SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2022, Vol. 65 132302:1–132302:15
https://doi.org/10.1007/s11432-020-3105-1

WARX: efficient white-box block cipher based on ARX primitives and random MDS matrix

Jun LIU[1,2*], Vincent RIJMEN[2,3*], Yupu HU[1], Jie CHEN[1] & Baocang WANG[1]

[1]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an* 710071*, China;*
[2]*Interuniversity Microelectronics Centre & Computer Security and Industrial Cryptography Group, KU Leuven, Leuven* 3001*, Belgium;*
[3]*Department of Informatics, University of Bergen, Bergen* 5020*, Norway*

Received 25 August 2020/Revised 20 October 2020/Accepted 3 November 2020/Published online 27 August 2021

Abstract White-box cryptography aims to provide secure cryptographic primitives and implementations for the white-box attack model, which assumes that an adversary has full access to the implementation of the cryptographic algorithms. Real-world applications require highly efficient and secure white-box schemes, whereas the existing proposals cannot meet this demand. In this paper, we design a new white-box block cipher based on addition/rotation/XOR (ARX) primitives and random maximal distance separable (MDS) matrix, white-box ARX (WARX), aiming for efficient implementations in both black- and white-box models. The implementation of WARX in the black-box model is nine times faster than SPNbox-16 from ASIACRYPT'16, and the implementation in the white-box model is more efficient than SPNbox-16 and WEM from CT-RSA'17. Moreover, the security of WARX in both black- and white-box models is analyzed, which ensures its practical applicability. The design of WARX shows that ARX primitives and random linear layer can improve the efficiency of a white-box block cipher. This article may inspire more provably secure and efficient white-box block ciphers and help to narrow the gap between provably secure white-box schemes from academia and highly applicable schemes in great demand from industry.

Keywords white-box cryptography, block cipher, design, addition/rotation/XOR, efficiency improvement

Citation Liu J, Rijmen V, Hu Y P, et al. WARX: efficient white-box block cipher based on ARX primitives and random MDS matrix. Sci China Inf Sci, 2022, 65(3): 132302, https://doi.org/10.1007/s11432-020-3105-1

1 Introduction

White-box cryptography, introduced in [1], studies secure cryptographic primitives and implementations in the white-box attack context, where an adversary has total visibility on the primitives' implementation and full control on their execution platforms. In recent years, white-box cryptography has fast-rising demand in many scenarios, namely, digital rights management (DRM), mobile payment, memory-leakage resilient software, and many more. In contrast to the white-box attack context which depicts the threats in untrusted ends, the traditional black-box attack context assumes that the adversary can only observe the external execution behavior of a cryptographic algorithm. In real-world applications with classical client/server mode, the server, which is depicted as a black-box model, deals with a large number of user keys simultaneously, and the user device, which is depicted as a white-box model, deals with real-time transactions or plays real-time digital content. Therefore, a white-box cryptographic primitive requires high security and encryption/decryption efficiency in both black- and white-box models. Considering the limited computation resources of user devices, the white-box implementation also demands lower storage cost.

However, the existing white-box schemes cannot meet these requirements. Since 2002, a number of white-box implementations of data encryption standard (DES) and advanced encryption standard (AES) have been proposed [1–5], but their security was penetrated [6–12]. At present, there are no recognized secure white-box implementations of DES/AES. Since 2014, several provably secure white-box block

* Corresponding author (email: jliu6@stu.xidian.edu.cn, vincent.rijmen@esat.kuleuven.be)

ciphers have been proposed, e.g., ASASA [13] (broken in [14, 15]), SPACE-8/16/24/32 [16], SPNbox-8/16/24/32 [17], WhiteBlock [18], WEM [19], and ASASASA [20]. Most of them belong to a common design class of white-box block ciphers called space-hard ciphers. The strategy is to use components that are secure in the black-box model and that have bulky implementation code. Although secure so far, the aforementioned space-hard white-box block ciphers are limited by their efficiency. For instance, the black-box implementation of SPNbox-16 [17] needs a 320-round execution of scaled-down AES on 16-bit blocks and a 10-round multiplication over the finite field $\mathrm{GF}(2^{16})$; the white-box implementation of WEM needs a 60-round execution of AES; the white-box implementations of SPACE-32/SPNbox-32 need storages of 52/17 GB.

These limitations motivate us to design a new white-box block cipher with better black- and white-box efficiency. For black-box implementation, we consider addition-rotation-XOR (ARX) primitives for their high software performance, which have received steady attention in designing symmetric-key primitives [21–23], especially in the area of lightweight cryptography. For white-box implementation, the efficiency bottleneck lies in the linear layer because the nonlinear component is usually implemented by a lookup-table (LUT for short) which has equivalent efficiency for different ciphers. Hence, we are interested in the influence of the linear layer in white-box efficiency improvement. We consider a random maximal distance separable (MDS) matrix, aiming to reduce the number of rounds of the cipher. However, it is unclear whether ARX and random MDS matrix can improve the efficiency of a white-box block cipher, under the premise of not affecting security.

In this paper, a new white-box block cipher based on ARX primitives and random MDS matrix, WARX (white-box ARX), is proposed. The contributions include:

Efficient black-box implementation. The round-based black-box implementation of WARX has impressive software performance on Intel platforms. First, an ARX block cipher with the block size of 16 bits, mSparx-16, is proposed. This small-size block cipher is used to construct the key-dependent nonlinear component of WARX because it provides provable security against typical differential/linear cryptanalysis in the black-box model. Despite using more rounds to achieve a desired security bound against differential/linear cryptanalysis, ARX ciphers are faster than common substitution-permutation network (SPN) ciphers due to their simple and efficient arithmetic instructions. With the help of this building block, WARX outperforms SPNbox-16 by a factor of nine in the efficiency of the black-box implementation. Another benefit of ARX primitives is their decreased vulnerability to cache-timing attacks due to constant-time instructions.

Efficient white-box implementation. The LUT-based white-box implementation of WARX has competitive efficiency on both Intel and ARM platforms. We introduce randomness to the MDS matrix, which is wildly used in block ciphers to provide high security. By using the random MDS linear layer, WARX needs fewer rounds to achieve the best white-box security level. Specifically, WARX improves one round in this aspect, compared to most white-box block ciphers without a random linear layer. Consequently, WARX is superior in white-box efficiency.

Observation, limitation, and implication. The structure of WARX is a combination of a key-dependent non-linear layer and a random linear layer. The core contribution of this work is to study the black/white-box security and efficiency of this structure. WARX reveals that a random linear layer can help improve the white-box efficiency because the linear layer can contribute to white-box security, and hence the lower bound of the number of rounds for achieving a certain security level is decreased.

Unfortunately, the efficiency improvement is limited (one round). Such a limitation comes from a constant white-box security bound, meaning that the cipher can be broken in the white-box model with a fixed probability regardless of the number of rounds. In other words, the white-box security of a block cipher reaches maximal and stops growing after some rounds. So, we focus on the lower bound of the number of rounds that fulfills the maximal achievable white-box security. For instance, this bound is 7/8/8 for WARX/SPNbox-16/WEM.

Despite the limited outcome, we hope this work can bring up new structures in designing more efficient white-box block ciphers, and we expect the gap between provably secure white-box schemes from academia and highly applicable schemes in great demand from industry vanishes soon.

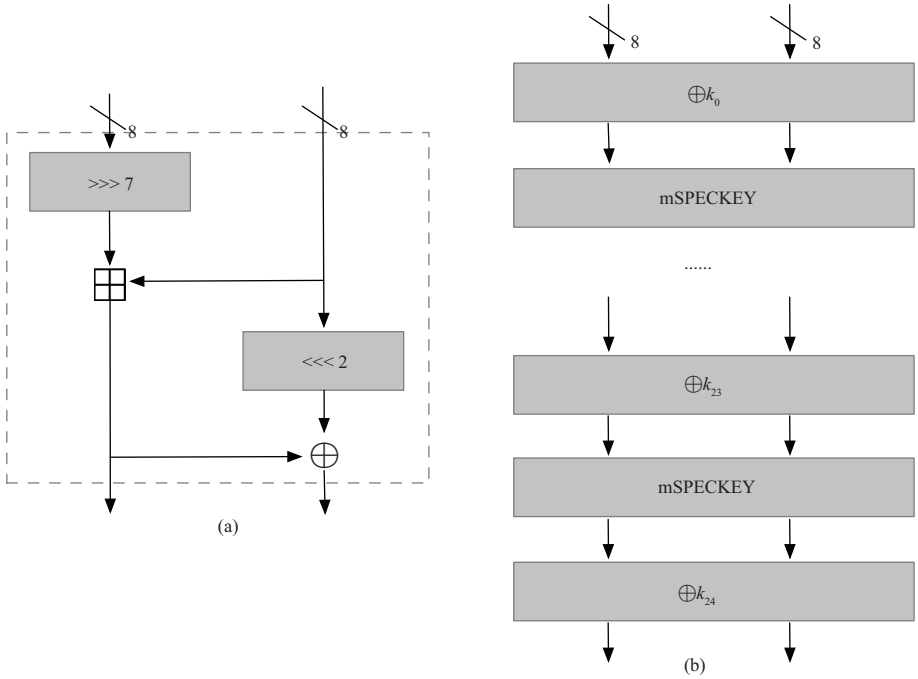

Figure 1 (a) 16-bit ARX-box mSPECKEY. $\ggg 7$ means right rotation by 7 bits, $\lll 2$ means left rotation by 2 bits, and $\boxplus$ means addition modulo 2^8. (b) 16-bit ARX block cipher mSPARX-16.

