IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Cryptanalysis of Shi's White-box Encryption Scheme

HYOUNGSHIN YIM[1], YONGJIN YEOM[1,2], AND JU-SUNG KANG .[1,2]
[1]Department of Financial information security, Kookmin University, Seoul 02707, South Korea
[2]Department of Information Security, Cryptology, and Mathematics Kookmin University, Seoul 02707, South Korea

Corresponding author: Yongjin Yeom (e-mail: salt@kookmin.ac.kr).

This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NO. 2021M1A2A2043893)

ABSTRACT Structural analysis is the study of finding component functions for a given function. In this paper, we proceed with structural analysis of structures consisting of the S (nonlinear Substitution) layer and the A (Affine or linear) layer. Our main interest is the $S^{(2)} \circ A \circ S^{(1)}$ structure with different substitution layers and large input/output sizes. The purpose of our structural analysis is to find the functionally equivalent oracle F^* and its component functions for a given encryption oracle $F = S^{(2)} \circ A \circ S^{(1)}$. As a result, we can construct the decryption oracle F^{*-1} explicitly and break the one-wayness of the building blocks used in a White-box implementation. Our attack consists of two steps: S layer recovery using multiset properties and A layer recovery using differential properties. We present the attack algorithm for each step and estimate the time complexity. Finally, we discuss the applicability of $S^{(2)} \circ A \circ S^{(1)}$ structural analysis in a White-box Cryptography environment.

INDEX TERMS Cryptanalysis, Structural analysis, White-box cryptography, White-box Implementation

I. INTRODUCTION

CRYPTOGRAPHIC technology is widely used in information and communication services for data protection and authentication. In encryption technology, encryption keys are essential for data and information and communication services authentication.

- Intro. to WBC
- Shi's model
- Structural analysis
- Our contribution

The security of the cryptosystem can be guaranteed only when the encryption key is safely protected from various attackers. The attacker models that threaten the security of cryptosystems include black-box attacks, gray-box attacks, and white-box attacks. The black-box attack is carried out through input and output values in unknown assumptions inside the cryptosystem. The gray-box attack is a technique that acquires and attacks side-channel information such as a cryptographic module's power and electromagnetic waves. Among them, the white-box attack assumes the most potent attacker. The white-box attack is a model in which an attacker takes control of the cryptosystem and neutralizes the cryptography. For example, there are dump and change of memory or register, monitoring the execution process, and the like. This has attracted attention to protect encryption keys used for copyright protection from exposure in media players and set-top boxes. Currently, the scope of use is expanding due to the safe execution of financial applications in a mobile environment and the prevention of firmware forgery in embedded devices [1]. In 2002, Chow et al [2, 3]. suggested the possibility of white-box cryptography of AES (Advanced Encryption Standard) and DES (Data Encryption Standard) along with the concept of white-box attackers, and various white-box cryptography technologies were proposed after that. The security of white-box cryptography generally aims at all or part of preventing exposure to encryption keys, one-wayness of encryption or decryption, and preventing the reproduction of cryptosystems [4]. It was analyzed that most of the white-box cryptography designed in a table reference method, including white-box cryptography such as Chow, does not satisfy any security goals. In general, one-wayness, the security of the white-box, cannot be maintained based on the analysis results of the SASAS structure consisting of the non-linear function S-box and the affine function proposed by Biruykov et al. [6, 7]. The analysis results of structures other than the SASAS structure are also the same. However, various attempts are still underway, and white-box cryptographic products that adopt undisclosed techniques are

also actively spreading [5]. This paper proposes an attack method on the light-weight white-box encryption scheme for securing distributed embedded devices presented by Shi et al. to IEEE Transaction on Computers in 2019 [8]. The LW-WBC (Light-Weight White-Box Cryptography) proposed by Shi et al. has a Feistel structure and protects the input and output value of the XOR (exclusive or) operation by using a table reference method in each round. In addition, it was argued that it was safe against the existing white-box attack method. The LW-WBC, a 60-bit $S^{(1)}AS^{(2)}$ structure with different nonlinear S-box sizes, increases attack complexity and enhances security by not exposing the linear functions used. However, as a result of applying Biruykov's SASAS structural analysis, it was confirmed that the inverse functions of each round could be efficiently obtained [8]. This cannot guarantee one-wayness, which is the security of the white-box cryptography. This paper presents the existing white-box cryptographic model and structural analysis studies. First, we implement LW-WBC based on C language and analyze the security evidence claimed by Shi. After that, we present the structural analysis algorithm of the $S^{(1)}AS^{(2)}$ structure and calculate the attack complexity based on it. Finally, the attack is carried out by applying structural analysis to the LW-WBC using Python language. The results of this study can be used in commercialized white-box cryptographic models with $S^{(1)}AS^{(2)}$ structures.

II. WHITE-BOX CRYPTOGRAPHY AND STRUCTURAL ANALYSIS

White-box cryptographic model uses obfuscation techniques by applying encoding to plaintext, ciphertext, and intermediate values. Various white-box cryptographic model design studies are underway, starting with the symmetric key cryptography AES white-box cryptographic model proposed in 2002 [2]. This white-box cryptographic model is very closely related to cryptographic logic, structural analysis. Structural analysis is a study that started with a different motive than white-box cryptography. This section examines the research trend of the white-box cryptographic model. In addition, we look at the structural analysis research trend closely related to the security of white-box cryptography.

