SCIENCE CHINA
Information Sciences

• LETTER •

March 2020, Vol. 63 139106:1–139106:3
https://doi.org/10.1007/s11432-019-9883-9

Cryptanalysis of the obfuscated round boundary technique for whitebox cryptography

Yongjin YEOM[1*], Dong-Chan KIM[1], Chung Hun BAEK[2] & Junbum SHIN[2]

[1] *Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul* 02707*, Republic of Korea;*
[2] *Samsung Research, Samsung Electronics, Seoul* 06765*, Republic of Korea*

Received 3 March 2019/Accepted 18 April 2019/Published online 10 February 2020

Citation Yeom Y, Kim D-C, Baek C H, et al. Cryptanalysis of the obfuscated round boundary technique for whitebox cryptography. Sci China Inf Sci, 2020, 63(3): 139106, https://doi.org/10.1007/s11432-019-9883-9

Dear editor,
The security of cryptographic techniques heavily relies on the confidentiality of the key, and everything except keys can be made open to public without affecting security. When working with cryptographic services, we must protect the keys from a variety of attackers, including eavesdroppers and malware. A whitebox adversary is an attacker with the the most powerful capabilities to monitor and modify the software control flow of the system. Whitebox cryptography is a technology which implements cryptographic algorithms without disclosing key information, even in the presence of whitebox adversaries.

Whitebox implementations of block ciphers have been studied since 2002. The most popular design rationale for whitebox cryptography is to represent an algorithm as a sequence of table lookups. The key is resolved into the tables together with random nonlinear encodings so that even whitebox attackers cannot extract the key from the tables or executable codes. However, it is difficult to find a successful whitebox solution for standard block ciphers. The first whitebox AES introduced by Chow et al. [1] was broken by Billet et al. [2] in 2004, who analyzed the tables for whitebox AES and recovered the encryption key by observing the input-output of various table compositions. In fact, most whitebox AESes proposed thus far turned out to be insecure.

In 2018, in order to withstand the structural cryptanalysis of tables, Xu et al. [3] proposed a new design rationale called the “obfuscated round boundary technique”, which hinders adversaries from identifying each round from the tables. Here, we analyze the security of the obfuscated round boundary technique and construct a key recovery attack on a whitebox implementation of AES called Rbo-Wbaes$(10, 500)$.

Whitebox AES with obfuscated round boundaries. A substitution linear transformation (SLT) cipher is defined as a 128-bit block cipher, the round function of which has two layers: the substitution layer S and linear transformation M. In the substitution layer S, each byte x_i is mapped by a nonlinear permutation S_i with round key rk_i addition as $x_i \mapsto S_i(x_i \oplus \mathrm{rk}_i)$. The linear transformation is a linear map M over $\mathrm{GF}(2)^{128}$ represented by matrix multiplication. The SLT cipher with N rounds is expressed as

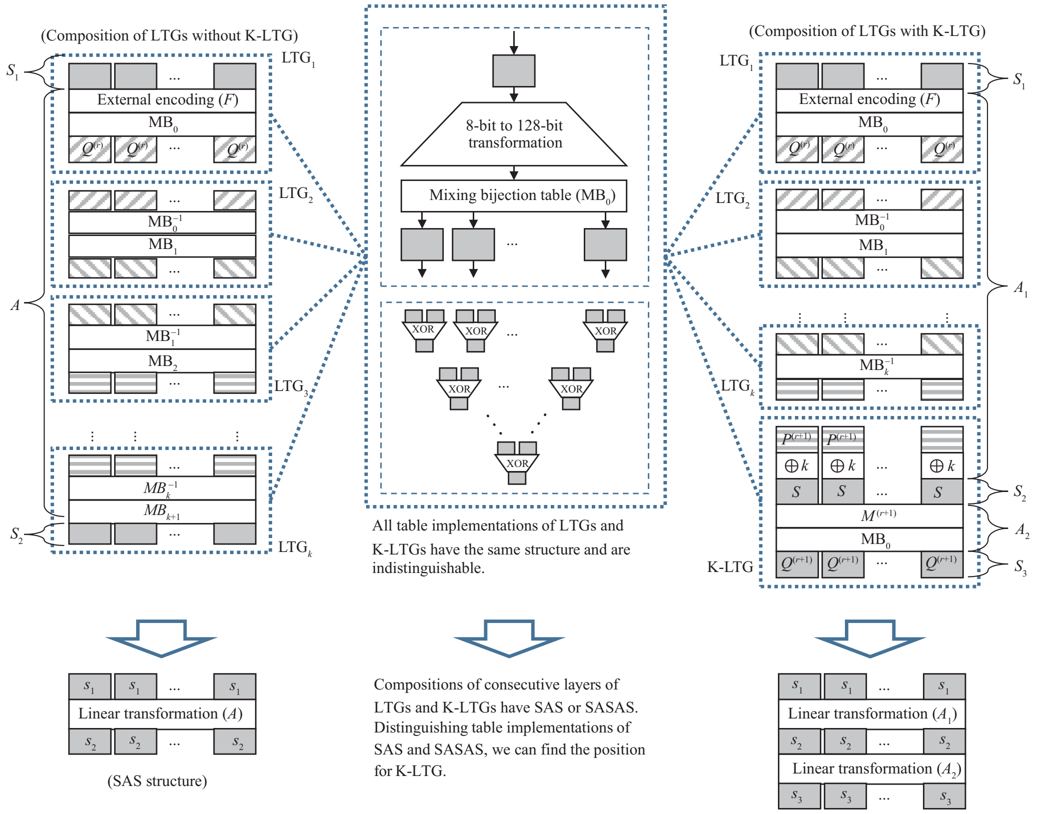
$$\underbrace{\left(M^{(N)} \circ S^{(N)}\right)}_{\text{round } N} \circ \cdots \circ \underbrace{\left(M^{(1)} \circ S^{(1)}\right)}_{\text{round } 1}.$$

