Lin TT, Lai XJ, Xue WJ *et al.* A new Feistel-type white-box encryption scheme. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(2): 386–395 Mar. 2017. DOI 10.1007/s11390-017-1727-x

A New Feistel-Type White-Box Encryption Scheme

Ting-Ting Lin [1,2], Xue-Jia Lai [1,*], Wei-Jia Xue [1], and Yin Jia [1]

[1] *Cryptography and Information Security Laboratory, Department of Computer Science, Shanghai Jiao Tong University Shanghai 200240, China*

[2] *Irdeto Canada, Ottawa, Ontario, K2K3G5, Canada*

E-mail: lintingting00@163.com; lai-xj@cs.sjtu.edu.cn; icelikejia@sjtu.edu.cn; jy09091001@163.com

Received December 1, 2015; revised December 1, 2016.

Abstract The white-box attack is a new attack context in which it is assumed that cryptographic software is implemented on an un-trusted platform and all the implementation details are controlled by the attackers. So far, almost all white-box solutions have been broken. In this study, we propose a white-box encryption scheme that is not a variant of obfuscating existing ciphers but a completely new solution. The new scheme is based on the unbalanced Feistel network as well as the ASASASA (where "A" means affine, and "S" means substitution) structure. It has an optional input block size and is suitable for saving space compared with other solutions because the space requirement grows slowly (linearly) with the growth of block size. Moreover, our scheme not only has huge white-box diversity and white-box ambiguity but also has a particular construction to bypass public white-box cryptanalysis techniques, including attacks aimed at white-box variants of existing ciphers and attacks specific to the ASASASA structure. More precisely, we present a definition of white-box security with regard to equivalent key, and prove that our scheme satisfies such security requirement.

Keywords white-box, equivalent key, Feistel network, cryptography, ASASASA

1 Introduction

The foundations of traditional cryptographic model originate from Shannon's model of communication system[1]. This model assumes that communication end-points and the operating environment of cryptographic primitives are trusted, and the attacker can only create disturbances in the communication channel, such as observation, manipulation, and replay. We denote this cryptographic model as the black-box model because it seems that the execution details of the cryptographic primitives are encapsulated in a black box.

Over the past decade, the field of cryptanalysis has experienced major changes, for example, timing attacks[2], simple and differential power analysis[3], electromagnetic emanation analysis[4], and some direct white-box attacks like HeyRays, OllyDbg, and HIEW. Attacks focused on the implementation of cryptographic primitives in devices were introduced.

In other words, one can no longer assume that an operating environment is trusted. An adversary may be able to observe and tamper with the implementation to extract information about the cryptographic key. Therefore the cryptographic algorithms that are secure in the black-box model will be vulnerable to such attacks.

As a result, many applications that make use of these algorithms as part of their security solution will be dangerous. For example, in a pay TV scenario, the content provider sends encrypted data to a user. The decryption process will be executed at an un-trusted end point. Then the owner of this end point may extract the key and benefit from illegally distributing the key for decrypting the TV to other users. There are still many other scenarios, like cloud computing[5-6], sensor networks[7-8] and malicious hosts[9].

Therefore, it is an extremely urgent requirement to design a relatively secure cryptographic scheme in

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61272440, 61472251, and U1536101, and China Postdoctoral Science Foundation under Grant Nos. 2013M531174 and 2014T70417.

*Corresponding Author

such extreme attack contexts. These contexts were first termed by Chow *et al.* as white-box (WB) attack context[10]: the attacker is assumed to have all the advantages of the hosts and have total access to the execution processes of the cryptographic algorithms.

Almost all recent research on white-box cryptography has focused on fixed-key symmetric algorithms. In 2002, Chow *et al.* proposed a white-box variant of Data Encryption Standard (DES) in [10] and a white-box variant of Advanced Encryption Standard (AES) in [11]. The basic idea is to break the cipher (AES or DES) into a number of steps, insert some randomly chosen mix bijections as obfuscation into every step, and use networked encodings (see Definition 3 in [11]) to cancel out the inserted parts in order to make the implementation functionally equivalent to the original cipher. The white-box variant is implemented with a set of key-dependent look-up tables.

However, for the white-box DES variant[10], in 2002, Jacob *et al.*[12] pointed out that the secret key can be recovered by injecting faults into the running program. In 2005, Link and Neumann[13] improved this variant and claimed that their improvement was secure against known attacks. In 2007, Wyseur *et al.*[14] proposed a differential attack on the schemes of Chow *et al.*[10] and Link and Neumann[13]. In the same year, Goubin *et al.*[15] presented a general cryptanalysis with a method similar to [14].

For Chow *et al.*'s white-box AES variant[11], in 2004, Billet *et al.*[16] presented an efficient attack to recover the secret key (we call it BGE attack) with a time complexity less than 2^{30}. In [17], Michiels *et al.* presented an algorithm to extract the secret key from the white-box variant of any substitution linear-transformation (SLT) cipher. In 2013, Lepoint *et al.*[18] proposed a more efficient attack that can be used to recover the secret key with a time complexity of 2^{22}.

In 2009, Xiao and Lai[19] improved Chow *et al.*'s white-box variant of AES to resist the BGE attack. However, in 2012, De Mulder *et al.*[20] attacked it with Biryukov *et al.*'s linear and affine equivalent algorithm[21].

In 2011, Karroumi[22] proposed an improved white-box variant of AES with a dual cipher and claimed that the complexity of BGE attack against this scheme is 2^{91}. Unfortunately, this scheme was also broken by Lepoint *et al.* in [18].