A. RESEARCH TRENDS IN WHITE-BOX CRYPTOGRAPHY

Chow et al. proposed AES' white-box cryptographic model in 2002 and presented a design idea that binds fixed encryption keys into tables, including XOR (exclusive-or) operations. Chow's design ideas are the basis for designing the white-box cryptographic model to date. However, it is not safe with a BGE attack [9], and encryption keys can be extracted by analyzing the table reference method regardless of whether encoding and obfuscation are applied. This is enough to obtain an encryption key from a given table in a few seconds in a PC environment. After that, there have been various studies to supplement Chow's white-box design, but most attack methods have been proposed within a few years. Xiao, Lai [10] presented 16-bit, 32-bit linear function encoding to improve the weakness of 4-bit unit nonlinear encoding in the table reference method. Still, a linear equivalence transformation attack method was discovered by Mulder et al. [11]. In addition, in 2020, vulnerabilities were found in the method of obfuscating the round boundary and adding dummy rounds proposed by Xu et al. [12]. As shown in TABLE I, research on white-box cryptography, which has been steadily improved in table reference methods, has continued until recently [13].

TABLE 1. Designs and Attacks in Whitebox cryptography

Whitebox Cryptography	Design	Attack
Whitebox AES	Chow (2002)	Billet (2004)
Whitebox DES	Chow (2002)	(2007)
Perturbated White-box AES	(2006)	(2010)
White-box AES with large linear encoding	(2009)	(2013)

To overcome the limitations of white-box cryptography for standard cryptography, research is also underway to propose white-box cryptography and a suitable cryptographic algorithm. This started in earnest with introducing the space-hard concept by Bogdanov et al. [14] in 2015. WEM (standing for white-box Even-Mansuor) of Chow et al. [26] proposed a new security concept and operation mode of white-box cryptography. Kwon et al. [27] announced FPL (Feistel cipher using Parallel table Look-ups) block ciphers that combine provable security using parallel table reference methods. Along with developing algorithms suitable for this white-box, the security concept was also discussed from various perspectives. Wyseur [4], Saxena [28] in 2009, and Delerablee et al. [29] in 2013 summarized the security concept that white-box cryptography should satisfy, but most of them are difficult to achieve. In 2020, Bock et al. [30, 31] proposed a security concept considering a practical environment and summarized the security of white-box cryptography based on HW-binding and SW-binding. There are various viewpoints on the security concept and goal of white-box cryptography, and commercial products mainly use private white-box cryptography technology that combines solid obfuscation [5, 32, 33]. The white-box cryptography design is also utilized in SM4, a Chinese standard block cipher algorithm. Various designs and analyses of white-box cryptography are in progress in China. Xioa, Lai [18] and Shang [24] and Yao, Chen [25] designed a white-box cryptography model based on SM4 in 2009, 2016, and 2020, but based on a collision attack, the results were announced that it is difficult to maintain security through the analysis method [19]. As a similar research case, an SM4-based light-weight white-box cryptography model suitable for WSNs (Wireless Sensor Networks) environment was proposed by Shi, Yang et al. in 2015. In 2019, a light-weight white-box cryptographic model

TABLE 2. Structural analysis of Substitution-Affine Iterations

Year	Topic	Authors
2001	Structural cryptanalysis of SASAS	A Biryukov et al.
2003	Affine Equivalance Algorithm	A Biryukov et al.
2015	Structural cryptanalysis of ASASA	I Dinur et al.
2015	Analytic Tools for White-box Cryptography	C.H. Baek et al.
2018	An improved Affine Equivalence Algorithm	I Dinur

suitable for distributed resource systems and combining nonlinear and affine functions was proposed [22, 8]. However, a vulnerability in the white-box cryptographic model was discovered in WSNs through collision-based attacks in 2021 [23]. The white-box cryptographic model proposed in 2019 can confirm its applicability to structural analysis attack [7].

B. RESEARCH TRENDS IN STRUCTURAL ANALYSIS

White-box cryptography is closely related to cryptographic logic, structural analysis. In 1997, Paratin et al. [34] attempted to create the function of public-key cryptography by combining S-box, which is the secret key cryptography logic, and higher-order polynomials. However, although it did not yield successful results, it led to a systematic structural analysis study in the future. As shown in TABLE II, security analysis is conducted on functions of various structures in which nonlinear and affine layers (or linear) with multiple S-boxes alternately appear.

Structural analysis is a study of a method of determining each component under conditions in which the structure of a function is known, but the specific function of each component is unknown. In other words, it is a technique of creating an equivalent function having the same function using only the input/output value of a given oracle function. In 2001, Biryukov et al. [6] considered a function of the SASAS structure as an oracle and discovered a way to find an oracle of the same structure with equal functionality. Using this method, you can discover the encryption key hidden inside the SASAS structure. Most of the white-box cryptography using the table reference method can be attacked by this analysis method. The BGE attack [9] can also be interpreted as this analysis method. Baek et al. proposed a toolbox that generalized structural analysis and presented systematic and quantitative attacks on various structures. This structural analysis has expanded its research to various structures such as ASASA and SASASASAS. Typical attacks on white-box cryptography include obtaining encryption keys and attacking one-wayness properties by constructing a decryption algorithm for a given encryption system. White-box cryptography, which combines non-linear and affine functions into tables, is difficult to maintain security through structural analysis. However, various studies are still in progress to design a white-box encryption model based on one-wayness.

