WBCD: White-box Block Cipher Scheme Based on Dynamic Library

Yatao Yang[1,2,*], Ye Zhang[1], Yuying Zhai[2], Zheng Yuan[1] , Guangwu Xu[3]

[1] Beijing Electronic Science and Technology Institute, Beijing 100070, China
yy2008@163.com zyuan@tsinghua.edu.cn
[2] School of Telecommunication Engineering, Xidian University, Xi'an, 710071, China
[3] Department of EE and CS, University of Wisconsin-Milwaukee, Milwaukee, USA

Abstract. The aim of white-box cryptography is to protect a secret key in a white-box environment in which an adversary has full control ability over the computer's execution process and the running environment. In order to solve the issues of lower security in static white-box algorithm and inconvenient application in traditional dynamic white-box algorithm, it is proposed that a white-box block cipher scheme based on dynamic library named WBCD. In this scheme, look-up tables and affine transformations are used to construct dynamic white-box library, which ensure that the different look-up tables can be used for each round of encryptions. In order to illustrate the effectiveness of WBCD, it is designed a novel white-box mechanism (WBDL) based on dynamic library, ,which adopt MDS matrix. In this mechanism, different round-keys have been employed to implement encryption by randomly selecting look-up tables in each round of operations. According to the analysis, WBDL mechanism can resist differential attack, linear attack, BGE attack and side channel energy attack against SM4. After being calculated and tested, WBDL mechanism requires 466.914KB of memory to store the look-up tables, maximum differential probability(MDP) of each round is 2^{-26}, maximum linear probability(MLP) of each round is $2^{-25.61}$, the encryption speed can reach to 0.273×10^{-3} Gbps, and decryption speed can achieve 0.234×10^{-3} Gbps. Our mechanism has better security and working efficiency, which can be used in mobile communication security and digital payment security.

Keywords: white-box cryptography; dynamic white-box; SM4; look-up table; affine transformation

1 Introduction

It is of great security significance to design secure and efficient white-box cryptographic algorithms. Therefore the ultimate purpose of applying the white-box cipher is to resist the white-box attack and provide effective security protection [1] . Since the concept was proposed, there have been two main branches for the research and construction of white-box cryptographic algorithms. The first strategy is to transform the standard cryptographic algorithm into white-box cryptographic algorithms, which can be used in the white-box attack environment [2] .The research results are mainly focused on block cipher algorithms. The second branch is to innovate the structure to construct a new white-box cryptography algorithm [3] . Including the constructions based on ASASA structure and its derivative structure, lightweight white-box encryption scheme and so on [3] .

1.1 White-box Related Work

In 2002, Chow *et al.* proposed the concept of white-box attack context WBAC [1] . It assumed that the attacker had completely controlled over the user's device terminal and owned special permission to monitor, observe and change the running process arbitrarily and internal data of the program. The attacker could get entire key under this circumstance. But Chow [2] also gave the countermeasures, he decomposed the cipher algorithm into multiple steps, and inserted some random mixing bijections into each step as fuzzy processing. He proposed the first white-box AES algorithm (Advanced Encryption Standard) [1] and the first white-box DES algorithm [2]. By using a set of key-dependent look-up tables. Later, Billet *et al.* [4] used BGE attack and Jacob *et al.* [5] used injection fault attack to extract the key successfully.

In 2006, Julien *et al.* [6] applied the idea of obfuscation to the design of white-box AES algorithm and proposed a method of white-box construction, the security of this approach due to polynomial. In 2009, Yaying Xiao and Xuejia Lai [7] proposed an improved algorithm for Chow's white-box AES algorithm, this improved algorithm broke the boundary of each round and recombined the operation steps, increased the time complexity of BGE attack. However, in 2010, Mulder *et al.* [8] cracked the Bringer's white-box AES algorithm by using algebraic analysis. In 2012, Mulder *et al.* [9] used linear/affine equivalent algorithm to crack the Xiao-Lai' white-box AES algorithm.

In 2010, Park *et al.*[10] proposed the concept of dynamic white-box cryptography, which used a specific mode with dual-key acceleration to speed up the calculation operations, so that the key could be updated in real time.In 2011, Karroumi [11] proposed a white-box AES algorithm with double keys, and the complexity of BGE attack was 2^{91}.In 2013, Lepoint *et al.* [12] broke Karroumi's white-box AES algorithm. In 2014, Rui Luo [13] proposed an improved white-box AES algorithm in which each look-up table contained a two-byte round key. In 2016, Baek *et al.* [14] introduced an idea to obfuscate two 128-bit algorithms simultaneously with 256-bit input/output encoding. In 2016, Sasdrich *et al.* [15] proposed the implementation of white-box AES algorithm in reconfigurable hardware platform and evaluated their method under the assumption of a grey box attacking situation, however, experiments showed that their method could not resist side channel attack (SCA) attacks. In 2016, Kunpeng Bai*et al* [16] proposed a white-box implementation scheme for SM4 algorithm based on look-up tables, which could protect large linear encoding from being counteracted. In 2017,Banik *et al.* [17] proposed a zero-difference enumeration (ZDE) attack, then, they designed a method to protect the white-box binary files under ZDE attack. In 2018, Tao XU *et al.* [18] constructed one white-box algorithm by using the method of obfuscating round boundaries. In 2018, Tao Xu *et al.* [19] proposed a white-box AES algorithm based on substitution and linear transformation, which could resist existing typical attacks.

In recent years, the research on the white-box of existing algorithms is not limited to block cipher algorithms. In 2017, Jie Zhou *et al.* [20] designed a software implementation method for white-box SM2 algorithm based on Chinese Remainder Theorem, which effectively protected the security of private key storage and reduced the storage space for calculation. In 2018, Dawu Gu *et al.* [21] described the implementation method of white-box SM2 algorithm in the digital signature scenario, which could ensure the security of the signature private key. In 2018, Debiao He *et al.* [22] published the white-box implementation method for SM9 digital signature, which could protect the private key of SM9's signature under the white-box attacks. In 2019, Bock [23] described new methods for evaluating the security of white-box implementations, which did not require extra look-up tables.

All of the above white-box algorithms were transformed from the respective standard cipher algorithms. However, in recent years, many researchers have developed new white-box cryptographic algorithms by designing novel and independent white-box cryptographic

structures.

In 2014, Alex Biryukov *et al.*[3] proposed a white-box algorithm based on the structure of Affine-Sbox, in which the key was encapsulated in a table in ASASA, however, this structure was attacked by Minaud *et al.* [24] and Dinur *et al.* [25].In 2015, Biryukov and Khovratovich [26] showed that the structures such as ASASASAS and SASASASAS could be attacked by decomposition algorithms if the block's length l and the box's size S satisfied the condition $s^2 <= l$. In 2015, Bogdanov and Isobe [27] proposed a special white-box scheme, they used the Feistel network and implemented the loop function. In 2017, Yang Shi *et al.* [28] proposed a lightweight white-box scheme, in which the encryption and decryption algorithms were based on a substitution-permutation network composed of random secure components. In 2016 and 2017, Jihoon Cho *et al.* proposed the mixed white-box scheme [29] and the WEM scheme respectively [30]. In 2017, Tingting Lin [31] proposed a new scheme based on non-equilibrium Feistel network and the structure of ASASASASASA, which had higher white-box diversity and white-box confusion.

With the deepening of research, the application field of white box passwords has gradually expanded. In 2015, Yang Shi *et al.* [32] proposed a lightweight white-box symmetric encryption algorithm for wireless sensor network, which incorporated several important steps of SM4 algorithm into the look-up tables by applying random mixed bijections. In 2018, Lu Zhou *et al.* [33] proposed a lightweight white-box block cipher for the internet of things. In 2019, Şengel *et al.* [34] developed a mobile payment system based on the white-box algorithm to protect consumers' digital cash and the confidentiality of their transaction information.

Based on the analysis about these academic literatures, the static white-box cryptographic algorithms with definitive secret key have been researched for a long time, the static white-box library will have to be updated regularly when it is being used, and its security also is lower. So, the dynamic white-box cryptographic algorithms are getting more and more concerns these years. However, the design and implementation for ordinary dynamic white-box cryptographic algorithms are complicated, and it is also inconvenient to use them in real information system. In order to solve these issues, we abandon the traditional concept on initial key in the static white-box cryptographic algorithms, a novel dynamic white-box block cipher scheme is proposed, which is easier to implement on software platform than the ordinary dynamic white-box algorithms.

1.2 MDS matrix

The MDS matrix can reach the upper bound of the number of branches, and it can effectively resist differential analysis and linear analysis, which is a common tool for constructing linear diffusion layers of block ciphers [35]. In 2003, literature [36] proposed that MDS matrices that are easy to implement can be searched from special matrices such as cyclic shift matrix, Cauchy matrix, and Hadamard matrix. In the 1990s, literature [37] and literature [38] proposed the idea and construction method of designing MDS matrices by using Cauchy matrix, but the number of MDS matrices that can be constructed is not many, that is, when the matrix series is given, this Both methods can only construct one or a few MDS matrices. In 2010, literature [39] based on the construction method of Cauchy matrix and Hadamard matrix gave a kind of construction method of Cauchy-Hadamard matrix. In 2016, Sarkar *et al.* [40] used a special structure matrix, Toeplitz matrix, to construct a 4∗4MDS matrix on the finite field F_{2^4} and F_{2^8}, XORs reached the optimal results at that time. In 2017, Jean *et al.* [41] presented a new XORs calculation method. The new calculation method does not consider the use of temporary registers. This is easier to implement in practical applications and can reduce the XORs of the matrix. Using the new XORs calculation method, literature [42] constructed a multi-dimensional MDS circulant matrix with XORs superior to the known optimal results at that time.

1.3 Our Contributions

The contributions in our article include:

(1) WBCD is proposed. We propose a white-box block cipher scheme based on dynamic library named WBCD. We abandon the concept of the traditional initial key in static white-box cryptosystem. Look-up tables and affine transformations are used to construct dynamic white-box library, which ensure that the different look-up tables can be used for each round of encryptions. By using Hadamard-type MDS matrices, both the security and computing efficiency have been improved.

(2) WBDL is designed and tested. In order to show the effectiveness of WBCD, we design a white-box mechanism (WBDL) based on the dynamic library. In this mechanism, different round-keys are employed to implement encryption by randomly selecting look-up tables in each round of operations. According to the analysis, WBDL mechanism can resist differential attack, linear attack, BGE attack and side channel energy attack against SM4. After being calculated and tested, the encryption speed in WBDL can reach 0.268×10^{-3} Gbps.The maximum differential probability MDP is $2^{-26} = 32/2^{31}$the maximum linear probability MLP is $2^{-25.61}(= 84 \cdot 2^{-32})$. Our mechanism has better security and working efficiency.