Another attempt to study white-box cryptography was based on the method of perturbations[23] by Bringer *et al.* The idea is to add specific terms as perturbations to the original cryptographic equations in order to dissimulate the algebraic structure, and these perturbations are canceled in the last round. However, this implementation has been shown to be insecure by De Mulder *et al.*[24]

In addition to AES and DES, there has been an attempt at a white-box variant of SMS4[25] that uses a method similar to Chow *et al.*'s white-box variant of AES[11], but it was also broken by Lin and Lai in [26].

Recently, Biryukov *et al.*[27] presented a white-box cryptographic solution based on the ASASA structure (where "A" means affine, and "S" means substitution). In the solution, an ASASA cipher was encapsulated in a table and used as a cell in a design to process a larger block. It was not a variant of any existing block cipher, but a new scheme. Unfortunately, this construction was attacked by Minaud *et al.*[28] and Dinur *et al.*[29] In particular, Biryukov and Khovratovich showed in [30] that structures like ASASASAS and SASASASAS can be attacked by decomposition algorithms if the block length l and the S-box size s satisfy the condition $s^2 \leqslant l$.

Some dedicated white-box schemes named as space-hard ciphers were proposed by Bogdanov and Isobe[31] in 2015. In the FIXED-SPACE schemes, they used a generalized Feistel network which has l branches and each branch corresponds to an n_a-bit block ($n = n_a \times l$ is the size of the plaintext). They also used a block cipher to implement the round function. The round function takes the block of the leftmost branch as its input, and obtains the output by XORing a constant r and a value determined by the most significant n_b ($= n - n_a$) bits of the outputs of the block cipher (the only look-up table in the schemes). Then the n_b-bit output of the round function and the concatenation of the rightmost $l-1$ block are combined into an n_b-bit result with an XOR. Finally, an n-bit value computed by concatenating the n_b-bit result with the leftmost block becomes the plaintext of next round.

The N-SPACE schemes use a similar way to encrypt n-bit plaintext. The distinction is that there are N different round functions used in N rounds respectively and recycled in every N rounds.

One of the white-box security of the SPACE schemes is key-extraction security, which can be reduced to the difficulty of key recovery for the underlying block ciphers (such as AES-128) in the black-box context. The other white-box security is space hardness, which is used to evaluate the difficulty of compressing the white-box implementation of the schemes, and quantify the security with the amount of code that

could be isolated from the implementation and still maintain the functionality. The SPACE schemes are proved to be secure based on the two security requirements, and there have been no efficient attacks as far as we know.

Due to its large size and the difficulty to find a compact implementation, the SPACE schemes keep safe in such situations where the attackers' available storages or capacities of the communication channels are limited. However, the efficiency of implementing a scheme with a huge size tends to be low, which makes the SPACE schemes hard to apply to "small" devices that require smaller code size, such as smart phone.

Our Contribution. On the basis of the fact that almost all white-box variants of existing block ciphers and the white-box ASASA cipher have been broken, and the large-sized SPACE schemes have limitation on their usage, in this paper, with the unbalanced Feistel network and ASASASA structure, we present a Feistel-type white-box encryption scheme that is not a variant of any existing encryption scheme, but an entirely new white-box solution. The solution can be used to protect secret information in an untrusted platform, such as pay TV scenario, cloud computing, and sensor networks. It has following strong points.

1) The total block length of our scheme is variable, such as 128, 256, and 512; hence the scheme can process a longer data series at one time.

2) Moreover, regardless of the total block length, the block length and the S-box size of the ASASASA structure are fixed to 16. This setting not only fails in satisfying the attack condition $s^2 \leqslant l$, thus thwarting the attack proposed in [30], but also makes look-up tables stay at a reasonable size.

3) Except for attacks in [30], no specific attack against the ASASASA structure has been found. Also, our scheme is a fresh scheme, and thus it is advantageous in bypassing attacks aimed at white-box variants of existing ciphers.

4) By accurate calculation, we show that our scheme has huge white-box diversity and white-box ambiguity, which are two basic requirements specific to white-box schemes.

5) Because of the unbalanced Feistel network, the space requirements of our solutions grow extremely slowly with the growth of the total block length. All the space requirements do not exceed 10 MB for blocks of 128, 256, and 512 bits.

We also present a definition of "white-box security with regard to equivalent key" by modifying the weak white-box security notion presented in [27]. The definition indicates that the ultimate purpose of a white-box adversary is to use the cryptographic functionality like the owner of the secret key and some unexpected information may act as the secret key and play an important role in recovering the cryptographic functionality. On the basis of this definition, we use two claims to prove that our scheme satisfies the security definition.

In general, our scheme performs well on the balance of efficiency and security. It is suitable for security requirements and efficient with regard to space requirements compared with other solutions on an equal security level. This scheme is expected to be a good candidate for white-box schemes.

2 Review of Feistel Network and ASASASA Structure

2.1 Feistel Network

The Feistel network was proposed by Horst Feistel in his design of Lucifer[32]. It is a general method for designing block ciphers and has been widely used in many well-known block cipher designs, such as DES[33], RC6[34], and Twofish[35]. Generally speaking, one round of a Feistel network is defined as follows.

Definition 1. *Let n be a block length, k_i be a k-bit key, x_i be an input to the round, x_{i+1} be an output to the round, $l(x_i)$ be the left $n/2$ bits of x_i, $r(x_i)$ be the right $n/2$ bits of x_i, and $F : \{0,1\}^{n/2} \times \{0,1\}^k \to \{0,1\}^{n/2}$ be an F function. The algorithm of one round of a balanced Feistel network is*:

$$x_{i+1} = r(x_i)||(F_{k_i}(l(x_i)) \oplus r(x_i)).$$

An unbalanced Feistel network is a Feistel network where the sizes of $l(x_i)$ and $r(x_i)$ are not equal.

