Performance Improvement of White-Box Algorithms Using Lightweight Designs

Hatice Kübra Güner, Ceyda Mangır, **Oğuz Yayla**

Cryptography Department
Institute of Applied Mathematics
Middle East Technical University
Ankara

February 23, 2023

Motivation

The motivation behind this study is finding an efficient implementation for (M,Z)-space hard white-box algorithms. With this purpose, we have studied linear lightweight components to find suitable one in the white-box settings.

Aim of White-Box Cryptography

- White-box cryptography suggests software protection using an appropriate method to hide the key in the algorithm phases.
- The key is generally embedded into the confusion layer with suitable methods.

Security Considerations

- Unbreakability
- One-wayness
- Tracibility
- Incompressibility

Security Considerations

Incompressibility is stated as one of the primitive properties of the white-box algorithms to **prevent extraction of the algorithm** from the device.

Space-Hard Ciphers

- Space-hard ciphers has been proposed by Bogdanov and Isobe in 2016.
- In this structure, large lookup tables, constructed with a small block cipher, are used to embed secret keys.
- The constructed tables are used as a nonlinear layer in the algorithm.

SPNbox

Bogdanov, A., Isobe, T., Tischhauser, E. (2016). **Towards Practical Whitebox Cryptography: Optimizing Efficiency and Space Hardness**. In: Cheon, J., Takagi, T. (eds) Advances in Cryptology ASIACRYPT 2016. ASIACRYPT 2016. Lecture Notes in Computer Science(), vol 10031. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-53887-6_5

SPNbox

- The SPNbox aims to improve performance against Feistel based space-hard cipher Space by taking advantage of parallelism and SIMD instructions.
- It is based on SPN structure using key-dependent lookup table T in the nonlinear layer.
- The T table is generated using a small scale block cipher with AES components.

WARX

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

WARX

- The motivation behind WARX is that existing white-box implementations run slowly or need large storage space.
- ARX (Addition/Rotation/XOR) approach is used to generate the lookup table used in the nonlinear layer.
- Speeding up the runtime performance of the white-box application by reducing the number of rounds by one according to SPNbox-16 with the **random MDS matrix** recommendation.

