Chinese Journal of Electronics
Vol.32, No.3, May 2023

Recover the Secret Components in a ForkCipher

HOU Tao, ZHANG Jiyan, and CUI Ting

(*PLA SSF Information Engineering University, Zhengzhou 450000, China*)

Abstract — Recently, a new cryptographic primitive has been proposed called ForkCiphers. This paper aims at proposing new generic cryptanalysis against such constructions. We give a generic method to apply existing decompositions againt the underlying block cipher $\mathcal{E}^r$ on the forking variant Fork$\mathcal{E}$-(r—1)-r_0-(r+1—r_0). As application, we consider the security of ForkSPN and ForkFN with secret inner functions. We provide a generic attack against ForkSPN-2-r_0-(4—r_0) based on the decomposition of SASAS. And also we extend the decomposition of Biryukov *et al.* against Feistel networks in SAC 2015 to get all the unknown round functions in ForkFN-r-r_0-r_1 for $r \leq 6$ and r_0+$r_1 \leq 8$. Therefore, compared with the original block cipher, the forking version requires more iteration rounds to resist the recovery attack.

Key words — Recovery attack, ForkCipher, Substitution-permutation network (SPN), Feistel network, Secret design criteria.

I. Introduction

ForkCipher is a new cryptographic primitive proposed by Andreeva *et al.* [1] to maintain the requirement of efficient encryptions and authentication of short messages in resource-constrained devices. These constructions encrypt a plaintext under a secret key, but compute two ciphertexts from this input. In order to achieve a better performance, the middle state is forked, and both ciphertext blocks are computed separately only from the middle. Owing to such construction, ForkCipher provides a new interface called Reconstruction (i.e., half-decryption then half-encryption) which takes one of the ciphertext blocks as input and returns the other one.

As a newly proposed cryptographic primitive, its security estimation attracts several interests. In its proposal, the designers estimated several sound cryptanalytic attacks such as differential cryptanalysis, related-tweakey cryptanalysis, and meet-in-the-middle cryptanalysis [1]. Later in [2], the impossible differential/rectangle/reflection differential/yoyo security of reconstruction of ForkCipher have been estimated. It is important to point out that most successful attacks utilize the reconstruction of the ForkCipher, which is slightly different from the traditional encryption/decryption structure but sometimes seems weaker than the original block cipher.

In this paper, we consider a new variant derived from the traditional ForkCipher by replacing all inner functions with secret ones. In this way, the secret information in ForkCipher increases significantly, and it seems that the security level of such a cipher could be very high. Our target is to recover all inner functions and rebuild the encryption/decryption of the original cipher, which is also called Decomposition. A natural question is: Does such structure have higher security level against existing decomposition methods? Facing this question, we mainly consider the security of ForkCipher against the recovery attack [3]. In such cryptanalysis, all of the internal functions are kept unknown or key-dependent except. Since adversaries can only make very limited assumptions on the secret functions, this attack is applicable to a broader class of cryptosystems. Hence, the recovery attack is quite useful in establishing general design rules for block cipher architectures and in dealing with secret-component-based ciphers.

In this paper, we concentrate on the recovery attacks against ForkCipher with secret round functions. As two direct applications, we take consider of 1) forking a substitution-permutation network (SPN) cipher with secret components (S-boxes and P-layer) in each round and 2) forking a Feistel cipher with secret round functions in each round.

We will explain how to recover all the secret in-

Manuscript Received Oct. 13, 2021; Accepted Jan. 19, 2022. This work was supported by the National Natural Science Foundation of China (61772547, 61902428, 61802438, 61672031) and the Foundation of Key Laboratory of National Defense Science and Technology (61421030107).

 DOI:10.23919/cje.2021.00.368

formation in these two categories. If we use p to denote the pliantext, X to denote the intermediate bifurcation, and c_0, c_1 to denoted ciphertexts, then the ForkCipher can be generalized to ForkCipher-r-r_0-r_1, where r, r_0, and r_1 denote the number of rounds from p to X, from X to c_0, and from X to c_1, respectively. The goal of our work is to recover all the details of such structure.