2.2 ASASASA Structure

The affine-substitution-affine (ASA) structure with secret affine transformation A and nonlinear transformation S is a fundamental method for designing ciphers. The ASASA structure was first discussed by Patarin and Goubin in [36] to construct asymmetric cryptosystems, but this suggestion was attacked by Biham in [37] because the S-box is not bijective owing to design constraints. Subsequently, the SASAS structure was also broken by Biryukov and Shamir in [38-39].

In 2014, Biryukov *et al.* suggested several schemes based on the ASASA structure as well as the ASASASA

structure[27]. Their ASASA constructions were attacked by Minaud *et al.* in [28] and Dinur *et al.* in [29]. Structures with more layers like ASASASAS and SASASASAS were proven to be insecure by Biryukov and Khovratovich in [30] if the block length l and the S-box size s satisfy the condition $s^2 \leqslant l$.

3 New Definition of White-Box Security

Generally speaking, in the traditional cryptographic context, the main goal of an adversary is to extract a secret key in a cryptographic scheme and use the key to accomplish cryptographic functionality (e.g., encrypt, sign) like the owner of the key. In the white-box context, an adversary has the similar goal (using a key to accomplish cryptographic functionality), while the difference is that the conception of the "key" needs to be extended because white-box attacks are much stronger than traditional cryptographic attacks and some unexpected information may be obtained and utilized to launch an attack. Such unexpected information can be viewed as a replacement of the secret key, and we call it the equivalent key.

Thus, we provide a definition of white-box security w.r.t. equivalent key by modifying the weak white-box security notion presented in [27]. In our definition, a white-box scheme is not specific to a white-box variant of an encryption scheme, but refers to a general white-box encryption scheme. Furthermore, we do not use a parameter T to depict the length of an equivalent key as well as the white-box security, because any information can probably be the equivalent key and it is difficult to establish a measurement criterion for the length.

Definition 2 (White-Box Security w.r.t. Equivalent Key). *Let (E, D) be an encryption scheme and $S(K)$ be the equivalent key set for E if for any $\tilde{K} \in S(K)$, there exists an algorithm $F : \tilde{K} \to \tilde{E}$ that generates a scheme $\tilde{E}$ functionally equivalent to E_k. An encryption scheme (E, D) is white-box secure w.r.t. equivalent key $\tilde{K}$ if for all probabilistic polynomial-time adversaries A that are given full access to E_k, it is computationally difficult to obtain $\tilde{K} \in S(K)$.*

4 New Feistel-Type White-Box Encryption Scheme

In this section, we propose a new white-box encryption scheme based on three operations of affine transformations, XORs, and look-up tables, where the look-up table consists of seven layers of key-dependent ASASASA structure. The decryption procedure is similar to the encryption procedure, except that the affine transformations and look-up tables are reversed.

4.1 Framework of the Scheme

We use an unbalanced Feistel structure for constructing the scheme with a total block length of $4n$ bits and a look-up table size of 16 bits. One round of our scheme is defined as:

$$\begin{aligned} & F(x_r, x_{r+1}, x_{r+2}, x_{r+3}) \\ = \ & (x_{r+1}, x_{r+2}, x_{r+3}, N^r(x_r) \oplus \\ & (T_1^r(y_1^r) \parallel T_2^r(y_2^r) \parallel \cdots \parallel T_j^r(y_j^r))), \end{aligned}$$

where x_i is an n-bit plaintext, $(y_1^r \parallel y_2^r \parallel \cdots \parallel y_j^r) = M^r(x_{r+1} \oplus x_{r+2} \oplus x_{r+3})$, M^r and N^r are affine transformations, T_j^r are look-up tables and named as T-box, y_j^r are 16-bit input, the output of a T-box is also 16 bits, $j = 1, 2, \ldots, n/16$, and $r = 1, 2, \ldots, 10$ is the round number.

The framework of our scheme is as follows (see Fig.1).

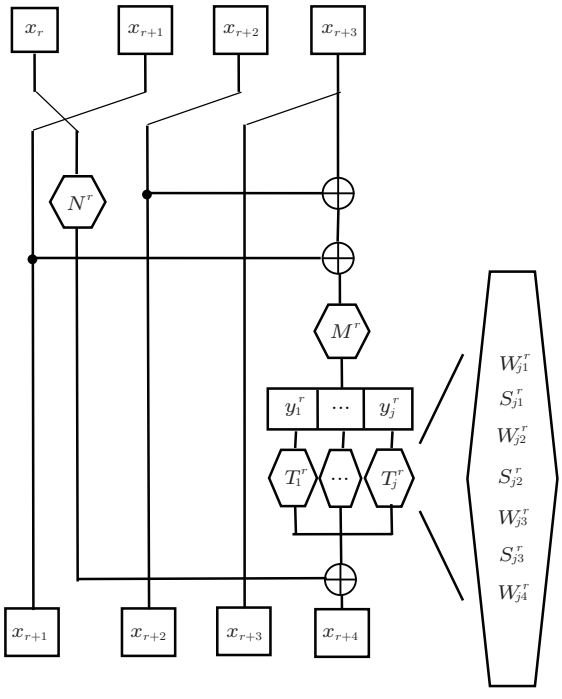


Fig.1. One round of our scheme.

