# On White-Box Cryptography and Obfuscation*

Amitabh Saxena[a] and Brecht Wyseur[b]

[a]DISI, University of Trento, Italy
amitabh@disi.unitn.it

[b]ESAT/SCD – COSIC/IBBT, K.U. Leuven, Belgium
brecht.wyseur@esat.kuleuven.be

**Abstract.** We study the relationship between obfuscation and white-box cryptography. We capture the requirements of any white-box primitive using a *White-Box Property (WBP)* and give some negative/positive results. Loosely speaking, the WBP is defined for some scheme and a security notion (we call the pair a *specification*), and implies that w.r.t. the specification, an obfuscation does not leak any "useful" information, even though it may leak some "useless" non-black-box information.

Our main result is a negative one - for most interesting programs, an obfuscation (under *any* definition) cannot satisfy the WBP for every specification in which the program may be present. To do this, we define a *Universal White-Box Property (UWBP)*, which if satisfied, would imply that under *whatever* specification we conceive, the WBP is satisfied. We then show that for every non-approximately-learnable family, there exist certain (contrived) specifications for which the WBP (and thus, the UWBP) fails.

On the positive side, we show that there exists an obfuscator for a non-approximately-learnable family that achieves the WBP for a certain specification. Furthermore, there exists an obfuscator for a non-learnable (but approximately-learnable) family that achieves the UWBP.

Our results can also be viewed as formalizing the distinction between "useful" and "useless" non-black-box information.

**Keywords.** White-Box Cryptography, Obfuscation, Security Notions

## 1 Introduction

Informally, an obfuscator $O$ is a probabilistic compiler that transforms a program $P$ into $O(P)$, an executable implementation of $P$ which hides certain functional characteristics of $P$. Starting from the seminal work of Barak *et al.* [2], several definitions for obfuscators have been proposed [20,21,24], each one based on some sort of *virtual black-box property (VBBP)*. Loosely speaking, the VBBP requires that *whatever we could do using the obfuscated program, we could also have done using black-box access to the original program.* The notion of "whatever" can be captured using several formalisms. The following are the common ones (in decreasing order of generality):

1. Computing something that is indistinguishable from the obfuscation [2,20,21,24].
2. Computing some function [2].
3. Computing some predicate [2].

* This work was partly supported by funds from the European Commission through the IST Program under Contract IST-021186-2 for the RE-TRUST project and in part by the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy).

### 1.1 White-box Cryptography

*White-box cryptography (WBC)*, which requires that *some given scheme must remain secure even if the adversary is given "white-box access" to a functionality instead of just black-box access*, is an active field of research. Informally, white-box access implies that the adversary is given an executable implementation of the algorithm that was used inside the black-box [11,12]. Existing notions of WBC only deal with the encryption algorithm of symmetric block-ciphers. In this work, we generalize this intuition to any cryptographic primitive. For instance, we can use WBC to convert a MAC into a signature scheme by white-boxing the verification algorithm.

White-Box Security. The (black-box) security of any primitive is captured using a *security notion* (e.g., IND-CPA) where the adversary is given black-box access to some functionality (e.g., encryption), and a white-box implementation can be required to satisfy that security notion when the adversary is given access to a *white-boxed* version of the functionality.

### 1.2 Motivation

One way to realize WBC is to obfuscate (using an obfuscator) the executable code of the algorithm and hope that the adversary cannot use it in a non-black-box manner. What we would like is, given an obfuscator satisfying some definition, a white-box implementation can be proved secure under some security notion. Furthermore, if a scheme is required to satisfy several security notions simultaneously (Authenticated Encryption (AE) [4] and the Obfuscated Virtual Machine (OVM) of [19] are two such examples, where both confidentiality and integrity needs to be satisfied), we would like the obfuscation to ensure that all the security notions are satisfied in the white-box variant if they are satisfied in the black-box variant. However, it is still not fully clear if any of the existing definitions of obfuscators can be used to achieve these goals. Hence, a natural question is:

> *Given an obfuscator satisfying the virtual black-box property for a program $P$ (in some sense), and some scheme that is secure when the adversary is given black-box access to $P$, can it be proved (without additional assumptions) that the scheme remains secure when the adversary is also given access to the obfuscated program $O(P)$?*

### 1.3 Our Contribution

1. In this paper we answer the above question in the negative - we show that under *whatever* definition of obfuscation we use, the answer to the above question is, in general, no. To do this, we first define the objective(s) of a white-box primitive, which we formalize using a *white-box property (WBP)*. Our main observation is that when considering obfuscation of most programs $P$, we must also take into account the scheme plus the security notion (i.e., the specification) in which $P$ is used. Furthermore, we show that for most programs $P$, there cannot exist an obfuscator that satisfies the WBP for all specifications in which $P$ might be present. To do this, we define a *universal white-box property (UWBP)* which, if satisfied, would imply that in whatever specification $P$ might be present, the obfuscated program $O(P)$ will not leak any "useful" information. We then show that for every non-approximately-learnable program $P$, there exists some specification in which the obfuscation leaks useful information, thereby failing the UWBP.
2. On the positive side, we have the following two results.
   (a) We show that under reasonable computational assumptions, there exists an obfuscator that satisfies the WBP w.r.t. some meaningful specification for a non-approximately-learnable program $P$.

(b) We show that there exist obfuscators that satisfy UWBP for a program that is non-learnable but approximately learnable.

## 2 Related Work

Practical white-box implementations of DES and AES encryption algorithms were proposed in [11,12]. However, no definitions of obfuscation were given, neither were there any proofs of security. With their subsequent cryptanalysis [6,17,25], it remains an open question whether or not such white-box implementations exist.

The notion of code-obfuscation was first given by Hada in [18], which introduced the concept of virtual black-box property (VBBP) using computational indistinguishability. In [2], Barak *et al.* defined obfuscation using the weaker predicate-based VBBP and showed that there exist unobfuscatable function families under their definition. Goldwasser and Kalai [13] extend the impossibility results of [2] w.r.t. auxiliary inputs.

On the positive side, there have been several results too. For instance, Lynn *et al.* show in [22] how to obfuscate point functions in the random oracle model. Wee in [24] showed how to obfuscate point functions without random oracles. Hohenberger *et al.* [21] used a stronger notion of obfuscation (average-case secure obfuscation) and showed how it can be used to prove the security of re-encryption functionality in a weak security model (i.e., IND-CPA). They also presented a re-encryption scheme under bilinear complexity assumptions. Hofheinz *et al.* [20] discuss a related notion of obfuscation and show that IND-CPA encryption and point functions can be securely obfuscated in their definition. Goldwasser and Rothblum [16] define the notion of "best-possible obfuscation" in order to give a qualitative measure of information leakage by an obfuscation (however, they do not differentiate between "useful" and "useless" information). Recently, Canetti and Dakdouk [10] give an obfuscator for point functions with multi-bit output for use in primitives called "digital lockers". Finally, Herzberg *et al.* [19] introduce the concept of *White-Box Remote Program Execution (WBRPE)* in order to give a meaningful notion of "software hardening" for all programs and avoid the negative results of [2].

However, till date, there has not been much work done on the relationship between arbitrary white-box primitives and obfuscation. This paper is intended to fill this gap.