III. SHI'S WHITE-BOX ENCRYPTION SCHEME: LW-WBES

In 2019, Shi et al. proposed a white-box encryption scheme for light-weight embedded devices including mobile phones and navigating systems. We denote their scheme by LW-WBES, which means a Light-Weight White-Box Encryption Scheme. LW-WBES has the following features:

- The block size (input/output size) is 120 bits.
- The number of rounds depends on the security level such as 16(default), 10(aggressive), or 32(conservative).
- The encryption process is designed as a variant of Feistel network.
- Two types of keys (black-box key and white-box key) are used for providing black-box security and white-box security simutaneously. Hence, the key size of LW-WBES is extremely large.

A. DESIGN RATIONALE

In order to overcome the difficulties of white-box implementations of standard ciphers, Shi et al. propose a new white-box friendly cipher secure against white-box attack context. Their design stretergies can be summarized as follows:

- The scheme has the secret components based on the Feistel network, which protect the secret white-box keys from white-box attacks including DCA and DFA.
- Since components of three different size (4, 5, 6-bit) are integrated, it is hard to mount the structural analysis directly.
- The secret components can be reused in each round for saving memory usage in light-weight devices.
- The scheme does not require additionaly external encodings nor obfuscation techniques.

We will show that the goal of design rationale cannot be satisfied and weak against structural cryptanalysis in particular.

B. SPECIFICATION

LW-WBES is a 120-bit block cipher and the number of round can be chosen based on the level of security and the constraints of resources. Here, we describe the encryption process of 16-round default version. 120-bit plaintext $PT = (L, R)$ is input for the Feistel network divided into 5-bit variables as:

$$PT = (L, R) = (L_0, L_1, \ldots, L_{11}, R_0, R_1, \ldots, R_{11}),$$

where $L_i, R_i \in GF(2)^5$ for $i = 0, 1, \ldots, 11$. In each round, the round function $F : GF(2)^{60} \times GF(2)^{72} \rightarrow GF(2)^{72}$ consumes 72-bit black-box round key rk by

$$F : (x, k) \mapsto (\Theta_0(x) \oplus rk_0, \ldots, \Theta_{11}(x) \oplus rk_{11}),$$

where Θ_i are nonlinear surjective functions whose outputs are 8-bits and rk is divided into 6-bit components rk_i for $i = 0, 1, \ldots, 11$. In fact, we do not need the details of Θ_i to construct our attack algorithm. Instead of usual mixing

by exclusive-or in Feistel network, LW-WBES uses nonlinear mixing function called T-box which contains white-box key component. The round transformation $(X_L, X_R) \mapsto (Y_L, Y_R)$ can be written as

$$\begin{cases} Y_L = X_R, \\ Y_R = T(X_L, F(X_R, rk)), \end{cases}$$

where T-box $T : GF(2)^{60} \times GF(2)^{72} \to GF(2)^{60}$ consists of 4 sub-components G, F', H^*, and M is defined as

$$T(x, y) = H^*(M(G(x) \oplus F'(y))) \tag{1}$$

- $G : GF(2)^{60} \to GF(2)^{60}$ is composed of 12 bjiections $G_0, G_1, \ldots, G_{11}$ in parallel so that
$$(x_0, x_1, \ldots, x_{11}) \overset{G}{\mapsto} (G_0(x_0), G_1(x_1), \ldots, G_{11}(x_{11})).$$
- $F' : GF(2)^{72} \to GF(2)^{60}$ takes output of round function F and squizes them into 60-bits.
$$(y_0, y_1, \ldots, y_{11}) \overset{F'}{\mapsto} (F'_0(y_0), F'_1(y_1), \ldots, F'_{11}(y_{11})).$$
- $M : GF(2)^{60} \to GF(2)^{60}$ is an invertible linear transformation represented by a binary matrix M whose 5-bit columns are $M_0, M_1, \ldots, M_{11}$.
$$M = (M_0 M_1 \cdots M_{11}),$$
where M_j are 60×5 submatrices for $j = 0, 1, \ldots, 11$.
- $H^* : GF(2)^{60} \to GF(2)^{60}$ is composed of fifteen 4-bit bijections $(h^*_0, h^*_1, \ldots, h^*_{14})$.

The component functions F', G, M, and H^* are white-box keys that cannot be exposed during the white-box implementation of encryption process.

In the white-box encryption algorithm, evaluations of T-box T are possible without knowing its component functions (white-box keys), since T is implemented as several steps of table look-ups. In fact, T-box T without final transformation H^*, say $\tilde{T}(x, y) := M(G(x) \oplus F'(y))$, can be represented by the tables of 12-bit input and 60-bit output as follows:

$$\tilde{T} : GF(2)^{60} \times GF(2)^{72} \to GF(2)^{60},$$

where its first input is $x = (x_0, x_1, \ldots, x_{11})$ and the second is $y = (y_0, y_1, \ldots, y_{11})$. Then we can rewrite $\tilde{T}$ as

$$\tilde{T} : GF(2)^{11} \to GF(2)^{60},$$
$$((x_0, y_0), (x_1, y_1), \ldots, (x_{11}, y_{11})) \overset{\tilde{T}}{\mapsto} (z_0, z_1, \ldots, z_{14}).$$

Define $\tilde{T}_j(x_j, y_j) = M_j(G(x_j) \oplus F'(y_j))$ for $0 \le j \le 11$. Then

$$\begin{aligned} \tilde{T}(x, y) &:= M(G(x) \oplus F'(y)) \\ &= M_0(G_0(x_0) \oplus F'_0(y_0)) \oplus \cdots \\ &\quad \cdots \oplus M_{11}(G_{11}(x_{11}) \oplus F'_{11}(y_{11})) \\ &= \tilde{T}_0(x_0, y_0) \oplus \cdots \oplus \tilde{T}_{11}(x_{11}, y_{11}). \end{aligned}$$

Note that each $\tilde{T}_j(x_j, y_j)$ can be implemented as a precomputed table with 2^{11} entries which takes 15,360 bytes. On the other hand, 60-bit output of $\tilde{T}$ can be divided into fifteen 4-bit subblocks $(z_0, z_1, \ldots, z_{14})$. Apply a 4-bit random nonlinear bijection h_k on each z_k $(k = 0, 1, \ldots, 14)$ in the output of lookup table $\tilde{T}_j(x_j, y_j)$. Then we have to use a cascade structure of masked adders to obtain the final output of T including H^* layer, as depicted in Chow's WB-AES.