1.4 Organization

The structure of this article is as follows: In the first section, we introduce the background and development situation of the white-box cryptography, the basic knowledge of static white-box and dynamic white-box is introduced in the second section. Afterwards, in the third section, the WBDL mechanism is introduced, the correctness of encryption and decryption processes in WBDL is validated. We conduct a comprehensive analysis for WBDL mechanism in the fourth section. The last section is the summary and prospect.

2 Preliminaries

2.1 Symbol description

Table 1: Symbol description

Symbol	Symbolic meaning
$+$	Add on real field
$-$	Sub on real field
$\times$	Multiplication on real field
$\bigoplus$	Add on finite field (modulo 2 addition)
$\cdot$	Multiplication on finite field
$\bigoplus_{i=0}^{n-1} a_i$	Addition summation of n elements on finite fields
A^T	Transposition of matrix A
$\circ$	Synthesis of invertible affine transformation operations
E	Identity matrix

2.2 Static and dynamic white-box analysis

The implementation of white-box cryptography can be divided into two types: static white-box and dynamic white-box. Both static white-box and dynamic white-box are generated by standard cryptographic algorithms processed by white-box. On the one hand, the secret keys in the static white-box are combined with the nonlinear transformation in the cryptography algorithm to form a number of look-up tables, then, these look-up tables will be uniformly stored to form the white-box library[7]. At this time, the keys are a fixed part in the look-up tables, and the user can directly input the plaintext in the static white-box library for encryption and decryption. On the other hand, the keys in the dynamic white-box are separable from the look-up tables[43]. The comparison of the static white-box and dynamic white-box is shown in Figure 1.

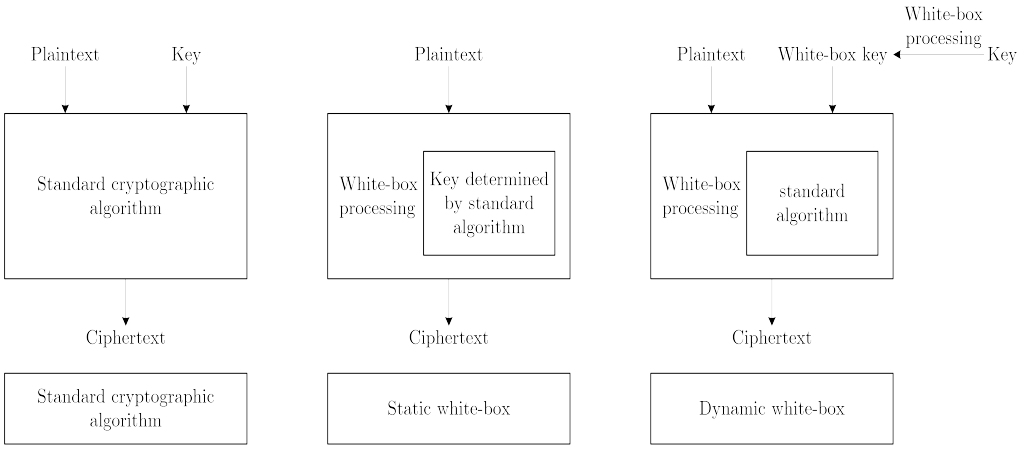

Figure 1: Comparison of static white-box and dynamic white-box

(1)Static white-box

In the static white-box cipher algorithms, the white-box library can be built by the keys processed by white-box cryptography, the white-box library has specific encryption and decryption functions, which can protect the security of the original keys effectively under the white-box attack environment. By regenerating the white-box library, the keys in the static white-box can be updated.

In 2002, Chow *et al*. first proposed the white-box AES scheme. The main idea was to transform the AES algorithm into a series of independent look-up tables, and they used the combination of internal and external encoding to hide the keys.

Xiao-Lai's white-box SM4 scheme[7]was a white-box construction based on the SM4 algorithm. Its plaintext length and key length were both 128 bits, and the ciphertext length was also 128 bits. Furthermore, the encryption process consisted of 32 rounds of iterative operations and reverse order transformations. Each round of operation was divided into three parts: Part 1, Part 2 and Part 3. The overall structure of Xiao-Lai's white-box SM4 scheme was shown in Figure 2.The network coding was adopted, the output encoding of the previous transformation was counteracted by the input encoding of the next transformation. So, the input scrambling codes and output scrambling codes were needed before and after each transformation. Moreover, the input/output scrambling codes of transformation were used to form a look-up table, which the keys would be hidden in it.

In general, the first part performed three affine transformations and two XOR operations, the second part performed four 8-bit to 32-bit look-up tables and three XOR operations, and the third part performed two affine transformations and one XOR operation. In this

way, the affine transformations were used to generate the input/ output encoding, and the keys were hidden in the look-up tables. Even if the cryptanalyst could read the computer memory, it was nearly impossible to obtain the key information.

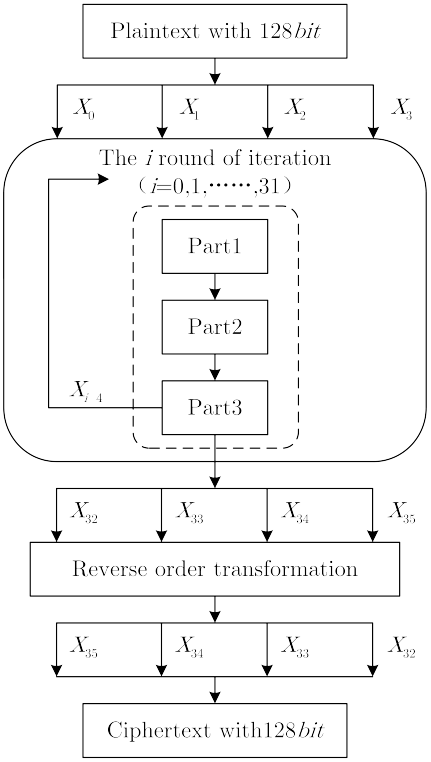

Figure 2: Xiao-Lai's static white-box structure

(2)Dynamic white-box

In the dynamic white-box algorithms, it is unnecessary to update secret keys after the dynamic white-box library being generated, which is a highlighted strongpoint compared with static white-box, the original keys were transformed into the white-box keys by the white-box cryptography technology. The attacker cannot obtain any useful information about the original keys by analyzing the white-box keys[10].

In a dynamic white-box, the white-box library files can be obtained through a combination parameter in commands from library file, and the white-box key files can be obtained through a combination parameter in commands from key file. It becomes very easier to update the secret keys by the encryption operation according to the white-box library file and the decryption operation with the white-box key file. Moreover, there is no need to update the white-box library file when updating the keys, of course, the dynamic white-box is also functionally compatible with the static white-box[43].

In dynamic white-box, the look-up table's source file can be generated dynamically and randomly according to the certain generation rules, thereby it should ensure that every look-up table's source files generated by each library files are different from each other, and the white-box library files generated by each keys are also different. When the keys are updated, the server only needs to send a changeable look-up table related to the particular key, the space occupancy will become very small. According to the user's business requirements and data processing ability, different encryption intensity can be configured flexibly, and the key updating mode is also more flexible and efficient.

2.3 SM4 algorithm

(1)Encryption/decryption process

The SM4 algorithm encryption/decryption process consists of 32 iteration operations and 1 reverse order transformation.

Define the reverse order transformation as:$R(A_0, A_1, A_2, A_3) = (A_3, A_2, A_1, A_0)$,$A_i \in Z_2^{32}$,$i$ = 0,1,2,3.

Assuming that the input of plaintext m is $(X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$,the output of ciphertext is $(Y_0, Y_1, Y_2, Y_3) \in (Z_2^{32})^4$,and the round key is $rk_i \in Z_2^{32}$,i = 0,1,2,3...31.and the white-box key files can be obtained through a combination parameter in commands from key file.

$$X_{i+4} = F(X_i, X_{i+1}X_{i+2}X_{i+3}rk_i) = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i), i = 0,1,2,3...31.$$

$$(Y_0, Y_1, Y_2, Y_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32}).$$

Where F is a round function and T is a synthetic permutation.

The structure of decryption transformation is the same as that of encryption transformation, the only difference is the order of using round keys. The order of encryption key is $(rk_0, rk_1, ..., rk_{31})$, and decryption key is $(rk_{31}, ..., rk_1, rk_0)$.

(2)Round function F

The SM4 algorithm uses a non-linear iterative structure and performs cryptographic operations in units of words. The round function is $F(X_i, X_{i+1}X_{i+2}X_{i+3}rk_i) = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i)$,where $(X_i, X_{i+1}X_{i+2}X_{i+3}) \in (Z_2^{32})^4$ are the input and $rk_i \in Z_2^{32}$ are the round key, and an iterative operation is a round of transformation, as shown in Figure 3.

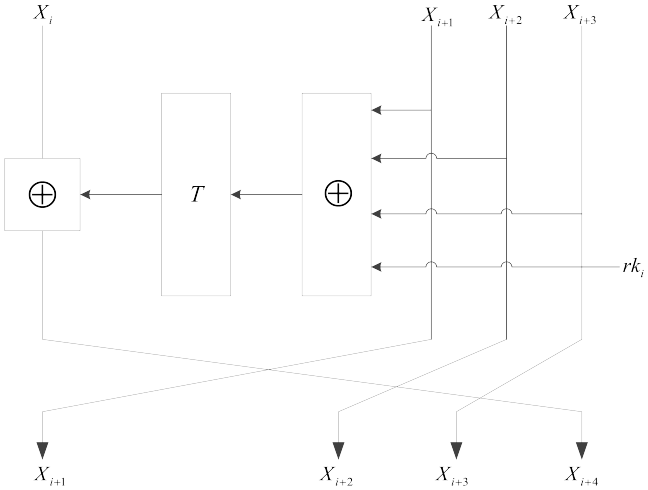

Figure 3: Schematic diagram of SM4 algorithm round function

The synthetic permutation $T : Z_2^{32} \to Z_2^{32}$ in SM4 algorithm is a reversible transformation, which is composed of nonlinear transformation τ and linear transformation$L$,$T(\cdot) = L(\tau(\cdot))$. Synthetic permutation T plays a role of confusion and diffusion in SM4 operation.