In order to hide the round boundaries, Xu et al. [3] proposed the insertion of a random number of dummy rounds such as

$$\left(\mathrm{MB}_k^{-1}\right) \circ \underbrace{\left(\mathrm{MB}_k \circ \mathrm{MB}_{k-1}^{-1}\right)}_{\text{dummy round}} \circ \cdots \circ \left(\mathrm{MB}_1 \circ M^{(r)} \circ S^{(r)}\right),$$

where MB_i is an invertible linear transformation.

[*] Corresponding author (email: salt@kookmin.ac.kr)



All table implementations of LTGs and K-LTGs have the same structure and are indistinguishable.

Compositions of consecutive layers of LTGs and K-LTGs have SAS or SASAS. Distinguishing table implementations of SAS and SASAS, we can find the position for K-LTG.

Figure 1 (Color online) Table implementations of LTG and KLTG, all of which have the same structure. The position for KLTG can be found by the distinguisher for SAS and SASAS structures.

The whitebox implementation is composed of a network of lookup tables constructed by adding byte-wise random encoding and decoding at the beginning and end of each round, respectively. As depicted in the middle of Figure 1, each lookup table group (LTG) uses 8-bit to 128-bit tables and 16-bit to 8-bit tables, which may or may not represent a dummy round. Note that external encodings are involved in the first and the last LTGs without any modifications to the table structures. Particularly, an LTG that depends on the round key is called a K-LTG. Since each round begins with a K-LTG, adversaries need to determine the exact locations of K-LTGs in the sequence of LTGs.

Xu et al.'s whitebox implementation of AES, called Rbo-Wbaes, consists of 500 LTGs including 10 K-LTGs. According to their assertion, without detecting round boundaries, one cannot find the corresponding round keys or eventually recover the encryption key.

Structural analysis of LTGs. A multiset is an extended concept of a set that allows the multiplicity of its elements. We denote an intermediate state of the encryption process by a state vector $(x_1, x_2, \ldots, x_{16})$, where $x_i \in \mathrm{GF}(2^8)$ $(1 \leqslant i \leqslant 16)$. We only consider multisets of 256 state vectors. Each byte position in a multiset M is said to have properties P, C, E, D, and B, which are defined as follows:

- P (permutation): all values appear once.
- C (constant): it contains a single value.
- E (even): each value occurs an even number of times (allowing no occurrences).
- D (dual): it is either P or E.
- B (balanced): the exclusive-or operation on all the values returns 0.

For instance, a multiset M defined as $M = \{(x_1, \ldots, x_{16}) \in \mathrm{GF}(2^8)^{16} \,|\, x_2 = \cdots = x_{16} = 0\}$ has the properties $(P, C, \ldots, C)$.

We define SAS and SASAS structures as described in Biryukov et al. [4], where the S layer is defined as a byte-wise bijection layer like $S = S_1 \| \cdots \| S_n$ and the A layer represents a linear or affine transformation. The round function of an SLT cipher has an SA structure, and the whitebox implementation of LTGs or K-LTGs has the SAS structure. We use well-known observations in [4] on multiset properties to derive the following theorem, which plays a key role in our analysis.

Theorem 1 (Distinguishing SAS and SASAS)**.** Let a multiset M of 256 state vectors have the properties $(P, C, \ldots, C)$. If we use M as an input to SAS and SASAS structures, respectively, then

(a) The output of the SAS structure has the properties $(D, D, \ldots, D)$ with probability 1.