WBI of WARX

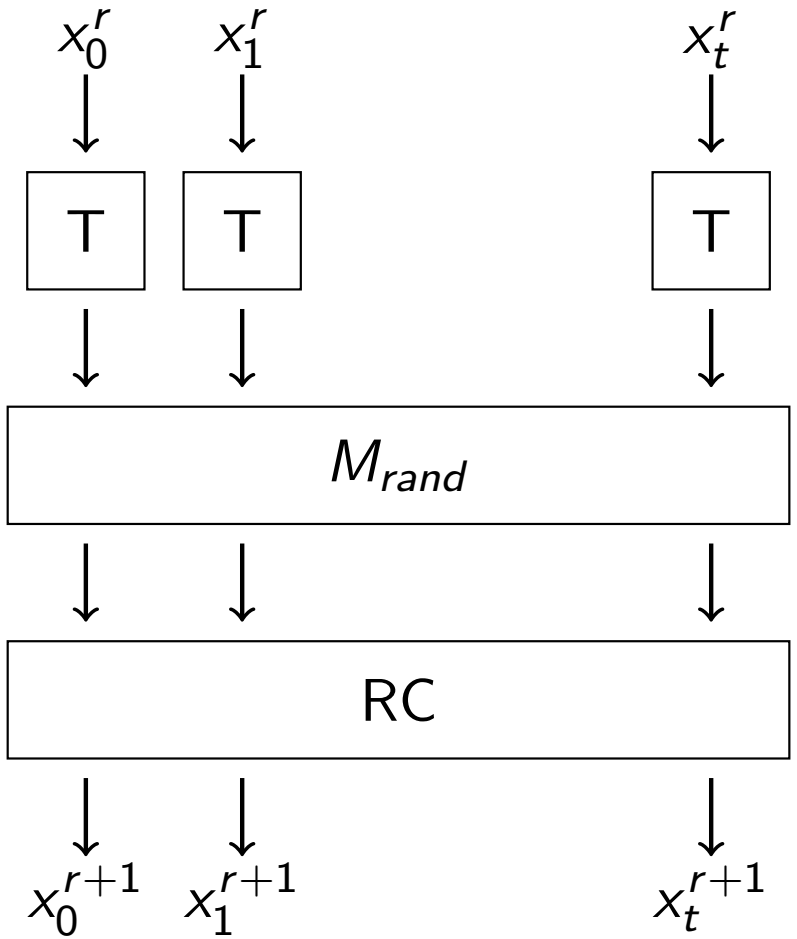

Figure 1: WBI of WARX

Table Construction of WARX

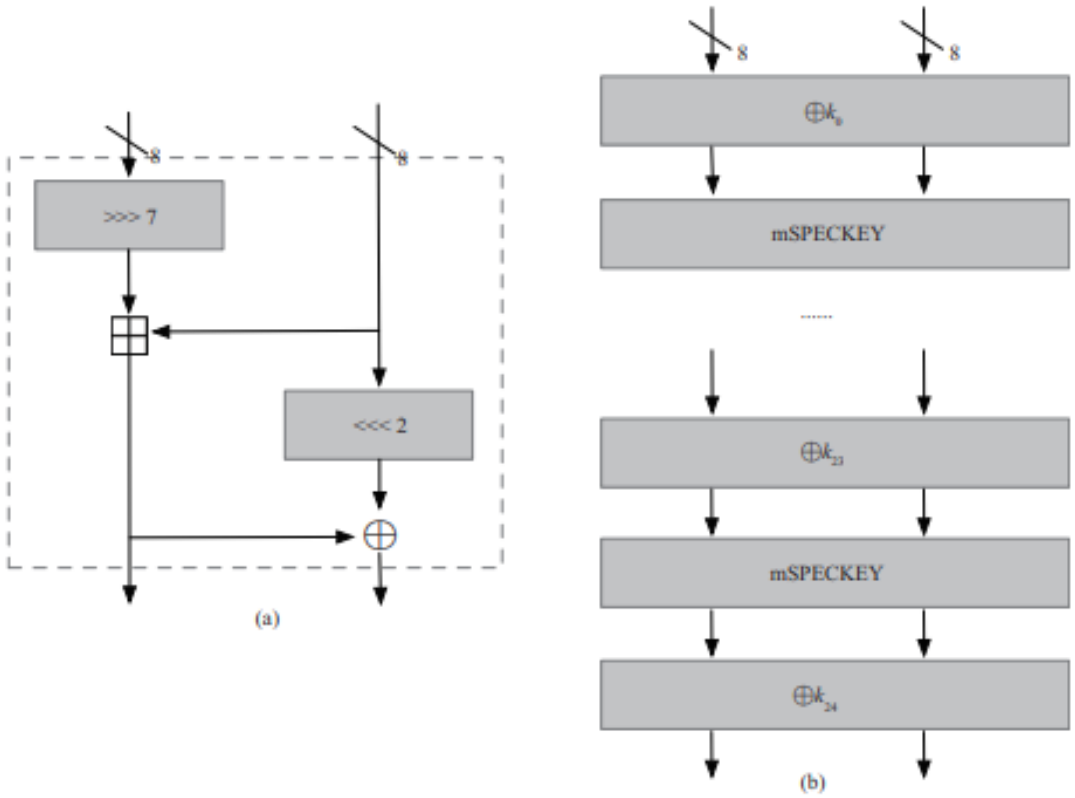

Figure 2: Table of WARX

WBI of WARX

$$M_{rand} = diag(r_{c_0}, ..., r_{c_7}) \cdot MDS$$

$$M_{rand} = diag(1, 2, 3, 4, 5, 6, 7, 8) * KHAZAD \in GF(2^{16})$$

Consequences

Important points are:

- Creating a secure table.
- Updating the table in the client side.
- Accelerating the white-box and black-box implementations.

