White-Box Cryptography: Formal Notions and (Im)possibility Results

Amitabh Saxena*
International University in Germany
Bruchsal 76646, Germany
amitabh123@gmail.com

Brecht Wyseur Bart Preneel
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10
3001 Heverlee, Belgium
bwyseur,preneel@esat.kuleuven.be

Abstract

A key research question in computer security is whether one can implement software that offers some protection against software attacks from its execution platform. While code obfuscation attempts to hide certain characteristics of a program P, white-box cryptography specifically focusses on software implementations of cryptographic primitives (such as encryption schemes); the goal of a white-box implementation is to offer a certain level of robustness against an adversary who has full access to and control over the implementation of the primitive. Several formal models for obfuscation have been presented before, but it is not clear if any of these definitions can capture the concept of white-box cryptography. In this paper, we discuss the relation between obfuscation and white-box cryptography, and formalize the notion of white-box cryptography by capturing the security requirement using a 'White-Box Property' (WBP). In the second part, we present positive and negative results on white-box cryptography. We show that for interesting programs (such as encryption schemes, and digital signature schemes), there are security notions that cannot be satisfied when adversaries have white-box access, while the notion is satisfied when the adversary has black-box access to its functionality. On the positive side, we show that there exists an obfuscator for a symmetric encryption scheme for which a useful security notion (such as CPA security) remains satisfied when an adversary has access to its white-box implementation.

1 Introduction

In recent years, we have witnessed a trend towards the use of complex software applications with strong security requirements. Think of banking applications, online games, and digital multimedia players. Prominent building blocks for these applications are cryptographic primitives, such as encryption schemes, digital signature schemes, and authentication mechanisms. Unfortunately, such building blocks (e.g., the AES encryption scheme) are guaranteed to be secure only when they are executed on a trustworthy system. It is known that several cryptographic primitives become insecure when the attacker has non-black-box (e.g., 'white-box', or side-channel) access to the computation (see for example [20]).

White-box cryptography (WBC) deals with protecting cryptographic primitives embedded in a program that the attacker has white-box access to. It aims to provide security when the program is executing in a hostile environment and the attacker can conduct non-black-box attacks (such as code inspection, execution environment modification, code modification, etc). Practical white-box implementations of DES and AES encryption algorithms were proposed in [9, 10]. However, no formal definitions of white-box cryptography were given, neither were there any proofs of security. With their subsequent cryptanalysis [4, 15, 23], it remains an open question whether or not such white-box implementations exist. In this paper, we initiate a study of rigorous security notions for the white-box setting.

One way to realize WBC is to obfuscate 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 that an obfuscator ensures that all the security notions are satisfied in a white-box attack context when they are satisfied in the black-box attack context. However, it is still not clear if any existing definitions of obfuscation can be used to achieve this goal. Hence, a natural question is:

Given an obfuscator $\mathcal{O}$ satisfying the virtual black-box property for a program P (in some sense), and a cryptographic scheme that is secure when the adversary is given black-box access to P, can it be shown that the scheme remains secure when the adversary is given white-box access to the program $\mathcal{O}(P)$?

*This work was initiated when the author was working at the University of Trento, Italy.

1.1 Our Contribution

The contributions of this work are two-fold. First, we develop the foundations of white-box cryptography by formalizing the notion of security which cryptographic primitives must satisfy. In order to do this, we define a *white-box property (WBP)* that captures the security of an obfuscated program with respect to some given security notion. The WBP, if satisfied, will imply that the obfuscation does not leak any useful information under that security notion (even though it may leak useful information under a different security notion).

Second, we present some (im)possibility results about reductions between WBP and obfuscation and answer the above question in the negative – we show that under any definition of obfuscation, the answer is, in general, no. In other words, we show that for most programs P, there cannot exist an obfuscator that satisfies the WBP for all security notions in which P might be present. We also show impossibility results for the composition of white-box implementations. On the positive side, we show that under reasonable computational assumptions, there exists an obfuscator that satisfies the WBP with respect to a meaningful security notion for a meaningful cryptographic primitive. We also show that there exist obfuscators that satisfy the WBP with respect to *every* security notion for a (contrived) non-learnable, but approximate learnable family.

To understand our results, it is important to note that obfuscation and WBC are two different concepts and should not be confused with each other. Obfuscation is captured using a Virtual Black-Box Property (VBBP), which is defined with respect to a program alone [1, 19, 11, 14, 22], while WBC is captured using a WBP, which is always defined with respect to a program and a security notion.

1.2 Notation and Preliminaries

Denote by $\mathbb{P}$ the set of all polynomials with non-negative integer coefficients and by $\mathbb{TM}$ the set of all Turing Machines (TMs). For $X \in \mathbb{TM}$, $|X|$ is the length of the string description of X. A mapping $f : x \ni \mathbb{N} \mapsto f(x) \in \mathbb{R}$ is negligible in x (written $f(x) \leq negl(x)$) if $\forall p \in \mathbb{P}, \exists x' \in \mathbb{N}, \forall x > x' : f(x) < 1/p(x)$.

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

1. **(Equality of TMs.)** $X, Y \in \mathbb{TM}$ *are equal (denoted as* $X = Y$*) 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. Let* $\mathbb{PTM}$ *be the set of all PTMs.*

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*. Let* $\mathbb{TMF}$ *be the set of all TMFs. For any* $Q \in \mathbb{TMF}$*:*

   (a) $Q[q] \in \mathbb{TM}$ *is the resulting TM when the key tape of* Q *contains string* q*.*

   (b) $\mathcal{K}_Q$ *is the key-space (valid strings for the key tape) of* Q*.*

   (c) *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 there must exist a polynomial* $\mathbf{P}_Q \in \mathbb{P}$ *s.t.:*

   $$\forall q \in \mathcal{K}_Q, \forall x \in \mathcal{I}_{Q,|q|} : |x| = \mathbf{P}_Q(|q|)\,.$$

