which proves 2.

To prove 3, assume $\mathcal{A}$ is an adversary against the $\mathcal{O}'$ obfuscator and $\{P'_n\}_{n\in\mathbb{N}}$ a sequence of predicates. Define a sequence of predicates $P_n : \{P([2n], 2n, [2n])\} \to \{0,1\}$ such that $P_n((Y_\rho, 2n, n)) = P'_n(\rho)$ for all $\rho \in 0,1^*$ and with arbitrary behavior on other inputs. By assumption there exists a simulator $\mathcal{S}$ for $(\mathcal{A}, P)$ that makes no black box queries and with negligible simulation error

$$\left| \Pr_{(Y_\rho,2n,n)\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathcal{O}(Y_\rho,2n,n)) = P_n(Y_\rho,2n,n)] - \Pr_{(Y_\rho,2n,n)\leftarrow\mathcal{D}_n}[\mathcal{S}(1^n) = P_n(Y_\rho,2n,n)] \right| .$$

But this simulation error is exactly equal to

$$\left| \Pr_{\rho\leftarrow\mathcal{D}'_n}[\mathcal{A}(\mathcal{O}'(\rho)) = P'_n(\rho)] - \Pr_{(\rho\leftarrow\mathcal{D}'_n}[\mathcal{S}(1^n) = P'_n(\rho)] \right| ,$$

so $\mathcal{S}$ is also a good simulator for $(\mathcal{A}, P')$, which proves that $\mathcal{O}'$ is $\mathcal{D}'$-DVBB secure. □

6.3 Security Guarantees for Construction of [3]

Since the obfuscator of [3] is an instantiation of the transformation of Theorem 7 with the big subset obfuscator whose VBB security we prove in Theorem 5 we can now derive security guarantees. In particular we derive the $2^{n/2}$-VBB security of the obfuscator and we prove that a sequence of distributions that has enough min-entropy is automatically DVBB secure. We prove one statement for distributions that output (Y, n, t) with $t \geq t_0$ for a certain t_0, and one statement

for distributions that output (Y, n, t) with a fixed $t = t_0$, and a fixed size of Y equal to t_1. The following follows immediately from combining Theorem 7 with Theorem 5 (note that VBB security is equivalent to 1-VBB security, because the T-VBB security definition hides polynomial factors in the runtime of the simulators).

Theorem 8 ($\mathcal{O}$ is $2^{n/2}$-VBB secure and 2^w-VBB secure.). *The obfuscator for pattern matching with wildcards from [3] is $2^{n/2}$-VBB secure. If the functionality is restricted to patterns with at most $w(n)$ wildcards, the obfuscator is 2^w-VBB secure.*

The DVVB security of the pattern matching obfuscator for a wide variety of distributions also follows.

Theorem 9 (DVBB security for min-entropy distributions). *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a sequence of distributions over $\{0, 1, \star\}^n$ and let $w(n)$ be a function with $0 \leq w(n) \leq n$, then*

1. *If the min-entropy of $\mathcal{D}_n$ is $n + \omega(\log n)$, then the obfuscation scheme is $\mathcal{D}$-DVBB secure with simulators that make no black box queries.*
2. *If $\mathcal{D}_n$ is supported on patterns with $w(n)$ wildcards, and its min-entropy is $\log(\binom{n}{w(n)}) + \omega(\log n)$, then the obfuscation scheme is $\mathcal{D}$-DVBB secure with simulators that make no black box queries.*

Proof. The embedding of pattern matching instances into big subset instances $\rho \mapsto (Y_\rho, 2n, n)$ is injective so it preserves min-entropy. Now point 1 of Lemma 2 with $t_0(2n) = n$ says that if the min-entropy of $\mathcal{D}' = (Y_\rho, 2n, n)$ is $n + \omega(\log n)$, then the obfuscation scheme for the big subset functionality is $\mathcal{D}'$-DVBB secure with a simulator that makes no black box queries. From this, Theorem 7 says that the obfuscation scheme for pattern matching is $\mathcal{D}$-VBB secure with simulators that make no black box queries.