4.2 Affine Parts

The affine transformations M^r and N^r are products of several affine transformations, which can be pre-

sented as $n \times n$ matrices and n-bit constants:

$$M^r = P^r \circ Q^r \circ E^r,$$
$$N^r = P^r \circ U^r,$$

where P^r, Q^r, E^r, and U^r are affine transformations and the corresponding $n \times n$ matrices and n-bit constants are generated by a pseudo random number generator (PRNG), such as a properly chosen block cipher. That is, with a randomly chosen primary key, a data stream is generated by the PRNG, and then it will be arranged as an $n \times n$ matrix and checked whether it is invertible (with time complexity $O(n^3)$); if not, the data stream is discarded. Then the n-bit constant is also generated.

We should note that P^1 is an external encoding of the scheme and used to cancel out the encoding of the encoded inputs, which are provided by a larger system that contains our scheme. This means that the platform running the scheme is expected to provide inputs with the inverse of P^1. With such external encoding, our scheme is customized in the implementation surroundings and cannot be used in other surroundings. The functionality of external encoding has been elaborated in [11], but we will also explain it in Claim 2 presented in Subsection 5.2.

4.3 T-Box Parts

In the T-box part, we propose to employ $n/16$ T-boxes, with each T-box constructed by seven layers of the form ASASASA. The layers A are 16-bit affine transformations W^r_{ji} $(j = 1, 2, ..., n/16;\ i = 1, 2, 3, 4)$, which are produced using the same method as described in Subsection 4.2. Layers S are 16-bit nonlinear look-up tables, and generated randomly by using a PRNG, i.e., with a primary key, a data stream is generated by the PRNG, and every 16×2^{16} bits of the data stream are arranged as a look-up table S^r_{jt}. After all the seven layers are generated, we composite them into a look-up table T^r_j:

$$T^r_j = W^r_{j4} \circ S^r_{j3} \circ W^r_{j3} \circ S^r_{j2} \circ W^r_{j2} \circ S^r_{j1} \circ W^r_{j1}.$$

5 Security Analysis of the Scheme

5.1 White-Box Diversity and White-Box Ambiguity

The white-box diversity and the white-box ambiguity are two basic standards for evaluating the security of a white-box scheme, which were presented and used by Chow *et al.* in [10-11].

White-Box Diversity. White-box diversity means the variability of implementations and is used to prevent pre-packaged attacks. We calculate the diversity by counting all possible constructions of every step in the implementation. Our implementation includes 10 rounds, and each round includes two affine transformations and $n/16$ 16-16 look-up tables. For transformations M^r and N^r, in the field $GF(2)$, the number of nonsingular matrices of order n is:

$$(2^n - 1) \times \prod_{j=1}^{n-1} (2^n - 1 - \sum_{k=1}^{j} \binom{j}{k}).$$

The number of n-bit constants is 2^n; thus the white-box diversity of M^r and N^r is

$$(2^n - 1) \times \prod_{j=1}^{n-1} (2^n - 1 - \sum_{k=1}^{j} \binom{j}{k}) \times 2^n.$$

For T-box, the possible number of each non-linear look-up table S^r_{ji} is $2^{16 \times 2^{16}}$ and the possible number of W^r_{ji} is 2^{270} (the number of nonsingular matrices of order 16 is approximately 2^{254}). Thus the white-box diversity of T-box is $(2^{270})^4 \times (2^{16 \times 2^{16}})^3$.

We use Table 1 to show the white-box diversity for each component of our scheme.

Table 1. White-Box Diversity for Each Component of Our Scheme

Component	White-Box Diversity
M^r	$(2^n - 1) \times \prod_{j=1}^{n-1} (2^n - 1 - \sum_{k=1}^{j} \binom{j}{k}) \times 2^n$
N^r	$(2^n - 1) \times \prod_{j=1}^{n-1} (2^n - 1 - \sum_{k=1}^{j} \binom{j}{k}) \times 2^n$
T-box	$(2^{270})^4 \times (2^{16 \times 2^{16}})^3$

White-Box Ambiguity. White-box ambiguity is used to estimate how many distinct constructions can produce the same component of the implementation. A large white-box ambiguity is helpful in preventing a white-box adversary from performing disambiguation. For example, for affine transformation M^r, different P^r, Q^r, and E^r may lead to the same M^r, and the adversary needs to determine the right collocation of these components.

Let $\boldsymbol{L}(\boldsymbol{M^r})$ and $\boldsymbol{c}(\boldsymbol{M^r})$ be the linear part and the constant part of affine M^r respectively. In a similar way, let $\boldsymbol{L}(\boldsymbol{E^r})$, $\boldsymbol{c}(\boldsymbol{E^r})$, $\boldsymbol{L}(\boldsymbol{P^r})$, $\boldsymbol{c}(\boldsymbol{P^r})$, $\boldsymbol{L}(\boldsymbol{Q^r})$ and $\boldsymbol{c}(\boldsymbol{Q^r})$ be the linear part and the constant part of affine E^r, P^r, and Q^r respectively, and let lm_{ij}, e_{kj}, p_{it}, and q_{tk} be the elements of matrices $\boldsymbol{L}(\boldsymbol{M^r})$, $\boldsymbol{L}(\boldsymbol{E^r})$, $\boldsymbol{L}(\boldsymbol{P^r})$

and $\boldsymbol{L(Q^r)}$ respectively. Because $\boldsymbol{L}(M^r)$ is an $n \times n$ matrix, each element in $\boldsymbol{L(M^r)}$ can be expressed by

$$lm_{ij} = \sum_{k=1}^{n}(e_{kj} \times \sum_{t=1}^{n}(p_{it} \times q_{tk})).$$

As there are $n^2 + 2n$ elements in the expression, the number of distinct constructions of a fixed lm_{ij} is 2^{n^2+2n}. Furthermore, as there are $n \times n$ elements in matrix $\boldsymbol{L(M^r)}$, the number of distinct constructions of $\boldsymbol{L(M^r)}$ is $(2^{n^2+2n})^{n\times n}$.

