D4.5 Annual Report on Functional Encryption Schemes for Richer Functionalities Y2

Document Identification			
Status	Final	**Due Date**	31/12/2019
Version	1.0	**Submission Date**	20/12/2019

Related WP	WP4	**Document Reference**	D4.5
Related Deliverable(s)	D4.4	**Dissemination Level(*)**	PU
Lead Participant	UEDIN	**Lead Author**	Hendrik Waldner
Contributors	UEDIN	**Reviewers**	Michel Abdalla (ENS) Clément Gentilucci (FUAS)

Keywords:
Functional Encryption, Indistinguishable Obfuscation, Virtual Black Box Obfuscation

This document is issued within the framework and for the purpose of the FENTEC project. This project has received funding from the European Union's Horizon2020 under Grant Agreement No. 780108. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

Document Information

List of Contributors	
Name	Partner
Hendrik Waldner	UEDIN

Document History			
Version	Date	Change editors	Changes
0.1	07/11/2019	Hendrik Waldner (UEDIN)	ToC
0.2	12/12/2019	Hendrik Waldner (UEDIN)	Version for reviewing
0.3	17/12/2019	Hendrik Waldner (UEDIN)	Addressed reviewers comments
1	18/12/2019	Hendrik Waldner (UEDIN)	Final version

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable Leader	Hendrik Waldner (UEDIN)	18/12/2018
Technical Manager	Michel Abdalla (ENS)	18/12/2018
Quality Manager	Diego Esteban (ATOS)	18/12/2018
Project Coordinator	Francisco Gala (ATOS)	18/12/2018

Table of Contents

List of Acronyms

Acronym	Description
D-VBB	Distributional Virtual Black Box
FE	Functional Encryption
iO	Indistinguishable Obfuscation
KOALA	Knowledge of OrthogonALity Assumption
PPT	Probabilistic Polynomial Time
VBB	Virtual Black Box

Executive Summary

Program obfuscation for all circuits is a long-standing open problem and a very active area of research. Proposed solutions for this problem rely on heavy mathematical tools such as multilinear maps and it is unknown how to instantiate this primitive from more standard assumptions. Nevertheless, it turned out that it is possible to construct obfuscators for more restricted functionality classes from standard assumptions. One recent result in this area consists of an obfuscator for the pattern matching with wildcards functionality, as shown in the work of Bishop et al. [3]. In this deliverable, we present practical obfuscators for two different functionality classes and build up on the results of Bishop et al. [3]. Due to the close relation between program obfuscation and functional encryption, this result tackles one of the main goals of WP4, the construction of practical functional encryption schemes.

In more detail, we describe a new knowledge assumption called KOALA introduced by Beullens and Wee [2] in the context of the FENTEC project. Using this assumption, we present the practical virtual black box (VBB) obfuscators for the big subset functionality and the pattern matching with wildcards functionality as constructed in [2]. The results of Beullens and Wee directly relate to the work of Bishop et al. [3], therefore, we also present the comparison and analysis made in [2] between the two different works.

1 Introduction

Functional encryption [5, 11] allows the delegation of fine-grained access control on encrypted data to different users. In comparison to public-key encryption schemes, and their accompanying "all-or-nothing" approach, functional encryption schemes allow for the generation of so-called functional keys sk_f. These keys are, as the name suggests, associated with a function f and, if used in the decryption procedure, they output the associated function f applied to the underlying plaintext $f(m)$ instead of the whole plaintext m. In this setting, the function f stems from a larger functionality class F.

