A Modular Approach to the Incompressibility of Block-Cipher-Based AEADs

Akinori Hosoyamada[1], Takanori Isobe[2,3,4], Yosuke Todo[1][0000−0002−6839−4777], and Kan Yasuda[1]

[1] NTT Social Informatics Laboratories, Tokyo, Japan
{akinori.hosoyamada.bh,yosuke.todo.xt,kan.yasuda.hy}@hco.ntt.co.jp
[2] University of Hyogo, Hyogo, Japan takanori.isobe@ai.u-hyogo.ac.jp
[3] National Institute of Information and Communications Technology, Tokyo, Japan
[4] PRESTO, Japan Science and Technology Agency, Tokyo, Japan

Abstract. Incompressibility is one of the most fundamental security goals in white-box cryptography. Given recent advances in the design of efficient and incompressible block ciphers such as SPACE, SPNbox and WhiteBlock, we demonstrate the feasibility of reducing incompressible AEAD modes to incompressible block ciphers. We first observe that several existing AEAD modes of operation, including CCM, GCM(-SIV), and OCB, would be all insecure against white-box adversaries even when used with an incompressble block cipher. This motivates us to revisit and formalize incompressibility-based security definitions for AEAD schemes and for block ciphers, so that we become able to design modes and reduce their security to that of the underlying ciphers. Our new security notion for AEAD, which we name whPRI, is an extension of the pseudo-random injection security in the black-box setting. Similar security notions are also defined for other cryptosystems such as privacy-only encryption schemes. We emphasize that whPRI ensures quite strong authenticity against white-box adversaries: existential unforgeability beyond leakage. This contrasts sharply with previous notions which have ensured either no authenticity or only universal unforgeability. For the underlying ciphers we introduce a new notion of whPRP, which extends that of PRP in the black-box setting. Interestingly, our incompressibility reductions follow from a variant of public indifferentiability. In particular, we show that a practical whPRI-secure AEAD mode can be built from a whPRP-secure block cipher: We present a SIV-like composition of the sponge construction (utilizing a block cipher as its underlying primitive) with the counter mode and prove that such a construction is (in the variant sense) public indifferentiable from a random injection. To instantiate such an AEAD scheme, we propose a 256-bit variant of SPACE, based on our conjecture that SPACE should be a whPRP-secure cipher.

Keywords: symmetric-key cryptography · white-box cryptography · incompressibility · mode of operation · public indifferentiability

1 Introduction

White-box cryptography, which has been introduced by Chow et al. for AES [25] and DES [26], is a technique to protect data in the presence of adversaries who have access to implementations of cryptographic algorithms. For two decades since Chow et al. published the seminal papers, target systems of white-box cryptography have spread out from digital rights management (DRM) to mobile payment and banking services [2, 41]. Today white-box cryptography is applied to a wide range of cryptographic algorithms [29], and in this paper we focus on symmetric-key encryption schemes.

Secure white-box implementations must resist key extraction and "code lifting" [29]. While the goal of key extraction is to retrieve a secret key from a white-box implementation, code lifting tries to isolate and copy (a part of) the functionality of the cryptographic algorithm. Security against code lifting is in general stronger than security against key extraction, as key extraction implies code lifting of the full functionality. Preventing code lifting is indispensable to realize secure white-box implementations because arbitrary message can be encrypted or decrypted once the program is copied.

Delerablée et al. [29] have introduced the notion of incompressibility to formalize resistance to code lifting. Roughly, a white-box program of an encryption scheme is incompressible if it is infeasible to compress the encryption program while keeping its functionality. Delerablée et al. have shown that incompressibility is achievable by an RSA-group-based construction. Follow-up work by Fouque et al. [35] has introduced variants of incompressibility regarding privacy (IND-COM) or limited authenticity of universal unforgeability (ENC-COM). They have presented randomized schemes ensuring each of the security notions but not both at the same time.[5] The more recent work by Bock et al. [18] has shown that an incompressible randomized encryption scheme can be built from one-way permutations. Closely related to incompressibility is the work by Bellare et al. on big-key symmetric encryption [9][6], which was later improved by Bellare and Dai [8]. They have provided efficient randomized encryption schemes with a high level of privacy (LIND) and without authenticity, in the setting where information of the key is partially leaked, by making the key big, say, 1GB.

While there exist other white-box security notions, we focus on incompressibility because it is achievable by relatively efficient schemes and without relying on special hardware. True that trusted execution environments are in common use today, but demands for software-only solutions are still high in various scenarios—e.g., cloud servers providing digital rights management based services, mobile phones running cloud-based payment services with host card emulations, and memory-leakage resilient software—as listed by Bogdanov et al. [22].

It should be noted that some pieces of previous work [21, 9, 35, 22] (and we also do) assume that a black-box adversary resides outside the target program

[5] A scheme in Section 2 of the paper [35] achieves authenticity but not privacy in the white-box setting, because its tag-generation part does not depend on keys.

[6] This work focuses on bounded retrieval model rather than white-box cryptography, but as Fouque et al. point out, its security notion almost matches IND-COM.

and tries to attack in the conventional sense. More precisely, the white-box adversary, which here we call a *lifter*, tries to isolate and copy the functionality of the encryption program. Then, the black-box adversary tries to break privacy and/or authenticity with the aid of leakage generated by the lifter. Here, the amount of leakage is properly restricted in agreement with the bounded retrieval model [27, 24] of leakage-resilient cryptography.

There is another line of research: Designing incompressible block ciphers. Bogdanov and Isobe [21] have introduced the concept of SPACE-hard white-box block ciphers and presented a concrete construction SPACE based on a dedicated design rather than on an obfuscated implementation of an existing cipher (such a direction of adopting dedicated designs for block ciphers was initiated with ASASA [14]). The notion of SPACE hardness is a variant of incompressibility and provides immunity against code lifting. Similar notions include weak white-box security [14] and ENC-TCOM [35]. Bogdanov and Isobe have shown that SPACE achieves SPACE hardness, assuming AES is secure. SPACE is reasonably efficient, running faster than a hundred cycles per byte on modern PCs. A number of follow-up SPACE-hard white-box block ciphers have been proposed, including SPNbox [22] and WhiteBlock [35].

Now our motivation behind this work becomes evident: There is a large gap between the two lines of research. Specifically, we would like to address the following issues:

1. There exist no modes of operation that turn incompressible block ciphers into incompressible authenticated encryption (AE) schemes. As described in Section 3 (and in the full version of this paper [40]), existing modes such as GCM [51], GCM-SIV [36], CCM [61], and OCB [46] would not yield incompressible AE even if combined with an incompressible block cipher. The state-of-the-art incompressible block ciphers mentioned above, though secure and reasonably efficient, are not utilized.
2. As mentioned above, there exist no AE schemes that simultaneously ensure both privacy and authenticity against white-box adversaries, unless one relies on special hardware. Moreover, the only type of authenticity that has been achieved in the context of incompressibility is universal unforgeability, which is much weaker than what has been done in the conventional setting. Similar discussions are provided in the previous work by Bock et al. [19] where the authors point out that "the definition of incompressibility does not capture any further security such as confidentiality and authenticity".
3. The lack of secure AE modes or schemes indicates the need for further investigation into the incompressibility notion. Specifically, we would like to come up with a usable definition of incompressible block ciphers as well as a new notion of incompressibility that captures more perfectly the privacy and authenticity requirements on AE schemes. Having done that, we should be able to design a mode that enjoys both privacy and authenticity in a strong sense, by relying on the underlying incompressible cipher.

1.1 Our Contributions

We introduce new incompressibility-based white-box security notions for AEAD schemes and BCs which we name whPRI and whPRP, respectively. Intuitively, with the two notions we attempt to define the best possible security such that any λ-bit leakage from a lifter (e.g., malware) does not allow adversaries to break privacy and/or authenticity, or equivalently indistinguishability, except for λ-bit ciphertexts. In particular, the notions demand authenticity in quite a strong sense: existential unforgeability beyond leakage. Our definition, we believe, should be the first one to formalize this notion concretely. Obviously, this is a much stronger requirement than universal unforgeability. We remark that whPRI and whPRP are extensions of pseudo-random injections (PRI) [58] and pseudo-random permutations (PRP) in the black-box setting, respectively: they exactly match in the extreme case of $\lambda = 0$. The security games for our new definitions involve both of black-box adversaries and lifters. These games become inherently multi-stage.

We properly bound the computational resource t_{lif} of a lifter and the leakage size λ. Especially, no security is guaranteed after either t_{lif} or λ reaches a certain threshold, e.g., $t_{\mathrm{lif}} = 2^{50}$ or $\lambda = 2^{20}$. We expect that an attack (malware activity) should be detectable, before the threshold is reached, by some means, e.g., monitoring active processes and/or outgoing packets. We conjecture that SPACE should satisfy whPRP-security under some reasonable parameter settings.

For completeness we study theoretical possibilities of security reductions of various symmetric-key schemes; we introduce similar notions for keyed functions and conventional (privacy-only) encryption schemes. Our notion for keyed functions, which we call whPRF, is an extension of the standard pseudo-random function (PRF). For conventional encryption schemes, we define two security notions which we name whIND\$-CPA and wh$\widetilde{\mathrm{SPRP}}$. The former is an extension of IND\$-CPA security (for random-IV schemes) in the black-box setting. The latter is obtained as a special case of whPRI where ciphertext lengths are always equal to message lengths. Thus, wh$\widetilde{\mathrm{SPRP}}$ is an extension of the tweakable strong PRP ($\widetilde{\mathrm{SPRP}}$) security for tweakable enciphering schemes [38] in the black-box setting. We observe that meaningful counterparts of MAC security and nonce-based security notions seem unachievable in our context. Table 1 gives comparisons between various security notions for (authenticated) encryption schemes.

We prove that a reduction between the new security notions is possible if the construction in hand satisfies a variant of public indifferentiability [32, 63], which we name *weak public indifferentiability.* Then we demonstrate that all the new notions can be reduced to whPRP, by presenting corresponding constructions that are weak public indifferentiable.

Finally, as an example of practical AEAD modes of block ciphers, we show that a composition of the sponge construction [12] and the counter mode (CTR) via SIV [58] is whPRI-secure if the underlying block cipehr E_K is whPRP-secure. Here, the underlying primitive E_K is used both by the sponge and by the CTR.

Table 1: Comparison of incompressibility or related notions for symmetric-key (authenticated) encryption schemes. We assume that AEADs always take a nonce (or IV) as a part of input. Especially, a nonce is included into inputs of deterministic AEADs.

Security notion	Target scheme	Leakage	Adversarial goal	
(λ, δ)-incompressibility [29, 18]	deterministic or randomized encryption (RSA group or OWP-based schemes [29, 18])	whole implementation	δ-functionality with code size $< \lambda$	
			(Privacy)	(Authenticity)
LIND [9]	randomized encryption (Big-Key Encryption [9, 8])	via function with output size $= \ell$	distinguishing	—
IND-COM [35]	randomized AE (WhiteKey [35])	via function with entropy left $\geqslant \mu$	distinguishing	—
ENC-COM [35]	randomized AE (WhiteKey+RO)		—	universal forgery
whPRI [Section 4.3]	deterministic AE (SIV+CTR (Section 7))	via lifter (malware) with output size $\leqslant \lambda$	distinguishing	existential forgery beyond leakage
whIND\$-CPA [Section 5.3]	randomized encryption (CTR (Section 6.3))	via lifter (malware) with output size $\leqslant \lambda$	distinguishing	—
wh$\widetilde{\text{SPRP}}$ [Section 5.3]	tweakable enciphering scheme (6-round Feistel (Section 6.3))	via lifter (malware) with output size $\leqslant \lambda$	distinguishing	—

Roughly speaking, if E_K is secure up to λ-bit leakage, the resulting AEAD is secure as long as the amount of processed data is $\ll 2^{n/4}$ and leakage is less than λ. To instantiate E_K, we propose to use a 256-bit-block variant of SPACE which we name SPACE256. We conjecture that SPACE256 is secure up to 2^{20} bits of leakage. The resulting AEAD scheme is implemented on an Intel platform for experiments, and we confirm that the performance is practical. The size of the program is in an order of KB or MB, which is reasonably small for mobile applications. Unlike previous schemes achieving incompressibility, our scheme does not need random nonces. This is an advantage in the white-box setting because random number generators may be compromised by adversaries.

Note that our notions do not supersede previous ones but rather coexist with other white-box security approaches such as binding [20, 19]. Which security approaches, definitions or solutions one should choose changes depending on use cases and what one wants to achieve. Specifically, when trusted hardware is available or when lifters have much more limited access to programs, other security notions would be more suitable.

1.2 Related Work

Other Security Notions in White-Box Cryptography. The initial goal set by Chow et al. was to protect software implementations of existing block ciphers from key extraction when an attacker is given an unlimited access to a white-box implementation. Many pieces of previous work have proposed such implementations, but none of them remains unbroken [13, 47, 53, 62]. Some of the state-of-the-art work focus on limited white-box adversaries such as DCA and a certain class of algebraic attacks [23, 5, 16, 17].

Several solutions outside incompressibility have been suggested to mitigate code lifting. Chow et al. suggested external encoding [25], which yields a white-box implementation of $E'_K = G \circ E_K \circ F^{-1}$ for some functions F and G instead of E_K. The problem is that even an ordinary user needs a separate implementation of G^{-1} or F to compute E_K. Thus, white-box adversaries would also be able to peel off the external encoding, unless the encoding is stored in trusted hardware. Delerablée et al. suggested one-wayness [29], which formalizes the notion that one is unable to perform decryption even if an encryption program is given. They also suggested traceability [29], which allows a program distributor to trace malicious users who leak their encryption programs. Both are interesting, but they do not encompass resistance to copying encryption programs. Other works have discussed the possibility of binding [20, 19, 1], where the execution of encryption is bound by trusted hardware or applications. Unfortunately, cryptographically secure binding requires, together with secure hardware, primitives such as indistinguishability obfuscation (iO) or LWE, which are richer than usual symmetric-key primitives.

Symmetrically and Asymmetrically Hard Cryptography. Biryukov and Perrin [15] introduced the HSp mode (and its instantiation WHALE), which can be used to build an incompressible VIL/VOL hash function from a usual sponge hash (like SHA-3) and an FIL/FOL incompressible function. The mode is proven to achieve a universal-unforgeability-like security notion on incompressibility. Their result seems close to ours (in Section 6.3) that the sponge construction becomes a VIL/VOL whPRF if the underlying primitive is a whPRP (or FIL/FOL whPRF). Still, there are two differences between theirs and ours. First, they proved only universal-unforgeability-like security while we proved existential-unforgeability-like security (i.e., whPRF-security). Second, their proof is in the random oracle model while ours is in the standard model in that the existence of a whPRP (or a FIL/FOL whPRF) is a falsifiable assumption.

Leakage Resilient Cryptography. An important area related to white-box cryptography is *leakage resilient cryptography*, which aims to achieve provable security against side-channel attacks. Security models in leakage resilient cryptography are roughly classified into two types[7], depending on whether (1) an adversary is allowed to obtain arbitrary leakage from the secret key as long as the leakage length is bounded by a certain parameter, or (2) some form of security is assumed on memory or storage, and/or leakage is obtained only when some computation (e.g., encryption) is performed through a special class of functions such as the Hamming weight of internal states with some noise.