## 3 Preliminaries

Denote by $\mathbb{P}$ the set of all positive polynomials and by $\mathbb{TM}$ the set of all Turing Machines (TMs). All TMs considered in this paper are deterministic (a probabilistic TM is simply a deterministic TM with randomness on the input tape). A mapping $f : x \ni \mathbb{N} \mapsto f(x) \in \mathbb{R}$ is negligible in $x$ (written $f(x) \leq negl(x)$) if $\forall p \in \mathbb{P}, \exists x' \in \mathbb{N}, \forall x > x' : f(x) < 1/p(x)$.

For simplicity, we define the input-space of arbitrary TMs to be $\{0,1\}^*$, the set of all strings. If, however, the input-space of a TM is well defined and efficiently samplable (for instance, the strings should be of a particular encoding), then we implicitly imply that the inputs are chosen from the input-space sampled using a string from $\{0,1\}^*$. All our definitions and results apply in this extended setting without any loss of generality.

**Definition 1.** *In the following, unless otherwise stated, a TM is assumed to have only one input tape.*

1. **(Equality of TMs.)** $X, Y \in \mathbb{TM}$ *are equal (written* $X = Y$*) if* $\forall a : X(a) = Y(a)$
2. **(Polynomial TM.)** $X \in \mathbb{TM}$ *is a Polynomial TM (PTM) if there exists* $p \in \mathbb{P}$ *s.t.* $\forall a : X(a)$ *halts in at most* $p(|a|)$ *steps. Denote the set of all PTMs by* $\mathbb{PTM}$.

3. **(PPT Algorithms.)** *A PPT algorithm (such as an adversary or an obfuscator) is a PTM with an unknown source of randomness input via an additional random tape. We denote the set of PPT algorithms by* $\mathbb{PPT}$. *The running time of a PPT algorithm must be polynomial in the length of the known inputs.*
4. **(TM Family.)** *A TM Family (TMF) is a TM having two input tapes: a* key tape *and a* standard input tape. *We denote by* $\mathbb{TMF}$ *the set of all TMFs. Let* $Q \in \mathbb{TMF}$. *Then:*
   (a) *The symbol* $Q^q$ *indicates that the key tape of* $Q$ *contains string* $q$.
   (b) *We denote by* $\mathcal{K}_Q$ *the key-space (valid strings for the key tape) of* $Q$.
   (c) *Let* $q \in \mathcal{K}_Q$. *In our model, the input-space (valid strings for the standard input tape) of* $Q^q$ *is fully defined by the parameter* $|q|$. *We denote this space by* $\mathcal{I}_{Q,|q|}$. *Furthermore the following must hold:*
$$\exists p \in \mathbb{P}, \forall q \in \mathcal{K}_Q, \forall x \in \mathcal{I}_{Q,|q|} : |x| = p(|q|).$$
5. **(Polynomial TM Family.)** $Q \in \mathbb{TMF}$ *is a* Polynomial *TMF (PTMF) if there exists* $p \in \mathbb{P}$ *such that* $\forall q \in \mathcal{K}_Q, \forall a \in \mathcal{I}_{Q,|q|} : Q^q(a)$ *halts in at most* $p(|q|)$ *steps. We denote the set of all PTMFs by* $\mathbb{PTMF}$.
6. **(Learnable Family.)** $Q \in \mathbb{TMF}$ *is learnable if* $\exists (L, p) \in \mathbb{PPT} \times \mathbb{P}$ *s.t.*
$$\forall k : \Pr[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q; X \leftarrow L^{Q^q}(1^{|q|}, Q) : X = Q^q] \geq 1/p(k)$$
   *(the probability taken over the coin tosses of* $L$*) and:*
   (a) $\forall a$ *: if* $Q^q(a)$ *halts after* $t$ *steps then* $X(a)$ *halts after at most* $p(t)$ *steps.*[1]
   (b) $|X| \leq p(|q|)$.
   $L$ *is called the learner for* $Q$. *We denote the set of all learnable families by* $\mathbb{LF}$.
7. **(Approx. Learnable Family.)** $Q \in \mathbb{TMF}$ *is approx. learnable if* $\exists (L, p) \in \mathbb{PPT} \times \mathbb{P}$ *s.t.*
$$\forall k : \Pr[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q; a \xleftarrow{R} \mathcal{I}_{Q,k}; X \leftarrow L^{Q^q}(1^{|q|}, Q) : X(a) = Q^q(a)] \geq 1/p(k)$$
   *(the probability taken over the coin tosses of* $L$*), and:*
   (a) $\forall a$ *: if* $Q^q(a)$ *halts after* $t$ *steps then* $X(a)$ *halts after at most* $p(t)$ *steps.*
   (b) $|X| \leq p(|q|)$.
   *We denote the set of all approx. learnable families by* $\mathbb{ALF}$.

**Lemma 1.** *If* $Q_1 \in \mathbb{PTMF} \backslash (\mathbb{A})\mathbb{LF}$, *then the following holds:*
$$\left[\exists (Q_2, p) \in \mathbb{PTMF} \times \mathbb{P}, \forall q_1 \in \mathcal{K}_{Q_1}, \exists q_2 \in \mathcal{K}_{Q_2} : Q_1^{q_1} = Q_2^{q_2} \wedge |q_2| \leq p(|q_1|)\right] \rightarrow Q_2 \notin (\mathbb{A})\mathbb{LF}.$$
*The symbol* $(\mathbb{A})$ *indicates that* $\mathbb{A}$ *is optional in the above statement.*

*Proof.* Assume for contradiction that for any given $Q_1 \in \mathbb{PTMF}$ that is not learnable, there exists some $(Q_2, p) \in \mathbb{PTMF} \times poly$ such that the LHS of the above implication is satisfied but RHS is not. Let $L_1$ and $L_2$ be the learners for $Q_1$ and $Q_2$ respectively. $L_1$ runs $L_2$ using its own oracle to answer $L_2$'s queries. If $Q_2$ is learnable, then $L_2$ will output $X_2 \approx Q_2^{q_2}$ in a polynomial (of $|q_2|$) number of steps, which is a polynomial function of $|q_1|$ by assumption, a contradiction. □

[1] This condition is to prevent an exponential time learner from becoming polynomial time by hard-wiring the learning algorithm and queries/responses inside $X$.

## 4 Obfuscators

In this work, we only consider obfuscation of PTMFs with a uniformly selected key, and not of a single PTM. As is common in cryptography, we define the functionality of the obfuscator using a *correctness* property and the security using a *soundness* property. In contrast to existing works, however, we define an obfuscator using only the correctness property. This is to consider different notions of "white-box" security (which might be unrelated to soundness) and still be able to use the word "obfuscator" in a formal sense.

### 4.1 Obfuscator (Correctness)

**Definition 2.** *A randomized algorithm* $O : \mathbb{PTMF} \times \{0,1\}^* \mapsto \mathbb{TM}$ *satisfies* correctness *for* $Q \in \mathbb{PTMF}$ *if the following two properties are satisfied:*

1. ***Approx. functionality:***

$$\forall q \in \mathcal{K}_Q, \forall a \in \mathcal{I}_{Q,|q|} : \Pr[O(Q,q)(a) \neq Q^q(a)] \leq negl(|q|),$$

   *the probability taken over the coin tosses of* $O$.[2]
2. ***Polynomial slowdown and expansion:*** *There exists* $p \in \mathbb{P}$ *s.t.*

$$\forall q \in \mathcal{K}_Q : |O(Q,q)| \leq p(|q|),$$

   *and* $\forall a$, *if* $Q^q(a)$ *halts in* $t$ *steps then* $O(Q,q)(a)$ *halts in at most* $p(t)$ *steps.*

*We say the* $O$ *is efficient if* $O \in \mathbb{PPT}$.

