Cybersecurity

RESEARCH

Open Access

WAS: improved white-box cryptographic algorithm over AS iteration

Yatao Yang[1,2*†], Yuying Zhai[2], Hui Dong[1†] and Yanshuo Zhang[1]

Abstract

The attacker in white-box model has full access to software implementation of a cryptographic algorithm and full control over its execution environment. In order to solve the issues of high storage cost and inadequate security about most current white-box cryptographic schemes, WAS, an improved white-box cryptographic algorithm over AS iteration is proposed. This scheme utilizes the AS iterative structure to construct a lookup table with a five-layer ASASA structure, and the maximum distance separable matrix is used as a linear layer to achieve complete diffusion in a small number of rounds. Attackers can be prevented from recovering the key under black-box model. The length of nonlinear layer S and affine layer A in lookup table is 16 bits, which effectively avoids decomposition attack against the ASASA structure and makes the algorithm possess anti-key extraction security under the white-box model, while WAS possesses weak white-box (32 KB, 112)-space hardness to satisfy anti-code lifting security. WAS has provable security and better storage cost than existing schemes, with the same anti-key extraction security and anti-code lifting security, only 128 KB of memory space is required in WAS, which is only 14% of SPACE-16 algorithm and 33% of Yoroi-16 algorithm.

Keywords White-box cryptography, Block cipher, Substitution permutation network structure, Anti-key extraction, Anti-code lifting

Introduction

Modern cryptography is widely used in symmetric cryptographic schemes for data encryption and asymmetric cryptographic schemes for digital signatures and key establishment, etc. Most of the above cryptographic schemes are analyzed for security under the black-box attack model, i.e., assuming that the communication endpoint is trusted and the internal state and algorithmic laws of the algorithm are unknown to the adversary. The attacker in the black-box model can only access the input and output of the algorithm and does not know the detailed information generated during the execution of the encryption and decryption algorithm, but the black-box model is vulnerable to attacks (Kocher 1109). In the gray-box attack model, the attacker can not only access information through untrusted channels, but also side channel information such as electromagnetic radiation, current flow, and running time during the encryption and decryption of the algorithm. By analyzing the side channel information, he can effectively obtain part of the algorithm's operation laws of the cryptographic algorithm and thus recover the key by various means. The analysis under the gray box model is also known as Side Channel Analysis (SCA), which can be used to obtain the side channel information statistically through electromagnetic analysis (Chari et al. 2523) and other means, and then obtain useful key information, etc.

In recent years, Digital Rights Management(DRM), smartphones and cloud services have emerged, and

†Yatao Yang and Hui Dong contributed equally to this work and should be considered co-first authors.

*Correspondence:
Yatao Yang
yy2008@163.com
[1] Department of Electronic and Communication Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China
[2] School of Telecommunication Engineering, Xidian University, Xi'an 710071, China

more and more cryptographic algorithms are running in untrusted endpoint environments. At the same time, various attacks have emerged, such as key whitening attacks, entropy attacks, and software static analysis. In 2002, Chow et al. (2002a) introduced the first white-box attack model, in which an attacker has full access to the software implementation of a cryptographic algorithm and full control over its execution environment, and can change the implementation details of the cryptographic algorithm at will, with full visibility of the algorithm's computational process, posing a huge potential threat to data security. How to securely implement cryptographic algorithms and secure keys in the program has become an urgent problem. Therefore, the study of new white-box cryptographic algorithms can effectively guarantee the security of keys in a white-box environment, enabling the cryptographic algorithm cope with a variety of attacks and help data and information security.

An attacker can launch key extraction attack, decomposition attack and code lifting attack under the white-box model (Chen et al. 2021). In key extraction attack, the attacker tries to extract the key from the white-box implementation (Yao et al. 2020); in decomposition attack, the attacker tries to find a less costly implementation to maintain the exact same functionality as the original version; in code lifting attack, the attacker uses the original cryptographic program as a large valid key for encrypting/decrypting on different devices. White-box ciphers can provide high strength security in the above attack environment. In this paper, we design the WAS white-box block cipher algorithm using a new lookup table theoretical construction, and analyze and compare the security with other white-box cipher schemes to prove the advantages of this scheme.

Related work

In 2002, Chow et al. (2002a) introduced a white-box attack model and designed a white-box Advanced Encryption Standard (AES) algorithm using a set of key-dependent lookup tables. Chow et al. also proposed a white-box implementation of Data Encryption Standard (DES) by interleaving the use of affine transformation and de-linearization techniques (Chow et al. 2002b), whose main design idea was to hide the key by constructing a lookup table. In 2010, Xiao et al. (2010) improved the white-box AES proposed in the reference (Chow et al. 2002a) by using a 16-bit linear encoding for the obfuscation operation and abandoning the use of nonlinear encoding, which required considerable memory space for the implementation of this scheme as a way to achieve a higher level of security.

In 2009, Xiao and Lai (2009a) used the first white-box implementation of the SM4 algorithm (denoted as the Xiao-SM4 white-box scheme) by constructing a lookup table, which transforms each round of the round function operation into the computation of the affine transform as well as the lookup table, and then the output of the lookup table is dissociated. Karroumi (2010) proposed an alternative white-box implementation of AES in 2010, which allows the expected security level of the scheme to be increased from 2^{30} to 2^{91} by using an additional set of coefficients obtained from the pairwise representation of AES. In 2014, Luo et al. (2014) improved on Xiao et al.'s white-box AES using nonlinear encoding. In 2016, Bai and Wu (2016) proposed a white-box implementation of the SM4 algorithm (denoted as Bai-Wu SM4 white-box scheme) using the same construction of lookup tables. To improve efficiency, the Bai-Wu SM4 white-box scheme uses two types of lookup tables: one to perform output decoding and encoding of new inputs, and the other to compute the round function of the standard SM4 algorithm. In 2020, Si and Jie (2020) proposed a novel white-box implementation of SM4 by first representing the linear transformation of the SM4 algorithm as a matrix, then constructing each round function as four lookup tables with 8-bit inputs and 64-bit outputs, and extracting meaningful 32-bit data after the lookup table using the shift matrix. In 2021, an attack against this scheme was proposed in the reference (Lu and Li 2021).