Models of the First Type. A typical model of the first type closely related to our results is the Bounded Retrieval Model (BRM) [27, 33], where large (e.g., 1GB) keys are used to prevent key exfiltration. The BRM and related notions

[7] This classification is based on (still not completely the same as) the one in [43, 44].

have been studied in a long line of research [27, 33, 24, 4, 3, 9, 8]. Among others, Bellare et al. [9] showed practical symmetric-key encryption schemes achieving confidentiality in the BRM, which was later improved by Bellare and Dai [8].

As pointed out by Fouque et al [35], the goals of Bellare et al. [8] and incompressibility are quite close. Still, each of the BRM and incompressibility has its own advantages. An advantage of the BRM is that, for well designed schemes such as the one by Bellare et al. [8], bounding the running time of a lifter (malware) is not mandatory (it is mandatory for incompressible ciphers because the secret key sizes are very small). Meanwhile, no previous works on symmetric encryption scheme in the BRM achieve both confidentiality and authenticity simultaneously[8], while we prove that SIV+CTR achieves whPRI.

Models of the Second Type. Major models of the second type include the "only computation leaks information" (OCL) model [52] and wire-probing leakage [42]. In models of the second type, lots of previous works have shown various leakage resilient schemes including AEADs [52, 34, 55, 48, 59, 11, 30, 7, 28, 31, 10, 37, 45]. Especially, Krämer and Struck [45] showed that the security of a leakage-resilient AEAD can be reduced to the security of leakage-resilient PRFs in the "only comoputation leaks information" model [52]. However, these results are incomparable to ours because they essentially assume that attackers do not have a direct full access to memory or storage that stores the secret key.

A clear advantage of the second type is that the size of implementations can be small, compared to incompressibility and the first type. When we can assume that adversaries do not have a full direct access to memory or storage (e.g., leakage can be obtained only by measuring power consumption of a circuit), models of the second type will be more suitable than incompressibility and the first type. When we cannot, incompressibility or the first type will be suitable.

1.3 Paper Organization

Section 2 introduces basic notations and definitions, and review basics on (public) indifferentiability. Section 3 shows an observation that GCM is unlikely to achieve incompressibility. In Section 4, we introduce whPRI, a new security notions for AEADs. New security notions for other schemes are introduced in Section 5. Section 6 introduces weak public indifferentiability and shows that weak public indifferentiability implies white-box security reductions. The section also demonstrates that our new notions on various schemes can be reduced to whPRP, by showing (weak) public indifferentiable constructions. In Section 7 we show that a practical whPRI-secure AEAD mode of whPRP can be realized as a composition by SIV of the sponge construction and the counter mode.

[8] A scheme by Bellare et al. [9] also achieves authenticity, but only in the absence of leakage (See also Table 1).

2 Preliminaries

Throughout the paper, $\mathrm{len}(M)$ denotes the bit length for a bit string M. Given a positive integer $m < \mathrm{len}(X)$, we write $(A, B) \xleftarrow{m,*} X$ to mean assignment of bit strings, the leftmost m bits of X to A and the remaining bits to B. The variable A or B may be omitted with the symbol "$\cdot$" in which case the corresponding bits are not assigned to any variable. When we write like $(X_1, X_2, \ldots, X_\ell) \xleftarrow{n} X$ we mean partitioning X into n-bit blocks and assigning them to $X_1, X_2, \ldots, X_\ell$ where $\ell = \lceil \mathrm{len}(X)/n \rceil$ and the last X_ℓ is possibly fractional, i.e., $\mathrm{len}(X_\ell) < n$. The symbol $\|$ stands for concatenation of bit strings and the symbol $\oplus$ exclusive OR of two bit stings of the same length. By block length of M we denote $\lceil \mathrm{len}(M)/n \rceil$ when the parameter n is clear from the context. For an invertible function F by $F^{\pm}$ we denote the oracles of F and F^{-1}. We denote the empty bit string by ε and define $\{0,1\}^0 := \{\varepsilon\}$. $\{0,1\}^*$ denotes the set of all bit strings of arbitrary length. For positive integers x and n, by $x \bmod n$ we denote the minimum positive integer i such that $i \equiv x \bmod n$. We say an m-input function $f : (\mathbb{Z}_{\geq 0})^{\times m} \to \mathbb{R}_{\geq 0}$ is non-decreasing if $f(x_1, \ldots, x_i + z, \ldots, x_m) \geq f(x_1, \ldots, x_m)$ holds for arbitrary $1 \leq i \leq m$, $(x_1, \ldots, x_m) \in (\mathbb{Z}_{\geq 0})^{\times m}$, and $z \in \mathbb{Z}_{\geq 0}$.

Definition 1 (Variable-key and fixed-key random injection). *Let $\tau \geq 0$ be an integer and $\mathrm{Inj}_\tau(\mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{M}, \mathbf{C})$ denote the set of functions $F : \mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{M} \to \mathbf{C}$ such that $F_{K,N,A} := F(K, N, A, \cdot)$ is an injection for each (K, N, A) and $\mathrm{len}(F(K, N, A, M)) = \mathrm{len}(M) + \tau$. A* variable-key random injection *F is an injection chosen uniformly at random from $\mathrm{Inj}_\tau(\mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{M}, \mathbf{C})$. The inverse $F^{-1} : \mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{C} \to \mathbf{M} \cup \{\perp\}$ is defined so that $F^{-1}(K, N, A, F(N, A, M)) = M$ for each (K, N, A, M) and $F^{-1}(K, N, A, C) = \perp$ for all $C \notin F_{K,N,A}(\mathbf{M})$. If $\mathbf{K}$ is a set that contains exactly a single element, we say F is a* fixed-key *random injection and omit to write $\mathbf{K}$ and K.*

Syntax of Symmetric-Key Cryptosystems and Basic Constructions.

Keyed Functions. A keyed function is a function $f : \{0,1\}^\kappa \times \mathbf{X} \to \mathbf{Y}$. Here, κ is a positive integer and $\{0,1\}^\kappa$ is called the key space. We write $f_K(M)$ and $f(K, M)$ interchangeably.

Block Ciphers. A block cipher is a keyed function $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ such that $E(K, \cdot)$ is a permutation for each K. The inverse function $(E_K)^{-1}$ is denoted by D_K, and we write $D_K(C)$ and $D(K, C)$ interchangeably. E and D are called the encryption and decryption functions.

AEADs. An AEAD scheme is a tuple $\Pi = (\mathcal{E}, \mathcal{D})$. The first element of Π is an encryption function $\mathcal{E} : \mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{M} \to \mathbf{C}$. Here, $\mathbf{K}$ is the key space from which the secret key is chosen uniformly at random. The set $\mathbf{N} = \{0,1\}^\nu$ is a nonce space with the nonce length ν being a non-negative integer. The sets $\mathbf{A}, \mathbf{M}, \mathbf{C}$ correspond to the spaces of associated data, plaintext and ciphertext, respectively, where $\mathbf{M} = \mathbf{C} = \{0,1\}^*$. We write interchangeably

Algorithm 1: CTR(K, IV, M)

```
$(M_1, M_2, \dots, M_\ell) \xleftarrow{n} M$
for $i = 1$ to $\ell$ do
    $y \leftarrow f_K(IV + i - 1)$,
    $(y', \cdot) \xleftarrow{\mathrm{len}(M_i), *} y$,
    $C_i \leftarrow M_i \oplus y'$
return $C_1 \parallel C_2 \parallel \cdots \parallel C_\ell$
```

Algorithm 2: $\mathcal{E}_{K_1,K_2}(N, A, M)$

```
$IV \leftarrow f_{K_1}(N, A, M)$
$C' \leftarrow \mathcal{E}'_{K_2}(IV, M)$
return $C \leftarrow IV \parallel C'$
```

Algorithm 3: $\mathcal{D}_{K_1,K_2}(N, A, C)$

```
$(IV, C') \xleftarrow{\tau, *} C$
$M \leftarrow \mathcal{D}_{K_2}(IV, C')$
$T \leftarrow f_{K_1}(N, A, M)$
if $IV = T$ then
    return $M$
else
    return $\bot$
```

$\mathcal{E}(K, N, A, M) = \mathcal{E}_K(N, A, M) = \mathcal{E}_{K,N,A}(M)$. For each $(K, N, A) \in \mathbf{K} \times \mathbf{N} \times \mathbf{A}$ we demand that $\mathrm{len}\big(\mathcal{E}_{K,N,A}(M)\big) = \mathrm{len}(M) + \tau$ should hold for all $M \in \mathbf{M}$, where τ a fixed non-negative integer. The second element of Π is a decryption function $\mathcal{D} : \mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{C} \to \mathbf{M} \cup \{\bot\}$. Here, the symbol $\bot$ signifies rejection. We write interchangeably $\mathcal{D}(K, N, A, M) = \mathcal{D}_K(N, A, M) = \mathcal{D}_{K,N,A}(M)$. For each (K, N, A, M) we demand that $\mathcal{D}_{K,N,A}(\mathcal{E}_{K,N,A}(M)) = M$ should hold.

Conventional Encryption Schemes. The syntax for a conventional (privacy-only) encryption scheme is essentially the same as that of AEAD except that it does not take any associated data, i.e., $\mathbf{A} = \{\varepsilon\}$, and $\tau = 0$. In addition, nonce N and nonce space $\mathbf{N}$ are renamed as IV and $\mathbf{IV}$. We assume IV is chosen uniformly at random for every encryption query or arbitrarily chosen by adversary depending on security notions we focus on.

Counter Mode. Counter mode (CTR) is the construction to convert a keyed function into a conventional encryption scheme. Let $f : \{0,1\}^\kappa \times \{0,1\}^m \to \{0,1\}^n$ be a keyed function. The encryption function of CTR based on f, which we denote by CTR(K, IV, M), is computed as in Algorithm 1. The key, IV, and message spaces are $\{0,1\}^\kappa$, $\{0,1\}^m$, and $\{0,1\}^*$, respectively. The decryption function is identical to the encryption function.

SIV. SIV is the construction introduced by Rogaway and Shrimpton to realize a deterministic AEAD [58]. Let $\mathbf{N}$ and $\mathbf{A}$ be arbitrarily chosen space of nonces and associated data. (We assume $\mathbf{N} = \{0,1\}^\nu$ for some $\nu \in \mathbb{Z}_{>0}$ and $\mathbf{A} = \{0,1\}^*$ unless otherwise noted.) Let $f : \{0,1\}^{\kappa_1} \times (\mathbf{N} \times \mathbf{A} \times \{0,1\}^*) \to \{0,1\}^\tau$ be a keyed function and $\Pi' = (\mathcal{E}', \mathcal{D}')$ be a conventional encryption scheme with the key space $\{0,1\}^{\kappa_2}$, IV space $\{0,1\}^\tau$, and message space $\{0,1\}^*$. The SIV construction based on f and Π is an AEAD with key space $\{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2}$, nonce space $\mathbf{N}$, associated data space $\mathbf{A}$, and message space $\{0,1\}^*$. The encryption function $\mathcal{E}$ and decryption function $\mathcal{D}$ are defined as in Algorithm 2 and Algorithm 3, respectively. We call an output of f a *tag* and f a tag-generation part.

Programs and White-Box Compilers. We follow the abstraction and notation used by Delerablée et al. [29] for dealing with programs and compilers. A program implements an algorithm, specific to some explicit language and execution model. A program can be read, copied and modified at will. A program can be viewed as a bit string, and its binary code can be executed locally. A program is inherently stateless. A program may, via APIs including system calls, make use of external resources such as random coins and additional functionalities. A white-box compiler $\mathcal{C}_E$ of a block cipher E is an algorithm that takes $K \in \{0,1\}^\kappa$ as an input and outputs a program that implements E_K. We use the notation $[\![E_k]\!]$ to denote a white-box implementation of E_K in a context where explicitly indicating the compiler is unnecessary. Moreover, we call $[\![E_K]\!]$ *white-box block cipher* simply. A white-box compiler may be probabilistic, outputting different programs for the same key[9]. White-box compilers of other cryptosystems are defined in the same way.

Indifferentiability. Let T^P be an algorithm (a cryptographic scheme, e.g., a VIL hash function) making queries to P, where P is an ideally random primitive (e.g., a FIL random oracle). In addition, let R be an ideally random scheme corresponding to T^P with the same input-output interface (e.g., a VIL random oracle). Then, the indifferentiability advantage of $\mathcal{A}$ against $(\mathsf{T}^\mathsf{P}, \mathsf{R})$ with respect to a simulator $\mathcal{S}$ is defined as $\mathsf{Adv}^{\text{indiff}}_{\mathsf{T},\mathsf{R},\mathcal{S}}(\mathcal{A}) := \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{T}^\mathsf{P},\mathsf{P}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{R},\mathcal{S}^\mathsf{R}}\right]$. Informally, we say T^P is indifferentiable from $\mathcal{R}$ if there is an efficient simulator $\mathcal{S}$ such that the above advantage becomes negligibly small for any efficient $\mathcal{A}$. We call T^P (resp., $\mathcal{R}$) a construction oracle and $\mathcal{P}$ a primitive oracle. We call queries to T^P or R (resp., P or $\mathcal{S}^\mathsf{R}$) construction queries (resp., primitive queries).

The most important feature of indifferentiability is the general "composition theorem" [50, 56]: Suppose the following (1)-(3) hold: (1) A scheme (or protocol) Π^R depending on the ideal object R is proven secure. (2) T^P is indifferentiable from R. (3) The security of Π is defined by single-stage games. Then the composition theorem guarantees that $\Sigma^{\mathsf{T}^\mathsf{P}}$ is secure [50]. Note that not only (1) and (2) but also (3) is crucial; the composition theorem does not necessarily hold for schemes of which security is defined by multi-stage games [56]. We do not get into further details because it is not directly related to our results.

Indifferentiability of Sponge. Let $r, c > 0$ and $f : \{0,1\}^{r+c} \to \{0,1\}^{r+c}$ be a function. Let $\mathsf{pad} : \{0,1\}^* \to (\{0,1\}^r)^+$ be an injective padding function such that the last (r-bit) block of $\mathsf{pad}(X)$ is not 0^r for every X.[10] The sponge construction Sponge^f maps bit strings of arbitrary length to bit strings of any requested length as in Algorithm 4 (i.e., Sponge^f can be regarded as a function from $\{0,1\}^* \times \mathbb{N}$ to $\{0,1\}^\infty$). The parameters r and c are called rate and capacity.

[9] In practice, many white-box implementations of AES are the output of the probabilistic compiler. On the other hand, the dedicated white-box block cipher such as SPACE uses the deterministic compiler in general.

[10] In what follows, we assume the padding function pads "1" and the minimum number of zeroes so that the total length of the padded string becomes multiple of r, i.e., $\mathsf{pad}(X) := X||1||0^{\text{len}(X) \bmod r-1}$.

Algorithm 4: $\mathsf{Sponge}^f(X)$ with requested output length m

```
$(X_1, \ldots, X_\ell) \xleftarrow{r} \mathsf{pad}(X)$, $s \leftarrow 0^{r+c}$, $y \leftarrow \varepsilon$
for $i = 0$ to $\ell - 1$ do
    $s \leftarrow f(s \oplus (X_{i+1} || 0^c))$
for $i = \ell$ to $\ell + \lceil \frac{m}{r} \rceil - 1$ do
    $y \leftarrow y ||$(the upper $r$ bits of $s$), $s \leftarrow f(s)$
return $y$
```

Bertoni et al. [12] proved that the sponge construction is indifferentiable from a VIL/VOL random oracle $\mathsf{RO} : \{0,1\}^* \to \{0,1\}^\infty$ if f is an ideally random function. More precisely, they showed the following theorem[11].