The constant part of M^r is $\boldsymbol{c(M^r) = L(E^r) \cdot L(Q^r) \cdot c(P^r) \oplus L(E^r) \cdot c(Q^r) \oplus c(E^r)}$, because $\boldsymbol{L(E^r)}$ and $\boldsymbol{L(Q^r)}$ have been determined in the previous step, and the undetermined elements are $\boldsymbol{c(P^r)}$, $\boldsymbol{c(Q^r)}$ and $\boldsymbol{c(E^r)}$. The number of distinct constructions of $\boldsymbol{c(M^r)}$ is 2^{3n}, because each undetermined element is an n-dimensional vector.

On the basis of the reasoning above, the white-box ambiguity of affine transformation M^r is $(2^{n^2+2n})^{n\times n} \times 2^{3n} = 2^{n^4+2n^3+3n}$. In the same way, the white-box ambiguity of N^r is $2^{n^4+2n^3+3n}$.

T-box is simpler: because all the inputs and outputs are known for a fixed T-box, if we can determine six components out of ASASASA, the remaining one will be uniquely determined. Thus the white-box ambiguity of a T-box is $(2^{270})^4 \times (2^{16\times 2^{16}})^2$.

We use Table 2 to show the white-box ambiguity for each component of our scheme.

Table 2. White-Box Ambiguity for Each Component of Our Scheme

Component	White-Box Ambiguity
M^r	$2^{n^4+2n^3+3n}$
N^r	$2^{n^4+2n^3+3n}$
T-box	$(2^{270})^4 \times (2^{16\times 2^{16}})^2$

5.2 White-Box Security w.r.t. Equivalent Key

In this subsection, we use two claims to show that our white-box scheme satisfies the white-box security w.r.t. equivalent key.

Claim 1. *Our scheme is white-box secure w.r.t. equivalent key* $\tilde{K} = \{P^r, Q^r, E^r, U^r, W^r_{jt}, S^r_{jn}\}$, *where* $r = 1, 2, \ldots, 32$, $j = 1, 2, \ldots, n/16$, $t = 1, 2, 3, 4$, *and* $n = 1, 2, 3$.

Proof. For $r = 1, 2, \ldots, 32$, $j = 1, 2, \ldots, n/16$, $t = 1, 2, 3, 4$, and $n = 1, 2, 3$, the key $\tilde{K} = \{P^r, Q^r, E^r, U^r, W^r_{jt}, S^r_{jn}\}$ is an equivalent key because the components in $\tilde{K}$ can be used to recover the white-box implementation of our scheme. In the following, we will explain why it is difficult to obtain the equivalent key.

First, as we have elaborated in Subsection 4.1, the huge white-box ambiguity prevents a white-box adversary from confirming P^r, Q^r, E^r, and U^r with M^r and N^r in our scheme.

Second, the security of the ASASASA structure prevents a white-box adversary from recovering W^r_{jt}, S^r_{jn}.

In fact, Biryukov and Shamir proposed a structural cryptanalysis of SASAS by using the multiset properties in [38-39], where each A layer is an affine transformation and each S layer contains some bijective S-boxes. With the structural cryptanalysis, the layers S and A are determined and peeled off successively. Even so, S layers in our scheme are generated randomly by using a PRNG, and they are not necessarily bijective S-boxes and do not satisfy multiset properties. Thus, the structural cryptanalysis cannot work in our scheme.

In [28], Minaud *et al.* presented an attack against the white-box ASASA design, where the last linear/affine layer was peeled off with an algebraic method, and the remaining layers were recovered by using Biryukov and Shamir's attack in [38]. As we mentioned before, the attack in [38] cannot work in our scheme, and thus the attack in [28] cannot work as well.

In [30], Biryukov and Khovratovich showed that the ASASASAS and SASASASAS structures can be decomposed if the block length l and the smaller S-box size s satisfy the condition $s^2 \leqslant l$, where the smaller S-box is one of the concatenated S-boxes in the nonlinear layer. However, both the block length and the S-box size are 16 in our scheme; thus the scheme is expected to be immune to such decomposition. □

Claim 2. *Our scheme is white-box secure w.r.t. equivalent key* $\tilde{K} = E$.

Proof. Our encryption scheme E is an equivalent key as it is functionally equivalent to itself.

In fact, if a white-box adversary cannot find a way to recover every component of our scheme, a straightforward and practical way to attack our scheme is to lift the code, i.e., to obtain the encryption E. However, we adopt a protection method called external encoding, which is used to guarantee that our scheme is a contained component in a larger system and difficult to be moved out.

In general, external encodings are randomly selected bijections and independent of any other components in a white-box solution. In our scheme, the external en-

coding is the randomly generated affine transformation P^1, and it is integrated into the affine transformations M^1 and N^1. The high white-box ambiguity computed in Subsection 5.1 indicates that it is hard to determine P^1 from M^1 or N^1.

On the other hand, the annihilating encoding $(P^1)^{-1}$ could be built in other parts of the system, e.g., a user authentication code, or a server side. It is separated from the module of our scheme and protected with some other methods which are beyond the consideration of this paper.

That is to say, the containing system will provide a manipulated input (encoded input with the form $(P^1)^{-1}(*)$) to our scheme, and the right output can be obtained only if the external encoding P^1 counteracts the effect of $(P^1)^{-1}$. This makes the scheme no longer an isolated component. Even if the adversary lifts the code of our scheme, it cannot encrypt or decrypt successfully because the external encoding cannot be eliminated. □

As no specific attack against our structure has been found, we conjecture the security level of our scheme to 2^{128}, according to the statement of the security of ASASASA structure in [27].