Another design idea for white-box ciphers is to design new white-box block cipher algorithms that are secure under the white-box model. Usually this design is based on key-related components, such as S-boxes. A common property of new white-box algorithm designs is incompressibility, also known as weak white-box security or spatial hardness.

In 2014, Biryukov et al. (2014) designed a strong white-box public key cryptographic scheme, a weak white-box block cryptographic scheme and a black-box block cryptographic scheme based on the ASASA structure (i.e., nonlinear layer S and affine layer A iterative structure). In 2015, Bogdanov and Isobe (2015) proposed a dedicated white-box scheme called SPACE for spatially hard ciphers. The SPACE reduces the security against key extraction and decomposition attacks under the white-box attack model to the key recovery problem for block ciphers under the black-box attack model; the design idea used is to construct lookup tables from AES by restricting plaintexts and truncating ciphertexts, which makes the attacker cannot recover the key used to generate the lookup table based on the security of AES alone. The concept of (M, Z)-space hardness security was also proposed for evaluating the strength of white-box ciphers against code lifting attacks, which is a generalization of the weak white-box security concept in the reference (Biryukov et al. 2014). In 2016, Bogdanov et al. (2016) proposed a

new efficient white-box block cipher SPNbox, which is designed using a Substitution Permutation Network (SPN) structure as key-dependent S-boxes. The SPNbox can provide all important white-box security properties of quantifiable spatial hardness. In 2017, Lin et al. (2017) designed a white-box cryptographic scheme based on the ASASASA structure. In 2021, Koike and Isobe (2021) designed the white-box block cipher Yoroi to improve the security of code lifting attacks against continuous data leakage by updating incompressible tables. Moreover, Yoroi only needs to update the lookup table periodically and does not require updating the key.

For attacks on white-box ciphers, in 2004, Billet et al. (2004) proposed an attack against the white-box AES scheme in the reference (Chow et al. 2002a), denoted as the BGE attack; the key was successfully recovered by means of combining lookup tables and offsetting nonlinear encoding. In recent years, attacks against the Xiao-SM4 white-box scheme have continued to emerge. In 2013, Lin and Lai (Lin and Lai 2013) proposed an attack that could recover the key with a time complexity of 2^{47}. In 2018, Pan et al. (2018) pointed out some complexity bias in the analysis of Lin et al. In 2021, Zhang et al. (2021) proposed an attack against the Xiao-SM4 white-box scheme, and they proposed Intermediate-Values Mean Difference Analysis (IVMDA) based on Differential Computation Analysis (DCA), and successfully recovered the round key of the Xiao-SM4 white-box scheme. Meanwhile, Zhang et al. proposed an improved scheme that can resist IVMDA by protecting the output of the lookup table with nonlinear encoding. The above scholars have proposed different design schemes and analysis methods in terms of security analysis and performance enhancement of white-box ciphers, but most of them are based on the modification of existing cryptographic algorithms, which are still deficient in terms of security and space performance.

The main contributions of this paper are as follows.

(1) An improved white-box cryptographic algorithm over AS iteration is proposed. Using the AS iterative structure, a lookup table with a five-layer ASASA structure is constructed, and the linear layer uses the Maximum Distance Separable (MDS) matrix. In the black-box model, the WAS can effectively prevent the attacker from recovering the key. The length of both the nonlinear layer S and the affine layer A in the lookup table is 16 bits, which effectively avoids the decomposition attack against the ASASA structure and makes the algorithm secure against key extraction under the white-box model.
(2) The security of the WAS algorithm is proved. The algorithm possesses weak white-box (32 KB, 112)-space hardness and satisfies the anti-code lifting security. Compared with other white-box cryptographic schemes, this scheme occupies less memory space and satisfies the design goals of security and efficiency. The lookup table of ASASA structure is treated as an S-box for testing and validation. After theoretical analysis and experimental testing, the data generated using the pseudo-random number generator has a nonlinearity of about 95.00 and a differential uniformity of 0.039.

Preliminary knowledge

The SPN structure was originally proposed by Shannon and the structure belonged to the same iterative algorithm as the Feistel structure. The cryptographic algorithm of the SPN structure uses two key steps of diffusion and obfuscation for multiplicative iteration, which is constructed as follows (Kong 2021).

(1) Specify the plaintext grouping and key length, and add the generated subkeys to each iteration of the algorithm's encryption and decryption operations.
(2) The plaintext and the key after the operation are subjected to a dissimilarity operation, and then the substitution and replacement operations are performed respectively. The components of Substitution and Permutation are called S-boxes and P-boxes, respectively. The substitution is a nonlinear operation, while the permutation eliminates the statistical properties of the input plaintext and thus better protects the key information.
(3) The ciphertext is obtained by repeated iterations with different iteration rounds designed.

AES is an SPN type block cipher algorithm with a group length of 128 bits and supports three different key sizes, i.e., 128/192/256 bits, denoted as AES-128, AES-192 or AES-256, respectively. In general, AES consists of R rounds with $R+1$ 128-bit round keys, which are obtained from the AES key using the AES key scheduling algorithm; R depends on the key size, i.e., $R=10$, 12 or 14 in the case of AES-128, AES-192 or AES-256. The initial and final states are the plaintext and ciphertext of AES, respectively, and an AES state is represented by a 4×4 byte array $\left[state_{i,j}\right]\left(0 \le i,j \le 3\right)$, called the state array.

Each AES round contains the following four operations, in particular, a key addition operation is performed before the start of the first round, and the last round has no column mixing operation.

(1) SubBytes: The AES S-box is applied to each byte of the state. AES uses a fixed S-box, denoted by $\boldsymbol{S}$, which is a nonlinear bijective mapping from eight

bits to eight bits. The S-box of AES has a high algebraic count.

(2) ShiftRows: In the case of $0 \le i \le 3$, shift each row i of the state array i bytes to the left. The row indexed by $i=0$ remains unchanged.

(3) Mixcolumns: It is a linear operation in $\mathbb{F}_{256}^{16}$, specific definition:

$$\begin{pmatrix} state'_{0,j} \\ state'_{1,j} \\ state'_{2,j} \\ state'_{3,j} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} state_{0,j} \\ state_{1,j} \\ state_{2,j} \\ state_{3,j} \end{pmatrix} \tag{1}$$

(4) AddRoundKey: It is a XOR orientation of the 128-bit round key and the state array.