   All TMFs in this paper are deterministic (i.e., for any $(Q, q) \in \mathbb{TMF} \times \mathcal{K}_Q$ *the output of* $Q[q]$ *is fully defined by the input). For modeling probabilistic algorithms using TMFs we assume that randomness is encoded in* q *and/or the input.*

5. **(Polynomial TM Family.)** $Q \in \mathbb{TMF}$ *is a* Polynomial *TMF (PTMF) if:*

   (a) *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.*

   (b) *Deciding if some* $x \overset{?}{\in} \mathcal{K}_Q$ *is easy (or* $\mathcal{K}_Q \in P$*).*

   We denote the set of all PTMFs by $\mathbb{PTMF}$*.*

6. **(Learnable Family.)** $Q \in \mathbb{TMF}$ *is learnable if*

   $$\exists (L, p) \in \mathbb{PPT} \times \mathbb{P} \text{ s.t. } \forall k : \Pr\left[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q; \; X \leftarrow L^{Q[q]}(1^{|q|}, Q) : X = Q[q]\right] \geq 1/p(k)$$

   (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) $|X| \leq p(|q|)$*.*

   Informally, a function Q *is learnable when, by means of a limited number of queries to its functionality, an equivalent function* X *can be constructed.* L *is called the learner for* Q*.* $\mathbb{LF}$ *is the set of all learnable families.*

7. **(Approximate Learnable Family.)** $Q \in \mathbb{TMF}$ *is approximate learnable if* $\exists (L, p) \in \mathbb{PPT} \times \mathbb{P}$ *s.t.*

$$\forall k : \Pr\Big[q \xleftarrow{R} \{0,1\}^k \cap \mathcal{K}_Q; a \xleftarrow{R} \mathcal{I}_{Q,k};$$
$$X \leftarrow L^{Q[q]}(1^{|q|}, Q) : X(a) = Q[q](a)\Big] \geq 1/p(k)$$

(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) $|X| \leq p(|q|)$.

$\mathbb{ALF}$ *denotes the set of all approximate learnable families.*

2 Obfuscation

Informally, an obfuscator $\mathcal{O}$ is a probabilistic compiler that transforms a program P into $\mathcal{O}(P)$, a functionally equivalent implementation of P which hides certain characteristics of P.

2.1 Related Work

The notion of code-obfuscation was first given by Hada [16], who introduced the concept of virtual black-box property (VBBP) using computational indistinguishability. In [1], 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 [11] extend the impossibility results of [1] with respect to auxiliary inputs.

On the positive side, Lynn *et al.* [21] show how point functions can be obfuscated the random oracle model. Wee [22] showed how to obfuscate point functions without random oracles. Hohenberger *et al.* [19] 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.* [18] 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 [14] 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 [8] give an obfuscator for point functions with multi-bit output for use in primitives called "digital lockers". Finally, Herzberg *et al.* [17] 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 [1].

2.2 Obfuscators

Denote by Q a family of cryptographic primitives (a PTMF), for which their description is publicly known. Denote by $Q[q]$ a primitive instantiated with secret key q selected from some distribution. We consider the obfuscation of $Q[q]$. We capture the functionality of an obfuscator using *correctness* property and the security using *soundness*.

2.2.1 Obfuscator (Correctness)

Definition 2. *A PPT algorithm* $\mathcal{O} : \mathbb{PTMF} \times \{0,1\}^* \mapsto \mathbb{TM}$ *is an (efficient) obfuscator for* $Q \in \mathbb{PTMF}$ *if it satisfies* correctness *defined using the following two properties:*

1. ***Approximate functionality:*** $\forall q \in \mathcal{K}_Q, \forall a \in \mathcal{I}_{Q,|q|}$:

$$\Pr[\mathcal{O}(Q,q)(a) \neq Q[q](a)] \leq negl(|q|),$$

the probability taken over the coin tosses of $\mathcal{O}$.

2. ***Polynomial slowdown and expansion:*** *There exists* $p \in \mathbb{P}$ *s.t.* $\forall q \in \mathcal{K}_Q : |\mathcal{O}(Q,q)| \leq p(|q|)$, *and* $\forall a$, *if* $Q[q](a)$ *halts in* t *steps then* $\mathcal{O}(Q,q)(a)$ *halts in at most* $p(t)$ *steps.*

Remark 1. *We consider the functionality of Q only in a deterministic sense and do not explicitly consider the notion of obfuscation of "probabilistic functions" (used, for example, in [18, 19]). However, our negative results (of Sect. 4.1) also apply to probabilistic functions using an appropriately defined notion of* probabilistic PTMFs (PPTMFs) *(and a corresponding notion of approximate functionality for PPTMFs). See Sect. 4.4.*

2.2.2 Obfuscator (Soundness)

Several definitions of soundness have been proposed in the literature, all based on a *Virtual Black-Box Property (VBBP)* [1, 18, 19, 21, 22]. Let $Q \in \mathbb{PTMF}$ and let $q \in \{0,1\}^*$. Then, the VBBP requires that any information about q a PPT adversary computes given the obfuscation $\mathcal{O}(Q,q)$, 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 on 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 $\mathcal{O}(Q,q)$. We define these two notions below.

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

Predicate Virtual black-box property (PVBBP): *Let π be an efficiently verifiable predicate. $\mathcal{O}$ satisfies PVBBP for Q if $\forall A \in \mathbb{PPT}, \exists S \in \mathbb{PPT} : Adv^{pvbbp}_{A,S,\mathcal{O},Q}(k) \leq negl(k)$, where $Adv^{pvbbp}_{A,S,\mathcal{O},Q}(k) =$*

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

the probability taken over the coin tosses of $\mathcal{O}, A, S$.

Computational Indistinguishability (IND): *$\mathcal{O}$ satisfies IND for Q if $\forall A \in \mathbb{PPT}, \exists S \in \mathbb{PPT} : Adv^{ind}_{A,S,\mathcal{O},Q}(k) \leq negl(k)$, where $Adv^{ind}_{A,S,\mathcal{O},Q}(k) =$*

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

the probability taken over the coin tosses of $\mathcal{O}, A, S$.