*If* $O$ *satisfies correctness for* $Q$, *we say that* $O$ *is an* obfuscator *for* $Q$.

### 4.2 Obfuscator (Soundness)

Over recent years, several definitions of soundness have been proposed, all based on some sort of *Virtual Black-Box Property (VBBP)* [2,20,21,22,24]. Let $Q \in \mathbb{PTMF}$ and let $q \in \{0,1\}^*$. Loosely speaking, the VBBP requires that whatever information about $q$ a PPT adversary computes given the obfuscation $O(Q,q)$, a PPT simulator could also have computed using only black-box access to $Q^q$. All existing notions of VBBP can be classified into one of two broad categories. At one extreme (the weakest) are the predicate-based definitions, where the adversary and the simulator are required to compute some predicate of $q$. At the other extreme (the strongest) are definitions based on computational indistinguishability, where the simulator is required to output something that is indistinguishable from $O(Q,q)$. We define these two notions below. Our definitions are based on that of [13], where an auxiliary input is also considered.

**Definition 3.** *An obfuscator* $O$ *for* $Q \in \mathbb{PTMF}$ *satisfies* soundness *for* $Q$ *if at least one of the properties given below is satisfied.*

[2] For now, we consider the functionality of $Q$ only in a deterministic sense. That is, we do not consider the notion of obfuscation of "probabilistic functions" (used, for example, in [20,21]). However, our negative results (presented in §6.1) also apply to probabilistic functions using an appropriately defined notion of *probabilistic PTMFs (PPTMFs)* (and a corresponding notion of approx. functionality for PPTMFs). This aspect will be further discussed in §7.

1. ***Predicate Virtual black-box property (PVBBP):*** *Let $\pi$ be any efficiently verifiable predicate on $\mathcal{K}_Q$. $O$ satisfies PVBBP for $Q$ if*

$$\forall (A,p) \in \mathbb{PPT} \times \mathbb{P}, \exists (S,k') \in \mathbb{PPT} \times \mathbb{N}, \forall k > k' : Adv^{pvbbp}_{A,S,O,Q}(k) \leq negl(k),$$

*where*

$$Adv^{pvbbp}_{A,S,O,Q}(k) = \max_{\pi} \max_{z \in \{0,1\}^{p(k)}} \left| \begin{array}{c} \Pr[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q : A^{Q^q}(1^k, O(Q,q), z) = \pi(q)] \\ - \Pr[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q : S^{Q^q}(1^k, z) = \pi(q)] \end{array} \right|,$$

*the probability taken over the coin tosses of $O, A, S$.*[3]

2. ***Computational Indistinguishability (IND):*** *$O$ satisfies IND for $Q$ if*

$$\forall (A,p) \in \mathbb{PPT} \times \mathbb{P}, \exists (S,k') \in \mathbb{PPT} \times \mathbb{N}, \forall k > k' : Adv^{ind}_{A,S,O,Q}(k) \leq negl(k),$$

*where*

$$Adv^{ind}_{A,S,O,Q}(k) = \max_{z \in \{0,1\}^{p(k)}} \left| \begin{array}{c} \Pr[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q : A^{Q^q}(1^k, O(Q,q), z) = 1] \\ - \Pr[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q : A^{Q^q}(1^k, S^{Q^q}(1^k, z), z) = 1] \end{array} \right|,$$

*the probability taken over the coin tosses of $O, A, S$.*

*Depending on the property satisfied, we call it IND-soundness or PVBBP-soundness (note that the former implies the latter).*

It has been noted (but never proved) in several papers (e.g., [2,21]) that the PVBBP is too weak for practical purposes. Furthermore, it has been noted that the IND-soundness is too strong to be satisfied in practice [21,24]. In fact, it is easy to prove:

**Proposition 1.** *If there exists an obfuscator satisfying IND-soundness for some $Q \in \mathbb{PTMF}$ then $Q \in \mathbb{ALF}$.*[4]

Nevertheless, it is conceivable that a definition of soundness can be formulated falling somewhere between the two extremes, which is neither too weak nor too strong, and can be used for proving security of arbitrary white-box primitives. We show this is not the case. Specifically, we show that, under *whatever* definition of soundness we use, for every family $Q \notin \mathbb{ALF}$, there exist (contrived) specifications for which white-box security fails but the corresponding black-box construction is secure.[5]

## 5 White-box Cryptography (WBC)

In this section, we formalize the notion of WBC by defining a *white-box property (WBP)*. A key concept of our model is the notion of a (cryptographic) *specification*. Informally, a specification is a self-contained description (in some formal language) of a cryptographic scheme (such as RSA-OAEP) along with a corresponding security notion (such as IND-CPA). We follow the basic principles of various "game-based" approaches [3,5,14,15] where a security notion is captured using an interactive game between an adversary and a challenger. In our model, the role of the challenger is played by an *experiment* and the corresponding game is called a *simulation*. We denote by $\mathbb{SPEC}$ the set of all specifications.

[3] The definition of PVBBP given here is slightly weaker than the one used in [2] because they require this property to hold for every $q$, while we require it to hold only for uniformly selected $q$.

[4] This result does not hold if the definition of approx. functionality in correctness is extended to probabilistic functions. See §7 for details.

[5] In related work, the authors of [20] show that a slightly different notion of the IND property - one based on probabilistic functions - is insufficient for proving the white-box IND-CCA2-security of encryption schemes, even if white-box IND-CPA is satisfied. Our results are more general because they apply to every $Q \notin \mathbb{ALF}$.

### 5.1 Black-Box Simulation

Let *spec* $\in \mathbb{SPEC}$ denote the specification of some scheme (e.g., "IND-CPA security notion for symmetric encryption scheme X"). Every such *spec* defines a *Black-box simulation* (or simply simulation) between an *experiment* and an *adversary*.

**Experiment.** The experiment for *spec*, written $\mathtt{Expt}^{spec}$ is a TM having six tapes: (1) a read-only *experiment-input* tape, (2) a writable *adversary-input* tape, (3) a read-only *query-input* tape, (4) a writable *query-response* tape, (5) a read-only *adversary-output* tape, and (6) a writable *experiment-output* tape

**Adversary.** The adversary $A \in \mathbb{PPT}$ is an algorithm having four tapes (along with an unknown source of randomness via a random input tape): (1) a read-only *adversary-input* tape, (2) a writable *query-input* tape, (3) a read-only *query-response* tape, and (4) a writable *adversary-output* tape

**Simulation.** A simulation is an interactive protocol between the experiment and the adversary when their tapes coincide, and is started by invoking the experiment via the experiment-input tape.