Related recovery attacks The recovery attack is far from being new. In 2001, Biryukov and Shamir [3] investigated the recovery of iterated SPN ciphers named SASAS and proposed a multiset cryptanalysis. In ASIACRYPT 2014, Biryukov *et al.* proposed a recovery attack on ASASA scheme, which is designed by claiming that it can resist traditional attacks [4]. Soon this result was improved by Dinur *et al.* in [5], and a more efficient recovery algorithm was proposed. In [6], Tyge *et al.* proposed a recovery attack on variant AES (in which the S-box is chosen secretly, but the rest parts kept unchanged). Their attack was based on an improved integral attack, and can recover all the secret information up to 6 rounds. In FSE 2016, Biryukov *et al.* introduced the security estimation of longer generic SP structures with secret inner components and provided several parameters. They claimed that these parameters can achieve a required level of security [7]. The recovery attack against Feistel network was firstly studied by Biryukov *et al.*: If the functions are completely unknown, it is still vulnerable to yoyo attacks for 7-round Feistel networks [8].

Our contribution We put forward a framework of generic recovery attacks against ForkCipher. This result shows that if an r-round underlying cipher $\mathcal{E}$ can be decomposed within complexity $\mathcal{N}$, then the complexity to decompose all the internal round functions of Fork$\mathcal{E}$-$(r-1)$-r_0-r_1 is at most $2\times\mathcal{N}$, the only limit is $r_0+r_1\le r+1$. We extend the SASAS decomposition on ForkSPN and then the Feistel decomposition on ForkFN. It is notable that from any direction of the fork cipher, our results (see Table 1 for details) provide longer recovery compared with the original attack, which indicates that the fork version of a cipher seems more vulnerable to the decomposition attack.

Table 1. Summary of decomposition results

Construction	Method	Time	Ref.
SASAS	Multiset	$\frac{n}{m}2^{3m}$	[3]
ASASA	Integral	$n2^{\frac{3m}{2}}$	[5]
5r-Feistel	Yoyo	2^{2n}	[8]
6r-Feistel	Yoyo	$2^{n2^{n-1}+2n}$	[8]
7r-Feistel	Yoyo	$2^{n2^{n}+2n}$	[8]
ForkSPN-2-r_0-$(4-r_0)$	Multiset	$\frac{n}{m}2^{3m+1}$	Section IV
ForkFN-4-r_0-$(6-r_0)$	Yoyo	2^{2n+1}	Section V
ForkFN-5-r_0-$(7-r_0)$	Yoyo	$2^{n2^{n-1}+2n+1}$	Section V
ForkFN-6-r_0-$(8-r_0)$	Yoyo	$2^{n2^{n}+2n+1}$	Section V

Organization The rest of this paper is organized as follows. Section II introduces several basic concepts. Section III gives a framework of generic recovery attacks against ForkCipher. Section IV and Section V apply our attack on ForkSPN and ForkFN, respectively. Finally, Section VI concludes the paper.

II. Preliminaries

Throughout this paper, we use the following symbols.

$\oplus$ — the XOR operation;

$g\circ f$ — composition of function f and g, i.e., $g\circ f(x)=g(f(x))$;

$\mathcal{E}^r$ — r-round iteration of the block cipher $\mathcal{E}$;

$\mathbb{Z}_m$ — the set of $\{0,1,2,\ldots,m-1\}$.

In our cryptanalysis, we take special interests in substitution-permutation network and Feistel network, which are two of the most popular ciphers nowadays.

Substitution permutation networks (SPN) Let $s_{i,0},\ldots,s_{i,m-1}:\{0,1\}^n\rightarrow\{0,1\}^n$ $(i=1,2,3,\ldots)$ be secret (or key-related) nonlinear bijections, the substitution layer of the i-th round is defined by

$$S_i(x_0,\ldots,x_{m-1})=(s_0(x_0),\ldots,s_{m-1}(x_{m-1}))$$

and the secret linear bijection of the i-th round is defined by $P_i:\{0,1\}^{mn}\rightarrow\{0,1\}^{mn}$, then the i-th round secret SPN cipher is defined as $F(x)=P_i(S_i(x))$.

Feistel networks (FN) Let χ and f_i be two mappings over $\{0,1\}^n\times\{0,1\}^n$, $\chi(x,y)=(y,x)$ and

$$f_i(L_i,R_i):=(L_{i+1},R_{i+1})=(L_i\oplus F_i(R_i),R_i)$$

where $i=0,1,2,\ldots$. Then we define $\chi\circ f_i$ be the i-th round of Feistel cipher, where F_i denotes the secret (or key-related) mappings defined over $\{0,1\}^n$.