We assumed above that the key q is selected uniformly. To consider keys selected from a distribution different from random, assume WLOG that the keys are selected uniformly from an appropriate subset of the original key-space.

2.3 Discussion

Although several definitions of obfuscation have been proposed in the literature, none of them are accepted as a standard. One of the problems is that in most cases, the IND-soundness definition is too strong in practice to yield any interesting results [19, 22], since deterministic functions can only satisfy the obfuscation definition when they are learnable. This intuition is formalized in Proposition 1.

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

This rules out an obfuscator satisfying IND-soundness for most interesting (deterministic) function families such as pseudorandom functions, encryption and digital signature schemes. Hofheinz *et al.* [18] extended this towards approximate obfuscators for similar types of families.

On the other hand, it has been pointed out in several papers (e.g., [1, 19]) that the PVBBP-soundness definition is too weak to capture any meaningful result, since useful non-black-box information might still leak (a concrete example of this is Theorem 1).

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 white-box security of arbitrary cryptographic primitives. We show this is not the case. Specifically, we show that, under *every* definition of soundness we use, for every family $Q \notin \mathbb{ALF}$, there exist (contrived) security notions for which white-box security fails but the corresponding black-box construction is secure.

3 White-Box Cryptography (WBC)

In this section, we formalize the notion of WBC by defining a *white-box property (WBP)*. We follow the basic principles of various "game-based" approaches [2, 3, 12, 13] where an attack is captured using an interactive game with an adversary. Loosely speaking the WBP is defined using two objects: a TMF (such as an encryption algorithm family) and a **security notion** (such as IND-CPA). A security notion (SN) is a formal description of the security desired from a cryptographic scheme [2, 3]. It defines what capabilities the attacker is given and what constitutes a successful attack. For instance, in IND-CPA for a symmetric encryption scheme, the SN will define that the adversary has access to the encryption oracle and needs to guess a secret bit. For convenience, we make the following assumptions: (1) a SN is specific to a cryptographic scheme (thus, for instance, IND-CPA-Paillier, and IND-CPA-ElGamal are two different SNs), (2) all interaction with the adversary is done via oracle queries made by the adversary.

Definition 4. *A **Security Notion** (SN) is a 5-tuple $(n, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win}) \in \mathbb{N} \times \mathbb{P} \times \mathbb{PTMF}^n \times \mathbb{PTM} \times \mathbb{PTM}$ such that:*

- *$\mathbf{Q} = (Q_1, Q_2, \ldots, Q_n) \in \mathbb{PTMF}^n$ is an array of (the descriptions of) n PTMFs.*
- *Extr and Win are (the description of) PTMs of the type $\{0,1\}^{p_{in}(k)} \mapsto \times_{i=1}^{n} \mathcal{K}_{Q_i}$ and $\{0,1\}^* \mapsto \{0,1\}$ respectively.*

We denote by $\mathbb{SN}$ the set of all security notions. For any $sn = (n, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win}) \in \mathbb{SN}$ and any $Q \in \mathbb{PTMF}$, we say $Q \in sn$ if $Q \in \mathbf{Q}$.

Definition 5. *A **Black-box Game** is a TM describing an interactive protocol with the adversary $A \in \mathbb{PPT}$. It has a standard structure given below. It takes as input a tuple $(1^k, sn, r)$, where $sn = (n, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win}) \in \mathbb{SN}$ is a security notion, and $r \in \{0,1\}^{p_{in}(k)}$ is a string (representing randomness supplied to the game). It outputs 0 or 1. The black-box game is given in Algorithm 1 (GameBB_A).*

$\mathsf{Queries}$ *is a set representing oracle queries made by A during the game. Each element j of this set is an ordered tuple of the type*

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

input : $1^k, sn, r$
Parse sn as $(n, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win})$
Parse $\mathbf{Q}$ as $(Q_1, Q_2, \ldots, Q_n)$
/* extract keys for each family */
$(q_1, q_2, \ldots q_n) \leftarrow \mathsf{Extr}(r)$
/* interact with adversary */
$s \leftarrow A^{Q_1[q_1], Q_2[q_2], \ldots, Q_n[q_n]}(1^k, sn)$
/* decide if adversary won */
$\mathsf{result} \leftarrow \mathsf{Win}(r, \mathsf{Queries}, s)$
output: result

Algorithm 1: $\mathsf{GameBB}_A(1^k, sn, r)$

indicating respectively, the time, oracle number, input, and the output of each query.[1] *The game has two important rules: (1) at any instant A can query at most one oracle, and (2) each query by the adversary takes one unit time irrespective of the amount of computation involved.*

Define the black-box advantage of an adversary A in the black-box game, $\mathsf{AdvBB}_A^{sn}(k)$ as

$$\Pr\left[r \xleftarrow{R} \{0,1\}^{p_{in}(k)} : \mathsf{GameBB}_A(1^k, sn, r) = 1\right],$$

the probability taken over the coin tosses of A.

See Appendix A for an example of the IND-CCA2 security notion for encryption and Appendix B for the sUF-CMA security notion for signatures.

Discussion. Let us recall the CCA2-type security notions for a symmetric encryption scheme (see Appendix A). The game with the adversary in the CCA2 notion consists of three stages: (1) the adversary is allowed to query the encryption/decryption oracles on arbitrary inputs; (2) the adversary obtains a challenge ciphertext corresponding to an unknown challenge plaintext; and, (3) the adversary queries the oracles as in stage 1 except that decryption queries on the challenge ciphertext are disallowed. The adversary wins if it can guess some property of the challenge plaintext.

Let $\mathcal{E} = (G, E, D)$ be a CCA2-secure symmetric encryption scheme, with the encryption/decryption key instantiated to K. Then, $D[K]$ cannot be obfuscated, since this would render the corresponding asymmetric scheme $\mathcal{E}$ insecure under CCA2: once the adversary has obtained an executable implementation of $D[K]$, the third phase of the CCA2 game cannot prevent the adversary querying the decryption function on the challenge ciphertext (the adversary does not even have to 'break' any obfuscation in order to do this). On the other hand $E[K]$ is a perfect candidate for obfuscation since the winning condition does not depend on what was queried to $E[K]$. Definition 6 is introduced to address this issue, and captures when a cryptographic primitive (i.e., family) is a suitable candidate for obfuscation.