Bringer et al. (2006) used a white-box implementation of the AES algorithm by inserting scrambled terms. The means used is to add scrambling to the original scheme, which makes the algebraic structure more complex and thus much more difficult for an attacker to carry out the attack.

Design of WAS

WAS algorithm lookup table construction

The lookup table in this scheme is denoted as S, which consists of a five-layer ASASA structure, and the specific process of lookup table generation is as follows.

(1) Generate a sufficient number of pseudo-random bits using a key. Specifically, the sequence of pseudo-random bits is generated in counter mode using the block cipher E, encrypted with the master key. CTR_DRBG is chosen as the PRNG (Pseudo Random Number Generator) and AES-CTR as the underlying architecture of CTR_DRBG. For example, in the case of E=AES-128, the key of AES-CTR is set as the 128-bit secret master key of this scheme, and the 128-bit plaintexts 0, 1, … (as many as possible) are encrypted by encrypting the 128-bit plaintexts 0, 1, … (as many as possible) to finally generate the desired sequence of pseudo-random bits.

(2) Arrange the pseudo-random bits generated in (1) into a 16×16 matrix and check whether the matrix is invertible. If it is invertible, it is left as an invertible affine transform of the A-layer together with any 16-bit constant, which is denoted as Q_j ($j=1, 2, 3$), if it is not invertible, it is discarded.

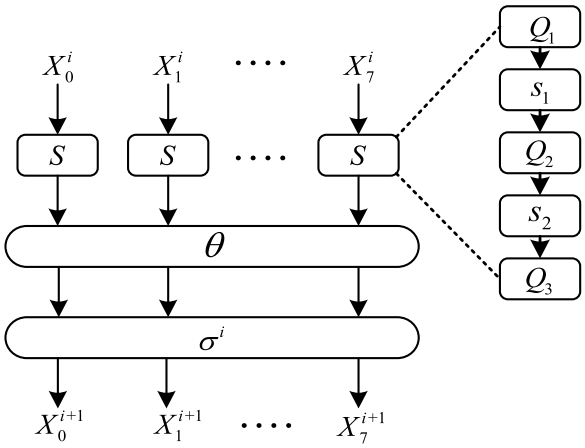

Fig. 1 Structure of WAS round function

(3) The pseudo-random bits generated in (1) are used to generate a 16-bit random permutation using the random permutation generation algorithm (Bacher et al. 2017), denoted as s_j ($j=1, 2$), as the S-layer in the lookup table. In the Additional file 1, we give an example of the random number lookup table formed using the above method.

The generated Q_j and s_j are arranged together to be used as the secret key-related S-box in WAS as follows:

$$S = Q_3 \circ s_2 \circ Q_2 \circ s_1 \circ Q_1. \tag{2}$$

Specific design of WAS algorithm

The WAS uses the SPN structure with MDS matrix as the underlying structure and uses the MDS matrix in the linear layer. The group length of WAS is 128 bits and there are ten rounds of iterations. The state of WAS is defined as a vector consisting of eight elements with 16 bits each: $X=\{X_0,...X_7\}$. The plaintext X^0 is transformed to the ciphertext X^{10} by a round function operation:

$$X^{10} = \left(\bigcirc_{i=1}^{10}\left(\sigma^i \circ \theta \circ \gamma\right)\right)\left(X^0\right) \tag{3}$$

where i is the number of iteration rounds and $\bigcirc$ denotes the composite of the function. The structure of the round function is shown in Fig. 1. The encryption algorithm of WAS is shown in Algorithm 1 and the decryption algorithm is shown in Algorithm 2.

Algorithm 1 WAS encryption algorithm

INPUT: 128-bit plaintext $X = (X_0, X_1, \ldots, X_7)$

OUTPUT: 128-bit ciphertext $Y = (Y_0, Y_1, \quad, Y_7)$

1: **for** $i = 1$ to 10 **do**

2: $Y = (S(X_0), S(X_1), \ldots, S(X_7))$

3: $Y = Y \cdot M_{16}$

4: $Y = Y \oplus (8(i-1)+1, 8(i-1)+2, \ldots, 8(i-1)+8)$

5: **end for**

6: **return** Y

Algorithm 2 WAS decryption algorithm

OUTPUT: 128-bit ciphertext $Y = (Y_0, Y_1, \ldots, Y_7)$

INPUT: 128-bit plaintext $X = (X_0, X_1, \ldots, X_7)$

1: **for** $i = 10$ to 1 **do**

2: $X = Y \oplus (8(i-1)+1, 8(i-1)+2, \ldots, 8(i-1)+8)$

3: $X = X \cdot M_{16}^{-1}$

4: $X = (S^{-1}(X_0), \ldots, S^{-1}(X_7))$

5: **end for**

6: **return** X

Each round of WAS contains three layers, which are nonlinear layer γ, linear layer θ, and affine layer σ^i.

(1) Nonlinear layer γ: It is a nonlinear layer constructed from eight secret key-related S-boxes. Since the S-box can be isolated separately by a white-box attacker, it must be an independent primitive that can ensure the n-bit security of the key even if the attacker obtains the complete ciphertext of the S-box. Here, the AS iteration structure is used to generate a lookup table S with a five-layer ASASA structure as the key-related S-box in the nonlinear layer. The nonlinear substitution layer γ is defined as follows.

$$\begin{aligned} &\gamma : \left(\mathbb{F}_2^{16}\right)^8 \to \left(\mathbb{F}_2^{16}\right)^8 \\ &(X_0, \ldots, X_7) \mapsto (S(X_0), \ldots, S(X_7)) \end{aligned} \tag{4}$$

(2) Linear layer θ: It is a linear layer that plays a diffusion role. The linear layer θ applies an 8×8 MDS matrix with the matrix used in the block cipher (Barreto and Rijmen 2000), defined as follows:

$$M_{16} = had(1_x, 3_x, 4_x, 5_x, 6_x, 8_x, \mathrm{b}_x, 7_x).$$

Linear layer θ is defined as follows.

$$\begin{aligned} &\theta : \left(\mathbb{F}_2^{16}\right)^8 \to \left(\mathbb{F}_2^{16}\right)^8 \\ &(X_0, \ldots, X_7) \mapsto (X_0, \ldots, X_7) \cdot M_{16} \end{aligned} \tag{5}$$

(3) Affine layer σ^i: It is an affine layer, it XOR with C^i which is the constant associated with the round function in round i, defined as follows

$$\begin{aligned} &\sigma^i : \left(\mathbb{F}_2^{16}\right)^8 \to \left(\mathbb{F}_2^{16}\right)^8 \\ &(X_0, \ldots, X_7) \mapsto \left(X_0 \oplus C_0^i, \ldots, X_7 \oplus C_7^i\right) \end{aligned} \tag{6}$$

where $C_j^i = 8(i-1) + j + 1, \quad 0 \le j \le 7$.