①Non-linear transformation τ

The nonlinear transformation τ consists of 4 parallel S boxes. Let input be $A = (a_0, a_1, a_2, a_3) \in (Z_2^8)^4$ and output be $B = (b_0, b_1, b_2, b_3) \in (Z_2^8)^4$, then $(b_0, b_1, b_2, b_3) = \tau(A) = (Sbox(a_0), Sbox(a_1), Sbox(a_2), Sbox(a_3))$

② Linear transformation L

The output of the nonlinear transformation τ is the input of the linear transformation L. Let the input be $B \in (Z_2^{32})$ and the output be $C \in (Z_2^{32})$, then $C = L(B) = B \oplus (B \lll 2) \oplus (B \lll 10) \oplus (B \lll 18) \oplus (B \lll 24)$.

(3)Key expension encipherment

The round key of the above encryption algorithm is generated by the encryption key through a key expansion algorithm. The SM4 algorithm key expansion process is shown in Figure 4.

Assuming that the encryption key is $MK = (MK_0, MK_1, MK_2, MK_3)$,$MK_i \in (Z_2^{32})$,$K_i \in (Z_2^{32})$,$i = 0$,1,...31,and the round key is $rk_i \in (Z_2^{32})$, the method for generating the round key is:

① $(K_0, K_1, K_2, K_3) = ((MK_0) \oplus (FK_0), (MK_1) \oplus (FK_1), (MK_2) \oplus (FK_2), (MK_3) \oplus (FK_3))$;

② $rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i)$;

Where, the transformation T' is basically the same as that in the round function F of the encryption algorithm, except that the linear transformation L therein becomes L' , which is expressed as follows:

$L'(B) = B \oplus (B \lll 13) \oplus (B \lll 23)$

The value of the system parameter FK is expressed in hexadecimal:

$FK_0 = (a3b1bac6), FK_1 = (56aa3350), FK_2 = (677d9197), FK_3 = (b27022dc)$

The fixed parameter CK is valued according to the following method $CK_i = (ck_{i,0}, ck_{i,1}$ $,ck_{i,2}ck_{i,3}) \in (Z_2^8)^4$, Let $ck_{i,j}$ be the j-th byte of $CK_i(i = 0, 1, ..., 31; j = 0, 1, 2, 3)$,$ck_{i,j} = (4i + j) \times 7 mod(256)$.

2.4 MDS matrix

Differential analysis and linear analysis are the most effective attack methods for block cipher algorithms. The minimum value of the sum of the input differential and output differential non-zero active bits of the diffusion layer is called the number of differential branches of the diffusion layer. The minimum value of the sum of 0 active bits is the number of linear branches of the diffusion layer, which can be used to give a bound on the number of S-boxes for block cipher activities, and then to measure the ability of the cryptographic algorithm to resist differential analysis and linear analysis. Therefore, the number of branches is an index to measure the diffusion performance of a diffusion layer. The larger the number of branches, the better the diffusion effect and the higher the safety.

Maximum distance separable(MDS) matrices has the largest number of branches, the best diffusion effect, better resistance to differential analysis and linear analysis, and the highest safety. It is widely used in SM4, AES, CLEFIA , Twofish, FOX and other block cipher algorithms in the diffusion layer design. The MDS matrices in the design of common diffusion layers are generally cyclic matrices, Hadamard matrices, Cauchy matrices, Cauchy-Hadamard (C-H) and other types of matrices. According to reference [44], through the different indicators of the number of elements, the generation efficiency, the performance of various platforms, and the ability to match these four indicators, the summaries of the above four types of matrix generation conditions and engineering applicability summary, the results are as following Table 2.

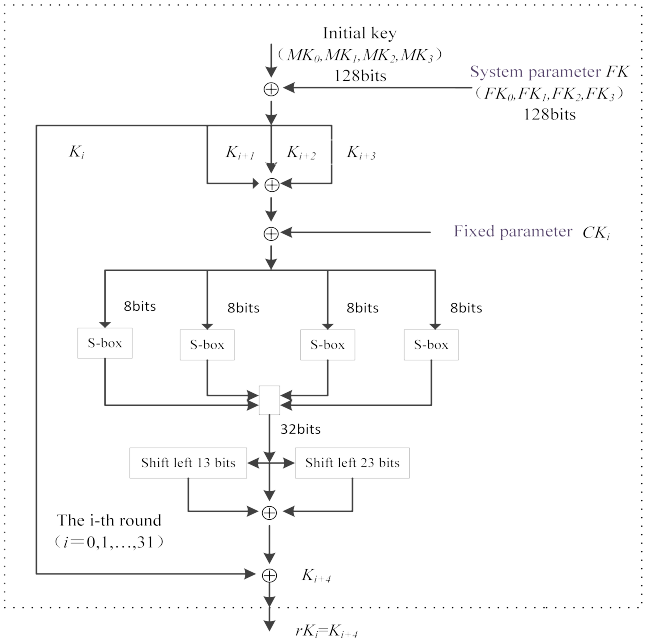

Figure 4: Key expension encipherment

Table 2: Comparison of conditions and performance of 4th order MDS matrix generation

	Number of different elements	Generation efficiency	Achieve performance	Involutorial
Cir	$3 \ll n \ll 4$	good	great	N
Had	$n = 4$	good	great	Y
Cau	$n \gg 4$	great	poor	Y
C-H	$n = 4$	good	good	Y

(level: great> good> poor)

If the circulant matrix is an MDS matrix, the matrix must not be coincident. Therefore, the application of the circulant matrix has limitations; the Cauchy matrix is not conducive to the implementation of the algorithm on various platforms due to the large Hamming weight of the matrix elements. It is not commonly used in algorithm design; Hadamard matrix element Hamming weight is small, and it can be the MDS matrix of the coincidence, which has the characteristics of easy realization. Therefore, this paper uses Hadamard type MDS matrix.

The definition of Hadamard matrix is as follows$A = (a_{i,j})_{2^m \times 2^m}$ is the $2^m \times 2^m$ matrix on GF(2^m), if $a_{i,j} = a_{0,i+j}(0 \le i, j \le 2^m - 1)$then A is called a Hadamard matrix on the finite field GF(2^n), and is abbreviated as $A = Had(a_{0,0}, a_{0,1}, ...a_{0,2^m-1})$.

The Hadamard matrix has the following properties [45]:

Property 1: Necessary and sufficient condition $\bigoplus_{i=0}^{2^{m-1}} a_{0,i} \neq 0$for the invertibility of the 2^m order Hadamard matrix on the finite field GF(2^n).

Property 2:The inverse matrix of the Hadamard type MDS matrix on the finite field GF(2^n) is also the Hadamard type MDS matrix.

Property 3:The necessary and sufficient condition for the 4th order Hadamard matrix

on the finite field GF(2^n) to be an MDS matrix is $\prod_{i=0}^{3} a_{0,i} \neq 0, \bigoplus_{i=0}^{3} a_{0,i} \neq 0$ and $a_0 \cdot a_3 \neq a_1 \cdot a_2$,$a_0 \cdot a_1 \neq a_2 \cdot a_3$,$a_0 \cdot a_2 \neq a_1 \cdot a_3$ and $a_i \neq a_i (i \neq j)(0 \leq i, j \leq n-1)$.

Property 4:The necessary and sufficient condition of 2^m order Hadamard matrix on finite field GF(2^n) is the involutory matrix $\bigoplus_{i=0}^{2^{m-1}} a_{0,i} = 1$.

Property 5:The necessary and sufficient conditions for the 4th order Hadamard matrix on the finite field GF(2^n) to be a coincident MDS matrix are $\prod_{i=0}^{3} a_{0,i} \neq 0, \bigoplus_{i=0}^{3} a_{0,i} \neq 1$ and $a_0 \cdot a_3 \neq a_1 \cdot a_2$,$a_0 \cdot a_1 \neq a_2 \cdot a_3$,$a_0 \cdot a_2 \neq a_1 \cdot a_3$ and $a_i \neq a_i (i \neq j)(0 \leq i, j \leq n-1)$.

Proof of property 4 is as follows:

It is known that A is a Hadamard matrix, so

$$A \cdot A = (\bigoplus_{i=0}^{2^{m-1}} a_{0,i}^2)E = (\bigoplus_{i=0}^{2^{m-1}} a_{0,i})^2 E$$

.

The necessary and sufficient condition for matrix A to be coincidence is $A \cdot A = E$, so

$$(\bigoplus_{i=0}^{2^{m-1}} a_{0,i})^2 = 1$$

.

Equivalent to

$$\bigoplus_{i=0}^{2^{m-1}} a_{0,i} = 1$$

.

3 WBDL Design

WBCD can be applied to many types of block cipher algorithms, in order to show the effectiveness of this scheme, the white-box mechanism (WBDL) is designed by using dynamic library.

3.1 WBDL mechanism

The basic thought of WBDL mechanism can be described as following. Each round in SM4 algorithm is divided into three parts, and the crucial second part is transformed into the dynamic look-up tables, which are confused by the reversible affine transformation as the input scrambling code and the output scrambling code. In order to prevent the attacker from obtaining the secret key, we hide the key information in the dynamic look-up tables. The whole mechanism can be transformed into the process of calculating affine transformations and look-up tables, which has been shown in Figure 5.

The structure of encryption round-function in WBSM4 mechanism has been shown in Figure 4, the scrambling codes are all in the form of reversible affine transformation, and its mathematical expression is:$P_i(x) = l[P_i](x) \oplus c[P_i]$.where P_i represents an affine transformation, $l[P_i]$ is an invertible matrix, which is the linear part of P_i ,$c[P_i]$ is a column vector, which is the constant term of P_i.Due to the existence of the MDS matrix, any input difference will nonlinearly affect all states after one round. In this paper, in order to increase the ability to resist differential analysis and linear analysis in each round, the linear part of the affine transformation adopts involution Hadamard type MDS matrix. The matching matrix can ensure the consistency of the decryption structure and the encryption structure, while reducing the storage space.