The ciphertext of LW-WBES is produced by iterating this process 16 times with their corresponding black-box and white-box keys at each round. Note that there are no output encoding at the end of the final round, since nonlinear bijections appear in the T-box T.

C. WHITE-BOX IMPLEMENTATION AND ITS SECURITY

In order to hide white-box keys in T-box T, LW-WBES uses 12 tables $T_i : GF(2)^5 \times GF(2)^6 \to GF(2)^{60}$ for $i = 0, 1, \ldots, 11$ definded as

$$T_i(x_i, y_i) := H'_i(\tilde{T}_i(x_i, y_i)),$$

where H'_i consists of 4-bit random bijections $(h'_{i,0}, \ldots, h'_{i,14})$ so that each output can be written as

$$T_i(x_i, y_i) = (h'_{i,0}(z_{i,0}), h'_{i,1}(z_{i,1}), \ldots, h'_{i,14}(z_{i,14})).$$

In fact, T-box produces output end with nonlinear bijection H^* as (1):

$$\begin{aligned} T(x, y) &= H^*(M(G(x) \oplus F'(y))) \\ &= H^*(\tilde{T}(x, y)) \\ &= H^*\left(\tilde{T}_0(x_0, y_0) \oplus \cdots \oplus \tilde{T}_{11}(x_{11}, y_{11})\right) \\ &= H^*\Big(H'^{-1}_0 \circ T_0(x_0, y_0) \oplus \cdots \\ &\qquad \cdots \oplus H'^{-1}_{11} \circ T_{11}(x_{11}, y_{11})\Big) \end{aligned} \tag{2}$$

Note that exclusive-or operations can be considered as operations on each 4-bit components. For instance, the first 4-bit output of $T(x, y)_{[0:3]}$ is computed as

$$h^*_0\left(h'^{-1}_{0,0}(T_0(x_0, y_0)_{[0:3]}) \oplus \cdots \oplus h'^{-1}_{11,0}(T_{11}(x_{11}, y_{11})_{[0:3]})\right).$$

When we have two masked variables $w_1 = h_1(z_1)$ and $w_2 = h_2(z_2)$ with random bijection h_1 and h_2, respectively, the masked exclusive-or with bijection h^* can be computed by

$$\begin{aligned} h^*(z_1 \oplus z_2) &= h^*\left(h_1^{-1}(h_1(z_1)) \oplus h_2^{-1}(h_2(z_2))\right) \\ &= h^*\left(h_1^{-1}(w_1) \oplus h_2^{-1}(w_2)\right) \end{aligned} \tag{3}$$

If each variable in (3) represents n-bit data, $2n$-bit to n-bit lookup table $L_{h^*} : (w_1, w_2) \mapsto h^*(z_1 \oplus z_2)$ hide its internal components h_1, h_2, and h^*. This technique called 'Masked Adder' enables us to obtain $T(x, y)$ without revealing H^* by using a cascade of lookup tables.

In the white-box implementation of T-box, the encrytopn process accesses the lookup tables $T_0, \ldots, T_{11}$ and masked adders. Hence, encryptor (as well as white-box adversary) cannot extract the white-box keys such as G, F' and H^*. Furthermore, LW-WBES is designed to provide one-wayness.

That is, it is not feasible to perform decryption process only with the white-box implementation of encyption.

To sum up, LW-WBES is claimed to be secure against black-box attack as well as white-box attack contexts. Its Feistel structure with black-box round keys resists against black-box attack and a large set of white-box keys implemented in T-boxes enhances the security against white-box adversaries. However, we will show later that it is possible to recover the plaintext from the corresponding ciphertext by accessing white-box encryptin orable only. In fact, inverting T-box can be done by structural analysis of the SAS variant explained in the next section.

IV. STRUCTURAL ANALYSIS

For a given black-box function F, the goal of structrural analysis is to reveal its internal components explicitly. informally,

> Given a function F with known internal structure, **structural analysis** is defined as the analysis of F to find an equivalent function F^* by determining its internal components explicitly.

Suppose that we have a bijective function F as

$$f : GF(2)^N \to GF(2)^N.$$

Additionally, we assume following conditions on F:

- Given x, anyone can compute $F(x)$ with ease.
- Positive integer N is large enough so that it is infeasible to invert F. i.e., for a given y, it is not possible to find x satisfying $F(x) = y$ within a reasonable time.
- The number of its internal components and the size of input and output of each component are known to public.
- The structure(how to combine components) of F is open to public but each component itself is not known except for its input and output sizes. Thue, given x, we merely calculate $F(x)$ without knowing intermediate values.