By introducing the secret (or keyed) components, the subkey participation can be merged. Thus, we can remove the influence of such traditional secret information and only consider the secret components.

In order to keep the similarity in both encryption and decryption, we omit the last the linear transformation in an r-round SPN structure, and also the final exchanging operation χ of r-round Feistel network.

ForkCipher The basic structure is shown in Fig.1. Let $R_\bullet/R'_\bullet$ be a single round of n-bit (tweakable) block cipher $\mathcal{E}$, then the forking of $\mathcal{E}$, Fork$\mathcal{E}$: $\{0,1\}^n \mapsto \{0,1\}^n \times \{0,1\}^n$, is defined by

$$\text{Fork}\mathcal{E}(p) = (c_0, c_1) := ((\circ_{i=r+1}^{r+r_0} R_i)(X), (\circ_{i=r+1}^{r+r_1} R'_i)(X))$$

where $X = (\circ_{i=1}^{r} R_i)(p)$.

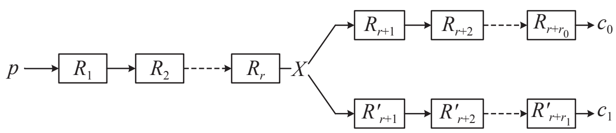


Fig. 1. The encryption of Fork$\mathcal{E}$

We can start from one of the ciphertext block of Fork$\mathcal{E}$ as the input and get the other one by using the inverse of the first half of the computation and then the ordinary round function in the second half. This process is named by Reconstruction [1], i.e.,

$$c_1 = R'_{r+r_1} \cdots \circ R'_{r+1} \circ R_{r+1}^{-1} \cdots \circ R_{r+r_0}^{-1}(c_0)$$

III. Recovering the Secret Components in ForkCipher

In this section, we show the basic idea of recovering secret components in Fork$\mathcal{E}$ for arbitrary block cipher $\mathcal{E}$. If $\mathcal{E}^r$ can be recovered, then the decomposition of Fork$\mathcal{E}$-$(r-1)$-r_0-$(r+1-r_0)$ can also be executed quite efficiently.

As is mentioned by [2], the reconstruction is quite different from the encryption/decryption of the underlying cipher. Therefore, this process may provide us a shortcut. We will first take a closer look at the branching points of ForkSPN and ForkFN.

Branching points of ForkSPN and ForkFN

Assume the two states of encrypting one round after the branching point be C^0 and C^1, respectively, then we have $C^1 = R'_{r+1} \circ R_{r+1}^{-1}(C^0)$. Once we specify the underlying structure R to be one round of SPN or FN, we can combine these two secret layers, i.e., by introducing $\mathcal{S} := S'_{r+1} \circ S_{r+1}^{-1}$ and $\mathcal{F} := f_{r+1} \circ f'_{r+1}$, we have

$$C^1 = P'_{r+1} \circ (S'_{r+1} \circ S_{r+1}^{-1}) \circ P_{r+1}^{-1}(C^0) = P'_{r+1} \circ \mathcal{S} \circ P_{r+1}^{-1}(C^0)$$

for ForkSPN and

$$\begin{aligned}(C_L^1, C_R^1) &= (C_L^0, C_R^0 \oplus F_{r+1}(C_L^0) \oplus F'_{r+1}(C_L^0)) \\ &= f'_{r+1} \circ f_{r+1}(C_L^0, C_R^0) \\ &= \mathcal{F}(C_L^0, C_R^0)\end{aligned}$$

for ForkFN (see Fig.2).

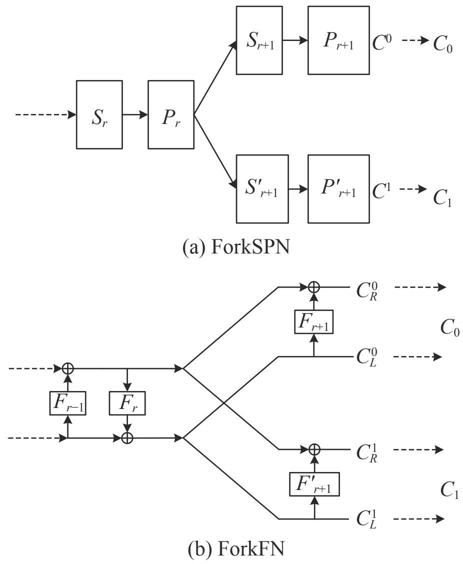


Fig. 2. The forking points of (a) ForkSPN and (b) ForkFN.