- The experiment-input tape contains two inputs: (1) a string of $k$ 1s, where $k$ is a security parameter, and (2) a random string $r$ of $p_{in}(k)$ bits for some $p_{in} \in \mathbb{P}$.
- During the simulation, the experiment and the adversary interact using the common tapes. The adversary terminates after writing a string on the adversary-output tape.
- The simulation ends when the experiment writes a *result* on the experiment-output tape.
- We require the result to be either 0 (indicating $A$ lost) or 1 ($A$ won).
- We denote by $\mathtt{Expt}_A^{spec}$ the simulation, and by $\mathtt{Expt}_A^{spec}(1^k, r)$ the result when the experiment-input tape contains $(1^k, r)$.
- Every experiment must be based on the following template:

  1. $\mathtt{Expt}_A^{spec}(1^k, r)$ :
  2.     `/*` Description of $n$ families $Q_1, Q_2, \ldots, Q_n \in \mathbb{PTMF}$ `*/`
  3.     `/*` Description of PTM $f : \{0,1\}^{p_{in}(k)} \mapsto \times_{i=1}^{n} \mathcal{K}_{Q_i}$ `*/`
  4.     $(q_1, q_2, \ldots q_n) \leftarrow f(r)$
  5.     $s \leftarrow A^{Q_1^{q_1}, Q_2^{q_2}, \ldots, Q_n^{q_n}}(1^k, spec)$
  6.     `If (win(`$r, QuerySet, s$`)) output 1 else output 0`

  The following discussion is based on the above template.

  - We do not allow the oracles used by $A$ to maintain state between successive queries[6] and assume that a query takes one unit time irrespective of the amount of computation involved.
  - We require that at any instant $A$ can query at most one oracle.
  - We require that if $r$ is uniformly distributed then so are the keys $q_i$ $(1 \le i \le n)$.
  - The run-time of $A$ is upper-bounded by $p_{run}(k)$ steps for some $p_{run} \in \mathbb{P}$ (specified in *spec*).
  - *QuerySet* is a set representing the queries made by $A$ during the simulation. Each element $j$ of this set is an ordered tuple of the type

    $$(\mathbf{t}_j, \mathbf{i}_j, \mathbf{in}_j, \mathbf{out}_j) \in \mathbb{N} \times \{1, 2, \ldots, n\} \times \{0,1\}^* \times \{0,1\}^*,$$

    indicating respectively, the time, oracle number, input, and the output of each query.
  - `win` is (the PTM description of) an efficiently computable predicate on $(r, QuerySet, s)$.
  - We say that a family $Q \in spec$ if $Q \in \{Q_i\}_{1 \le i \le n}$.

---

[6] If state is to be maintained, for instance, each response to the query must use different randomness (and so a query counter must be maintained), then we first assume that adversary can make at most $x$ queries to this oracle, and we replicate the oracle $x$ times, each with different randomness. In the winning condition, we test that each such oracle was queried at most one time.

**Definition 4.** *We define*

$$Adv_A^{spec}(k) = \Pr[r \xleftarrow{R} \{0,1\}^{p_{in}(k)} : \textbf{\textit{Expt}}_A^{spec}(1^k, r) = 1],$$

*the probability taken over the coin tosses of $A$.*

**Definition 5.** *(**Obfuscatable family**) For any PTMF $Q_i \in spec$, define*

$$QuerySet_i = \{(\mathbf{t}_j, \mathbf{i}_j, \mathbf{in}_j, \mathbf{out}_j) | (\mathbf{t}_j, \mathbf{i}_j, \mathbf{in}_j, \mathbf{out}_j) \in QuerySet \wedge \mathbf{i}_j \neq i\}$$

*We say that $Q_i$ is* obfuscatable *in spec (written $Q_i \in_{obf} spec$) if*

$$\forall r, QuerySet, s : \mathtt{win}(r, QuerySet, s) = \mathtt{win}(r, QuerySet_i, s).$$

*(In other words, $Q_i \in spec$ is obfuscatable if every element of QuerySet corresponding to oracle $Q_i^{q_i}$ can be removed without affecting the* `win` *predicate).*

*Remark 1.* We claim that it is meaningless to talk about white-box security of specifications where the PTMF to be white-boxed is not-obfuscatable, since it is impossible to keep track of "queries" made by an adversary to an obfuscated program. As an example, it is meaningless to talk about obfuscating the decryption oracle of an encryption scheme (or the 'signing' oracle of a MAC scheme).

An example of a specification for the IND-CCA2 notion of some symmetric encryption scheme is given in Appendix A.

### 5.2 White-box Simulation

Let $\mathtt{Expt}_A^{spec}$ capture the security of some $spec \in \mathbb{SPEC}$ (using the template of §5.1). Let $O$ be an obfuscator for some $Q_i$ with $1 \leq i \leq n$ such that $Q_i \in_{obf} spec$. Define the corresponding white-box experiment for $(spec, Q_i)$ as follows:

1. $\mathtt{ExptWB}_{A,O}^{spec,Q_i}(1^k, r)$ :
2.     `/*` Description of $n$ families $Q_1, Q_2, \ldots, Q_n \in \mathbb{PTMF}$ `*/`
3.     `/*` Description of PPT $f : \{0,1\}^{p_{in}(k)} \mapsto \times_{i=1}^{n} \mathcal{K}_{Q_i}$ `*/`
4.     $(q_1, q_2, \ldots q_n) \leftarrow f(r)$
5.     $s \leftarrow A^{Q_1^{q_1}, Q_2^{q_2}, \ldots Q_n^{q_n}}(1^k, spec, i, O(Q_i, q_i))$
6.     `If` $(\mathtt{win}(r, QuerySet, s))$ `output 1 else output 0`

– As before, we bound the running time of $\mathtt{ExptWB}_{A,O}^{spec,Q_i}$ to $p_{run}(k)$ steps.

**Definition 6.** *We define*

$$Advwb_{A,O}^{spec,Q_i}(k) = \Pr[r \xleftarrow{R} \{0,1\}^{p_{in}(k)} : \textbf{\textit{ExptWB}}_{A,O}^{spec,Q_i}(1^k, r) = 1],$$

*the probability taken over the coin tosses of $A, O$.*

**Definition 7.** *(**White-box Property (WBP)**) Let $O$ be an obfuscator for $Q_i \in \mathbb{PTMF}$ and let spec be such that $Q_i \in_{obf}$ spec. We say that $O$ satisfies WBP for $(Q_i, spec)$ if the following holds:*

$$\min_{A \in \mathbb{PPT}} \left| Advwb_{A,O}^{spec,Q_i}(k) - Adv_A^{spec}(k) \right| \leq negl(|k|).$$

*The term $\min_{A \in \mathbb{PPT}} \left| Advwb_{A,O}^{spec,Q_i}(k) - Adv_A^{spec}(k) \right|$ is called the* **white-box advantage** of $(O, Q_i, spec)$, *and serves a measure of "useful information leakage" by the obfuscation.*

**Definition 8.** *(**Universal White-box Property (UWBP)**) Let $O$ be an obfuscator for $Q \in \mathbb{PTMF}$. We say that $O$ satisfies UWBP for $Q$ if for every spec $\in \mathbb{SPEC}$ with $Q \in_{obf}$ spec, $O$ satisfies white-box property for $(Q, spec)$.*

# 6 WBC and Obfuscation

In this section we give some useful relationships between obfuscators, WBP and UWBP.

## 6.1 Negative results

We note that Barak *et al.*'s impossibility results [2] also apply our definitions. In our model, their results can be interpreted as the following:

*There exists a pair* $(Q, spec) \in \mathbb{PTMF} \times \mathbb{SPEC}$ *with* $Q \in_{obf} spec$ *such that every obfuscator for* $Q$ *fails to satisfy WBP for* $(Q, spec)$.

In other words, there cannot exist a obfuscator that satisfies UWBP for every $Q$. However, their results do not rule out an obfuscator that satisfies the UWBP for some useful family $Q$. We show that even this is not possible unless $Q$ is at least approx. learnable.