5.3 Resistance to Popular White-Box Attacks

Since most white-box schemes are white-box variants of existing cryptographic primitives, such as white-box AES implementation[11,19,22], the corresponding attacks account for a large proportion of the white-box attacks. We denote them as popular white-box attacks. The purpose of such attacks like BGE attack[16], differential attack[15], and cryptanalysis of the Xiao-Lai white-box AES implementation[20] is to recover the fixed secret keys of the original cryptographic primitives in their white-box implementations. Meanwhile, the structural and public parameters of the existing cryptographic primitive play the pivotal role of the attacks.

For example, in the first stage of BGE attack, an isomorphism is constructed with the inner structure of AES, then the non-linear parts of the external encodings are recovered with the isomorphism and the affine parts are left. In the second stage, the affine parts are also determined with the structure of AES. In the last stage, the secret key of AES is computed with the MixColumn coefficients. The details of the attack are omitted due to the space limitation, and more information can be found in [16].

However, our scheme is a new white-box solution, but not a variant of any existing cryptographic primitive. Therefore, there is no secret key of any existing cryptographic primitive hidden in the scheme, which frustrates the purpose of the popular white-box attacks. Furthermore, all the components in the scheme are generated randomly, and any structures or parameters are not relevant to an existing cryptographic primitive. Thus, the popular white-box attacks fail to work on our scheme.

6 Example of the Scheme

We choose $n = 32$ as an example of our scheme.

Affine Transformations. We choose AES-256 as a data stream generator. The master key of AES is properly chosen and the data-stream is produced by encrypting the plaintext with the following form:

$$L||r||i,$$

where L is a user's license number of size 128 bits, r is the round number of size 8 bits, and i is a counter of size 120 bits.

For every 1 024 bits generated by the AES-256, we arrange them into a 32×32 matrix and check whether this matrix is invertible; if so, we assign this matrix and another 32 bits generated by the AES-256 as a constant part to P^r. In the same way, Q^r, U^r, E^r, and W^r_{ji} are produced separately.

We multiply the transformations and obtain $M^r = P^r \circ Q^r \circ E^r$ and $N^r = P^r \circ U^r$. W^r_{ji} is used for constructing the T-box.

Nonlinear Transformation. The non-linear layer is implemented by a look-up table. We generate such look-up tables by generating 16×2^{16}-bit data stream with AES-256. Similarly, a new master key of AES is properly chosen, and the plaintext is constructed as follows:

$$L||r||p||i,$$

where L is a user's license number of size 128 bits, r is the round number of size 8 bits, p is the number of look-up table of size 8 bits, and i is the counter of size 112 bits.

After three S-boxes are generated, we combine them with four transformations W^r_{ji} and obtain a larger look-up table T-box T^r_j.

Implementation. This scheme is implemented in $r = 10$ rounds, with every round involving two matrix multiplications, two table look-ups and four XORs. The memory requirement of this implementation is $10 \times (32 \times 32 + 32 \times 32 + 2 \times 16 \times 2^{16}) = 20\,992\,000$ bits $= 2\,562.5$ KB ≈ 2.5 M.

The costly operations in this scheme are table look-ups. There are two tables in each round, and the size of each table is $2^{16} \times 16 = 128$ KB. They are suited for being stored in L1/L2 cache. Assuming that each random access to the tables takes five cycles, this white-box scheme (with 128-bit block size and 10 rounds) will take about six cycles per byte.

7 Summary and Comparison

Summary. We summarize the white-box diversity, white-box ambiguity, and estimated security level of our scheme for different parameters n in Table 3. Mainly because of the use of the unbalanced Feistel network, we can see that with the growth of n, the size of the scheme grows slowly, while the white-box diversity and ambiguity grow fast.

Comparison. Because the only public white-box ASASASA we can find was presented by Biryukov *et al.* in [27], in Table 4, we compare some measures for 128-bit block cipher between white-box ASASASA and our scheme. The table shows that, with the same security level and block length, both the number of look-up tables and the space requirement of our scheme are less than those of other schemes.

8 Some Discussions

SPACE Ciphers. We note that our scheme and the SPACE schemes look similar on the design: both of them use the unbalanced Feistel network and choose new transformations as the round functions. But there are some essential differences between them. The first difference is that our scheme uses external encoding while the SPACE schemes do not. In our opinion, external encoding and large-sized code are two efficient ways to thwart the code lifting attack. They all have their own restrictions but they all have their markets too. The second difference is that our scheme uses randomly generated transformations to construct the round function while SPACE schemes use a block cipher to construct the round function. The SPACE schemes win in the aspect of the security, while our scheme wins in the aspect of the size.

Black-Box Security. We omit the black-box security analyses of our scheme. In fact, the white-box context is much severer than the black-box context. For example, the classical black-box attacks (like differential, linear, and integral attacks) are generally applied on a cipher with several rounds "encapsulated in a black box", and the attackers have no idea of the internal details of the "black box", such as the inputs and the outputs of the inner rounds, while in white-box context, the attackers are assumed to control all the implementation details of the cipher. The white-box attackers can obtain not only all the inputs and the outputs of each round, but also every component in every round, such as N^r, M^r, and look-up tables in our scheme, thereby there is no need to do black-box attacks.

In other words, the white-box attackers own much more information than the black-box attackers and the white-box attacks are stronger than the black-box attacks. For this reason, we believe that a scheme, which is secure in the white-box context, will be bound to keep secure in the black-box context.

