Hybrid WBC: Secure and efficient encryption schemes using the White-Box Cryptography

Jihoon Cho, Kyu Young Choi, and Dukjae Moon

Network & Security Lab, Samsung SDS, Inc., Seoul, Korea
{jihoon1.cho, ky12.choi, dukjae.moon}@samsung.com

Abstract. We analyse and define practical requirements in white-box attack environment, and then propose secure and effective cryptographic constructions combining white-box cryptography (WBC) primitive and standard block cipher, providing security and reasonable performance. The proposed design also delivers great effectiveness in the commercial development of cryptographic systems, transforming the existing cryptographic libraries secure in the black-box model to those secure in the white-box model. Furthermore, the suggested architecture potentially gives a novel direction of the design of WBC primitives.

1 Introduction

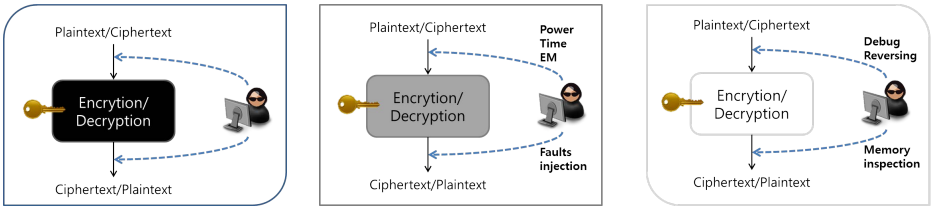
With widespread use of smartphone devices and public cloud services, cryptographic primitives become increasingly implemented as software. Certain security-critical services are provided with support of hardware security features such as secure element or TrustZone in mobile devices or hardware security modules in cloud. Most services, however, are implemented as software operating within Rich OS mainly because of cost, development efficiency and the complicated ecosystems. The cryptographic implementations are then vulnerable to a wide variety of attacks, which include remote software attacks using mobile malware such as root-kit as well as physical attacks.

Furthermore, we have been long discussing security issues regarding resource-constraint devices, but, more seriously, such devices become increasingly connected allowing far larger scale of attacks ever in the Internet of Things (IoT) environment [1, 2]. We thus have sufficient reasons to reconsider the design cryptographic primitives to mitigate attacks against cryptographic implementations working in unsecure environment.

Until the early 1990s the endpoints of communication channels were assumed to be secure, and most cryptographic primitives had been designed assuming such an Black-box model (see Figure 1 (a)) [3]. Kocher, however, introduced a more realistic attack model, called the Gray-box model (see Figure 1 (b)) [4]. In the suggested model, an adversary can access side channels, including execution time, power consumption and electromagnetic radiation [5–7], which is called the side-channel attack (SCA).

Yung and Chang suggested to reconsider the existing security model, defining Midgame attack which includes physical invasive attacks such as cold boot

attack and memory dump [8, 9]. In 2002, Chow et. al formally introduced the white-box model (Figure 1 (c)) [10, 11], assuming the capabilities of adversary as follows. First, an adversary shares a host device operating the cryptographic implementation under attack, having complete access to the implementation of algorithms. Second, dynamic execution with instantiated cryptographic keys can be observed, and internal details of cryptographic algorithms are both completely visible and alterable at will.



(a) Black-box model (b) Gray-box model (c) White-box model

Fig. 1. Attack Models

The cryptographic construction designed to remain secure under white-box attack, called white-box cryptography (WBC), delivers several advantages. As software implementation, it could be easily deployed and updated across various platforms, without requirement of hardware cost.

Our contribution: Since the existing WBC algorithms are not able to provide reasonable performance, they were thus mainly applied to protect relatively small size of data. For this reason, there have been approaches of improving performance [12, 13]. More specifically, they employ certain mode of operations with both WBC version of AES and standard AES as input.

We first define reasonable requirements for WBC algorithms in terms of security and applicability considering cryptographic environment such as IoT. We show that the previous schemes are not secure, and then propose secure and effective cryptographic constructions combining a WBC primitive and standard block cipher as underlying primitives, assuming those underlying WBC and block cipher is secure. The proposed design delivers great effectiveness in the commercial development of cryptographic systems, transforming the existing cryptographic libraries secure in the black-box model to those secure in the white-box model. Furthermore, the suggested architecture potentially gives a novel direction of the design of WBC primitives.