**Result 1: No UWBP For "Interesting" Families.** (Informal) Obfuscators satisfying UWBP for "interesting" families do not exist. More formally,

**Theorem 1.** *For every family* $Q \in \mathbb{PTMF} \backslash \mathbb{ALF}$*, there exists a (contrived) spec* $\in \mathbb{SPEC}$ *such that* $Q \in_{obf} spec$ *but every obfuscator for* $Q$ *fails to satisfy the WBP for* $(Q, spec)$.

*Proof.* Let $Q \in \mathbb{PTMF} \backslash \mathbb{ALF}$. Consider $spec = find\text{-}q'$ captured below.

$\mathtt{Expt}_A^{find\text{-}q'}(1^k, r := \langle q, q', a \rangle)$ :
    Family $Q$(Key $q$, Input $X$) {
        /* Description of $Q$ (used as a black-box) */
    }
    Family $Q_1$(Key $q_1 := \langle q, q', a \rangle$, Input $Y$) {
        /* We assume $Y \in \mathbb{PTM}$ */
        If $(Y(a) = Q^q(a))$ output $q'$ else output $0$
        /* In the above, (the description of) $Q$ is used as a black-box */
        /* $Y$ is allowed to run for at most $p(|a|)$ steps (for some $p \in \mathbb{P}$) */
    }
    Function $f$(Input $r$) {
        Parse $r$ as $\langle q, q', a \rangle$
        /* we require that $a \in \mathcal{I}_{Q,|q|} \wedge |q'| = |q|$ */
        set $q_1 \leftarrow \langle q, q', a \rangle$
        output $q, q_1$
    }
    $q, q_1 \leftarrow f(r)$
    $s \leftarrow A^{Q^q,\, Q_1^{q_1}}(1^k, find\text{-}q')$
    If ($s = q'$ and at most one query to $Q_1^{q_1}$) output 1 else output 0

Observe that $Q \in_{obf} find\text{-}q'$. Since $Q \notin \mathbb{ALF}$, therefore by virtue of Definitions 1.7 and 2.1, for sufficiently large random $q, q', a$ (and thus, $k$), the following inequalities are guaranteed to hold:

$$\forall A \in \mathbb{PPT} : 0 \leq Adv_A^{find\text{-}q'}(k) < \alpha(k)$$

$$\forall A \in \mathbb{PPT} : 1 \geq Advwb_{A,O}^{find\text{-}q',Q}(k) \geq 1 - \beta(k),$$

where $\alpha, \beta$ are negligible functions. Hence, we have:

$$\min_{A \in \mathbb{PPT}} \left| Advwb_{A,O}^{find\text{-}q',Q}(k) - Adv_A^{find\text{-}q'}(k) \right| > 1 - \alpha(k) - \beta(k),$$

which is non-negligible in $k$. This proves the theorem. $\square$

*Remark 2.* The above result applies because the approx. functionality requirement for obfuscators (in the correctness definition of §2) is defined only for PTMFs considered as deterministic. What about the extended definitions (such as in [20,21]) which allow probabilistic families? It turns out that a similar technique can be used for probabilistic families using appropriately extended definitions. This aspect is discussed in §7.

*Remark 3.* Although we define $\mathbb{ALF}$ to be the set of families which can be approximately-learned with a non-negligible advantage (which is quite broad), we note that the above result can be further strengthened by narrowing down the definition of $\mathbb{ALF}$ to only families that can be approximately-learned with an overwhelming advantage.

Our next result deals with multiple obfuscations.

**Result 2: Simultaneous Obfuscation May Be Insecure.** (Informal) Simultaneous obfuscation of two families may be insecure even if obfuscation of each family alone is secure. We give a definition before stating this formally.

**Definition 9.** *(Multiple obfuscations) Let spec $\in \mathbb{SPEC}$ be the specification defined by $\texttt{Expt}_A^{spec}$ using the above template. Let $Q_i, Q_j \in_{obf}$ spec for some $1 \leq i, j \leq n$. Let $O$ be an obfuscator for $Q_i, Q_j$. Extend the white-box simulation $\texttt{ExptWB}_{A,O}^{spec,i}$ of §5 by defining a corresponding simulation $\texttt{ExptWB}_{A,O}^{spec,i,j}$ in which $A$ gets as input in Step 5, the tuple $(1^k, i, j, O(Q_i, q_i), O(Q_j, q_j))$. Finally define,*

$$Advwb_{A,O}^{spec,i,j}(k) = \Pr[r \xleftarrow{R} \{0,1\}^{p_{in}(k)} : \texttt{ExptWB}_{A,O}^{spec,i,j}(1^k, r) = 1],$$

*the probability taken over the coin tosses of $A, O$.*

*We say that $O$ satisfies WBP for $((Q_i, Q_j), spec)$ if the following holds:*

$$\min_{A \in \mathbb{PPT}} \left| Advwb_{A,O}^{spec,i,j}(k) - Adv_A^{spec}(k) \right| \leq negl(|k|).$$

**Theorem 2.** *Let $Q_i, Q_j \in \mathbb{PTMF} \backslash \mathbb{ALF}$. Then there exists a spec $\in \mathbb{SPEC}$ with $Q_i, Q_j \in_{obf}$ spec such that even if there exists an obfuscator for $Q_1, Q_2$ satisfying WBP for $(Q_i, spec)$ and $(Q_j, spec)$, every obfuscator fails to satisfy WBP for $((Q_i, Q_j), spec)$*

The proof is similar to the proof of Theorem 1.

### 6.2 Positive Results

Although the above results rule out the possibility of obfuscators satisfying UWBP for most non-trivial families, they do not imply that a meaningful definition of security for white-box cryptography cannot exist. In fact, any asymmetric encryption scheme can be considered as a white-boxed version of the corresponding symmetric scheme (where the encryption key is also secret). We use this observation as a starting point of our first positive result. A similar observation was used in the positive results of [20].

**Result 3: WBP For "Useful" Families.** (informal) There exists a non-approx. learnable family (in fact many), and an obfuscator that satisfies WBP for that family under some useful specification. This is stated formally in Theorem 3.

**Theorem 3.** *Under standard computational assumptions, there exists a pair $(Q, spec) \in \mathbb{PTMF}\backslash\mathbb{ALF} \times \mathbb{SPEC}$ with with $Q \in_{obf}$ spec and an efficient obfuscator $O$ for $Q$ satisfying WBP for $(Q, spec)$.*

*Proof.* We prove this using construction. We will use an encryption scheme based on the BF-IBE scheme [7]. First we describe a primitive known as a bilinear pairing. Let $G_1$ and $G_2$ be two cyclic multiplicative groups both of prime order $w$ such that computing discrete logarithms in $G_1$ and $G_2$ is intractable. A bilinear pairing is a map $\hat{e} : G_1 \times G_1 \mapsto G_2$ that satisfies the following properties [7,8,9].

1. *Bilinearity*: $\hat{e}(a^x, b^y) = \hat{e}(a, b)^{xy}\ \forall a, b \in G_1$ and $x, y \in \mathbb{Z}_w$.
2. *Non-degeneracy*: If $g$ is a generator of $G_1$ then $\hat{e}(g, g)$ is a generator of $G_2$.
3. *Computability*: The map $\hat{e}$ is efficiently computable.