2 WARX: specifications

WARX is a 7-round block cipher with an iterated SPN structure. The block size n and key size k are 128 bits. The cipher state is organized as a row vector of eight elements with each in $\mathrm{GF}(2^{16})$: $\boldsymbol{x} = (x_0, x_1, \ldots, x_7)$.

Round transformation. Each round of WARX consists of three layers, the nonlinear layer, and then the linear layer followed by the constant addition layer. The nonlinear layer applies eight identical key-dependent S-boxes in parallel: $\boldsymbol{x} = (S(x_0), S(x_1), \ldots, S(x_7))$. The linear layer applies an 8×8 MDS matrix in $\mathrm{GF}(2^{16})$: $\boldsymbol{x} = (\boldsymbol{M}_R \cdot (x_0, x_1, \ldots, x_7)^{\mathrm{T}})^{\mathrm{T}}$ where T means transposition of a vector or matrix. The constant addition layer adds some round-dependent constants: $\boldsymbol{x} = (x_0, x_1, \ldots, x_7) \oplus (8(r-1)+1, 8(r-1)+2, \ldots, 8(r-1)+8)$ where r denotes the r-th round ($r = 1, 2, \ldots, 7$).

S-box generation. The key-dependent S-box S is generated by the 16-bit ARX block cipher mSparx-16 whose design is inspired by Sparx [22]. First, a 16-bit ARX-box mSpeckey is constructed. The notion of ARX-box was introduced by Biryukov et al. at FSE'16 (see Definition 1), together with two examples of 32-bit ARX-boxes Speckey and MARX [24]. The structure of mSpeckey, shown in Figure 1(a), comes from the scaled-down version of Speckey. Then, mSparx-16 is constructed as a 24-round key alternating cipher [25] which intersects the key addition with ARX-box, as shown in Figure 1(b). Finally, to generate the 16-bit S-box S, each element e is seen as a plaintext and encrypted by mSparx-16, and the ciphertext is used as the output of this element.

Definition 1 (ARX-box [24])**.** An ARX-box is a permutation on m bits that relies entirely on addition, rotation, and XOR to provide both non-linearity and diffusion. It is a particular type of S-box.

MDS matrix generation. The random MDS matrix $\boldsymbol{M}_R$ is generated by multiplying a public predefined MDS matrix $\boldsymbol{M}_P$, e.g., the MDS matrix used in the block cipher Khazad [26], by some random numbers. First, a vector $\boldsymbol{r} = (rc_0, rc_1, \ldots, rc_7)$ with each element nonzero and randomly chosen from $\mathrm{GF}(2^{16})$ is generated. Then, $\boldsymbol{M}_R$ is generated simply by multiplying these numbers by the corresponding rows of $\boldsymbol{M}_P$: $\boldsymbol{M}_R = \mathrm{diag}(rc_0, \ldots, rc_7) \cdot \boldsymbol{M}_P$, where $\mathrm{diag}(rc_0, \ldots, rc_7)$ is a block diagonal matrix. The irreducible polynomial is $x^{16} + x^5 + x^3 + x + 1$.

Key schedule. The 16-bit round keys are generated by a key derivation function (KDF) [27] as in SPNbox-16, e.g., an extendable-output function (XOF) [28]. Specifically, the 128-bit master key K is expanded to 25 round keys by the KDF: $(k_0, \ldots, k_{24}) = \mathrm{KDF}(K, 400)$, where $k_0, \ldots, k_{24}$ are used for generating the S-box.

Implementations. WARX involves four algorithms, SETUP, MSPARX, BBI, and WBI, whose pseudo-code are given in Algorithms 1–4, respectively. Taking DRM as an example, SETUP first runs the key schedule to generate the round keys and then generates the random MDS matrix and S-box by calling MSPARX, which is the encryption algorithm of the 16-bit block cipher mSparx-16. BBI is the round-based black-box encryption implementation of WARX with the S-box implemented by calling MSPARX. WBI is the LUT-based white-box decryption implementation of WARX with the (inverse) S-box implemented by looking up a table. Normally, SETUP, MSPARX, and BBI are all running on the server, while WBI is running on a user's device.

Algorithm 1 SETUP: setup procedure of WARX

Input: Master key K.
Output: Round keys $(k_0, \ldots, k_{24})$, MDS matrix $\boldsymbol{M}_R$, S-box S.
1: $(k_0, \ldots, k_{24}) = \mathrm{KDF}(K, 400)$; // Generate the round keys
2: **for** $i = 0$ to $2^{16} - 1$ **do**
3: $S(i) = \mathrm{MSPARX}\ (i, k_0, \ldots, k_{24})$; // Generate the S-box
4: **end for**
5: **for** $j = 0$ to 7 **do**
6: **while** $rc_j = 0$ **do**
7: $rc_j \leftarrow_R \{0,1\}^{16}$; // Randomly generate eight numbers
8: **end while**
9: **end for**
10: $\boldsymbol{M}_R = \mathrm{diag}(rc_0, \ldots, rc_7) \cdot \boldsymbol{M}_P$. //Generate the random MDS matrix

Algorithm 2 MSPARX: encryption algorithm of the 16-bit ARX block cipher mSPARX-16

Input: Plaintext x; round keys $(k_0, \ldots, k_{24})$.
Output: Ciphertext y.
1: $y = x \oplus k_0$; // Pre-whitening key
2: **for** $i = 1$ to 24 **do**
3: $y = A(y) \oplus k_i$; // Key alternating. A is the 16-bit ARX-box mSPECKEY
4: **end for**
5: Return y.

Algorithm 3 BBI: round-based black-box implementation of WARX

Input: Plaintext $\boldsymbol{x} = (x_0, x_1, \ldots, x_7)$; round keys $(k_0, \ldots, k_{24})$.
Output: Ciphertext $\boldsymbol{y} = (y_0, y_1, \ldots, y_7)$.
1: **for** $i = 1$ to 7 **do**
2: **for** $j = 0$ to 7 **do**
3: $y_j = \mathrm{MSPARX}(x_j, k_0, \ldots, k_{24})$; // Nonlinear layer
4: **end for**
5: $\boldsymbol{y} = (y_0, y_1, \ldots, y_7)$;
6: $\boldsymbol{y} = \boldsymbol{y} \cdot \boldsymbol{M}_R^{\mathrm{T}}$; // MDS matrix
7: $\boldsymbol{y} = \boldsymbol{y} \oplus (8(i-1)+1, 8(i-1)+2, \ldots, 8(i-1)+8)$; // Round constant addition
8: **end for**
9: Return $\boldsymbol{y}$.

Algorithm 4 WBI: LUT-based white-box implementation of WARX

Input: Ciphertext $\boldsymbol{y} = (y_0, y_1, \ldots, y_7)$; LUT T_{16} representing the 16-bit inverse S-box S^{-1}.
Output: Plaintext $\boldsymbol{x} = (x_0, x_1, \ldots, x_7)$.
1: **for** $i = 7$ to 1 **do**
2: $\boldsymbol{x} = \boldsymbol{y} \oplus (8(i-1)+1, 8(i-1)+2, \ldots, 8(i-1)+8)$; // Inverse of round constant addition
3: $\boldsymbol{x} = \boldsymbol{x} \cdot (\boldsymbol{M}_R^{-1})^{\mathrm{T}}$; // Inverse of MDS matrix
4: $\boldsymbol{x} = (T_{16}(x_0), T_{16}(x_1), T_{16}(x_2), T_{16}(x_3), T_{16}(x_4), T_{16}(x_5), T_{16}(x_6), T_{16}(x_7))$; // Inverse of nonlinear layer
5: **end for**
6: Return $\boldsymbol{x}$.

3 WARX: design rationale

3.1 Non-linear layer

The key-dependent S-box should be a standalone primitive with strong security because the white-box adversary can see its LUT. This primitive is expected to be more efficient than building blocks of prior

Table 1 Automatic search results for mSPECKEY with rotation constants (7,2). ODC is the optimal differential characteristic with input difference-output difference; OLT is the optimal linear trail with input mask-output mask

Rounds	1	2	3	4	5	6	7	8	9	10
ODCP	0	1	3	5	8	10	13	16	19	21
ODC	40,80-00,02	40,00-81,83	28,10-81,83	28,10-86,88	28,10-85,a7	48,d0-81,8b	48,50-74,a7	09,1a-74,a7	40,80-04,14	04,08-81,83
OLTC	0	0	1	2	3	4	6	7	9	10
OLT	00,10-40,40	80,41-04,04	a0,d1-04,04	8c,65-04,04	40,b0-c7,c6	80,79-8a,89	02,07-85,85	8c,56-c7,c6	80,71-1c,19	40,a8-04,04

white-box block ciphers, e.g., SPNbox-16. To this end, we consider implementing the 16-bit S-box by a 16-bit ARX block cipher, yet few block ciphers with the block size of 16 exist in the field of lightweight symmetric cryptography [29]. To the best of our knowledge, LAX-16 [22] is the only candidate that provides a 16-bit block size. However, the provable security of LAX against linear cryptanalysis remains an open problem. Therefore, a new ARX block cipher mSparx-16 is constructed, which has provable security bounds against current black-box attacks.