Another notion, which is very close to the concept of functional encryption, is obfuscation [1]. Program obfuscation tries to achieve the goal of making computer programs "unintelligible". Unfortunately, it has been proven that it is impossible to achieve "black-box" program obfuscation (so called virtual black box obfuscation) for general programs, i.e. there exists functions that are unobfuscatable. Due to this impossibility result, the notion of indistinguishable obfuscation $i\mathcal{O}$ has been introduced. An indistinguishable obfuscator $i\mathcal{O}$ for a class of circuits $\mathcal{C}$ guarantees that given two equivalent circuits $C_1, C_2 \in \mathcal{C}$, the two distribution of obfuscations $i\mathcal{O}(C_1)$ and $i\mathcal{O}(C_2)$ should be computationally indistinguishable.

It has been proven, in the work of Garg et al. [10], that indistinguishable obfuscation for all circuits implies functional encryption for all circuits, which proves the strong connection between these two primitives. The fact that functional encryption implies indistinguishable obfuscation has been proven in the work of Bitansky and Vaikuntanathan [4]. This shows the equivalence of these two primitives. Unfortunately, no practical instantiation for a general obfuscator, and therefore a general functional encryption scheme, is known at the moment. Nevertheless, as in the case of functional encryption, it turned out that it is possible to obfuscate more specific classes of functions, even under the stronger notion of virtual black box obfuscation. In this deliverable, we present recent results on virtual black box obfuscation for two different functionalities: the big subset functionality and the pattern matching with wildcards functionality.

Before we describe the functionality classes and their obfuscators, we need to introduce a complexity assumption called the Knowledge of OrthogonALity Assumption (KOALA).

Knowledge of OrthogonALity Assumption. The Knowledge of OrthogonALity Assumption (KOALA) is a natural decisional analogue of Damgard's KEA assumption [8], 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 $\mathbf{zM} = \mathbf{0}$. The assumption is a natural decisional analogue of the recent algebraic group model [9], which essentially asserts that the only way an adversary can compute a new group element is to take a linear combination of previous ones. Due to its analogy to the algebraic group model it follows directly that KOALA is weaker than the generic group model [12].

Big Subset Functionality. The big subset functionality, for input size n, consists of functions that 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. We also show that the big subset functionality directly implies the pattern matching with wildcards functionality and therefore a VBB secure obfuscator for the big subset functionality covers obfuscation for the pattern matching with wildcards functionality.

Pattern Matching with Wildcards Functionality. The pattern matching with wildcards functionality, for input size n, consists of functions that are parameterized by a pattern $\rho \in \{0,1,\star\}^n$. The function $f_\rho : \{0,1\}^n \to \{0,1\}$ takes a string $\mathbf{x} \in \{0,1\}^n$ as input and outputs 1 if and only if $\mathbf{x}$ matches the pattern,

i.e. if $\rho_i = x_i$ or $\rho_i = \star$ for all $i \in [n]$. Recently, Bishop et al. [3] presented a simple and efficient method for obfuscating pattern matching with wildcards 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 $0.75n$. The work of Beullens and Wee [2] shows how to improve the work of Bishop et al. [3] in a number of directions. First, they 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 secure. Moreover, the attack shows that there are high entropy distributions for which the scheme is not DVBB secure. On the other hand they prove stronger security claims by proving the scheme to be VBB secure. To achieve this, Beullens and Wee show that the obfuscator for the big subset functionality also fulfills the requirements for the pattern matching with wildcards functionality. This can be done by embedding of $\{0, 1, \star\}^n$ into $(\mathcal{P}([2n]), 2n, n)$ where the i'th symbol is replaced with either $2i - 1, 2i$ or both. Indeed, this was the approach (implicitly) taken in [3].

1.1 Purpose of the Document

A primary goal of this deliverable is to give an overview of the contributions made by the FENTEC partners in the area of functional encryption for expressive functionalities. Towards this goal, we present different virtual black box obfuscators, covering the big subset functionality and the pattern matching with wildcards functionality. This obfuscators rely on a knew knowledge assumption called KOALA. The knowledge assumption and the resulting obfuscators have been introduced by Beullens and Wee [2] in the context of the FENTEC project. The presented obfuscators are reasonably efficient and represents the current state of the art in the area of virtual black box obfuscation for specific functionalities.