Define a *symmetric* encryption scheme $\mathcal{E} = (G, E, D)$ as follows.

1. *Key Generation (G):* Let $\hat{e} : G_1 \times G_1 \mapsto G_2$ be a bilinear pairing over cyclic multiplicative groups as defined above (such maps are known to exist). Let $|G_1| = |G_2| = w$ (prime) such that $\lfloor \log_2(w) \rfloor = l$. Pick random $g \xleftarrow{R} G_1 \backslash \{1\}$ and define $\mathcal{H} : G_2 \mapsto \{0,1\}^l$ to be a hash function. Finally pick $x \xleftarrow{R} G_1$ and define $key = \langle \hat{e}, G_1, G_2, w, g, \mathcal{H}, x \rangle$. The encryption/decryption key is $key$.
2. *Encryption (E):* The encryption family $E$ using key $key$ is defined as follows. Parse $key$ as $\langle \hat{e}, G_1, G_2, w, g, \mathcal{H}, x \rangle$. Let $m \in \{0,1\}^l$ be a message and $\alpha \in \mathbb{Z}_w$ be a random string. Set $(c_1, c_2) \leftarrow E^{key}(x, \alpha)$, where:
$$E^{key} : \{0,1\}^l \times \mathbb{Z}_w \ni (m, \alpha) \mapsto (\mathcal{H}(\hat{e}(x^\alpha, g)) \oplus m, g^\alpha) \in \{0,1\}^l \times G_1.$$
3. *Decryption (D):* The decryption family $D$ using key $key$ is defined as follows. Parse $key$ as $\langle \hat{e}, G_1, G_2, w, g, \mathcal{H}, x \rangle$ and compute $m = D^{key}(c_1, c_2)$, where:
$$D^{key} : \{0,1\}^l \times G_1 \ni (c_1, c_2) \mapsto \mathcal{H}(\hat{e}(c_2, x)) \oplus c_1 \in \{0,1\}^l.$$
It can be verified that $D^{key}(E^{key}(m, \alpha)) = m$ for valid values of $(m, \alpha)$

The scheme can be proven to be CPA secure if $\mathcal{H}$ is a random oracle and $w$ is sufficiently large. We construct an obfuscation of the $E^{key}$ oracle that converts $\mathcal{E}$ into a CPA secure *asymmetric* encryption scheme under a computational assumption.

**The obfuscator $O$:** The input is $(E, key)$.

1. Parse $key$ as $\langle \hat{e}, G_1, G_2, w, g, \mathcal{H}, x \rangle$ and set $y \leftarrow \hat{e}(x, g) \in G_2$.
2. Set $key' \leftarrow \langle \hat{e}, G_1, G_2, w, g, \mathcal{H}, y \rangle$ and define family $F$ with key $key'$ as:
$$F^{key'} : \{0,1\}^l \times \mathbb{Z}_w \ni (m, \alpha) \mapsto (\mathcal{H}(y^\alpha) \oplus m, g^\alpha) \in \{0,1\}^l \times G_1,$$
where $key'$ is parsed as $\langle \hat{e}, G_1, G_2, w, g, \mathcal{H}, y \rangle$.
3. Output $F^{key'}$.

*Claim.* $O$ is an efficient obfuscator for $E$ satisfying WBP for $(E, spec)$, where $spec$ = "IND-CPA security of $\mathcal{E}$", assuming that the *bilinear Diffie-Hellman assumption* [7] holds in $(G_1, G_2)$ and $\mathcal{H}$ can be considered equivalent to a random oracle.

*Proof.* We refer the reader to Appendix A for the formal definition of IND-CPA security. (The IND-CPA game is a restricted version of the IND-CCA2 game given there, by adding the additional check "no queries to $\mathbf{D}^{key}$" to the `win` predicate.)

First note that the obfuscator satisfies correctness for $E$ because $F^{key'} = E^{key}$. The proof of the above claim follows from the security of the $\mathsf{BasicPUB}$ encryption scheme of [7]. $\square$

*Claim.* If $\mathcal{H}$ is a one-way hash function then $E \in \mathbb{PTMF} \backslash \mathbb{ALF}$.

*Proof.* Clearly, $F \in \mathbb{PTMF}$ and the following holds:

$$\exists p \in \mathbb{P}, \forall key \in \mathcal{K}_E, \exists key' \in \mathcal{K}_F : F^{key'} = E^{key} \wedge |key'| = |key| + p(|key|).$$

By virtue of Lemma 1, in order to prove that $E \notin \mathbb{ALF}$, it is sufficient to prove that $F \notin \mathbb{ALF}$. Finally, it can be proved that if $\mathcal{H}$ is a one-way hash function then indeed $F \notin \mathbb{ALF}$.[7] $\square$

This completes the proof of Theorem 3. $\square$

*Remark 4.* An interesting observation from Theorem 3 is that even though the obfuscator $O$ satisfies WBP for $(E, spec)$, it does not satisfy soundness for $E$ (under Definition 3). This indicates that the soundness property and WBP are in general independent of each other.

*Remark 5.* A reader might wonder why we used the specific encryption scheme in the proof Theorem 3, when we could have used just about any asymmetric scheme (such as RSA), or even the re-encryption scheme of [21]. We justify our choice with the following reasons:

1. *Why not RSA, El Gamal, etc?*
   (a) Textbook RSA does not enjoy the security notion of IND-CPA. Furthermore, even in RSA variants that are IND-CPA, it is impossible to prove $E \notin \mathbb{ALF}$ without relying on additional computational assumptions.
   (b) Encryption in El Gamal (and its variants) is learnable.
2. *Why not re-encryption scheme of [21]?*
   The obfuscator of [21] does not satisfy approx. functionality as we define. and so their scheme is unsuitable for the proof. (However, the scheme of [21] is the ideal candidate for an analogous example of §7.)

### 6.3 UWBP For Non-Trivial Families

Let $Q \in \mathbb{PTMF} \cap \mathbb{LF}$. Then it is easy to construct an obfuscator satisfying UWBP for $Q$ with a non-negligible probability (same as that of learning $Q$). We call such families *trivial.*

Although Result 1 rules out the possibility of an obfuscator satisfying UWBP for some $Q \in \mathbb{PTMF} \backslash \mathbb{ALF}$ (which includes most non-trivial families), it does not rule out the possibility of an obfuscator satisfying UWBP for some non-trivial family $Q \in \mathbb{PTMF} \cap \mathbb{ALF}$ (i.e., $Q \in \mathbb{PTMF} \cap \mathbb{ALF} \backslash \mathbb{LF}$). Our next positive result shows that, under reasonable assumptions, this is indeed the case.

[7] Note that for proving IND-CPA security, we need a stronger assumption on $\mathcal{H}$, namely that it is equivalent to a random oracle. However, for proving that $E \notin \mathbb{ALF}$, the assumption that $\mathcal{H}$ is a one-way hash function is sufficient.

**Result 4: UWBP for a non-trivial family.** (informal) There exists an obfuscator satisfying UWBP for a non-trivial (but contrived) family $Q$. Formally,

**Theorem 4.** *Under reasonable assumptions, there exists a family* $Q \in \mathbb{PTMF} \cap \mathbb{ALF} \backslash \mathbb{LF}$ *and an obfuscator* $O$ *for* $Q$ *that satisfies UWBP for* $Q$.

