Several Methods for Constructing White-box Solutions

Tingting Lin¹*, Hailun Yan¹, Xuejia Lai¹, Weidong Qiu²

1. School of Electrical and Information Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.
2. School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China.
*Corresponding author: lintingting00@163.com

Abstract. White-box cryptography is an important countermeasure for providing software security in untrusted environments. While almost all public white-box solutions have been broken, the methods for constructing such solutions are worth learning. In this paper, we introduce several classic methods for constructing white-box solutions. The methods are divided into two categories: those that make the white-box solutions functionally equivalent to existing ciphers, and those that do not. In each category, we explain each method using steps and figures. Our introduction covers almost all the general methods for constructing white-box solutions and aims at presenting a full overview of white-box designs.

1. Introduction

These in the last 10 years, several techniques that use side-channel information to implement cryptanalysis have been developed. Because side-channel information, such as power consumption, electromagnetic radiation and execution timing, is always related to the key and difficult to deploy protection in a platform, they are useful in a cryptanalysis. Even worse, execution details such as memory states, CPU calls, and dynamic data can also be inspected or intercepted by an adversary. Given such a context, cryptographic schemes are executed on an untrusted platform, and the adversaries are assumed to have full control over the internal details of the execution. We denote these attacks as white-box attacks. Traditional cryptographic schemes cannot provide protection for private information in white-box attack contexts; thus, white-box cryptography was proposed to address this problem.

Normally, the understanding is that white-box cryptography is aimed at protecting secret keys from being extracted from the cryptographic software, because early research on white-box cryptography was focused on constructing software implementations of block ciphers. As research continues, it is now a well-accepted idea that white-box implementation can be applied on more cryptographic primitives, and white-box cryptographic techniques can provide robustness to the implementations of cryptographic systems or construct cryptographic systems that can succeed in their functionality (such as encryption, decryption, and authentication) under white-box attack scenarios.

However, white-box cryptanalysis techniques come under attacks incessantly. For example, fault propagation correlation [1], guess and determine attacks [2], algebraic cryptanalysis [3], affine equivalence algorithm [4], differential computation analysis [5] etc., have been used to attack many white-box implementations successfully. To resist white-box attacks, various white-box cryptographic schemes have been proposed.

In this paper, we introduce several methods for constructing white-box cryptographic solutions by summarizing the currently known white-box schemes. We aim to present a full overview of the white-box designs, which can serve as a reference for future research.
2. Related Work
The first white-box implementations were presented by Chow et al. [6, 7], when lookup tables were first applied to white-box solutions. However, in 2002, Jacob et al. in [8] demonstrated that the white-box DES implementation was vulnerable to fault-injection attack. Although it was improved by Link et al. in [9], Wyseur et al. [10] and Goubin et al. [11] independently recovered the secret keys in their work. For the white-box AES implementation, because the S-box and MixColumn matrices of the AES are public, it was broken by Billet et al. [3] and Lepoint et al. [12]. In 2009, an attack focused on the white-box implementation of any substitution linear transformation (SLT) cipher was proposed in [13]. Based on Chow et al.’s design, some white-box solutions have been presented [14-17], however the attempts at solutions have proven to be insecure [18, 12, 19].

From another point of view, Biryukov et al. [20] encapsulated ASASA (affine-substitution-affine-substitution-affine) transformation in a table and constructed a white-box solution. However, this was also broken by Minaud et al. [21] and Dinur et al. [22]. In 2017, Lin et al. [23] improved the method and used ASASA transformation and unbalanced Feistel structure to construct a white-box solution.

Moreover, some dedicated white-box schemes, called space-hard ciphers, were proposed by Bogdanov et al. [24] in 2015. Additionally, to resist side-channel attack DCA (differential computation analysis), masking-based countermeasures were proposed in [25, 26].

3. The Methods to Construct White-box Solutions
As stated above, several white-box cryptographic solutions have been proposed in recent years. These solutions can be divided into two categories: those equivalent to existing ciphers, and those not equivalent to existing ciphers.