2 White Box Cryptography

Since 2002, there have been several constructions of white-box cryptography (WBC) using table-based or polynomial-based implementations. They translated the description of the given block cipher into a series of lookup tables (Chow et al. [10, 11], Xiao-Lai [14] and Karroumi [15]) or multivariate polynomials (Bringer et al. [16] and Biryukov et al. [17]).

More specifically, Chow et al. [10, 11] proposed a method of constructing a WBC version of AES using large tables and encoding schemes. The basic idea is to change several internal steps of a given algorithm into a network of large look-up tables, where the key is embedded in the parts of tables. To hide the internal steps of the algorithm they used encoding schemes for the input and output of each table. Bringer et al. [16] proposed the WBC construction of AES using the isomorphism of polynomials. They change round functions of AES into a set of a multi-variate equations with perturbations. The perturbations and random systems were used for the generation of a random equation set.

Table 1 lists the implementation size of the previously suggested WBC construction, assuming 128-bit security level. Note that the standard AES could be implemented within 4KB with all tables [18]. There have been attempts of reducing the size of WBC implementations, for example, re-using S-Box tables or internal encodings [19]. Unfortunately, however, all the previous WBC constructions have been cryptanalysed. [20–22] as in Table 1. The algebraic attacks were used to successfully extract the embedded 128-bit master key with a small work factor.

Table 1. Analysis of size and attack results

Implementation	Size	Attack Results
Chow et al. [11] Karroumi [15]	753KB	2^{22} [22]
Xiao-Lai [14]	20MB	$\geq 2^{32}$ [21, 23]
Bringer et al. [16]	568MB	2^{16} [24]
Biryukov et al. [17]	8MB	$\geq 2^{28}$ [25]

3 Hybrid White-Box Cryptography

In this section we propose robust and efficient cryptographic constructions secure in the white-box model, namely hybrid WBC (h-WBC), and discuss the implications from both academia and industry.

3.1 Design goals

Cryptographic primitives are designed mainly focusing on security, performance and cost. That is, they should be resistant against defined attacks, provide throughput reasonable in applied environment, and also minimize cost involved in development.

We first discuss on practical approaches regarding the adversarial capabilities in the white-box model. In the cryptographic perspective, it would not be reasonable assumption that an adversary has an access to the cryptographic operation environment at any time. Any cryptographic implementations would be of no use against such an attack, where an adversary may observe memory buffer for decryption rather than trying to recover cryptographic keys, for example. We may instead assume that an adversary would attempt a relatively expensive white-box attack once and exploit the obtained information to launch less expensive black-box attacks.

The above assumption, in particular, is well suited in IoT environment. IoT devices are usually manufactured in production line simply assembling flash memories with the same binary programmed including cryptographic keys, i.e. the same cryptographic keys are shared across multiple devices. This is because it would be quite expensive to embed separate keys into each devices either in production lines or by consumers; additional key-embedding process and related key management, as well as adding UX layers to IoT devices, generally require considerable cost. In such an IoT environment, for instance, an adversary may implement the white-box attack for a single device, and try to compromise whole system using the obtained key or any critical information using the capabilities from the conventional black-box model. To sum up, cryptographic constructions need to be designed to minimise the damage from one-time compromise. That is, any compromised information should not give any additional advantage to an adversary in the subsequent black-box attack model.

Previously suggested WBC constructions are 12–55 times slower than the standard AES [11, 26]. WBC primitives thus have been used to protect relatively small size of data such as data encryption keys. For this reason, in the architecture of DRM service, for example, a content encryption key (CEK) are typically encrypted using a WBC primitive. We note that such security-critical CEK is eventually exposed in memory[1], and such a cracked key could be used for illegal distribution of contents. It would be desirable to construct cryptographic architecture to deliver reasonable performance whenever possible.

Lastly, but most importantly, the architecture should be designed to minimise the modification of the existing development or manufacturing process related to cryptographic implementations. Interestingly, it could be most important factor for commercial adoption in reality. Industry believes that security is essential, but often regard its implementation as an additional cost.

We now define the design goals of the proposed architecture as follows.

[1] Of course, various obfuscation tools are used for protection.