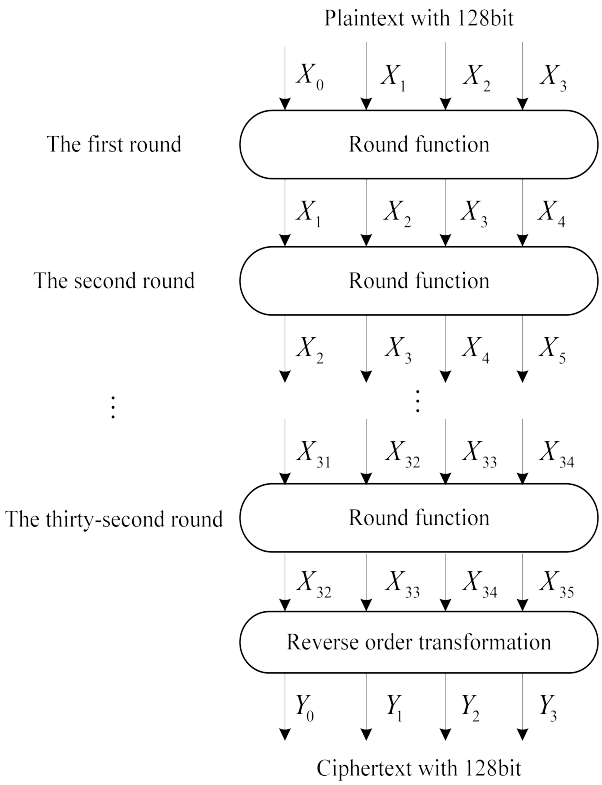

Figure 5: Overall structure of WBDL mechanism

The first part is to calculate X.

$$X = X'_{i+1} \oplus X'_{i+2} \oplus X'_{i+3} = E_{r_i}^{-1} \circ P_{i+1}^{-1}(X_{i+1}) \oplus E_{r_i}^{-1} \circ P_{i+2}^{-1}(X_{i+2}) \oplus E_{r_i}^{-1} \circ P_{i+3}^{-1}(X_{i+3})$$

$$P_{i+j}(x) = A_{i+j}(x) \oplus a_{i+j}$$

Where $\circ$ represents the synthesis of reversible affine transformation operations,P_{i+j} and E_{r_i} are all invertible affine transformations, A_{i+j} is 32×32 invertible matrix on $GF(2)$,a_{i+j} is a constant with 32bit; $E_{r_i} = diag(E_{r_i0}, E_{r_i1}, E_{r_i2}, E_{r_i3})$, E_{r_i0}, E_{r_i1}, E_{r_i2}, E_{r_i3} are 8-bit to 8-bit invertible affine transformations on $GF(2)$; Because of P_{i+j} and E_{r_i} being selected randomly and with secret status, we only save one 32bit to 32bit affine transformation, which is denoted by $M_{i+j} = E_{r_i}^{-1} \circ P_{i+j}^{-1}$.

The second part is to change X with 32bit to Y with 32bit through the r_i-th look-up table.

There are n white-box look-up tables in the dynamic white-box library in the WBDL mechanism, and the white-box tables can be named as table 0, table 1, table 2... table n-1. In the i-th round, the input value is X_{i+2} , X_{i+3} , and the selection factor (SF) for look-up table is H_i , they can perform XOR operation to get W_i

$$W_i = X_{i+2} \oplus X_{i+3} \oplus H_i$$

Where, W_i is 32bit. In WBDL, we adopt n=37. For instance, the first 8 bits of W_i is 11000001, then r_i can be obtained by the following formula.

$$r_i = 11000001 \bmod n = (128 + 64 + 1) \bmod 37 = 8$$

Where, r_i is the first 8 bits of (W_i mod n). Then, the No.8 look-up table will be used in this round.

The third part is to calculate X_{i+4}.

$$X_{i+4} = X''_i \oplus Y' = P_{i+4} \circ P_i^{-1}(X_i) \oplus P'_{i+4} \circ Q_{r_i}^{-1}(Y)$$

Where Q_{r_i} is selected randomly as a 32bit to 32bit invertible affine transformation,

which is employed as the output encoding of the look-up tables. $P'_{i+4}(x) = P_{i+4}(x) \oplus C'_{i+4}$, P'_{i+4} is also an invertible affine transformation and it will be partially counteracted by P_{i+4} in the next round.Because of P'_{i+4} and Q_{ri} being selected randomly and with secret status, it is saved that two 32 bit affine transformations, which are denoted by $C_i = P'_{i+4} \circ Q_{r_i}^{-1}$, $D_i = P_{i+4} \circ P_i^{-1}$.

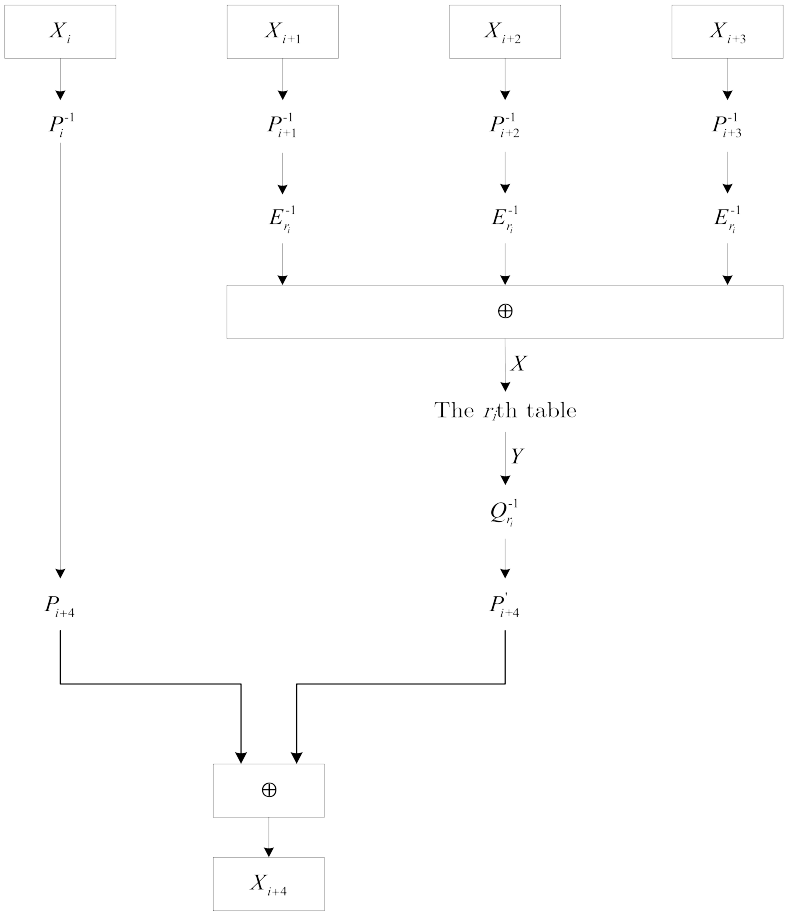

Figure 6: The workflow of encryption round-function

The work flow of decryption round-function in WBDL mechanism is shown in Figure 7, the structure of decryption transformation is similar to that of encryption transformation. The only difference is the order in which look-up tables WBDL used.

3.2 Verification of the correctness of WBDL scheme decryption

To start with, we will introduce the property of invertible affine transformation.

The form of invertible affine transformation can be denoted as $P(x) = l[P](x) \oplus c[P]$, then the inverse transformation of this affine transformation is:$P^{-1}(x) = (l[P])^{-1}(x) \oplus (l[P])^{-1} \cdot c[P]$.

The correctness of the above property can be proved as follows:

Supposing the affine transformation B is $B(x) = (l[P])^{-1}(x) \oplus (l[P])^{-1} \cdot c[P]$,then,

$$\begin{aligned} B \circ [P]^{-1}(x) &= B(l[P](x)) \oplus c[P] \\ &= (l[P])^{-1}(l[P](x) \oplus c[P]) \oplus (l[P])^{-1} \cdot c[P] \\ &= (l[P])^{-1} \cdot l[P](x) \oplus (l[P])^{-1} \cdot c[P] \oplus (l[P])^{-1} \cdot c[P] \\ &= (l[P])^{-1} \cdot l[P](x) \oplus ((l[P])^{-1} \cdot c[P] \oplus (l[P])^{-1} \cdot c[P]) \end{aligned}$$

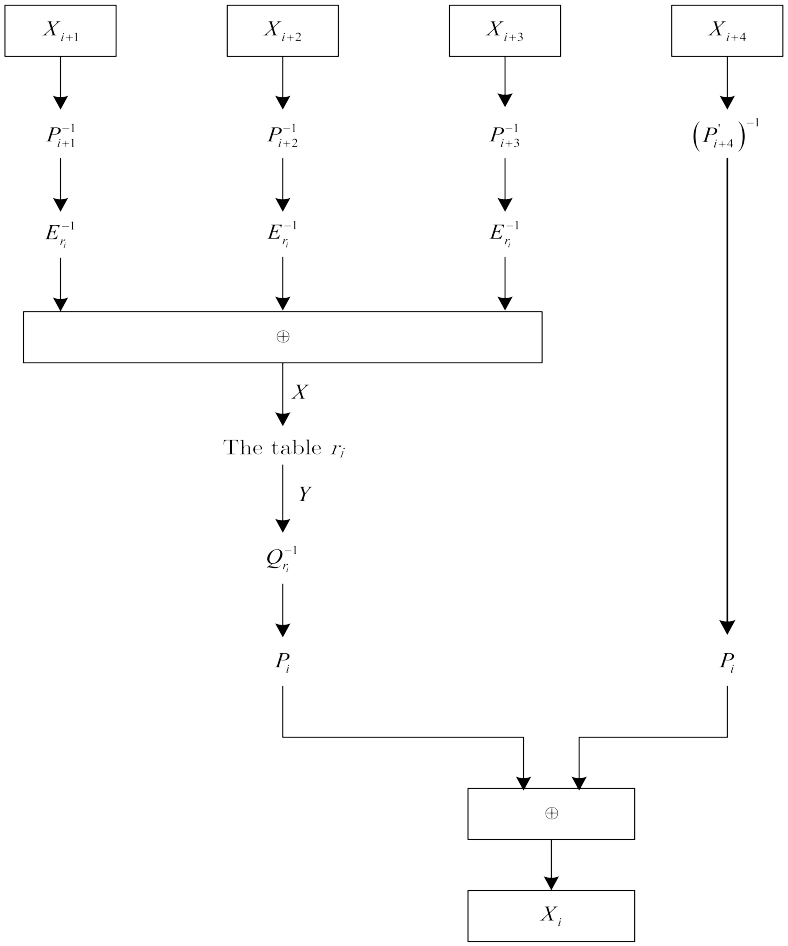

Figure 7: The workflow of decryption round-function

$$= E(x) + E(x)$$
$$= E(x)$$

Where, $\circ$ is regarded as the synthesis of reversible affine transformation operations, $\cdot$ represents matrix multiplication, E represents the unit matrix.

From the uniqueness of affine transformation, we can calculate that the inverse transformation of invertible affine transformation P is B.

$$P^{-1}(x) = (l[P])^{-1}(x) \oplus l[P]^{-1} \cdot c[P]$$

Because $l[P]$ is a match $(l[P])^{-1} = l[P]$, thus $P^{-1}(x) = (l[P])(x) \oplus l[P] \cdot c[P]$

Next, we prove the decryption's correctness in WBDL mechanism.

(1) The decryption correctness of each round.