To prove 2, we use point 2 of Lemma 2 with $t_0(2n) = n$ and $t_1(2n) = n+w(n)$. This tells us that if the min-entropy of $\mathcal{D}_n$ is $\log(\binom{n}{w(n)}) + \omega(\log n)$ then the big subset-obfuscator is DVBB secure. From this, Theorem 7 says that the pattern matching-obfuscator is $\mathcal{D}$-DVVB secure. □

The min-entropy bounds of theorem 9 are almost optimal. The generalized attack of Lemma 4 gives a distribution which has min-entropy larger than $n - \omega(\log n)$. Similarly, we can construct distributions of functions with exactly $w(n)$ wildcards for which the scheme is not DVBB secure that have min-entropy at least $\log(\binom{n}{w(n)}) - \omega(\log n)$.

From the min-entropy criteria it follows immediately that the obfuscator of [3] is DVVB secure for uniform distributions, and uniform distributions with a fixed number of wildcards.

Theorem 10 (DVBB security for uniform distributions). *Let $\mathcal{O}$ be the obfuscator from [3], and let $w(n)$ be a function with $n \leq w(n) \leq n$, such that $n - w(n)$ is $\omega(\log n)$ then*

1. *$\mathcal{O}$ is DVBB secure for the sequence of uniform distributions of patterns of length n, and*
2. *$\mathcal{O}$ is DVBB secure for the sequence of uniform distributions of length n patterns with $w(n)$ wildcards*

Proof. There are 3^n patterns of length n, so the min-entropy of the uniform distributions is $\log(3)n$, which is clearly $n + \omega(\log n)$. The claim now follows from Theorem 9.

For 2, there are $\binom{n}{w(n)}2^{n-w(n)}$ patterns, so the min-entropy of the distribution is $\log(\binom{n}{w(n)}) + n - w(n)$, so again the claim follows from Theorem 9. □

Remark 3. The condition that $n - w(n)$ is $\omega(\log n)$ is essentially optimal, because if the number of non-wildcards is $O(\log n)$, then an adversary can find an accepting input in polynomial time and recover the entire pattern with n additional black box queries.

7 Generalizing the Scheme

A natural question is whether we can generalize the scheme of [3] to create a scheme that is fully VBB secure. For example, one could hope to introduce some extra error terms to the scheme to prevent the attack of Sect. 6.1 and get a fully VBB secure scheme. However, we formulate a big class of generalizations of the scheme and show that all these schemes suffer from an attack similar the one in Sect. 6.1. On the positive side we give a variant of the scheme of [3] which has exactly the same security, but where the obfuscation only consists of $n+1$ group elements instead of $2n$.

7.1 Framework

At a high level, the construction of [3] consist of a mapping $\mathbf{u} : \{0,1\}^n \to \mathbb{Z}_p^m$ that maps an input $\mathbf{x}$ to a vector $\mathbf{u}_\mathbf{x}$ of length m (In the construction we have $m = 2n$), and a mapping V that assigns a vector space V_ρ to each pattern ρ in $\{0,1,\star\}^n$. An obfuscation of the pattern ρ is then $[\mathbf{v}]_g$, where $\mathbf{v}$ is a vector, chosen uniformly from V_ρ. To evaluate the obfuscated program at input $\mathbf{x}$ the evaluator computes $[\mathbf{u}_\mathbf{x}^\top \cdot \mathbf{v}]_g$. If this inner product is $[0]_g$ the evaluator outputs 1, otherwise it outputs 0. For correctness, we require that $\mathbf{u}_\mathbf{x}$ is orthogonal to V_ρ if and only $f_\rho(\mathbf{x}) = 1$. This ensures that the obfuscated program outputs 1 if $f_\rho(\mathbf{x}) = 1$ with probability 1, and 0 if $f_\rho(\mathbf{x}) = 0$ with overwhelming probability $1 - 1/p$.