Security analysis of WAS algorithm

Security analysis under the black-box model

In the black-box attack model, the attacker's goal is to recover the key. In this case, the attacker attacks by brute-force cracking and its primary goal is to guess the secret component, which in this scenario refers to the lookup table S. The time complexity of the attack here is about $2^{2^{16} \cdot 16}$; if the attacker guesses the master key, the time complexity required here is higher than 2^{128}. Next, the security analysis of WAS under the black-box model is discussed.

The ability of a cryptographic algorithm to resist differential cryptanalysis under a black box model can be assessed by calculating the number of differentially active round functions in the cryptographic algorithm, and the maximum differential probability of the round functions. A round function is said to be differentially active if the input differential of the round function is not zero.

(1) Differential cryptanalysis (Biham and Shamir 2012): for differential cryptanalysis, given an input differential a and an output differential b, $a \in \mathbb{F}_2^n, b \in \mathbb{F}_2^n$. Then for a function f: $\mathbb{F}_2^n \to \mathbb{F}_2^n$, its input is n bits and output is n bits.

$$\Pr[a,b] = \frac{\#\{x|f(x) \oplus f(x \oplus a) = b\}}{2^n} \tag{7}$$

The number of differential branches of the linear layer in WAS is nine. Here, assume that the maximum differential probability of the secret S-box in WAS is $\frac{16}{2^{16}} \times 2 = \frac{16}{2^{15}} = 2^{-11}$, then the maximum differential probability of the WAS round function is 2^{-99}. The 16 here refers to the fact that each branch of the WAS algorithm processes 16 bits of data, 2^{15} refers to the fact that during the construction of the S-box, pseudo-random bits are generated using the random permutation generation algorithm to generate 16 bits of random permutations, and there are two S-layers in the ASASA structure, and according to the above Eq. (7), the maximum differential probability of the WAS round function is 2^{-99}.

On the other hand, since the components in the secret S-box are randomly generated and kept secret, it is difficult for an attacker to obtain the actual differential features. The WAS has at least 18 active S-boxes after four rounds, and if an attacker wants to extend the number of rounds of the differential distinguisher, the increase in the number of rounds corresponds to the rise in the amount of guessing keys required, and the amount of keys to be guessed for each extended round is $2^{2^{16} \cdot 16}$. From the above analysis, it is clear that ten rounds of WAS can resist differential cryptanalysis.

(2) Linear cryptanalysis (Matsui 1993): for linear cryptanalysis, given an input mask α and an output mask β, $\alpha \in \mathbb{F}_2^n, \beta \in \mathbb{F}_2^n$. Then for a function f: $\mathbb{F}_2^n \to \mathbb{F}_2^n$, whose input is n bits and output is n bits.

$$\Pr[\alpha,\beta] = \left(\frac{\#\{x|\alpha \cdot x \oplus \beta \cdot f(x) = 0\}}{2^{n-1}} - 1\right)^2 \tag{8}$$

The number of linear branches of the linear layer in WAS is nine. Here, assume that the maximum linear probability of the secret S-box in WAS is $\left(\frac{2^{16-1}+2^{16-1}\times\frac{2}{5\times16}}{2^{16-1}} - 1\right)^2 \approx 2^{10.62}$. Then the maximum linear probability of the WAS round function is $2^{-95.58}$. The 16 here refers to the fact that each branch of the WAS algorithm processes 16 bits of data, The S-box consists of a five-layer structure, two of which are S-layers, with each branch handling 16 bits of data. According to the constructure of WAS's S-box and Eq. (8), the WAS round function linear probability at this point should be $2^{-95.58}$ approximately.

On the other hand, since the components in the secret S-box are randomly generated and kept secret, it is difficult for an attacker to obtain the actual linear approximation of the round function. The WAS has at least 18 active S-boxes after four rounds, and if the attacker wants to extend the number of rounds of the linear distinguisher, accordingly, the increase in the number of rounds corresponds to the rise in the amount of required guessing keys, and the amount of keys to be guessed for each extended round is $2^{2^{16} \cdot 16}$. From the above analysis, it is clear that ten rounds of WAS can resist linear cryptanalysis.

The lookup table $\boldsymbol{S}$ designed after our white-box construction is not an S-box in the traditional sense, so its value is not fixed. Nevertheless, we can still use the metrics for evaluating S-boxes to measure the performance of these substitution matrices, as they all play a certain role in obfuscating and diffusing the data. An S-box is essentially a mapping $S(x) = \big(f_1(x), f_2(x), \ldots, f_m(x)\big), F_2^n \to F_2^m$. The S-boxes are able to reflect the overall performance of the block cipher through the following metrics (Feng and Wu 2000).

- Nonlinearity
  $S(x) = \big(f_1(x), f_2(x), \ldots, f_m(x)\big), F_2^n \to F_2^m$ is a multi-output function, then there is:

  $$N_s = \min_{\substack{l \in L_n \\ 0 \neq u \in F_2^m}} d_H(u \cdot s(x), l(x))$$

  N_s is the degree of nonlinearity of s(x). where L_n denotes the set of all n-element linear and affine functions. To construct an effective linear approximation of a round function in linear cryptanalysis, it is necessary to construct a linear approximation of the S-box. For linear cryptanalysis, the greater the nonlinearity of the S-box, the better.
- Differential uniformity
  $S(x) = \big(f_1(x), f_2(x), \ldots, f_m(x)\big), F_2^n \to F_2^m$ is a multi-output function, then there is:

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

  δ is the differential uniformity of the S-box. The resistance of a block cipher against differential analysis can be measured using differential uniformity. For differential cryptanalysis, the smaller the differential uniformity, the better.