Last two decades, the structrural analysis has been studied for functions with layered structure. In 2002, Biryukov et al. proved that a function of SASAS structure can be analyzed successfully so that one can construct an equivalent function explicitly, where S layer is composed of small nonlinear S-boxes in parallel and A layer is an affine or a linear transformation. Later, several types of structural analysis such as ASASA have been considered.

For a given F, our goal for structural analysis is to find an equivalent function F^* explicitly.

A. MULTISET PROPERTIES

We define a multiset by a set

B. STRUCTURAL ANALYSIS OF $S^{(1)}AS^{(2)}$

Multiset properties play a key role in the structural analysis. For example, suppose that N-bit function F has SASAS structure and each S-layer consists of m-bit S-boxes. Then if we use input multiset of the form $(P, C, C, \ldots, C)$, output of F has B(balanced) property [Biryukov].

We focus on the $S^{(1)}AS^{(2)}$ structure for T-box function. LW-WBES uses a variant of SAS structure for T-box which consists of two S layers with different size of S-boxes and a linear function between them. In this section, we focus on bijective functions with $S^{(1)}AS^{(2)}$ structure as follows:

- $S^{(1)} : GF(2)^{m_1 \cdot k_1} \to GF(2)^{m_1 \cdot k_1}$

$$S^{(1)}(x_0, \ldots, x_{k_1-1}) = s_0^{(1)}(x_0) \| \cdots \| s_{k_1-1}^{(1)}(x_{k_1-1}).$$

- $A : GF(2)^{m_1 \cdot k_1} \to GF(2)^{m_1 \cdot k_1}$ is a linear transformation on the vector space over $G(2)$ which can be represented by an $m_1 k_1 \times m_1 k_1$ matrix.
- $S^{(2)} : GF(2)^{m_2 \cdot k_2} \to GF(2)^{m_2 \cdot k_2}$

$$S^{(2)}(x_0, \ldots, x_{k_2-1}) = s_0^{(2)}(x_0) \| \cdots \| s_{k_2-1}^{(2)}(x_{k_2-1}).$$

Since the function is a bijection, we observe that

$$N := m_1 \cdot k_1 = m_2 \cdot k_2.$$

LW-WBES chooses $m_1 = 5$, $k_1 = 12$ and $m_2 = 4$, $k_2 = 15$ so that $N = 5 \times 12 = 4 \times 15 = 60$ bits.

We can remove $S^{(2)}$-layer efficiently by Theorem 4.1.

Theorem 4.1: For a given function $F := S^{(2)} \circ A \circ S^{(1)}$, we can find a function $\tilde{S}^{(2)}$ such that

$$(\tilde{S}^{(2)})^{-1} \circ F = \tilde{A} \circ S^{(1)}.$$

proof. Choose 2^{m_1} input data

$$X_i := (x_{0,i}, x_{1,i}, \ldots, x_{k_1-1,i}), \quad i = 0, 1, \ldots, 2^{m_1} - 1$$

that form a multiset with property $(D, D, \ldots, D)$. Then $S^{(1)}$-layer preserves the multiset property and the input of $S^{(2)}$-layer has balanced property $(B, B, \ldots, B)$. The output of F can be written as

$$Y_i := (y_{0,i}, y_{1,i}, \ldots, y_{k_2-1,i}), \quad i = 0, 1, \ldots, 2^{m_1} - 1.$$

It follows from the above balanced property that

$$\bigoplus_{i=0}^{2^{m_1}-1} (s_k^{(2)})^{-1}(y_{k,i}) = 0, \quad k = 0, 1, \ldots, k_2 - 1. \tag{4}$$

In order to determine the component $s_k^{(2)}$, we introduce new variables $z_{k,0}, z_{k,1}, \ldots, z_{k,2^{m_2}-1}$ such that

$$z_{k,j} = (s_k^{(2)})^{-1}(j), \quad j = 0, 1, \ldots, 2^{m_2} - 1.$$

For a fixed k, the equation (4) can be interpreted as the equation for unknowns $z_{k,0}, z_{k,1}, \ldots, z_{k,2^{m_2}-1}$. Choose another input multiset and repeating this process until we obtain "*sufficiently many*" linearly independent equations. Then we can pick a solution of the system of linear equations, which means that we determine the internal component $s_k^{(2)}$ of F. Note that the solution is not unique. If we find a solution $s_k^{(2)}$, then $s_k^{(2)} \circ a_k$ is also a solution, where a_k is an affine map.

On the other hand, suppose that F is designed as

$$F = S^{(2)} \circ A \circ S^{(1)}$$

and we also know all functions used as internal components.

Then it is easy to find a class of equivalent functions

$$F^* = \tilde{S}^{(2)} \circ \tilde{A} \circ S^{(1)}, \tag{5}$$

by inserting an affine layer a^* and its inverse between $S^{(2)}$ and A, where

$$a^* = (a_0^*, a_1^*, \ldots, a_{k_2-1}^*),$$

and a_k^* $(k = 0, 1, \ldots, k_2 - 1)$ is an m_2-bit affine bijection. If we choose $\tilde{S}^{(2)} := S^{(2)} \circ (a^*)^{-1}$ and $\tilde{A} := a^* \circ A$, then $F^* = F$ as a black-box function. Thus each $s_k^{(2)}$ has $\mathcal{N}(m_2)$ equivalent forms, where $\mathcal{N}(m_2)$ is the number of m_2-bit affine bijections.