In each round, X_{i+4} is calculated by the following formula:

$$X_{i+4} = X_i'' \oplus Y' = P_{i+4} \circ P_i^{-1}(X_i) \oplus P'_{i+4} \circ {Q_{r_i}}^{-1}(Y)$$

According to the above definition:

$$P'_{i+4}(x) = P_{i+4}(x) \oplus c'_{i+4}$$

It can be deduced :

$$X_{i+4} = P_{i+4} \circ P_i^{-1}(X_i) \oplus P'_{i+4} \circ Q_{r_i}^{-1}(Y) \oplus c'_{i+4}$$

Then:

$$\begin{aligned} X_{i+4} \oplus c'_{i+4} &= P_{i+4} \circ P_i^{-1}(X_i) \oplus P_{i+4} \circ Q_{r_i}^{-1}(Y) \\ &= P_{i+4} \cdot (P_i^{-1}(X_i) \oplus Q_{r_i}^{-1}(Y)) \end{aligned}$$

It will be calculated:

$$\begin{aligned} P_i^{-1}(X_i) \oplus Q_{r_i}^{-1}(Y) &= P_{i+4}^{-1} \circ (X_{i+4} \oplus c'_{i+4}) \\ &= P_{i+4}^{-1} \circ X_{i+4} \oplus P_{i+4}^{-1} \circ c'_{i+4} \end{aligned}$$

From the above properties, the following result should be got:

$$P_i^{-1}(X_i) \oplus Q_{r_i}^{-1}(Y) = (P'_{i+4})^{-1}(X_{i+4})$$

Through the calculation:

$$P_i^{-1}(X_i) = (P'_{i+4})^{-1}(X_{i+4}) \oplus Q_{r_i}^{-1}(Y)$$

The result can be obtained:

$$X_i = P_i \circ (P'_{i+4})^{-1}(X_{i+4}) \oplus P_i \circ Q_{r_i}^{-1}(Y)$$

The decryption correctness of each round in WBDL mechanism can is proved by the above process.

(2) The overall decryption correctness of WBDL mechanism.

The data transform in the encryption process in WBDL mechanism is as follows:

$$\begin{gathered} (X_0, X_1, X_2, X_3) \\ \downarrow \\ (X_1, X_2, X_3, X_4) \\ \downarrow \\ \ldots \\ (X_{32}, X_{33}, X_{34}, X_{35}) \\ \downarrow \\ (X_{35}, X_{34}, X_{33}, X_{32}) \end{gathered}$$

$$(X_{35}, X_{34}, X_{33}, X_{32}) = (Y_0, Y_1, Y_2, Y_3)$$

The data transform in the decryption process in WBDL mechanism is as follows:

$$(Y_0, Y_1, Y_2, Y_3) = (X_{35}, X_{34}, X_{33}, X_{32})$$

$$\begin{gathered} (X_{35}, X_{34}, X_{33}, X_{32}) \\ \downarrow \\ (X_{34}, X_{33}, X_{32}, X_{31}) \\ \downarrow \\ \ldots \\ (X_3, X_2, X_1, X_0) \\ \downarrow \\ (X_0, X_1, X_2, X_3) \end{gathered}$$

The last step in encryption and decryption process is reversed, so $D(E(X_0,X_1,X_2,X_3))$ $=(X_0,X_1,\ X_2,X_3)$. It can be seen that the WBDL mechanism is reversible, that is, the whole decryption process in WBDL mechanism is correct.

3.3 Design of WBDL dynamic white-box Library

There are n look-up tables being applied in the dynamic white-box library in WBDL mechanism.

Firstly, three initial keys with 128bit are selected randomly:
$k_0 = (MK_0, MK_1, MK_2, MK_3)$, $k_1 = (MK_4, MK_5, MK_6, MK_7)$,
$k_2 = (MK_8, MK_9, MK_{10}, MK_{11})$,

Next, these three initial keys are expanded by using the key expansion algorithm in standard SM4 algorithm to output 96 round-keys with 32bit. Then 69 round-keys are randomly selected from these 96 round-keys, of which 37 round-keys are used to generate 37 look-up tables, and these 37 round-keys are renamed as $(rk_0, rk_1, ..., rk_{36})$, another 32 round-keys are used as selection factors (SF) of look-up tables, and these 32 SFs are renamed as $(H_1, H_2, ..., H_{31})$.

Then, as shown in Figure 8, in order to form n look-up tables, we encode the original T transform after inputting $X = X'_{i+1} \oplus X'_{i+2} \oplus X'_{i+3} = (a_0, a_1, a_2, a_3)$ with 32bit and round-key $rk_h (h = 0, 1, ..., 36)$ with 32bit. In WBDL mechanism, the round-keys are combined with the S-box, S-box will also be hidden in the look-up tables. Setting up:

$$rk_h = (rk_{h,0}, rk_{h,1}, rk_{h,2}, rk_{h,3}) \quad h = 0, 1, ..., 36$$

And supposing

$$S_{hj}(x) = Sbox(x \oplus rk_{h,j}) \quad h = 0, 1, ..., 36 \quad j = 0, 1, 2, 3$$

Each S_{hj} is related to the round-keys. Because the S-box in the standard SM4 algorithm is S box is public, the cryptanalyst still could get the key information when he gets S_{hj}, so it is necessary to scramble S_{hj}, the scrambling method can be denoted by the following expression:

$$Y = Q_h(T(E_h(X) \oplus rk_h)) =$$
$$Q_h(L(S_{h0}(E_{h0}(a_0)), S_{h1}(E_{h1}(a_1)), S_{h2}(E_{h2}(a_2)), S_{h3}(E_{h3}(a_3))))$$

Where, L represents the linear transformation of T transformation in SM4 algorithm, which is a 32×32 invertible matrix on $GF(2)$.$E_{h0}, E_{h1}, E_{h2}, E_{h3}$ are 8bit to 8bit invertible affine transformations on $GF(2)$, which is used as the input encoding of the look-up tables. And Q_h is a 32bit to 32bit invertible affine transformations on $GF(2)$, which is encoded as the output of the look-up tables.

By using the above round-keys with n=37, we can generate the look-up tables with n=37. These look-up tables are all 32bit to 32bit transformations, and the size of these look-up tables is $2^{32} \times 37$ .Such large look-up tables are not suitable for practical white-box cipher application; therefore, it can be split that one big look-up table into four small look-up tables as following.

Recording the value of $X = (a_0, a_1, a_2, a_3)$ as $(z_{h0}, z_{h1}, z_{h2}z_{h3})$ after the transformations of E_{hj} and S_{hj} ,thus, a_j and $z_{hj} (j = 0, 1, 2, 3)$ are one-to-one correspondence.

$$Y = Q_h \circ L \cdot \begin{bmatrix} z_{h0} \\ z_{h1} \\ z_{h2} \\ z_{h3} \end{bmatrix} = l[Q_h] \cdot \left(L \cdot \begin{bmatrix} z_{h0} \\ z_{h1} \\ z_{h2} \\ z_{h3} \end{bmatrix} \right) \oplus c[Q_h] = (R_{h0}, R_{h1}, R_{h2}, R_{h3}) \cdot \begin{bmatrix} z_{h0} \\ z_{h1} \\ z_{h2} \\ z_{h3} \end{bmatrix} \oplus c[Q_h]$$

$$= (R_{h0} \cdot z_{h0}) \oplus (R_{h1} \cdot z_{h1}) \oplus (R_{h2} \cdot z_{h2}) \oplus (R_{h3} \cdot z_{h3} \oplus c[Q_h])$$
$$= v_{h0}, v_{h1}, v_{h2}, v_{h3}$$

Where, $R_{hj} (j = 0, 1, 2, 3)$ is a 32×8 invertible matrix on $GF(2)$, it can be seen from the above formulas, v_{hj} and $z_{hj} (j = 0, 1, 2, 3)$ are also one-to-one correspondence, further

a_j and $v_{hj}(j = 0, 1, 2, 3)$ are one-to-one correspondence, thus the whole transformation can be divided into four 8bit to 32bit look-up tables.

Therefore, in actual application, the process of transforming the 32bit X from a look-up table to the 32bit Y can be converted into the process of four look-up tables and four XOR operations.

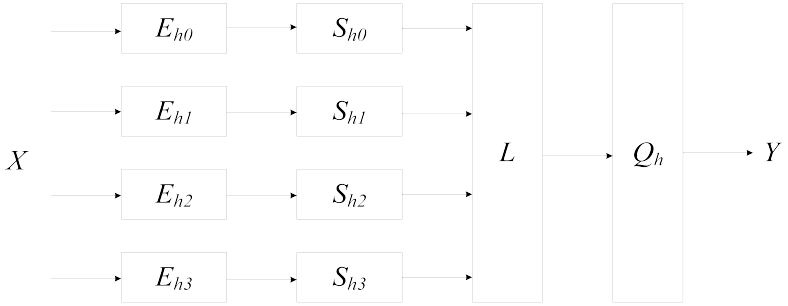

Figure 8: Generation process of dynamic white box library

4 Comprehensive analysis

4.1 Security Analysis

(1) Differential Cryptanalysis

Differential attack is to analyze the key with the greatest possibility by analyzing and comparing whether the difference of the corresponding ciphertext has increased after the specific plaintext encryption and transformation.One of the performance indicators of the S-box, the differential uniformity, can specifically determine the strength of a cryptographic algorithm against differential attacks.

$$\delta_s = \frac{1}{2^n} \max_{\substack{\alpha \in F_2^n \\ \alpha \neq 0}} \max_{\beta \in F_2^m} \{X \in F_2^n : S(X \oplus \alpha) - S(X) = \beta\} |$$

Where δ_s is called the differential uniformity of the S box. In general, the smaller the differential uniformity of the S box in a cryptographic algorithm, the stronger the ability to resist differential attacks. If the value is 0, the S box is called a differential active box. It can be found that the maximum differential probability of the S box of the SM4 algorithm is2^{-6} , and the maximum differential probability of the S box of the scheme in this paper is also 2^{-6}.

We analyze the differential properties of each round function of the WBDL mechanism. Given the input difference a and the output difference b, the differential probability of the function f is defined as [42].

$$DP(a, b) = \{(v, u)|u \oplus v = a \; and \; f(v) \oplus f(u) = b\}$$

Where $v, u \in \{0, 1\}^{32}$.The maximum differential probability(MDP) bounds are as follows:

$$\Pr(\frac{\boldsymbol{n}ln2}{2^{n-1}ln\boldsymbol{n}} \leq \text{MDP} < \frac{n}{2^{n-1}}) \approx 1$$

Therefore, the maximum differential probability(MDP) of each round of the WBDL mechanism is $2^{-26} = 32/2^{31}$

(2) Linear Cryptanalysis.