1.2 Structure and Methodology

Section 2 first recalls the notions and security definitions we use in the remainder of the document. Section 3 describes the new Knowledge of OrthogonALity Assumption and how to construct a point obfuscator from it. The former and the latter are results of the work of Beullens and Wee [2]. In Section 4, we recap their definition of the big subset functionality and present their corresponding obfuscator based on KOALA. In the final section, Section 5, we present the analysis of Beullens and Wee of the existing obfuscator for the pattern matching with wildcards functionality of Bishop et al. [3] and recap their proof that the obfuscator for the big subset functionality of Section 4 implies an obfuscator for the pattern matching with wildcards functionality.

2 Preliminaries

In this section, we recall some of the security definitions and basic tools that will be used in the remainder of the document.

2.1 Notation and conventions

Let $\mathbb{N}$ denote the set of natural numbers. If $n \in \mathbb{N}$, then $\{0,1\}^n$ denotes the set of n-bit strings, and $\{0,1\}^*$ is the set of all bit strings. We use use $[n]$ to denote the set $\{1, \cdots, n\}$. More generally, if S is a set, then S^n is the set of n-tuples of elements of S. If x is a string then $|x|$ denotes its length, and if S is a set then $|S|$ denotes its size. If S is finite, then $x \leftarrow S$ denotes the random assignment to x of an element chosen uniformly at random from S. If $\mathcal{A}$ is an algorithm, then $y \leftarrow \mathcal{A}(x)$ denotes the assignment to y of the output of $\mathcal{A}$ on input x. Unless otherwise indicated, an algorithm may be randomized. Most of the time we denote by λ the security parameter. A function $\nu : \mathbb{N} \to [0,1]$ is said to be *negligible* if for every $c \in \mathbb{N}$ there exists a $\lambda_c \in \mathbb{N}$ such that $\nu(\lambda) \leq \lambda^{-c}$ for all $\lambda > \lambda_c$. 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 $\{g^{v_i}\}_{i\in[n]}$. The *min-entropy* of a random variable X is $H_\infty := -\log(\max_x \Pr[X = x])$.

2.2 Obfuscation Security Definitions

In this section we define Virtual Black Box [1] (VBB) and Distributional Virtual Black Box [6] (DVBB) security. We also recap the definition of T-VBB security, a variant of VBB security where the simulator is allowed to run in super-polynomial time $O(T)$ as introduced by Beullens and Wee.

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 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 $O(1^n, f)$ that:

1. preserves functionality: For any n, $f \in \mathcal{F}_n$ and $\mathbf{x} \in \{0,1\}^n$ we have that $O(1^n, f)(\mathbf{x}) = f(\mathbf{x})$ 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 $O(1^n, f)$ runs in time that is $\mathsf{poly}(\lambda)(n, T(f))$, 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 O in the rest of the deliverable.

If an obfuscator reveals no more information about the function $f \in \mathcal{F}_n$ than what can be 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 O for the functionality $\mathcal{F} := \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ 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 a n_0 such that for any $n \geq n_0$ and any $f \in \mathcal{F}_n$*

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

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 obfuscates 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 $O(T * \mathsf{poly}(\lambda)(n))$ and a n_0 such that for any $n \geq n_0$ and $f \in \mathcal{F}_n$*

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

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*

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

The fact that VBB security implies distributional VBB security for any arbitrary distribution is trivial. However, Beullens and Wee proved that VBB security also implies DVBB security with simulators that do not make black-box queries for distributions that are evasive and that T-VBB implies DVBB with simulators that make no black-box queries for distributions which are T-evasive. We recap the definition and the corresponding lemma here.

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.*

The proof for this lemma can be found in [2].