The lookup table of ASASA structure is treated as an S-box for testing and validation. After theoretical analysis and experimental testing, the data generated using the pseudo-random number generator has a nonlinearity of

about 95.00 and a differential uniformity of 0.039. According to the test results, it is seen that because of the random generation method, our S-box performance is not as good as the classical AES algorithm in terms of nonlinearity and differential uniformity. But our biggest advantage lies in the pseudo-random number generation and white-boxed lookup table construction. This makes it difficult for the attacker to get the correct key information from the fixed S-box or the actual operation of the algorithm, thus guaranteeing the security in the white-box environment.

(3) Structural cryptanalysis (Biryukov and Shamir 2010): structural cryptanalysis usually exploits the propagation properties of a collection of plaintexts with a special structure in the structure of a cryptographic algorithm to initiate an attack. This analysis method generally focuses on the structure of the cryptographic algorithm, independent of the specific algorithmic details, and is particularly suitable for cryptographic algorithms with secret components. A 2.5-round generalized integral distinguisher for SPN-type structures that can recover the secret S-box requires a data complexity of 2^{32} selected plaintexts and a time complexity of 2^{48}. However, no feasible structural cryptanalysis has been found to break a 10-round WAS, so the current 10-round WAS can resist structural cryptanalysis.

Security analysis of anti-key extraction attack under white-box model

First, a description of the key extraction attack under the white-box model is given.

(1) Decomposition attack on AS iteration structure. The key extraction attack on a white-box block cipher algorithm with AS iterative structure under the white-box model can be reduced to a decomposition attack on the AS iterative structure under the black-box model. Biryukov et al. (Biryukov and Khovratovich 2015) proposed a decomposition attack on the ASASA scheme. It is shown that the decomposition attack on a white-box block cipher algorithm with AS iterative structure holds when the size m of the S-box and the block length n satisfy the conditions $m^2 \leq n$, e.g., 8-bit S-box and 128-bit block length.

(2) Key recovery attack on the underlying block cipher. The key extraction attack under the white-box model for white-box block ciphers that use the entire codebook of a small block cipher as a lookup table can be reduced to the key recovery attack on the underlying block cipher under the black-box model. The underlying block cipher should be secure against key recovery attacks, i.e., there is no more effective attack than a generic attack such as a brute force attack. For example, the underlying block cipher in SPACE is AES-128, and despite the extensive cryptanalysis work (Bogdanov et al. 2011; Derbez et al. 2013), no effective key recovery attack has been performed so far. More precisely, the key extraction advantage $\mathrm{Adv}_{\mathrm{KE\text{-}WB}}$ of SPACE in the white-box model, is limited by the key recovery advantage $\mathrm{Adv}_{\mathrm{KR\text{-}BB}}$ of the underlying block cipher in the black-box model: $\mathrm{Adv}_{\mathrm{KE\text{-}WB}} \leq \mathrm{Adv}_{\mathrm{KR\text{-}BB}}$.

(3) Key recovery attacks on random permutations. The key extraction attack under the white-box model of the white-box block cipher algorithm that uses random permutations as lookup tables can be reduced to the key recovery attack on random permutations. In this case, a PRNG with some security and a random permutation generation algorithm are used to generate random permutations with each other, e.g., AES-CTR. Although an attacker who knows the complete ciphertext can find the pseudo-random bit string used in the generation by reversing the random permutation generation algorithm, he can only get some plaintext ciphertext pairs of AES-CTR, which cannot be used to recover the key. The security of AES-CTR makes it difficult for an attacker to recover the master key.

In the following, we analyze the security of anti-key extraction under the white-box model.

In the white-box model, an attacker can observe not only the inputs and outputs of the cryptographic algorithm, but also has full access to all entries of the lookup table, i.e., all input–output pairs in the table. For key extraction security, an attacker cannot extract the keys embedded in the white-box scheme. Key extraction attack is the most typical attack strategy against white-box schemes today.

Biryukov and Khovratovich (2015) proposed a decomposition attack on the ASASA scheme. The authors show that the decomposition attack on the ASASA scheme holds when the small S-box, here the size m of the S-box in the nonlinear layer of the ASASA structure and the length n of the block cipher satisfy the conditions $m^2 \leq n$, for example, an 8-bit S-box and a 128-bit block length. As mentioned above, the input and output of the lookup table in WAS with a five-layer ASASA structure are 16 bits, and the S-layer of the lookup table is composed of 16-bit random permutations, and the A-layer is a 16-bit reversible affine transformation, i.e., both m and n are 16 bits, which does not satisfy the condition $m^2 \leq n$, so

the lookup table in this scheme is secure against such a decomposition attack.

From the above analysis, it is clear that the ASASA structure of the lookup table in this scheme can resist the key extraction attack. Therefore, the attacker cannot recover the A and S layers of the lookup table, and cannot successfully extract the master key K_{AES} of AES-CTR.

Security analysis of anti-code lifting attack under white-box model

In a mobile payment application scenario, an attacker located in the user's phone, for example, malware may try to lift the decryption key and use it to recover the transaction credentials; or it may copy the entire application to run on the phone to communicate with the payment terminal. Therefore, white-box ciphers should also have security against code lifting attack. In this paper, the definition of weak white-box space hardness (Bogdanov et al. 2016) is used to measure the strength of WAS against code lifting attacks.

The group length of WAS is 128 bits, and there are eight secret S-boxes in the nonlinear layer, each with 16 bits of input and output, for a total of ten iterations. In order to increase the probability of correctly decrypting a random ciphertext, an attacker can use a space of size less than M to store the explicit ciphertext pairs. Assuming that the cryptographer is able to store a total of 1/4 of the amount of code, the amount of code saved by the attacker is given as $M_a = 16 \times 2^{14} bit$.

The set of plaintext pairs that the attacker has saved is denoted X_s, the ciphertext is denoted x_0, the plaintext is denoted x_i, and the attacker's guess of the plaintext is denoted $\hat{x}_i$. If the attacker has saved the plaintext corresponding to this random ciphertext, then he can decrypt this ciphertext with probability 1. If the attacker has not saved the plaintext corresponding to this ciphertext, then the attacker needs to guess the plaintext corresponding to this random ciphertext.