The basic method of linear attack is that the attacker knows the plaintext, and the existing expression $A[i, j, \cdots, k] = A[i] \oplus A[j] \oplus \cdots \oplus A[k]$. It is assumed that both the plaintext packet length and the cipher text packet length are n bits, and the key packet length is m bits. Remember that the plaintext group is $P[1], P[2], \cdots, P[n]$, the cipher text group is $C[1], C[2] \cdots C[n]$, , and the key group is $K[1], K[2], \cdots, K[m]$. Then the goal of linear cryptanalysis is to find effective linear equations of the following form:

$$P[i_1, i_2, \cdots, i_a] \oplus C[j_1, j_2, \cdots, j_b] = K[k_1, k_2, \cdots k_c], \text{where } 1 \le a \le n, 1 \le b \le n, 1 \le c \le m$$

If the probability of satisfying an equation p is the largest, the equation is said to be the most efficient linear approximation. After obtaining some effective linear approximations, the maximum likelihood law method is used to improve the attack efficiency. Let N denote the plaintext number, and T be the plaintext number that makes the left side of the equation 0. If $T > N/2$, then:

$$K[k_1, k_2 \cdot \cdot, k_c] = \begin{cases} 0\ p > \frac{1}{2} \\ 1\ p < \frac{1}{2} \end{cases} \quad if \ \ \text{T<N/2 , then let K}[k_1, k_2, \cdots k_c] = \begin{cases} 0\ p > \frac{1}{2} \\ 1\ p < \frac{1}{2} \end{cases}$$

Thus we can get a linear equation about the key bits. Repeat the above process for different plaintext ciphertext pairs to obtain a set of linear equations about the key, thereby determining the key bit.

The nonlinearity of the boolean function $f(x)$ can be used to express the ability to resist linear attacks, and it is related to the core device of the block cipher algorithm, the S box. The greater the non-linearity of the S-box, the stronger the resistance to linear attacks and the greater the security strength of the entire algorithm.

The nonlinearity of the S box is defined as: let $S(X) = (f_1(x), f_2(x), \cdots, f_m(x)) : F_2^n \to F_2^m$ be a multi-output function, then

$$N_s = \min_{\substack{l \in L_n \\ 0 \ne u \in F_2^m \\ l \in L_n \\ 0 \ne u \in F_2}} (u \cdot S(X), L(X))$$

is the nonlinearity of $S(X)$. Where L_n represents the entire set of n-ary affine functions, and $d_H(f, l)$represents the hamming distance between f and l. The non-linearity of the S-box is the arbitrary linear combination of output bits and the minimum Hamming distance of all affine functions related to the input. It can be found that the S-box nonlinearity of the SM4 algorithm is 112, and the S-box nonlinearity of the scheme in this paper is also 112.

Now we analyze the linear nature of the 32bit to 32bit permutation of the round function of the WBDL mechanism. Given the input mask and the output mask $\beta, \alpha, \beta \in \{0,1\}^{32}$, the function f: $\{0,1\}^{32} \to \{0,1\}^{32}$linear approximation (α, β) related definitions from [42].

$$Cor = 2^{-n_{in}\left\{\boldsymbol{x} \in \{0,1\}^{-n_{in}} | \alpha \cdot x \oplus \beta \cdot f(x) = 0\right\} - \left\{x \in \{0,1\}^{-n_{in}} | \alpha \cdot x \oplus \beta \cdot f(x) = 1\right\}} .$$

The linear probability LP of (α, β) is defined as Cor. For fixed-key block ciphers, the maximum linear probability MLP normal distribution is average $\approx (1.38 \cdot 2n - ln(1.38 \cdot 2n) + 1) \cdot 2^{-n}$and standard deviation $\approx 2.6 \times 2^{-n}$.

Then the maximum linear probability MLP of each round function of the WBDL mechanism is $2^{-25.61} (= 84 \cdot 2^{-32})$.

(3) White-box diversity

White-box diversity is used to measure the number of all possible distinct constructions with the look-up tables, which depends on the number of input/output scrambling encoding

and the number of scrambling encoding's steps. If the look-up table has n steps, and each step has c_i kinds of choices, then the white-box diversity of this look-up table is $\prod c_i$. Because different input/output scrambling encoding and secret keys may generate the same look-up table, the value of the white-box diversity could be much larger than the actual number of look-up tables. Theoretically, the larger the value of the white-box diversity, the harder it is for the cryptanalyst to analyze the input/output scrambling encoding and the key information hidden in the look-up tables[7].

The number $N_m(n)$ of m-order invertible matrices on Z_n can be calculated by the following theorem.

Theorem[46]: Let $n \geq 2$ be an integer,$n = p_1^{r_1} p_2^{r_2} ... p_s^{r_s}$ be the approximate factor of n, and $r_i \geq 1$ $p_1, p_2, ..., p_s$ be the mutual prime number, then $N_m(n) = \prod_{i=1}^{z} \prod_{j=0}^{m-1} (p_i^m - p_i^j) p_i^{(r_i-1)m^2}$. Therefore, the number of m-order invertible matrices on $GF(2)$ is:$N_m(n) = \prod_{j=0}^{m-1} (2^m - 2^j)$

According to the above formula, we can calculate that, the number of 8-order invertible matrices on $GF(2)$ is about 2^{62}, the number of 16-order invertible matrices is about 2^{254}, and the number of 32-order invertible matrices is about 2^{922}.

In WBDL, the estimated value of white-box diversity is as shown in Table 3.

Table 3: White-box diversity in WBDL

Part	White-box diversity
Part 1	$(2^{922} \times 2^{32})^3 \times (2^{62} \times 2^8)^3 = 2^{3142}$
Part 2	$37 \times (2^{62} \times 2^8)^4 \times 2^{32} \times (2^{922} \times 2^{32}) = 37 \times 2^{1266}$
Part 3	$(2^{922} \times 2^{32})^3 \times 2^{32} = 2^{2894}$

(4) White-box ambiguity

White-box ambiguity is a measuring indicator that how many different constructors can it produce the same look-up table in one white-box algorithm implementation. The larger of white-box ambiguity's value, the more input-output encoding methods in the look-up table, the more difficult for inferring the hidden codes and keys[7].

In WBDL, white-box ambiguity is estimated as shown in Table 4.

Table 4: White-box ambiguity in WBDL

Part	White-box ambiguity
Part 1	$(2^{62} \times 2^8)^4 = 2^{280}$
Part 2	$37 \times (2^{62} \times 2^8)^4 \times 2^{32} = 37 \times 2^{312}$
Part 3	$(2^{922} \times 2^{32}) \times 2^{32} = 2^{986}$

Based on the analysis, WBDL, the scheme in [7] and the scheme in [47] can resist the BGE attack. From Table 5, the white-box diversity and the white-box ambiguity in WBDL are better than that of in [7] and [47].That is, the WBDLmechanismhas better security.

Table 5: Comparison of white-box diversity/ambiguity

Scheme	part	White box diversity	White box ambiguity
[7]	part 1	2^{3142}	2^{280}
[47]	part 1	2^{3402}	2^{540}
WBDL	part 1	2^{3142}	2^{280}
[7]	part 2	2^{1266}	2^{312}
[47]	part 2	2^{1256}	2^{572}
WBDL	part 2	37×2^{1266}	37×2^{312}
[7]	part 3	2^{2894}	2^{986}
[47]	part 3	2^{2894}	2^{986}
WBDL	part 3	2^{2894}	2^{986}

(5)Resistance to BGE attack

The BGE attack is an attack method proposed by Billet, Gilbert, and Ech-Chatbi, its can obtain key within a time complexity of $O(2^{30})$.

The main idea of BGE attack is as follows:

Due to the local security and diversity of the white-box design, it is difficult to obtain the key information by attacking the separate look-up table. However, by combining multiple look-up tables, the complete AES cryptography algorithm can be obtained, thus that the look-up tables corresponding to four input blocks and four output blocks in each round of operation can be obtained. This is due to the reciprocal relationship between the output scrambling code and the input scrambling code.

If r is used to represent the transformation of the white-box AES r-th wheel, Inr denotes the input scrambling code of the r-th wheel, and $Outr$ represents the output scrambling code of the r-th wheel, then $Out^r = (In^{r+1})^{-1}$, thus the white-box AES round-up look-up table is combined in together, the internal encoding will be offset, leaving only the effect of the external encoding, and then by analyzing the algebraic structure that makes up the look-up table, the key information can be obtained. Although the BGE attack method is proposed for the white-box AES designed by Chow *et al.*, the nature of the white-box algorithm based on the look-up table method to construct the block cipher algorithms is similar to the construction method, thus we design the WBDL white-box based on the look-up table. The algorithm should also consider whether it can resist BGE attacks when conducting security assessments.

One of the most important reasons why the BGE attack method can break Chow's white box AES is that its output scrambling code for each round and the input scrambling code for the next round are reciprocal, and can be combined when adjacent lookup tables are combined. Cancel each other out.In the WBDL mechanism,$Out^r \neq (In^{r+1})^{-1}$, there is a constant difference between them, but the cryptanalyst cannot know it, nor can it be obtained from a lookup table or affine transformation.

In the WBDL mechanism,$P_{i+j}(x) = A_{i+j}(x) \oplus a_{i+j}, P'_{i+4}(x) = P_{i+4}(x) \oplus c'_{i+4}$. Combining the lookup table in the second part of the WBDL mechanism, part of the affine transformation in part three, and part of the affine transformation in the first part of the next round, we can get: Q_i and its inverse completely cancel, P^{-1}_{i+4} and P'_{i+4} cancel There is still a constant left $A^{-1}_{i+4} \cdot c'_{i+4}$. This constant is randomly selected when we select the scrambling parameter, and the cryptanalyst cannot know it, and thus cannot calculate the

key information hidden in the lookup table. Therefore, the WBDL mechanism can resist BGE attacks.

(6) Anti side channel energy attack method for SM4 cipher round function output

Side-channel attacks such as SCA attacks and DCA attacks usually focus and target cryptographic implementations. According to the analysis of literature [16], if the white box algorithm has only internal encoding , it is not enough to resist attacks like SCA and DCA. For WBDL algorithm, in order to ensure the integrity of encoding and decoding, we introduce the concept of external encoding, that is, the input encoding added before the first round of input and the output encoding added after the last round. The external coding not only increases white box diversity and white box ambiguity, but also successfully resists SCA attacks and DCA attacks when the internal coding successfully resists BGE attacks.