3 Knowledge of OrthogonALity Assumption (KOALA)

Beullens and Wee introduced a new assumption, the Knowledge of OrthogonALity Assumption (KOALA). Based on this assumption, they proved the security of the different obfuscation schemes we present in this deliverable. Informally, the assumption states 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. More formally:

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*

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

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

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

3.1 Obfuscating Point Functions from KOALA

To demonstrate the power of KOALA, Beullens and Wee proved the VBB security of the simple point function obfuscator of [7]. To obfuscate the function that tests whether an input $x \in \mathbb{Z}_p$ 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 1 (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.*

The proof of this theorem can be found in [2].

Before we can recap the definition of VBB self composability [2], we need to recap the definition of an array of functions [2]. An array of functions is basically a function containing k different functions that takes two inputs (i, x) and outputs the value of the i'th function on input x, i.e. $f_i(x)$. More formally:

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

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

The point function obfuscator from [7] fulfills VBB self composability.

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

A detailed proof can be found in [2].

As shown by Beullens and Wee, the point obfuscator can also be used to obfuscate multi-bit output point functions. A multi-bit output function is a function that is parameterized by two different values **a** and **b**. The function outputs the value **b**, if it receives **a** as an input. Otherwise it outputs $\perp$. More formally:

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

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

Theorem 3 (Obfuscating multi-bit output point functions) *Suppose O is a VBB self-composable obfuscator for point functions, then there exists a VBB self-composable obfuscator O' for point functions with multi-bit output*

The construction of O' from O, makes use of the self composability of O. The detailed construction can be found in [2].

4 Obfuscator for the Big Subset Functionality

Now, we introduce the big subset functionality as defined by Beullens and Wee. The big subset functionality, for input size n, is a function $f_{Y,n,t} : \mathcal{P}([n]) \to \{0,1\}$ parameterized by a subset $Y \subset [n]$ and a threshold value $0 \leq t \leq n$. It takes as an input a subset $X \subset [n]$ and outputs 1 if and only if X is a big enough subset of Y (i.e. $|X| \geq t$), i.e. $f_{Y,n,t}(X) = 1 \Leftrightarrow |X| \geq t$ and $X \subseteq Y$.

Formally:

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*

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

4.1 VBB Secure Obfuscation of the Big Subset Functionality

We can now describe how the VBB secure obfuscator for the big subset functionality of [2] works.

To build an obfuscator for $f_{Y,n,t}$, it is necessary to comprise n group elements $[v_1]_g, \ldots, [v_n]_g$ 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, the obfuscator simply returns 1 if and only if the reconstruction "in the exponent" over the shares corresponding to X returns $[0]_g$.

In other words: 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 $\{(i, o_i) \mid i \in X\}$ lie on a degree $|X|-1$ polynomial h_x with $h_x(0) = 0$.

Under KOALA this construction is a VBB secure obfuscator.

Theorem 4 (O is VBB secure) *Let O be the obfuscator for the big subset functionality defined above, using a family of cyclic groups that satisfies KOALA. Then O is VBB secure.*

This security proof can be found in [2].

Beullens and Wee showed that the pattern matching with wildcards obfuscator of [3] contains an obfuscator for the big subset functionality. In the next chapter, we recap their results by describing their 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 the pattern matching with wildcards functionality. This transformation preserves VBB security at the cost of

a slowdown of the simulator by a factor of $2^{n/2}$. 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, Beullens and Wee proved its VBB security with a super-polynomial simulator and Distributional VBB security for a wide variety of distributions.

5 Obfuscator for Pattern Matching with Wildcards

Now, we recap the definition of the pattern matching with wildcards functionality of [3] as described in [2]. The class of functions for the pattern matching with wildcards functionality is parametrized by length n strings over the alphabet $\{0, 1, \star\}$. For a pattern $\rho = (\rho_i)_{i\in[n]}$ in $\{0, 1, \star\}^n$, 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

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

5.1 A simple and efficient construction

In the work of Bishop et al. [3], the authors present a simple obfuscation scheme for the pattern matching with wildcards functionality. We recap the idea of the scheme informally:

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 rejects it. 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 $1 - \frac{1}{p}$.