White-Box Conversions

- The linear layers of NIST Lightweight competition finalists and second round candidates are examined to evaluate suitable designs for (M,Z)-space hard white-box algorithms.
- S-box, key addition and key schedule layers are discarded if they exists in the design.
- The lookup table is used as a substitution box in the nonlinear layer, while the linear component of the lightweight design is used as a linear layer in the white-box implementation.
- The tables are created using the WARX method for 16-bit word size algorithms and the SPNbox method for 32-bit word size algorithms.

Linear Layers of White-Box Conversions

- In White-box Saturnin, each round uses the **MDS matrix** to provide linearity, while *SR_slice* is applied to each odd round and *SR_sheet* to each even round.
- White-box Sparkle is based on ARX design. The ARX design uses the LTS approach, based on the powerful built-in Sbox and the use of a linear layer with reduced computational cost.
- The bitslicing is used in the linear layer of white-box Shadow to prevent side channel analysis.

White-Box Security

- Key extraction security
- Code lifting security

Key Extraction Security

- A block cipher is used to construct lookup tables in order to prevent key extraction attacks.
- White-box key extraction problem turns into black-box key recovery problem.

Code Lifting Security

Code lifting attack

Attacker uses the original implementation as a large secret key for encryption and decryption on a different device.

- Incompressible tables are needed to prevent code lifting attacks.
- Code lifting security is defining with weak (M,Z)-space hardness.
- In this approach, **the table needs to be updated when the leak size reaches the defined limit.**

Space-Hard Ciphers

Weak (M,Z)-space hardness

A white-box block cipher is called **weak (M, Z)-space hard** if **it is not possible to encrypt/decrypt** a randomly selected text **with a probability greater than** 2^{-Z} until the size of **the leakage from the code (table) is reached to M bits**.

White-Box Security

$$p = \left(\frac{M}{T} + \left(\frac{1}{T-M}\right) \cdot \left(1 - \frac{M}{T}\right)\right)^{r \cdot t}$$

- As in WARX, correctly guessing the corresponding entry of the table is included if the entry is not located in the leaked part of the lookup table.
- Security size is limited to $M \cdot 2^{-keysize}$ instead of $T \cdot 2^{-keysize}$ to make more precise computations for round numbers.
- Round numbers of the conversions are computed according to the white-box security criteria.

WBI Results

Table 1: Performance results of the white-box implementations.

Algorithm	Key Size (bit)	Round	MAS (bit)	Cycle (per byte)	Code Size (KB)
WARX	128	7	114	304	4.8
Saturnin	256	8	242	33	6.1
SPNbox-32	128	10	98	942	2.2
Sparkle	256	15	226	93	5
Shadow	384	15	354	120	6.6

WBI Results

- White-box Saturnin is almost twelve times faster than the WARX.
- White-box Sparkle is ten times faster than SPNbox-32 with 226-bit security.
- The white-box Shadow is seven times faster than SPNbox-32 with a 354-bit maximum achievable security level.
- White-box implementation of Sparkle32 is 22% faster than the Shadow256.

BBI Results

Table 2: Performance results of the 16-bit word size black-box implementations.

Algorithm	Key Size (bit)	Round	Cycle (per byte)	Code Size (KB)
WARX	128	7	498	6.1
Saturnin	256	8	422	10.9
SPNbox-32	128	10	73695	3.2
Sparkle	256	15	115079	9.7
Shadow	384	15	116355	12.3

BBI Results

- The black-box Saturnin is 15% faster than the WARX.
- The black-box Sparkle is 1% faster than Shadow.
- Nevertheless, both of the black-box implementations are slower than SPNbox-32.

Table Leakage

Table 3: Table leakage size for 2^{-114} and 2^{-98} success probability.