Definition 10 (Linear-in-the-exponent obfuscation scheme). *A linear-in-the-exponent obfuscation scheme (for pattern matching with wildcards) is a tuple $(\mathbf{u}, V, m(n), \mathcal{O})$, where $\mathbf{u}$ is a mapping $\{0,1\}^n \to \mathbb{Z}_p^{m(n)}$ and V is a mapping that sends patterns in $\{0,1,\star\}^n$ to subspaces of $\mathbb{Z}_p^{m(n)}$ such that*

$$\mathbf{u}_\mathbf{x} \in V_\rho^\perp \Leftrightarrow f_\rho(\mathbf{x}) = 1,$$

and $\mathcal{O}$ is the obfuscation scheme that in input ρ outputs $[v]_g$, for a uniformly chosen vector $v \in V_\rho$. Note that $m(n)$ has to be bounded by a polynomial, because otherwise $\mathcal{O}$ does not have a polynomial slowdown.

Concretely, in the construction of [3] the mapping $\mathbf{u}$ assigns to input $\mathbf{x} \in \{0,1\}^n$ the length-$2n$ vector $\mathbf{u}_\mathbf{x}$ whose $(2i-j)$-th component is the correct polynomial interpolation coefficient if $\mathbf{x}_i = j$, and 0 otherwise. For a pattern ρ, the vector space $V_\rho = C + E_\rho$, where

$$C = \left\{ \{h(i)\}_{i\in[n]} \,|h \text{ a degree } t-1 \text{ polynomial with } h(0) = 0 \right\},$$

and E_ρ is the subspace with basis $\{\mathbf{e}_{2i-j} \,|\, \rho_i \neq \star \text{ and } \rho_i \neq j\}$. We have correctness because $\mathbf{u}_\mathbf{x}$ is orthogonal to C regardless of $\mathbf{x}$, and orthogonal to E if and only if $f_\rho(\mathbf{x}) = 1$.

7.2 Compression

We observed that $\mathbf{u}_\mathbf{x}$ is orthogonal to C, regardless of $\mathbf{x}$. This is because $\mathbf{u}_\mathbf{x} \cdot \mathbf{v}$ corresponds to looking at certain coefficients of $\mathbf{u}$ and doing polynomial interpolation on them at 0, while the entries of $\mathbf{u} \in C$ are precisely the evaluation of a low degree polynomial h with $h(0) = 0$. This shows that $\mathbf{u}$ sends all the inputs $\mathbf{x}$ to a vector in $C^\perp$, which is a subspace of dimension $n+1$. So the scheme is not using the additional $n-1$ dimensions of $\mathbf{Z}_p^{2n}$, which is wasteful. We can "cut out" these extra dimensions to get a scheme $(\mathbf{u}', V')$ which has more compact obfuscated programs consisting of $n+1$ group elements instead of $2n$, but still has the same security. This compression can be performed for any linear-in-the-exponent obfuscation scheme $(\mathbf{u}, V)$ if $\langle \mathbf{u}_\mathbf{x} \,|\, \mathbf{x} \in \{0,1\}^n \rangle \neq \mathbb{Z}_p^m$.

Theorem 11 (Compressing linear-in-the-exponent schemes). *Let $(\mathbf{u}, V, m, \mathcal{O})$ be a linear-in-the-exponent obfuscation scheme, then there exists a scheme $(\mathbf{u}', V', m', \mathcal{O}')$ such that*

$$m'(n) = \dim(\langle \mathbf{u}_\mathbf{x} \,|\, \mathbf{x} \in \{0,1\}^n \rangle) = \dim(\langle \mathbf{u'}_\mathbf{x} \,|\, \mathbf{x} \in \{0,1\}^n \rangle).$$

Let T be a function and $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a sequence of distributions of length-n patterns. If $\mathcal{O}$ is VBB, T-VBB or $\mathcal{D}$-DVBB secure then $\mathcal{O}'$ is VBB, T-VBB or $\mathcal{D}$-DVBB secure respectively.