1. Security. It should minimise the damage from one-time compromise. That is, any compromised information in the white-box attack should not give any additional advantage to an adversary in the subsequent attacks in black-box model.
2. Performance. It should be securely protect an arbitrary size of data, providing reasonable performance.
3. Cost. It should be designed to minimise the modification of the existing development or manufacturing process related to cryptographic implementations.

3.2 Related work

In order to achieve the goals defined in the previous section, there have been approaches, so-called the composition modes [12, 13], using WBC as a building block for secure constructions.

In 2010, Park et al. [12] proposed a novel method of using WBC to enhance performance. They make use of PCBC mode with dual keys as input to WBC-AES[2] and AES, where a WBC primitive encrypts only one block of plaintext. It is trivial that their scheme runs as fast as AES for large size of data. With the AES key obtained by the white-box attack, however, an adversary is easily able to decrypt any other ciphertext blocks encrypted using the same keys, preceding the data block encrypted using a WBC primitive.

Yoo et al. [13] proposed another types of composition mode, choosing CBC and OFB modes. In the specific construction using CBC mode, a random value is generated and then XORed with the first block of plaintext before AES encryption. The random value is also encrypted using a WBC primitive, and concatenated with ciphertext blocks. With the AES key compromised by the white-box attack, an adversary is also able to decrypt any other ciphertext blocks encrypted using the same keys in the black-box attack model, only except the first data block.

Park et al. and Yoo et al. both claims that CTR mode is not applicable in the their composition mode, but in the subsequent section we shows that a WBC primitive and standard encryption algorithm could be securely and efficient combined using CTR mode.

3.3 Description of secure h-WBC implementations

Considering the design goals, we describe outline of proposed cryptographic constructions of h-WBC as described in Figure 2. We choose a certain mode of operations such as PCBC, CTR and OFB for secure and efficient construction using a WBC primitive and any standard encryption algorithm, e.g. AES. The proposed basic construction uses two separate keys for a WBC pritmive and standard encryption scheme, and a WBC algorithm is only used for encryption

[2] A WBC primitive is designed based on AES, and it gives the same output as AES given the same input.

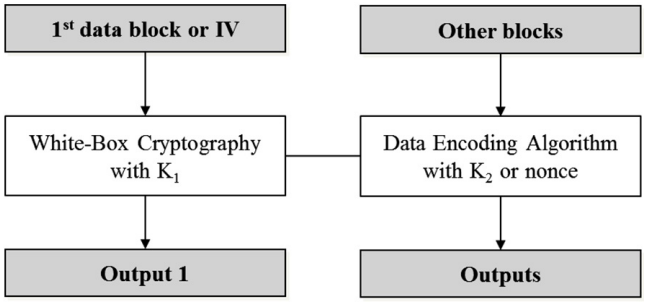


Fig. 2. The basic structure of proposed h-WBC

of an initial vector (IV) or counter and AES for encryption of pliantexts. We also give an additional example construction using a single key.

We now give example constructions of h-WBC using PCBC, CTR, and OFB modes, assuming that the underlying WBC primitive is secure under the white-box attack, i.e. an adversary cannot recover a key, k_1.[3] Figure 3 describes a h-WBC construction using PCBC mode, PCBC-WBC say. The proposed architecture is exactly the same as PCBC mode using standard block ciphers, but an IV is encrypted using a WBC primitive.

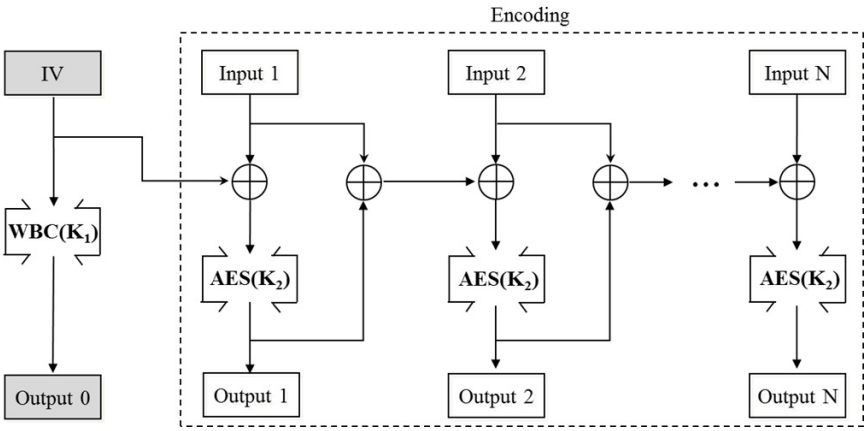


Fig. 3. Construction of h-WBC using PCBC mode: PCBC-WBC

[3] It may be possible to lift the whole WBC binary and then run it in simulator, in which case a WBC primitive itself is a key. We assume that the system has an counter-measure against such code-lifting attacks, e.g. additional coding scheme, node-locking technologies, etc.