Accordingly, for ForkSPN-r-r_0-r_1, we detail the reconstruction by

$$\begin{aligned}c_1 =& S'_{r+r_1} \circ P'_{r+r_1-1} \circ \cdots \circ P'_{r+1} \circ (S'_{r+1} \circ S_{r+1}^{-1}) \circ P_{r+1}^{-1} \\ & \circ \cdots \circ P_{r+r_0-1}^{-1} \circ S_{r+r_0}^{-1}(c_0) \\ =& S'_{r+r_1} \circ P'_{r+r_1-1} \circ \cdots \circ S'_{r+2} \circ P'_{r+1} \circ \mathcal{S} \circ P_{r+1}^{-1} \\ & \circ \cdots \circ P_{r+r_0-1}^{-1} \circ S_{r+r_0}^{-1}(c_0)\end{aligned}$$

and for ForkFN-r-r_0-r_1, the reconstruction process can be rewritten by

$$\begin{aligned}c_1 =& f'_{r+r_1} \circ \chi \circ \cdots \circ f'_{r+2} \circ \chi \circ f'_{r+1} \circ f_{r+1} \circ \chi \\ & \circ \cdots \circ f_{r+r_0-1} \circ \chi \circ f_{r+r_0}(c_0) \\ =& f'_{r+r_1} \circ \chi \circ \cdots \circ f'_{r+2} \circ \chi \circ \mathcal{F} \circ \chi \\ & \circ \cdots \circ f_{r+r_1-1} \circ \chi \circ f_{r+r_0}(c_0)\end{aligned}$$

To summarize, after combining the secret components at the branching point of ForkSPN/ForkFN-$*$-r_0-r_1, one converts the reconstruction into $(r_0 + r_1 - 1)$ iterations of the original structure.

Assume that we can access an r-round-decomposition machine D of the underlying block cipher $\mathcal{E}$, our machine may use some specific property based on the structural weakness of $\mathcal{E}^r$ to recover all the secret components. For instance, we can choose the SASAS attack [3] for SPN structure, or yoyo game attack [8] for 5/6/7-round Feistel structure.

Now we will introduce how to decompose Fork$\mathcal{E}$-$(r-1)$-r_0-r_1 by using two calls of the machine D. The only restriction is that the sum of positive integer r_0 and r_1 equals to $r+1$.

The attack works as follows:

- Call D and recover all the details in the reconstruction of Fork$\mathcal{E}$-$(r-1)$-r_0-r_1, i.e., we get the exact values of $R_{r+1},\dots,R_{r+r_0-1}$, $R'_{r+1},\dots,R'_{r+r_1-1}$ and $R'_r\circ R_r^{-1}$ (or we get an equivalent decomposition).
- Remove the influences of $R_{r+1},\dots,R_{r+r_0-1}$, and call D again for

$$R_r\circ R_{r-1}\circ\cdots\circ R_1$$

then recover all the secret components in the fork cipher.

Note Under the circumstances of getting an equivalent decomposition, without loss of generality we assume that the equivalents are

$$(\mathcal{R}_{r+1},\dots,\mathcal{R}_{r+r_0-1},\mathcal{R}'_{r+1},\dots,\mathcal{R}'_{r+r_1-1},\mathcal{R}'_r\circ\mathcal{R}_r^{-1})$$

We should check whether the middle state decrypted by the partly-equivalent function sequences matches up with the original plaintext. More accurately, for any plaintext p and its ciphertext (c_0,c_1), we compute

$$\hat{c}:=\mathcal{R}_{r+1}^{-1}\circ,\dots,\circ\mathcal{R}_{r+r_0-1}^{-1}(c_0)$$

and check if there exists instance rounds $\mathcal{R}_\bullet$ of the structure $\mathcal{E}$, such that

$$\mathcal{R}_r\circ\mathcal{R}_{r-1}\circ\cdots\circ\mathcal{R}_1(p)=\hat{c}$$

The next step should be executed only if this condition is satisfied.

IV. Recovery Attack Against ForkSPN

In this section, we concentrate on the recovery problem on ForkSPN based on the existing recovery results, namely, the SASAS recovery [3]. It is worthwhile to declare that once a better cryptanalysis result is achieved (for example, with some extra conditions, 5-round SPN can also be recovered [9]), the new decomposition may allow us to recovery more rounds of the fork version similarly.