(b) The output of the SASAS structure cannot have any of the properties P, C, D, and B with a meaningful probability.

With the help of Theorem 1, we can easily distinguish SAS and SASAS structures by using the 256 chosen state vectors.

Cryptanalysis of Xu's Rbo-Wbaes. We observe that each LTG has an SAS structure, and we cannot distinguish LTGs and K-LTGs in a whitebox implementation, since they have the same structure. In fact, a lookup table group in Rbo-Wbaes can be interpreted as both an LTG as well as a K-LTG depending on the choice of random encodings. However, we can determine the locations for K-LTGs through Algorithm 1.

Algorithm 1 Round boundary detection algorithm

```
Input: Whitebox implementation of Rbo-Wbaes(10, 500),
    Rbo-Wbaes(10, 500) = $\mathrm{LTG}_{500} \circ \cdots \mathrm{LTG}_2 \circ \mathrm{LTG}_1$, 10
    K-LTGs are contained in $\{\mathrm{LTG}_1, \ldots, \mathrm{LTG}_{500}\}$.
Output: Starting positions for K-LTGs: $\{p_1, \ldots, p_{10}\}$.
Prepare a multiset of the form $(P, C, \ldots, C)$ with $2^8$
    inputs to Rbo-Wbaes(10, 500);
$k \leftarrow 0$, $r \leftarrow 1$;
$f \leftarrow \mathrm{LTG}_1$;
while $r < 500$ do
    if (output of $f$) $\neq (D, \ldots, D)$ then
        $p_k \leftarrow r$;
        $k \leftarrow k + 1$;
        $f \leftarrow \mathrm{LTG}_{r+1}$;
    else
        $f \leftarrow \mathrm{LTG}_{r+1} \circ f$;
    end if
    $r \leftarrow r + 1$;
end while
```

Let p_1 be the position of the first K-LTG, which contains the first round key. That is, LTG_{p_1} is the first K-LTG. Then, the compositions of

$$\mathrm{LTG}_i \circ \cdots \circ \mathrm{LTG}_1$$

have the SAS structure until $i < p_1$ because the byte-wise random encodings and MB_j, MB_j^{-1} of adjacent LTGs are canceled out. When appending the K-LTG layer, we observe that the composite function has the SASAS structure. This is how Algorithm 1 detects the round boundaries.

Unlike Xu et al.'s estimations, we only use a multiset to test whether the currently appended LTG is Type II. In fact, less than $256 \times 500 \approx 2^{17}$ encryptions are sufficient to find all the locations of key-dependent LTG layers from Rbo-Wbaes(10, 500).

As mentioned in [3], with exact round boundary information, we can extract the corresponding round key by mounting the BGE attack in [2] or more efficient attacks such as those in [5, 6]. Note that one can divide a round function of Rbo-Wbaes(10, 500) into 4 column-wise transformations.

Since the encryption key of AES is completely determined by the 16-byte round key of any round, a key-recovery attack against Rbo-Wbaes(10, 500) succeeds with a work factor of the order of 2^{22}.

Conclusion. In 2018, Xu et al. proposed a new whitebox implementation technique called the obfuscated round boundary technique. We analyze the security of their construction and provide an attack algorithm to extract encryption keys from the whitebox implementation even when external encoding is adopted. We utilize Biryukov's early work [4] to present a negative result for whitebox design based on the table lookups. Our attack relies on the observation of the difference between SAS and SASAS structures. Since whitebox implementations using lookup tables inevitably use large tables with the SAS structure, the key information concealed in the tables can be extracted through structural analysis.

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chow S, Eisen P, Johnson H, et al. White-box cryptography and an AES implementation. In: Proceedings of International Workshop on Selected Areas in Cryptography. Berlin: Springer, 2003. 250–270

2 Billet O, Gilbert H, Ech-Chatbi C. Cryptanalysis of a white box AES implementation. In: Proceedings of International Workshop on Selected Areas in Cryptography. Berlin: Springer, 2005. 227–240

3 Xu T, Wu C K, Liu F, et al. Protecting white-box cryptographic implementations with obfuscated round boundaries. Sci China Inf Sci, 2018, 61: 039103

4 Biryukov A, Shamir A. Structural cryptanalysis of SASAS. In: Proceedings of Eurocrypt 2001. Berlin: Springer, 2001. 395–405

5 Baek C H, Cheon J H, Hong H. White-box AES implementation revisited. J Commun Netw, 2016, 18: 273–287

6 Lepoint T, Rivain M, de Mulder Y, et al. Two attacks on a white-box AES implementation. In: Proceedings of International Workshop on Selected Areas in Cryptography. Berlin: Springer, 2014. 265–285