The structure of mSparx-16 is inspired by Sparx [22], the first provably secure ARX instance against differential and linear cryptanalysis. mSparx-16 uses a 16-bit ARX-box mSpeckey, whose security properties determine the number of rounds.

The construction of mSpeckey comes from scaling down Speck32 [21]. We performed an automatic search for the best differential characteristic and the linear trail of mSpeckey with different rounds under the independence assumption [30]. For the optimal differential characteristic probability and linear trail correlation, cryptoSMT [31] and MILP [32] were employed, respectively. To choose the rotation constants of mSpeckey, all possible pairs of constants (α, β) were tested where α and β is the right and left rotation constant, respectively. Let ODCP denote the optimal differential characteristic probability ($-\log_2$ scale) and OLTC denote the optimal linear trail correlation ($-\log_2$ scale). It was found that rotation constants (7,2), (7,3), and (5,6) stood out in respect of ODCP and OLTC. However, our further security analysis ruled out (5,6) (see Subsection 5.1 for more details). Finally, (7,2) was chosen due to the implementation cost. The ODCP/OLTC result for (7,2), as seen in Table 1, was verified by experiments. Appendix A provides the experiments result and ODCP/OLTC result for (7,3) and (5,6). From Table 1, at least 9 rounds are needed for mSpeckey to make ODCP and OLTC less than 2^{-16} and 2^{-8}, respectively. Therefore, mSparx-16 applies 24 rounds to provide a 62% ($\frac{24-9}{24}$) security margin.

3.2 Linear layer

For the linear layer, two methods were mainly considered, the random invertible binary matrix as in LowMC [33,34] and the random MDS matrix. The second method was chosen because of its lower sample complexity [35] and higher black-box security. To introduce randomness, the scalar multiplication class of an MDS matrix [36] is employed. Let $\boldsymbol{M}_p$ denote a $p \times p$ MDS matrix in GF(2^q). The scalar multiplication class of $\boldsymbol{M}_p$ is defined as multiplying a scalar value $V = (v_0, v_1, \ldots, v_p)$, where each v_i is a nonzero value in GF(2^q), by $\boldsymbol{M}_p$ with the following rule: v_i is multiplied by each element in the row i of $\boldsymbol{M}_p$.

3.3 Determining the number of rounds

The choice of the number of rounds was based on the best existing white-box attacks. For a better tradeoff between white-box security and efficiency, the minimal number of rounds necessary to reach the maximal achievable white-box security was considered. Section 5 provides a detailed analysis of white-box security of WARX, from which the lower bound of the number of rounds is deduced as 7. Finally, the number of rounds was set as 7 to maximize the efficiency of the white-box implementation.

3.4 Key schedule

The choice of KDF as the key schedule was motivated by security consideration. An important cryptographic criterion for key schedules in key alternating ciphers is that the round keys are mutually independent and random. From the perspective of implementation, it is assumed that a large amount of data is encrypted under the same key and the key schedule is called only rarely. In this case, the key schedule can be neglected [37]. Also, it is reasonable to assume that a server has enough resources to support the computation power a KDF such as SHAKE128 [28] requires. Moreover, efficient key schedule functions achieving fast key mixing for small-size block ciphers are unknown to the best of our knowledge.

4 Security evaluation in black-box model

In the black-box model, the adversary can only observe the input-output of the given key-fixed black-box implementation of WARX. The adversary's goal is to recover the round keys or the secret components, i.e., the secret S-box and MDS matrix.

Brute force attacks. A naive key recovery attack is to exhaustively search the round keys in the nonlinear layer, which are $25 \times 16 = 400$ bits, and search the MDS matrix in the linear layer, which is from the scalar multiplication class of a pre-defined MDS matrix. The scalar multiplication class $\mathcal{SMC}$ of a $p \times p$ MDS matrix has order $O_{\mathcal{SMC}} = (O(\mathrm{GF}(2^q)) - 1)^p$, where $O(\mathrm{GF}(2^q))$ is the order of the base field. For WARX, the base field is $\mathrm{GF}(2^{16})$. Therefore, the time complexity is estimated as $2^{400+\log_2[(2^{16}-1)^8]}$. The second attack in this scope is to exhaustively guess the 128-bit master key, then generate the round keys, and then guess the MDS matrix as mentioned before. The time complexity is $2^{128+\log_2[(2^{16}-1)^8]}$. Another attack is to guess the secret S-box directly without guessing the round keys, which has time complexity $2^{16\times 2^{16}+\log_2[(2^{16}-1)^8]1)}$. All in all, these attacks have impractical time complexity.

Meet-in-the-middle attacks. The meet-in-the-middle (MITM for short) approach can be used to reduce the time complexity of an exhaustive search [38]. The idea of a basic MITM attack is to split a cipher into two parts such that the subkeys in both parts can be guessed independently. In WARX, the nonlinear layer of each round involves 25 round keys, i.e., the whole 128-bit master key. In this case, the use of key bits is not biased among rounds. Therefore, MITM cannot give the adversary more convenience.

Differential/Linear cryptanalysis. The most powerful attacks for block ciphers tend to be differential [39] and linear cryptanalysis [40]. Differential cryptanalysis employs differential propagation with probability significantly higher than 2^{1-n} to attack an n-bit cipher, while linear cryptanalysis employs linear propagation with correlation significantly higher than $2^{n/2}$ to attack an n-bit cipher. According to the search result for the best differential characteristic and linear trail of mSpeckey in Subsection 3.1, the optimal differential characteristic probability and linear trail correlation of 9-round mSparx-16 are 2^{-19} and 2^{-9}, which are lower than 2^{-15} and 2^{-8}, respectively. So it is assumed that the maximum differential probability and linear correlation of mSparx-16 are 2^{-15} and 2^{-8}. Since the MDS matrix is employed in the linear layer, it is trivial to follow the wide trail argument [25] to derive the security bound against differential and linear cryptanalysis. For WARX, there are at least 9 active S-boxes after 2 rounds due to the 8×8 MDS matrix in the linear layer. Therefore, full-round WARX is secure against differential and linear cryptanalysis.

Structural attacks. Structural attack [41], also known as integral attack [42] or square attack [43], applies to ciphers with unknown or key-dependent internal functions, because such attacks focus on the syntactic interaction between different building blocks of a cipher, but ignore their concrete definition. Note that 2.5-round WARX, i.e., two rounds plus the nonlinear layer of the third round, is rather weak. Utilizing the technique in [41], the secret S-box and the MDS matrix can be recovered with complexities of only 2^{29} chosen plaintexts and 2^{48} time. However, no useful structural attacks targeting full-round WARX were found.

Decomposition attacks. WARX can be formally described as an 'SASASASASASASA' structure where 'S' denotes a non-linear layer composed of S-boxes and 'A' an affine layer. In WARX, both A and S layers are secret. For such secret SPN, decomposition attacks exist [44–46]. According to [44–46], the degree of an 'SA' structure is upper bounded by $n - \lceil \frac{n-\deg(\mathrm{A})}{m-1} \rceil$, where n and m are block size and S-box size, respectively, and $\deg(\mathrm{A})$ is the degree of the affine layer, which is 1. For WARX, the 16-bit S-box is modeled as a PRP, hence its algebraic degree is bounded by 15. This is reasonable because the S-box is constructed by the 16-bit block cipher mSparx-16. Therefore, the degree of the S-box is assumed as 15. Then the algebraic degree of 2-round WARX is upper bounded by $128 - \lceil \frac{128-15}{15} \rceil = 120$; and hence WARX reaches the maximum algebraic degree only after three rounds. Given the above, the complexity of generic decomposition attacks is very high.

Slide attacks. Iterated block ciphers with a high degree of self-similarity are vulnerable to slide attacks [47–49]. Such attacks are independent of the number of rounds of the cipher. WARX has the same S-box and the MDS matrix in each round, while the addition of round-dependent constants breaks the symmetry of the cipher. Therefore, general slide attacks are inapplicable.

1) Representing a random m-bit permutation requires $(m - 1.44)m$ bits, but the difference is only by a small multiplicative factor of about $1 - 1.44/m$.

Algebraic attacks. The idea of algebraic attack can be traced back to Shannon, who states that 'breaking a cipher should be as difficult as solving a system of simultaneous equations in a large number of unknowns of a complex type' [50]. The basic principle is to construct a system of multivariate polynomial equations from a cryptographic primitive such that solving this system can reveal some information of the key [51, 52]. Algebraic attacks do not threaten the security of WARX because of its high algebraic degree and complicated equation system.

5 Security evaluation in white-box model

Given the cipher's implementation, the goal of a white-box adversary is to extract the key, or to extract and copy the whole implementation code without recovering the secret key and use it in a standalone manner, which is called code lifting. Therefore, white-box security is composed of key extraction and code lifting security.

5.1 Key extraction security

In the white-box model, the adversary can see the LUT of the key-dependent S-box and the matrix of the linear layer, from which the adversary can get all plaintext-ciphertext pairs generated by mSparx-16 with regard to a fixed key in the black-box model. Therefore, key extraction can be reduced to the key recovery problem for mSparx-16 in the single key black-box model, and the adversary's advantage of key extraction in the white-box model is upper bounded by the advantage of key recovery for mSparx-16 in the black-box model. Therefore, we analyze the black-box security of mSparx-16.