Theorem 1 (Theorem 1 of [12]). *Let* $\epsilon(q) := 1 - \prod_{i=1}^{q}\left(1 - \frac{1}{2^c}\right)$. *There exists a simulator* $\mathcal{S}$ *making queries of total length at most* $\lceil \frac{r+c}{r} \rceil q^2$ *such that* $\mathbf{Adv}^{\mathrm{indiff}}_{\mathsf{Sponge},\mathsf{RO},\mathcal{S}}(\mathcal{A}) \leq \epsilon(q)$ *holds for any adversary* $\mathcal{A}$ *that calls* f *at most* $q (< 2^c)$ *times in the real world, either directly or indirectly through* Sponge^f.

Indifferentiable AEAD Schemes. Barbosa and Farshim studied indifferentiable AEAD schemes [6], where the ideal oracle is a *variable-key* random injection F and its inverse F^{-1} (see Definition 1)[12]. Note that a variable random injection takes not only nonce, associated data, and message (or ciphertext) but also a key as an input. They especially showed that indifferentiable AEADs cannot be achieved by some generic compositions such as SIV, and that indifferentiable constructions can be built by Encode-then-Encipher (EtE) or 3-round Feistel-based scheme. In particular, by using the sponge construction for round functions of the Feistel-based scheme, an indifferentiable AEAD can be built from a FIL/FOL random function. See the full version of this paper [40] for details.

Public Indifferentiability. Again, let T^{P} be a construction calling an ideal primitive P, and R be an ideal object of which interface is compatible with T^{P}. Public indifferentiability [32, 63] is defined in the same way as the original indifferentiability, except that a simulator $\mathcal{S}$ is allowed to observe all the queries by

[11] The theorem roughly says Sponge^f is secure up to $2^{c/2}$ queries because $\epsilon(q) \approx 1 - e^{-\frac{q(q+1)}{2^{c+1}}} < \frac{q(q+1)}{2^{c+1}}$ holds for $q \ll 2^c$. The original theorem in [12] did not mention the exact number of queries by $\mathcal{S}$ but we can deduce it is at most $\lceil \frac{r+c}{r} \rceil q$ by checking the details of the proof.

[12] The parameter τ (the length of ciphertext-stretch) is also considered as an input to AEADs and random injections in [6], but this paper considers the special case where is τ fixed to a constant.

adversaries to R and the responses. (Public-indifferentiability is actually a special case of indifferentiability rather than a variant. However, we regard it as a variant for readability.) More precisely, in the ideal world, there is an additional oracle-query interface to reveal the list of all the queries made so far to R and the responses, and an access to this interface is given to $\mathcal{S}$ (but not to $\mathcal{A}$). We call this interface the *revealing interface*, and denote by $\mathsf{Rev}[\mathsf{R}]$. This models the condition that every input to R (and the output) is visible to all the parties involved in a security game, and the general "composition theorem" on public-indifferentiability holds only for schemes of which security games satisfy such a condition. The restriction that the "composition theorem" does not necessarily hold for multi-stage games also applies to public-indifferentiability, but the theorem holds for single-stage games as long as this condition holds. The public indifferentiability advantage is defined as $\mathsf{Adv}^{\text{pub-indiff}}_{\mathsf{T},\mathsf{R},\mathcal{S}}(\mathcal{A}) := \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{T}^{\mathsf{P}},\mathsf{P}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{R},\mathcal{S}^{\mathsf{R},\mathsf{Rev}[\mathsf{R}]}}\right]$. Informally, we say T^{P} is public indifferentiable from $\mathcal{R}$ if there exists an efficient simulator $\mathcal{S}$ such that the above advantage becomes negligibly small for any efficient $\mathcal{A}$. The Merkle-Damgård construction is proven public indifferentiable [32].

Remark 1. While it is straightforward to show the composition of two indifferentiable constructions becomes again indifferentiable[13], its seems quite hard (or even impossible) to prove that the composition of two public indifferentiabile constructions becomes again public indifferentiable. (See Section 6.1 for details).

3 Code Lifting on GCM

This section briefly explains that GCM [51] is unlikely to achieve incompressibility in the presence of a lifter given an unlimited access to an implementation, even when used with an incompressible block cipher. Recall that GCM is an AEAD mode of 128-bit block cipher composed of CTR and a universal hash function called GHASH (see Fig. 4 of the full version [40] for details). As an input, the encryption function of GCM takes a tuple of a nonce N, associated data A, and a message M. Given an input (N, A, M), CTR first encrypts M into a ciphertext C' with an IV derived from N. Then, a tag value T is computed as $T := \mathrm{GHASH}_{E_K(0^{128})}(A, C') \oplus E_K(N||1)$. The output of the encryption function is $T||C'$. GCM is proven secure in the nonce-respecting scenario where each nonce is never repeated for encryption queries[14]. When a nonce is repeated, GCM is broken even in the black-box setting.

An important feature of GCM is that the authenticity heavily relies on the value $E_K(0^{128})$: Suppose we know $E_K(0^{128})$ in addition to the tag T and the ciphertext C' for an input (N, A, M). Then, for arbitrary $\tilde{A}$ and $\tilde{M}$ with $\mathrm{len}(\tilde{M}) \leq \mathrm{len}(M)$, we can produce the tag $\tilde{T}$ and the ciphertext $\tilde{C}'$ corresponding to

[13] If $\mathcal{S}$ (resp., $\mathcal{S}'$) is a simulator for a construction T^{P} (resp., U^{Q}) making the indifferentiability advantage small (and if the interfaces are compatible), then ${\mathcal{S}'}^{\mathcal{S}}$ makes the advantage for $\mathsf{T}^{\mathsf{U}^{\mathsf{Q}}}$ small.

[14] Note that nonce reuse for *decryption* is allowed.

$(N, \tilde{A}, \tilde{M})$ *without knowing the secret key* K as $\tilde{C}' = \tilde{M} \oplus$(the upper len$(\tilde{M})$ bits of $M \oplus C$) and $\tilde{T} = \mathrm{GHASH}_{E_K(0^{128})}(A, C') \oplus T \oplus \mathrm{GHASH}_{E_K(0^{128})}(\tilde{A}, \tilde{C}')$. This means the universal forgery attack is possible and the authenticity of GCM is completely broken once an adversary retrieves $E_K(0^{128})$.

In the black-box setting, the value $E_K(0^{128})$ is hidden from adversaries and GCM achieves authenticity. However, in the white-box setting where a lifter has an unlimited access to a white-box implementation of GCM, the lifter could copy and leak the value $E_K(0^{128})$ to an attacker to break authenticity[15]. This attack works even if the underlying block cipher E_K is incompressible. Just copying a single 128-bit string $E_K(0^{128})$ would not be difficult no matter how hard copying the full functionality of E_K is.

The above attack shows that GCM fails to inherit incompressibility from E_K: A relatively small amount of data $E_K(0^{128})$ leaks information on an exponentially many input-output pairs of GCM. Similar attacks exist for other AE modes such as CCM, GCM-SIV, and OCB. See the full version [40] for details.

4 New AEAD Security Notion

This section gives us a formal definition of incompressibility-based white-box security of an AEAD implementation. Security notions for other cryptosystems are given later based on the definition for AEADs.

The attack in the previous section (and the ones in the full version of the paper [40]) shows that, with raw implementation of AEAD modes such as GCM, a small amount of leakage from the underlying white-box cipher could lead to giving the adversary a great deal of information concerning valid ciphertext values of the overlying AEAD scheme. Clearly this is an undesirable situation.

Basically, we want that a small amount of leakage would only lead to a small amount of valid ciphertext information, but there is a subtlety. A white-box attacker, or *lifter* (e.g., malware) could locally encrypt a large number of messages and then compute leakage of a small size from the obtained ciphertexts. As a result, the leakage, as a function, may depend on a large number of ciphertext values. Intuitively, we want that:

1. The leakage should not contain information yielding ciphertext values that have not been computed by the lifter, so that the ciphertexts that the adversary can compute from the leakage are limited to those that have been already computed by the lifter, and
2. The number of ciphertexts that the adversary can compute from the leakage should be small likewise the leakage size.

We establish a security notion that formalizes these requirements.

[15] The value $E_K(0^{128})$ could be protected from some white-box attacks with software or hardware countermeasures. Still, the effectiveness of such countermeasures would be limited, given that existing white-box implementations of AES ensure security only when adversaries have limited access to implemented algorithms. In addition, our aim is to achieve white-box security without assuming trusted hardware.

4.1 White-Box AEAD Attack Model

This section shows our attack model on white-box AEADs. We discuss on the security *after* the code lifting because no security can be guaranteed before and during the code lifting.

First, we provide an intuitive observation on what kind of attackers we have to take into account. Assume a white-box AEAD scheme is running on a target device, e.g., a remote server or a smartphone. Real-world attackers will behave as follows: First, an attacker performs advance preparation on the target scheme, making black-box queries to the encryption and decryption functions if possible. Then the attacker creates a lifter, e.g., a malware or an analysis tool, and give the lifter access to the white-box implementation by any means[16]. After analyzing the implementation, the lifter leaks some information on the scheme to the attacker. Finally, the attacker tries to break the privacy or authenticity of the scheme by using the leakage.

Based on the above observation, we reach the following attack model. Formally, let $\Pi = (\mathcal{E}, \mathcal{D})$ be an AEAD and $\mathcal{C}_\Pi$ its compiler. A *white-box adversary* $\mathcal{A} = (\mathcal{A}_{\text{create}}, \mathcal{A}_{\text{dist}})$ is a pair of oracle-aided, probabilistic random-access machines (RAMs.) The adversary $\mathcal{A}$ attacks $\mathcal{C}_\Pi$, running in two stages, as follows:

Initialization. A key K is chosen uniformly at random. Then using this key we put $\mathcal{P} \leftarrow \mathcal{C}_\Pi(K)$.

1st stage: creating a lifter. The first-stage is run by the sub-adversary $\mathcal{A}_{\text{create}}$ which has only black-box access to $\mathcal{P}$, making queries to oracles $\mathcal{E}_K$ and $\mathcal{D}_K$. The goal of $\mathcal{A}_{\text{create}}$ is to output a deterministic RAM $\mathcal{L}$ which we call a *lifter.*

Lifter execution. Once created, the lifter $\mathcal{L}$ gets full access to the AEAD program $\mathcal{P}$. The lifter $\mathcal{L}$ tries to extract some useful information out of the implementation $\mathcal{P}$, for example key material or compressed codes, and sends leakage data L to the adversary. The size of L is restricted to λ bits, which are properly smaller than the description of $\mathcal{P}$.

2nd stage: distinguishing. Upon receiving leakage L from the lifter, the second-stage sub-adversary $\mathcal{A}_{\text{dist}}$ resumes querying to $\mathcal{E}_k$ and $\mathcal{D}_k$, and finally outputs a bit string.

We could consider various sorts of adversarial goals, such as key recovery, plaintext recovery and ciphertext forgery. Of these, we choose the distinguishing attack, extending the "gold-standard" IND-CCA in the black-box setting: We assume $\mathcal{A}_{\text{dist}}$ finally outputs a bit b. The final goal of the adversary $\mathcal{A}$ is to distinguish between the real world ($b = 1$) and the ideal world ($b = 0$), i.e., whether the oracles $\mathcal{E}_K$ and $\mathcal{D}_K$ and the leakage have been real, or they have been some random and simulated ones.

Of course, white-box implementation is not present in the black-box security definitions, so we shall define how the leakage is computed in the ideal world.

[16] For instance, if the target device is a remote server, the lifter would be a malware that sneaks into the server. If the device is a smartphone, the lifter would be an analysis tool and the attacker may take the advantage of a slight opportunity to analyze the smartphone while the owner does not pay attention to it.

Consequently, we shall later introduce a simulator that imitates the behavior of the lifter $\mathcal{L}$.

Even if it is impossible to prevent lifters from getting access to the white-box implementation, we expect it is still possible to notice an attack is being mounted when non-negligible amount of data is sent to a strange and suspicious direction, by monitoring outgoing packets. Hence we define security notions when the leakage size λ is limited, e.g., up to 2^{20} bits. No security is guaranteed after λ reaches the limitation. In addition, basically we assume the running time of the lifter t_{lif} is much smaller than that of the adversary t (e.g., $t_{\mathrm{lif}} = 2^{50}$ while $t = 2^{112}$) and the intrusion of the lifter can be detected after t_{lif} time has passed.

We do not formalize the attack model in such a way $\mathcal{L}$ communicates with $\mathcal{A}$ since $\mathcal{L}$ can do everything $\mathcal{A}$ can do, and thus communications do not help much to break the scheme.

4.2 Ideal Oracles and Simulators

It remains to describe the ideal world in order to give a formal definition of white-box AEAD security.

When *black-box* security of nonce-based AEAD is studied, typically the ideal encryption (resp., decryption) oracle is set to be the one that always returns a random ciphertext (resp., the reject symbol). The adversary is prohibited to forward outputs from the encryption oracle to the decryption oracle to exclude trivial attacks.

On the other hand, in our white-box setting we cannot set the ideal oracles like above because the adversary can distinguish the ideal decryption oracle from the real one if the lifter leaks a valid ciphertext C and the adversary queries C to the decryption oracle.

Thus we set *a fixed-key random injection* F and its inverse F^{-1} as the ideal oracles (see Definition 1), following previous works on pseudorandom injection (PRI) security of AEADs [58, 39]. In particular, our security notion will completely match the black-box PRI security when $\lambda = 0$.

Note that the black-box PRI security matches the misuse-resistant AE (MRAE) security [58, 39] if the tag length τ is sufficiently long: Roughly speaking, the difference between the PRI advantage and the MRAE advantage of an AEAD scheme is upper-bounded by $O(q^2/2^\tau)$, where q is the number of black-box oracle queries [39, Thorem 1]. Thus our white-box security notion will require a secure scheme to be at least MRAE-secure in the black-box model.

Simulators. Now what remains of the real world is the program $\mathcal{P}$ and the lifter $\mathcal{L}$. $\mathcal{P}$ does not exist in the ideal world and it is non-trivial how we should define the behavior of $\mathcal{L}$. To remedy this, we introduce a simulator that imitates the behavior of $\mathcal{L}$.

Recall our intuition on the property that a secure white-box scheme must meet: Any leakage on a secure scheme does not contain information enabling an adversary to compute ciphertext values that have not been computed by a lifter. In other words, information that a lifter $\mathcal{L}$ can send to an adversary $\mathcal{A}_{\mathrm{dist}}$ (with reasonable computational resources) is only those computable or *simulatable* from some input-output pairs of $\mathcal{E}_K$ and $\mathcal{D}_K$.

We model this situation by existence of a simulator $\mathcal{S}$ working as follows. Given the description of a lifter $\mathcal{L}$[17] and oracle access to F and F^{-1} in the ideal world, $\mathcal{S}$ produces a bit string L_{ideal} which is, to $\mathcal{A}_{\text{dist}}$, indistinguishable from leakage L_{real} by $\mathcal{L}$ in the real world.