We next give the second construction of h-WBC using CTR mode, namely CTR-WBC, as in Figure 4. Similarly to PCBC-WBC, a CTR value is now encrypted using a WBC primitive. Figure 5 describes the third construction of h-WBC using OFB mode, namely OFB-WBC. Yoo et al. [13] mentioned an idea of using OFB mode without specific description. Similarly to the above the constructions, OFB, an IV value is encrypted using a WBC primitive.

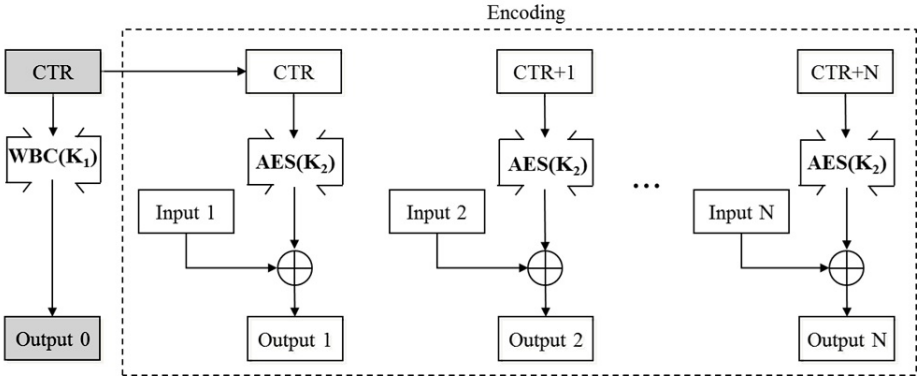


Fig. 4. Construction of h-WBC using CTR mode: CTR-WBC

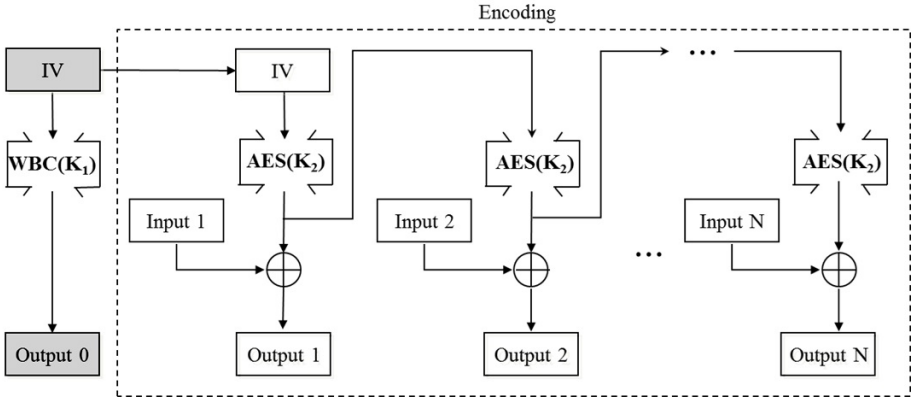


Fig. 5. Construction of h-WBC using OFB mode: OFB-WBC

We have given examples of h-WBC with two separate keys as input, but it would be problematic to manage such two keys in certain operating environment. Finally, we give an additional example construction of h-WBC using a single key as input to WBC only as in Figure 6.