Proof. For any n, let $U_n = \langle \mathbf{u}_\mathbf{x} \,|\, \mathbf{x} \in \{0,1\}^n \rangle$ be the space spanned by the $\mathbf{u}_\mathbf{x}$. Let $m'(n) = \dim(U_n)$ and let $\mathbf{u}_1, \cdots, \mathbf{u}_{m'(n)}$ be a basis for U and extend this to a basis $u_1, \cdots, u_n$ for all of $\mathbb{Z}_p^{m(n)}$. Let M be the matrix whose columns are the u_i, and $\overline{M^\top}$ and $\overline{M^{-1}}$ the first $m'(n)$ rows of $M^\top$ and M^{-1} respectively. Now we define $\mathbf{u}'_\mathbf{x} = \overline{M^{-1}}\mathbf{u}_\mathbf{x}$ and $V'_\rho = \overline{M^\top} V_\rho$. Let $\mathbf{u}$ be a vector from V_ρ and $\mathbf{v}' = \overline{M^\top}\mathbf{v}$, then we have

$$\mathbf{u}'^\top_\mathbf{x} \cdot \mathbf{v}' = \mathbf{u}^\top_\mathbf{x} \cdot \overline{M^{-1}}^\top \cdot \overline{M^\top} \cdot \mathbf{v} = \mathbf{u}^\top_\mathbf{x} \cdot \mathbf{v}\,.$$

This shows that the new linear-in-the-exponent obfuscation scheme $(\mathbf{u}', V', m', \mathcal{O}')$ is correct if the original scheme is.

To prove that the compression preserves security, let $\mathcal{A}'$ be an adversary that breaks VBB, T-VBB or $\mathcal{D}$-DVBB security of $\mathcal{O}'$, then it is easy to see that the adversary $\mathcal{A}$ that on input $[v]_g$ computes $[\overline{M^\top}\mathbf{v}]_g$ and outputs $\mathcal{A}'([\overline{M^\top}\mathbf{v}]_g)$ is an adversary that breaks VBB, T-VBB or $\mathcal{D}$-DVBB security of $\mathcal{O}$ respectively. □

7.3 Impossibility Result

We have proven that the construction of [3] is essentially only $2^{n/2}$-VBB secure. So constructing a simple, efficient and fully VBB secure construction is still an open problem. A priori, one can hope to find another construction that follows the linear-in-the-exponent paradigm which is fully VBB secure, or perhaps something that is $2^{\sqrt{n}}$-VBB secure. Unfortunately we show that our attack on the construction of [3] generalizes to a wide class of "natural" linear-in-the exponent constructions. Recall that our attack on the scheme of [3] allowed to check whether the first half of an obfuscated pattern consists of wildcards. This was done by interpolating on the first n values. In the language of the linear-in-the-exponent framework this means there is a vector $\mathbf{o}$ (which corresponds to polynomial interpolation) that is orthogonal to V_ρ for every pattern ρ that has wildcards in the first $n/2$ positions. So, given an obfuscated program $\mathcal{O}(\rho) = [\mathbf{v}]_g$, one can test if the first half of the obfuscated pattern ρ consists of wildcards by checking if $[\mathbf{o}^\top \cdot \mathbf{v}]_g = [0]_g$. One crucial element here for the attack work is that $[\mathbf{o}^\top \cdot \mathbf{v}]_g \neq [0]_g$ with a large probability if $[\mathbf{v}]_g$ is the obfuscation of a pattern that does have non-wildcard characters in the first half of the pattern. For the construction of [3] this is obviously true.