Since F is an ideally random object, $\mathcal{S}$ in the ideal world cannot leak more than λ bits of information on $F^{\pm}$ via λ-bit leakage L_{ideal}. Hence, intuitively, if $\mathcal{A}_{\text{dist}}$ cannot L_{real} and L_{ideal}, then $\mathcal{L}$ in the real world cannot leak more than λ bits of information on $\mathcal{E}_K$ and $\mathcal{D}_K$ via λ-bit leakage L_{real}.

More specifically, a simulator $\mathcal{S}$ is an oracle-aided RAM. We give $\mathcal{S}$ the ability to do its job as follows:

1. We give $\mathcal{S}$ as its input the lifter $\mathcal{L}$ just as it is. Then $\mathcal{S}$ can perform static and dynamic analyses on $\mathcal{L}$. The code of $\mathcal{L}$ can be read, dissected and studied, so that $\mathcal{S}$ can determine the functionality of $\mathcal{L}$.
2. Needless to say, we let $\mathcal{S}$ have oracle access to $F^{\pm 1}$.
3. We give $\mathcal{S}$ sufficient computational power and do not explicitly bound its running time. We only demand that the algorithm $\mathcal{S}$ be a finite sequence of well-defined instructions and operations. By doing so, we believe that our security notion should become achievable by a sound portion of AEAD programs while dismissing the rest.

In addition, we assume that $\mathcal{S}$ can observe all the queries to $F^{\pm}$ by $\mathcal{A}_{\text{create}}$ and the responses. The reasons that we assume this is as follows. First, if we define a security notion for conventional encryption schemes similarly *without this assumption*, then a conventional encryption scheme (random-IV CTR) which intuitively seems white-box-secure is deemed insecure (see the full version of this paper [40] for details). However, if the assumption is included in the definition, random-IV CTR can be proven secure (Section 6.3). Thus it seems reasonable to include the assumption into the definition for conventional encryption schemes. Second, We would like to make security definitions for various cryptosystems consistent as much as possible. Thus we include this assumption not only for conventional encryption schemes but also for AEADs.

4.3 Formal Security Notion: whPRI

Now we are ready to define new security notion of AEAD programs. We call our notion *white-box pseudo-random injection security (whPRI)*.

We consider a white-box adversary $\mathcal{A} = (\mathcal{A}_{\text{create}}, \mathcal{A}_{\text{dist}})$ running in two different experiments called games. The real white-box PRI game ($\boxed{\text{PRI}}$-real) is an experiment in the real world as described in Sect. 4.1. We assume that the white-box program $\mathcal{P}$ contains an implementation of not only encryption but also decryption. The ideal white-box PRI game ($\boxed{\text{PRI}}$-ideal) is an experiment

[17] Note that a lifter is also made by a first-stage adversary $\mathcal{A}_{\text{create}}$ in the ideal world, but the black-box oracles given to $\mathcal{A}_{\text{create}}$ are (F, F^{-1}) instead of $(\mathcal{E}_K, \mathcal{D}_K)$.

Experiment 5: $\mathbf{Exp}^{\boxed{\text{PRI}}\text{-real}}_{\Pi,\mathcal{C}_\Pi,\mathcal{A}}$

```
1: $K \xleftarrow{\$} \mathbf{K}$, $\mathcal{P} \leftarrow \mathcal{C}_\Pi(K)$
2: $(\mathcal{L}, S) \leftarrow {\mathcal{A}_{\text{create}}}^{\mathcal{E}_K,\mathcal{D}_K}()$
3: $L \leftarrow \mathcal{L}(\mathcal{P})$
4: $\beta \leftarrow {\mathcal{A}_{\text{dist}}}^{\mathcal{E}_K,\mathcal{D}_K}(S, L)$
5: return $\beta$
```

Experiment 6: $\mathbf{Exp}^{\boxed{\text{PRI}}\text{-ideal}}_{\mathcal{S},\mathcal{A}}$

```
1: $F \xleftarrow{\$} \mathrm{Inj}_\tau(\mathbf{N} \times \mathbf{A} \times \mathbf{M}, \mathbf{C})$
2: $(\mathcal{L}, S) \leftarrow {\mathcal{A}_{\text{create}}}^{F,F^{-1}}()$
3: $L \leftarrow \mathcal{S}^{F,F^{-1}}(\mathcal{L}, \mathsf{List}_{\text{create}})$
4: $\beta \leftarrow {\mathcal{A}_{\text{dist}}}^{F,F^{-1}}(S, L)$
5: return $\beta$
```

Fig. 1: Experiments for whPRI. In the ideal experiment, $\mathsf{List}_{\text{create}}$ denotes the list of queries by $\mathcal{A}_{\text{create}}$ to $F^{\pm}$ and the responses.

in the ideal world, where the oracles and the lifter are replaced with a random injection and a simulator, respectively. These two games are formally defined in Exp. 5 and Exp. 6.

Now, given an AEAD scheme Π and its compiler $\mathcal{C}_\Pi$, let us define the *wh-PRI advantage* of a white-box adversary $\mathcal{A} = (\mathcal{A}_{\text{create}}, \mathcal{A}_{\text{dist}})$ with respect to a simulator $\mathcal{S}$ as $\mathbf{Adv}^{\text{whPRI}}_{\Pi,\mathcal{C}_\Pi,\mathcal{S}}(\mathcal{A}) := \Pr\left[\mathbf{Exp}^{\boxed{\text{PRI}}\text{-real}}_{\Pi,\mathcal{C}_\Pi,\mathcal{A}} = 1\right] - \Pr\left[\mathbf{Exp}^{\boxed{\text{PRI}}\text{-ideal}}_{\mathcal{S},\mathcal{A}} = 1\right]$.

Definition 2 (whPRI). *The pair of an AEAD scheme Π and a compiler $\mathcal{C}_\Pi$ is $(\lambda, t, q, \sigma, t_{\text{lif}}, q_{\text{sim}}, \sigma_{\text{sim}}, \epsilon)$-whPRI-secure white-box AEAD if the following condition is satisfied: Let $\mathcal{A}$ be an arbitrary adversary running in time t and making queries at most q times. The lengths of the queries are at most σ in total*[18]*. In addition, $\mathcal{A}$ creates a lifter that runs in time t_{lif} and outputs at most λ-bit leakage. For arbitrary such $\mathcal{A}$, there exists a simulator $\mathcal{S}$ that makes at most q_{sim} queries of which lengths are at most σ_{sim} in total, and satisfies an inequality $\mathbf{Adv}^{\text{whPRI}}_{\Pi,\mathcal{C}_\Pi,\mathcal{S}}(\mathcal{A}) < \epsilon$.*

Informally, suppose the following claim holds: For any "efficient" $\mathcal{A}$, there exists a simulator $\mathcal{S}$ that makes a "reasonable amount of" queries and making the whPRI-advantage small[19]. Then we say that Π is whPRI-secure.

The attacks on GCM, GCM-SIV, CCM, OCB in Section 3 (or in the full version of this paper [40]) show that, for each of those schemes, there exists a lifter $\mathcal{L}$ that leaks the information on exponentially many number of input-output pairs by only a small amount of leakage. In the ideal world, the information of

[18] The unit of length can be set arbitrarily (e.g., bit or block) depending on the context.

[19] We set quantifiers as $\forall\mathcal{A}\exists\mathcal{S}$ rather than $\exists\mathcal{S}\forall\mathcal{A}$ so that the possibility of existence of primitives will increase, and the order of the quantifiers seems to have little impact on whether a practical scheme is judged secure or not. Indeed, our proofs in later sections, in addition to the discussions about the attacks on GCM, GCM-SIV, CCM, OCB mentioned below, work regardless of the order of the quantifiers.

Experiment 7: $\mathbf{Exp}^{\boxed{\mathrm{PRP}}\text{-real}}_{E,\mathcal{C}_E,\mathcal{A}}$

```
$K \xleftarrow{\$} \{0,1\}^{\kappa}$, $\mathcal{P} \leftarrow \mathcal{C}_E(K)$
$(\mathcal{L}, S) \leftarrow {\mathcal{A}_{\mathrm{create}}}^{E_K}()$
$L \leftarrow \mathcal{L}(\mathcal{P})$
$\beta \leftarrow {\mathcal{A}_{\mathrm{dist}}}^{E_K}(S, L)$
return $\beta$
```

Experiment 8: $\mathbf{Exp}^{\boxed{\mathrm{PRP}}\text{-ideal}}_{\mathcal{S},\mathcal{A}}$

```
$P \xleftarrow{\$} \mathrm{Perm}(n)$
$(\mathcal{L}, S) \leftarrow {\mathcal{A}_{\mathrm{create}}}^{P}()$
$L \leftarrow \mathcal{S}^{P,P^{-1}}(\mathcal{L}, \mathsf{List}_{\mathrm{create}})$
$\beta \leftarrow {\mathcal{A}_{\mathrm{dist}}}^{P}(S, L)$
return $\beta$
```

Fig. 2: Experiments for whPRP. $\mathsf{List}_{\mathrm{create}}$ in Experiment 2 denotes the list of queries to P by $\mathcal{A}_{\mathrm{create}}$ and the responses.

input-output pairs of the black-box oracle $F^{\pm}$ that a simulator can output by a λ-bit leakage is at most λ-bit. Hence no simulator will be able to mimic the behavior of such $\mathcal{L}$. Therefore those modes are unlikely to achieve whPRI-security.

5 New White-Box Security Notions for Other Schemes

This section introduces white-box security notions for block ciphers, keyed functions, and conventional encryption schemes.

5.1 whPRP: Secure White-Box Block Ciphers

We call the new security notion for white-box block ciphers *white-box pseudorandom permutation security* (*whPRP*). The definition of whPRP is similar to that of whPRI; the oracles $\mathcal{E}_K$ and $\mathcal{D}_K$ are now just E_K, and its counterpart in the ideal game is a random permutation $P \in \mathrm{Perm}(n)$, where $\mathrm{Perm}(n)$ denotes the set of permutations on $\{0,1\}^n$.

We again consider a white-box adversary $\mathcal{A} = (\mathcal{A}_{\mathrm{create}}, \mathcal{A}_{\mathrm{dist}})$ running in two games: the real white-box PRP game ($\boxed{\mathrm{PRP}}$-real) which is formally defined in Exp. 7 and the ideal white-box PRP game ($\boxed{\mathrm{PRP}}$-ideal) in Exp. 8. We assume that the white-box program given to a lifter contains an implementation of not only encryption but also decryption. Then, given a block cipher E and its compiler $\mathcal{C}_E$, let us define the *whPRP advantage* of a white-box adversary $\mathcal{A} = (\mathcal{A}_{\mathrm{create}}, \mathcal{A}_{\mathrm{dist}})$ with respect to a simulator $\mathcal{S}$ as $\mathbf{Adv}^{\mathrm{whPRP}}_{E,\mathcal{C}_E,\mathcal{S}}(\mathcal{A}) := \Pr\left[\mathbf{Exp}^{\boxed{\mathrm{PRP}}\text{-real}}_{E,\mathcal{C}_E,\mathcal{A}} = 1\right] - \Pr\left[\mathbf{Exp}^{\boxed{\mathrm{PRP}}\text{-ideal}}_{\mathcal{S},\mathcal{A}} = 1\right]$.

Definition 3 (whPRP). *The pair of a block cipher E and a compiler $\mathcal{C}_E$ is a $(\lambda, t, q, t_{\mathrm{lif}}, q_{\mathrm{sim}}, \epsilon)$-secure whPRP if the following condition is satisfied: Let $\mathcal{A}$ be an arbitrary adversary running in time t and making at most q queries. $\mathcal{A}$ makes a lifter that runs in time t_{lif} and outputs at most λ-bit leakage. For arbitrary such*

$\mathcal{A}$, there exists a simulator $\mathcal{S}$ that runs in time t_{sim}, makes at most q_{sim} queries, and satisfies an inequality $\mathbf{Adv}^{\mathrm{whPRP}}_{E,\mathcal{C}_E,\mathcal{S}}(\mathcal{A}) < \epsilon$.

Informally, suppose the following claim holds: For any "efficient" $\mathcal{A}$, there exists a simulator $\mathcal{S}$ that makes a "reasonable amount of" queries and making the whPRP-advantage small. Then we say that Π is whPRP-secure.

The definition of whPRP is a strengthening of the conventional black-box PRP, as the latter corresponds to the case $\lambda = 0$. It should be noted that we allow the simulator $\mathcal{S}$ to make queries to P^{-1}.

We can also consider the strong PRP version, whSPRP, where $\mathcal{A}$ is given oracle access to not only E_K but also E_K^{-1}. It is strictly stronger than whPRP.

As a candidate of whPRP, we conjecture[20] that SPACE-n_a ($n_a \in \{8, 16, 24, 32\}$) is a $(\lambda, t, q, t_{\mathrm{lif}}, q_{\mathrm{sim}}, \epsilon)$-secure whPRP with $t \approx 2^\kappa$, $q \approx 2^n$, $\lambda \approx (n - n_a) \cdot 2^{n_a - 2}$, and $\epsilon \ll 1$, as long as $t_{\mathrm{lif}} \ll q_{\mathrm{sim}} (< 2^n)$. Here, n and κ denote the block and key length, which are 128. See the full version [40] for more details.

5.2 whPRF: Secure White-Box Keyed Functions

We call the new security notion for white-box keyed functions *white-box pseudo-random function security* (*whPRF*)[21], which is defined in the same way as wh-PRP except that the black-box oracle given to the adversary is a random function RF instead of a random permutation P, and that simulators have access to RF instead of P and P^{-1}. Real and ideal experiments in addition to a distinguishing advantage are defined in the same way as those for whPRP.

5.3 White-Box Security of Conventional Encryption Schemes

We define two security notions on conventional IV-based encryption schemes, which we name *tweakable strong PRP security* (wh$\widetilde{\mathrm{SPRP}}$) and *white-box IND\$-CPA security* (whIND\$-CPA).

wh$\widetilde{\mathbf{SPRP}}$. The most natural way to obtain a definition of conventional IV-based encryption schemes is to consider the special case of whPRI where $\mathbf{A} = \{\varepsilon\}$ and $\tau = 0$. This is an extension of (VIL) tweakable strong PRP ($\widetilde{\mathrm{SPRP}}$) security for enciphering schemes in the black-box setting [38], and thus we call it wh$\widetilde{\mathrm{SPRP}}$.

[20] Note that it is unrealistic to "prove" whPRP-security of SPACE-256-16 in the same sense as proving PRP security of AES is unrealistic. Generally, the only realistic way to be confident with security of a block cipher is to see whether it withstands various attempts of cryptanalysis by experts. Recently, the security of some space-hard block ciphers was reviewed against a similar adversary to whPRP in [60].

[21] We define a white-box version of PRF security but does not for MAC security such as existential unforgeability. This is because a lifter can leak a valid message-tag pair that has not been queried to oracles before, and thus it seems hard to achieve a sound white-box version of existential unforgeability. It might be possible to define a white-box version of weaker notions such as universal unforgeability, but such notions are out of the scope of this paper. Studying weaker notions is a future work.

whIND\$-CPA. Though wh$\widetilde{\text{SPRP}}$ is naturally derived from whPRI, many popular conventional encryption schemes such as CTR and CBC are not $\widetilde{\text{SPRP}}$-secure even in the black-box setting. Thus we seek for another definition extending ones that CTR and CBC meet in the black-box setting. Since CTR and CBC cannot achieve indistinguishability against CCAs, we focus on security against CPAs.