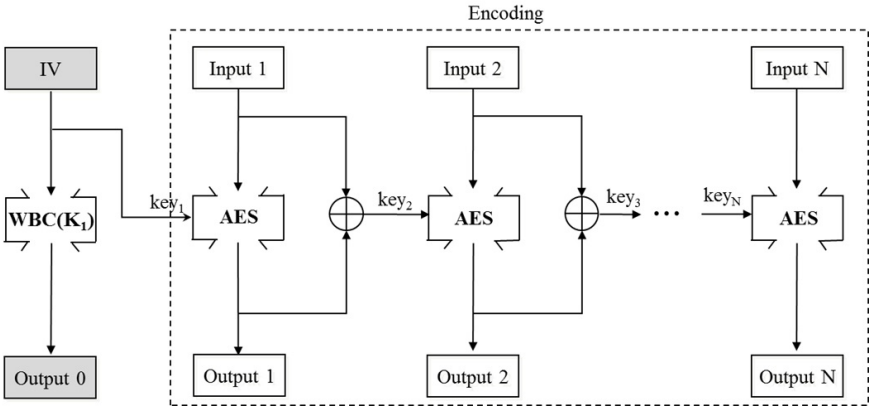


Fig. 6. Construction of h-WBC using PCBC mode: PCBC-WBC with a single key

3.4 Discussion about security and performance

We fist show that the specific constructions of h-WBC satisfy the security requirement. That is, even though an adversary obtains any information except a WBC key in white-box attack, other ciphertexts encrypted using the same keys remain secure in the conventional black-box model.

In the PCBC-WBC with two keys, an adversary cannot decrypt any block of ciphetexts, i.e. Output 1 and after in Figure 3, since each decryption requires the preceding plaintext and AES key. The first ciphertext block also remains secure since an adversary cannot decrypt the encrypted version of IV without k_1. The other constructions of h-WBC could be shown to be secure with similar analysis. We note that CTR mode also satisfies security requirement, unlike the claims [12, 13] that CTR mode is not applicable.

Table 2. Comparison of encryption performance (unit: second)

Scheme	**1MB**	**5MB**	**10MB**	**50MB**	**100MB**	**500MB**
WBC [27] (open source)	5.07	28.52	57.79	288.32	607.59	2860.15
h-WBC (WBC[27]+AES)	0.0033	0.0130	0.0254	0.3118	0.6232	3.6787
AES	0.0029	0.0121	0.0233	0.3106	0.6184	3.6689

(CPU: Intel Core i7-4770 8Core 3.4GHz, RAM: 4.0GB, OS: 64-bit Ubuntu 14.04 LTS)

Table 2 shows that the encryption performance between the open source WBC [27], our hybrid WBC, and AES based on CTR mode. It is trivial that the overall performance of h-WBC constructions would be almost the same as

standard modes of operation for relatively large size of data. In particular, CTR-WBC may greatly improve performance compared to other h-WBC contructions due to the possibilities of parallel computations.

3.5 Implications

The basic example constructions of h-WBC, i.e. PCBC/CTR/OFB-WBC, give significant implications from both industry and academia. Development of commercial solutions including cryptographic implementations often considers two critical aspects, i.e. compliance and cost. It is increasingly required for cryptographic libraries to comply with standards, e.g. FIPS 140-2, and also critical to minimise the implementation scope for cost savings related to development and test.

The proposed construction merely requires to implement an adaptor including a WBC algorithm and to integrate it with the existing cryptographic libraries. This is possible because, prior to giving an IV or counter as a parameter, the adaptor layer could simple encrypt them using a WBC primitive.

Another implication is that, given the h-WBC architecture, the design of WBC does not have to be based on the existing block ciphers, e.g. AES or 3-DES, since a WBC primitive only encrypts an IV or counter without affecting conventional mode of operations. This is not the case of Yoo et al.'s scheme [13], where a WBC primitives should have to use WBC-AES. In the h-WBC, any WBC primitive could enhance data security as long as it remains secure under white-box attacks. Most of all, the proposed design removes serious concerns on performance issues of WBC primitives. The overall performance of h-WBC converges to the one of underlying standard block cipher as the size of data blocks increases.

4 Conclusion

We have given practical requirements in terms of security, performance and cost, and then proposed the secure and effective construction combining WBC algorithm and block cipher as underlying primitives. The proposed design delivers great effectiveness in the commercial development of cryptographic systems, transforming the existing cryptographic libraries secure in the black-box model to the ones secure in the white-box model. Finally, we gave a novel direction of WBC primitives designs.

References

1. J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand and D. Boyle, From machine-to-machine to the internet of things: Introduction to a new age of intelligence, Elsevier, 2014.
2. B. Jun, Make way for the internet of things, RSA conference, 2014.