5.1.1 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.

Beullens and Wee investigated the security of the obfuscation scheme of [3] further. On the negative side, they found an attack that allows an adversary to learn if the first half of the pattern consists of wildcards. This shows that the scheme cannot be VBB secure, and even that the scheme is not DVBB secure for some high entropy distributions. On the positive side however, they could show that the scheme is VBB secure for a super-polynomial simulator.

Beside this, they also proved that any distribution of patterns that has at least $n + \omega(\log n)$ bits of min-entropy automatically guarantees security and they were able to give similar bounds for distributions that output patterns with a fixed number of wildcards. The attacks presented by Beullens and Wee match these min-entropy bounds and therefore they show their optimality. Their bounds immediately proved that the scheme is DVBB secure for uniform patterns and uniform patterns with a fixed number of wildcards up to $n - \omega(\log n)$. Therefore, they presented a stronger result than the result of [3], which only proves DVBB security for uniform distributions of up to $\frac{3n}{4}$ wildcards. Beullens and Wee also analyzed that for $n - O(\log n)$ wildcards a pattern can be recovered through black box queries in polynomial time and VBB security is trivial, therefore, having up to $n - \omega(\log n)$ wildcards, is the optimal case. If there are only $O(\log n)$ non wildcards, then after polynomially many black box queries at random inputs it is possible to get an accepting input. Once an accepting input $\mathbf{x} \in \{0,1\}^n$ is found it is possible to learn the entire pattern with n additional black box queries on the n inputs that differ from $\mathbf{x}$ at exactly one position.

5.2 The Construction of Bishop et al. is not VBB Secure

Now, we present the attack for VBB security for the scheme of Bishop et al. [3] presented in [2].

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. More technically:

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 2 (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.*

Theorem 5 (O is not $2^{0.5n}n^{-\omega(1)}$-VBB secure.) *Let O be the obfuscation scheme for pattern matching with wildcards from [3], then O is not $2^{0.5n}n^{-\omega(1)}$-VBB secure.*

The detailed proofs for Lemma 2 and Theorem 5 can be found in [2].

The distribution of Lemma 2 has $n/2 + 1$ bits of min-entropy, but, as described by Beullens and Wee, 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 identify this structure. If $a = \omega(\log n)$ then a uniform sample from these patterns is an evasive distribution. So, similar to Lemma 2 this leads to an insecure distribution.

Lemma 3 (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 $a=\omega(\log n)$.*

For a detailed proof of Lemma 3 we refer to the proof of Lemma 2.

This proves that the construction is not VBB secure. By looking at $O(\rho)$ it is possible to learn something about ρ in polynomial time, which would take $O(2^{n/2})$ black box queries to learn otherwise. In Section 5.3.1, we recap the result of [2], that this is essentially the best attack under KOALA. Specifically, 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 7).

5.3 Pattern Matching from Big Subset

Next, we present the result of Beullens and Wee how to derive an obfuscation scheme for the pattern matching with wildcards functionality, by relying on an obfuscator for the big subset functionality.

Theorem 6 (Pattern matching with wildcards obfuscator from big subset obfuscator) *For an obfuscation scheme O for the big subset functionality, there exists an obfuscator O' for the pattern matching with wildcards functionality such that:*

1. *If O is T-VBB secure with simulators making Q black box queries, then O' is $(T+Q2^{n/2})$-VBB secure.*
2. *If O is T-VBB secure with simulators making Q black box queries, then 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 O is $\mathcal{D}$-DVBB secure with simulators that don't make black box queries, then O' is $\mathcal{D}'$-DVBB secure with simulators that don't make black box queries.

The obfuscator O' in Theorem 6 works as follows:

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

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

The details and a formal proof for Theorem 6 can be found in [2].