In the black-box setting, we have three scenarios depending on how IVs for encryption queries are chosen.

1. Arbitrary IV (or, nonce-misuse) scenario: IVs are chosen by adversaries completely arbitrarily.
2. Nonce IV (or, nonce-respecting) scenario: IVs are chosen by adversaries arbitrarily, but repeated uses are prohibited (i.e., once an IV value is used for a query, it is never be used again).
3. Random IV scenario: An IV is chosen uniformly at random for every encryption query.

CTR and CBC cannot achieve indistinguishability in the first scenario. The second scenario is popular in the black-box setting but not suitable in our context since a lifter may leak information on a valid message-ciphertext pair w.r.t. an unused nonce. Thus we focus on the random IV scenario.

We follow [54] for the black-box security notion against CPAs for conventional random-IV encryption scheme. The notion is defined by real and ideal experiments. In the real experiment, an adversary has an access to a modified version of the encryption oracle $\mathcal{E}_K$, which we denote by $\mathcal{E}_{K,\mathrm{rnd}}$. For each encryption query, $\mathcal{E}_{K,\mathrm{rnd}}$ chooses IV uniformly at random, and returns $(IV, \mathcal{E}_K(IV))$. In the ideal experiment, $\mathcal{E}_{K,\mathrm{rnd}}$ is replaced with an oracle $\$(\cdot)$ that just returns a random IV and a random ciphertext of the same length as the message. A scheme is defined to be secure if an adversary with a reasonable amount of computational resources cannot distinguish the two experiments. We call this black-box security notion IND\$-CPA[22].

Our new notion whIND\$-CPA is defined by extending IND\$-CPA in the same way as whPRI is defined extending PRI security. In the real world, the black-box oracle given to $\mathcal{A}$ is $\mathcal{E}_{K,\mathrm{rnd}}$ only. In the ideal world, the oracle $\$(\cdot)$ is given to both of $\mathcal{A}$ and $\mathcal{S}$. A complete description of the real and ideal experiments can be found in the full version of this paper [40]. The advantage $\mathbf{Adv}^{\mathrm{whIND\$-CPA}}_{\Pi,\mathcal{C}_\Pi,\mathcal{S}}(\mathcal{A})$ is defined as before. We assume that the white-box program given to a lifter contains an implementation of not only encryption but also decryption.

6 Weak Public Indifferentiability and White-Box Security Reductions

This section first introduces a weaker version of public indifferentiability which we name *weak public indifferentiability*. Second, we show that weak public indif-

[22] This name is from [57], though it is defined for nonce-based scheme rather than random-IV schemes.

ferentiability implies reductions between our white-box security notions introduced in Sections 4 and 5. Third, we provide feasibility results that our white-box security notions on various schemes can be reduced to whPRP, by showing weak public indifferentiable constructions.

6.1 Weak Public Indifferentiability and Compositions

An important point to be aware of about public indifferentiability is that it seems quite hard to prove a composition of two arbitrary public indifferentiable scheme become again public indifferentiable. This is because the general “composition theorem” is not applicable to show public indifferentiability of composite schemes due to the following reason. Suppose a scheme U^{Q} (Q is an ideally random primitive) is public indifferentiable from a random object P (e.g., a random oracle). Then, what the general ”composition theorem” for public indifferentiability says is that we can safely replace P in a protocol or construction with U^{Q} if the security of the protocol/construction is defined by single-stage games satisfying the following condition: Queries to P by any party involved in the security games can be made public without affecting the security. (We denote this condition by (C).) Now, assume that there is another scheme T^{P} that is public indifferentiable from a random object R. If (C) were satisfied by the security games of T^{P} (i.e., by the security games of public indifferentiability), public indifferentiability of U^{Q} and the ”composition theorem” would imply public indifferentiability of $\mathsf{T}^{\mathsf{U}^{\mathsf{Q}}}$. However, (C) is not satisfied because queries by a simulator must not be visible to an adversary in the ideal game. Thus the general ”composition theorem” is not applicable to prove public indifferentiability of $\mathsf{T}^{\mathsf{U}^{\mathsf{Q}}}$. (See also Remark 2.)

However, infeasibility of compositions is inconvenient because security proofs cannot be provided in a modular way. To remedy this, we introduce a weaker variant which we name *weak public indifferentiability.* Let T^{P} be a construction querying to an ideally random primitive P, and let R be a random object of which input-output interfaces are compatible with T^{P}. Now, let $\mathsf{Rev}'[\mathsf{R}]$ be a variant of the revealing interface $\mathsf{Rev}[\mathsf{R}]$ that returns the list of all the queries made so far by $\mathcal{A}$, but not by $\mathcal{S}$, together with the responses[23]. We define *weak public indifferentiability* in the same way as public indifferentiability is defined except that the revealing interface is $\mathsf{Rev}'[\mathsf{R}]$ instead of $\mathsf{Rev}[\mathsf{R}]$. Weak public indifferentiability advantage is defined as $\mathsf{Adv}^{\text{weak-pub-indiff}}_{\mathsf{T},\mathsf{R},\mathcal{S}}(\mathcal{A}) := \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{T}^{\mathsf{P}},\mathsf{P}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{R},\mathcal{S}^{\mathsf{R},\mathsf{Rev}'[\mathsf{R}]}}\right]$.

[23] Note that lists returned by $\mathsf{Rev}'[\mathsf{R}]$ contain more useful information for $\mathcal{S}$ than lists returned by $\mathsf{Rev}[\mathsf{R}]$. This is because (1) $\mathcal{S}$ can record what it has queried to R so far by itself, and (2) Sometimes $\mathcal{S}$ cannot tell which queries recorded in a list by $\mathsf{Rev}[\mathsf{R}]$ have been queried by $\mathcal{A}$: If a value x had been queried to R for the first time by $\mathcal{S}$ but not $\mathcal{A}$, there is no means for $\mathcal{S}$ to know whether $\mathcal{A}$ queried x to R afterwards.

A public indifferentiable scheme is weak public indifferentiable[24]. This is because a simulator $\mathcal{S}$ for public indifferentiability can be converted into a one for weak public indifferentiability just by recording queries that $\mathcal{S}$ makes to R.

On Compositions of Two Weak Public Indifferentiable Schemes. Here we explain that a composition of two weak public indifferentiable schemes become weak public indifferentiable if a few additional conditions are satisfied. To explain this, we formally define *random-IV schemes.* Note that we say a construction T^{P} is deterministic if, for an arbitrary input X, the output value $\mathsf{T}^{\mathsf{P}}(X)$ is unchanged during each game.

Definition 4. *A construction* T^{P} *is a* random-IV scheme *if it is a public-coin protocol. Namely, there exists a deterministic construction* $\tilde{\mathsf{T}}^{\mathsf{P}}$ *and a set* $\mathbf{IV}$ *such that, on arbitrary input* X, T^{P} *runs as follows: (1) Take a value* IV *from* $\mathbf{IV}$ *uniformly at random. (2) Return* $(IV, \tilde{\mathsf{T}}^{\mathsf{P}}(IV, X))$.

The following lemma shows the composition of two weak public indifferentiable schemes is again weak public indifferentiable if a few additional conditions are satisfied. Here we provide only an informal version due to page limitation. See the full version of this paper [40] for a formal version and a proof.

Lemma 1 (Composition of weak public indifferentiable schemes, informal). *Suppose the following (1)-(3) hold: (1)* T^{P} *is a* deterministic or random-IV *scheme calling an ideally random primitive* P *and is weak public indifferentiable from* R*, (2)* U^{Q} *is another* deterministic *construction calling an ideally random primitive* Q *and is weak public indifferentiable from* P*, and (3)* P *and* Q *are deterministic. Then* $\mathsf{T}^{\mathsf{U}^{\mathsf{Q}}}$ *is also weak public indifferentiable from* R*, regarding* Q *as the primitive oracle.*

All compositions of (weak public) indifferentiable schemes appearing in this paper satisfy (1)-(3).

Intuition of the Proof. Here we explain a sketch of the proof when all the functions and constructions are deterministic. Suppose the ideal game for T^{U} is being executed with an adversary $\mathcal{A}$.

Let $\mathcal{S}_{\mathsf{T}}$ (resp., $\mathcal{S}_{\mathsf{U}}$) be a "good" simulator for T (resp., U) making the indifferentiability advantage small. Then, a "good" simulator $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ for T^{U} is defined as follows, by using $\mathcal{S}_{\mathsf{T}}$ and $\mathcal{S}_{\mathsf{U}}$ as subroutines: When a value x is queried to $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$, it first runs $\mathcal{S}_{\mathsf{U}}$ on the input x as a subroutine. Intuitively, $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ tries to convince the subroutine $\mathcal{S}_{\mathsf{U}}$ that "now $\mathcal{S}_{\mathsf{U}}$ is run as a part of $\mathcal{A}^{\mathsf{T}^{\mathsf{P}},\mathcal{S}_{\mathsf{U}}^{\mathsf{P},\mathsf{Rev}'[\mathsf{P}]}}$". When $\mathcal{S}_{\mathsf{U}}$ returns an output, $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ returns it to $\mathcal{A}$ as its own output. To achieve this, $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ simulates the oracles P and $\mathsf{Rev}'[\mathsf{P}]$ for $\mathcal{S}_{\mathsf{U}}$. P is simulated just by running $\mathcal{S}_{\mathsf{T}}^{\mathsf{R},\mathsf{Rev}'[\mathsf{R}]}$. (Note

[24] It seems hard to prove weak public indifferentiability implies public indifferentiability, but currently we are not aware of any separation example that is weak public indifferentiable but not public indifferentiable.

that $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ is given oracle access to R and $\mathsf{Rev}'[\mathsf{R}]$.) The non-trivial part is how to simulate the oracle $\mathsf{Rev}'[\mathsf{P}]$.

What the subroutine $\mathcal{S}_{\mathsf{U}}$ is expecting to receive when it makes a query to the revealing interface is a list storing queries (and the responses) to P that are made so far by $\mathcal{A}$ through T (but not by $\mathcal{S}_{\mathsf{U}}$) while running $\mathcal{A}^{\mathsf{T}^{\mathsf{P}},\mathcal{S}_{\mathsf{U}}^{\mathsf{P},\mathsf{Rev}'[\mathsf{P}]}}$. Hence, when the subroutine $\mathcal{S}_{\mathsf{U}}$ makes a query to the revealing interface, $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ simulates the oracle $\mathsf{Rev}'[\mathsf{P}]$ as follows. First, $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ queries to $\mathsf{Rev}'[\mathsf{R}]$ to get the list $\mathsf{List}_{\mathcal{A}}[\mathsf{R}]$ of queries made so far to R by $\mathcal{A}$ (but not by $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$). Then $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ computes the function $\mathsf{T}^{\mathcal{S}_{\mathsf{T}}^{\mathsf{R},\mathsf{Rev}'[\mathsf{R}]}}$ on the input X for each entry (X, Y) in $\mathsf{List}_{\mathcal{A}}[\mathsf{R}]$, recording all the queries by T to $\mathcal{S}_{\mathsf{T}}^{\mathsf{R},\mathsf{Rev}'[\mathsf{R}]}$ into a list $\mathsf{List}_{\mathrm{prim}}$, together with the responses. Finally, $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ returns $\mathsf{List}_{\mathrm{prim}}$ to $\mathcal{S}_{\mathsf{U}}$ as a response. The simulation works well because $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ can tell which value has been queried to R so far by $\mathcal{A}$ (but not by $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$). See the full version of this paper [40] for further details.

Remark 2. The above idea does not work for (original) public indifferentiability. Here we explain which part fails for public indifferentiability. The non-trivial part of the proof is again how to simulate $\mathsf{Rev}[\mathsf{P}]$ for $\mathcal{S}_{\mathsf{U}}$. The issue in simulating $\mathsf{Rev}[\mathsf{P}]$ is also again how to determine the values queried to P through T by $\mathcal{A}$ *but not by* $\mathcal{S}_{\mathsf{U}}$. Now, the procedure "First, $\mathcal{S}_{\mathsf{T}^{\mathsf{U}}}$ queries to $\mathsf{Rev}'[\mathsf{R}]$ to get..." does not work for public indifferentiability due to the property (2) in Footnote 23.

6.2 Weak Public Indifferentiability Implies White-Box Reduction

Let $(\pi, \mathcal{C}_\pi)$ be a white-box symmetric-key scheme that are either of a keyed function, block cipher, AEAD, or a conventional IV-based encryption scheme. In addition, let $(\Sigma^\pi, \mathcal{C}_{\Sigma^\pi})$ be another white-box symmetric-key scheme built on $(\pi, \mathcal{C}_\pi)$. We assume Σ calls π in a black-box manner not only at a level of syntax but also at a level of implementation, i.e., the following conditions are satisfied.

1. The implementation of π (denoted by $[\![\pi]\!]$) is included into the implementation of Σ^π (denoted by $[\![\Sigma^\pi]\!]$). In particular, $[\![\pi]\!]$ and an implementation of an oracle-aided algorithm Σ (which is independent from π) is explicitly separated in $[\![\Sigma^\pi]\!]$.
2. The implementation of Σ calls $[\![\pi]\!]$ in a black-box manner.

Our goal is to reduce the security of $(\Sigma^\pi, \mathcal{C}_{\Sigma^\pi})$ to the security of $(\pi, \mathcal{C}_\pi)$. By sec-const (resp., sec-prim) we denote the security notion corresponding to $(\Sigma^\pi, \mathcal{C}_{\Sigma^\pi})$ (resp., $(\pi, \mathcal{C}_\pi)$), which is whPRI, whPRP, whPRF, wh$\widetilde{\text{SPRP}}$, or whIND\$-CPA[25].

By abuse of notations, we use the same symbols Σ^π and π to denote the corresponding keyed black-box oracles given to $(\mathcal{A}_{\mathrm{create}}, \mathcal{A}_{\mathrm{dist}})$ in the white-box security definitions. We assume Σ^π is a deterministic or random-IV scheme: If

[25] We assume the interfaces of π that Σ accesses to are only those given to $\mathcal{A}$ as black-box oracles in the security games of sec-prim. For instance, if π is a block cipher E_K and sec-prim is whPRP, we assume that Σ calls only E_K and does not call E_K^{-1} (though simulators in the ideal game of whPRP access to both of P and P^{-1}).

it is a random-IV scheme, there exists a scheme $\tilde{\Sigma}^\pi$ and a set $\mathbf{IV}$ such that Σ^π runs as follows on arbitrary input X: (1) IV is chosen uniformly at random from $\mathbf{IV}$. (2) Return $(IV, \tilde{\Sigma}^\pi(IV, X))$.

Let R and P denote the ideal oracles given to a simulator in the ideal games of the security definition of $(\Sigma^\pi, \mathcal{C}_{\Sigma^\pi})$ and $(\pi, \mathcal{C}_\pi)$, respectively. Suppose there exist non-decreasing functions $q_\Sigma(\cdot,\cdot)$ and $\sigma_\Sigma(\cdot,\cdot)$ satisfying the following property: If Σ^π is evaluated on q inputs of which lengths [26] are σ in total during a game, Σ makes at most $q_\Sigma(q,\sigma)$ queries to π and the lengths of the queries are at most $\sigma_\Sigma(q,\sigma)$ in total. In addition, assume we have the following three algorithms.