For one round of WAS, the probability of a successful guess by the attacker is denoted as p_1, as follows.

$$\begin{aligned} p_1 &= \Pr[\hat{x}_1 = x_1 | x_0 \in X_s] \cdot \Pr[x_0 \in X_s] + \Pr[\hat{x}_1 = x_1 | x_0 \notin X_s] \cdot \Pr[x_0 \notin X_s] \\ &= 1 \times \frac{(16 \times 2^{14})/128}{2^{128}} + \left(\frac{1}{2^{128} - (16 \times 2^{14})/128} \right) \cdot \left(1 - \frac{(16 \times 2^{14})/128}{2^{128}} \right) \\ &\approx 2^{-112} \end{aligned} \tag{9}$$

Therefore, the WAS has weak white-box (32 KB, 112)-space hardness, which means that even if an attacker succeeds in stealing a quarter of the entire lookup table, he cannot correctly decrypt a randomly selected ciphertext with a probability greater than 2^{-112}. The WAS and other schemes do not have the same anti-code lifting security, but the WAS has good performance for code lifting attacks.

Side channel attack analysis

Bos et al. (2016) proposed a new class of side channel analysis means called Differential Computation Analysis (DCA). The DCA can be considered as a software version of the equivalent of Differential Power Analysis (DPA) applied to hardware. This analysis exploits memory access patterns during the execution of white-box AES software, which allows attackers to execute binaries and simultaneously use dynamic binary tool frameworks such as PIN and Valgrind; by acquiring software traces, they can record read and write access traces to memory. Software traces are used to record the memory addresses accessed by the program during the encryption process. These traces also include other information that can be monitored using binary instrumentation, such as stack reads or register values. Software traces are used to determine which encryption algorithm is implemented, to determine the approximate location of the encryption algorithm in the software implementation, and to perform statistical analysis to extract the secret key. Side channel analysis takes advantage of the fact that each lookup table depends on only a small portion of the key, such as eight or 16 bits of the key. The DCA can efficiently extract a small portion of the key with the help of side channel leakage. However, the lookup table in this scheme contains the full 128 bits of key information. Therefore, even if an attacker can completely monitor the memory access pattern of the target key-related lookup table, the amount of information that the attacker has to guess is 2^{128}. Therefore, the WAS is resistant to DCA.

White-box diversity and white-box ambiguity analysis

White-box diversity (Chow et al. 2002a): refers to the total number of possible lookup tables constructed in a white-box scheme. Therefore, the greater the white-box diversity, the more difficult it is for a cryptanalyst to break the scrambled code, and the more secure the white-box scheme is (Xiao and Lai 2009b).

White-box ambiguity (Chow et al. 2002a): refers to the number of possible constructions for a given white-box cipher or lookup table. The larger the white-box ambiguity, the more difficult it is for the analyst to compute the

key disambiguation code and the initial key, and the more secure the scheme is.

The number of integrable matrices of order 16 on $\mathbb{F}_2^n$ is approximately 2^{254}, thus

$$\text{White - box diversity}: \left(2^{16\times 2^{16}}\right)^2 \times \left(2^{16}\times 2^{254}\right)^3$$

$$\text{White - box ambiguity}: \left(2^{16\times 2^{16}}\right)^2 \times \left(2^{16}\times 2^{254}\right)^3.$$

Decomposition attacks against the ASASA structure

In 2018, Minaud et al. (2018) proposed a new algebraic key-recovery attack, and was able to break the secret-key scheme as well as the remaining public-key scheme, in time complexity 2^{63} and 2^{39}, respectively. This attack method targets the ASASA structure proposed by Biryukov et al. in ASIACRYPT 2014 (Biryukov et al. 2014). After testing and theoretical analysis, the attack achieved good results. Contrary to the attack of Gilbert et al. (2015), this attack is no hope of patching the scheme by increasing the number of perturbation polynomials. In addition, Dinur et al. (2015) proposed decomposition attacks and declared that these attacks are able to break all the proposed concrete ASASA constructions with practical complexity.

Based on the above scholars' proposed decomposition attack against the ASASA structure, the resistance of our WAS algorithm to this type of attack is analyzed.

The attack method proposed by Minaud et al. (2018) focus on removing the first linear layer, through creating a ranking function $\boldsymbol{F}$ to recognize whether an input difference δ activates one or two S-boxes in the first S-box layer. During the attack, the δ's are randomly selected. Then 16 linearly independent δ's are found and verified by a ranking function, and whether a single S-box is activated. Each δ has probability 2^{-8} of activating a single S-box.

Our WAS algorithm uses eight independent S-boxes that are constructed by white-box through multiple layers of iterations, $S = Q_3 \circ s_2 \circ Q_2 \circ s_1 \circ Q_1$. Pseudo-random bit sequences Q_j and random permutations s_j enhance the obfuscation of the data. According to the decomposition attack, the probability of eight S-boxes in a round being activated is $\left(2^{-8}\right)^8 = 2^{-64}$. The probability that the key information in all ten rounds are cracked is $\left(\left(2^{-8}\right)^8\right)^{10} = 2^{-640}$. This value clearly exceeds the predetermined computational complexity of the attacker's attack.

In addition, the decomposition attack assumes that the attacker has access to the full codebook. Even in an actual white-box attack environment, this is quite difficult. Based on actual attack tests using the MDS matrix, both ranking functions distinguish δ's that activate one or two S-boxes much less efficiently. The WAS algorithm is constructed using the MDS matrix, which increases the difficulty of activation during the attack to a certain extent.

The resistance of WAS also exceeds the required computational complexity in Gilbert's attack (Gilbert et al. 2015) method. Also the introduction of the MDS matrix and pseudo-random bit sequences increases the time complexity for the attacker to execute an effective attack. The scheme by Gilbert et al. concentrates on the cracking and analysis of public key ciphers. This method is not fully applicable for WAS that uses random permutations and pseudo-random series for iteration to construct a white-box scheme.

Therefore, combining the above three approaches against the ASASA structure, the WAS scheme has good resistance to decomposition attacks and can be applied in a white-box environment.