1. Decomposition machines of SASAS

Firstly, we take a brief overview on the decomposition machines of SASAS, more details may refer to the original works.

In [3], Biryukov and Shamir develop the multiset cryptanalysis to a generalized SPN structure defined by $S_3\circ A_2\circ S_2\circ A_1\circ S_1$, which consists of three substitution layers $S_\bullet$ separated by two affine layers $A_\bullet$. The SASAS attack finds an equivalent three-round SPN structure $g_3\circ A_2^*\circ g_2\circ A_1^*\circ g_1$, which is compatible to the codebook, i.e., for any message p, we have

$$g_3\circ A_2^*\circ g_2\circ A_1^*\circ g_1(p)=S_3\circ A_2\circ S_2\circ A_1\circ S_1(p)$$

Definition 1 Let σ be a permutation defined on $\mathbb{Z}_m$, Λ be a mapping defined on $\{0,1\}^{n\times m}$, if

$$\Lambda(x_0,x_1,\dots,x_{m-1})=(x_{\sigma(0)},x_{\sigma(1)},\dots,x_{\sigma(m-1)})$$

then Λ is called a n-bit word shuffle over $\{0,1\}^{mn}$, where $x_\bullet\in\{0,1\}^n$.

Definition 2 Let L be a mapping over $\{0,1\}^{n\times m}$, if

$$L(x_0,x_1,\dots,x_{m-1})=(T_0(x_0),T_1(x_1),\dots,T_{m-1}(x_{m-1}))$$

then L is said to be an affine mapping layer, where $T_\bullet$ denotes affine bijections over n-bit.

By the results of [3], the functions we obtained and those of the real ones satisfy

$$\left\{\begin{array}{lcl} g_3 &=& S_3\circ L_1\\ A_2^* &=& L_1^{-1}\circ A_2\circ L_2^{-1}\circ\Lambda^{-1}\\ g_2 &=& \Lambda\circ L_2\circ S_2\circ\Lambda\circ L_3^{-1}\\ A_1^* &=& L_3\circ\Lambda^{-1}\circ A_1\circ L_4^{-1}\\ g_1 &=& L_4\circ S_1\end{array}\right.\tag{1}$$

where Λ denotes an arbitrary n-bit word shuffle over $\{0,1\}^{mn}$, and L_1,L_2,L_3,L_4 denote (unknown) affine mapping layers.

2. Decompose ForkSPN-2-2-2

Concerning the reconstruction process of ForkSPN-2-2-2, call the decomposition machine and we find a mapping sequence $g_3,A_1^*,g_4,g'_3,A_2^*,g'_4$, such that

$$S'_4\circ P'_3\circ S'_3\circ S_3^{-1}\circ P_3^{-1}\circ S_4^{-1}=g'_4\circ A_2^*\circ g'_3\circ g_3\circ A_1^*\circ g_4$$

Then by (1), it holds

$$\left\{\begin{array}{lcl} g'_4 &=& S'_4\circ L_1\\ A_2^* &=& L_1^{-1}\circ P'_3\circ L_2^{-1}\circ\Lambda^{-1}\end{array}\right.$$

where Λ denotes an unknown n-bit word shuffle and L_1,L_2 denote two unknown affine mapping layers.

Next we remove g_4 and A_2^* from the reconstruction of ForkSPN-2-2-2. For each plaintext p and its ciphertexts of two branches, c_0 and c_1, we compute $\hat{c}:=A_2^{*-1}\circ g_4'^{-1}(c_1)$. Subsequently, recall the encryption process and we have

$$\hat{c}=(\Lambda\circ L_2)\circ S'_3\circ P_2\circ S_2\circ P_1\circ S_1(p)$$

Observation 1 Let S be a substitution layer, Λ be a n-bit word shuffle over $\{0,1\}^{mn}$ and L be an affine mapping layer on $\{0,1\}^{mn}$, then there exists a substitution layer ς such that $\varsigma\circ\Lambda=\Lambda\circ L\circ S$.