1. An adversary $\mathcal{A} = (\mathcal{A}_{\text{create}}, \mathcal{A}_{\text{dist}})$ against $(\Sigma^\pi, \mathcal{C}_{\Sigma^\pi})$. The running time is at most $t_\mathcal{A}$. The number of black-box oracle queries by $\mathcal{A}$ is at most $q_\mathcal{A}$ and the lengths of queries are at most $\sigma_\mathcal{A}$ in total. $\mathcal{A}$ creates a lifter running in time t_{lif} and outputs at most λ-bit leakage.
2. A simulator $\mathcal{S}_{\text{prim}}$ for $(\pi, \mathcal{C}_\pi)$ on sec-prim. $\mathcal{S}_{\text{prim}}$ makes at most $q_{\mathcal{S}_{\text{prim}}}$ queries to the ideal oracle P. The lengths of queries are at most $\sigma_{\mathcal{S}_{\text{prim}}}$ in total.
3. A simulator $\mathcal{S}_{\text{indiff}}$ for weak public indifferentiability of Σ^{P} from R[27]. There exist non-decreasing functions $q_{\mathcal{S}_{\text{indiff}}}(\cdot,\cdot,\cdot,\cdot)$ and $\sigma_{\mathcal{S}_{\text{indiff}}}(\cdot,\cdot,\cdot,\cdot)$ satisfying the following properties: If an adversary makes at most q_c (resp., q_p) construction (resp., primitive) queries of which lengths are at most σ_c (resp., σ_p) in total in the ideal game of weak public indifferentiability, $\mathcal{S}_{\text{indiff}}$ makes at most $q_{\mathcal{S}_{\text{indiff}}}(q_c, \sigma_c, q_p, \sigma_p)$ queries to the ideal oracle R. The lengths of the queries are $\sigma_{\mathcal{S}_{\text{indiff}}}(q_c, \sigma_c, q_p, \sigma_p)$ in total.

Theorem 2. *Let $\mathcal{A}$, $\mathcal{S}_{\text{prim}}$, and $\mathcal{S}_{\text{indiff}}$ be as above. Then there exists an adversary $\mathcal{A}' = (\mathcal{A}'_{\text{create}}, \mathcal{A}'_{\text{dist}})$ against $(\pi, \mathcal{C}_\pi)$, a simulator $\mathcal{S}_{\text{const}}$ for $(\Sigma^\pi, \mathcal{C}_{\Sigma^\pi})$, and an algorithm $\mathcal{A}''$ against weak public indifferentiability of Σ such that*

$$\mathbf{Adv}^{\text{sec-const}}_{\Sigma^\pi, \mathcal{C}_{\Sigma^\pi}, \mathcal{S}_{\text{const}}}(\mathcal{A}) = \mathbf{Adv}^{\text{sec-prim}}_{\pi, \mathcal{C}_\pi, \mathcal{S}_{\text{prim}}}(\mathcal{A}') + \mathbf{Adv}^{\text{weak-pub-indiff}}_{\Sigma, \mathsf{R}, \mathcal{S}_{\text{indiff}}}(\mathcal{A}'') \qquad (1)$$

holds. Here, we can construct $\mathcal{S}_{\text{const}}$, $\mathcal{A}'$, and A'' so that (a) $\mathcal{A}'$ does not depend on $\mathcal{S}_{\text{prim}}$ and $\mathcal{S}_{\text{indiff}}$, (b) $\mathcal{S}_{\text{const}}$ does not depend on $\mathcal{A}$, (c) $\mathcal{A}''$ does not depend on $\mathcal{S}_{\text{indiff}}$, and the following conditions hold: (1) $\mathcal{S}_{\text{const}}$ makes at most $q_{\mathcal{S}_{\text{indiff}}}(q_\mathcal{A}, \sigma_\mathcal{A}, q'_\Sigma + q_{\mathcal{S}_{\text{prim}}}, \sigma'_\Sigma + \sigma_{\mathcal{S}_{\text{prim}}})$ queries to R. The lengths of the queries are at most $\sigma_{\mathcal{S}_{\text{indiff}}}(q_\mathcal{A}, \sigma_\mathcal{A}, q'_\Sigma + q_{\mathcal{S}_{\text{prim}}}, \sigma'_\Sigma + \sigma_{\mathcal{S}_{\text{prim}}})$ in total. Here, $q'_\Sigma := q_\Sigma(q_\mathcal{A}, \sigma_\mathcal{A})$ and $\sigma'_\Sigma := q_\Sigma(q_\mathcal{A}, \sigma_\mathcal{A})$ (2) $\mathcal{A}'$ runs in time $O(t_\mathcal{A} + \sigma'_\Sigma)$ and makes at most q'_Σ queries to a black-box oracle. The lengths of the queries are at most σ'_Σ in total. $\mathcal{A}'$ creates a lifter $\mathcal{L}'$ that runs in time $O(t_{\text{lif}})$ and outputs at most λ-bit leakage. (3) $\mathcal{A}''$ makes at most $q_\mathcal{A}$ construction queries of which lengths are at most $\sigma_\mathcal{A}$ in total, and makes at most $q'_\Sigma + q_{\mathcal{S}_{\text{prim}}}$ primitive queries of which lengths are at most $\sigma'_\Sigma + \sigma_{\mathcal{S}_{\text{prim}}}$ in total.

[26] The unit of length can be set arbitrarily (e.g., bit or block) depending on the context.

[27] Since P is the oracle given to a simulator while π is the black-box oracle given to an adversary in the security games of sec-prim, Σ may access to only a part of the interfaces of P: If sec-prim is whPRP and $\pi = E_K$, P is the pair (P, P^{-1}) (here, P is a random permutation) but Σ accesses only to P (and not to P^{-1}) because the black-box oracle interface given to an adversary $\mathcal{A}$ in the definition of whPRP is only E_K (and E_K^{-1} is not given to $\mathcal{A}$).

Interpretation of Theorem 2. The above theorem indeed shows that $(\Sigma^{\pi}, \mathcal{C}_{\Sigma^{\pi}})$ is a secure white-box scheme w.r.t. sec-const if the underlying scheme $(\pi, \mathcal{C}_{\pi})$ is secure w.r.t. sec-prim and Σ^{P} is weak public indifferentiable from R: Let $\mathcal{A}$ be an adversary attacking $(\Sigma^{\pi}, \mathcal{C}_{\Sigma^{\pi}})$. Then, we can construct an adversary $\mathcal{A}'$ to attack $(\pi, \mathcal{C}_{\pi})$ as in Theorem 2. If $(\pi, \mathcal{C}_{\pi})$ is secure (w.r.t. sec-prim), then there is a simulator $\mathcal{S}_{\text{prim}}$ for $(\pi, \mathcal{C}_{\pi})$ that makes $\mathbf{Adv}^{\text{sec-prim}}_{\pi,\mathcal{C}_{\pi},\mathcal{S}_{\text{prim}}}(\mathcal{A}')$ small. In addition, if Σ^{P} is weak public indifferentiable from R, then there exists a simulator $\mathcal{S}_{\text{indiff}}$ making $\mathbf{Adv}^{\text{weak-pub-indiff}}_{\Sigma,\mathcal{S}_{\text{indiff}}}(\mathcal{A}'')$ small, where $\mathcal{A}''$ is the adversary built from $\mathcal{A}$ and $\mathcal{S}_{\text{prim}}$ as in the theorem. Again, Theorem 2 assures that we can construct $\mathcal{S}_{\text{const}}$ from $\mathcal{S}_{\text{prim}}$ and $\mathcal{S}_{\text{indiff}}$ such that $\mathbf{Adv}^{\text{sec-const}}_{\Sigma^{\pi},\mathcal{C}_{\Sigma^{\pi}},\mathcal{S}_{\text{const}}}(\mathcal{A})$ satisfies Eq. (1). If all the parameters appearing in Theorem 2 are not so large, the advantage $\mathbf{Adv}^{\text{sec-const}}_{\Sigma^{\pi},\mathcal{C}_{\Sigma^{\pi}},\mathcal{S}_{\text{const}}}(\mathcal{A})$ is sufficiently small.

Intuition of the Proof. Here we provide a rough sketch on why Σ^{π} becomes secure if π is secure and Σ^{P} satisfies the *original* indifferentiability. Let $\mathcal{S}_{\text{indiff}}$ be a simulator making the indifferentiability advantage of Σ^{P} small. We consider the following three games.

1. [The real world (for Σ^{π} on sec-const).] The adversary $\mathcal{A} = (\mathcal{A}_{\text{create}}, \mathcal{A}_{\text{dist}})$ is given a black-box oracle access to Σ^{π}. A lifter L is given a white-box implementation of Σ^{π}.
2. [Intermediate world.] The black-box oracle of π and the lifter L in the real world are replaced with a random permutation P and a simulator $\mathcal{S}_{\text{prim}}$ (for π on sec-prim), respectively. The adversary $\mathcal{A} = (\mathcal{A}_{\text{create}}, \mathcal{A}_{\text{dist}})$ and $\S_{\text{prim}}$ are given oracle access to Σ^{P} and P, respectively. Especially, this game executes three algorithms $\mathcal{A}^{\Sigma^{\mathsf{P}}}_{\text{create}}$, $\mathcal{S}^{\mathsf{P}}_{\text{prim}}$, and $\mathcal{A}^{\Sigma^{\mathsf{P}}}_{\text{dist}}$.
3. [The ideal world] The black-box oracle given to $\mathcal{A} = (\mathcal{A}_{\text{create}}, \mathcal{A}_{\text{dist}})$ is R. In addition, the simulator (for Σ^{π} on sec-const) is defined to be $\mathcal{S}^{\mathcal{S}_{\text{indiff}}}_{\text{prim}}$. $\mathcal{S}^{\mathcal{S}_{\text{indiff}}}_{\text{prim}}$ is also given an oracle access to R. Especially, this game executes three algorithms $\mathcal{A}^{\mathsf{R}}_{\text{create}}$, $\mathcal{S}^{\mathcal{S}^{\mathsf{R}}_{\text{indiff}}}_{\text{prim}}$, and $\mathcal{A}^{\mathsf{R}}_{\text{dist}}$.

If π is secure, then we can replace π (in Σ^{π}) and a lifter in the real world with P and a simulator $\mathcal{S}_{\text{prim}}$, respectively, with a small security loss. That is, the difference between the first and the second worlds is small. Next, regarding the tuple $(\mathcal{A}_{\text{create}}, \mathcal{S}_{\text{prim}}, \mathcal{A}_{\text{dist}})$ as a single algorithm, we can regard the intermediate world as a game where a single-stage adversary $(\mathcal{A}_{\text{create}}, \mathcal{S}_{\text{prim}}, \mathcal{A}_{\text{dist}})$ runs relative to the oracles $(\Sigma^{\mathsf{P}}, \mathsf{P})$. Moreover, we can also regard the ideal wold as a game where the single algorithm $(\mathcal{A}_{\text{create}}, \mathcal{S}_{\text{prim}}, \mathcal{A}_{\text{dist}})$ runs relative to the oracles $(\mathsf{R}, \mathcal{S}^{\mathsf{R}}_{\text{indiff}})$. Especially, the difference between the intermediate and ideal worlds matches the indifferentiability advantage of the *single* algorithm $(\mathcal{A}_{\text{create}}, \mathcal{S}_{\text{prim}}, \mathcal{A}_{\text{dist}})$ against T^{P} and R with respect to $\mathcal{S}_{\text{indiff}}$[28]. Since Σ^{P} is indifferentiable from R by $\mathcal{S}_{\text{indiff}}$, the difference between the intermediate world and the ideal world is also small.

[28] This is the reason that we can utilize the indifferentiability of Σ^{P} from R to show the security of Σ^{π} although the security games of Σ^{π} are not single-stage games.

In fact there are some subtleties on how to simulate list of queries passed to $\mathcal{S}_{\mathrm{prim}}$. Moreover, when we consider weak public indifferentiability instead of original indifferentiability, we also have to consider how to simulate the revealing interface $\mathsf{Rev}'[\mathsf{R}]$. See the full version of this paper [40] for details.

6.3 Feasibility Results

This section shows feasibility results that various white-box security notions can be reduced to that of block ciphers (whPRP) and FIL/FOL keyed functions (whPRF) like in the black-box setting. We only prove (weak) public indifferentiability of the constructions because Theorem 2 shows white-box security reductions follow from (weak) public indifferentiability.

whPRP-whPRF Switch. Let P be an n-bit random permutation. Then, regarding (P, P^{-1}) as a primitive oracle, P is public indifferentiable from a random function $\mathsf{RF} : \{0,1\}^n \rightarrow \{0,1\}^n$. (In the real world, the construction oracle is P and the primitive oracle is (P, P^{-1}).) Specifically, the proposition below holds.

Proposition 1. *There is a simulator $\mathcal{S}$ making at most q_p queries to* RF *satisfying $\mathbf{Adv}^{\text{pub-indiff}}_{P,\mathsf{RF},\mathcal{S}}(\mathcal{A}) \leq \frac{(q_c+q_p)^2}{2^n}$ for any adversary $\mathcal{A}$ making at most q_c and q_p queries to the construction and primitive oracles, respectively.*

The proof is quite straightforward. See the full version of this paper [40] for a complete proof. Together with Theorem 2, this proposition implies that a whPRP-secure BC is a whPRF-secure keyed function.

Reduction from whPRP to whPRF. The 6-round Feistel construction is public indifferentiable from a random invertible permutation when round functions are random functions [49]. Thus we can build a whPRP-secure BC from a whPRF-secure keyed function.

Reduction from VIL/VOL-whPRF to FIL/FOL-whPRF. The indifferentiability result of the sponge construction (Theorem 1) implies that we can build VIL/VOL-whPRF from FIL/FOL-whPRF. We can also build VIL/FOL-whPRF from FIL/FOL-whPRF by the Merkle-Damgård construction since it is public indifferentiable [32].

Reduction from whPRI to FIL/FOL-whPRF. By the result of Barbosa and Farshim [6], an indifferentiable AEAD can be constructed from a FIL/FOL random function by a scheme based on (unbalanced) 3-round Feistel that uses the sponge construction as round functions. (See Theorem 5 of the full version of this paper [40] and the explanation below for more details.) Thus we can build a whPRI-secure AEAD from a whPRF-secure FIL/FOL keyed function. In Section 7 we show a more practical construction.

Reduction from wh$\widetilde{\textbf{SPRP}}$ to whPRF. A weak public indifferentiable VIL tweakable ideally random permutation can be built from a FIL/FOL random function f, by a (balanced) 6-round Feistel construction of which round functions are the sponge construction using f as an underlying primitive. See the full version of this paper [40] for more details.

Reduction from whIND\$-CPA to whPRF. Let us modify the encryption oracle of CTR in such a way that (1) a uniformly random IV is chosen for each encryption query (rather than IV is chosen by adversary) and IV is returned together with the ciphertext, and (2) the underlying keyed function of CTR is replaced with a random function ρ. We denote the resulting encryption oracle by $\mathcal{E}^{\rho}_{\mathrm{rnd}}$. Then $\mathcal{E}^{\rho}_{\mathrm{rnd}}$ is public indifferentiable from $\$(\cdot)$ that appears in the ideal experiment of whIND\$-CPA. More precisely, the following proposition holds.