Comparison

The memory space required for WAS is 16×2^{16} 128 KB, the space comparison with other white-box schemes is shown in Table 1, the security comparison of different white-box schemes is shown in Table 2.

In the reference (Biryukov et al. 2014), the authors used the ASASA structure to construct a white-box cryptographic scheme that occupies 8 MB of memory space, but it is currently compromised and cannot resist key extraction attacks as well as code lifting attacks under the white-box model. The SPACE-16 scheme in the reference (Bogdanov and Isobe 2015) occupies 918 KB of memory space and the SPACE-24 scheme in the reference (Bogdanov and Isobe 2015) occupies 218 MB of memory space. The SPACE scheme adopts a very conservative design strategy and its internal round function needs to call a full ten rounds of AES-128, thus it is less efficient. The Whiteblock-16 scheme in the reference (Fouque et al.

Table 1 Space comparison of different white-box schemes

Scheme	Space occupancy
Whitebox ASASA (Biryukov et al. 2014)	8 MB
SPACE-16 (Bogdanov and Isobe 2015)	918 KB
SPACE-24 (Bogdanov and Isobe 2015)	218 MB
Whiteblock-16 (Fouque et al. 2016)	2 MB
FPL-AES-128 (Kwon et al. 2020)	10.63 MB
WARX (Liu et al. 2022)	128 KB
Yoroi-16 (Koike and Isobe 2021)	384 KB
Yoroi-32 (Koike and Isobe 2021)	48 GB
WAS	128 KB

Table 2 Security comparison of different white-box schemes

Scheme	Key extraction	Code lifting	SCA	Differential analysis	Linear analysis
Whitebox ASASA (Biryukov et al. 2014)	–	–	–	√	√
SPACE-16 (Bogdanov and Isobe 2015)	√	√	–	√	√
SPACE-24 (Bogdanov and Isobe 2015)	√	√	–	√	√
Whiteblock-16 (Fouque et al. 2016)	√	√	–	–	–
FPL-AES-128 (Kwon et al. 2020)	√	√	–	√	√
WARX (Liu et al. 2022)	√	√	√	√	√
Yoroi-16 (Koike and Isobe 2021)	√	√	√	–	–
Yoroi-32 (Koike and Isobe 2021)	√	√	√	–	–
WAS	√	√	√	√	√

2016) occupies 2 MB of memory space and achieves incompressibility by using key-dependent pseudorandom functions. The FPL-AES scheme in the reference (Kwon et al. 2020) uses parallel lookup tables to design white-box block ciphers with high storage cost and requires 13.75 MB of storage space. The WARX scheme in reference (Liu et al. 2022) uses addition, rotation, XOR operations and MDS matrix, and the WARX is more efficient than SPNbox-16 and WEM. The Yoroi-16 scheme in reference (Koike and Isobe 2021) enhances the security of code lifting attacks against persistent leaks by updating incompressible tables, but requires multiple lookup tables and 384 KB of storage space. The Yoroi-32 scheme in the reference (Koike and Isobe 2021) requires a large storage space of 48 GB. Compared with other white-box cryptographic algorithms, the WAS has a lower storage cost of 128 KB and the size of both S and A layers in its lookup table is 16 bits.

As a result of the above analysis, our WAS algorithm can resist a wide range of attacks and has a small space, and it can satisfy the security requirements in a white-box environment. It has good overall performance.

Conclusion

In this paper, we propose the WAS, an improved white-box cryptographic algorithm over AS iteration and MDS matrix. The design uses the AS iterative structure to construct a lookup table with a five-layer ASASA structure, and uses the SPN structure with MDS matrix as the underlying structure to reduce the number of rounds of the algorithm and improve the implementation efficiency of WAS by taking advantage of the good diffusion property of MDS matrix.

The security analysis shows that the WAS can effectively prevent the attacker from recovering the key under the black-box model. The lookup table of ASASA structure is treated as an S-box for testing and validation. Firstly, after theoretical analysis and experimental testing, the data generated using the pseudo-random number generator has a nonlinearity of about 95.00 and a differential uniformity of 0.039. Secondly, since the size of both S-layer and A-layer in the lookup table is 16 bits, they can resist the decomposition attack against the ASASA structure, and WAS has anti-key extraction security under the white-box model. Finally, the strength of WAS against code lifting attacks is evaluated using weak white-box space hardness. Compared with other white-box cryptographic algorithms, this scheme takes up less memory space. With the same anti-key extraction security and anti-code lifting security, the WAS requires 128 KB of memory space, which is only 14% of SPACE-16 and 33% of Yoroi-16, meeting the design goal of security and efficiency, and can be used for digital rights management, mobile payment, etc.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s42400-023-00192-7.

Additional file 1. The Example of Random Number Lookup Table.

Acknowledgements
We thank the experts for their careful review and valuable comments, and the editorial experts for their hard work.

Author contributions
Conceptualization, ZY; methodology, YY, DH; software, DH; validation, ZY, DH; formal analysis, ZY; investigation, YY; data curation, ZY; writing-original draft preparation, DH writing—review and editing, YY, DH; visualization, ZY; supervision, YY; funding acquisition, YY, DH. All authors have read and agreed to the published version of the manuscript.

Funding
This work was supported by Beijing Natural Science Foundation (No: 4232034); the Fundamental Research Funds for the Central Universities (No: 328202222).