[1] Note that the last element of this tuple (the output) is redundant because it is efficiently computable from the input alone. However, including it makes some definitions simpler (for instance IND-CCA2 security).

Definition 6. *For any $sn \in \mathbb{SN}$ and any PTMF $Q_i \in sn$, define $\mathsf{Queries}_i$ to be the following set:*

$$\{(\mathbf{t}_j, \mathbf{i}_j, \mathbf{in}_j, \mathbf{out}_j) | (\mathbf{t}_j, \mathbf{i}_j, \mathbf{in}_j, \mathbf{out}_j) \in \mathsf{Queries} \wedge \mathbf{i}_j \neq i\}$$

*Q_i is **obfuscatable** in sn (written $Q_i \in_{obf} sn$) if it satisfies the* statelessness *property given below:*

$$\forall r, \mathsf{Queries}, s : \mathsf{Win}(r, \mathsf{Queries}, s) = \mathsf{Win}(r, \mathsf{Queries}_i, s).$$

Note: If $Q \notin sn$ then $Q \notin_{obf} sn$.

In other words, $Q_i \in_{obf} sn$ if: (1) $Q_i \in sn$, and (2) in the corresponding black-box game, the output of Win is invariant with respect to the entries of Queries for $Q_i[q_i]$.

We claim that a meaningful notion of white-box security cannot be obtained for a family under a security notion in which it is not obfuscatable. For instance, white-boxing the decryption function of a symmetric encryption scheme, or the 'signing' function of a MAC scheme under any standard security notion is not meaningful (however, it is possible to construct specialized/contrived security notions in which this becomes meaningful).

Definition 7. *A **White-Box Game** is defined for the tuple $(A, \mathcal{O}, Q_i)$ by extending the black-box game. For a security notion $sn = (n, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win}) \in \mathbb{SN}$ with $(\mathbf{Q} = (Q_1, Q_2, \ldots, Q_n)$, assume that $Q_i \in_{obf} sn$ $(1 \leq i \leq n)$ and that $\mathcal{O} \in \mathbb{PPT}$ is an obfuscator for Q_i. The white-box game is given in Algorithm 2 ($\mathsf{GameWB}_{A,\mathcal{O},Q_i}$).*

input : $1^k, sn, r$
Parse sn as $(n, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win})$
Parse $\mathbf{Q}$ as $(Q_1, Q_2, \ldots, Q_n)$
/* extract keys for each family */
$(q_1, q_2, \ldots q_n) \leftarrow \mathsf{Extr}(r)$
/* interact with adversary */
$s \leftarrow A^{Q_1[q_1], Q_2[q_2], \ldots, Q_n[q_n]}(1^k, sn, i, \mathcal{O}(Q_i, q_i))$
/* decide if adversary won */
$\mathsf{result} \leftarrow \mathsf{Win}(r, \mathsf{Queries}, s)$
output: result

Algorithm 2: $\mathsf{GameWB}_{A,\mathcal{O},Q_i}(1^k, sn, r)$

Denote by $\mathsf{AdvWB}_{A,\mathcal{O},Q_i}^{sn}(k)$, the advantage of an adversary in the white-box game, defined as

$$\Pr\left[r \xleftarrow{R} \{0,1\}^{p_{in}(k)} : \mathsf{GameWB}_{A,\mathcal{O},Q_i}(1^k, sn, r) = 1\right],$$

the probability taken over the coin tosses of A.

Definition 8. *Let $\mathcal{O}$ be an obfuscator for $Q \in \mathbb{PTMF}$ and let $sn \in \mathbb{SN}$ such that $Q \in sn$. The **White-box Advantage** of $\mathcal{O}$ for (Q, sn), $AdvWB^{sn}_{\mathcal{O},Q}(k)$, is*

$$\left| \max_{A \in \mathbb{PPT}} AdvWB^{sn}_{A,\mathcal{O},Q}(k) - \max_{A \in \mathbb{PPT}} AdvBB^{sn}_{A}(k) \right| .$$

The WBA serves as a measure of useful information leakage by an obfuscation.

The term 'useful information' within the context of the security notion is any information that aids the adversary in conducting a successful attack.

Definition 9. *Let $\mathcal{O}$ be an obfuscator for $Q \in \mathbb{PTMF}$ and let $sn \in \mathbb{SN}$ such that $Q \in sn$. $\mathcal{O}$ satisfies **White-box Property (WBP)** for (Q, sn) if $AdvWB^{sn}_{\mathcal{O},Q}(k) \leq negl(|k|)$.*

This captures the notion of obfuscation, in the sense that the best adversary in a white-box setting is not able to extract significantly more useful information than the best adversary in a black-box setting.

Definition 10. *Let $\mathcal{O}$ be an obfuscator for $Q \in \mathbb{PTMF}$. $\mathcal{O}$ satisfies **Universal White-box Property (UWBP)** for Q if for every $sn \in \mathbb{SN}$ with $Q \in_{obf} sn$, $\mathcal{O}$ satisfies WBP for (Q, sn).*

Definition 9 and 10 give us a formal, sensible meaning of what the objective of white-box cryptography is. The WBP, if satisfed would imply that, given a cryptographic primitive that is secure in the black-box sense, its white-box implementation also remains secure with respect to the desired security notion. In the next section, we investigate what can be achieved within the context of our model.

4 (Im)possibility Results

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

4.1 Negative Results

Barak *et al.* [1] gave several impossibility results on obfuscation, some of them quite strong (for instance they present an unobfuscatable encryption scheme). Our negative results are even stronger than theirs. To put our main negative result in context with that of [1], we first mention their result using our notation.

Proposition 2. (Barak *et al.* [1]) There exists a pair $(Q, sn) \in \mathbb{PTMF} \times \mathbb{SN}$ with $Q \in_{obf} sn$ such that every obfuscator for Q fails to satisfy WBP for (Q, sn).