Proposition 2. *There exists a simulator $\mathcal{S}$ making at most q_c queries to the $\$(\cdot)$, where the lengths of queries are at most σ blocks in total, that satisfies $\mathbf{Adv}^{\text{pub-indiff}}_{\mathcal{E}_{\mathrm{rnd}},\$(\cdot),\mathcal{S}}(\mathcal{A}) \leq \frac{\sigma^2}{2^m} + \frac{\sigma(\sigma+q_p)}{2^m}$ for any adversary $\mathcal{A}$ making at most q_c queries to the construction oracle of which lengths are at most σ blocks in total and q_p queries to the primitive oracle.*

Intuition of the Proof. For simplicity, we assume $\mathrm{len}(M)$ is always a multiple of n and denote the i-th block of M by M_i, i.e., $M = M_1 || \cdots || M_{\mathrm{len}(M)/n}$. Roughly speaking, the simulator $\mathcal{S}$ runs as follows: Let $\mathsf{List}[\$(\cdot)]$ be the list storing queries made so far to $\$(\cdot)$ and the responses. When a fresh value x is queried to the interface corresponding to ρ, the simulator first queries to the revealing interface to get $\mathsf{List}[\$(\cdot)]$. If there exists $(M, (IV, C)) \in \mathsf{List}[\$(\cdot)]$ such that $x = IV + i - 1$ for $1 \leq i \leq \mathrm{len}(M)/n$, the adversary may be trying to compute the i-th block of C itself. Thus the simulator sets the value $\rho(x)$ as $\rho(x) := M_i \oplus C_i$ so that the adversary cannot notice that ρ is simulated. If such $(M, (IV, C))$ does not exist in $\mathsf{List}[\$(\cdot)]$, the value $\rho(x)$ is just randomly sampled.

The simulator may not be able to sample the value $\rho(x)$ in compatible with C and fail if the following (a) or (b) happen: (a) when a message M is queried to $\$(\cdot)$ and IV is randomly chosen, $IV + i = IV' + j$ holds for some $(M', (IV', C')) \in \mathsf{List}[\$'(\cdot)]$, where $0 \leq i < \mathrm{len}(M)/n$ and $0 \leq j < \mathrm{len}(M')/n$. (b) when a message M is queried to $\$(\cdot)$ and IV is randomly chosen, the value $IV + i$ $(0 \leq i < \mathrm{len}(M)/n)$ collides with a value x on which the output value of ρ is already defined. The events (a) and (b) correspond to the terms $\frac{\sigma^2}{2^m}$ and $\frac{\sigma(\sigma+q_p)}{2^m}$ in the security bound, respectively. If both of (a) and (b) do not happen in the ideal world, then outputs of ρ are appropriately simulated in compatible with ciphertexts, and thus an adversary cannot distinguish the ideal world from the real world. See the full version of this paper [40] for a complete proof.

On Reduction of Pairs and Generic Compositions We also observe feasibility of reductions of pairs (i.e., providing proof in a modular way), and infeasibility of generic compositions for AEADs. See the full version [40] for details.

7 A Search for a Practical whPRI-Secure AEAD Mode

This section shows a practical AEAD mode to convert whPRP into whPRI. Section 7.1 shows that SIV with CTR is public indifferentiable from a fixed-key random injection when the tag-generation (or, MAC) part is a single VIL/FOL random function f and the underlying keyed function of CTR is a FIL/FOL random function ρ. Then, in Section 7.2, we replace ρ and f with keyed functions built from a single whPRP, and observe that the resulting scheme is a whPRI-secure AEAD.

7.1 Public Indifferentiability of SIV+CTR

Let $\mathsf{CTR}^{\rho}(IV, M)$ denote the encryption function of the counter mode with the underlying keyed function being replaced with a random function $\rho : \{0,1\}^{\tau} \to \{0,1\}^{n}$ $(\tau \leq n)$. In addition, let $\mathit{\Pi} = (\mathcal{E}^{f,\rho}, \mathcal{D}^{f,\rho})$ be the SIV construction of which keyed function for tag-generation is replaced with a random function $f : \mathbf{N} \times \mathbf{A} \times \{0,1\}^{*} \to \{0,1\}^{\tau}$ and conventional encryption scheme is replaced with CTR^{ρ}. Let $\mathsf{enc} : \mathbf{N} \times \mathbf{A} \times \{0,1\}^{*} \to \{0,1\}^{*}$ be an arbitrary encoding function that encodes each tuple (N, A, X) into a single bit string in a uniquely decodable manner. We let $\mathrm{len}(N, A, X) := \mathrm{len}(\mathsf{enc}(N, A, X))$ and call $\lceil \mathrm{len}(X)/n \rceil$ the block length of a bit string of X. The following theorem shows $\mathit{\Pi}$ is public indifferentiable from a random injection.

Theorem 3. *Let $F : \mathbf{N} \times \mathbf{A} \times \{0,1\}^{*} \to \{0,1\}^{*}$ be a fixed-key random injection with message space $\{0,1\}^{*}$ and such that $\mathrm{len}(F(N, A, M)) = \mathrm{len}(M) + \tau$. There exists a simulator $\mathcal{S}$ for public indifferentiability of $\mathit{\Pi}$ from $F^{\pm}$, where a primitive oracle is (f, ρ), such that the number of queries by $\mathcal{S}$ to the construction oracle is at most q_f and the block lengths of the queries are at most σ_f in total, and*

$$\mathbf{Adv}^{\text{pub-indiff}}_{\mathit{\Pi},F^{\pm},\mathcal{S}}(\mathcal{A}) \leq \frac{(\sigma_c + \sigma_f)^2}{2^{\tau}} + \frac{(\sigma_c + \sigma_f)(q_\rho + \sigma_c)}{2^{\tau}} + \frac{3q_c}{2^{\tau}} + \frac{(q_c + q_f)^2}{2^{\tau}} \quad (2)$$

holds for any adversary $\mathcal{A}$ of which computational resources are as follows: To the construction oracle, $\mathcal{A}$ makes at most q_c queries of which block lengths are at most σ_c in total. To the first primitive oracle (corresponding to f), $\mathcal{A}$ makes at most q_f queries of which block lengths are at most σ_f in total. To the second primitive oracle (corresponding to ρ), $\mathcal{A}$ makes at most q_ρ queries. Here, we assume $(q_c + q_f) \leq 2^{\tau-1}$.

Intuition of the Proof. For simplicity, we assume $\mathrm{len}(M)$ is always a multiple of n. For each (N, A, M), we assume $F(N, A, M)$ is divided as $F(N, A, M) = IV||C_1|| \cdots ||C_\ell$, where $IV \in \{0,1\}^{\tau}$ and $C_1, \ldots, C_\ell \in \{0,1\}^{n}$. The simulation of ρ is almost the same as that for the proof of random-IV CTR (See the explanation below Proposition 2. Here, $\$(\cdot)$ in Proposition 2 is replaced with F.). Simulation of $f(N, A, M)$ is done just by querying (N, A, M) to F and return $F_0(N, A, M)$. Intuitively, the simulation does not work well if the simulation of ρ fails (the events (a) and (b) in the explanation below Proposition 2), or (c) an adversary computes $\mathcal{D}^{f,\rho}(N, A, C)$ itself for a tuple (N, A, C) such that C has never been returned from F, and $\mathcal{D}^{f,\rho}(N, A, C)$ happens to be a value that is *not* $\bot$. The events (a) and (b) correspond to the terms $\frac{(\sigma_c+\sigma_f)^2}{2^\tau}$ and $\frac{(\sigma_c+\sigma_f)(q_\rho+\sigma_c)}{2^\tau}$ of Eq. (2), respectively. Due to (c), an additional term $\frac{q_c}{2^\tau}$ is added. Moreover, we need another term $\frac{(q_c+q_f)^2+2q_c}{2^\tau}$ to deal with lazy sampling of a random injection. See the full version of this paper [40] for a complete proof.

7.2 Instantiation with Block Ciphers

This section discusses how to combine the scheme in the previous subsection with a whPRP-secure block cipher to build a whPRI-secure AEAD.

Assume $\tau < n$ and let $P : \{0,1\}^n \to \{0,1\}^n$ a random permutation. Define $P_0 : \{0,1\}^{n-1} \to \{0,1\}^{n-1}$ and $P_1 : \{0,1\}^\tau \to \{0,1\}^n$ by $P_0(x) :=$ (The lower $(n-1)$ bits of $P(0||x)$) and $P_1(x) := P(1^{n-\tau}||x)$. Set $\mathbf{N} = \{0,1\}^{n/2}$. Let enc be an encoding function such that $\mathsf{enc}(N, A, M) = N||A||M||\mathrm{len}(M)$. (We assume $\mathrm{len}(M)$ is represented as an $n/4$-bit string.) In addition, define $\mathrm{MAC}^P(N, A, M) := \mathsf{Sponge}^{P_0}(\mathsf{enc}(N, A, M))$. Replace f (tag generation function) and ρ (underlying function of CTR) of Π in Theorem 3 with MAC^P and P_1, and denote the resulting scheme by $\Pi[P]$.

Then, $\Pi[P]$ is weak public indifferentiable from a fixed-key random injection $F^\pm$ when $P^\pm$ is regarded as a primitive oracle[29]. Furthermore, if we replace P with a whPRP-secure block cipher E_K, the resulting scheme $\Pi[E_K]$ becomes a whPRI-secure AEAD by Theorem 2. (Here, we assume $\Pi[E_K]$ is implemented in such a way that the implementation of the mode is explicitly separated from the implementation of E_K and the former calls the latter in a black-box manner, so that Theorem 2 can be applied.) More precisely, the following corollary holds. (See the full version [40] for more details on how to derive the corollary.)

Corollary 1. *Let $\mathcal{A}$ be an adversary against $(\Pi[E_K], \mathcal{C}_{\Pi[E_K]})$ on whPRI-security. The running time of $\mathcal{A}$ is at most t. The number of queries by $\mathcal{A}$ to a black-box oracle is at most q and the block lengths of the queries are at most σ in total. $\mathcal{A}$ creates a lifter running in time t_{lif} and outputs at most λ-bit leakage. In addition, let $\mathcal{S}_{\mathrm{prim}}$ be a simulator for $(E, \mathcal{C}_E)$ on whPRP-security. $\mathcal{S}_{\mathrm{prim}}$ makes at most q_{sim} queries to $P^\pm$. Then there exists a simulator $\mathcal{S}_{\mathrm{const}}$ for $(\Pi[E_K], \mathcal{C}_{\Pi[E_K]})$ on whPRI-security and an adversary $\mathcal{A}'$ against $(E, \mathcal{C}_E)$ on whPRP-security such that $\mathbf{Adv}^{\mathrm{whPRI}}_{\Pi[E_K],\mathcal{C}_{\Pi[E_K]},\mathcal{S}_{\mathrm{const}}}(\mathcal{A})$ is upper bounded by $\mathbf{Adv}^{\mathrm{whPRP}}_{E_K,\mathcal{C}_{E_K},\mathcal{S}_{\mathrm{prim}}}(\mathcal{A}')+$ $\frac{\lceil\frac{n}{r}\rceil^6(10\lceil\frac{n}{r}\rceil\sigma+q_{\mathrm{sim}})^4}{2^\tau}+\frac{\lceil\frac{n}{r}\rceil^4(10\lceil\frac{n}{r}\rceil\sigma+q_{\mathrm{sim}})^3}{2^\tau}+\frac{\lceil\frac{n}{r}\rceil^2(10\lceil\frac{n}{r}\rceil\sigma+q_{\mathrm{sim}})^2}{2^\tau}+\frac{\lceil\frac{n}{r}\rceil^2(9\lceil\frac{n}{r}\rceil\sigma+2q_{\mathrm{sim}})^2}{2^n}+$ $\frac{3q_c}{2^\tau}+\epsilon\left(2\left\lceil\frac{n}{r}\right\rceil\left(8\left\lceil\frac{n}{r}\right\rceil\sigma+q_{\mathrm{sim}}\right)\right)$, where $\epsilon(j) = 1 - \Pi_{i=1}^{j}(1-\frac{1}{2^c})$. $\mathcal{S}_{\mathrm{const}}$ makes at most $\left\lceil\frac{n}{r}\right\rceil\left(9\left\lceil\frac{n}{r}\right\rceil\sigma+q_{\mathrm{sim}}\right)$ queries to $F^\pm$ and the lengths of the queries are at most*

[29] This is because (1) an invertible permutation is public indifferentiable from a random function (Proposition 1), (2) the sponge construction is indifferentiable from a random oracle (Theorem 1), (3) the scheme Π in Theorem 3 is public indifferentiable from a fixed-key random injection, (4) composition of deterministic weak public indifferentiable schemes are again weak public indifferentiable (Lemma 1, or its formal version in the full version of this paper [40]).

$(\lceil \frac{n}{r} \rceil^3 (9 \lceil \frac{n}{r} \rceil \sigma + q_{\mathrm{sim}})^2)$ blocks in total. $\mathcal{A}'$ runs in time $O(t+\sigma)$ and makes at most $(2\sigma + \lceil \frac{n}{r} \rceil q)$ black-box oracle queries. $\mathcal{A}'$ outputs a lifter that runs in time $O(t_{\mathrm{lif}})$ and outputs at most λ bits of leakage.

Interpretation of Corollary 1. Let us set $\tau = n-1$ and $(r,c) = (n/2, n/2-1)$. Then, Corollary 1 says that $\mathbf{Adv}^{\mathrm{whPRI}}_{\Pi[E_K],\mathcal{C}_{\Pi[E_K]},\mathcal{S}_{\mathrm{const}}}(\mathcal{A})$ becomes small as long as $\mathbf{Adv}^{\mathrm{whPRP}}_{E,\mathcal{C}_E,\mathcal{S}_{\mathrm{prim}}}(\mathcal{A})$ is small and $q_{\mathrm{sim}}, q, \sigma \ll 2^{n/4}$. This means the following: Let λ and t_{lif} be some reasonable parameters ($\ll 2^{n/4}$) and assume the underlying block cipher E_K is a secure whPRP. More concretely, let $\mathcal{A}'$ be an adversary attacking E_K with $t \ll 2^{\kappa}$ and $q \ll 2^{n/4}$, and $\mathcal{L}$ be a lifter running in time $t_{\mathrm{lif}}(< 2^{n/4})$ that leaks at most λ-bit leakage. Suppose, for any such $\mathcal{A}'$ and $\mathcal{L}$, there exists $\mathcal{S}_{\mathrm{prim}}$ with making $q_{\mathrm{sim}}(\ll 2^{n/4})$ queries such that $\mathbf{Adv}^{\mathrm{whPRP}}_{E,\mathcal{C}_E,\mathcal{S}_{\mathrm{prim}}}(\mathcal{A}')$ is sufficiently small. Then $\Pi[E_K]$ is whPRI-secure against an adversary $\mathcal{A}$ as long as (1) the running time of $\mathcal{A}$ is $\ll 2^{\kappa}$, (2) the length of messages processed by $\Pi[E_K]$ is $\ll 2^{n/4}$ blocks in total while running $\mathcal{A}$ in the real world, and (3) the running time and leakage of a lifter (output by $\mathcal{A}$) are at most t_{lif} and λ bits [30].