Single trail differential/linear cryptanalysis. Because mSparx-16 is a key alternating cipher, the optimal differential characteristic probability and linear trail correlation (see Table 1) of 9-round mSparx-16 are 2^{-19} and 2^{-9}, respectively. Hence there is no differential/linear trail covering nine rounds with probability/correlation larger than $2^{-16}/2^{-8}$. Therefore, 24-round mSparx-16 is secure against single-trail differential/linear cryptanalysis. The difference/linear effect, i.e., the effect of trail clustering [25], is not considered, but it is expected that trail clustering does not greatly influence the security of the cipher. The effect of trail clustering in Speck and Sparx have been observed in [53, 54], and the results show that attacks exploiting this effect do not threaten security.

Integral cryptanalysis. mSparx-16 is a scaled-down version of Speck32 with a modified key addition. A 6-round integral distinguisher based on the division-property for mSparx-16 might exist [55]. This was verified with the automatic search tool Solvatore [56]. It was found that there always exists an integral distinguisher for rotation constants (5,6) for any number of rounds, but no 7-round distinguishers were found for rotation constants (7,2) and (7,3).

Algebraic attacks. Algebraic attacks pose a threat to the ciphers with a weak algebraic structure, especially for those with a low algebraic degree. In the A-box of mSparx-16, modular addition operation can be expressed by a vectorial boolean function from $\mathrm{GF}(2^{16})$ to $\mathrm{GF}(2^8)$ and the algebraic degree of this function is defined as the highest degree of its coordinates. According to the algebraic normal form of each coordinate of modular addition operation [57, 58], one modular addition has degree almost eight, which appears in the carry bit. This means the algebraic degree of one A-box is upper bounded by eight. By iterating the algebraic representation of modular addition, mSparx-16 reaches the full degree, i.e., 15, only after two rounds.

Other attacks. KDF is employed to preclude periodic sequence of round keys, e.g., $(k_1, k_2, k_1, k_2, \ldots)$, so mSparx-16 is secure against slide attacks. A simple MITM attack [38] that splits mSparx-16 into two sub-ciphers with every 12 rounds costs $2^{13\times 16}$ time complexity, but no MITM attacks on full-round mSparx-16 with significantly lower complexity were found. Besides, attacks in the related-key model such as rotational-xor attacks [59] are not applicable.

5.2 Code lifting security: space hardness bound

In space-hard designs, code lifting resistance is achieved by the incompressible white-box implementation such that the adversary cannot easily extract or copy the implementation code from the execution environment due to channel limitation. The security notion of (M, Z)-space hardness (see Definition 2) was proposed to quantitively measure code lifting resistance of a white-box block cipher.

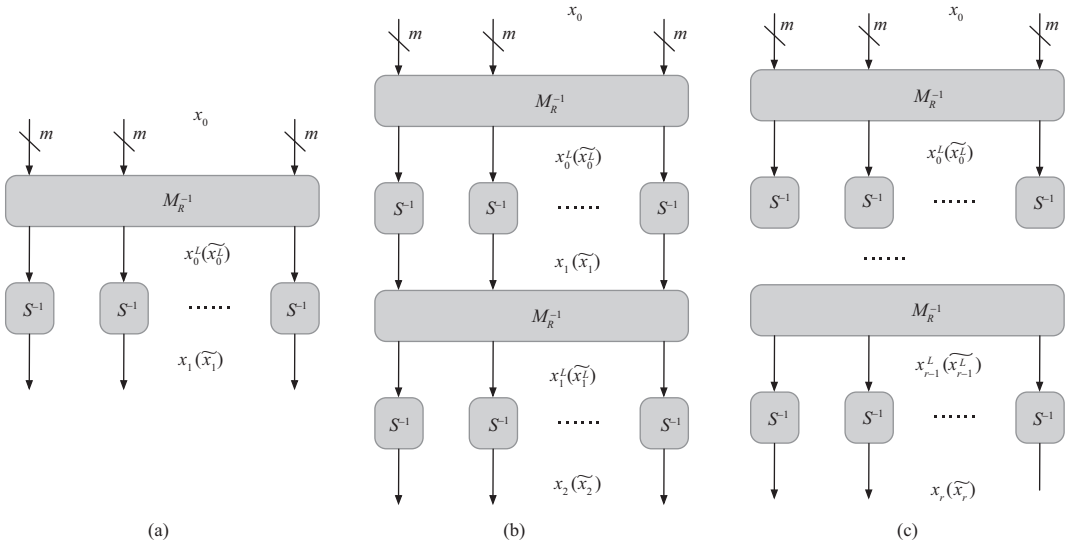

Figure 2 (a) 1-round case; (b) 2-round case; (c) r-round case.

Definition 2 ((M, Z)-space hardness, [16])**.** An implementation of a block cipher is said to have (M, Z)-space hardness if it is infeasible for an adversary to decrypt any randomly drawn ciphertext with the success probability greater than 2^{-Z}, given code of size less than M.

To calculate the space hardness of a cipher, all code compression attacks should be concerned. Because WARX employs a different structure with the former white-box block ciphers, it is beyond the scope that the current code compression attacks can cover. Therefore, we define new code compression attacks. According to the type of code that the adversary stores, code compression attacks are classified as non-linear layer, linear layer, cipher, and hybrid compression attack. In a non-linear layer compression attack, the adversary stores some entries of the LUT. In a linear layer compression attack, the adversary stores some constants of the linear layer matrix. In a cipher compression attack, the adversary stores some plaintext/ciphertext pairs. In a hybrid compression attack, the adversary stores anything among the three but with a proportion. Notice that although the linear layer occupies small space compared to the nonlinear layer and thus should be stored without limitation for the adversary, the situation that the adversary does not store the linear layer is still discussed. It will be shown that when the adversary only stores plaintext/ciphertext pairs with the given space, the success probability is maximal.

To conduct the analysis in a general way, WARX is parameterized as an (n, m, t, r) cipher with n representing the block size, m representing the S-box size, t representing the number of S-boxes in one nonlinear layer, and r representing the number of rounds. Then, the parameter $\lambda = \frac{M_a}{M_w}$ is introduced, where M_w is the memory size of the white-box implementation and M_a is the memory size given to the adversary. For WARX, $M_w = m2^m + t \cdot \log_2(2^m - 1)$ bits. For ease of comparison with current white-box block ciphers, we fix $\lambda = \frac{1}{4}$, so $M_a = m2^{m-2} + \frac{1}{4}t \cdot \log_2(2^m - 1)$ bits. Finally, we define the (λ, a, b, c)-hybrid compression attack, where $a/b/c$ denotes the proportion of the space that the adversary uses to store the entries/constants/pairs. Obviously, $a + b + c = 1$, and a non-linear layer/linear layer/cipher compression attack corresponds to a (1/4,1,0,0)/(1/4,0,1,0)/(1/4,0,0,1)-hybrid compression attack. To represent all kinds of attacks, we consider all possible combinations of a, b, and c, leading to three cases: $(1/4, 0, b, c)$, $(1/4, a, b, 0)$, and $(1/4, a, b, c)$. In all these cases, b can take any value such that $bM_a/m = 0, 1, 2, \ldots, t$ under the assumption that the adversary stores some constants completely rather partially.

For each type of attack, 1-round and 2-round cases are discussed and then the upper bound of the adversary's success probability is deduced for full-round WARX. The results are revealed in Theorems 1–3 as below. For a vector $\boldsymbol{v} = (v_0, v_1, \ldots, v_{t-1}) \in [\mathrm{GF}(2^m)]^t$, v_i is called a bundle, and it is called a zero bundle if $v_i = 0$. For the 1-round case, x_0, x_0^L, x_1 respectively denote the target ciphertext, intermediate value after the inverse linear layer, and plaintext (see Figure 2(a)). $\widetilde{x_0^L}$ and $\widetilde{x_1}$ denote the adversary's guesses for x_0^L and x_1, respectively. Similar notations are used for the 2- and r-round cases (Figures 2(b) and (c), respectively). Let $g_0, \ldots, g_{t-1}$ denote the adversary's guess for $rc_0^{-1}, \ldots, rc_{t-1}^{-1}$. In the next analysis, it is assumed that the input values of each table lookup are uniformly distributed.

Theorem 1. For a $(1/4, 0, b, c)$-hybrid compression attack under which the adversary can store $(cM_a)/n$

plaintext/ciphertext pairs and $(bM_a)/m$ constants of $\boldsymbol{M}_R^{-1}$, the upper bound of the adversary's success probability is a constant which is independent of r and only relies on n and m.

Proof. **1-round case.** Denote the set of stored plaintext/ciphertext pairs as $\boldsymbol{X}_s$. The adversary can directly decrypt if the target ciphertext is already stored, and otherwise can directly guess the plaintext or guess the inverse linear layer and missed entries of S^{-1}. Let p_1 and p_2 respectively denote the adversary's success probability by directly guessing the plaintext of the missed ciphertext and by guessing the matrix and the entries of the missed ciphertext, and then

$$\begin{aligned} p_1 &= \Pr[\widetilde{x_1} = x_1 | x_0 \in \boldsymbol{X}_s] \cdot \Pr[x_0 \in \boldsymbol{X}_s] + \Pr[\widetilde{x_1} = x_1 | x_0 \notin \boldsymbol{X}_s] \cdot \Pr[x_0 \notin \boldsymbol{X}_s] \\ &= 1 \cdot \frac{(cM_a)/n}{2^n} + \left(\frac{1}{2^n} - (cM_a)/n\right) \cdot \left(1 - \frac{(cM_a)/n}{2^n}\right). \end{aligned} \tag{1}$$

Let $\widetilde{S^{-1}}$ denote the adversary's guess for the missed entries of S^{-1}, and then

$$\begin{aligned} p_2 &= \Pr[\widetilde{x_1} = x_1 | x_0 \in \boldsymbol{X}_s] \cdot \Pr[x_0 \in \boldsymbol{X}_s] + \Pr[\widetilde{x_1} = x_1 | x_0 \notin \boldsymbol{X}_s]' \cdot \Pr[x_0 \notin \boldsymbol{X}_s] \\ &= 1 \cdot \frac{(cM_a)/n}{2^n} + \Pr[\widetilde{S^{-1}}(\widetilde{x_0^L}) = x_1] \cdot \left(1 - \frac{(cM_a)/n}{2^n}\right) \\ &= 1 \cdot \frac{(cM_a)/n}{2^n} + \left(\frac{1}{2^m}\right)^t \cdot \left(1 - \frac{(cM_a)/n}{2^n}\right). \end{aligned} \tag{2}$$

Obviously, $p_1 > p_2$. Therefore, the success probability is upper bounded by 2^{m-n} $(b = 0, c = 1)$.

