IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 4, pp. 167–187. DOI:10.46586/tches.v2023.i4.167-187

On Provable White-Box Security in the Strong Incompressibility Model

Estuardo Alpirez Bock[1], Chris Brzuska[2] and Russell W. F. Lai[2]

[1] Xiphera LTD, Espoo, Finland estuardo.alpirezbock@xiphera.com
[2] Aalto University, Espoo, Finland chris.brzuska@gmail.com,russel.lai@aalto.fi

Abstract. *Incompressibility* is a popular security notion for white-box cryptography and captures that a large encryption program cannot be compressed without losing functionality. Fouque, Karpman, Kirchner and Minaud (FKKM) defined *strong incompressibility*, where a compressed program should not even help to *distinguish* encryptions of two messages of equal length. Equivalently, the notion can be phrased as indistinguishability under chosen-plaintext attacks and key-leakage (LK-IND-CPA), where the leakage *rate* is high.

In this paper, we show that LK-IND-CPA security with superlogarithmic-length leakage, and thus strong incompressibility, cannot be proven under standard (i.e. single-stage) assumptions, if the encryption scheme is *key-fixing*, i.e. a polynomial number of message-ciphertext pairs uniquely determine the key with high probability. Our impossibility result refutes a claim by FKKM that their big-key generation mechanism achieves strong incompressibility when combined with any PRG or any conventional encryption scheme, since the claim is not true for encryption schemes which are key-fixing (or for PRGs which are injective). In particular, we prove that the cipher block chaining (CBC) block cipher mode is key-fixing when modelling the cipher as a truly random permutation for each key. Subsequent to and inspired by our work, FKKM prove that their original big-key generation mechanism can be combined with a *random oracle* into an LK-IND-CPA-secure encryption scheme, circumventing the impossibility result by the use of an idealised model.

Along the way, our work also helps clarifying the relations between incompressible white-box cryptography, big-key symmetric encryption, and general leakage resilient cryptography, and their limitations.

Keywords: White-Box Cryptography · Incompressibility · Bounded-Retrieval · Leakage Resilience · Impossibility · Provable Security

1 Introduction

1.1 White-Box Cryptography and Big-Key Encryption

Chow, Eisen, Johnson, and van Oorschot [CEJv03, CEJvO03] introduced the white-box attack model, where an adversary obtains full access to the implementation code of a cryptographic algorithm and is in full control of its execution environment. White-box cryptography aims at designing cryptographic implementations which remain secure in the presence of a white-box adversary. As of today, white-box programs are largely deployed for Digital Rights Management (DRM) and protecting mobile payment applications [EMV19, Sma14]. Typically, white-box programs correspond to white-box versions of popular symmetric encryption schemes, such as the Advanced Encryption Standard (AES).

Defining security for white-box cryptography is intricate and has been studied by quite a number of works [SWP09, Wys11, DLPR14, FKKM16a, ABF+20, AABM20, ABCW23, HITY22]. In particular, Delerablée, Lepoint, Paillier, and Rivain (DLPR) [DLPR14] noted

Received: 2023-04-15 Accepted: 2023-06-15 Published: 2023-08-31

that necessary security goals for white-box programs include not leaking the key (security against key extraction) and not leaking the message (one-wayness). In addition, the authors also pointed out that white-box programs might be susceptible to so called *code-lifting* attacks, where an adversary simply copies the entire execution code of the program and runs a copy on any device of its choice [Bre12]. As a mitigation technique against code-lifting attacks, DLPR proposed the property of *incompressibility*. Constructions achieve incompressibility by designing large programs implementing the desired cryptographic operations which only remain functional in their complete forms, i.e. if such a program is compressed, or if fragments of it are removed, the program loses its functionality. The idea behind incompressibility is that a white-box adversary should not be able to copy the complete program due to its large size, e.g. it should be difficult to move such a large program over the network. At the same time, seeing (or copying) only some fractions of the program should not give enough information to the adversary for breaking the security of the program. For instance, an adversary who sees the incompressible program should not be able to derive a functionally equivalent program of smaller size, or should not be able to copy small fragments of the program and use them for decrypting arbitrary ciphertexts.

1.1.1 Incompressibility

As mentioned above, incompressibility was introduced in [DLPR14] as a means for white-box programs to mitigate code-lifting attacks. In the syntax of an incompressible program, we depart from a symmetric encryption scheme of conventional size and use a *white-box compiler* to derive an incompressible encryption (or decryption) program. The incompressible program can then be used in combination with the encryption scheme for performing encryptions (or decryptions) in the white-box attack model, while the inverse operation can be performed using the original program of conventional size (and thus retaining the original efficiency). Incompressibility seems particularly interesting for white-box programs as it provides a means of mitigating code-lifting attacks without needing to rely on any external hardware or software components (in contrast to e.g. device-binding [ABF+20, AABM20, ABCW23]).

Since its introduction, a line of works [DLPR14, FKKM16a, BBK14, BI15, BIT16, AAB+19, KLLM20, KI21, HITY22] has presented constructions and concrete designs for incompressible ciphers according to various security definitions [DLPR14, FKKM16a, BI15, BIT16, AAB+19, KI21, HITY22]. All these definitions capture incompressibility intuitively as described in the beginning of this section, but differ on how an adversary obtains leakage from the incompressible white-box program, or how/whether the adversary may interact with encryption oracles after receiving key leakage. In particular, Fouque, Karpman, Kirchner, and Minaud (FKKM) [FKKM16a] introduced the *strong incompressibility* model, which is our main focus.

Adopting the terminology of IND-CPA-security of encryption schemes, the strong incompressibility model can be seen as *indistinguishability under chosen plaintext attacks and key-leakage (LK-IND-CPA)*. The LK-IND-CPA model indeed allows the adversary to make encryption queries *after* seeing key leakage.[1] Intuitively, if the incompressible construction still has enough min-entropy conditioned on the leakage, e.g. a compressed version of the program, then the adversary should not be able to distinguish between encryptions of different messages. In other words, this model captures *confidentiality*, which implies security against key extraction and one-wayness.

[1] Here, the term *adversary* refers to the algorithm trying to break the winning condition, e.g. encrypt a message, distinguish between two values etc. In white-box security notions, the leakage algorithm or a compressed adversary *generator* is sometimes referred to as an adversary and the entity trying to break the winning condition is sometimes referred to as a *decompression algorithm*.

1.1.2 Big-Key Encryption

A very similar model was considered by Bellare, Kane and Rogaway (BKR) in the context of *big-key (symmetric) encryption* [BKR16]. In big-key encryption, keys of very large size are used as a means to achieve security in the bounded retrieval model (BRM), where we assume that the adversary can only exfiltrate a limited amount of data from a system under attack. BKR propose to use such large keys for deriving shorter subkeys, which can then be used for encrypting messages using conventional encryption schemes. For capturing security, BKR consider IND-CPA under leakage (which BKR refer to as LIND) and thus a model which is essentially identical to strong incompressibility. The two models only differ on their definitional style, since instead of bounding min-entropy directly, BKR consider a bounded-length leakage function which implies an upper bound on the lost min-entropy of the key. Throughout this paper, *LK-IND-CPA* refers to said IND-CPA under leakage model and we consider this definition in the context of incompressible white-box cryptography. For completeness, Section 5 shows formally that LK-IND-CPA is equivalent to *strong incompressibility* by FKKM and to the LIND model by BKR.

1.1.3 Similarities and Differences

White-box incompressibility and big-key encryption share not only common intuitive goals, as discussed above, but also possible use cases. Bogdanov and Isobe [BI15] discuss that incompressibility may be useful for thwarting mass surveillance, which BKR [BKR16] also mention as a main motivation of big-key encryption. The idea is that it may be admissible for a local user to store large keys on their own device. However, a large-scale surveillance project might not be able to store the keys of many users, if all users employ large keys.

The main syntactical difference between incompressible white-box cryptography and big-key symmetric encryption is the process of how large keys are generated. In big-key encryption, a large key is simply generated at random, and the same large key is used for both encryption and decryption. In white-box cryptography, however, the large key (or incompressible construction) is derived from a conventional (small-key) encryption scheme via a *white-box compiler*. The compiler takes the short key of the encryption scheme and compiles it into a functionally equivalent large key/program which performs encryptions secure under leakage.

Having functionally equivalent short and large keys is particularly useful in scenarios where only either encryption or decryption will be performed in the presence of a leakage adversary. For instance, if encryption and decryption are performed in a safe environment and a leaky environment respectively, then short key may be used for encryption while the large key should be used for decryption.

1.2 On Strong Incompressibility from Standard Assumptions

The central question that we address in this work is the following:

> Can white-box incompressible schemes based on conventional ciphers *really* be provably secure under standard assumptions?

This question is motivated by the use of white-box cryptography for conventional ciphers such as AES using conventional cipher modes such as CBC mode.

We answer the above question in the negative. We show that, for some types of encryption schemes, it is impossible to achieve LK-IND-CPA security even with very mild leakage under simple assumptions. More concretely, we present a general negative result showing that, if a symmetric encryption scheme satisfies a property called *key-fixing*, then it cannot be proven secure under a super-logarithmic amount of leakage based on single-stage assumptions. A single-stage assumption is defined by a game played by a single adversary,

which is the case for most standard definitions. For example, IND-CPA security (cf. game $\mathsf{indcpa}^b_{\mathsf{ske},\mathcal{A}}(1^n)$ in Fig. 1) is a single-stage assumption. In turn, LK-IND-CPA security (cf. game $\mathsf{lkindcpa}^b_{\mathsf{bke},\mathcal{G},\beta,m}(1^n)$ in Fig. 1) is a two-stage assumption since the leakage-producing adversary $\mathcal{A}_1$ and the main adversary $\mathcal{A}_2$ only share a short leakage *lkg* produced by $\mathcal{A}_1$, but not their complete states.