Reconsider the meaning of "*sufficiently many*" mentioned above in the proof. If we collect $2^{m_2} - m_2 - 1$ linearly independent equations for $z_{k,j}$'s, then a system of linear equations of the form (4) has $m_2 + 1$ dimensional kernel by the rank-nullity theorem [Strang]. Then there are $m_2 + 1$ free variables among $z_{k,j}$'s. We have $\mathcal{N}(m_2)$ possible cases since we have to choose them so that $s_k^{(2)}$ is invertible.

To sum up, if we have $2^{m_2} - m_2 - 1$ linearly independent equations for for $z_{k,j}$'s and choose a solution to define $\tilde{s}_k^{(2)}$, then there exists an affine bijection a_k^* such that

$$\tilde{s}_k^{(2)} = s_k^{(2)} \circ a_k^*.$$

Set

$$\tilde{S}^{(2)} := \tilde{s}_0^{(2)} \| \cdots \| \tilde{s}_{k_2-1}^{(2)},$$
$$\tilde{A} := \left((a_0^*)^{-1} \| \cdots \| (a_{k_2-1}^*)^{-1}\right) \circ A.$$

Then $F = \tilde{S}^{(2)} \circ \tilde{A} \circ S^{(1)}$. This completes the proof. □

Note that we can construct input multisets $(D, D, \ldots, D)$ easily by adding two layers at the begining of F so that the resulting function has $SASAS$ structure. Applying input as $(P, C, C, \ldots, C)$, we expect the same result. Thus we can make as many input multisets as we wish by choosing constant parts arbitrarily. Thus it is easy to collect $2^{m_2} - m_2 - 1$ lineary independent equations. Algorithm 1 provides a way to make $S^{(2)} \circ A \circ S^{(1)}$ to $\tilde{A} \circ S^{(1)}$.

Algorithm 1 Recovering $S^{(2)}$-layer

Input: $F = S^{(2)} \circ A \circ S^{(1)}$ as a black-box function
Output: $\tilde{S}^{(2)}$ such that $(\tilde{S}^{(2)})^{-1} \circ F = \tilde{A} \circ S^{(1)}$

for $k = 0$ to $k_2 - 1$ **do**
  Assign variables $z_j := (s_k^{(2)})^{-1}(j)$ for $0 \le j < k_2$.
  $A \leftarrow \phi$ ▷ A: Set of equations
  while $n < 2^{m_2} - m_2 - 1$ **do**
    Choose a multiset with property $(D, \ldots, D)$ as $\{X_i := (x_{0,i}, \ldots, x_{k_1-1,i}) : i = 0, 1, \ldots, 2^{m_1} - 1\}$.
    Store the corresponding output as $Y_i = F(X_i)$ $\{Y_i := (y_{0,i}, \ldots, y_{k_1-1,i}) : i = 0, 1, \ldots, 2^{m_1} - 1\}$.
    Construct an equation (eq) as

$$(eq) : \bigoplus_{i=0}^{2^{m_1}-1} (s_k^{(2)})^{-1}(y_{k,i}) = 0$$

    if (eq) is linearly independent of A **then**
      Add the equation (eq) to A
    end if
  end while
  Solve the system of linear equations in A:
    Determine z_j by Gaussian elimination.
  Define $\tilde{s}_k^{(2)}(z_j) = j$.
end for
return $\tilde{S}^{(2)} \leftarrow \tilde{s}_0^{(2)} \| \cdots \| \tilde{s}_{k_2-1}^{(2)}$.

From Algorithm 1, we can successfully remove the $S^{(2)}$-layer and obtain $\tilde{F}$ with SA-structure. Considering the differential charateristic of $\tilde{F}$, we can remove its linear part, too. In the following two lemmas, we recall elementary concepts of linear algebra such as rank of matrices and difference preserving property of linear transformation.

Lemma 4.1: Let $L_A : GF(2)^m \to GF(2)^n$ and $L_B : GF(2)^n \to GF(2)^n$ be linear transformations whose corresponding matrices are A and B, respetively. Then we have the associated property of the rank

$$\mathrm{rank}(L_B \circ L_A) \le \mathrm{rank}(L_A).$$

Lemma 4.2: Let $L : GF(2)^n \to GF(2)^n$ be a linear transormation. Then L preserve the difference as $\Delta L(x) = L(\Delta x)$. More precisely, for input data x_1, x_2 and their difference $\Delta x := x_1 \oplus x_2$,

$$\Delta L(x) = L(x_1) \oplus L(x_2) = L(x_1 \oplus x_2) = L(\Delta x).$$

Theorem 4.2: For a given $\tilde{F} = \tilde{A} \circ S^{(1)}$, we can find a function $\tilde{A}^*$ such that

$$(\tilde{A}^*)^{-1} \circ \tilde{F} = \tilde{S}^{(1)}.$$

proof. Let U_{in} be a set of all possible differences for $GF(2)^{m_1}$. Initialize two sets Δ_{in} and Δ_{out} as empty, which will store input and output differences. Randomly select d_{in} from U_{in} and construct a pair of input for $\tilde{F}$ with its difference $D_{in} := (d_{in}, 0, \ldots, 0)$. Then we have the corresponding output difference $\Delta Z := \tilde{F}(D_{in})$ as

$$D_{out} := \tilde{F}(x_0, 0, \ldots, 0) \oplus \tilde{F}(x_1, 0, \ldots, 0).$$

Update two setes as

$$\Delta_{in} \leftarrow \Delta_{in} \cup \{D_{in}\}, \quad \Delta_{out} \leftarrow \Delta_{out} \cup \{D_{out}\}.$$