2-round case. When the adversary directly guesses $\widetilde{x_2}$, the probability is the same as in the 1-round case. If the adversary guesses the missed constants and entries, the same analysis for the 1-round case holds, and $p_2 = 1 \cdot \frac{(cM_a)/n}{2^n} + \left(\frac{1}{2^m}\right)^{2t} \cdot \left(1 - \frac{(cM_a)/n}{2^n}\right)$. Again, the upper bound is determined by p_1. Hence, the probability is upper bounded by 2^{m-n}.

r-round case. As in the 2-round case, the success probability is upper bounded by 2^{m-n}, which is only determined by the block size and the S-box size. This means for an (n, m, t, r) white-box block cipher, its space hardness under the $(1/4, 0, 0, 1)$-hybrid compression attack is upper bounded by 2^{m-n}, regardless of the number of rounds. For WARX/SPNbox-16/WEM with $n = 128$, $m = 16$, this bound is 2^{-112}.

Theorem 2. For a $(1/4, a, b, 0)$-hybrid compression attack under which the adversary can store $(aM_a)/m$ entries of S^{-1} and $(bM_a)/m$ constants of $\boldsymbol{M}_R^{-1}$, the upper bound of the adversary's success probability is $[\frac{2^{m-2} - 3t(m-1)/4m + 1}{2^m}]^{rt}$.

Proof. **1-round case.** The adversary first guesses the missed parts of $\boldsymbol{M}_R^{-1}$, and then guesses the missed entries of S^{-1}. There is $\Pr[\widetilde{x_1} = x_1] = \Pr[\widetilde{x_1} = x_1 | \widetilde{x_0^L} = x_0^L] \cdot \Pr[\widetilde{x_0^L} = x_0^L] + \Pr[\widetilde{x_1} = x_1 | \widetilde{x_0^L} \neq x_0^L] \cdot \Pr[\widetilde{x_0^L} \neq x_0^L]$. It is easy to see that when $\widetilde{x_0^L} = x_0^L$, whether $\widetilde{x_1} = x_1$ depends on whether the looked entries of x_0^L are stored. If they are, then the probability of $\widetilde{x_1} = x_1$ is one, and if not, the adversary can still guess the missed entries of S^{-1}. So, $\Pr[\widetilde{x_1} = x_1 | \widetilde{x_0^L} = x_0^L] = [1 \cdot \frac{(aM_a)/m}{2^m} + \frac{1}{2^m - (aM_a)/m} \cdot (1 - \frac{(aM_a)/m}{2^m})]^t$.

The situation is more complicated when $\widetilde{x_0^L} \neq x_0^L$, because $\widetilde{x_0^L}$ might have some common bundles with x_0^L. For these common bundles, it is easy to determine the probability that the corresponding bundles of $\widetilde{x_1}$ and x_1 are equal, while for the other bundles, this probability depends on whether the looked entries of those bundles of $\widetilde{x_0^L}$ are stored. If so, then the probability is zero, and if not, then the adversary can still guess the missed entries and get the correct outputs of those uncommon bundles. Let $\#(\widetilde{x_0^L}, x_0^L)$ denote the number of common bundles in $\widetilde{x_0^L}$ and x_0^L, and then

$$\begin{aligned} \Pr[\widetilde{x_1} = x_1 | \widetilde{x_0^L} \neq x_0^L] = &\left[1 \cdot \frac{(aM_a)/m}{2^m} + \frac{1}{2^m - (aM_a)/m} \cdot \left(1 - \frac{(aM_a)/m}{2^m}\right)\right]^{\#(\widetilde{x_0^L}, x_0^L)} \\ &\cdot \left[0 \cdot \frac{(aM_a)/m}{2^m} + \frac{1}{2^m - (aM_a)/m} \cdot \left(1 - \frac{(aM_a)/m}{2^m}\right)\right]^{t - \#(\widetilde{x_0^L}, x_0^L)}. \end{aligned} \tag{3}$$

Here, $\#(\widetilde{x_0^L}, x_0^L)$ can have any value from 0 to $t-1$. $\forall i \in [0, 1, \ldots, t-1]$, $\Pr[\widetilde{x_0^L}_i = x_0^L{}_i] = \frac{1}{2^m}$. So, $\forall j \in [0, 1, \ldots, t-1]$, $\Pr[\#(\widetilde{x_0^L}, x_0^L) = j] = \binom{t-1}{j} \cdot \left(\frac{1}{2^m}\right)^j \cdot \left(1 - \frac{1}{2^m}\right)^{t-1-j}$ where $\binom{u}{v}$ denotes v combinations

of u. Therefore,

$$\Pr[\widetilde{x_1} = x_1 | \widetilde{x_0^L} \neq x_0^L] = \sum_{j=0}^{t-1} \binom{t-1}{j} \cdot \left(\frac{1}{2^m}\right)^j \cdot \left(1 - \frac{1}{2^m}\right)^{t-1-j} \cdot \left[1 \cdot \frac{(aM_a)/m}{2^m} + \frac{1}{2^m - (aM_a)/m} \cdot \left(1 - \frac{(aM_a)/m}{2^m}\right)\right]^j \cdot \left[\frac{1}{2^m - (aM_a)/m} \cdot \left(1 - \frac{(aM_a)/m}{2^m}\right)\right]^{t-j}. \quad (4)$$

Now, only $\Pr[\widetilde{x_0^L} = x_0^L]$ needs to be calculated. $\widetilde{x_0^L} = x_0^L$ iff $\boldsymbol{M}_P^{-1} \cdot \mathrm{diag}(g_0, \ldots, g_{t-1}) \cdot x_0 = x_0^L$, which holds iff $\mathrm{diag}(g_0, \ldots, g_{t-1}) \cdot x_0 = \boldsymbol{M}_P \cdot x_0^L$, because $\boldsymbol{M}_P^{-1}$ has full rank. Let $y_0^L = \boldsymbol{M}_P \cdot x_0^L$. Because $\mathrm{diag}(g_0, \ldots, g_{t-1}) \cdot x_0 = (g_0 \cdot x_{00}, \ldots, g_{t-1} \cdot x_{0\,t-1})$, we need $(g_0 \cdot x_{00}, \ldots, g_{t-1} \cdot x_{0\,t-1}) = y_0^L = (y_{0\,0}^L, \ldots, y_{0\,t-1}^L)$. $\forall i \in [0, 1, \ldots, t-1]$, the solution of g_i such that $g_i \cdot x_{0\,i} = y_{0\,i}^L$ depends on whether x_{0i} is a zero bundle. If $x_{0i} = 0$, then any nonzero constant in the base field $\mathrm{GF}(2^m)$ makes $g_i \cdot x_{0\,i} = y_{0\,i}^L$ hold, while if $x_{0i} \neq 0$, there is only one solution for g_i. Since the adversary has stored $(bM_a)/m$ constants of $\boldsymbol{M}_R^{-1}$, there are $t - (bM_a)/m$ constants remaining to be guessed. So, $\Pr[\widetilde{x_0^L} = x_0^L] = [1 \cdot \frac{1}{2^m} + \frac{1}{2^m - 1} \cdot (1 - \frac{1}{2^m})]^{t-(bM_a)/m}$. Naturally, $\Pr[\widetilde{x_0^L} \neq x_0^L] = 1 - (1 \cdot \frac{1}{2^m} + \frac{1}{2^m - 1} \cdot (1 - \frac{1}{2^m}))^{t-(bM_a)/m}$.

To sum up, there is

$$\Pr[\widetilde{x_1} = x_1] = [f(a)]^t \cdot [f(m)]^{t-(bM_a)/m} + \left[\sum_{j=0}^{t-1} \binom{t-1}{j} \cdot \left(\frac{1}{2^m}\right)^j \cdot \left(1 - \frac{1}{2^m}\right)^{t-1-j} \cdot [f(a)]^j \cdot [f'(a)]^{t-j}\right] \cdot [1 - [f(m)]^{t-(bM_a)/m}], \quad (5)$$

where $f(a) = 1 \cdot \frac{(aM_a)/m}{2^m} + \frac{1}{2^m - (aM_a)/m} \cdot (1 - \frac{(aM_a)/m}{2^m})$, $f'(a) = \frac{1}{2^m - (aM_a)/m} \cdot (1 - \frac{(aM_a)/m}{2^m})$, and $f(m) = 1 \cdot \frac{1}{2^m} + \frac{1}{2^m - 1} \cdot (1 - \frac{1}{2^m})$.