5.3.1 Security Guarantees for the Construction of [3]

The security of the obfuscator of [3] directly follows from the fact that it is an instantiation of the obfuscator for the big subset functionality (Theorem 6), i.e. the security is derived from the VBB security of the big subset obfuscator (Theorem 4). In particular, Beullens and Wee showed how to derive the $2^{n/2}$-VBB

security of the obfuscator and that a sequence of distributions that has enough min-entropy is automatically DVBB secure. They proved 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 next theorem is an immediate implication of Theorem 6 and Theorem 4 as presented in [2].

Theorem 7 (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 is stated as follows.

Theorem 8 (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.*

A detailed proof for Theorem 7 and Theorem 8 can be found in [2].

As analyzed by Beullens and Wee, the min-entropy bounds of Theorem 8 are almost optimal. The generalized attack of Lemma 3 results in a distribution which has min-entropy larger than $n - \omega(\log n)$. Similarly, it is possible to construct distributions of functions with exactly $w(n)$ wildcards and min-entropy at least $\log(\binom{n}{w(n)}) - \omega(\log n)$ for which the scheme is not DVBB secure.

From the min-entropy criteria it follows immediately that the obfuscator of [3] is DVBB secure for uniform distributions, and uniform distributions with a fixed number of wildcards, as shown in [2].

Theorem 9 (DVBB security for uniform distributions) *Let 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. *O is DVBB secure for the sequence of uniform distributions of patterns of length n, and*
2. *O is DVBB secure for the sequence of uniform distributions of length n patterns with $w(n)$ wildcards*

The proof for this Theorem can be found in [2].

6 Conclusion

In this document, we presented the results of Beullens and Wee made in [2] towards VBB obfuscation for specific functionalities. In Section 3, we recapped their newly introduced knowledge assumption called KOALA and the resulting VBB secure obfuscator for the big subset functionality. Beside this, we presented the analysis made in [2] regarding the current state of the art for obfuscation schemes covering the pattern matching with wildcards functionality. This is summarized in Section 5.1 and Section 5.2. Finally, we present how Beullens and Wee improved this result by presenting their transformation of an obfuscator for the big subset functionality into an obfuscator for the pattern matching with wildcards functionality in Section 5.3.

References

[1] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany. doi:10.1007/3-540-44647-8_1. (Pages 1 and 3.)

[2] Ward Beullens and Hoeteck Wee. Obfuscating simple functionalities from knowledge assumptions. In Dongdai Lin and Kazue Sako, editors, *PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 254–283, Beijing, China, April 14–17, 2019. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-17259-6_9. (Pages iv, 2, 4, 5, 6, 7, 9, 10, 11, 12, and 13.)

[3] Allison Bishop, Lucas Kowalczyk, Tal Malkin, Valerio Pastro, Mariana Raykova, and Kevin Shi. A simple obfuscation scheme for pattern-matching with wildcards. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 731–752, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-96878-0_25. (Pages iv, 2, 7, 8, 9, 10, 11, and 12.)

[4] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 171–190, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press. doi:10.1109/FOCS.2015.20. (Page 1.)

[5] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273, Providence, RI, USA, March 28–30, 2011. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-19571-6_16. (Page 1.)

[6] Zvika Brakerski and Guy N. Rothblum. Obfuscating conjunctions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 416–434, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-40084-1_24. (Page 3.)

[7] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany. doi:10.1007/BFb0052255. (Pages 5 and 6.)

[8] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Heidelberg, Germany. doi:10.1007/3-540-46766-1_36. (Page 1.)

[9] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018,*

Part II, volume 10992 of *Lecture Notes in Computer Science*, pages 33–62, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-96881-0_2. (Page 1.)

[10] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press. doi:10.1109/FOCS.2013.13. (Page 1.)

[11] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. http://eprint.iacr.org/2010/556. (Page 1.)

[12] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany. doi:10.1007/3-540-69053-0_18. (Page 1.)