The same thing happens for general linear-in-the-exponent obfuscation schemes. We show in Lemma 5 that if $a \leq \frac{n}{\log(m)}$, then there exist a subset $A \subset [n]$ of size a and a non-zero attack vector $\mathbf{o}$ such that $\mathbf{o}$ is orthogonal to V_ρ for every pattern ρ that has only wildcards outside of A. If $\mathbf{o}$ is not orthogonal to obfuscations of uniformly chosen patterns with a non-negligible probability, then this breaks VBB security (and even $2^{\frac{n}{\omega(\log n)}}$-VBB security). Note that without loss of generality we can assume that no vector is orthogonal to *every* V_ρ, because otherwise we can use the compression trick to obtain a more efficient and equally secure scheme. Schemes for which each vector is not orthogonal to a significant fraction of the V_ρ are called natural. We then prove that there are no natural linear-in-the-exponent obfuscation schemes.

Definition 11 (Natural linear-in-the-exponent schemes). *A linear-in-the-exponent obfuscation scheme $(\mathbf{u}, V, m, \mathcal{O})$ is called* natural *if there exists a polynomial $p(n)$ such that for all vectors $\mathbf{o}$*

$$\Pr_{\rho \leftarrow \{0,1,\star\}^n}[\mathbf{o} \notin V_\rho^\perp] \geq \frac{1}{p(n)}\,.$$

Remark 4. All schemes where $m(n) = n + 1$ are natural.

Lemma 5 (There exist vectors orthogonal to patterns with wildcards at fixed positions). *Let $(\mathbf{u}, V, m, \mathcal{O})$ be a linear-in-the-exponent obfuscation scheme for pattern matching with wildcards. Then if $a(n) \leq \frac{n}{\log(m(n))}$ then there exist subsets $A_n \subset [n]$ of size $|A_n| = a(n)$ and non-zero vectors $\mathbf{o}_n$ such that $\mathbf{o}_n$ is orthogonal to V_ρ for any pattern ρ that has wildcards outside of A_n.*

Proof. Suppose $(\mathsf{u}, V, m, \mathcal{O})$ is a linear-in-the-exponent obfuscation scheme for pattern matching with wildcards. Let $a(n)$ be a function such that $2^{\lfloor \frac{n}{a(n)} \rfloor} > m(n)$ then we will prove that there is a subset $A \subset [n]$ of size $|A| = a$ together with a non-zero vector $\mathbf{o}$ such that $\mathbf{o}$ is orthogonal to V_ρ for all patterns ρ such that $\rho_i = \star$ for all i outside of A. It suffices to show this in the case that the V_ρ are maximal given the correctness constraints, i.e.

$$V_\rho = \langle \mathbf{u}_\mathbf{x} \,|\, \mathbf{x} \in \{0,1\}^n : \rho(\mathbf{x}) = 1 \rangle^\perp .$$

Clearly, if we prove there exists a vector $\mathbf{o}$ that is orthogonal to the maximal V_ρ, than this $\mathbf{o}$ will also be orthogonal to whatever the V_ρ are in any other linear-in-the-exponent obfuscation scheme with the same $\mathbf{u}$ map.

For $i \in [n]$ and $j \in \{0, 1, \star\}$ let $e_{i,j}$ be the pattern that has the character j at position i and wildcards at all other positions. Then we have for a general pattern ρ that $f_\rho(\mathbf{x}) = 1$ if and only if $f_{e_{i,\rho_j}}(\mathbf{x}) = 1$ for all $i \in [n]$. Therefore we have that

$$\begin{aligned} V_\rho &= \langle \mathbf{u}_\mathbf{x} \,|\, \mathbf{x} \in \{0,1\}^n : f_{e_{i,\rho_i}}(\mathbf{x}) = 1 \ \forall i \in [n] \rangle^\perp \\ &= \sum_{i=1}^{n} \langle \mathbf{u}_\mathbf{x} \,|\, \mathbf{x} \in \{0,1\}^n : f_{e_{i,\rho_i}}(\mathbf{x}) = 1 \rangle^\perp = \sum_{i=1}^{n} V_{e_{i,\rho_i}} . \end{aligned}$$