Let $E_{5\mathrm{R}}$ denote the right-hand-side expression of Eq. (5). We evaluated $E_{5\mathrm{R}}$ for all possible pairs of a and b and the upper bound appears when $(a, b) = (1 - \frac{t \cdot \log_2(2^m - 1)}{M_a}, \frac{t \cdot \log_2(2^m - 1)}{M_a})$, meaning the adversary stores the whole $\boldsymbol{M}_R^{-1}$. The bound is expressed by $E_{5\mathrm{R}} < [\frac{2^{m-2} - 3t(m-1)/4m + 1}{2^m}]^t$.

2-round case. The 2-round case differs from the 1-round case because the adversary guesses $\boldsymbol{M}_R^{-1}$ once in the first round and it is determinate in the second round. Following the same analysis as in the 1-round case, the probability that the target ciphertext can be correctly decrypted is calculated by $\Pr[\widetilde{x_2} = x_2] = \Pr[\widetilde{x_2} = x_2 | \widetilde{x_1} = x_1] \cdot \Pr[\widetilde{x_1} = x_1] + \Pr[\widetilde{x_2} = x_2 | \widetilde{x_1} \neq x_1] \cdot \Pr[\widetilde{x_1} \neq x_1]$.

When $\widetilde{x_1} = x_1$, $\Pr[\widetilde{x_2} = x_2] = \Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} = x_1^L] \cdot \Pr[\widetilde{x_1^L} = x_1^L] + \Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} \neq x_1^L] \cdot \Pr[\widetilde{x_1^L} \neq x_1^L]$. Again, it is easy to see that $\Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} = x_1^L] = [f(a)]^t$. When $\widetilde{x_1^L} \neq x_1^L$, whether $\widetilde{x_2} = x_2$ is unrelated to the number of rounds, so

$$\Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} \neq x_1^L] = \sum_{j=0}^{t-1} \binom{t-1}{j} \cdot \left(\frac{1}{2^m}\right)^j \cdot \left(1 - \frac{1}{2^m}\right)^{t-1-j} \cdot [f(a)]^j \cdot [f'(a)]^{t-j}. \quad (6)$$

Then only $\Pr[\widetilde{x_1^L} = x_1^L]$ remains to be determined. Here, $\widetilde{x_1^L} = x_1^L$ iff $\boldsymbol{M}_P^{-1} \cdot \mathrm{diag}(g_0, \ldots, g_{t-1}) \cdot x_1 = x_1^L$. Although $\mathrm{diag}(g_0, \ldots, g_{t-1})$ has been guessed and is determinate in the first round, whether $\widetilde{x_1^L} = x_1^L$ also depends on whether the guesses are correct, which is independent of r, so $\Pr[\widetilde{x_1^L} = x_1^L] = [f(m)]^{t-(bM_a)/m}$. Finally, it is natural to determine $\Pr[\widetilde{x_1^L} \neq x_1^L]$.

To sum up, $\Pr[\widetilde{x_2} = x_2 | \widetilde{x_1} = x_1] = \Pr[\widetilde{x_1} = x_1]$.

When $\widetilde{x_1} \neq x_1$, it is not easy to determine $\Pr[\widetilde{x_2} = x_2]$. Again, $\Pr[\widetilde{x_2} = x_2] = \Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} = x_1^L] \cdot \Pr[\widetilde{x_1^L} = x_1^L] + \Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} \neq x_1^L] \cdot \Pr[\widetilde{x_1^L} \neq x_1^L]$. First, $\Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} = x_1^L] = [f(a)]^t$. Then

$$\Pr[\widetilde{x_2} = x_2 | \widetilde{x_1^L} \neq x_1^L] = \sum_{j=0}^{t-1} \binom{t-1}{j} \cdot \left(\frac{1}{2^m}\right)^j \cdot \left(1 - \frac{1}{2^m}\right)^{t-1-j} \cdot [f(a)]^j \cdot [f'(a)]^{t-j}. \quad (7)$$

Now, only $\Pr[\widetilde{x_1^L} = x_1^L]$ remains to be determined. $\widetilde{x_1^L} = x_1^L$ iff $(g_0 \cdot \widetilde{x_{10}}, \ldots, g_{t-1} \cdot \widetilde{x_{1\,t-1}}) = y_1^L = (y_{1\,0}^L, \ldots, y_{1\,t-1}^L)$, where $y_1^L = \boldsymbol{M}_P \cdot x_1^L$. If g_i has been stored at the beginning for some $i \in [0, \ldots, t-1]$, then the probability that $g_i \cdot \widetilde{x_{1\,i}} = y_{1\,i}^L$ depends on whether $\widetilde{x_{1\,i}} = x_{1\,i}$, and this probability is $\frac{1}{2^m}$. However, if g_i was guessed in the first round, then the probability of $g_i \cdot \widetilde{x_{1\,i}} = y_{1\,i}^L$ depends on whether $\widetilde{x_{1\,i}}$ is a zero bundle, and this probability is $[f(m)]^{t-(bM_a)/m}$. Finally, it is natural to determine $\Pr[\widetilde{x_1^L} = x_1^L]$.

To sum up,

$$\begin{aligned}\Pr[\widetilde{x_2} = x_2 | \widetilde{x_1} \neq x_1] = {} & [f(a)]^t \cdot \left[\left(\frac{1}{2^m} \right)^{(bM_a)/m} \cdot [f(m)]^{t-(bM_a)/m} \right] \\ & + \left[\sum_{j=0}^{t-1} \binom{t-1}{j} \cdot \left(\frac{1}{2^m} \right)^j \cdot \left(1 - \frac{1}{2^m} \right)^{t-1-j} \cdot [f(a)]^j \cdot [f'(a)]^{t-j} \right] \\ & \cdot \left(1 - \left(\frac{1}{2^m} \right)^{(bM_a)/m} \cdot [f(m)]^{t-(bM_a)/m} \right). \end{aligned} \tag{8}$$

All in all, there is $\Pr[\widetilde{x_2} = x_2] = E_{5\mathrm{R}}^2 + E_{8\mathrm{R}} \cdot (1 - E_{5\mathrm{R}})$ where $E_{8\mathrm{R}}$ denotes the right-hand-side expression of Eq. (8). Again, the upper bound appears when $(a, b) = (1 - \frac{t \cdot \log_2(2^m - 1)}{M_a}, \frac{t \cdot \log_2(2^m - 1)}{M_a})$. In this case, $E_{8\mathrm{R}}$ is negligible and we regard $E_{5\mathrm{R}}^2$ as the upper bound of the success probability. This bound is expressed by $E_{5\mathrm{R}}^2 < [\frac{2^{m-2} - 3t(m-1)/4m+1}{2^m}]^{2t}$.

r-round case. From the analysis in the 2-round case, it is deduced that the success probability for the r-round case is upper bounded by $[\frac{2^{m-2} - 3t(m-1)/4m+1}{2^m}]^{rt}$. For WARX with $t = 8, m = 16$, this bound is 2^{-16r}.

Theorem 3. For a $(1/4, a, b, c)$-hybrid compression attack under which the adversary can store $(aM_a)/m$ entries of S^{-1}, $(bM_a)/m$ constants of $\boldsymbol{M}_R^{-1}$, and $(cM_a)/n$ plaintext/ciphertext pairs, the upper bound of the success probability is $[\frac{2^{m-2} - 3t(m-1)/4m+1}{2^m}]^{rt}$.

Proof. **1-round case.** The adversary can randomly guess the plaintext, or guess $\boldsymbol{M}_R^{-1}$ and the missed entries of S^{-1} for the target ciphertext if it is not stored. Let p_1' and p_2' respectively denote the adversary's success probability by directly guessing the plaintext of the missed ciphertext and by guessing the matrix and the entries of S^{-1} used by the missed ciphertext. Following the analysis in the $(1/4, 0, b, c)$-hybrid compression attack, $p_1' = 1 \cdot \frac{(cM_a)/n}{2^n} + (\frac{1}{2^n} - (cM_a)/n) \cdot (1 - \frac{(cM_a)/n}{2^n})$, while for p_2', the analysis in the $(1/4, a, b, 0)$-hybrid compression attack holds: $p_2' = 1 \cdot \frac{(cM_a)/n}{2^n} + E_{5\mathrm{R}} \cdot (1 - \frac{(cM_a)/n}{2^n})$. Because $E_{5\mathrm{R}} > (\frac{1}{2^n} - (cM_a)/n)$, $\Pr[\widetilde{x_1} = x_1]$ is upper bounded by p_2', which is further upper bounded by $E_{5\mathrm{R}}$.

2-round case. Following the analysis in the 1-round case, $\Pr[\widetilde{x_2} = x_2]$ is upper bounded by $E_{5\mathrm{R}}^2 + E_{8\mathrm{R}} \cdot (1 - E_{5\mathrm{R}})$. We regard $E_{5\mathrm{R}}^2$ as the bound, which is expressed by $E_{5\mathrm{R}}^2 < [\frac{2^{m-2} - 3t(m-1)/4m+1}{2^m}]^{2t}$.

r-round case. Following the analysis in the 2-round case, it is deduced that $\Pr[\widetilde{x_r} = x_r]$ is upper bounded by $[\frac{2^{m-2} - 3t(m-1)/4m+1}{2^m}]^{rt}$. For WARX with $t = 8$, $m = 16$, this bound is 2^{-16r}.