Moreover, we show that the CBC mode of symmetric ciphers is key-fixing, when modelling the cipher as an ideal random permutation for each key. In other words, it is impossible to provably compile a CBC-mode-based encryption into its strongly incompressible version under single-stage assumptions. Our result highlights the importance of taking the cipher mode into account when arguing about the incompressibility of white-box encryption schemes built from symmetric ciphers. Reasoning only about the security of the key generator does not suffice. For example, our result shows that FKKM's key generator cannot be used in combination with key-fixing symmetric-encryption schemes if we wish to have a provably secure white-box encryption scheme.

1.3 Technical Overview

Our goal is to show that the LK-IND-CPA security of a symmetric encryption scheme with super-logarithmic-length leakage cannot be reduced to any single-stage assumption in a black-box way. In the following, we will refer to the length of the leakage as $\beta = \omega(\log n)$, where n is the security parameter. We will also assume that the a secret key is a bit string in $\{0,1\}^m$ with $m = \mathsf{poly}(n)$. We will use the term WBE to refer to an incompressible white-box encryption scheme. We shall keep in mind, however, that our result also applies to big-key encryption.

Concretely, we prove via a meta-reduction that for any single-stage assumption modelled by a game G^b, no PPT reduction $\mathcal{R}$ turns a pair of PPT adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ with non-negligible advantage against LK-IND-CPA security into a PPT algorithm $\mathcal{R}^{(\mathcal{A}_1,\mathcal{A}_2)}$ with non-negligible distinguishing advantage against the game G^b.

1.3.1 Meta-Reduction Outline

Our proof technique is inspired by the meta-reduction approach of Wichs [Wic13], who shows an impossibility result about leakage-resilient injective one-way functions. We here paraphrase the intuition given by Wichs in our context: Building a reduction to prove leakage-resilience or incompressibility is difficult, because the reduction first needs to give a key k to a first adversary $\mathcal{A}_1$, and then receives some leakage *lkg* about k. Then, the reduction gives *lkg* to a second adversary $\mathcal{A}_2$ and additionally simulates an encryption oracle for $\mathcal{A}_2$. The reduction does not know how adversary $\mathcal{A}_1$ computes the leakage. Therefore, it seems that the only way in which the reduction can simulate the leakage *lkg* and the encryption oracle ENC correctly together is by knowing the key k already (see Section 6 for a discussion). If the answers to the encryption oracle fix the key k *uniquely*, then we can actually formalise the above. Technically, Wichs would have implemented the above intuition using the two following steps.

1. First, sample a pair of *inefficient* adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ which *do not share any state* and break the security of the WBE with non-negligible advantage.

2. Next, design a *PPT* simulator $\mathcal{S}$ that simulates both adversaries with a *shared state* such that $(\mathcal{A}_1, \mathcal{A}_2) \overset{\text{comp}}{\approx} \mathcal{S}$.

Assume towards contradiction that there exists a single-stage assumption modelled by a game G^b and a PPT reduction $\mathcal{R}$ which bases the LK-IND-CPA security of the encryption scheme on said assumption, then it follows that $\mathcal{R}^{\mathcal{S}}$ is PPT (because both $\mathcal{R}$ and $\mathcal{S}$ are), and

has non-negligible advantage against G^b (since $(\mathcal{A}_1, \mathcal{A}_2) \overset{\text{comp}}{\approx} \mathcal{S}$ implies $\mathcal{R}^{(\mathcal{A}_1,\mathcal{A}_2)} \overset{\text{comp}}{\approx} \mathcal{R}^{\mathcal{S}}$), which contradicts the single-stage assumption.

Simulation vs. Attack. The above simulation does not give a valid attack against the scheme, because the two adversaries "cheat" by sharing a state. In fact, the shared state gives the second adversary the *entire* key rather than just the leakage. This difference is significant because there could be exponentially many possible keys which are consistent with the leakage, and thus it is infeasible for the second adversary to guess the key if given just the leakage.

However, if the reduction cannot detect this cheat, then it must be as successful when interacting with the pair of cheating adversaries as with the real (inefficient) adversaries $(\mathcal{A}_1, \mathcal{A}_2)$. Thus, a key point in the analysis will be to argue that, from the reduction's point of view, the pair of inefficient adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ is indistinguishable from the efficiently simulated pair of cheating adversaries. Below, we outline the inefficient adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ and their simulator in more detail.

Sampling Inefficient $(\mathcal{A}_1, \mathcal{A}_2)$. On a high-level, our simulatable attack works as follows: The adversary pair $(\mathcal{A}_1, \mathcal{A}_2)$ has a hardwired random function RO, with input space $\{0,1\}^m$ and output space $\{0,1\}^\beta$. Note that $\beta = \omega(\log n)$ therefore it is difficult to guess the output of RO on any given input. As required by the security definition of WBE (Definition 4), $\mathcal{A}_1$ and $\mathcal{A}_2$ do not share any state.

Adversary $\mathcal{A}_1$. We construct the first-stage adversary $\mathcal{A}_1$ such that it simply outputs $lkg \leftarrow \mathsf{RO}(K)$ where K is the big key of the WBE. By construction, the length of the leakage is β, as permitted.

Adversary $\mathcal{A}_2$. The second-stage adversary $\mathcal{A}_2$ receives leakage lkg as input and is given access to an encryption oracle encO. It queries sufficiently many messages to encO, until the key K is uniquely determined[2] and then finds K by exhaustive search. If K does not match the leakage lkg, i.e. $lkg \neq \mathsf{RO}(K)$, the adversary $\mathcal{A}_2$ returns 1. In turn, if K matches the leakage, then it will break the IND-CPA game by sending two distinct messages msg_0 and msg_1 to the challenge oracle encO, getting back a challenge ciphertext $ctxt^*$, decrypting $ctxt^*$ using K, and returning 0 if the resulting plaintext is msg_0, and 1, otherwise.

Efficiently Simulating $(\mathcal{A}_1, \mathcal{A}_2)$. The adversary $\mathcal{A}_2$ is inefficient since it searches for K exhaustively. Nevertheless, it turns out that we can simulate the *pair* $(\mathcal{A}_1, \mathcal{A}_2)$ efficiently by a simulator $\mathcal{S}$ which emulates $\mathcal{A}_1$ and $\mathcal{A}_2$ *jointly*, i.e. the simulated adversaries are allowed to share a state. The simulated $\mathcal{A}_1$ (which takes K as input) does not have a hardwired true random function. Instead, it implements RO jointly with the simulated $\mathcal{A}_2$ via lazy sampling, and with queries and responses recorded in a table. The simulated $\mathcal{A}_2$ will not perform an exhaustive search. Instead, it only tries out all polynomially-many values in the table, i.e. values of K that were used for querying $\mathcal{A}_1$.

1.3.2 Conceptual Reflections

We highlight some key points in the analysis of the above meta-reduction.

Soundness of Simulation. To detect simulation, the reduction $\mathcal{R}$ needs to provide a leakage $lkg = \mathsf{RO}(K^*)$ to $\mathcal{A}_2$ without querying $\mathcal{A}_1$ on K^*. Here, if the reduction is interacting with the simulated $\mathcal{A}_2$, it would not be successful, since the simulator wouldn't

[2] Achieving uniqueness is not always possible. We elaborate on this issue shortly.

have the correct value of K^* stored in its state. On the other hand, the real (inefficient) $\mathcal{A}_2$ would successfully break IND-CPA as described above. However, $\mathcal{R}$ is only able to implement this distinguishing strategy with negligible probability, since random values of length $\beta = \omega(\log n)$ are hard to guess.

Key-Fixing. In order to make the above approach work, we need to assume that the key K is uniquely determined after seeing sufficiently many encryption queries. This property, analogous to Wichs' injectivity requirement [Wic13], seems necessary for our adversary to be successful. This is because, if there were too many keys, then the inefficient $\mathcal{A}_2$ would pick an arbitrary key from multiple possibilities with no guarantee in decryption correctness. Indeed, this is how Hazay, Lopez-Alt, Wee, and Wichs (HLWW) [HLWW13] obtain feasibility results for leakage-resilient encryption. In turn, it is still quite unclear how practical encryption schemes could reasonably avoid key-fixing. As we will see in Section 4, CBC-mode is key-fixing when the cipher is modelled as a truly random permutation for every key. We also show that CBC-mode continues to be key-fixing when using different fractions of a large key for each encryption.

Different Models. Our lower bound conceptually also applies to both strong incompressibility and leakage-resilient encryption when the leakage bounds are chosen appropriately. We prove the equivalence between strong incompressibility and leakage-resilient encryption in Section 5.

1.3.3 Outline

Section 2 provides additional background and definitions. Section 3 states and proves our main impossibility result. Section 4 illustrates that practical encryption schemes tend to be key-fixing by showing that CBC-mode is key-fixing even when using different chunks from a big key rather than only a small key. This result models the underlying cipher as a random independent permutation for every key. Section 5 relates strong incompressibility and leakage-resilient encryption. Section 6 discusses consequences of our impossibility result and possible avenues to circumvent it, both in theory and in practice.