In other words, there cannot exist an obfuscator that satisfies UWBP for every $Q \in \mathbb{PTMF}$. 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 approximate learnable.

4.1.1 No UWBP for "Interesting" Families.

Obfuscators that satisfy the UWBP for "interesting" families do not exist. That is, in Theorem 1, we show that any non-approximately-learnable family, there exists a security notion that cannot be satisfied when an adversary has white-box access to the white-box implementation.

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

Proof. Let $Q \in \mathbb{PTMF} \backslash \mathbb{ALF}$. Consider the security notion $guess\text{-}x = (2, p_{in}, (Q, Q_1), \mathsf{Extr}, \mathsf{Win}) \in \mathbb{SN}$, with $p_{in}(k) = 2k + \mathbf{P}_Q(k)$ and other details in Algorithm 3.

Function $Q_1[q_1]$ (***Input*** Y_1) {
  /* Assume: $Y_1 \in \mathbb{PTM}$ */
  Parse q_1 as (q, x, a);
  if $(Y_1(a) = Q[q](a))$ **then** output x **else** output 0;
  /* Q is used as a black-box */
  /* Let $Q[q]$ halt in t steps */
  /* Y_1 is halted after $p(t)$ steps */
  /* for some (large) $p \in \mathbb{P}$ */
}
Function Extr(***Input*** r) {
  Parse r as (q, x, a);
  /* Assume: $q \in \{0,1\}^k$ */
  /* Assume: $x \in \{0,1\}^k$ */
  /* Assume: $a \in \{0,1\}^{\mathbf{P}_Q(k)}$ */
  /* WLOG assume the following: */
  /* Assume: $q \in \mathcal{K}_Q$ */
  /* Assume: $a \in \mathcal{I}_{Q,k}$ */
  set $q_1 \leftarrow (q, x, a)$
  output q, q_1
}
Function Win(***Input*** $(r, \mathsf{Queries}, s)$) {
  Parse r as (q, x, a);
  if $(s \neq x) \vee$ (*more than one query to* $Q_1[q_1]$) **then** output 0 **else** output 1
}

Algorithm 3: Q_1, Extr and Win for $guess\text{-}x$.

Observe that $Q \in_{obf} guess\text{-}x$. Since $Q \notin \mathbb{ALF}$, therefore by virtue of Definitions 1(7) and 2(1), for sufficiently large k, the following inequalities are guaranteed to hold:

$$\forall A \in \mathbb{PPT} : 0 \leq \mathrm{AdvBB}^{guess\text{-}x}_{A}(k) < \alpha(k)$$

$$\exists A \in \mathbb{PPT} : 1 \geq \text{AdvWB}^{guess\text{-}x}_{A,\mathcal{O},Q}(k) \geq 1 - \beta(k),$$

where α, β are negligible functions. Hence, we have:

$$\text{AdvWB}^{guess\text{-}x}_{\mathcal{O},Q}(k) > 1 - \alpha(k) - \beta(k),$$

which is non-negligible in k. This proves the theorem. □

Remark 2. *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.

4.1.2 Simultaneous Obfuscation may be Insecure.

A desired property is the composition of obfuscations. When two implementations are securely obfuscated, one would desire that the combination of the two remains secure, as this opens perspectives to many practical applications. Wee [22] and Canneti *et al.* [8] have investigated this question for point functions, while Lynn *et al.* [21] have found a negative answer to this question for generic programs. In Definition 11, we capture the concept of composition within the context of white-box cryptography. In Theorem 2, we show that simultaneous white-boxing of two families may be insecure even if white-boxing each family is secure.

Definition 11. *(Multiple obfuscations) Let $sn \in \mathbb{SN}$ be a security notion and let $Q_i, Q_j \in_{obf} sn$ for some $1 \leq i, j \leq n$. Let $\mathcal{O}$ be an obfuscator for Q_i, Q_j. Extend the white-box game $\mathsf{GameWB}^{sn}_{A,\mathcal{O},Q_i}$ of Definition 7 by defining a new game $\mathsf{GameWB1}^{sn}_{A,\mathcal{O},Q_i,Q_j}$ in which A gets as input the tuple $(1^k, sn, i, j, \mathcal{O}(Q_i, q_i), \mathcal{O}(Q_j, q_j))$. Denote the advantage of an adversary for this new game as $AdvWB^{sn}_{A,\mathcal{O},Q_i,Q_j}(k) =$*

$$\Pr\left[r \xleftarrow{R} \{0,1\}^{p_{in}(k)} : \mathsf{GameWB1}^{sn}_{A,\mathcal{O},Q_i,Q_j}(1^k, r) = 1\right],$$

the probability taken over the coin tosses of $A, \mathcal{O}$. The obfuscator $\mathcal{O}$ satisfies WBP for $((Q_i, Q_j), sn)$ if

$$\left| \max_{A \in \mathbb{PPT}} AdvWB^{sn}_{A,\mathcal{O},Q_i,Q_j}(k) - \max_{A \in \mathbb{PPT}} AdvBB^{sn}_{A}(k) \right|$$

is negligible in $|k|$.

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

The proof is similar to the proof of Theorem 1.

4.2 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 [18].

4.2.1 WBP for "Useful" Families.

In Theorem 3, we formally state that there exists a non-approximately learnable family, and an obfuscator that satisfies the WBP for that family for a useful security notion.

Theorem 3. *There exists a tuple $(\mathcal{O}, Q, sn) \in \mathbb{PPT} \times \mathbb{PTMF}\backslash\mathbb{ALF} \times \mathbb{SN}$ such that $Q \in_{obf} sn$ and $\mathcal{O}$ is an obfuscator satisfying WBP for (Q, sn) under reasonable computational assumptions.*

Before we prove Theorem 3, we describe a primitive known as bilinear pairing in Definition 12, which we require for the proof.