In 2015, Zhibo Du *et al.* designed a side channel energy attack method for the output of SM4 cryptographic wheel function. They chose the plaintext or ciphertext input, so one byte of $(X_{i+1} \oplus X_{i+2} \oplus X_{i+3})$ is a random number and the other bytes are the same fixed number. At the outset, they first attacked the bytes of the wheel key and fixed numbers in the linear transformation, using the side channel energy attack method. Afterward, they attacked all other fixed numbers in the linear transformation, and performed corresponding operations on the attacked data to obtain the round-keys rk_i , where $i = 0, 1, 2, 3$. Finally, according to the first four round-key, rk_0 , rk_1,rk_2 andrk_3 ,the initial key could be calculated reversely through the key expansion algorithm.

The WBDL scheme uses the obfuscation scrambling code in the form of reversible affine transformation to scrambling the input of $X_{i+1} \oplus X_{i+2} \oplus X_{i+3}$ into $X'_{i+1} \oplus X'_{i+2} \oplus X'_{i+3} = E_{r_i}^{-1} \circ P_{i+1}^{-1}(X_{i+1}) \oplus E_{r_i}^{-1} \circ P_{i+2}^{-1}(X_{i+2}) \oplus E_{r_i}^{-1} \circ P_{i+3}^{-1}(X_{i+3})$. This side channel attack can not attack the bytes of wheel key and the fixed number in linear transformation, so the WBDL scheme can resist this side channel energy attack against the output of SM4 cryptographic wheel function.

Table 6: Comparison of resistance against existing attacks.

Implementation	BGE attack	MGH attack	De Mulder et al. attack	Lin–Lai attack	SCA attack
The Xiao–Lai AES[7]	Y	/	N	N	/
HadThe Xiao–Lai SM4[7]	Y	/	N	N	/
Karroumi's AES[11]	Y	N	/	/	/
Luo-Lai-You AES[13]	Y	/	Y	/	/
Bai Kunpeng SM4[16]	Y	Y	Y	Y	N
Tao Xu AES[18]	Y	Y	Y	Y	/
Our WBDL	Y	Y	Y	Y	Y

It is shown in Table 6 that compares security property of WBDL with other white-box

implementations,where "Y" means that the implementation can resist the attack, and "N" means that the implementation cannot resist the attack. "/" mean that no previous work shows whether the implementation can resist the attack or not.

4.2 Analysis of space occupation

WBDL is a white-box algorithm based on look-up tables, which can be implemented by affine transformation and look-up tables. Therefore this mechanism will occupy additional space which could be measured by the times of look-up tables, XOR operations and affine transformations. There are three parts in each round of process. The first part includes three affine transformations and two XOR operations. The second part includes four look-up tables and five XOR operations. The third part includes two affine transformations and one XOR operations. The whole WBDL mechanism needs 128 look-up tables, 160 affine transformations with 32bit to 32bit and 256 XOR operations.

The analysis of space occupation with 32 rounds in WBDL mechanism is shown in Table 7.

Table 7: Space occupancy in WBDL

Part	Space occupancy (bit)
Part 1	$34 \times 37 \times (32 \times 32 + 32) = 1328448$
Part 2	$37 \times 4 \times (32 \times 2^8) = 1212416$
Part 3	$32 \times (32 \times 32 + 32) + 32 \times 37 \times (32 \times 32 + 32) = 1284096$

Therefore, the entire space occupied in the WBDL mechanism is:

$$1328448 + 1212416 + 1284096 = 3824960bit = 478120B = 466.914KB$$

Table 8: Execution components comparison

Scheme	Occupancy space	Number of table look-ups	Number of XORs	Number of affine transformation
SM4-128	544B	0	256	64
AES-128	4352B	0	300	0
[1]	752KB	3104	0	0
[7]	148.625KB	128	192	160
[47]	16.012MB	64	128	160
WBDL	466.914KB	128	256	160

By analyzing the data in Table 8, in terms of the occupied space, we can see that, scheme [1] is 752KB, scheme [7] is 148.625KB, scheme [47] is 16.012MB, and WBDL mechanism is only 466.914KB, which is 62.29% of [1], 3.14 times of [7], only 2.85% of [47]. In terms of the number of look-up tables, we can see that, there are only 128 times in WBDL, which is 2.06% of [1], the same as [7]. Moreover, there are 256 times XOR operations and 160 times affine transformations in WBDL.

Generally compared with other related algorithms, WBDL mechanism has a slight reduction in efficiency, resource utilization and other practical performance; however, it can meet the practical application in view of its better security and convenience.

4.3 Efficiency analysis

Many input/output encoding, look-up tables and invertible affine transformations have been used in WBDL mechanism. The execution efficiency of these operations in the program has an important impact on the running speed of the entire scheme. The actual test environment is Intel Core i7 processor, 8GB memory, Windows 7 64bit operating system, and the execution software is Microsoft Visio studio 2010.

As mentioned in Section 2.4, this paper uses the Hadamard-type MDS matrix of the coincidence. In this paper, the $4 * 4$ matrix $[x^8 + x^7 + x^6 + x + 1]$ designed by literature [42], the irreducible polynomial is, and the matrix is $\mathrm{Had}(x^{-2} + x^2 + x + 1)$.

In the design of the diffusion layer of the block cipher algorithm, due to the overall requirements of the algorithm, engineering applicability and other requirements, it is usually desirable that the diffusion layer of the algorithm has some good cryptographic properties, such as the smaller matrix element Hamming weight sum, matrix The match. Our selection criteria for effective MDS matrices are different from the widely studied lightweight hardware implementation field: in software, the cost of changing at any position is high, which means that the matrix has a small coefficient, but a high theoretical XOR count can be As a result, SIMD implementation is more efficient. It can be seen from the property 5 in section 2.4 that the first row matrix elements of the Hadamard matrix on the finite field must be different from each other. Therefore, there is at most one "01" in each row of matrix elements, which limits the implementation of the algorithm on various platforms. Compared with cyclic matrices, Hadamard matrices are at a disadvantage in terms of implementation performance, but Hadamard MDSs can be coincident. We put 34 matching Hadamard MDS matrices in the appendix.

We test the encryption and decryption speed for the WBDL mechanism, the results are shown in Figure 9.

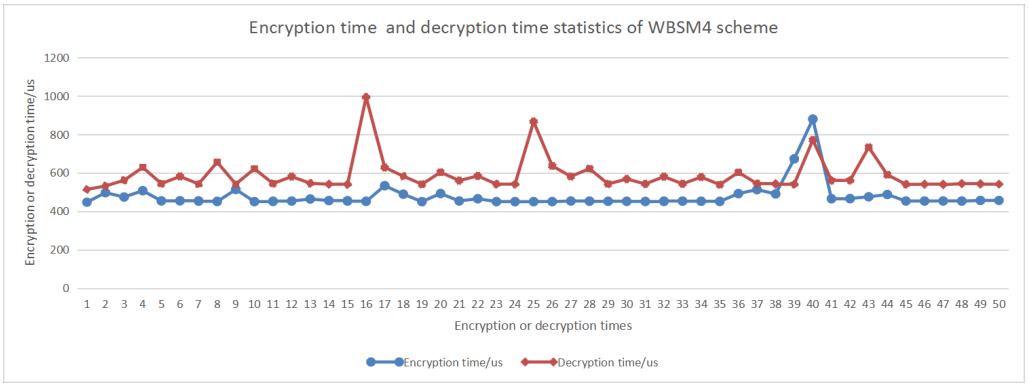

Figure 9: Speed test of encryption and decryption

It can be seen that the average encryption time of WBDL mechanism is 476.778us for 128bit data, and the average decryption time is 586.414us for 128bit ciphertext. After calculation, the average encryption rate of WBDL mechanism is 0.268×10^{-3} Gbps, and the average decryption rate is 0.218×10^{-3} Gbps.

In order to illustrate the working performance of WBDL further, the comparison of encryption speed with other research achievements can be presented as Table 9. Although the test platforms are different, it is obvious that WBDL has very fast encryption speed.

Table 9: Comparison of encryption speed

Scheme	Processor	Memory	Operating system	Speed of encryption
WBDL	core i7	8GB	windows 7	0.268×10^{-3}Gbps
[47]	core i3	4GB	windows 7	3.475×10^{-6}Gbps

5 Conclusion

We propose a novel effective dynamic white-box block cipher scheme named WBCD. Through the key expansion function in the block cipher algorithm, multiple initial keys selected are converted into multiple round-keys. Then, a number of white-box look-up tables will be generated by the round-keys which are randomly selected by a specific rule; hence, the dynamic encryption and decryption white-box library is built. In the WBCD scheme, the selection of the look-up tables in each round depends on the intermediate variables and the selection factors. This scheme can be applied to many kinds of block ciphers, such as SM4 algorithm.

In order to present the effectiveness of WBCD, the white-box SM4 mechanism (WBDL) based on the dynamic white-box library is designed, which adopt MDS matrix. In the WBDL mechanism, each round of SM4 algorithm is divided into three parts, and the crucial second part is transformed into the dynamic look-up tables, which are confused by the reversible affine transform as input scrambling code and output scrambling code. Then the correctness of WBDL mechanism is also proved.

WBDL mechanism has higher security. On the one hand, WBDL mechanism inherits the advantages of block cipher, which can resist differential attacks, such as linear attack, BGE attack and side channel energy attack against SM4.On the other hand, the white-box diversity, white-box ambiguity and other indicators are not lower than the previous mechanism.

Through the analysis of space occupation and working efficiency, the WBDL mechanism has better running speed compared with the standard SM4 algorithm and other related achievements. Much more specifically, the WBDL mechanism is more practical, which can be used in mobile communication security and digital payment security in the future.

References

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

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

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

[4] Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi. Cryptanalysis of a white box AES implementation. In Helena Handschuh and Anwar Hasan, editors, SAC 2004, volume 3357 of LNCS, pages 227–240. Springer, Heidelberg, August 2004.

[5] Matthias Jacob, Dan Boneh, and Edward Felten. Attacking an obfuscated cipher by injecting faults. In Joan Feigenbaum, editors, Digital Rights Management (DRM 2002). volume 2696 of LNCS, pages 16-31. Springer, Heidelberg, 2004.

[6] Bringer Julien, Chabanne Herve, and Dottax Emmanuelle. White box cryptography: another attempt. In IACR Cryptology ePrint Archive, volume 2006, pages 468. Research Gate, January 2006.

[7] Yaying Xiao and Xuejia Lai. A secure implementation of white-box AES. In 2009 2nd International Conference on Computer Science and its Applications, pages 1–6, Dec 2009.

[8] Yoni De Mulder, Brecht Wyseur, Bart Preneel. Cryptanalysis of a perturbated white-box AES implementation. In Gong G., Gupta K.C., editors, Progress in Cryptology - INDOCRYPT 2010, volume 6498 of LNCS, pages 292-310. Springer, Berlin, 2010.