3.1. Equivalent to Existing Cipher
The white-box cryptographic solutions that are functionally equivalent to existing ciphers can be viewed as software implementation variations of the ciphers in a manner such that the implementations remain secure even when they are subject to white-box attacks. The inputs/outputs of the white-box implementation will coincide with the inputs/outputs of the original cipher. For an easy understanding, we call these white-box solutions as white-box implementation of the cipher, for example, white-box implementation of AES.

Networked lookup tables based on annihilating encoding The idea of using networked lookup tables to implement a cipher was first proposed by Chow et al. in the construction of white-box implementations of DES and AES [6, 7]. For a fixed-key, the white-box techniques convert a cipher into a series of lookup tables. The secret key of the cipher is hard-coded into some of the lookup tables, and all the lookup tables are protected by annihilating encodings. Figuer. 1 shows the procedure of the white-box technique.

![Figure 1. Constructing networked lookup tables](image)

The technique of networked lookup tables is normally applied on block ciphers, especially block ciphers with a substitution-permutation network (SPN). The general steps of the technique are...
described below:

Break the cipher into a number of steps.

Decompose each step into several lookup tables, make sure the secret key is hard-coded into the substitution steps (S-box).

Apply a randomly chosen mix bijection on the inputs/output of each lookup table, make sure the bijection applied on the outputs of a lookup table is the inverse of the bijection applied on the inputs of next adjacent lookup table. Because the adjacent bijections can be annihilated in the end, we call them annihilating encodings. The annihilating encodings will keep the implementation functionally equivalent with the cipher.

The cipher is implemented with a collection of the randomized look-up tables.

Most literature used the networked lookup tables to implement ciphers, such as white-box implementation of DES [6, 9], white-box implementation of AES[7, 14, 15], and white-box implementation of SMS4 [17, 27]; however, almost all of them have been proven to be insecure.

Masking-based networked lookup tables Outside of academia, there are practical attacks that have proven to be more efficient at breaking many white-box solutions. To improve the security of white-box implementations against such attacks, the most common countermeasure is to add masks.

Masking-based networked lookup tables Outside of academia, there are practical attacks that have proven to be more efficient at breaking many white-box solutions. To improve the security of white-box implementations against such attacks, the most common countermeasure is to add masks.

As shown in Figure 2, the basic steps of masking-based networked lookup tables are as follows:

Select random values as masks.

Apply masking technique before encoding the outputs of a lookup table.

Use the same linear transformations to encode the masks and generate new lookup tables. Such lookup tables will be used to annihilate the masks in the end. Differential computation analysis (DCA) proposed by Bos et al.[5] is one of the powerful attacks. The idea is to collect the software execution traces from the implementation and apply techniques from side-channel attacks. Because uniformly selected masks can be used to adjust bias and balance of the linear and non-linear encodings in a white-box implementation, masking-based obfuscation techniques are efficient in weakening DCA attacks.

To better utilize the masking-based techniques, we must make sure the masks are uniform, and using non-linear masking technique is also recommended. For more details, please refer to [25].

3.2. Non-equivalent to Any Existing Cipher

White-box cryptographic solutions that are not equivalent to any existing ciphers are, in reality, brand new cryptographic primitives. We call them new white-box cryptographic schemes. There are three typical ways to design such schemes: replace standard components with customized components in an
existing cipher, use a non-standard component with a non-standard structure, and use a well studied cipher as a component in a structure.

Replace standard components with customised components Based on Kerckho’s principle, a cipher should have everything public, except for the key. For almost all the white-box implementation of existing ciphers, they are easy to be broken mainly because of the public components, for example, the public S-box and linear transformation matrix in a block cipher. This motivated a natural idea that we can replace the standard components with some other unknown components. This countermeasure would probably create a challenge for white-box attackers when they attack the white-box implementations. In [27, 28], this method is used to construct white-box implementations of derivate AES and derivate SMS4, respectively. We use SMS4 as an example to demonstrate the basic idea of this method (Figure 3).