On Underlying Block Cipher. The above discussions show that $\Pi[E_K]$ is whPRI-secure if E_K is whPRP-secure and the amount of data processed by $\Pi[E_K]$ is $\ll 2^{n/4}$. As a candidate of whPRP-secure BC, we conjecture that SPACE is whPRP-secure for some parameter settings (see Section 5.1). However, the block length of SPACE is basically $n = 128$ only, when $2^{n/4} = 2^{32}$. In practical use cases, the limitation of 2^{32} is inconvenient and unsatisfactory.

Thus, we propose a 256-bit block variant of SPACE-16, which we name SPACE256-16. Its details are provided in the full version of this paper [40], where we discuss its security against various attacks following the convention of block cipher designs. We conjecture[31] that SPACE256-16 is a $(\lambda, t, q, t_{\mathrm{lif}}, q_{\mathrm{sim}}, \epsilon)$-secure whPRP with $\lambda \approx 2^{20}$, $t \approx 2^{128}$, $q \approx 2^{64}$, $q_{\mathrm{sim}} \approx 2^{64}$, and $\epsilon \ll 1$, as long as $t_{\mathrm{lif}} \ll q_{\mathrm{sim}}$. Assuming our conjecture is true, $\Pi[E_K]$ with E_K instantiated with SPACE256-16 is secure until the amount of processed data is $\ll 2^{64}$ (and the amount of leakage is $< 2^{20}$).

To evaluate the performance, we implemented $\Pi[E_K]$ using SPACE256-16 on a single core in a laptop PC with Intel Core i7-1065G7, being Turbo Boost and hyperthreading disabled. The implementation size is in the order of KB or MB. As a result, the performance reaches about 530 CPB when a 1KB message is processed. Considering the performance of raw SPACE-16 is 305.11 cpb [22], we believe our mode of operation achieves relevant performance.

The limit of leakage for SPACE256-16 is not large. Still, in the same way as (the original, 128-bit-block) SPACE-32 and SPACE-24 provide better security

[30] Note that λ does not explicitly appear in the upper bound of $\mathbf{Adv}^{\mathrm{whPRI}}_{\Pi[E_K],\mathcal{C}_{\Pi[E_K]},\mathcal{S}_{\mathrm{const}}}(\mathcal{A})$ in the corollary. This is because we (implicitly) assume $\lambda \le q_{\mathrm{sim}}$ and the effect of λ is absorbed into q_{sim} in the security bound.

[31] This conjecture is obtained by changing the settings of n and n_a in our conjecture in Section 5.1 to $n = 256$ and $n_a = 16$. κ is still 128.

than SPACE-16 does, a better limit could be achieved by 256-bit-block versions of SPACE-32 or SPACE-24 (at the cost of performance). We introduced a 256-bit-block version of SPACE-16 rather than SPACE-32 or SPACE-24 to balance security and performance. Improving the performance and the limit of the leakage is an interesting future work. This could be achieved by improving SPACE-hard block ciphers, modes of operations, or both.

References

1. Agrawal, S., Bock, E.A., Chen, Y., Watson, G.J.: White-box cryptography with device binding from token-based obfuscation and more. IACR Cryptology ePrint Archive 2021/767 (2021)
2. Alliance, S.C.: A smart card alliance mobile & nfc council white paper, host card emulation (hce) 101 (2014)
3. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert, H. (ed.) EUROCRYPT 2010, Proceedings. LNCS, vol. 6110, pp. 113–134. Springer (2010)
4. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009, Proceedings. LNCS, vol. 5677, pp. 36–54. Springer (2009)
5. Banik, S., Bogdanov, A., Isobe, T., Jepsen, M.B.: Analysis of software countermeasures for whitebox encryption. IACR Trans. Symmetric Cryptol. **2017**(1), 307–328 (2017)
6. Barbosa, M., Farshim, P.: Indifferentiable authenticated encryption. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Proceedings, Part I. LNCS, vol. 10991, pp. 187–220. Springer (2018)
7. Barwell, G., Martin, D.P., Oswald, E., Stam, M.: Authenticated encryption in the face of protocol and side channel leakage. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Proceedings, Part I. LNCS, vol. 10624, pp. 693–723. Springer (2017)
8. Bellare, M., Dai, W.: Defending against key exfiltration: Efficiency improvements for big-key cryptography via large-alphabet subkey prediction. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017, Proceedings. pp. 923–940. ACM (2017)
9. Bellare, M., Kane, D., Rogaway, P.: Big-key symmetric encryption: Resisting key exfiltration. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Proceedings, Part I. LNCS, vol. 9814, pp. 373–402. Springer (2016)
10. Berti, F., Guo, C., Pereira, O., Peters, T., Standaert, F.: Tedt, a leakage-resist AEAD mode for high physical security applications. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2020**(1), 256–320 (2020)
11. Berti, F., Pereira, O., Peters, T., Standaert, F.: On leakage-resilient authenticated encryption with decryption leakages. IACR Trans. Symmetric Cryptol. **2017**(3), 271–293 (2017)
12. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: On the indifferentiability of the sponge construction. In: Smart, N.P. (ed.) EUROCRYPT 2008, Proceedings. LNCS, vol. 4965, pp. 181–197. Springer (2008)
13. Billet, O., Gilbert, H., Ech-Chatbi, C.: Cryptanalysis of a White Box AES Implementation. In: SAC 2004, Revised Selected Papers. pp. 227–240 (2004)

14. Biryukov, A., Bouillaguet, C., Khovratovich, D.: Cryptographic schemes based on the ASASA structure: Black-box, white-box, and public-key (extended abstract). In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Proceedings, Part I. LNCS, vol. 8873, pp. 63–84. Springer (2014)
15. Biryukov, A., Perrin, L.: Symmetrically and asymmetrically hard cryptography. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Proceedings, Part III. LNCS, vol. 10626, pp. 417–445. Springer (2017)
16. Biryukov, A., Udovenko, A.: Attacks and countermeasures for white-box designs. In: Peyrin, T., Galbraith, S.D. (eds.) ASIACRYPT 2018, Proceedings, Part II. LNCS, vol. 11273, pp. 373–402. Springer (2018)
17. Biryukov, A., Udovenko, A.: Dummy shuffling against algebraic attacks in white-box implementations. In: Canteaut, A., Standaert, F. (eds.) EUROCRYPT 2021, Proceedings, Part II. LNCS, vol. 12697, pp. 219–248. Springer (2021)
18. Bock, E.A., Amadori, A., Bos, J.W., Brzuska, C., Michiels, W.: Doubly half-injective prgs for incompressible white-box cryptography. In: Matsui, M. (ed.) CT-RSA 2019, Proceedings. LNCS, vol. 11405, pp. 189–209. Springer (2019)
19. Bock, E.A., Amadori, A., Brzuska, C., Michiels, W.: On the security goals of white-box cryptography. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2020**(2), 327–357 (2020)
20. Bock, E.A., Brzuska, C., Fischlin, M., Janson, C., Michiels, W.: Security reductions for white-box key-storage in mobile payments. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Proceedings, Part I. LNCS, vol. 12491, pp. 221–252. Springer (2020)
21. Bogdanov, A., Isobe, T.: White-box cryptography revisited: Space-hard ciphers. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015, Proceedings. pp. 1058–1069. ACM (2015)
22. Bogdanov, A., Isobe, T., Tischhauser, E.: Towards practical whitebox cryptography: Optimizing efficiency and space hardness. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Proceedings, Part I. LNCS, vol. 10031, pp. 126–158 (2016)
23. Bos, J.W., Hubain, C., Michiels, W., Teuwen, P.: Differential computation analysis: Hiding your white-box designs is not enough. In: Gierlichs, B., Poschmann, A.Y. (eds.) CHES 2016, Proceedings. LNCS, vol. 9813, pp. 215–236. Springer (2016)
24. Cash, D., Ding, Y.Z., Dodis, Y., Lee, W., Lipton, R.J., Walfish, S.: Intrusion-resilient key exchange in the bounded retrieval model. In: Vadhan, S.P. (ed.) TCC 2007, Proceedings. LNCS, vol. 4392, pp. 479–498. Springer (2007)
25. Chow, S., Eisen, P.A., Johnson, H., van Oorschot, P.C.: White-box cryptography and an AES implementation. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002, Revised Papers. LNCS, vol. 2595, pp. 250–270. Springer (2002)
26. Chow, S., Eisen, P.A., Johnson, H., van Oorschot, P.C.: A white-box DES implementation for DRM applications. In: Feigenbaum, J. (ed.) Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002, Revised Papers. LNCS, vol. 2696, pp. 1–15. Springer (2002)
27. Crescenzo, G.D., Lipton, R.J., Walfish, S.: Perfectly secure password protocols in the bounded retrieval model. In: Halevi, S., Rabin, T. (eds.) TCC 2006, Proceedings. LNCS, vol. 3876, pp. 225–244. Springer (2006)
28. Degabriele, J.P., Janson, C., Struck, P.: Sponges resist leakage: The case of authenticated encryption. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Proceedings, Part II. LNCS, vol. 11922, pp. 209–240. Springer (2019)
29. Delerablée, C., Lepoint, T., Paillier, P., Rivain, M.: White-box security notions for symmetric encryption schemes. In: Lange, T., Lauter, K.E., Lisonek, P. (eds.) SAC 2013, Revised Selected Papers. LNCS, vol. 8282, pp. 247–264. Springer (2013)

30. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Unterluggauer, T.: ISAP - towards side-channel secure authenticated encryption. IACR Trans. Symmetric Cryptol. **2017**(1), 80–105 (2017)
31. Dobraunig, C., Mennink, B.: Leakage resilience of the duplex construction. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Proceedings, Part III. LNCS, vol. 11923, pp. 225–255. Springer (2019)
32. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging merkle-damgård for practical applications. In: Joux, A. (ed.) EUROCRYPT 2009, Proceedings. LNCS, vol. 5479, pp. 371–388. Springer (2009)
33. Dziembowski, S.: Intrusion-resilience via the bounded-storage model. In: Halevi, S., Rabin, T. (eds.) TCC 2006, Proceedings. LNCS, vol. 3876, pp. 207–224. Springer (2006)
34. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008, Proceedings. pp. 293–302. IEEE Computer Society (2008)
35. Fouque, P., Karpman, P., Kirchner, P., Minaud, B.: Efficient and provable white-box primitives. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Proceedings, Part I. LNCS, vol. 10031, pp. 159–188 (2016)
36. Gueron, S., Lindell, Y.: GCM-SIV: full nonce misuse-resistant authenticated encryption at under one cycle per byte. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015, Proceedings. pp. 109–119. ACM (2015)
37. Guo, C., Pereira, O., Peters, T., Standaert, F.: Towards low-energy leakage-resistant authenticated encryption from the duplex sponge construction. IACR Trans. Symmetric Cryptol. **2020**(1), 6–42 (2020)
38. Halevi, S., Rogaway, P.: A tweakable enciphering mode. In: Boneh, D. (ed.) CRYPTO 2003, Proceedings. LNCS, vol. 2729, pp. 482–499. Springer (2003)
39. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Proceedings, Part I. LNCS, vol. 9056, pp. 15–44. Springer (2015)
40. Hosoyamada, A., Isobe, T., Todo, Y., Yasuda, K.: A Modular Approach to the Incompressibility of Block-Cipher-Based AEADs, To appear on IACR Cryptology ePrint Archive
41. intertrust: Intertrust white paper, taking steps to protect financial mobile applications (2018)
42. Ishai, Y., Sahai, A., Wagner, D.A.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003, Proceedings. LNCS, vol. 2729, pp. 463–481. Springer (2003)
43. Kalai, Y.T., Reyzin, L.: A survey of leakage-resilient cryptography. In: Goldreich, O. (ed.) Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali, pp. 727–794. ACM (2019)
44. Kalai, Y.T., Reyzin, L.: A survey of leakage-resilient cryptography. IACR Cryptol. ePrint Arch. p. 302 (2019)
45. Krämer, J., Struck, P.: Leakage-resilient authenticated encryption from leakage-resilient pseudorandom functions. In: Bertoni, G.M., Regazzoni, F. (eds.) COSADE 2020, Revised Selected Papers. LNCS, vol. 12244, pp. 315–337. Springer (2020)
46. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011, Revised Selected Papers. LNCS, vol. 6733, pp. 306–327. Springer (2011)
47. Lepoint, T., Rivain, M., Mulder, Y.D., Roelse, P., Preneel, B.: Two Attacks on a White-Box AES Implementation. In: SAC 2013, Revised Selected Papers. pp. 265–285 (2013)

48. Longo, J., Martin, D.P., Oswald, E., Page, D., Stam, M., Tunstall, M.: Simulatable leakage: Analysis, pitfalls, and new constructions. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Proceedings, Part I. LNCS, vol. 8873, pp. 223–242. Springer (2014)
49. Mandal, A., Patarin, J., Seurin, Y.: On the public indifferentiability and correlation intractability of the 6-round feistel construction. In: Cramer, R. (ed.) TCC 2012, Proceedings. LNCS, vol. 7194, pp. 285–302. Springer (2012)
50. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004, Proceedings. LNCS, vol. 2951, pp. 21–39. Springer (2004)
51. McGrew, D.A., Viega, J.: The security and performance of the galois/counter mode (GCM) of operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004, Proceedings. LNCS, vol. 3348, pp. 343–355. Springer (2004)
52. Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004, Proceedings. LNCS, vol. 2951, pp. 278–296. Springer (2004)
53. Mulder, Y.D., Roelse, P., Preneel, B.: Cryptanalysis of the Xiao - Lai White-Box AES Implementation. In: SAC 2012, Revised Selected Papers. pp. 34–49 (2012)
54. Namprempre, C., Rogaway, P., Shrimpton, T.: Reconsidering generic composition. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014, Proceedings. LNCS, vol. 8441, pp. 257–274. Springer (2014)
55. Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009, Proceedings. LNCS, vol. 5479, pp. 462–482. Springer (2009)
56. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with composition: Limitations of the indifferentiability framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011, Proceedings. LNCS, vol. 6632, pp. 487–506. Springer (2011)
57. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM CCS 2002, Proceedings. pp. 98–107. ACM (2002)
58. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006, Proceedings. LNCS, vol. 4004, pp. 373–390. Springer (2006)
59. Standaert, F., Pereira, O., Yu, Y.: Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Proceedings, Part I. LNCS, vol. 8042, pp. 335–352. Springer (2013)
60. Todo, Y., Isobe, T.: Hybrid code lifting on space-hard block ciphers: Application to Yoroi and SPNbox. IACR Trans. Symmetric Cryptol. **2022**(3), 368–402 (2022)
61. Whiting, D., Housley, R., Ferguson, N.: AES Encryption & Authentication Using CTR Mode & CBC-MAC. IEEE P802.11 Wireless LNAs (2002)
62. Wyseur, B., Michiels, W., Gorissen, P., Preneel, B.: Cryptanalysis of White-Box DES Implementations with Arbitrary External Encodings. In: SAC 2007, Revised Selected Papers. pp. 264–277 (2007)
63. Yoneyama, K., Miyagawa, S., Ohta, K.: Leaky random oracle. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **92-A**(8), 1795–1807 (2009)