2 Preliminaries

Let $\mathbb{N} = \{1, 2, \ldots\}$ be the set of positive integers and $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$ be the set of non-negative integers. For $\ell \in \mathbb{N}$, write $[\ell] := \{1, 2, \ldots, \ell\}$. We write log for $\log_2$, the base-2 logarithm. We denote the security parameter by $n \in \mathbb{N}$. In big-key and white-box primitives, algorithms often take a length parameter $m \in \mathbb{N}$ as additional input. For conciseness, we often make the parameters n and m implicit. If S is a finite set, we write $x \leftarrow\!\!\$\ S$ for sampling from S uniformly at random. For a probabilistic algorithm $\mathcal{A}$, we write $y \leftarrow\!\!\$\ \mathcal{A}(x)$ for running $\mathcal{A}$ on the input x with implicit uniform randomness. If the randomness r is explicit, we write $y \leftarrow \mathcal{A}(x; r)$ instead.

We formalise security under leakage as a game played by a pair of adversaries $(\mathcal{A}_1, \mathcal{A}_2)$. Adversary $\mathcal{A}_1$ computes leakage of the secret key and passes it to $\mathcal{A}_2$.

Definition 1 (Two-Stage Adversary Generators)**.** Let $\mathcal{G}(1^n)$ be an algorithm which outputs a pair of two algorithms $(\mathcal{A}_1, \mathcal{A}_2)$ forming a two-stage adversary. We call the algorithm $\mathcal{G}$ a *two-stage adversary generator* if $\mathcal{G}$ is probablistic polynomial-time and $\mathcal{A}_1$ and $\mathcal{A}_2$ also run in time polynomial in n.

$\mathsf{indcpa}^b_{\mathsf{ske},\mathcal{A}}(1^n)$	$\mathsf{lkindcpa}^b_{\mathsf{bke},\mathcal{G},\beta,m}(1^n)$	$\mathsf{lkindcpa}^b_{\mathsf{wbe},\mathcal{G},\beta,m}(1^n)$
$k \leftarrow\!\!\$\ \{0,1\}^n$	$(\mathcal{A}_1, \mathcal{A}_2) \leftarrow\!\!\$\ \mathcal{G}(1^n)$	$(\mathcal{A}_1, \mathcal{A}_2) \leftarrow\!\!\$\ \mathcal{G}(1^n)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{encO}}(1^n)$	$K \leftarrow\!\!\$\ \mathsf{Kgen}(1^n, 1^m)$	$k \leftarrow\!\!\$\ \{0,1\}^n$
return b'	$lkg \leftarrow\!\!\$\ \mathcal{A}_1(K)$	$K \leftarrow\!\!\$\ \mathsf{wbkgen}(k, 1^m)$
	$b' \leftarrow (\lvert lkg\rvert \leq \beta)$	$lkg \leftarrow\!\!\$\ \mathcal{A}_1(K)$
$\mathsf{encO}(msg_0, msg_1)$	$b'' \leftarrow\!\!\$\ \mathcal{A}_2^{\mathsf{KencO}}(lkg)$	$b' \leftarrow (\lvert lkg\rvert \leq \beta)$
assert $\lvert msg_0\rvert = \lvert msg_1\rvert$	**return** $b' \wedge b''$	$b'' \leftarrow\!\!\$\ \mathcal{A}_2^{\mathsf{encO}}(lkg)$
return $\mathsf{enc}(k, msg_b)$		**return** $b' \wedge b''$
$\mathsf{KencO}(msg_0, msg_1)$		
assert $\lvert msg_0\rvert = \lvert msg_1\rvert$		
return $\mathsf{Kenc}(K, msg_b)$		

Figure 1: (β, m)-LK-IND-CPA-security for SKE, BKE, and WBE.

2.1 Encryption Schemes

Definition 2 (SKE)**.** A symmetric-key encryption (SKE) scheme $\mathsf{ske} = (\mathsf{enc}, \mathsf{dec})$ is a pair of PPT algorithms (which implicitly input 1^n).

- $ctxt \leftarrow\!\!\$\ \mathsf{enc}(k, msg)$: The randomised encryption algorithm enc takes a key $k \in \{0,1\}^n$ and a message $msg \in \{0,1\}^*$ and produces a ciphertext $ctxt \in \{0,1\}^*$.
- $msg \leftarrow \mathsf{dec}(k, ctxt)$: The deterministic decryption algorithm dec takes a key $k \in \{0,1\}^n$ and decrypts a ciphertext $ctxt \in \{0,1\}^*$ to a message $msg \in \{0,1\}^*$.

An SKE scheme ske is correct if for any $n \in \mathbb{N}$, $k \in \{0,1\}^n$, and $msg \in \{0,1\}^*$,

$$\mathsf{dec}(k, \mathsf{enc}(k, msg)) = msg.$$

2.1.1 White-Box Encryption

A white-box encryption scheme is derived from a symmetric encryption scheme with a key of conventional size. The small key of the latter can be transformed into a functionally equivalent big key.

Definition 3 (WBE)**.** A white-box encryption (WBE) scheme

$$\mathsf{wbe} = (\mathsf{wbkgen}, \mathsf{enc}, \mathsf{dec}, \mathsf{Kenc}, \mathsf{Kdec})$$

is a tuple of six PPT algorithms with the following properties:

White-Box Key Generation: The randomised white-box key generation algorithm $K \leftarrow\!\!\$\ \mathsf{wbkgen}(k, 1^m)$ generates a big key $K \in \{0,1\}^m$ given a small key $k \in \{0,1\}^n$ and a length parameter $m \in \mathbb{N}$.

Small-Key Mode: $(\mathsf{enc}, \mathsf{dec})$ is an SKE scheme running on the short key.

Big-Key Mode: $(\mathsf{Kenc}, \mathsf{Kdec})$ is an SKE scheme running on the large key.

A WBE scheme wbe is correct if the following properties are satisfied:

Small-Key Correctness: $(\mathsf{enc}, \mathsf{dec})$ is correct (as an SKE scheme).

Big-Key Correctness: $(\mathsf{Kenc}, \mathsf{Kdec})$ is correct (as an SKE scheme).

Encryption Equivalence: For any PPT distinguisher $\mathcal{D}$, any $n, m \in \mathbb{N}$, $m \geq n$, any $k \in \{0,1\}^n$, any $K \in \mathsf{wbkgen}(k, 1^m)$, any $\mathsf{O}_1, \mathsf{O}'_1 \in \{\mathsf{enc}(k,\cdot), \mathsf{Kenc}(K,\cdot)\}$, and any $\mathsf{O}_2, \mathsf{O}'_2 \in \{\mathsf{dec}(k,\cdot), \mathsf{Kdec}(K,\cdot)\}$,

$$\left|\Pr\left[\mathcal{D}^{\mathsf{O}_1,\mathsf{O}_2}(1^n, 1^m)\right] - \Pr\left[\mathcal{D}^{\mathsf{O}'_1,\mathsf{O}'_2}(1^n, 1^m)\right]\right|$$

is negligible.

In Definition 3, the small-key mode of a WBE works independently of the length parameter chosen for the big-key mode. In other words, a small key of a WBE scheme could have multiple equivalent big keys of different lengths.

We next define LK-IND-CPA security for white-box encryption, as introduced by HLWW, adapted to WBE, with the only difference being the key generation process. Namely, for white-box cryptography, we generate a pseudorandom large key from a smaller key. Note that the LK-IND-CPA model considered by FKKM [FKKM16a] (named strong incompressibility model), provides the adversary access to an additional encryption oracle, so it can obtain ciphertexts for chosen plaintexts. Our models in Fig. 1 do not explicitly provide access to such an encryption oracle, but the adversary can still obtain ciphertexts for chosen plaintexts simply by querying the challenge oracle with two equal messages $msg_0 = msg_1$.

Definition 4. [(β, m)-LK-IND-CPA for WBE] Let β, m be functions of n. A WBE scheme wbe is (β, m)-LK-IND-CPA-secure if for any PPT two-stage adversary generator $\mathcal{G}$

$$\mathsf{Adv}^{\mathsf{lkindcpa}}_{\mathsf{wbe},\mathcal{G},\beta,m}(n) := \left|\Pr\left[\mathsf{lkindcpa}^0_{\mathsf{wbe},\mathcal{G},\beta,m}(1^n) = 1\right] - \Pr\left[\mathsf{lkindcpa}^1_{\mathsf{wbe},\mathcal{G},\beta,m}(1^n) = 1\right]\right|$$

is negligible, where the experiment $\mathsf{lkindcpa}^b_{\mathsf{wbe},\mathcal{G},\beta,m}$ for $b \in \{0,1\}$ is defined in Fig. 1.

2.2 Big-Key Symmetric Encryption

We recall the definition of *big-key encryption (BKE)* [BKR16]. The syntax of BKE is almost identical to that of SKE, except that the (big-)key generation algorithm additionally inputs a length parameter $m \in \mathbb{N}$ which determines the size of the secret-key.

Definition 5 (BKE)**.** A big-key encryption (BKE) scheme $\mathsf{bke} = (\mathsf{Kgen}, \mathsf{Kenc}, \mathsf{Kdec})$ is a tuple of three PPT algorithms with the following properties:

- $K \leftarrow\!\!\$\ \mathsf{Kgen}(1^n, 1^m)$: The randomised big-key generation algorithm Kgen generates a secret key $K \in \{0,1\}^m$.
- $ctxt \leftarrow\!\!\$\ \mathsf{Kenc}(K, msg)$: The randomised big-key encryption algorithm Kenc produces a ciphertext $ctxt \in \{0,1\}^*$ given a big key $K \in \{0,1\}^m$ and a message $msg \in \{0,1\}^*$.
- $msg \leftarrow \mathsf{Kdec}(K, ctxt)$: The deterministic big-key decryption algorithm Kdec takes a big key $K \in \{0,1\}^m$ and decrypts a ciphertext $ctxt \in \{0,1\}^*$ to a message $msg \in \{0,1\}^*$.