Repeat this process from selecting new d_{in} and update Δ_{in} and Δ_{out} only if D_{out} is linearly independent of differences which are already stored in Δ_{out}. We can check this by comparing the ranks of the matrices generated by Δ_{out} and

$\Delta_{out} \cup \{D_{out}\}$. Suppose that we have ℓ output differences in Δ_{out} as

$$\Delta_{out} = \{w_0, w_1, \ldots, w_{\ell-1}\},$$

where w_i are interpreted as column vectors for $i = 0, 1, \ldots, \ell - 1$ and we have a new difference $w_\ell := \tilde{F}(D_{in})$ to be added in Δ_{out}. If the rank increases,

$$\text{rank}\,[w_0 \cdots w_{\ell-1} w_\ell] = \text{rank}\,[w_0 \cdots w_{\ell-1}] + 1,$$

then update $\Delta_{out} \leftarrow \Delta_{out} \cup \{w_\ell\}$.

Iterating this process, we can reach the maximal rank m_1 and obtain m_1 output differences in $\Delta_{out} = \{w_0, w_1, \ldots, w_{m_1-1}\}$.

Note that by Lemma, 4.1 the maximal rank is m_1, since we make input differences in m_1-bit S-box $s_0^{(1)}$.

Consider two inputs $(x_1, 0, \ldots, 0)$ and $(x_2, 0, \ldots, 0)$ with $d_{in} := x_1 \oplus x_2$. Since $\tilde{F}$ is a black-box function, we do not know the intermediate difference

$$\delta y := S^{(1)}(D_{in}) = S^{(1)}(x_1, 0, \ldots, 0) \oplus S^{(1)}(x_2, 0, \ldots, 0).$$

However, we have the final difference D_{out} interpreted as

$$D_{out} = \tilde{F}(D_{in}) = A(\delta y).$$

With $\Delta_{out} = \{w_0, w_1, \ldots, w_{m_1-1}\}$, let $\delta y_0, \delta y_1, \ldots, \delta y_{m_1-1}$ be the corresponding intermediate differences. Then the linear transformation $\tilde{A}$ maps

$$(\delta y_j, 0, \ldots, 0) \mapsto w_j, \quad \text{for } j = 0, 1, \ldots, m_1 - 1.$$

There is a linear map $L_0 : GF(2)^{m_1} \to GF(2)^{m_1}$ such that

$$\delta y_j \mapsto \hat{e}_j,$$

where $\hat{e}_j = (e_{j0}, e_{j1}, \ldots, e_{jm_1-1})$ defind as $e_{ji} = 1$ for $i = j$ and $e_{ji} = 0$, otherwise.

Define a linear map $A_0^* : GF(2)^{m_1} \to GF(2)^n$ by

$$A_0^*(x) := \tilde{A}(L_0(x), 0, \ldots, 0).$$

Its matrix representation is $W_0 := [w_0 \cdots w_{m_1-1}]$.

In a similar way, we successively define linear maps A_j^* and matrices W_j for $j = 1, \ldots, k_1 - 1$. Combining these matrices, we can define $\tilde{A}^* : GF(2)^n \to GF(2)^n$ as

$$\tilde{A}^*(x_0, \ldots, x_{k_1-1}) := \tilde{A}(L_0(x_0), \ldots, L_{k_1-1}(x_{k_1-1}))$$

represented by $n \times n$ matrix $[W_0 \cdots W_{k_1-1}]$.

It is easy to check $(\tilde{A}^*)^{-1} \circ \tilde{F}$ is divided into k_1 S-boxes in parallel. In fact, since $\tilde{A}$ is linear, we may write

$$\begin{aligned}(\tilde{A}^*)^{-1} \circ \tilde{F} &= (L_0^{-1} \| \cdots \| L_{k_1-1}^{-1}) \circ \tilde{A}^{-1} \circ \tilde{A} \circ S^{(1)} \\ &= (L_0^{-1} \| \cdots \| L_{k_1-1}^{-1}) \circ S^{(1)} \\ &= (L_0^{-1} \circ s_0^{(1)} \| \cdots \| L_{k_1-1}^{-1} \circ s_{k_1-1}^{(1)}) \\ &= (\tilde{s}_0^{(1)} \| \cdots \| \tilde{s}_{k_1-1}^{(1)}).\end{aligned}$$

Without knowing the functions L_j's in the middle, we can remove the linear part $\tilde{A}$. □.

Applying Theorem 4.1 and 4.2, we can find all internal components of the black-box function F accodring to the steps in Algorithm 2. Note that all components of the function F cannot be determined uniquely. So far, we have shown that SAS-structure can be successfully analyzed even though S-boxes of different size are used in S-layers.

Algorithm 2 Recovering A-layer

Input: $\tilde{F} := \tilde{A} \circ S^{(1)}$ as a black-box function
Output: $\tilde{A}^*$ such that $(\tilde{A}^*)^{-1} \circ \tilde{F} = \tilde{S}^{(1)}$