*Proof.* For simplicity, we prove the above result in the random oracle model. Then under the assumption that there exist hash functions equivalent to random oracles, our result can be lifted to the plain model.

Consider the family $Q$ defined below:

1. `Family` $Q($`Key` $q,$ `Input` $X)$ $\{$
2. `If Random-Oracle`$_{|q|}(q||X) = q$ `output` $1$ `else output` $0$
3. $\}$

Here, `Random-Oracle`$_{|q|}$ is a random oracle mapping arbitrary strings to $|q|$-bit strings. First note that indeed $Q \in \mathbb{PTMF} \cap \mathbb{ALF} \backslash \mathbb{LF}$. It can be proved that $\forall D \in \mathbb{PPT}$ (the distinguisher),

$$\left| \Pr[b \xleftarrow{R} \{0,1\}; q_0, q_1 \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q : D^{Q^{q_b}}(1^k, q_0, q_1) = b] - \frac{1}{2} \right| \leq negl(k), \tag{1}$$

the probability taken over the coin tosses of $D$. For any $k$, let $q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q$. Consider an obfuscator $O$ that takes in as input $(Q, q)$ and simply outputs a description of $Q^q$ as the obfuscation of $Q^q$. Let $spec \in \mathbb{SPEC}$ be such that $Q \in_{obf} spec$ but $O$ does not satisfy WBP for $(Q, spec)$ w.r.t. some adversary $A \in \mathbb{PPT}$ (that is, the white-box advantage of $(O, Q, spec)$ is non-negligible), then $A$ can be directly converted into a distinguisher $D$ such that Equation 1 does not hold, thereby arriving at a contradiction. □

## 7 The Case Of Probabilistic PTMFs

In this section, we consider probabilistic (i.e., randomized) functions based on the definitions of [20,21]. In contrast to conventional constructions of PTMFs (such as the encryption algorithm of the probabilistic encryption scheme in the proof of Theorem 3 and Appendix A), where randomness is considered as part of the input tape, the definitions of [20,21] consider randomness as part of the key tape.[8] We call a PTMF of the latter type, a probabilistic PTMFs (PPTMF).

Intuitively, a PPTMF is simply an ordinary PTMF $Q$ with part of the key used for randomness, so that two different keys are "equivalent" provided only their random bits are different.

Formally, a PPTMF is any pair of the type $(Q, \tau)$, where $Q \in \mathbb{PTMF}$ and $\tau$ is an equivalence relation on $\mathcal{K}_Q$ that partitions $\mathcal{K}_Q$ into equivalence classes, s.t.

$$\forall q_1, q_2 \in \mathcal{K}_Q : \tau(q_1, q_2) = 1 \iff \text{only the random bits of } q_1, q_2 \text{ are different.}$$

We denote the set of all PPTMFs by $\mathbb{PPTMF}$.

**Definition 10.** *In the following, let* $(Q, \tau) \in \mathbb{PPTMF}$.

1. *Let* $q \in \mathcal{K}_Q$. *Then:*

[8] In the construction of [21], the PTMF has additional randomness on the input tape (a.k.a. 're-randomization values', which are supplied by the adversary). We ignore this additional randomness in our discussion (adversary cannot be trusted to supply randomness), since we are focusing on the security of the *obfuscator of some given PTMF*, and not of the *specification in which the PTMF is used.*

– *For any* $(a, z) \in \mathcal{I}_{Q,|q|} \times \{0,1\}^*$, *we say that* $z$ *is* $\tau$-equal *to* $Q^q(a)$ *(written* $z =_\tau Q^q(a)$*) if*

$$\exists q' \in \mathcal{K}_Q : z = Q^{q'}(a) \wedge \tau(q, q') = 1.$$

– *For any* $X \in \mathbb{TM}$ *we say that* $X$ *is* $\tau$-equal *to* $Q^q$ *(written* $X =_\tau Q^q$*) if*

$$\forall a \in \mathcal{I}_{Q,|q|} : X(a) =_\tau Q^q(a).$$

2. *We define a* $\tau$-(approx.) learnable *family by replacing "=" with "$=_\tau$" in the definition of (approx.) learnable families. The following claim is easy to prove.*

   *Claim. If* $Q$ *is not* $\tau$*-approx. learnable then* $Q \notin \mathbb{ALF}$.

3. *Let* $O : \mathbb{PTMF} \times \{0,1\}^* \mapsto \mathbb{TM}$ *be a randomized algorithm. Then:*
   – $O$ *satisfies* $\tau$-approx. functionality *for* $Q$ *if*

$$\forall q \in \mathcal{K}_Q, \forall a \in \mathcal{I}_{Q,|q|} : \Pr[O(Q,q)(a) \neq_\tau Q^q(a)] \leq negl(|q|),$$

   *the probability taken over the coin tosses of* $O$.
   – $O$ *satisfies* $\tau$-correctness *for* $Q$ *if Definition 2 (of §4.1) holds when "approx. functionality" is replaced by "*$\tau$*-approx. functionality".*
   – $O$ *is a* $\tau$-obfuscator *for* $Q$ *if it satisfies* $\tau$*-correctness for* $Q$.
4. $Q$ *is* $\tau$-decidable *if there exist* $p, p' \in \mathbb{P}$ *and for every* $k \in \mathbb{N}$, *there exists an efficiently computable map*

$$\mathbf{f}_k : \{0,1\}^{p(k)} \mapsto \{0,1\}^k \cap \mathcal{K}_Q \times \mathbb{PTM},$$

   *such that for all* $(q, Z) \leftarrow \mathbf{f}_k(r)$, *the following holds:*
   – $\forall a, z \in \{0,1\}^* : Z(a, z) = 1 \iff z =_\tau Q^q(a)$.
   – *If* $r$ *is uniformly distributed then so is* $q$.
   – $|Z| \leq p'(k)$.

We claim that in any meaningful PPTMF construction $(Q, \tau)$, the family $Q$ must be $\tau$-decidable (this is true for the constructions of [20,21]). Theorem 5 below is the equivalent of Theorem 1 for PPTMFs. The proof follows directly after replacing "=" with "$=_\tau$" in the proof of Theorem 1.

**Theorem 5.** *For every* $(Q, \tau) \in \mathbb{PPTMF}$ *with* $Q$ $\tau$*-decidable but not* $\tau$*-approx. learnable, there exists spec* $\in \mathbb{SPEC}$ *such that* $Q \in_{obf}$ *spec but every* $\tau$*-obfuscator for* $Q$ *fails to satisfy the WBP for* $(Q, spec)$.

### 7.1 An Open Question: WBP and Soundness

Let $((Q, \tau), spec) \in \mathbb{PPTMF} \times \mathbb{SPEC}$ such that the following is true:

1. $Q$ is not $\tau$-approx. learnable.
2. $Q \in_{obf}$ *spec*
3. $Q$ is $\tau$-decidable
4. $O$ is a $\tau$-obfuscator for $Q$ satisfying IND-soundness (§4.2, Definition 3).

A useful question is: *Given spec, can we decide if $O$ satisfies WBP for* $(Q, spec)$*?*

*Why is it useful?* Ideally, we would like the WBP to be satisfied. However, WBP is defined w.r.t. a family and a specification while soundness is defined w.r.t. a family, independent of the specification. On the one hand, due to this simplified definition, obfuscator designers may find it appealing. On the other hand, it is possible that IND-soundness may be too strong to be satisfied even though WBP is (w.r.t. to some specification), as in the example in the proof of Theorem 3. Nevertheless, we consider it an interesting question to characterize cases when the WBP can be reduced to IND-soundness. The assumption that $O$ is a $\tau$-obfuscator (rather than an obfuscator) for $Q$ is necessary to falsify Proposition 1, which rules out interesting families.