Putting Theorems 1–3 together, the constant bound under the $(1/4, 0, 0, 1)$-hybrid compression attack gives a generic maximal achievable space hardness when considering all code compression attacks. Building upon this, an expression can be formulated to derive the number of rounds needed for the desired 112-bit security level for WARX: $2^{-16r} \leqslant 2^{-112}$, from which we get $r \geqslant 7$. As the goal of WARX is to optimize efficiency, the number of rounds can be set as 7.

For white-box block ciphers without a random linear layer such as SPNbox-16 [17] and WEM [19], a similar analysis can be conducted. Given M_a size to the adversary who can only store some entries of the S-box and plaintext/ciphertext pairs, the upper bound of the adversary's success probability is 2^{-112} under the $(1/4, 0, 0, 1)$-hybrid compression attack and 2^{-15r} under the $(1/4, 1, 0, 0)$-hybrid compression attack. Therefore, their minimal number of rounds is determined by $2^{-15r} \leqslant 2^{-112}$, from which we deduce $r \geqslant 8$.

To sum up, by adopting a random linear layer, the minimal number of rounds required to achieve the maximal code lifting security level is improved by one round.

Table 2 Components of SPNbox-16, WEM, and WARX. AES-16 denotes the 16-bit AES used in SPNbox-16. r_i denotes the number of rounds of the small block cipher

Cipher	r	r_i	Components		
			Key schedule	Non-linear layer (S-box generation)	Linear layer
SPNbox-16 [17]	10	32	KDF	AES-16	MDS matrix
WEM [19]	12.5	–	AES	PRNG(AES-CTR) and Fisher-Yates shuffle	5-round AES-128
WARX	7	24	KDF	mSPARX-16	MDS matrix

Table 3 Software performance of BBIs. Performance on Intel is measured by cycles per byte. Cache-timing issues means whether the implementation is vulnerable to some potential cache-timing attacks

Cipher	Implementation method		Performance on Intel	Cache-timing issues
	Non-linear layer	Linear layer		
SPNbox-16 [17]	Arithmetic instructions	Givaro*	7821	YES
WEM [19]	LUT	Arithmetic instructions	970	YES
WARX	Arithmetic instructions	Givaro	863	NO

* A C++ library for arithmetic and algebraic computations for efficient finite field multiplication. https://github.com/linbox-team/givaro.

Table 4 Software performance of WBIs. Performance on Intel and performance on ARM are all measured by cycles per byte. Space hardness is under the best compression attack

Cipher	Implementation method		Performance on Intel	Performance on ARM	M_w	Space hardness
	Non-linear layer	Linear layer				
SPNbox-16 [17]	LUT	Givaro	778	11024	128 KB	(32 kB, 112)
WEM [19]	LUT	Arithmetic instructions	970	12586	128 KB	(32 kB, 112)
WARX	LUT	Givaro	572	8739	128 KB	(32 kB, 112)

6 Software performance and comparison

We mainly compare WARX with SPNbox-16 [17] and WEM [19], the two most competitive white-box block ciphers so far. All of the three ciphers have the SPN structure, but use different components, as shown in Table 2. To clarify the efficiency improvement, we implemented BBIs and WBIs of them (codes are available on the website[2)]). Considering scenarios such as cloud-based DRM and mobile payment, all BBIs were tested under the x86_64 architecture. Specifically, a laptop equipped with a 1.6-GHz Intel Core i5 CPU was chosen and the performance result is shown in Table 3. WBIs were tested under the same x86_64 and ARMv8 architectures, for which a QEMU 5.0 ARMv8 virtual machine with a Cortex-A53 CPU was built. The performance result is shown in Table 4. Performance of BBI is measured by taking the average over 100000 repetitions, each time encrypting a random message of 2048 bytes. Performance of WBI is measured similarly while decrypting a message of the same length.

From Tables 3 and 4, the black- and white-box implementations of WARX perform best among the three ciphers on both Intel and ARMv8 platforms. Especially, the BBI of WARX is about nine times faster than SPNbox-16. Obviously, for all the three ciphers, the table-based WBI outperforms the round-based BBI on the same platform. This result is consistent with [17]. We should mention that some implementation methods might introduce cache-timing issues. For example, a BBI with key-dependent table lookups for secret S-boxes might be vulnerable to cache-timing attacks; hence, its secret components might be recovered.

7 Conclusion

This paper proposes a new white-box block cipher, WARX, which is based on ARX primitives and random MDS matrix. Under the precondition of ensuring security, WARX accomplishes better black- and white-box efficiency than the existing space-hard white-box block ciphers. This design shows the effectivity of a random linear layer in reducing the number of rounds of an SPN-type white-box block cipher and thereby improving its efficiency. Our findings indicate several future research directions. First, it would be interesting to explore different random linear layers. In this case, the space hardness will perform differently, but we predict that the constant space hardness bound cannot be broken. Second, the space

2) https://github.com/JunLiu9102.

hardness model defined in this work also applies to other SPN-type white-box block ciphers. This means the constant space hardness bound exists for new designs following the SPN structure. Therefore, the number of rounds cannot be reduced further, also because the security under some black-box attacks still relies on the number of rounds. Finally, an open question is to search fast and secure key mixing functions for small-size block ciphers, which can further optimize the efficiency of the setup procedure in practice. It is worth noting that there exists a gap between theoretically secure and realistically imperative white-box schemes, while our work took a step towards closing this gap.

Acknowledgements This work was supported by the National Key R&D Program of China (Grant No. 2017YFB0802000), National Natural Science Foundations of China (Grant Nos. 61672412, 61972457, 61902303, U19B2021), National Cryptography Development Fund of China (Grant Nos. MMJJ20170104, MMJJ20180219), China Scholarship Council (Grant No. 201806960067), Key Research and Development Program of Shaanxi (Grant No. 2020ZDLGY08-04), and Natural Science Basic Research Program of Shaanxi (Grant No. 2020JQ-832). We thank Adrián Ranea for fruitful discussions and reviewers for their effort.

References

1 Chow S, Eisen P, Johnson H, et al. White-box cryptography and an AES implementation. In: Proceedings of the 9th International Workshop on Selected Areas in Cryptography, 2002. 250–270
2 Chow S, Eisen P, Johnson H, et al. A white-box DES implementation for DRM applications. In: Proceedings of ACM CCS-9 Workshop on Digital Rights Management, 2002. 1–15
3 Bringer J, Chabanne H, Dottax E. White box cryptography: another attempt. 2006. https://eprint.iacr.org/2006/468.pdf
4 Xiao Y Y, Lai X J. A secure implementation of white-box AES. In: Proceedings of the 2nd International Conference on Computer Science and Its Applications, 2009. 1–6
5 Karroumi M. Protecting white-box AES with dual ciphers. In: Proceedings of the 13th International Conference on Information Security and Cryptology, 2010. 278–291
6 Billet O, Gilbert H, Ech-Chatbi C. Cryptanalysis of a white box AES implementation. In: Proceedings of the 11th International Workshop on Selected Areas in Cryptography, 2004. 227–240
7 Lepoint T, Rivain M, de Mulder Y, et al. Two attacks on a white-box AES implementation. In: Proceedings of the 20th International Workshop on Selected Areas in Cryptography, 2013. 265–285
8 Wyseur B, Michiels W, Gorissen P, et al. Cryptanalysis of white-box DES implementations with arbitrary external encodings. In: Proceedings of the 14th International Workshop on Selected Areas in Cryptography, 2007. 264–277
9 de Mulder Y, Wyseur B, Preneel B. Cryptanalysis of a perturbated white-box AES implementation. In: Proceedings of the 11th International Conference on Cryptology, 2010. 292–310
10 de Mulder Y, Roelse P, Preneel B. Cryptanalysis of the Xiao-Lai white-box AES implementation. In: Proceedings of the 19th International Workshop on Selected Areas in Cryptography, 2012. 34–49
11 Michiels W, Gorissen P, Hollmann H D L. Cryptanalysis of a generic class of white-box implementations. In: Proceedings of the 15th International Workshop on Selected Areas in Cryptography, 2008. 414–428
12 Derbez P, Fouque P A, Lambin B, et al. On recovering affine encodings in white-box implementations. IACR Trans Cryptogr Hardw Embed Syst, 2018, 3: 121–149
13 Biryukov A, Bouillaguet C, Khovratovich D. Cryptographic schemes based on the ASASA structure: black-box, white-box, and public-key (extended abstract). In: Proceedings of the 20th International Conference on the Theory and Application of Cryptology and Information Security, 2014. 63–84
14 Gilbert H, Plût J, Treger J. Key-recovery attack on the ASASA cryptosystem with expanding s-boxes. In: Proceedings of the 35th Annual Cryptology Conference, 2015. 475–490
15 Minaud B, Derbez P, Fouque P A, et al. Key-recovery attacks on ASASA. J Cryptol, 2018, 31: 845–884
16 Bogdanov A, Isobe T. White-box cryptography revisited: space-hard ciphers. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015. 1058–1069
17 Bogdanov A, Isobe T, Tischhauser E. Towards practical whitebox cryptography: optimizing efficiency and space hardness. In: Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security, 2016. 126–158
18 Fouque P A, Karpman P, Kirchner P, et al. Efficient and provable white-box primitives. In: Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security, 2016. 159–188
19 Cho J, Choi K Y, Dinur I. WEM: a new family of white-box block ciphers based on the Even-Mansour construction. In: Proceedings of the Cryptographers' Track at the RSA Conference, 2017. 293–308
20 Lin T T, Lai X J, Xue W J, et al. A new Feistel-type white-box encryption scheme. J Comput Sci Technol, 2017, 32: 386–395
21 Beaulieu R, Shors D, Smith J, et al. The SIMON and SPECK families of lightweight block ciphers. 2013. https://eprint.iacr.org/2013/404.pdf
22 Dinu D, Perrin L, Udovenko A, et al. Design strategies for ARX with provable bounds: SPARX and LAX. In: Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security, 2016. 484–513
23 Beierle C, Biryukov A, Santos L C, et al. Alzette: a 64-bit ARX-box (feat. CRAX and TRAX). In: Proceedings of the 40th Annual International Cryptology Conference, 2020. 419–448
24 Biryukov A, Velichkov V, Le Corre Y. Automatic search for the best trails in ARX: application to block cipher SPECK. In: Proceedings of the 23rd International Conference on Fast Software Encryption, 2016. 289–310
25 Daemen J, Rijmen V. The wide trail design strategy. In: Proceedings of the 8th IMA International Conference on Cryptography and Coding, 2001. 222–238
26 Barreto P, Rijmen V. The Khazad legacy-level block cipher. NESSIE Project, 2000. https://www.researchgate.net/publication/228924670_The_Khazad_legacy-level_block_cipher
27 National Institute of Standards and Technology. Recommendation for key derivation using pseudorandom functions. NIST SP 800-108. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-108.pdf
28 National Institute of Standards and Technology. SHA-3 standard: permutation-based hash and extendable-output functions. FIPS PUB 202. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
29 Biryukov A, Perrin L. State of the art in lightweight symmetric cryptography. 2017. https://eprint.iacr.org/2017/511.pdf