Proof We assume

$$S(x_0,\ldots,x_{m-1}) = (s_0(x_0),\ldots,s_{m-1}(x_{m-1}))$$
$$L(x_0,x_1,\ldots,x_{m-1}) = (T_0(x_0),T_1(x_1),\ldots,T_{m-1}(x_{m-1}))$$
$$\Lambda(x_0,x_1,\ldots,x_{m-1}) = (x_{\sigma(0)},x_{\sigma(1)},\ldots,x_{\sigma(m-1)})$$

If we define $s_i^* := T_i \circ s_i$, then we have

$$\begin{aligned}
&\Lambda \circ L \circ S \circ \Lambda^{-1}(x_0,\ldots,x_{m-1})\\
&= \Lambda \circ L \circ S(x_{\sigma^{-1}(0)},\ldots,x_{\sigma^{-1}(m-1)})\\
&= \Lambda(s_0^*(x_{\sigma^{-1}(0)}),\ldots,s_{m-1}^*(x_{\sigma^{-1}(m-1)}))\\
&= (s_{\sigma(0)}^*(x_0),\ldots,s_{\sigma(m-1)}^*(x_{m-1}))
\end{aligned}$$

Therefore, we end the proof by introducing

$$\varsigma(x_0,\ldots,x_{m-1}) = (s_{\sigma(0)}^*(x_0),\ldots,s_{\sigma(m-1)}^*(x_{m-1}))$$

According to Observation 1, if we introduce a new affine mapping $\theta := \Lambda \circ P_2$, then we can rewrite $\hat{c}$ by $\varsigma_3 \circ \theta \circ S_2 \circ P_1 \circ S_1(p)$, which implies that there exists an SASAS construction, such that

$$\hat{c} = \varsigma_3 \circ \theta \circ S_2 \circ P_1 \circ S_1(p)$$

Thus, all the rest details of ForkSPN-2-2-2 can be recovered by the second call of decomposition machine of SASAS.

V. Recovery Attack Against ForkFN

In [8], Biryukov *et al.* proposed a yoyo-based decomposition against 5-round Feistel network, and this decomposition was extended to 6/7-round Feistel networks at the cost of extra computational complexity. Their attack requires about 2^{2n} time complexity and the full codebook to execute a 5-round recovery against Feistel network. Additionally, for 6- and 7-round decomposition, the time complexities increase to $2^{n2^{n-1}+2n}$ and 2^{n2^n+2n}, respectively.

In this section, we employ the basic 5-round attack [8] as the decomposition machine of Feistel network, and the aim is to recover the inner functions in ForkFN-4-3-3. It is worthwhile declaring that the attack is also compatible with 6/7-round decomposition of the underlying block ciphers. In other words, we may launch recovery attacks against ForkFN-5-4-3 and ForkFN-6-4-4 by a similar extension.

After calling the decomposition machine for the reconstruction phase from c_1 to c_0 (refer to the left part of Fig.3), we may get affine equivalent decompositions G_6 and G_7 instead of the real functions F_6 and F_7 [8], i.e.,

$$\begin{cases} G_6(x) &= F_6(x \oplus \alpha_2) \oplus \alpha_1 \\ G_7(x) &= F_7(x) \oplus \alpha_2 \end{cases}$$

where α_1 and α_2 indicate two unknown constants in $\{0,1\}^n$. Since the influence of these two constants α_1 and α_2 can be absorbed in the recovery process of F_4

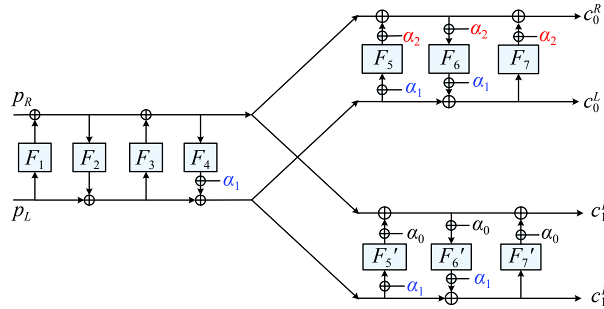


Fig. 3. The equivalent structure of ForkFN-4-3-3

and F_5 (refer to the right part of Fig.3), which means by employing appropriate round functions, we can still receive the ciphertext $(c_0^L \oplus G_6(G_7(c_0^L)\oplus c_0^R), G_7(c_0^L)\oplus c_0^R)$ by encrypting (p_L, p_R) with 5 rounds of Feistel network. Hence, by a second call of the 5-round FN decomposition, we get all the details remaining in ForkFN-4-3-3. In this way, the whole computation can be completed within time complexity 2^{2n+1} and memory 2^{2n}.