*An Open Question:* If *spec* is such that the only oracle available to $A$ is the one being obfuscated, then the WBP for *spec* indeed holds if IND-soundness holds. The result also holds if *spec* has additional oracles that always output the same string (which can be given as an auxiliary input to the distinguisher - cf. "distinguishable attack property" of [21]). At this stage, an open question is: *how to characterize specs where $A$ is given additional oracles which output query-dependent strings?*

## 8 Conclusion

In this work, we initiated a formal study of White-Box Cryptography (WBC) and investigated its relationship with obfuscation. We presented definitions and (im)possibility results for obfuscators of *specific* classes of 'interesting' programs families - those that are not approx. learnable. The security requirements of WBC is captured by means of a *White-Box Property (WBP)*, which is defined for some scheme and a security notion (we call the pair a specification). The security requirement of an obfuscator is captured using a *soundness* property. We showed that WBP and soundness are in general quite independent of each other by giving some examples where one is satisfied but the other is not.

Although the WBP is defined for a particular (family, specification) pair, soundness is only defined for a given family and is independent of the specification. A natural question is whether there exist non-trivial families for which the WBP w.r.t. every specification can be reduced to the soundness of an obfuscator for that family. Loosely speaking, an obfuscator that achieves this is said to satisfy the *Universal White-Box Property (UWBP)* for that family. We showed that the UWBP fails for every family that is not approx. learnable. However, we show that under reasonable assumptions there exists an obfuscator $O$ satisfying UWBP for a non-learnable but approx. learnable family. Furthermore, the specification we used for our negative result is quite contrived. Hence, it seems reasonable to expect that a meaningful notion of security for WBC based on WBP can still be achieved for "normal" specifications. As a possible example of this, we presented a (non-trivial) non-approx. learnable family for which there does exist an obfuscator satisfying the WBP in a real-world specification. Additionally, we showed that there exists a (contrived) family $Q \in \mathbb{ALF} \backslash \mathbb{LF}$ for which there exists an obfuscator satisfying UWBP for $Q$.

**Acknowledgement:**

We are thankful to Amir Herzberg for useful discussions and for motivating the topic. We would also like to thank Dennis Hofheinz, Gregory Neven and Ioannis Atsonios for their valuable feedback.

## References

1. Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors. *Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers*, volume 4876 of *Lecture Notes in Computer Science*. Springer, 2007.

2. Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Advances in Cryptology - Crypto 2001*, LNCS 2139:1–18, 2001.
3. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO '98: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, pages 26–45, London, UK, 1998. Springer-Verlag.
4. Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.
5. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.
6. Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi. Cryptanalysis of a white box aes implementation. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 227–240. Springer, 2004.
7. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
8. Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.
9. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532, London, UK, 2001. Springer-Verlag.
10. Ran Canetti and Ronny R. Dakdouk. Obfuscating point functions with multibit output. In *EUROCRYPT'08*, volume to appear of *Lecture Notes in Computer Science*. Springer, to appear.
11. Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. White-Box Cryptography and an AES Implementation. In *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, volume 2595 of *Lecture Notes in Computer Science*, pages 250–270. Springer, 2002.
12. Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. A white-box DES implementation for drm applications. In *Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002, Washington, DC, USA, November 18, 2002, Revised Papers*, volume 2696 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2002.
13. Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 553–562, Washington, DC, USA, 2005. IEEE Computer Society.
14. Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377. ACM, 1982.
15. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
16. Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In Vadhan [23], pages 194–213.
17. Louis Goubin, Jean-Michel Masereel, and Michaël Quisquater. Cryptanalysis of white box des implementations. In Adams et al. [1], pages 278–295.
18. Satoshi Hada. Zero-knowledge and code obfuscation. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 443–457. Springer, 2000.
19. Amir Herzberg, Haya Shulman, Amitabh Saxena, and Bruno Crispo. Towards a theory of white-box security. Cryptology ePrint Archive, Report 2008/087, 2008. `http://eprint.iacr.org/`.
20. Dennis Hofheinz, John Malone-Lee, and Martijn Stam. Obfuscation for cryptographic purposes. In Vadhan [23], pages 214–232.
21. Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. In Vadhan [23], pages 233–252.
22. Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 20–39. Springer, 2004.
23. Salil P. Vadhan, editor. *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*. Springer, 2007.
24. Hoeteck Wee. On obfuscating point functions. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 523–532, New York, NY, USA, 2005. ACM.
25. Brecht Wyseur, Wil Michiels, Paul Gorissen, and Bart Preneel. Cryptanalysis of white-box des implementations with arbitrary external encodings. In Adams et al. [1], pages 264–277.

## APPENDIX

## A An Example Specification

Let $\mathcal{E} = (G, E, D)$ be a symmetric encryption scheme. We define the IND-CCA2 specification using the following simulation. The corresponding specification is called *ind-cca*2-$\mathcal{E}$. In the following, the key generation algorithm, $G$ takes in as input the security parameter $(1^k)$ and a $k$ bit random string $\gamma$. It outputs a $k$ bit encryption/decryption key *key*.

$\texttt{Expt}_{A,O}^{ind\text{-}cca2\text{-}\mathcal{E}}(1^k, r)$ :
  `Family` $\mathbf{E}$(`key` $key$, `Input` $\langle \alpha, m \rangle$) {
      `/*` $\alpha$ is randomness `*/`
    `output` $E(key, \alpha, m)$
  }
  `Family` $\mathbf{D}$(`key` $key$, `Input` $c$) {
    `output` $D(key, c)$
  }
  `Family` $\mathbf{C}$(`key` $\langle b, key, \beta \rangle$, `Input` $\langle m_0, m_1 \rangle$) {
      `/*` $\mathbf{C}$ is the challenge oracle. $b \in \{0,1\}$ is a bit. $\beta$ is randomness `*/`
    `output` $E(key, \beta, m_b)$
  }
  `Function` $f$(`Input` $r$) {
      `/*` $f : \{0,1\}^{2k+1} \mapsto \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^{2k+1}$ `*/`
    `parse` $r$ `as` $\langle \gamma, \beta, b \rangle$
    $key \leftarrow G(1^{|\gamma|}, \gamma)$
    `output` $key, key, \langle b, key, \beta \rangle$
  }
  $key, key, \langle b, key, \beta \rangle \leftarrow f(r)$
  $s \leftarrow A^{\mathbf{E}^{key}, \mathbf{D}^{key}, \mathbf{C}^{\langle b, key, \beta \rangle}}(1^k, ind\text{-}cca2\text{-}\mathcal{E})$
  `If (win(`$r, QuerySet, s$`)) output 1 else output 0`

Here `win` := "If (At most one query to $\mathbf{C}^{\langle b,key,\beta \rangle}$) $\wedge$ (No query to $\mathbf{D}^{key}$ on output of $\mathbf{C}^{\langle b,key,\beta \rangle}$ after query to $\mathbf{C}^{\langle b,key,\beta \rangle}$) $\wedge$ $(s = b)$".

Clearly, $\mathbf{E} \in_{obf}$ *ind-cca*2-$\mathcal{E}$.