30 Lai X J, Massey J, Murphy S. Markov ciphers and differential cryptanalysis. In: Proceedings of the 10th Workshop on the Theory and Application of Cryptographic Techniques, 1991. 17–38
31 Kölbl S, Leander G, Tiessen T. Observations on the SIMON block cipher family. In: Proceedings of the 35th Annual Cryptology Conference, 2015. 161–185
32 Fu K, Wang M Q, Guo Y H, et al. MILP-based automatic search algorithms for differential and linear trails for SPECK. In: Proceedings of the 23rd International Conference on Fast Software Encryption, 2016. 268–288
33 Albrecht M, Rechberger C, Schneider T, et al. Ciphers for MPC and FHE. In: Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015. 430–454
34 Dinur I, Kales D, Promitzer A, et al. Linear equivalence of block ciphers with partial non-linear layers: application to LowMC. In: Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019. 343–372
35 Randall D. Efficient generation of random nonsingular matrices. Random Struct Alg, 1993, 4: 111–118
36 Murtaza G, Ikram N. Direct exponent and scalar multiplication classes of an MDS matrix. 2011. https://eprint.iacr.org/2011/151.pdf
37 Daemen J, Rijmen V. The Design of Rijndael: the Advanced Encryption Standard (AES). Berlin: Springer, 2020
38 Diffie W, Hellman M E. Special feature exhaustive cryptanalysis of the NBS data encryption standard. Computer, 1977, 10: 74–84
39 Biham E, Shamir A. Differential Cryptanalysis of the Data Encryption Standard. Berlin: Springer, 1993
40 Matsui M. Linear cryptanalysis method for DES cipher. In: Proceedings of the 12th Workshop on the Theory and Application of Cryptographic Techniques, 1993. 386–397
41 Biryukov A, Shamir A. Structural cryptanalysis of SASAS. In: Proceedings of the 20th International Conference on the Theory and Application of Cryptographic Techniques, 2001. 395–405
42 Knudsen L, Wagner D. Integral cryptanalysis (extended abstract). In: Proceedings of the 9th International Workshop on Fast Software Encryption, 2002. 112–127
43 Daemen J, Knudsen L, Rijmen V. The block cipher square. In: Proceedings of the 4th International Workshop on Fast Software Encryption, 1997. 149–165
44 Biryukov A, Khovratovich D. Decomposition attack on SASASASAS. 2015. https://eprint.iacr.org/2015/646.pdf
45 Perrin L. Cryptanalysis, reverse-engineering and design of symmetric cryptographic algorithms. Dissertation for Ph.D. Degree. Luxembourg: University of Luxembourg, 2017
46 Boura C, Canteaut A, Cannière C. Higher-order differential properties of KECCAK and Luffa. In: Proceedings of the 18th International Workshop on Fast Software Encryption, 2011. 252–269
47 Biryukov A, Wagner D. Slide attacks. In: Proceedings of the 6th International Workshop on Fast Software Encryption, 1999. 245–259
48 Biryukov A, Wagner D. Advanced slide attacks. In: Proceedings of the 19th International Conference on the Theory and Application of Cryptographic Techniques, 2000. 589–606
49 Bar-On A, Biham E, Dunkelman O, et al. Efficient slide attacks. J Cryptol, 2018, 31: 641–670
50 Shannon C E. Communication theory of secrecy systems. Bell Syst Tech J, 1949, 28: 656–715
51 Courtois N T, Pieprzyk J. Cryptanalysis of block ciphers with overdefined systems of equations. In: Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security, 2002. 267–287
52 Albrecht M. Algorithmic algebraic techniques and their application to block cipher cryptanalysis. Dissertation for Ph.D. Degree. London: Royal Holloway, University of London, 2010
53 Ankele R, List E. Differential cryptanalysis of round-reduced SPARX-64/128. In: Proceedings of the 16th International Conference on Applied Cryptography and Network Security, 2018. 459–475
54 Ankele R, Kölbl S. Mind the GAP-a closer look at the security of block ciphers against differential cryptanalysis. In: Proceedings of the 25th International Conference on Selected Areas in Cryptography, 2018. 163–190
55 Sun L, Wang W, Wang M Q. Automatic search of bit-based division property for ARX ciphers and word-based division property. In: Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security, 2017. 128–157
56 Eskandari Z, Kidmose A B, Kölbl S, et al. Finding integral distinguishers with ease. In: Proceedings of the 25th International Conference on Selected Areas in Cryptography, 2018. 115–138
57 Sun L, Wang W, Liu R, et al. MILP-aided bit-based division property for ARX ciphers. Sci China Inf Sci, 2018, 61: 118102
58 Braeken A, Semaev I. The ANF of the composition of addition and multiplication mod 2^n with a boolean function. In: Proceedings of the 12th International Workshop on Fast Software Encryption, 2005. 112–125
59 Liu Y W, de Witte G, Ranea A, et al. Rotational-XOR cryptanalysis of reduced-round SPECK. IACR Trans Symmetric Cryptol, 2017, 3: 24–36

Appendix A Experiments for Subsection 3.1

We used cryptoSMT to search for the best differential probability of the ARX-box mSPECKEY, and MILP to search for the linear correlation. We modified the word size to 8 and ran the tools for all possible pairs of rotation constants. After that, we verified the differential probability and linear correlation using 10000 random keys and 10000 pairs of plaintext for the rotation constant (7,2). For the practical probability, we took the average value. When the number of rounds is low, the verified differential probability/linear correlation is close to the result returned by the tools, while for higher numbers of rounds, the verified probability differs more because of the trail clustering effect. See Tables A1 and A2 for the automatic search results of rotation constants (7,3) and (5,6). See Table A3 for the verification result of differential probability and linear correlation.

Table A1 Automatic search results (extra) of rotation constant (7,3)

Rounds	1	2	3	4	5	6	7	8	9	10
ODCP	0	1	3	6	9	12	15	18	20	22
ODC	40,80-00,04	40,80-04,24	24,08-81,85	84,03-85,a1	35,68-c2,d0	42,80-71,7e	72,44-81,85	40,80-20,21	01,02-80,84	01,02-c2,d0
OLTC	0	0	1	2	4	5	7	8	9	10
OLT	00,10-80,80	80,21-08,08	80,21-d8,c0	80,21-3c,0d	98,15-73,7a	82,bd-ac,8d	10,24-e9,e9	00,a3-ac,8d	86,a8-0c,0c	04,0d-08,08

Table A2 Automatic search results (extra) of rotation constant (5,6)

Rounds	1	2	3	4	5	6	7	8	9	10
ODCP	0	1	3	5	9	11	13	16	19	22
ODC	10,80-00,20	10,00-84,a4	1a,40-84,a4	0a,40-80,a9	0a,40-a5,cf	20,01-20,28	94,84-20,28	94,84-29,23	9a,40-29,23	10,80-02,a6
OLTC	0	0	1	3	4	6	6	8	9	11
OLT	00,10-04,04	00,80-49,48	20,05-92,90	20,9d-21,00	20,95-94,84	2a,37-6d,6c	20,05-49,48	e6,31-49,48	20,cd-6d,6c	06,f0-e4,24

Table A3 Verification result for differential probability and linear correlation of mSPECKEY (all $-\log_2$ scale)

Rounds	1	2	3	4	5	6	7	8	9	10
Differential probability (cryptoSMT)	0	1	3	5	8	10	13	16	19	21
Differential probability (verify)	0	1.0	3.0	5.0	8.0	10.0	12.8	15.0	15.0	15.8
Linear correlation (MILP)	0	0	1	2	3	4	6	7	9	10
Linear correlation (verify)	0	0	7.6	2.0	3.0	4.0	5.7	6.6	6.9	6.8