A BKE scheme bke is *correct* if for any $n, m \in \mathbb{N}$, $K \in \mathsf{Kgen}(1^n, 1^m)$, and $msg \in \{0,1\}^*$,

$$\mathsf{Kdec}(K, \mathsf{Kenc}(K, msg)) = msg.$$

Definition 6 ((β, m)-LK-IND-CPA for BKE)**.** Let β, m be functions of n. A BKE scheme bke is (β, m)-LK-IND-CPA-secure if for any PPT two-stage adversary generator $\mathcal{G}$

$$\mathsf{Adv}^{\mathsf{lkindcpa}}_{\mathsf{bke},\mathcal{G},\beta,m}(n) := \left|\Pr\left[\mathsf{lkindcpa}^0_{\mathsf{bke},\mathcal{G},\beta,m}(1^n) = 1\right] - \Pr\left[\mathsf{lkindcpa}^1_{\mathsf{bke},\mathcal{G},\beta,m}(1^n) = 1\right]\right|$$

is negligible, where Fig. 1 defines the experiment $\mathsf{indcpa}^b_{\mathsf{bke},\mathcal{G},\beta,m}$ for $b \in \{0,1\}$.

3 Impossibility Result

In this section, we prove that the (β, m)-LK-IND-CPA security of a WBE scheme cannot be black-box-reduced to any single-stage assumption, provided that the WBE scheme is *key-fixing*, and that the leakage $\beta = \omega(\log n)$. In Section 3.1, we formalise single-stage assumptions and black-box reductions which establish the (β, m)-LK-IND-CPA security of WBE schemes as well as key-fixing. We then present our impossibility result in Section 3.2.

3.1 Single-Stage Assumptions, Black-Box Reductions, and Key-Fixing

Notation. In this section, we use the notation $\mathcal{B} \to G^b(1^n)$ for the interaction between adversary $\mathcal{B}$ and the game $G^b(1^n)$. We denote by $1 = \mathcal{B} \to G^b(1^n)$ the even that the adversary returns 1 when interacting with game $G^b(1^n)$. This notation makes the adversary the "main routine" which calls the game (rather than making the game call the adversary).

Remark. The $\to$ notation is useful, because we can later write $\mathcal{B} = \mathcal{R}^{\mathcal{A}_1,\mathcal{A}_2}$ and $\mathcal{R}^{\mathcal{A}_1,\mathcal{A}_2} \to G^b(1^n)$. This way, the notation distinguishes between $\mathcal{R}$'s black-box interface to the two-stage adversary $(\mathcal{A}_1, \mathcal{A}_2)$ and to the game $G^b(1^n)$. In this notation, the security parameter is given to $\mathcal{B}$ and $\mathcal{R}$ implicitly.

Definition 7 (Single-Stage Assumption)**.** A single-stage assumption is defined via two PPT games G^0 and G^1 which provide the same set of oracles $\mathcal{Q}$ to an adversary $\mathcal{B}$. The assumption is then that for all PPT adversaries $\mathcal{B}$,

$$\left|\Pr\left[1 = \mathcal{B} \to G^0(1^n)\right] - \Pr\left[1 = \mathcal{B} \to G^1(1^n)\right]\right|$$

is negligible in n.

Example. The IND-CPA security game $\mathsf{indcpa}^b_{\mathsf{ske},\mathcal{A}}(1^n)$ (cf. Fig. 1) is a single-stage game, while the LK-IND-CPA game $\mathsf{lkindcpa}^b_{\mathsf{bke},\mathcal{G},\beta,m}(1^n)$ (cf. Fig. 1) is a two-stage assumption since the leakage-producing adversary $\mathcal{A}_0$ and the main adversary $\mathcal{A}_1$ do not share their complete state.

Style. Definition 7 encodes security without an experiment environment, but instead the adversary $\mathcal{B}$ is the main procedure and calls oracles which are exposed by the game. This encoding is w.l.o.g., since one can always add an oracle for setup. It is convenient, since the interface between the adversary and the game is explicit. Moreover, encoding single-stage games as distinguishing games is w.l.o.g., since every search game with efficiently checkable winning condition can be encoded as a decision game by adding an oracle in the real world which returns 1 when the winning condition is satisfied, but always returns 0 in the ideal world. This encoding of single-stage games is borrowed from [BDF+18] and requires query restrictions on adversaries to be formulated via *silencing* oracles à la Rogaway-Zhang [RZ18].

We now define what it means for a reduction to base security of a big-key encryption scheme on a single-stage assumption.

Definition 8 (Black-Box Reduction)**.** Let (G^0, G^1) be two games defining a single-stage assumption. Let wbe be a white-box encryption scheme. A PPT oracle algorithm $\mathcal{R}$ bases the (β, m)-LK-IND-CPA security of wbe on (G^0, G^1) if for all (possibly inefficient) distributions $\mathcal{G}$ over (possibly inefficient) two-stage adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ the following holds: If $\mathsf{Adv}^{\mathsf{lkindcpa}}_{\mathsf{wbe},\mathcal{G},\beta,m}(n)$ is non-negligible, then

$$\begin{aligned} &\mathsf{Adv}(\mathcal{R}^{\mathsf{O}_0[\mathcal{G}],\mathsf{O}_1[\mathcal{G}],\mathsf{O}_2[\mathcal{G}]}, G^0, G^1) \\ := &\left|\Pr\left[1 = \mathcal{R}^{\mathsf{O}_0[\mathcal{G}],\mathsf{O}_1[\mathcal{G}],\mathsf{O}_2[\mathcal{G}]} \to G^0(1^n)\right] - \Pr\left[1 = \mathcal{R}^{\mathsf{O}_0[\mathcal{G}],\mathsf{O}_1[\mathcal{G}],\mathsf{O}_2[\mathcal{G}]} \to G^1(1^n)\right]\right| \end{aligned}$$

$\mathsf{O}_0[\mathcal{G}](i, 1^n)$
if $\mathcal{A}_{i,1} = \bot$:
  $(\mathcal{A}_{i,1}, \mathcal{A}_{i,2}) \leftarrow\!\!\$\ \mathcal{G}(1^n)$
return ()

$\mathsf{O}_1[\mathcal{G}](i, K)$
if $\mathcal{A}_{i,1} = \bot$: **return** $\bot$
$lkg \leftarrow\!\!\$\ \mathcal{A}_{i,1}(K)$
return lkg

$\mathsf{O}_2[\mathcal{G}]^{\mathsf{encO}}(i, lkg)$
if $\mathcal{A}_{i,2} = \bot$: **return** $\bot$
$b^* \leftarrow\!\!\$\ \mathcal{A}_{i,2}^{\mathsf{encO}}(lkg)$
return b^*

Figure 2: Oracles given to black-box reductions which establish (β, m)-LK-IND-CPA security of wbe. The oracle encO is implemented by the caller of O_2.

is also non-negligible. See Fig. 2 for the behaviour of O_0, O_1, and O_2. We call such an algorithm $\mathcal{R}$ a black-box reduction.

The reduction $\mathcal{R}$ can query $\mathsf{O}_0[\mathcal{G}]$ with an integer i to sample a new pair of adversaries $\mathcal{A}_{i,1}$ and $\mathcal{A}_{i,2}$. It can then run $\mathsf{O}_1[\mathcal{G}](i, K)$ which returns $lkg \leftarrow\!\!\$\ \mathcal{A}_{i,1}(K)$. It can query $\mathsf{O}_1[\mathcal{G}]$ on as many inputs (i, K) as it likes, repeating both i and K. Similarly, $\mathcal{R}$ can query $\mathsf{O}_2[\mathcal{G}]$ on (i, lkg). It will then obtain several encryption requests from $\mathsf{O}_2[\mathcal{G}]$ and eventually obtain a bit b^*. Our proof allows rewinding. However, all oracles are stateless and it is w.l.o.g. to assume that $\mathcal{A}_{i,1}$ and $\mathcal{A}_{i,2}$ do not have randomness beyond the randomness that was used by $\mathcal{G}$ to generate them. Similarly, we model rewinding of the adversaries by being able to query them multiple times with different inputs.

We now define key-fixing, which roughly means that a polynomial number of ciphertexts uniquely determine the key with high probability. Formally, we say that a scheme is key-fixing if there exists an algorithm UniqueChecker which determines with good probability whether the key associated with a given set of ciphertexts is unique. The number of required ciphertexts to determine uniqueness depends on the key size m, because if their accumulated length is less than m, the ciphertexts cannot determine the key information-theoretically. Additionally, for local constructions, we need to have sufficiently many samples to access each key bit sufficiently frequently. We do not make the number of necessary ciphertexts explicit, but instead also allow it to depend on the scheme.

Definition 9 (Key-Fixing)**.** Let m, ℓ be polynomials in n. A symmetric-key encryption scheme $(\mathsf{enc}, \mathsf{dec})$ is (m, ℓ)-*key-fixing* if there is a PPT algorithm UniqueChecker such that the following hold:

Overwhelmingly Unique For any $K \in \{0,1\}^m$,

$$\Pr_{\forall i \in [m],\ c_i \leftarrow\!\!\$ \mathsf{enc}(K, 0^\ell)}[\mathsf{UniqueChecker}(c_1, \ldots, c_m) \neq 1]$$

is negligible in n.

Correctness For any PPT algorithm $\mathcal{R}$,

$$\Pr_{(K,(c_i)_{i=1}^m) \leftarrow\!\!\$ \mathcal{R}(1^n)} \left[\begin{array}{l} \mathsf{UniqueChecker}(c_1, \ldots, c_m) = 1 \\ \wedge\ \exists K' \neq K,\ \forall i \in [m],\ \mathsf{dec}(K, c_i) = \mathsf{dec}(K', c_i) = 0^\ell \end{array} \right]$$

is negligible in n.