9 Conclusions

The white-box attack context reflects a class of extreme attacks that have turned into reality with the progress of technology. Traditional cryptographic algorithms are completely insecure against these threats. White-box cryptography is therefore the most appropriate tool to study the security of cryptographic algo-

Table 3. Various Measures of Our Scheme for Different Parameters n

n	Block Size	Space Requirement (M)	White-Box Diversity	White-Box Ambiguity	Estimated Security Level
32	128	2.5	See Table 1 ($n = 32$)	See Table 2 ($n = 32$)	128
64	256	5.0	See Table 1 ($n = 64$)	See Table 2 ($n = 64$)	128
128	512	10.0	See Table 1 ($n = 128$)	See Table 2 ($n = 128$)	128

Table 4. Comparison of ASASASA-Constructed White-Box Schemes

	Number of Look-Up Tables	Number of Rounds	Security Level	Space Requirement
White-box ASASASA(I)	64	8	128	8 MB
White-box ASASASA(II)	64	8	128	384 MB
White-box ASASASA(III)	25	5	128	20 GB
Our scheme	20	10	128	2.5 MB

rithms implemented in untrusted environments.

As almost all present white-box solutions have been shown to be insecure, in this paper, we introduced a fresh white-box encryption scheme and a new white-box security definition. The proposed scheme is based on the unbalanced Feistel network and ASASASA structure. It is not a variant of any existing cryptographic primitive, but a completely new white-box solution. The input block size of the scheme is optional while the size of look-up table is fixed. In the look-up table, the size of the affine layer is the same with that of the non-linear layer. Because of its special framework, the scheme is advantageous in bypassing existing white-box attacks and proven to meet the security definition. Furthermore, benefited by the unbalanced Feistel network, our scheme can process a longer data series at one time. From a more practical point of view, our solution has a smaller size while achieving better performance.

References

[1] Shannon C E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2001, 5(1): 3-55.

[2] Kocher P C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Proc. the 16th Annual International Cryptology Conference on Advances in Cryptology*, August 1996, pp.104-113.

[3] Kocher P, Jaffe J, Jun B. Differential power analysis. In *Proc. the 19th Annual International Cryptology Conference*, August 1999, pp.388-397.

[4] Quisquater J J, Samyde D. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *Proc. the International Conference on Research in Smart Cards: Smart Card Programming and Security*, September 2001, pp.200-210.

[5] Wang H. Privacy-preserving data sharing in cloud computing. *Journal of Computer Science and Technology*, 2010, 25(3): 401-414.

[6] Mi H B, Wang H M, Zhou Y F, Lyu M R, Cai H. Localizing root causes of performance anomalies in cloud computing systems by analyzing request trace logs. *Science China Information Sciences*, 2012, 55(12): 2757-2773.

[7] Wang X M, He Z B, Zhao X Q, Lin C, Pan Y, Cai Z P. Reaction-diffusion modeling of malware propagation in mobile wireless sensor networks. *Science China Information Sciences*, 2013, 56(9): 1-18.

[8] Ma X L, Hu H F, Li S F, Xiao H M, Luo Q, Yang D Q, Tang S W. DHC: Distributed, hierarchical clustering in sensor networks. *Journal of Computer Science and Technology*, 2011, 26(4): 643-662.

[9] Zhou C, Sun Y Q. SPMH: A solution to the problem of malicious hosts. *Journal of Computer Science and Technology*, 2002, 17(6): 738-748.

[10] Chow S, Eisen P, Johnson H, van Oorschot P C. A white-box DES implementation for DRM applications. In *Lecture Notes in Computer Science 2696*, Feigenbaum J (ed.), Springer, 2003, pp.1-15.

[11] Chow S, Eisen P, Johnson H, van Oorschot P C. White-box cryptography and an AES implementation. In *Lecture Notes in Computer Science 2595*, Nyberg K, Heys H (eds.), Springer, 2003, pp.250-270.

[12] Jacob M, Boneh D, Felten E. Attacking an obfuscated cipher by injecting faults. In *Lecture Notes in Computer Science 2696*, Feigenbaum J (ed.), Springer, 2003, pp.16-31.

[13] Link H E, Neumann W D. Clarifying obfuscation: Improving the security of whitebox DES. In *Proc. International Conference on Information Technology: Coding and Computing*, April 2005, pp.679-684.

[14] Wyseur B, Michiels W, Gorissen P, Preneel B. Cryptanalysis of white-box DES implementations with arbitrary external encodings. In *Proc. the 14th International Conference on Selected Areas in Cryptography*, August 2007, pp.264-277.

[15] Goubin L, Masereel J M, Quisquater M. Cryptanalysis of white box DES implementations. In *Proc. the 14th International Conference on Selected Areas in Cryptography*, August 2007, pp.278-295.

[16] Billet O, Gilbert H, Ech-Chatbi C. Cryptanalysis of a white box AES implementation. In *Proc. the 11th International Conference on Selected Areas in Cryptography*, August 2005, pp.227-240.

[17] Michiels W, Gorissen P, Hollmann H D L. Cryptanalysis of a generic class of white-box implementations. In *Lecture Notes in Computer Science 5381*, Avanzi R M, Keliher L, Sica F (eds.), Springer, 2009, pp.414-428.

[18] Lepoint T, Rivain M, De Mulder Y, Roelse P, Preneel B. Two attacks on a white-box AES implementation. In *Lecture Notes in Computer Science 8282*, Lange T, Lauter K, Lisoněk P (eds.), Springer, 2014, pp.265-285.

[19] Xiao Y Y, Lai X J. A secure implementation of white-box AES. In *Proc. the 2nd International Conference on Computer Science and its Applications*, December 2009, pp.153-158.