3. A. J. Meneses, P. C. van Oorschot and S. A. Vanstone, Hnadbook of applied cryptography, CRC Press, 1997.
4. P. C. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, CTYPTO 1996, LNCS 1109, pp. 104–113, Springer, 1996.
5. D. Brumley and D. Boheh, Remote timing attacks are practical, SSYM 2003, pp. 1–14, USENIX Association Berkeley, 2003.
6. P. C. Kocher, J. Jaffe and B.Jun, Differential power analysis, CRYPTO 1999, LNCS 1666, pp. 388–397, Springer, 1999.
7. J. Quisquater and D. Samyde, Electromagnetic analysis (EMA: Measures and counter-measures for smart cards, E-smart 2001, LNCS 2140, pp. 200–210, Springer, 2001.
8. D. H. Chang and M. Yung, Midgame attacks (and their consequences), CRYPTO rump session, 2012.
9. M. Yung, Lets get real about the real world: cryptography facing extreme system breaks, Real world cryptography workshop, 2014.
10. S. Chow, P. Elsen, H. Johnson and P. C. A. van Oorschot, A white-box DES implementation for DRM applications, DRM 2002, LNCS 2696, pp. 1–15, Springer, 2003.
11. S. Chow, P. Elsen, H. Johnson and P. C. A. van Oorschot, White-box cryptography and an AES implementation, SAC 2002, LNCS 2595, pp. 250–270, Springer, 2003.
12. J. Park, O. Yi and J. Choi, Methods for practical whitebox cryptography, IETC 2010, pp. 474–479, IEEE, 2010.
13. J. Yoo, H. Jeong and D. Won, A method for secure and an efficient block cipher using white-box cryptography, ICUIMC 2012, No. 89, ACM, 2012.
14. Y. Xiao and X. Lai, A secure implementation of white-box AES, CSA 2009, pp. 1–6, IEEE, 2009.
15. M. Karroumi, Protecting white-box AES with dual ciphers, ICISC 2010, LNCS 6829, pp. 278-291, Springer, 2010.
16. J. Bringer, H. Chabanne, and E. Dottax, White Box Cryptography: Another Attempt, ePrint 2006-468, 2006.
17. A. Biryukov, C. Bouillaguet and D. Khovratovich, Cryptographic Schemes Based on the ASASA Structure: Black-Box, White-Box, and Public-Key, ASIACRYPT 2014, LNCS 8873, pp. 63-84, Springer, 2014.
18. D. A. Osvik, J. W. Boa, D. Stefan and D. Canright, Fast Software AES Encryption, FSE 2010, LNCS 6147, pp. 75–93, Springer, 2010.
19. M. Ciet, A. J. Farrugia and F. T. Paun, Systems and methods for implementing block cipher algorithms on attacker controlled systems US 20100054461, Apple, 2008.
20. O. Billet, H. Gilbert and C. Ech-Chatbi, Cryptanalysis of a white box AES implementation, SAC 2004, LNCS 3357, pp. 227–240, Springer, 2004.
21. W. Michiels, P. Gorissen and H. D. L. Hollmann, Cryptanalysis of a generic class of white-box implementations, SAC 2008, LNCS 5381, pp. 414–428, Springer, 2008.
22. T. Lepoint, M. Rivain, Y. De Mulder, P. Roelse and B. Preneel, Two attacks on a white-box AES implementation, SAC 2013, LNCS 8282, pp. 265–285, Springer, 2014.
23. Y. De Mulder, P. Roelse and B. Preneel, Cryptanalysis of the Xiao-Lai white-box AES implementation, SAC 2012, LNCS 7707, pp. 34–49, Springer, 2013.
24. Y. De Mulder, B. Wyseur and B. Preneel, Cryptanalysis of a Perturbated White-Box AES implementation, Indocrypt 2010, LNCS 6498, pp. 292–310, Springer, 2010.

25. I. Dinur, O. Dunkelman, T. Kranz and G. Leander, Decomposing the ASASA Block Cipher Construction, ePrint 2015-507, 2015.
26. B. Wyseur, White-Box Cryptography, PhD thesis, Katholieke Universiteit Leuven, 2009.
27. Open source for WBC: Whitebox AES implementation in C++. Available at https://github.com/ph4r05/Whitebox-crypto-AES