Algorithm	Security Size (bit)	Table Size (T)	Leak Limit (bit)
WARX	114	128 KB	$T/2^{2}$
Saturnin	114	128 KB	$T/2^{0.89}$
Sparkle-16	114	128 KB	$T/2^{0.89}$
SPNbox-32	98	16 GB	$T/2^{2.45}$
Sparkle-32	98	16 GB	$T/2^{0.82}$
Shadow256	98	16 GB	$T/2^{0.82}$
Shadow384	98	16 GB	$T/2^{0.54}$

Table Leakage

- Shadow384's block size is larger than the others, providing the same level of security until a larger table leak.
- The table leak size of SPNbox is calculated as $T/2^{2.45}$ for the 98-bit security level.

Design of the Table

- The LS-design is a block cipher aimed at preventing side-channel attacks with bitslicing implementation.
- The 4-bit S-box is taken from Spook design. The linear layer L-box is taken from *Mysterion* algorithm.
- The S-box is applied to 4-bit columns, while the L-box is applied to 8-bit rows.
- The algorithm consists of 32 rounds to provide 256-bit security.

Design of the Table

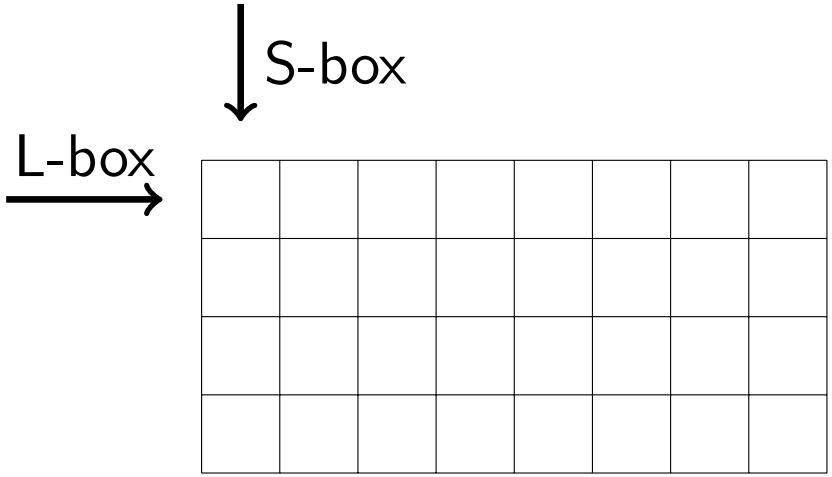

Figure 3: State of the Input

Design of the Table

- The differential and linear probability of S-box is 2^{-2}, while the algebraic degree is 3.
- The L-box is a recursive MDS matrix obtained from an [16,8,9] MDS code with the branch number 9.
- The 8×8 MDS matrix is M = [0,8,3,f,5,f,3,8] related reduction polynomial is p=0x13.

Security Considerations

- The LS-design is based on WTS approach.
- Differential and linear probability of the S-box is 2^{-2} and branch number of L-box is 9.

$$Pr_{diff}(2r) \leq Pr_{diff}^{max}(S)^{r \cdot B(L)}$$

$$Pr_{lin}(2r) \leq Pr_{lin}^{max}(S)^{r \cdot B(L)}$$

Security Considerations

- The table generation method provides 128-bit security after 14 rounds and 256-bit security after 28 rounds.
- There are 63 active S-boxes for 14 rounds and 126 active S-boxes for 28 rounds.

Run-time Results

Table 4: Performance results of the BBI.

Algorithm	Key Size	Table	Table Round	Cycle (per byte)
Shadow256	256	LS-design	32	111100
Shadow256	256	SPNbox	18	133733
SPNbox-32	128	SPNbox	16	117848

Conclusion

- The use of lightweight components in white-box settings enables reasonably fast algorithm designs.
- According to the performance results, all white-box transformations are faster than (M, Z)-space hard algorithms WARX and SPNbox-32 without decreasing the white-box security level.

Thank you for your attention