Working towards a contradiction, suppose no $A \subset [n]$ of size $|A| = a$ and $\mathbf{o}$ exists. This means that for every set $A \subset [n]$ of size $|A| = a$ we have

$$\cap \{ V_\rho^\perp \,|\, \rho : \rho_i = \star \forall i \notin A \} = \{0\} ,$$

which is equivalent to $\sum \{ V_\rho \,|\, \rho : \rho_i = \star \forall i \notin A \} = \mathbb{Z}_p^{m(n)}$. Using the fact that $V_\rho = \sum_{i=0}^{n} V_{e_{i,\rho_i}}$ this is the same as

$$\left(\sum_{i \in A} V_{e_{i,0}} \right) + \left(\sum_{i \in A} V_{e_{i,1}} \right) = \mathbb{Z}_p^{m(n)} . \tag{6}$$

Pick $\lceil \frac{n}{a} \rceil$ disjoint subsets $A_1, \cdots, A_k$, each of size a, and define the vector spaces

$$V_i^j = \sum_{i \in A_i} V_{e_{i,j}} .$$

Then Eqn. 6 says that for any i we have $V_i^0 + V_i^1 = \mathbb{Z}_p^{m(n)}$. At the same time we have for any $\mathbf{y} \in \{0,1\}^k$ that

$$V_\mathbf{y} = \sum_{i=1}^{k} V_i^{\mathbf{y}_i} \neq \mathbb{Z}_p^n ,$$

because if ρ is the pattern such that $\rho_i = \mathbf{y}_j$ if $i \in A_j$ and $\rho_i = \star$ otherwise, then $V_\rho = \sum_{i=1}^{k} V_i^{y_i}$, and V_ρ is not equal to $\mathbb{Z}_p^m(n)$ because by correctness it is orthogonal to $\mathbf{x}$ for any $\mathbf{x}$ that is accepted by ρ.

Now we show that the 2^k none of the spaces ${V_{\mathbf{y}}}^{\perp}$ are included in the sum of the other $2^k - 1$ ones. Indeed, suppose $V_{\mathbf{y}}^{\perp} \subset \sum_{\mathbf{y}' \neq \mathbf{y}} V_{\mathbf{y}'}^{\perp}$, which is equivalent to $V_{\mathbf{y}} \supset \bigcap_{\mathbf{y}' \neq \mathbf{y}} V_{\mathbf{y}'}$, then after adding $V_{\mathbf{y}}$ to both sides we get

$$V_{\mathbf{y}} \supset \bigcap_{\mathbf{y}' \neq \mathbf{y}} (V_{\mathbf{y}'} + V_{\mathbf{y}}) \ .$$

But this is a contradiction because the left hand side is not equal to $\mathbb{Z}_p^{m(n)}$ while each space in the intersection is equal to $\mathbb{Z}_p^{m(n)}$ because if $\mathbf{y}_i \neq \mathbf{y}'_i$, then $V_{\mathbf{y}'} + V_{\mathbf{y}}$ contains $V_i^0 + V_i^1 = \mathbb{Z}_p^{m(n)}$. The fact that none of 2^k subspaces of $\mathbb{Z}_p^{m(n)}$ is included in the sum of the other $2^k - 1$ ones implies that $m(n) \geq 2^k = 2^{\lfloor \frac{n}{a} \rfloor}$, which contradicts the assumption that $m(n) < 2^{\lfloor \frac{n}{a} \rfloor}$. □

The following theorem follows readily from Lemma 5 and the discussion above.

Theorem 12 (Limitations of linear-in-the-exponent obfuscation). *There are no natural linear-in-the-exponent obfuscation schemes that are $2^{\frac{n}{\omega(\log n)}}$-VBB secure.*

Since every scheme with the minimal dimensionality of $m = n+1$ is automatically natural, this implies that no VBB secure constructions with $m(n) = n+1$ exist.