A white-box encryption scheme $(\mathsf{wbkgen}, \mathsf{enc}, \mathsf{dec}, \mathsf{Kenc}, \mathsf{Kdec})$ is (m, ℓ)-key-fixing if the symmetric-key encryption scheme $(\mathsf{Kenc}, \mathsf{Kdec})$ is (m, ℓ)-key-fixing.

The key-fixing property above which we require for our impossibility result on strong incompressibility corresponds to Wichs' [Wic13] injectivity requirement for their impossibility result on leakage-resilient one-way functions. Essentially, key-fixing is a probabilistic version of injectivity, saying that for honestly generated ciphertexts, the probability that they fix the key is overwhelming. We remark that the choice of the all zero message 0^ℓ is arbitrary – it suffices for the property to hold for some arbitrarily fixed message sequence.

$\underline{\mathcal{G}(1^n)}$

$\mathsf{RO} \leftarrow\!\$\ \{f : \{0,1\}^m \to \{0,1\}^\beta\}$

return $(\mathcal{A}_1[\mathsf{RO}], \mathcal{A}_2[\mathsf{RO}])$

$\underline{\mathcal{A}_1[\mathsf{RO}](K)}$

$lkg \leftarrow \mathsf{RO}(K)$

return lkg

$\underline{\mathcal{A}_2[\mathsf{RO}]^{\mathsf{encO}}(lkg)}$

$\mathcal{C} \leftarrow \emptyset$

for $j \in [\ell]$:

  $c_j \leftarrow\!\$\ \mathsf{encO}(0^\ell, 0^\ell)$

if $\mathsf{UniqueChecker}(c_1, \ldots, c_\ell) \neq 1$: **return** 1

find K^* :

$$\begin{cases} \mathsf{RO}(K^*) = lkg \\ \forall j,\ 0^\ell = \mathsf{dec}(K^*, c_j) \end{cases}$$

if no such K^* : **return** 1

$c^* \leftarrow\!\$\ \mathsf{encO}(0^\ell, 1^\ell)$

if $\mathsf{dec}(K^*, c^*) = 0^\ell$:

  return 0

return 1

$\underline{\hat{\mathcal{G}}(1^n)}$

$T \leftarrow$ empty table

return $(\hat{\mathcal{A}}_1^T, \hat{\mathcal{A}}_2^T)$

$\underline{\hat{\mathcal{A}}_1^T(K)}$

if $T[K] = \perp$:

  $T[K] \leftarrow\!\$\ \{0,1\}^\beta$

$lkg \leftarrow T[K]$

return lkg

$\underline{\hat{\mathcal{A}}_2^{T,\mathsf{encO}}(lkg)}$

$\mathcal{C} \leftarrow \emptyset$

for $j \in [\ell]$:

  $c_j \leftarrow\!\$\ \mathsf{encO}(0^\ell, 0^\ell)$

if $\mathsf{UniqueChecker}(c_1, \ldots, c_\ell) \neq 1$: **return** 1

find K^* :

$$\begin{cases} T(K^*) = lkg \\ \forall j,\ 0^\ell = \mathsf{dec}(K^*, c_j) \end{cases}$$

if no such K^* : **return** 1

$c^* \leftarrow\!\$\ \mathsf{encO}(0^\ell, 1^\ell)$

if $\mathsf{dec}(K^*, c^*) = 0^\ell$:

  return 0

return 1

Figure 3: An inefficient stateless adversary distribution $\mathcal{G}$ (left) and its efficient stateful (with shared tables T) simulation (right), both parametrised by (m, ℓ, β).

3.2 Impossibility Result

Theorem 1. *Let m, ℓ be polynomials in n. Let* wbe *be a correct (m, ℓ)-key-fixing white-box encryption scheme. If $\beta = \omega(\log n)$, then there exists no black-box reduction which bases the (β, m)-LK-IND-CPA-security on a single-stage assumption, or the assumption is false.*

In other words, the above theorem states that for all PPT reductions $\mathcal{R}$ and all (true) single-state assumption modelled by a pair of games (G^0, G^1), the advantage $\mathsf{Adv}(\mathcal{R}^{\mathsf{O}_0, \mathsf{O}_1, \mathsf{O}_2}, G^0, G^1)$ is negligible. Of course, for an incorrect assumption, there might still be a successful adversary.

Proof of Theorem 1. Let the games (G^0, G^1) model a single-stage assumption. If the assumption (G^0, G^1) is false, then we are done. Below, assume that the assumption (G^0, G^1) is true.

Recall that $\beta = \omega(\log n)$ and wbe is correct and (m, ℓ)-key-fixing. Assume towards contradiction that there exists a PPT black-box reduction which bases the (β, m)-LK-IND-CPA-security of wbe on the assumption (G^0, G^1).

Consider the distribution $\mathcal{G}$ of inefficient stateless adversaries defined in the left column of Fig. 3, and the distribution $\hat{\mathcal{G}}$ of efficient stateful adversaries defined in the right column of Fig. 3. To prove Theorem 1, we will prove the following two claims:

Claim 1 (Successful Adversary). $\mathsf{Adv}^{\mathsf{lkindcpa}}_{\mathsf{wbe},\mathcal{G},\beta,m}(n)$ *is non-negligible.*

Claim 2 (Efficient Simulation). *For any PPT reduction* $\mathcal{R}$ *and all* $b \in \{0,1\}$*,*

$$\left|\Pr\left[1 = \mathcal{R}^{\mathsf{O}_0[\mathcal{G}],\mathsf{O}_1[\mathcal{G}],\mathsf{O}_2[\mathcal{G}]} \to G^b(1^n)\right] - \Pr\left[1 = \mathcal{R}^{\mathsf{O}_0[\hat{\mathcal{G}}],\mathsf{O}_1[\hat{\mathcal{G}}],\mathsf{O}_2[\hat{\mathcal{G}}]} \to G^b(1^n)\right]\right|$$

is negligible.

Assuming that both claims hold, and that $\mathcal{R}$ is a black-box reduction from the (β, m)-LK-IND-CPA security of wbe to the single-stage assumption modeled by (G^0, G^1), we conclude that

$$\left|\Pr\left[1 = \mathcal{R}^{\mathsf{O}_0[\hat{\mathcal{G}}],\mathsf{O}_1[\hat{\mathcal{G}}],\mathsf{O}_2[\hat{\mathcal{G}}]} \to G^0(1^n)\right] - \Pr\left[1 = \mathcal{R}^{\mathsf{O}_0[\hat{\mathcal{G}}],\mathsf{O}_1[\hat{\mathcal{G}}],\mathsf{O}_2[\hat{\mathcal{G}}]} \to G^1(1^n)\right]\right|$$

is non-negligible in n, which contradicts the single-stage assumption. It remains to prove the two claims.

Proof of Claim 1. We argue that $\mathcal{G}$ almost always breaks (β, m)-LK-IND-CPA-security when directly interacting with the security game $\mathsf{lkindcpa}^b_{\mathsf{wbe},\mathcal{G},\beta,m}(n)$, i.e. the advantage $\mathsf{Adv}^{\mathsf{lkindcpa}}_{\mathsf{wbe},\mathcal{G},\beta,m}(n)$ is overwhelming. For sanity check, we note that for any $(\mathcal{A}_1, \mathcal{A}_2)$ generated by $\mathcal{G}$, $\mathcal{A}_1$ has output length exactly β. It therefore remains to show that $\mathcal{A}_2$ guesses the hidden bit b correctly except with negligible probability.

We first note that, by the *overwhelmingly unique* condition of key-fixing (Definition 9),

$$\mathsf{UniqueChecker}(c_1, \ldots, c_\ell) = 1$$

with overwhelming probability. Next, by the *correctness* condition of key-fixing, there is no second $K' \neq K^*$ which could have yielded the same ciphertexts. Hence, K must be equal to K^*. Finally, by correctness of the encryption scheme, we must have $\mathsf{dec}(K^*, c^*) = b^\ell$.

Proof of Claim 2. Observe that the only difference between $\mathcal{G}$ and $\hat{\mathcal{G}}$ is that the random function hardwired in $(\mathcal{A}_1, \mathcal{A}_2)$ is simulated by lazy sampling in $(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$. Define $\mathcal{R} := \mathcal{R}^{\mathsf{O}_0[\mathcal{G}],\mathsf{O}_1[\mathcal{G}],\mathsf{O}_2[\mathcal{G}]} \to G^b(1^n)$ and define $\hat{\mathcal{R}}$ analogously. We observe that $\mathcal{R}$ and $\hat{\mathcal{R}}$ produce identically distributed outputs except when there exists a tuple (i, K^*, lkg) where all four following events and conditions are satisfied:

- $\mathcal{R}$ did not query $\mathsf{O}_1[\mathcal{G}](i, K^*)$,
- $\mathcal{R}$ queried $\mathsf{O}_2[\mathcal{G}](i, \mathit{lkg})$,
- $\mathsf{RO}(K^*) = \mathit{lkg}$,
- $\mathsf{UniqueChecker}(c_1, \ldots, c_\ell) = 1$

where RO is the random function hardwired to the i-th instance of $(\mathcal{A}_1, \mathcal{A}_2)$.

Since $\mathsf{UniqueChecker}(c_1, \ldots, c_\ell) = 1$, by the correctness property of key-fixing (Definition 9), there is only a unique key K^* which is consistent with these ciphertexts. Since $\mathcal{R}$ did not query $\mathsf{O}_1[\mathcal{G}](i, K^*)$, the value $\mathsf{RO}(K^*)$ for this unique key K^* is information-theoretically hidden from $\mathcal{R}$ and thus the probability that the value lkg chosen by $\mathcal{R}$ is equal to $\mathsf{RO}(K^*)$ is negligible.