Definition 12. *(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 [5, 6, 7]:*

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.*

Proof. (of Theorem 3) We prove this by construction. We will use an encryption scheme based on the BF-IBE scheme [5].
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. Pick $x \xleftarrow{R} G_1$ and define $K = (\hat{e}, G_1, G_2, w, g, \mathcal{H}, x)$. The encryption/decryption key is K.

2. *Encryption (E):* Parse K as $(\hat{e}, G_1, G_2, w, g, \mathcal{H}, x)$. Let $m \in \{0,1\}^l$ be a message and $\alpha \in \mathbb{Z}_w$ be a random string. The encryption family E is defined as:

$$\begin{aligned} E[K] : \{0,1\}^l \times \mathbb{Z}_w &\mapsto \{0,1\}^l \times G_1 \\ (m, \alpha) &\mapsto (\mathcal{H}(\hat{e}(x^\alpha, g)) \oplus m, g^\alpha)\,. \end{aligned}$$

3. *Decryption (D):* Parse K as $(\hat{e}, G_1, G_2, w, g, \mathcal{H}, x)$. The decryption family D is defined as:

$$\begin{aligned} D[K] : \{0,1\}^l \times G_1 &\mapsto \{0,1\}^l \\ (c_1, c_2) &\mapsto \mathcal{H}(\hat{e}(c_2, x)) \oplus c_1\,. \end{aligned}$$

It can be verified that $D[K](E[K](m, \alpha)) = m$ for valid values of (m, α). 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 $E[K]$ oracle that converts $\mathcal{E}$ into a CPA secure *asymmetric* encryption scheme under a computational assumption.

The obfuscator $\mathcal{O}$: The input is (E, K).

1. Parse K as $(\hat{e}, G_1, G_2, w, g, \mathcal{H}, x)$. Set $y \leftarrow \hat{e}(x, g)$.
2. Set $K' \leftarrow (\hat{e}, G_1, G_2, w, g, \mathcal{H}, y)$ and define family F with key K' as:

$$\begin{aligned} F[K'] : \{0,1\}^l \times \mathbb{Z}_w &\mapsto \{0,1\}^l \times G_1 \\ (m, \alpha) &\mapsto (\mathcal{H}(y^\alpha) \oplus m, g^\alpha)\,. \end{aligned}$$

3. Output $F[K']$.

Claim 1. *$\mathcal{O}$ is an efficient obfuscator for E satisfying WBP for (E, sn), where $sn :=$ "IND-CPA security of $\mathcal{E}$", assuming that the* bilinear Diffie-Hellman assumption *[5] holds in (G_1, G_2) and $\mathcal{H}$ is a random oracle.*

Proof. The IND-CPA security notion captures an adversary that can only perform queries to an encryption oracle with arbitrary plaintexts. The objective is to obtain the plaintext corresponding to a challenge ciphertext. See Appendix A for the IND-CCA2 security notion. The IND-CPA security notion is a restricted version of this where the family $\mathbf{D}$ is absent.

First note that the obfuscator satisfies correctness for E because $F[K'] = E[K]$. The proof of the above claim follows from the security of the BasicPUB encryption scheme of [5]. □

Claim 2. *If $\mathcal{H}$ is a one-way then $E \in \mathbb{PTMF} \backslash \mathbb{ALF}$.*

Proof. If $\mathcal{H}$ is a one-way function and $x \neq 1$ then it is easy to prove that $E \notin \mathbb{ALF}$. We skip the details. □

This completes the proof of Theorem 3. □

Remark 3. *To justify our choice of the particular scheme (instead of RSA/ElGamal) in the above proof, observe that RSA does not enjoy the security notion of IND-CPA, while Encryption in ElGamal is learnable (to see this, consider access to the ElGamal encryption oracle and obtain encryption of 1 using randomness 1).*

4.3 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.

4.3.1 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 $\mathcal{O}$ for Q that satisfies UWBP for Q.*

Proof. For simplicity, we prove the above result in the random oracle model. Let $\mathcal{R}_{|q|}$ be a random oracle mapping arbitrary strings to $|q|$-bit strings. Consider the family Q defined using Algorithm 4:

Function $Q[q]$ (***Input*** Y) {
  if $(\mathcal{R}_{|q|}(q||Y) = q)$ **then** output 1 **else** output 0;
}

Algorithm 4: Family Q.

Observe that $Q \in \mathbb{PTMF} \cap \mathbb{ALF} \backslash \mathbb{LF}$. It can be proved that $\forall D \in \mathbb{PPT}$ (the distinguisher),

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

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 $\mathcal{O}$ that takes in as input (Q, q) and simply outputs a description of $Q[q]$ as the obfuscation of $Q[q]$. Let $sn \in \mathbb{SN}$ be such that $Q \in_{obf} sn$ but $\mathcal{O}$ does not satisfy WBP for (Q, sn) w.r.t. some adversary $A \in \mathbb{PPT}$ (that is, the white-box advantage

of $(\mathcal{O}, Q, sn)$ 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. □

4.4 Probablistic PTMFs

Our negative results apply because the approximate functionality requirement for obfuscators (in the correctness definition of Sect. 2) is defined only for deterministic PTMFs. What about extended models (such as in [18, 19]) which allow probabilistic families? It turns out that similar results can be obtained for such models using appropriately extended definitions. We give an overview below.

We have two types of PPTMs considering the way randomness is encoded: (1) randomness encoded in the input tape as in the encryption oracle of the example in Appendix A, or (2) randomness encoded in the key tape as in the challenge oracle of the same example. Our negative results assume that the obfuscated family is of Type (1). However, we can also talk about obfuscating Type (2) families (this was considered in [18, 19]). We call a PTMF of the latter type a probabilistic PTMF (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 a pair (Q, τ), where $Q \in \mathbb{PTMF}$ and τ is an equivalence relation on $\mathcal{K}_Q$ that partitions $\mathcal{K}_Q$ into equivalence classes, s.t. $\forall q_1, q_2 \in \mathcal{K}_Q$:

$\tau(q_1, q_2) = 1 \iff$ only random bits of q_1, q_2 are different.

The definitions of equality, (approximate) learnability, approximate functionality and correctness for PTMFs can be extended to PPTMFs using the relation τ.