Proof. Let $a(n) = \frac{n}{\log(m(n))} - 1$. Then Lemma 3 says that there exists an A_n of size $a(n)$ and vectors $\mathbf{o}_n$ such that $\mathbf{o}$ is orthogonal to V_ρ for all patterns ρ that have wilcards outside of A_n. Sampling uniformly from patterns that have wildcards outside of A_n and no wildcards at locations in A_n are $2^{a(n)}n^{-\omega(1)}$-elusive. But obfuscations of these patterns can be efficiently distinguished from obfuscations of uniformly random patterns (which are also elusive) with the vectors $\mathbf{o}_n$, because the former are orthogonal to $\mathbf{o}$, and the latter are not with non negligible probability (because of the naturality assumption). Then it follows from Lemma 1 that the scheme is not $2^{a(n)}n^{-\omega(1)}$-VBB secure. The claim follows because $2^{a(n)}n^{-\omega(1)})$ is eventually bigger than $2^{n/f(n)}$ for every $f(n)$ that is $\omega(\log n)$. □

Acknowledgements. This work started at ENS over the summer; we thank Luke Kowalczyk for telling us about [3], as well as Michel Abdalla, Georg Fuchsbauer and Hendrik Waldner for helpful discussions. This work was supported in part by the Research Council KU Leuven: C16/15/058, C14/18/067 and STG/17/019. In addition, this work was supported by the European Commission through the Horizon 2020 research and innovation programme under grant agreement H2020-DS-LEIT-2017-780108 FENTEC, by the Flemish Government

through FWO SBO project SNIPPET and by the IF/C1 on Cryptanalysis of post-quantum cryptography. Ward Beullens is funded by an FWO fellowship. Hoeteck Wee is supported by ERC Project aSCEND (H2020 639554).

References

1. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im) possibility of obfuscating programs. In: CRYPTO. pp. 1–18. Springer (2001)
2. Bartusek, J., Lepoint, T., Ma, F., Zhandry, M.: New techniques for obfuscating conjunctions. Cryptology ePrint Archive, Report 2018/936 (2018), https://eprint.iacr.org/2018/936
3. Bishop, A., Kowalczyk, L., Malkin, T., Pastro, V., Raykova, M., Shi, K.: A simple obfuscation scheme for pattern-matching with wildcards. In: CRYPTO. pp. 731–752. Springer (2018)
4. Bitansky, N., Canetti, R.: On strong simulation and composable point obfuscation. In: Annual Cryptology Conference. pp. 520–537. Springer (2010)
5. Brakerski, Z., Rothblum, G.N.: Obfuscating conjunctions. In: CRYPTO, Part II. pp. 416–434 (2013)
6. Brakerski, Z., Vaikuntanathan, V., Wee, H., Wichs, D.: Obfuscating conjunctions under entropic ring LWE. In: ITCS. pp. 147–156. ACM (2016)
7. Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: CRYPTO. pp. 455–469. Springer (1997)
8. Canetti, R., Dakdouk, R.R.: Obfuscating point functions with multibit output. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 489–508. Springer (2008)
9. Canetti, R., Rothblum, G.N., Varia, M.: Obfuscation of hyperplane membership. In: TCC. pp. 72–89. Springer (2010)
10. Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: CRYPTO. pp. 445–456. Springer (1991)
11. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: CRYPTO. pp. 33–62. Springer (2018)
12. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. SIAM Journal on Computing 45(3), 882–929 (2016)
13. Goyal, R., Koppula, V., Waters, B.: Lockable obfuscation. In: FOCS. pp. 612–621 (2017)
14. Hada, S.: Zero-knowledge and code obfuscation. In: ASIACRYPT. pp. 443–457 (2000)
15. Lynn, B., Prabhakaran, M., Sahai, A.: Positive results and techniques for obfuscation. In: International conference on the theory and applications of cryptographic techniques. pp. 20–39. Springer (2004)
16. Wee, H.: On obfuscating point functions. In: STOC. pp. 523–532. ACM (2005)
17. Wichs, D., Zirdelis, G.: Obfuscating compute-and-compare programs under LWE. In: FOCS. pp. 600–611 (2017)