4 CBC-Mode with a Random Permutation is Key-Fixing.

We now prove that CBC mode is key-fixing when modelling the cipher as a (family of) truly random permutation π, i.e. for any key $k \in \{0,1\}^n$, we treat $\pi(k,\cdot)$ as a random permutation and $\pi^{-1}(k,\cdot)$ its inverse. We first prove this statement for the case where the key is not large (Section 4.1) and then handle the case with a large key (Section 4.2). In order to formalise this statement, we consider an FKKM-like construction where, for each encryption, a number of random indices into the large key are sampled, run through an extractor (which we implement by a random matrix), and then used as a key for CBC-mode (Section 4.2). This second result proves key-fixing for one example of obtaining small keys from large keys. Similar results can be proven for other natural approaches using different extractors as well, as long as all inputs bits are treated equally.

4.1 Key-Fixing of Small-Key CBC-mode

We start by describing CBC-mode. Since the length of some messages is not divisible by n, we first need to encode the message into a multiple of n.

Definition 10 (Encoding scheme)**.** We call two functions $\mathsf{encode}_n : \{0,1\}^* \to \{0,1\}^*$ and $\mathsf{decode}_n : \{0,1\}^* \to \{0,1\}^*$ an *encoding scheme* if the following hold:

- encode_n and decode_n are computable in time polynomial in n and the input length.
- decode_n is the inverse of encode_n, i.e. for all $x \in \{0,1\}^*$, decode_n can recover x from $\mathsf{encode}_n(x)$, that is, we have $\mathsf{decode}_n(\mathsf{encode}_n(x)) = x$. In particular, encode_n is *injective*, i.e. if $x \neq x'$, then $\mathsf{encode}_n(x) \neq \mathsf{encode}_n(x')$.
- the length encode_n only depends on the length of the input, i.e. if $|x| = |x'|$, then $|\mathsf{encode}_n(x)| = |\mathsf{encode}_n(x')|$.
- For all $x \in \{0,1\}^*$, $|\mathsf{encode}_n(x)|$ is divisible by n.

We can construct a CBC-mode encryption scheme $\mathsf{ske}_{\pi,\mathtt{CBC}} = (\mathsf{enc}, \mathsf{dec})$ as follows:

$\mathsf{enc}(k, msg)$	$\mathsf{dec}(k, ctxt)$
$n \leftarrow \lvert k\rvert$	$n \leftarrow \lvert k\rvert$
$msg' \leftarrow \mathsf{encode}_n(msg)$	$\ell \leftarrow \frac{\lvert ctxt\rvert}{n} - 1$
$\mathsf{nc} \leftarrow\!\!\$\ \{0,1\}^n$	$(c_0, \ldots, c_\ell) \leftarrow ctxt$
$c_0 \leftarrow \mathsf{nc}$	**for** $i \in [\ell]$
$\ell \leftarrow \frac{\lvert msg'\rvert}{n}$	$\quad x_i \leftarrow c_{i-1} \oplus \pi^{-1}(k, c_i)$
for $i \in [\ell]$	$msg' \leftarrow x_1 \lvert\rvert \ldots \lvert\rvert x_\ell$
$\quad x_i \leftarrow msg'_{(i-1)n+1\ldots in}$	$msg \leftarrow \mathsf{decode}_{\lvert k\rvert}(msg')$
$\quad c_i \leftarrow \pi(k, x_i \oplus c_{i-1})$	**return** msg
$ctxt \leftarrow (c_0, \ldots, c_\ell)$	
return $ctxt$	

Theorem 2. *For $m = n$ and $\ell = 3n$, $\mathsf{ske}_{\pi,\mathtt{CBC}}$ is (m,ℓ)-key-fixing.*

Proof. Define $\mathsf{UniqueChecker}(c_1, \ldots, c_\ell)$ to always return 1. Therefore, the *overwhelmingly uniqueness* condition is satisfied. We now turn to correctness.

We prove that a PPT adversary $\mathcal{R}$ cannot come up with a ciphertext c_1 and a key k such that there exists a key k' such that $\mathsf{dec}(k, c_1) = \mathsf{dec}_{k'}(c_1) = 0^\ell$. Namely, for every

nonce nc and keys k and k', there is a probability of 2^{-n} that $\pi_k(\mathsf{nc}) = \pi_{k'}(\mathsf{nc})$. Now, let $y := \pi_k(\mathsf{nc})$ and assume that $y \neq \mathsf{nc}$. Then, again, the probability that $\pi_k(y) = \pi_{k'}(y)$ is 2^{-n}. Letting $y' := \pi_k(y)$ and again assuming that $y' \notin \{y, \mathsf{nc}\}$, the probability that $\pi_k(y') = \pi_{k'}(y')$ is 2^{-n}. And and we get the same probability for the next ciphertext block $y'' := \pi_k(y')$ Thus, for triple (k, k', nc) such that $|\{y'', y', y, \mathsf{nc}\}| = 4$, we have that

$$\Pr_\pi[\mathsf{dec}(k, c_1) = \mathsf{dec}_{k'}(c_1)] = 2^{-4n}.$$

Taking a union bound over all nonces and pairs of distinct keys (nc, k, k'), the probability over π that such a nonce and pair (k, k') exists, is 2^{-n}. Moreover, in a polynomial number of queries, $\mathcal{R}$ will not be able to find k and nc such that $\{\mathsf{nc}, y, y'\}$ contains a collision, so that $|\{y'', y', y, \mathsf{nc}\}| = 4$ for all values that the reduction can compute. This concludes the proof of Theorem 2.

4.2 Key-Fixing of Big-Key CBC-Mode

We now construct a big-key symmetric encryption scheme $\mathsf{bke}_{\pi,\mathtt{CBC}} = (\mathsf{Kenc}, \mathsf{Kdec})$ by augmenting the $\mathsf{ske}_{\pi,\mathtt{CBC}}$ scheme constructed above. In the construction below, we derive a subkey by multiplying a random binary matrix M to it. This gives us a subkey of conventional length, which we use for running the CBC-based encryption scheme from the previous subsection. Note that the construction below can equally be turned into an incompressible white-box encryption if the large key K is itself derived from a small key of conventional length.

$\mathsf{Kenc}(K, msg)$	$\mathsf{Kdec}(K, c)$
$M \leftarrow\!\!\$\ \{0,1\}^{m \times n}$	$(M, c') \leftarrow c$
$k \leftarrow K \cdot M$	$k \leftarrow K \cdot M$
$\mathsf{nc} \leftarrow\!\!\$\ \{0,1\}^n$	
$c' \leftarrow\!\!\$\ \mathsf{ske}_{\pi,\mathtt{CBC}}.\mathsf{enc}(k, msg, \mathsf{nc})$	$msg \leftarrow \mathsf{ske}_{\pi,\mathtt{CBC}}.\mathsf{dec}(k, c', \mathsf{nc})$
$c \leftarrow (M, c')$	
return c	**return** msg

Theorem 3. *For $m \geq n$ a polynomial in n and $n\ell \geq 2m - 1$, $\mathsf{bke}_{\pi,\mathtt{CBC}}$ is (m, ℓ)-key-fixing.*

Proof. As in the proof of Theorem 2, we construct $\mathsf{UniqueChecker}$ so that it almost always returns 1, but it now performs an additional check. Concretely, let M_i denote the matrix specified in c_i. $\mathsf{UniqueChecker}$ additionally checks that the matrix $\hat{M} := M_1 || \ldots || M_\ell$ is of full rank m over the binary field $\{0, 1\}$. If so, $\mathsf{UniqueChecker}$ outputs 1. Else, it outputs 0. For a random m-by-t matrix over $\{0, 1\}$ where $t \geq m$, the probability that the matrix is of full rank m is at least $1 - m/2^{t-m+1}$. Setting $t = n\ell$, the probability that $\hat{M}$ is of full rank m is at least $1 - m/2^m$ which is overwhelming in n. Thus the *overwhelmingly unique* property holds.

For correctness of $\mathsf{UniqueChecker}$, observe that (a) the analysis of Theorem 2 now applies to each *extracted* key individually. Bootstrapping from the individual keys and using the full-rank matrix $\hat{M}$, we then obtain uniqueness of the entire big-key K.

Remark. We remark that the above impossibility can be extended to the case where M is chosen to be a random sparse matrix (so that, for example, each encryption/decryption only depends on $\kappa \ll m$ bits of the big key K) as long as $\hat{M} = M_1 || \ldots || M_\ell$ is of full rank m with overwhelming probability for some ℓ polynomial in m.

$\mathsf{ENC}-\mathsf{COM}^{b}_{\mu,\mathcal{A}}(1^n)$	$\mathsf{encO}(m)$
$k \leftarrow\!\$\ \mathsf{kgen}(1^n)$	$c \leftarrow\!\$\ \mathsf{Kenc}(m)$
$K \leftarrow\!\$\ \mathsf{wbkgen}(k)$	**return** c
$\mathsf{leak}(\cdot) \leftarrow\!\$\ \mathcal{A}(1^n)$	
$lkg \leftarrow\!\$\ \mathsf{leak}(K)$	$\mathsf{encO}_b(m_0, m_1)$
$b' \leftarrow\!\$\ \mathcal{A}^{\mathsf{encO},\mathsf{encO}_b}(lkg)$	**if** $\lvert m_0\rvert = \lvert m_1\rvert$
$b'' \leftarrow (H_\infty(K\lvert lkg)) \geq \mu)$	$c_b \leftarrow\!\$\ \mathsf{Kenc}(m_b)$
return $b' \wedge b''$	**return** c_b

Figure 4: Strong incompressibility model.