Despite several known positive white-box results for PPTMFs [18, 19], our negative results also extend to such families assuming a 'τ-decidability' property, which roughly says that for every equivalence class of Q in τ, there must exist an efficient distinguisher that decides whether a given PTM is an member of that class or not (any meaningful PPTMF must satisfy this property). The corresponding statement of our main negative result is: for every PPTMF Q that is τ-decidable but not τ-approx. learnable, there exists $sn \in \mathbb{SN}$ such that $Q \in_{obf} sn$ but every τ-obfuscator for Q fails to satisfy the WBP for (Q, sn). The τ-decidability property allows us to extend the counterexample in the proof of Theorem 1. Details will be given in a forthcoming extended version of this paper.

5 Conclusion

The objective of White-Box Cryptography (WBC) is to implement cryptographic primitives in software in a *secure* way. Since many software implementations are subject to attacks from their execution hosts, WBC is of practical importance. Unfortunately, it lacks a theoretical foundation. This paper provides an initial step to bring the foundations of white-box cryptography to a same level as obfuscation.

This work made several contributions in this regard. We extended the notion of WBC to arbitrary cryptographic primitives and initiated a formal study of WBC by introducing precise definitions of what it means for a white-box implementation to be secure. To achieve this, we formalized the White-Box Property (WBP). The WBP is defined for a program family (e.g., encryption) with associated *Security Notion* (e.g., IND-CPA), and describes how much 'useful information' is leaked from the white-boxed program. We also showed how to encode security notions in a formal manner, which might be of independent interest.

This new theoretic model provides a context to investigate the security of white-box implementations. We present some (im)possibility results, and describe connections between WBC and obfuscation. Specifically, we show that any obfuscator fails to satisfy the *Universal White-Box Property* (UWBP) for non-learnable functions, in the sense that there exists a (contrived) security notion that is satisfied in 'black-box' setting, but fails when an adversary has white-box access to the obfuscated program. However, we show that UWBP can be achieved for non-learnable, but approximate learnable families. Furthermore, we show that under reasonable computational assumptions, there exists a non-learnable family and an obfuscator that satisfies the WBP for that family under a meaningful security notion. In particular, we described an obfuscator that turns a IND-CPA secure symmetric scheme into an IND-CPA secure asymmetric encryption scheme.

References

[1] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (Im)possibility of Obfuscating Programs. In *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 2001.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *Advances in Cryptology - CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer-Verlag, 1998.

[3] M. Bellare and P. Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer-Verlag, 2006.

[4] O. Billet, H. Gilbert, and C. Ech-Chatbi. Cryptanalysis of a White Box AES Implementation. In *Proceedings of the 11th International Workshop on Selected Areas in Cryptography*

(SAC 2004), volume 3357 of *Lecture Notes in Computer Science*, pages 227–240. Springer-Verlag, 2004.

[5] D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer-Verlag, 2003.

[7] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532, London, UK, 2001. Springer-Verlag.

[8] R. Canetti and R. R. Dakdouk. Obfuscating Point Functions with Multibit Output. In *Advances in Cryptology - EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 489–508. Springer-Verlag, 2008.

[9] S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot. White-Box Cryptography and an AES Implementation. In *Proceedings of the 9th International Workshop on Selected Areas in Cryptography (SAC 2002)*, volume 2595 of *Lecture Notes in Computer Science*, pages 250–270. Springer, 2002.

[10] S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot. A white-box DES implementation for DRM applications. In *Proceedings of the ACM Workshop on Security and Privacy in Digital Rights Management (DRM 2002)*, volume 2696 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2002.

[11] S. Goldwasser and Y. T. Kalai. On the Impossibility of Obfuscation with Auxiliary Input. In *Proceedings of the 46th Symposium on Foundations of Computer Science (FOCS 2005)*, IEEE Computer Society, pages 553–562, Washington, DC, USA, 2005. IEEE Computer Society.

[12] S. Goldwasser and S. Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the 14th ACM Symposium on Theory of Computing (STOC 1982)*, pages 365–377. ACM Press, 1982.

[13] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[14] S. Goldwasser and G. N. Rothblum. On Best-Possible Obfuscation. In *Proceedings of 4th Theory of Cryptography Conference (TCC 2007)*, volume 4392 of *Lecture Notes in Computer Science*, pages 194–213. Springer-Verlag, 2007.

[15] L. Goubin, J.-M. Masereel, and M. Quisquater. Cryptanalysis of White Box DES Implementations. In *Proceedings of the 14th International Workshop on Selected Areas in Cryptography (SAC 2007)*, volume 4876 of *Lecture Notes in Computer Science*, pages 278–295. Springer-Verlag, 2007.

[16] S. Hada. Zero-Knowledge and Code Obfuscation. In T. Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 443–457, London, UK, 2000. Springer-Verlag.

[17] A. Herzberg, H. Shulman, A. Saxena, and B. Crispo. Towards a theory of white-box security. Cryptology ePrint Archive, Report 2008/087, 2008. `http://eprint.iacr.org/`.

[18] D. Hofheinz, J. Malone-Lee, and M. Stam. Obfuscation for Cryptographic Purposes. In *Proceedings of 4th Theory of Cryptography Conference (TCC 2007)*, volume 4392 of *Lecture Notes in Computer Science*, pages 214–232. Springer-Verlag, 2007.

[19] S. Hohenberger, G. Rothblum, A. Shelat, and V. Vaikuntanathan. Securely Obfuscating Re-Encryption. In *Proceedings of 4th Theory of Cryptography Conference (TCC 2007)*, volume 4392 of *Lecture Notes in Computer Science*, pages 233–252. Springer-Verlag, 2007.

[20] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology - CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397, London, UK, 1999. Springer-Verlag.

[21] B. Lynn, M. Prabhakaran, and A. Sahai. Positive Results and Techniques for Obfuscation. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 20–39. Springer-Verlag, 2004.

[22] H. Wee. On Obfuscating Point Functions. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC 2005)*, pages 523–532, New York, NY, USA, 2005. ACM Press.