[20] De Mulder Y, Roelse P, Preneel B. Cryptanalysis of the Xiao-Lai white-box AES Implementation. In *Lecture Notes in Computer Science 7707*, Knudsen L R, Wu H P (eds.), Springer, 2013, pp.34-49

[21] Biryukov A, De Cannière C, Braeken A, Preneel B. A toolbox for cryptanalysis: Linear and affine equivalence algorithms. In *Lecture Notes in Computer Science 2656*, Biham E (ed.), Springer, 2003, pp.33-50.

[22] Karroumi M. Protecting white-box AES with dual ciphers. In *Lecture Notes in Computer Science 6829*, Rhee K H, Nyang D (eds.), Springer, 2011, pp.278-291.

[23] Bringer J, Chabanne H, Dottax E. White box cryptography: Another attempt. *IACR Cryptology ePrint Archive*, 2006.

[24] De Mulder Y, Wyseur B, Preneel B. Cryptanalysis of a perturbated white-box AES implementation. In *Lecture Notes in Computer Science 6498*, Gong G, Gupta K C (eds.), Springer, 2010, pp.292-310.

[25] Xiao Y Y. White-Box cryptography and implementations of AES SMS4. In *Proc. the Chaincrypto*, Nov. 2009, pp.24-34. (in Chinese)

[26] Lin T T, Lai X J. Efficient attack to white-box SMS4 implementation. *Journal of Software*, 2013, 24(9): 2238-2249. (in Chinese)

[27] Biryukov A, Bouillaguet C, Khovratovich D. Cryptographic schemes based on the ASASA structure: Black-box, white-box, and public-key (Extended Abstract). In *Lecture Notes in Computer Science 8873*, Sarkar P, Iwata T (eds.), Springer, 2014, pp.63-84.

[28] Minaud B, Derbez P, Fouque P A, Karpman P. Key-recovery attacks on ASASA. In *Lecture Notes in Computer Science 9453*, Iwata T, Cheon J H (eds.), Springer, 2015, pp.3-27.

[29] Dinur I, Dunkelman O, Kranz T, Leander G. Decomposing the ASASA block cipher construction. Cryptology ePrint Archive, Report 2015/507, 2015. http://eprint.iacr.org/2015/507, Jan. 2017.

[30] Biryukov A, Khovratovich D. Decomposition attack on SASASASAS. https://eprint.iacr.org/2015/646.pdf, Jan. 2017.

[31] Bogdanov A, Isobe T. White-box cryptography revisited: Space-hard ciphers. In *Proc. the 22nd ACM SIGSAC Conference on Computer and Communications Security*, October 2015, pp.1058-1069.

[32] Feistel H. Cryptography and computer privacy. *Scientific American*, 1973, 228(5): 15-23.

[33] Data Encryption Standard, Federal Information Processing Standard (FIPS). National Bureau of Standards, U.S. Department of Commerce, Washington D. C., Jan. 1977.

[34] Rivest R L, Robshaw M J B, Sidney R, Yin Y L. The RC6TM block cipher. In *Proc. the 1st Advanced Encryption Standard* (*AES*) *Conference*, August 1998, pp.82-104.

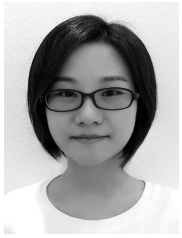
[35] Schneier B, Kelsey J, Whiting D, Wagner D, Hall C, Ferguson N. Twofish: A 128-bit block cipher. NIST AES Proposal, 1998. https://www.schneier.com/academic/archives/1998/06/twofish_a_128-bit_bl.html, Jan. 2017.

[36] Patarin J, Goubin L. Asymmetric cryptography with S-boxes: Is it easier than expected to design efficient asymmetric cryptosystems? In *Lecture Notes in Computer Science 1334*, Han Y F, Okamoto T, Qing S H (eds.), Springer, 1997, pp.369-380.

[37] Biham E. Cryptanalysis of Patarin's 2-round public key system with S boxes (2R). In *Lecture Notes in Computer Science 1807*, Preneel B (ed), Springer, 2000, pp.408-416.

[38] Biryukov A, Shamir A. Structural cryptanalysis of SASAS. *Journal of Cryptology*, 2010, 23(4): 505-518.

[39] Biryukov A, Shamir A. Structural cryptanalysis of SASAS. In *Lecture Notes in Computer Science 2045*, Pfitzmann B (ed.), Springer, 2001, pp.395-405.



Ting-Ting Lin received her Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, in 2016. Her research interests are theory and techniques of white-box cryptography, block cipher, software security, and obfuscation.

Xue-Jia Lai received his B.S. degree in electrical engineering in 1982 and M.S. degree in mathematics in 1984 from the Xidian University, Xi'an. He received his M.S. degree in 1988 and his Ph.D. degree in 1992 from the Swiss Federal Institute of Technology, Zurich. He has been evaluating, analysing, and improving several ciphers for several international companies and organisations, and involved in the European project KRISIS, ICE-CAR and PKI Challenge. He has been a professor of Shanghai Jiao Tong University, Shanghai, since 2004. His work has been concentrated in cryptography and PKI during the past 20 years, especially in the design and analysis of practical cryptosystems (including block ciphers and stream ciphers), differential cryptanalysis of block ciphers, and analysis of hash functions.

Wei-Jia Xue received her B.S. degree in computer science in 2011 at Shanghai Jiao Tong University, and she is working for her Ph.D. degree at Shanghai Jiao Tong University, Shanghai. Her major research interests include cryptography, the design and analysis of block-cipher.

Yin Jia is a Master student in computer science at Shanghai Jiao Tong University, Shanghai. Her major research interests include the design and analysis of white-box cryptography, and cyber security.