$\mathsf{LIND}^{b}_{\beta,\mathcal{A}}(1^n)$	$\mathsf{encO}_b(m_0, m_1)$
$\mathsf{leak}(\cdot) \leftarrow\!\$\ \mathcal{A}(1^n)$	**if** $\lvert m_0\rvert = \lvert m_1\rvert$
$K \leftarrow\!\$\ \mathsf{kgen}(1^n)$	$c_b \leftarrow\!\$\ \mathsf{Kenc}(K, m_b)$
$lkg \leftarrow \mathsf{leak}(K)$	**return** c_b
$b' \leftarrow\!\$\ \mathcal{A}^{\mathsf{encO}_b}(lkg)$	
$b \leftarrow \lvert lkg\rvert \leq \beta$	
return $b' \wedge b''$	

Figure 5: LIND security

5 Equivalence of Strong Incompressibility and Big-Key Encryption Security

In this section, we show that the strong incompressibility model of FKKM ([FKKM16a, Definition 4] or Definition 11 below) and the LIND model of BKR ([BKR16, Figure 10] or Definition 11 below) are equivalent to the LK-IND-CPA model when choosing appropriate leakage classes. We thus show formally that our impossibility result from Section 3 does not only apply to incompressible white-box cryptography, but also to big-key symmetric encryption and any other leakage-resilient cryptographic schemes whose security is defined via LK-IND-CPA.

Definition 11 (Strong Incompressibility [FKKM16a, Definition 4])**.** A WBE-scheme is μ-strongly incompressible (ENC-COM) if for any PPT stateful adversary $\mathcal{A}$ the following advantage

$$\left|\Pr\left[1 = \mathsf{ENC}-\mathsf{COM}^{0}_{\mu,\mathcal{A}}(1^n)\right] - \Pr\left[1 = \mathsf{ENC}-\mathsf{COM}^{1}_{\mu,\mathcal{A}}(1^n)\right]\right|,$$

is negligible, where the experiment $\mathsf{ENC}-\mathsf{COM}^{b}_{\mu,\mathcal{A}}(1^n)$ runs as described in Section 5.

Definition 12 (LIND-security [BKR16, Figure 10])**.** A BKE-scheme is β-LIND-secure if for any PPT stateful adversary $\mathcal{A}$ the following probability is negligible,

$$\Pr\left[1 = \mathsf{LIND}^{0}_{\beta,\mathcal{A}}(1^n)\right] - \Pr\left[1 = \mathsf{LIND}^{1}_{\beta,\mathcal{A}}(1^n)\right],$$

where Fig. 5 defines $\mathsf{LIND}^{b}_{\beta,\mathcal{A}}(1^n)$.

Our previously defined LK-IND-CPA game defines an upper bound β on the leakage. Instead, LK-IND-CPA can also consider adversary pairs $(\mathcal{A}_1, \mathcal{A}_2)$ which ensure that the

min-entropy of K conditioned on the leakage is greater than some value μ. With a strict upper bound β as defined previously, LK-IND-CPA is equivalent to LIND security, and with a min-entropy bound, LK-IND-CPA security is equivalent to strong incompressibility. We now show that if a scheme is LK-IND-CPA secure w.r.t. a strict upper bound β, then it is also LK-IND-CPA secure with a closely related upper bound on the leakage resulting from the following claim.

Claim 3 (Relation between leakage classes)**.** *Let* leak *be such that* $|\mathsf{leak}(K)| \leq \beta$ *for all* $K \in \{0,1\}^m$*. Let* $\delta = \omega(\log n)$*. Given leakage* $lkg := \mathsf{leak}(K)$*, the min-entropy* $H_\infty(K|lkg)$ *is at least* $m - \beta - \delta$ *except with negligible probability over the choice of* K*.*

Proof. The claim follows by a counting argument. There are 2^m values of K and at most 2^β leakage values of lkg. For any fixed leakage lkg, write

$$\mathcal{K}_{lkg} := \{K : \mathsf{leak}(K) = lkg\}.$$

Let $\mathcal{S}$ be the set of lkg such that

$$|\mathcal{K}_{lkg}| < 2^{m-\beta-\delta}.$$

Now, the union

$$\bigcup_{lkg \in \mathcal{S}} \mathcal{K}_{lkg}$$

contains at most

$$|\mathcal{S}| \cdot 2^{m-\beta-\delta} < 2^{m-\delta}$$

elements. Thus, the fraction of keys K which lead to leakage that has too low min-entropy is at most

$$\frac{2^{m-\delta}}{2^m} = 2^{-\delta},$$

which is negligible, which concludes the proof.

Therefore, when an adversary is a valid adversary w.r.t. leakage bounded by β, then the adversary is also a valid adversary w.r.t. min-entropy $|K| - \beta - \delta$ for any $\delta = \omega(\log n)$. Hence, our impossibility result also applies to models which consider min-entropy.

6 Discussion

We now discuss consequences of our impossibility result.

Implication to FKKM's Strong Incompressibility Model. FKKM [FKKM16a] use a symmetric-key cipher, e.g. AES, to generate large look-up tables as well as their input queries. These tables are intended as an incompressible key generator for deriving subkeys, to be fed into a PRG or any conventional encryption scheme for encrypting messages.

In their original paper, FKKM claim that this composition yields LK-IND-CPA-security based on the security of the underlying cipher. However, this contradicts our impossibility result, and indeed, FKKM did not provide arguments to support the claim. More concretely, the security proof only argues about *indistinguishability of keys*, but the reduction does not emulate encryption oracles. Therefore the security argument is insufficient for proving LK-IND-CPA-security, but only a weaker (or incomparable) model of security. Inspired by our impossibility result, FKKM revisited their security analysis [FKKM16b, Appendix D] and now prove that their key generator can be combined with a length-expanding *random oracle* to yield an LK-IND-CPA-secure encryption scheme in the random oracle model. The use of idealised models circumvents our impossibility result.

(Non-)Implication to Other Security Models. Our impossibility result does not apply to weaker incompressibility models *without* encryption oracles, as considered, for example, in [DLPR14, BI15, BIT16, BBK14, CCD+17, AAB+19, KLLM20, KI21]. Although we present our negative result in the context of incompressible white-box cryptography, it readily generalises to any leakage-resilient encryption scheme whose security implies LK-IND-CPA-security, such as big-key symmetric encryption. For completeness, we showed formally that both FKKM's strong incompressibility and the BKR's LIND-security are equivalent to LK-IND-CPA-security (cf. Section 5).

Circumventing our Impossibility Result. Given that our impossibility result rules out proving key-fixing schemes secure under single-stage assumptions in the standard model, natural approaches towards circumvention include:

- proving security in an idealised model such as the random oracle model (ROM),
- proving security under two-stage assumptions, or
- designing a non-key-fixing scheme.

Using Random Oracles or Two-Stage Assumptions. Highly efficient encryption with provable LK-IND-CPA-security with large leakage was achieved in the context of big-key encryption in BKR [BKR16] and a follow-up work by Bellare and Dai (BD [BD17]) in the random oracle model. In particular, BKR and BD seek to reduce the number of (blocks of) bits accessed by, i.e. the *locality* of, the encryption algorithm to securely derive a subkey, where the former is modelled as a function of (1) the key size and (2) the output length of the adversarially chosen leakage function.

BKR and BD further show that the random oracle can be *instantiated* using universal computational extractors (UCE) [BHK13]. UCEs are a strong two-stage-assumption for hash-functions which, in some cases, can be instantiated from indistinguishability obfuscation [BFM14] in a provably secure, yet inefficient way. As of now, no practically efficient, provably secure constructions of UCEs based on standard assumptions are known.

Although the BKR construction of big-key encryption falls short of being a white-box encryption scheme due to the lack of functionally equivalent small keys, the latter can be easily added by deriving the big key from a small key using a PRG.

Using Non-Key-Fixing Constructions. A very exciting feasibility result for building LK-IND-CPA-secure encryption schemes under the mere assumption of one-way functions was provided by HLWW [HLWW13]. While "natural" encryption schemes tend to satisfy key-fixing, as illustrated by our result for CBC-encryption in Section 4, HLWW demonstrate that there are *meaningful* ways of introducing redundancy which allows to prove leakage-resilience beyond $\log n$ many bits and thus circumvent our impossibility result.

Crucially, their scheme has key material which is *never* accessed for the generation of honest ciphertexts. A sequence of subtle game-hops then moves to a situation where, for the challenge ciphertext, the additional key material is accessed, so that one obtains information-theoretic security for the challenge ciphertext. The difficult argument is to show that this modified challenge ciphertext which accesses the additional information is indistinguishable from a real ciphertext, using *hash-proof systems*. Specifically, HLWW introduce *symmetric-key weak hash proof systems* (wHPS), which can be seen as a special type of PRFs which can take as input values from *valid* and *invalid* distributions. Given multiple (valid) input-output pairs and one random invalid input, the corresponding output on the invalid input should be uniformly random and statistically independent from the previously obtained input-output pairs.

HLWW construct wHPS from weak PRFs (wPRFs) and show how wHPS can be used for constructing leakage resilient wPRFs by simply applying a randomness extractor to

the output of the wHPS. Then, given such leakage-resilient wPRF, messages can easily be encrypted in an LK-IND-CPA-secure way by padding them with outputs of the leakage-resilient wPRF. Crucially, a leakage-resilient wPRF constructed via the above complicated process is not key-fixing, which circumvents our impossibility result.