[9] Yoni De Mulder, Peter Roelse, and Bart Preneel. Cryptanalysis of the Xiao Lai white-box AES implementation. In Lars R. Knudsen and Huapeng Wu, editors, SAC 2012, volume 7707 of LNCS, pages 34–49. Springer, Heidelberg, August 2013.

[10] Jong-Yeon Park, Ji-Sun Choi, and Okyeon Yi. Methods for practical whitebox cryptography. In International Conference on Information and Communication Technology Convergence (ICTC), pages 474-479. IEEE, Jeju, November 2010.

[11] Mohamed Karroumi. Protecting white-box AES with dual ciphers. In Kyung Hyune Rhee and DaeHun Nyang, editors, ICISC 10, volume 6829 of LNCS, pages 278–291. Springer, Heidelberg, December 2011.

[12] Tancrède Lepoint, Matthieu Rivain, Yoni De Mulder, Peter Roelse, and Bart Preneel. Two attacks on a white-box AES implementation. In Tanja Lange, Kristin Lauter, and Petr Lisonek, editors, SAC 2013, volume 8282 of LNCS, pages 265–285. Springer, Heidelberg, August 2014.

[13] Rui Luo. A white box AES implementation based on look-up table with nonlinear obfuscation.In Shanghai Jiao Tong University, Shanghai, 2015.

[14] Chung Hun Baek, Jung Hee Cheon, Hyunsook Hong. White-box AES implementation revisited. In Journal of Communications and Networks, volume 18(3), pages 273-287. IEEE, June 2016.

[15] Pascal Sasdrich, Amir Moradi, Tim Güneysu. White-box cryptography in the gray box. In Peyrin T, editors, FSE 2016. volume 9783 of LNCS, pages 185-203. Springer, Berlin, July 2016.

[16] Kunpeng Bai, and Chuankun Wu. A secure white-box SM4 implementation. In Security and Communication Networks, volume 9, pages 996-1006. Research Gate, January 2016

[17] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, and Martin Bjerregaard Jepsen. Analysis of software countermeasures for whitebox encryption. In Cryptology ePrint Archive, Report 2017/183, 2017.

[18] Tao Xu, Chuankun Wu, Feng Liu, and Ruoxin Zhao. Protecting white-box cryptographic implementations with obfuscated round boundaries. In Science China Information Sciences, volume 61, 039103. Springer, 2018.

[19] Tao Xu, Feng Liu, and Chuankun Wu. A white-box AES-like implementation based on key-dependent substitution-linear transformations. In Multimedia Tools and Applications, volume 77, pages 1811–18137. Springer, March 2017.

[20] Jie Zhou, Jian Bai and Hongzhang An. A white box software implementation method of quotient secret SM2 encryption algorithm based on remainder system. The 30th Research Institute of China Electronic Technology Group Corporation. China, 2017.

[21] Dawu Gu, Lei Wang, Ning Ding and Haining Lu. SM2 white box password implementation method. In Springer Berlin Heidelberg, China, 2018.

[22] Debiao He, Yudi Zhang, Yubo Zhang and Biwen Chen. A white box implementation method and device for SM9 digital signature. In Springer Berlin Heidelberg, China, 2018.

[23] Estuardo Alpirez Bock, Joppe W. Bos, Chris Brzuska, Charles Hubain, Wil Michiels, Cristofaro Mune, Eloi Sanfelix Gonzalez, Philippe Teuwen, and Alexander Treff. White-box cryptography: don't forget about grey-box attacks. In Journal of Cryptology, volume 32(4), pages 1095–1143. Springer, Heidelberg, Oct 2019.

[24] Brice Minaud, Patrick Derbez, Pierre-Alain Fouque, and Pierre Karpman. Key-recovery attacks on ASASA. In Tetsu Iwata and Cheon J, editors, Advances in Cryptology – ASIACRYPT 2015. ASIACRYPT 2015. Lecture Notes in Computer Science, volume 9453 of LNCS, pages 3-27. Springer, Heidelberg, December 2015.

[25] Itai Dinur, Orr Dunkelman, Thorsten Kranz and Gregor Leander. Decomposing the ASASA block cipher construction. In IACR Cryptology ePrint Archive, volume 2015, pages 507. 2015. https://eprint.iacr.org/2015/507

[26] Alex Biryukov and Dmitry Khovratovich. Decomposition attack on SASASASAS. In IACR Cryptology ePrint Archive, volume 2015, pages 646. 2015.https://eprint.iacr.org/2015/646.

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

[28] Yang Shi, Xiaoping Wang, and Hongfei Fan. Light-weight white-box encryption scheme with random padding for wearable consumeer electronic devices. In IEEE Transactions on Consumer Electronics, volume 63, pages=44-52. IEEE, February 2017.

[29] Jihoon Cho, Kyu Young Choi, Orr Dunkelman, Nathan Keller, Dukjae Moon, and Aviya Vaidberg. Hybrid WBC: secure and efficient white-box encryption schemes. In Foresti S and Persiano G, editors, CANS 2016. volume 10052 of LNCS, pages 749-754. Springer, Milan, October 2016.

[30] Jihoon Cho, Kyu Young Choi, Itai Dinur, Orr Dunkelman, Nathan Keller, Dukjae Moon, and Aviya Veidberg. WEM: a new family of white-box block ciphers based on the even-mansour construction. In Handschuh H, editors, Topics in Cryptology – CT-RSA 2017. CT-RSA 2017, volume 10159 of LNCS, pages 293-308. Springer, California, January 2017.

[31] Tingting Lin, Xuejia Lai, Weijia Xue and Yin Jia. A new feistel-type white-box encryption scheme. In J. Comput. Sci. Technol. volume 32, pages 386–395. Springer, March 2017. https://doi.org/10.1007/s11390-017-1727-x

[32] Yang Shi, Wujing Wei, and Zongjian He. A lightweight white-Box symmetric encryption algorithm against node capture for WSNs. In Leonhard M. Reindl, editors, Sensors. volume 15(5), pages 11928–11952. NCBI, Basel, May 2015.

[33] Lu Zhou, Chun-hu Su, Ya-min Wen, Wei-jie Li, and Zheng Gong. Towards practical white-box lightweight block cipher implementations for IoTs. In Future Generation Computer Systems, volume 86, pages 507-514. ScienceDirect, September 2018.

[34] Öznur Şengel, Muhammed Ali Aydin, and Ahmet Sertbaş. A Survey on white box cryptography model for mobile payment systems. In Boyaci A, Ekti A, Aydin M, and Yarkan S, editors, International Telecommunications Conference, volume 504 of LNEE, pages 215-225. Springer, Singapore, 2019.

[35] Kang Ju-Sung, Hong Seokhie, Lee Sangjin, Yi Okyeon, Park Choonsik and Lim Jongin. Practical and provable security against differential and linear cryptanalysisfor substitution-permutation networks. In Etri Journal, volume 23, pages 158-167, 2001.

[36] Xiao Lu and Heys Howard. Hardware design and analysis of block cipher omponents. In: Lee P.J., Lim C.H., editors, Information Security and Cryptology — ICISC 2002. ICISC 2002. Lecture Notes in Computer Science, , volume 2587. pages 164-181. Springer, Berlin, Heidelberg,2003

[37] YOUSSEF AMister S. and Tavares Stafford On the design of linear transformations for substitution permutation encryption networks In Workshop on Selected Areas in Cryptography-SAC'97, pages 40-48. Workshop record, Ottawa1997:

[38] Blomer Johannes, Kalfane Malik, Karp Richard, Karpinski Marek, Luby Michael and Zuckerman Davi An XOR-based erasure-resilient coding schemeIn International Computer Science Institute, Technical Report TR-95-048, 1995.

[39] Ting Cui and Chenhui Jin. Construction of MDS matrix of coincidence cauchy-hadamard type In Journal of Electronics and Information Technologyvolume 32, pages 500-503, 2010

[40] Sumanta Sarkar and Habeeb Syed. Lightweight diffusion layer: importance of toeplitz matrices. In IACR Transactions on Symmetric Cryptology, volume 2016,pages 95–113, Dagstuhl Seminars, Saarland, 2016.

[41] Jérémy Jean, Thomas Peyrin, Siang Meng Sim and Jade Tourteaux. Optimizing implementations of lightweight building blocks. In IACR Transactions on Symmetric Cryptol, volume 2017(4), pages 130-168, Dagstuhl Seminars, Saarland, 2017.

[42] Beierle Christof, Kranz Thorsten, Leander Gregor. Lightweight multiplication in GF(2n) with applications to mds matrices. In: Robshaw M., Katz J., editors, Advances in Cryptology – CRYPTO 2016. CRYPTO 2016. Lecture Notes in Computer Science, volume 9814, pages 625-653, Springer, Berlin, 2016.

[43] Zhigang Kan and Biao Chen and Quanzhou Wang and Zuohua Lu and Ning Fang. The method and device of encryption and decryption using white box library file and white box key file. China, 2018.

[44] Lihui Liu, Tao Pei and Zuping Zhang. Research on generation and application of MDS matrix. In Modern communication technology, volume 1, pages 26-29, 2014.

[45] Lihui Liu, Linjie Xu, Zuping Zhang and Yanping Li. Investigate for MDS matrix of hadamard type on finite fields. In Ship Electronic Engineering, volume 034, pages 41-45, 2014.

[46] Sheng Yuan Zhang. Count of m-order invertible matrix on Zn. In Journal of Fujian Normal University, volume 01, pages 18-20. Fujian, 1999.

[47] Pei Shang. Design and implementation of white box cipher algorithm for SMS4 algorithm. In University of Electronic Science and Technology of China. Chengdu, 2016.

[48] Hamilton E. Link, and William D. Neumann. Clarifying obfuscation: improving the security of white-box encoding, cryptology eprint archive. In International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II, volume 1, pages 679-684 Vol. 1. IEEE, Nevada, April 2005.

[49] Brecht Wyseur, Wil Michiels, Paul Gorissen, Bart Preneel. Cryptanalysis of white box DES implementations. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, SAC 2007, volume 4876 of LNCS, pages 264-277. Springer, Heidelberg, August 2007.

[50] Yoni De Mulder, Peter Roelse, Bart Preneel. Cryptanalysis of the Xiao – Lai white-box AES implementation. In Knudsen L.R., Wu H. editors, SAC 2012, volume 7707 of LNCS, pages 34-49. Springer, Berlin, 2012.

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

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

[53] Rui Luo, Xuejia Lai, Rong You. A new attempt of white-box AES implementation. In Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pages 423-429. IEEE, Wuhan, 2014.

[54] Wenlun Pan, Tihong Qin, Yin Jia, Liting Zhang. Analysis of two SM4 white box schemes. In Journal of Cryptologic Research, volume 5, pages 651-670. June 2018.