![Figure 3. Replace standard components with customized components](image)

In general, several steps are required in order to apply this method to design new white-box cryptographic schemes:

- Develop substitution (S-box) and/or linear transformation according to the design criteria of a black-box model. For example, we need to consider the algebraic complexity when constructing the S-box and the branch number when constructing the linear transformation matrix.

- Evaluate the black-box security for the new ciphers.

- Construct white-box implementation of the new ciphers.

Non-standard component with non-standard structure White-box solutions constructed by this method are more in line with the definition of brand new cryptographic primitives. With this method, the framework of the solutions can be any structure (they are not necessarily SPN structure or Feistel structure), and the components of the solutions are non-standard transformations.

In [20], ASASA transformation is used as the component of their white-box scheme, and a parallel structure is used as the framework. In [23], ASASA is also used as the component, however an unbalanced Feistel network is used as the framework. The basic idea of this method is shown in Fig. 4 (we use a construction in [20] as the example).

![Figure 4. Non-standard component and non-standard structure](image)
Use well-studied ciphers as components in a structure. Another method to construct white-box solutions is to use some well-studied ciphers as components, such as AES. The advantage of this kind of white-box solutions is that the well-studied ciphers already exhibit excellent performance in black-box security, and they can be used as a random permutation to make the inputs/outputs of the white-box solutions appear more uniform. Additionally, the entire cipher is always encapsulated in a lookup table, thus the difficulty in extracting the key from the white-box solution will be reduced to the difficulty in recovering the secret key from the underlying cipher in the black-box model. The basic idea of this method is shown in Fig. 5.

Choose a framework for the white-box solution (Feistel network is used in Fig. 5).

Choose a well-studied cipher as the components in the framework. The inputs/outputs of the cipher can be adjusted, for example, part of the inputs can be random values and part of the outputs can be discarded.

Construct lookup tables that encapsulate the cipher.

The Space-Hard ciphers proposed by Bogdanov et al. [24] use this method. Apart from black-box security, code-lifting-proof is one of the advantages of such white-box solutions, however the size of the lookup tables will be too large to be suitable for many devices.

![Diagram](image)

**Figure 5.** Using well-studied ciphers

4. **Conclusion**
In this paper, we introduced several methods for constructing white-box solutions. We divided the methods into two categories. In one category, a white-box solution is functionally equivalent to an existing cipher, which is called white-box implementation of a cipher. A series of networked lookup tables is the most common manifestation in this category. In the other category, a white-box solution is not an implementation version of any existing cipher but a brand new cryptographic primitive. There are three methods can be used to construct such solutions, however all of them require black-box security evaluations. As future work, we will explore more novel methods and study the best method for constructing white-box solutions with excellent performance.

5. **Acknowledgments**
This work was supported by National Natural Science Foundation of China (No. 61702331, 61472251, U1536101, 71774111), China Postdoctoral Science Foundation (No. 2017M621471). The authors are very grateful to the anonymous referees for their valuable comments and suggestions, helping them to improve the quality of this paper.

6. **References**
[1] Matthias J, Dan B, Edward W. Felten. Attacking an Obfuscated Cipher by Injecting Faults. In Proceedings of the ACM Workshop on Security and Privacy in Digital Rights Management (DRM 2002), volume 2696 of Lecture Notes in Computer Science, pages 1631. Springer, 2002.
[2] Hamilton E. Link and William D. Neumann. Clarifying Obfuscation: Improving the Security of White-Box DES. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2005), volume 1, pages 679-684, Washington, DC, USA, 2005. IEEE Computer Society.

[3] Olivier B, Henri G, and Charaf Ech-Chatbi. Cryptanalysis of a White Box AES Implementation. In Proceedings of the 11th International Workshop on Selected Areas in Cryptography (SAC 2004), volume 3357 of Lecture Notes in Computer Science, pages 227-240. Springer-Verlag, 2004.

[4] Biryukov A, De Canniere C, Braeken A, et al. A toolbox for cryptanalysis: Linear and affine equivalence algorithms. International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2003: 33-50.

[5] Bos J W, Hubain C, Michiels W, et al. Differential computation analysis: Hiding your white-box designs is not enough. International Conference on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2016: 215-236.