The HLWW construction as outlined above achieves a moderate leakage rate of $\frac{\log(n)}{n}$ and uses the complete secret key for each encryption. HLWW show that using t parallel repetitions of the above construction and random sampling yields a big-key encryption scheme with significantly improved locality while retaining the leakage rate $\frac{t \cdot \log(n)}{t \cdot n} = \frac{\log(n)}{n}$. Recently, Quach, Waters and Wichs (QWW) [QWW21] provide a construction secure against the same leakage rate, based on pseudo-entropy functions which are derived from targeted lossy functions based on injective PRGs. Unlike HLWW, the *ciphertext size* in QWW do not increase with the leakage bound, but the secret key size does.

Summary. Amongst all options for circumventing our impossibility result, it seems that opting for a weaker security model for incompressibility is not advisable, because the encryption oracle is close to a real-life attacker capability[3]. However, when seeking to prevent code-lifting attack [Bre12], if the hardware supports it, a promising alternative is to aim for *hardware-binding* [CdRP14, SdHM15, BBIJ17, AABM20], since provably secure constructions are feasible in this domain [ABCW23, ABF+20].

As HLWW and QWW show, developing tailor-made ciphers which are provably strongly incompressible under standard assumptions is theoretically feasible, but at present remains impractical. While this might change at some point in the future, the BKR and BD approaches of using strong (two-stage) assumptions on hash-functions, or the BKR, BD, and FKKM approaches to use a random oracle seem to lead to more practical constructions.

Additionally, it might be feasible to mix ideal-model analysis and cryptanalysis for a higher degree of confidence. For example, hash-functions are usually presented with a proof of indifferentiability, assuming only that smaller building blocks are ideal, and a similar style of results might be possible and desirable for incompressibility. In particular, this latter avenue potentially allows us to still prove security for *standard* encryption schemes.

In summary, from our perspective, one should either aim for device-binding instead of incompressibility, or, if one needs to aim for incompressibility, then the most promising way to achieve high-confidence security of real-life constructions is to employ a mix of cryptanalysis and ideal-model analysis.

Acknowledgement

We thank the CHES reviewers and, in particular, our shepherd Sebastian Berndt for very useful comments on the presentation and, in particular, on suggestions for explaining the proof technique of our impossibility result.

References

[AAB+19] Estuardo Alpirez Bock, Alessandro Amadori, Joppe W. Bos, Chris Brzuska, and Wil Michiels. Doubly half-injective PRGs for incompressible white-box cryptography. In Mitsuru Matsui, editor, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 189–209. Springer, Heidelberg, March 2019.

[AABM20] Estuardo Alpirez Bock, Alessandro Amadori, Chris Brzuska, and Wil Michiels. On the security goals of white-box cryptography. *IACR TCHES*, 2020(2):327–

[3] Realistically, ciphertexts can be obtained for partially known messages, if not for adversarially chosen messages. Conceptually, our argument also applies to many such weaker kinds of encryption oracles.

357, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8554.

[ABCW23] Shashank Agrawal, Estuardo Alpirez Bock, Yilei Chen, and Gaven J. Watson. White-box cryptography with global device binding from message-recoverable signatures and token-based obfuscation. In Elif Bilge Kavun and Michael Pehl, editors, *Constructive Side-Channel Analysis and Secure Design - 14th International Workshop, COSADE 2023, Munich, Germany, April 3-4, 2023, Proceedings*, volume 13979 of *Lecture Notes in Computer Science*, pages 241–261. Springer, 2023.

[ABF+20] Estuardo Alpirez Bock, Chris Brzuska, Marc Fischlin, Christian Janson, and Wil Michiels. Security reductions for white-box key-storage in mobile payments. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 221–252. Springer, Heidelberg, December 2020.

[BBIJ17] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, and Martin Bjerregaard Jepsen. Analysis of software countermeasures for whitebox encryption. *IACR Trans. Symm. Cryptol.*, 2017(1):307–328, 2017.

[BBK14] Alex Biryukov, Charles Bouillaguet, and Dmitry Khovratovich. Cryptographic schemes based on the ASASA structure: Black-box, white-box, and public-key (extended abstract). In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 63–84. Springer, Heidelberg, December 2014.

[BD17] Mihir Bellare and Wei Dai. Defending against key exfiltration: Efficiency improvements for big-key cryptography via large-alphabet subkey prediction. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 923–940. ACM Press, October / November 2017.

[BDF+18] Chris Brzuska, Antoine Delignat-Lavaud, Cédric Fournet, Konrad Kohbrok, and Markulf Kohlweiss. State separation for code-based game-playing proofs. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 222–249. Springer, Heidelberg, December 2018.

[BFM14] Christina Brzuska, Pooya Farshim, and Arno Mittelbach. Indistinguishability obfuscation and UCEs: The case of computationally unpredictable sources. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 188–205. Springer, Heidelberg, August 2014.

[BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 398–415. Springer, Heidelberg, August 2013.

[BI15] Andrey Bogdanov and Takanori Isobe. White-box cryptography revisited: Space-hard ciphers. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 1058–1069. ACM Press, October 2015.

[BIT16] Andrey Bogdanov, Takanori Isobe, and Elmar Tischhauser. Towards practical whitebox cryptography: Optimizing efficiency and space hardness. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 126–158. Springer, Heidelberg, December 2016.

[BKR16] Mihir Bellare, Daniel Kane, and Phillip Rogaway. Big-key symmetric encryption: Resisting key exfiltration. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 373–402. Springer, Heidelberg, August 2016.

[Bre12] Brecht Wyseur. White-box cryptography: hiding keys in software. MISC Magazine, 2012. http://www.whiteboxcrypto.com/index.php.

[CCD+17] Jihoon Cho, Kyu Young Choi, Itai Dinur, Orr Dunkelman, Nathan Keller, Dukjae Moon, and Aviya Veidberg. WEM: A new family of white-box block ciphers based on the Even-Mansour construction. In Helena Handschuh, editor, *CT-RSA 2017*, volume 10159 of *LNCS*, pages 293–308. Springer, Heidelberg, February 2017.

[CdRP14] Tim Cooijmans, Joeri de Ruiter, and Erik Poll. Analysis of secure key storage solutions on android. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, SPSM '14, pages 11–20. ACM, 2014.

[CEJv03] Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. White-box cryptography and an AES implementation. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 250–270. Springer, Heidelberg, August 2003.

[CEJvO03] Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. A white-box DES implementation for DRM applications. In Joan Feigenbaum, editor, *Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002*, volume 2696 of *LNCS*, pages 1–15. Springer, 2003.

[DLPR14] Cécile Delerablée, Tancrède Lepoint, Pascal Paillier, and Matthieu Rivain. White-box security notions for symmetric encryption schemes. In Tanja Lange, Kristin Lauter, and Petr Lisonek, editors, *SAC 2013*, volume 8282 of *LNCS*, pages 247–264. Springer, Heidelberg, August 2014.

[EMV19] EMVCo. Emv mobile payment: Software-based mobile payment security requirements, 2019. https://www.emvco.com/wp-content/uploads/documents/EMVCo-SBMP-16-G01-V1.2_SBMP_Security_Requirements.pdf.

[FKKM16a] Pierre-Alain Fouque, Pierre Karpman, Paul Kirchner, and Brice Minaud. Efficient and provable white-box primitives. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 159–188. Springer, Heidelberg, December 2016.

[FKKM16b] Pierre-Alain Fouque, Pierre Karpman, Paul Kirchner, and Brice Minaud. Efficient and provable white-box primitives. Cryptology ePrint Archive, Report 2016/642, 2016. https://eprint.iacr.org/2016/642.

[HITY22] Akinori Hosoyamada, Takanori Isobe, Yosuke Todo, and Kan Yasuda. A modular approach to the incompressibility of block-cipher-based AEADs. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 585–619. Springer, Heidelberg, December 2022.

[HLWW13] Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 160–176. Springer, Heidelberg, May 2013.

[KI21] Yuji Koike and Takanori Isobe. Yoroi: Updatable whitebox cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):587–617, 2021.

[KLLM20] Jihoon Kwon, ByeongHak Lee, Jooyoung Lee, and Dukjae Moon. FPL: White-box secure block cipher using parallel table look-ups. In Stanislaw Jarecki, editor, *CT-RSA 2020*, volume 12006 of *LNCS*, pages 106–128. Springer, Heidelberg, February 2020.

[QWW21] Willy Quach, Brent Waters, and Daniel Wichs. Targeted lossy functions and applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 424–453, Virtual Event, August 2021. Springer, Heidelberg.

[RZ18] Phillip Rogaway and Yusi Zhang. Simplifying game-based definitions - indistinguishability up to correctness and its application to stateful AE. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 3–32. Springer, Heidelberg, August 2018.

[SdHM15] Eloi Sanfelix, Job de Haas, and Cristofaro Mune. Unboxing the white-box: Practical attacks against obfuscated ciphers. Presentation at BlackHat Europe 2015, 2015. https://www.blackhat.com/eu-15/briefings.html.

[Sma14] Smart Card Alliance Mobile and NFC Council. Host card emulation 101. white paper, 2014. http://www.smartcardalliance.org/downloads/HCE-101-WP-FINAL-081114-clean.pdf.

[SWP09] Amitabh Saxena, Brecht Wyseur, and Bart Preneel. Towards security notions for white-box cryptography. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Agostino Ardagna, editors, *ISC 2009*, volume 5735 of *LNCS*, pages 49–58. Springer, Heidelberg, September 2009.

[Wic13] Daniel Wichs. Barriers in cryptography with weak, correlated and leaky sources. In Robert D. Kleinberg, editor, *ITCS 2013*, pages 111–126. ACM, January 2013.

[Wys11] Brecht Wyseur. White-box cryptography. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security, 2nd Ed*, pages 1386–1387. Springer, 2011.