1: Let U_{in} be a set of all possible input differences for an m_1-bit function.
2: **for** $k = 0$ to $k_1 - 1$ **do**
3: $\quad \Delta_{in} \leftarrow \phi$
4: $\quad \Delta_{out} \leftarrow \phi$
5: $\quad$ **repeat**
6: $\quad\quad$ Select $d_{in} \leftarrow U_{in}$ randomly.
7: $\quad\quad$ $D_{in} \leftarrow (0, \ldots, d_{in}, \ldots, 0)$
8: $\quad\quad$ **if** $\text{rank}\,\tilde{F}(\Delta_{in}) < \text{rank}\,\tilde{F}(\Delta_{in} \cup \{D_{in}\})$ **then**
9: $\quad\quad\quad$ $\Delta_{in} \leftarrow \Delta_{in} \cup \{D_{in}\}$
10: $\quad\quad\quad$ $\Delta_{out} \leftarrow \tilde{F}(\Delta_{in})$
11: $\quad\quad$ **end if**
12: $\quad$ **until** $\text{rank}(\Delta_{out}) = m_1$
13: $\quad$ From the set $\Delta_{out} = \{w_0, w_1, \ldots, w_{m_1-1}\}$, define

$$W_k := [w_0 \cdots w_{m_1-1}].$$

14: **end for**
15: Define a linear map $\tilde{A}^*$ using $n \times n$ matrix W:

$$W := [W_0 \cdots W_{k_1-1}].$$

16: Inverting the matrix W, define $\tilde{S}^{(1)}$ as

$$\tilde{S}^{(1)} := (\tilde{A}^*)^{-1} \circ \tilde{F} = (\tilde{s}_0^{(1)} \| \cdots \| \tilde{s}_{k_1-1}^{(1)}).$$

17: **return** $\tilde{S}^{(1)}$.

V. CRYPTANALYSIS OF SHI'S ALGORITHM

From the result of analysis in Section 4, we construct an attack algorithm for Shi's LW-WBES. The type of our attack is "plaintext recovery attack":

> For a given implementation of LW-WBES and ciphertexts, we can find its plaintexts by inverting each round successively.

- round inversion
- Attack algorithm of plaintext recovery attack
- Experimental results
- Possible countermeasures

VI. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate

on the importance of the work or suggest applications and extensions.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in American English is without an "e" after the "g." Use the singular heading even if you have many acknowledgments. Avoid expressions such as "One of us (S.B.A.) would like to thank . . . ." Instead, write "F. A. Author thanks . . . ." In most cases, sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.

REFERENCES

[1] S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot, "White-box cryptography and an AES implementation", *SAC 2002*, LNCS volume 2595, 2003.
[2] S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot, "A white-box DES implementation for DRM applications", *Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002*, LNCS volume 2696, 2003.
[3] B. Wyseur, "White-Box Cryptography", PhD thesis, Katholieke University Leuven, 2009.
[4] A. Biryukov, A. Shamir, "Structural cryptanalysis of SASAS", *Eurocrypt 2001, J. of Cryptology*, 23(4), 2010.
[5] H. Yim, J.-S. Kang, Y. Yeom, "An Efficient Structural Analysis of SAS and its Application to White-Box Cryptography", *IEEE TENSYMP*, 2021.
[6] Y. Shi, W. Wei, H. Fan, M. H. Au and X. Luo, "A Light-Weight White-Box Encryption Scheme for Securing Distributed Embedded Devices", *IEEE Transactions on Computers*, vol. 68, no. 10, 2019.
[7] O. Billet, H. Gilbert, C. Ech-Chatbi, "Cryptanalysis of a white box AES implementation", *SAC 2004*, LNCS volume 3357, 2004.
[8] Y. Xiao, X. Lai, "A secure implementation of white-box AES", *2nd International Conference on Computer Science and its Applications, IEEE CSA*, 2009.
[9] Y. De Mulder, P. Roelse, B. Preneel, "Cryptanalysis of the Xiao–Lai white-box AES implementation", *SAC 2013*, LNCS volume 7707, 2013.
[10] T. Xu, C. K. Wu, F. Liu, R. Zhao, "Protecting white-box cryptographic implementations with obfuscated round boundaries", *Sci. China Inform. Sci.*, 61(3), 2018.
[11] Y. Yeom, DC. Kim, C. H. Baek, J. Shin, "Cryptanalysis of the Obfuscated Round Boundary Technique for Whitebox Cryptography", *Sci. China Inform. Sci.*, 63, 2020.
[12] A. Bogdanov and T. Isobe, "White-box cryptography revisited: Space-hard ciphers", *ACM SIGSAG Conference on Computer and Communications Security*, ACM, 2015.
[13] J. Cho, Y. Choi, I. Dinur, O. Dunkelman, N. Keller, D. Moon, A. Veidberg, "WEM: A New Family of White-Box Black Ciphers Based on the Even-Mansour Construction", *CT-RSA 2017*, LNCS volume 10159, 2017.
[14] C. H. Baek, J. H. Cheon, H. Hong, "White-box AES implementation revisited", *Journal of Communications and Networks*, 2016.
[15] A. Biryukov, C. De Canniere, A. Braeken, B. Preneel, "A toolbox for cryptanalysis: Linear and affine equivalence algorithms", *EUROCRYPT 2003*. LNCS volume 2656, 2003.
[16] A. Biryukov, C. Bouillaguet, D. Khovratovich, "Cryptographic schemes based on the ASASA structure: Black-box, white-box, and public-key", *ASIACRYPT 2014*, LNCS volume 8873, 2014.
[17] I. Dinur, "An Improved Affine Equivalence Algorithm for Random Permutations", *EUROCRYPT 2018*. LNCS volume 10820, 2018.
[18] I. Dinur, O Dunkelman, T Karnz, G Leander, "Decomposing the ASASA Block Cipher Construction", IACR Cryptol, 2015.
[19] A. Biryukov, D. Khovratovich, "Decomposition attack on SASASASAS," *Cryptology ePrint Archive*, Report 2015/646, 2015.