[6] Chow S, Eisen P, Johnson H, et al. A white-box DES implementation for DRM applications. In Proc. Digital Rights Management, October 2003: pp.1-15.

[7] Chow S, Eisen P, Johnson H, et al. Whitebox cryptography and an AES implementation. In Proc. Selected Areas in Cryptography, August 2003: pp.250-270.

[8] Jacob M, Boneh D, Felten E. Attacking an obfuscated cipher by injecting faults. In Proc. Digital Rights Management, October 2003: pp.16-31.

[9] Link H E, Neumann W D. Clarifying obfuscation: improving the security of white-box DES. Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on. IEEE, 2005, 1: 679-684.

[10] Wyseur B, Michiels W, Gorissen P, et al. Cryptanalysis of white-box DES implementations with arbitrary external encodings. In Proc. Selected Areas in Cryptography, August 2007: pp.264-277.

[11] Goubin L, Masereel J M, Quisquater M. Cryptanalysis of white box DES implementations. In Proc. Selected Areas in Cryptography, August 2007: pp.278-295.

[12] Lepoint T, Rivain M, De Mulder Y, et al. Two attacks on a white-box AES implementation. International Conference on Selected Areas in Cryptography. Springer, Berlin, Heidelberg, 2013: 265-285.

[13] Michiels W, Gorissen P, Hollmann H D L. Cryptanalysis of a generic class of white-box implementations. International Workshop on Selected Areas in Cryptography. Springer, Berlin, Heidelberg, 2008: 414-428.

[14] Xiao Y, Lai X. A secure implementation of white-box AES. Computer Science and its Applications, 2009. CSA'09. 2nd International Conference on. IEEE, 2009: 1-6.

[15] Karroumi M. Protecting white-box AES with dual ciphers. International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2010: 278-291.

[16] Shi Y, Liu Q, Zhao Q. A secure implementation of a symmetric encryption algorithm in white-box attack contexts. Journal of Applied Mathematics, 2013, 2013.

[17] Xiao Y, Lai X. White-Box cryptography and implementations of SMS4. In Proc. the 2009 CACR Annual Meeting, November 2009, pp.34.

[18] De Mulder Y, Roelse P, Preneel B. Cryptanalysis of the XiaoLai white-box AES implementation. International Conference on Selected Areas in Cryptography. Springer, Berlin, Heidelberg, 2012: 34-49.

[19] Lin T, Lai X. Key-recovery attack to white-box SMS4 implementation. J. Softw, 2013, 24: 2238-2249.

[20] Biryukov A, Bouillaguet C, Khovratovich D. Cryptographic schemes based on the ASASA structure: Black-box, white-box, and public-key. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2014: 63-84.

[21] Minaud B, Derbez P, Fouque P A, et al. Key-recovery attacks on ASASA. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2014: 3-27.
[22] Dinur I, Dunkelman O, Kranz T, et al. Decomposing the ASASA Block Cipher Construction. IACR Cryptology ePrint Archive, 2015, 2015: 507.

[23] Lin T T, Lai X J, Xue W J, et al. A New Feistel-Type White-Box Encryption Scheme. Journal of Computer Science and Technology, 2017, 32(2): 386-395.

[24] Bogdanov A, Isobe T. White-box cryptography revisited: Space-hard ciphers. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015: 1058-1069.

[25] Biryukov A, Udovenko A. Attacks and Countermeasures for White-box Designs. IACR Cryptology ePrint Archive, 2018, 2018: 049.

[26] Lee S. A Masked White-box Cryptographic Implementation for Protecting against Differential Computation Analysis. IACR Cryptology ePrint Archive, 2017, 2017:267.

[27] Shi Y, Wei W, He Z. A lightweight white-box symmetric encryption algorithm against node capture for WSNs. Sensors, 2015, 15(5): 11928-11952.

[28] Xu T, Liu F, Wu C. A white-box AES-like implementation based on key-dependent substitution-linear transformations. Multimedia Tools and Applications, 2017: 1-21.