By a similar proof, the decompositions of ForkFN-5-r_0-$(7-r_0)$ and ForkFN-6-r_0-$(8-r_0)$ respectively cost twice as much as each of the underlying block cipher structures.

VI. Conclusions

In this paper, we mainly consider the decomposition problem of a ForkCipher, which is a new cryptographic primitive proposed especially for efficient encryption and authentication of small messages. Our results indicate that an r-round recovery against the underlying block cipher $\mathcal{E}$ can be transformed into the recovery of the forking variant Fork$\mathcal{E}$-$(r-1)$-r_0-$(r-r_0+1)$. Amazingly, the recovery complexity of Fork$\mathcal{E}$ is only about twice as much as recovering the original cipher structure. Since our results propose a new attack against ForkCipher capable of recovering the whole details of the secret functions without making any assumptions, it can be treated as a theoretical generic attack against ForkCipher. We have verified our attack on ForkSPN and ForkFN by experiments: Call the underlying structural decomposition twice, one receive a full recovery of these fork versions.

The architecture of Fork$\mathcal{E}$ takes out two processes of the dataflow, namely, the encryption process and the reconstruction process. Compared with the original decomposition against the underlying block cipher structure, our recovery works for more rounds from any dataflow direction. This work makes further understanding of the security of the architecture of ForkCipher: To achieve the same security level against the structural cryptanalysis, the forking version seems need more iterations than the original one.

References

[1] E. Andreeva, R. Reyhanitabar, K. Varici, *et al.*, "Forking a blockcipher for authenticated encryption of very short messages," *Cryptology ePrint Archive*, Paper 2018/916, Available at: *https://eprint.iacr.org/2018/916*, 2019.

[2] S. Banik, J. Bossert, A. Jana, *et al.*, "Cryptanalysis of ForkAES," in *Proceedings of the 17th International Conference on Applied Cryptography and Network Security*, Bogota, Colombia, pp.43–63, 2019.

[3] A. Biryukov and A. Shamir, "Structural cryptanalysis of SASAS," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, Innsbruck, Austria, pp.395–405, 2001.

[4] A. Biryukov, C. Bouillaguet, and D. Khovratovich, "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*, Kaoshiung, China, pp.63–84, 2014.

[5] I. Dinur, O. Dunkelman, T. Kranz, *et al.*, "Decomposing the ASASA block cipher construction," *Cryptology ePrint Archive*, Paper 2015/507, Available at: https://eprint.iacr.org/2015/507, 2015.

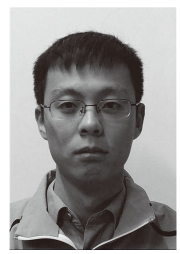
[6] T. Tiessen, L. R. Knudsen, S. Klbl, *et al.*, "Security of the AES with a secret S-Box," in *Proceedings of the 22nd International Workshop on Fast Software Encryption*, Istanbul, Turkey, pp.175–189, 2015.

[7] A. Biryukov, D. Khovratovich, and L. P. Perrin, "Multiset-algebraic cryptanalysis of reduced Kuznyechik, Khazad, and secret SPNs," *IACR Transactions on Symmetric Cryptology*, vol.2016, no.2, pp.226–247, 2016.

[8] A. Biryukov, G. Leurent, and L. Perrin, "Cryptanalysis of Feistel networks with secret round functions," in *Proceedings of the 22nd International Conference on Selected Areas in Cryptography*, Sackville, Canada, pp.102–121, 2015.

[9] A. Biryukov and D. Khovratovich, "Decomposition attack on SASASASAS," *Cryptology ePrint Archive*, Paper 2015/646, Available at: *https://eprint.iacr.org/2015/646*, 2015.

HOU Tao was born in 1996. He received the M.S. degree in 2021. His research interests include block cipher design and cryptanalysis.
(Email: houtao 1996@126.com)



ZHANG Jiyan was born in 1996. He received the Ph.D. degree in 2022. His research interests include block cipher design and cryptanalysis.
(Email: xdzhangjiyan@126.com)

CUI Ting (corresponding author) was born in 1985. He received the Ph.D. degree in 2013 and is currently a Professor at the PLA SSF Information Engineering University, China. His research interests include block cipher design and cryptanalysis.
(Email: cuiting_1209@126.com)