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 4 May 2023 Accepted: 18 September 2023
Published online: 08 December 2023

References

Bacher A, Bodini O, Hwang HK et al (2017) Generating random permutations by coin tossing: classical algorithms, new analysis, and modern implementation. ACM Trans Algorithms 13(2):1–43
Bai K, Wu C (2016) A secure white-box SM4 implementation. Secur Commun Netw 9(10):996–1006
Barreto P, Rijmen V (2000) The Khazad legacy-level block cipher. Primitive submitted to NESSIE 97(106)
Biham E, Shamir A (2012) Differential cryptanalysis of the data encryption standard. Springer
Billet O, Gilbert H, Ech-Chatbi C (2004) Cryptanalysis of a white-box AES implementation. In: 11th International workshop on selected areas in cryptography. LNCS, vol 3357, pp 227–240. Springer, Berlin
Biryukov A, Shamir A (2010) Structural cryptanalysis of SASAS. J Cryptol 23(4):505–518
Biryukov A, Khovratovich D (2015) Decomposition attack on SASASASAS. Cryptology ePrint Arch 2015:646
Biryukov A, Bouillaguet C, Khovratovich D (2014) Cryptographic schemes based on the ASASA structure: black-box, white-box, and public-key. In: 20th International conference on the theory and application of cryptology and information security. LNCS, vol 8873, pp 63–84. Springer, Berlin
Bogdanov A, Isobe T (2015) White-box cryptography revisited: space-hard ciphers. In: 22nd ACM SIGSAC conference on computer and communications security. ACM, New York, pp 1058–1069
Bogdanov A, Khovratovich D, Rechberger C (2011) Biclique cryptanalysis of the full AES. In: 17th International conference on the theory and application of cryptology and information security. LNCS, vol 7073, pp 344–371. Springer, Berlin
Bogdanov A, Isobe T, Tischhauser E (2016) Towards practical white-box cryptography: optimizing efficiency and space hardness. In: 22nd International conference on the theory and application of cryptology and information security. LNCS, vol 10031, pp 126–158. Springer, Berlin
Bos J, Hubain C, Michiels W, et al (2016) Differential computation analysis: hiding your white-box designs is not enough. In: International conference on cryptographic hardware and embedded systems. LNCS, vol 9813, pp 215–236. Springer, Berlin
Bringer J, Chabanne H, Dottax E (2006) White-box cryptography: another attempt. IACR Cryptol ePrint Arch 2006(51):468
Chari S, Rao JR, Rohatgi P (2002) Template attacks. In: 4th International workshop on cryptographic hardware and embedded systems. LNCS, vol 2523, pp 13–28. Springer, Berlin
Chen J, Tong P, Yao S (2021) A white-box implementation of lightweight block cipher GIFT. Inf Netw Secur 21(02):16–23
Chow S, Eisen P, Johnson H, et al (2002a) White-box cryptography and an AES implementation. In: 9th International workshop on selected areas in cryptography. LNCS, vol 2595, pp 250–270. Springer, Berlin
Chow S, Eisen P, Johnson H, et al (2002b) A white-box DES implementation for DRM applications. In: ACM workshop on digital rights management. LNCS, vol 2696, pp 1–15. Springer, Berlin
Derbez P, Fouque PA, Jean J (2013) Improved key recovery attacks on reduced-round AES in the single-key setting. In: 32nd Annual international conference on the theory and applications of cryptographic techniques. LNCS, vol 7881, pp 371–387. Springer, Berlin
Dinur I, Dunkelman O, Kranz T, et al (2015) Decomposing the ASASA block cipher construction. Cryptol ePrint Arch
Feng D, Wu W (2000) Design and analysis of block ciphers. Tsinghua University Press, Beijing
Fouque P, Karpman P, Kirchner P, et al (2016) Efficient and provable white-box primitives. In: 22nd International conference on the theory and application of cryptology and information security. LNCS, vol 10031, pp 159–188. Springer, Berlin
Gilbert H, Plût J, Treger J (2015) Key-recovery attack on the ASASA cryptosystem with expanding S-boxes. In: 35th Annual cryptology conference. LNCS, vol 9215, pp 475–490. Springer, Berlin
Karroumi M (2010) Protecting white-box AES with dual ciphers. In: 13th International conference on information security and cryptology. LNCS, vol 6829, pp 278–291. Springer, Berlin
Kocher P (1996) Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: 16th Annual international cryptology conference. LNCS, vol 1109, pp 104–113. Springer, Berlin
Koike Y, Isobe T (2021) Yoroi: updatable white-box cryptography. IACR Trans Cryptogr Hardw Embed Syst 2021(4):587–617
Kong M (2021) Differential fault attacks against Feistel and SPN cryptographic structures. Changsha University of Science and Technology, Changsha
Kwon J, Lee B, Lee J, et al (2020) FPL: white-box secure block cipher using parallel table lookups. In: Cryptographers' track at the RSA conference. LNCS, vol 12006, pp 106–128. Springer, Berlin
Lin T, Lai X (2013) An effective attack on white box SMS4 implementation. J Softw 24(9):2238–2249
Lin T, Lai X, Xue W et al (2017) A new Feistel type white-box encryption scheme. J Comput Sci Technol 32(2):386–395
Liu J, Rijmen V, Hu Y et al (2022) WARX: efficient white-box block cipher based on ARX primitives and random MDS matrix. Sci China Inf Sci 65(3):1–15
Lu J, Li J (2021) Cryptanalysis of two white-box implementations of the SM4 block cipher. In: 24th International conference on information security. LNCS, vol 13118, pp 54–69. Springer, Berlin
Luo R, Lai X, You R (2014) A new attempt of white-box AES implementation. In: IEEE International conference on security, pattern analysis, and cybernetics, pp 423–429. IEEE, New Jersey
Matsui M (1993) Linear cryptanalysis method for DES cipher. In: Workshop on the theory and application of cryptographic techniques. LNCS, vol 765, pp 386–397. Springer, Berlin
Minaud B, Derbez P, Fouque PA et al (2018) Key-recovery attacks on ASASA. J Cryptol 31(3):845–884
Pan W, Qin T, Jia Y et al (2018) Analysis of two SM4 white-box schemes. J Cryptol Res 5(6):651–671
Si Y, Jie C (2020) A new white box implementation of SM4 algorithm. J Cryptol Res 7(3):358–374
Xiao Y (2010) White box cryptography and implementation of AES and SMS4 algorithms. Shanghai Jiaotong University
Xiao Y, Lai X (2009a) White box cryptography and white box implementation of SMS4 algorithm. In: 2009 Annual meeting of Chinese cryptography society, pp 24–34. Science Press, Beijing
Xiao Y, Lai X (2009b) A secure implementation of white-box AES. In: 2nd IEEE international conference on computer science and its applications, pp 1–6
Yao S, Chen J, Yating G et al (2020) A new white-box implementation of CLEFIA algorithm. J Xidian Univ 47(05):150–158
Zhang Y, Xu D, Chen J (2021) Analysis and Improvement of white-box SM4 implementation. J Electron Inf Technol 43:1–11

Publisher's Note