[23] B. Wyseur, W. Michiels, P. Gorissen, and B. Preneel. Cryptanalysis of White-Box DES Implementations with Arbitrary External Encodings. In *Proceedings of the 14th International Workshop on Selected Areas in Cryptography (SAC 2007)*, volume 4876 of *Lecture Notes in Computer Science*, pages 264–277. Springer-Verlag, 2007.

A The IND-CCA2 Security Notion

Let $\mathcal{E} = (G, E, D)$ be a symmetric encryption scheme. The key generation algorithm, G takes in as input the security parameter (1^k) and a k bit random string. It outputs a k bit symmetric key K. As an example, we describe *Indistinguishability under Adaptive Chosen Ciphertext Attack* (IND-CCA2) of $\mathcal{E}$ using security notion $ind\text{-}cca2\text{-}\mathcal{E} = (3, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win})$, with $p_{in}(k) = 2k + 1$ and $\mathbf{Q} = (\mathbf{E}, \mathbf{D}, \mathbf{C})$ (see Algorithm 5).

Function $\mathbf{E}[K](\textbf{\textit{Input}}\ (\alpha, m))$ {
  /* This is encryption oracle */
  /* K is encryption key */
  /* α is randomness */
  /* m is plaintext */
  output $E(K, \alpha, m)$
}
Function $\mathbf{D}[K](\textbf{\textit{Input}}\ c)$ {
  /* This is decryption oracle */
  /* K is decryption key */
  /* c is ciphertext */
  output $D(K, c)$
}
Function $\mathbf{C}[(b, K, \beta)](\textbf{\textit{Input}}\ (m_0, m_1))$ {
  /* This is challenge oracle */
  /* $b \in \{0,1\}$ is a bit */
  /* K is encryption key */
  /* β is randomness */
  /* m_0, m_1 are plaintexts */
  if $(|m_0| = |m_1|)$ **then** output $E(K, \beta, m_b)$ **else** output 0.
}
Function $\mathsf{Extr}(\textbf{\textit{Input}}\ r)$ {
  /* Assume: $r \in \{0,1\}^{2k+1}$ */
  parse r as (γ, β, b)
  /* Assume: $\gamma \in \{0,1\}^k$ */
  /* Assume: $\beta \in \{0,1\}^k$ */
  /* Assume: $b \in \{0,1\}$ */
  $K \leftarrow G(1^{|\gamma|}, \gamma)$
  output $K, K, (b, K, \beta)$
}
Function $\mathsf{Win}(\textbf{\textit{Input}}\ (r, \mathsf{Queries}, s))$ {
  Parse r as (q, x, a);
  if ((*At most one query to* $\mathbf{C}[(b, K, \beta)]$) ***AND***
  (*No query to* $\mathbf{D}[K]$ *on output of* $\mathbf{C}[(b, K, \beta)]$ *after query to* $\mathbf{C}[(b, K, \beta)]$) ***AND*** $(s = b)$)
  then output 1 **else** output 0.
}

Algorithm 5: Security Notion $ind\text{-}cca2\text{-}\mathcal{E}$

$\mathbf{E} \in_{obf} ind\text{-}cca2\text{-}\mathcal{E}$ since the Win predicate does not consider the queries made to the encryption oracle $\mathbf{E}[K]$.

B The sUF-CMA Security Notion

As another example, we give a security notion for signatures called *strong Unforgeability under Chosen Message Attack* (sUF-CMA), defined for probabilistic schemes. In this, the adversary must output a valid (message, signature) pair that is not the (input, output) pair of any sign query. In fact, the notion we present is even stronger than standard sUF-CMA because we allow the adversary to choose randomness for the signing oracle.[2]

Let $\mathcal{S} = (G, S, V)$ be a signature scheme. The key generation algorithm, G takes in as input the security parameter (1^k) and a k bit random string. It outputs a k bit signing key K_s and a k bit verification key K_v. We define sUF-CMA security of $\mathcal{S}$ using security notion $suf\text{-}cma\text{-}\mathcal{S} = (2, p_{in}, \mathbf{Q}, \mathsf{Extr}, \mathsf{Win})$, with $p_{in}(k) = k$ and $\mathbf{Q} = (\mathbf{S}, \mathbf{V})$. Details are given in Algorithm 6.

Function $\mathbf{S}[K_s](\textbf{\textit{Input}}\ (\alpha, m))$ {
  /* This is signing oracle */
  /* K_s is signing key */
  /* α is randomness */
  /* m is message */
  output $S(K_s, \alpha, m)$
}
Function $\mathbf{V}[K_v](\textbf{\textit{Input}}\ \textit{NULL})$ {
  /* K_v is verification key */
  /* output verification key */
  output K_v
}
Function $\mathsf{Extr}(\textbf{\textit{Input}}\ r)$ {
  /* Assume: $r \in \{0,1\}^k$ */
  $(K_s, K_v) \leftarrow G(1^{|r|}, r)$
  output K_s, K_v
}
Function $\mathsf{Win}(\textbf{\textit{Input}}\ (r, \mathsf{Queries}, s))$ {
  $(K_s, K_v) \leftarrow G(1^{|r|}, r)$
  Parse s as (m, σ);
  if $((V(K_v, m, \sigma) = \mathit{True})$ ***AND***
  $(\nexists \alpha : ((\alpha, m), \sigma)$ *is input, output pair of* $\mathbf{S}[K_s]))$
  then output 1 **else** output 0.
}

Algorithm 6: Security Notion $suf\text{-}cma\text{-}\mathcal{S}$

Observe that $\mathbf{S} \notin_{obf} suf\text{-}cma\text{-}\mathcal{S}$.

[2] One way to define the weaker notion, where randomness is not selected by adversary, is to assume that the adversary can make at most ℓ sign queries and replicate the signing oracle ℓ times. Then set each oracle to use different randomness, which is now encoded in the key tape (as in the challenge oracle of $ind\text{-}cca2\text{-}\mathcal{E}$). Before declaring a win in the Win predicate, ensure that each signing oracle was queried at most once.