Breaking The FF3 Format-Preserving Encryption Standard Over Small Domains

F. Betül Durak[1] and Serge Vaudenay[2]

[1] Rutgers University
Department of Computer Science
fbdurak@cs.rutgers.edu

[2] Ecole Polytechnique Fédérale de Lausanne (EPFL)
LASEC - Security and Cryptography Laboratory
Lausanne, Switzerland
serge.vaudenay@epfl.ch

Abstract. The National Institute of Standards and Technology (NIST) recently published a Format-Preserving Encryption standard accepting two Feistel structure based schemes called FF1 and FF3. Particularly, FF3 is a tweakable block cipher based on an 8-round Feistel network. In CCS 2016, Bellare et. al. gave an attack to break FF3 (and FF1) with time and data complexity $\mathsf{O}(\mathsf{N}^5 \log(\mathsf{N}))$, which is much larger than the code book (but using many tweaks), where N^2 is domain size to the Feistel network. In this work, we give a new practical total break attack to the FF3 scheme (also known as BPS scheme). Our FF3 attack requires $\mathsf{O}(\mathsf{N}^{\frac{11}{6}})$ chosen plaintexts with time complexity $\mathsf{O}(\mathsf{N}^5)$. Our attack was successfully tested with $\mathsf{N} \leqslant 2^9$. It is a slide attack (using two tweaks) that exploits the bad domain separation of the FF3 design. Due to this weakness, we reduced the FF3 attack to an attack on 4-round Feistel network. Biryukov et. al. already gave a 4-round Feistel structure attack in SAC 2015. However, it works with chosen plaintexts and ciphertexts whereas we need a known-plaintext attack. Therefore, we developed a new generic known-plaintext attack to 4-round Feistel network that reconstructs the entire tables for all round functions. It works with $\mathsf{N}^{\frac{3}{2}} \left(\frac{\mathsf{N}}{2}\right)^{\frac{1}{6}}$ known plaintexts and time complexity $\mathsf{O}(\mathsf{N}^3)$. Our 4-round attack is simple to extend to five and more rounds with complexity $\mathsf{N}^{(\mathsf{r}-5)\mathsf{N}+\mathsf{o}(\mathsf{N})}$. It shows that FF1 with $\mathsf{N} = 7$ and FF3 with $7 \leqslant \mathsf{N} \leqslant 10$ do not offer a 128-bit security. Finally, we provide an easy and intuitive fix to prevent the FF3 scheme from our $\mathsf{O}(\mathsf{N}^5)$ attack.

1 Introduction

Format-Preserving Encryption (FPE) provides a method to encrypt data in a specific *format* into a ciphertext of the same *format*. A *format* in FPE schemes refers to a finite set of characters such as the decimal (or binary) numerals or alpha-numerals along with the length of the sequence of the characters that form the plaintexts. FPE has been staging in applied cryptography community due to the desirable functionality. It secures data while keeping the database scheme intact. For instance, given a legacy database system, upgrading the database security requires a way for encrypting credit card numbers (CCN) or social security numbers (SSN) in a transparent way to its applications.

Brightwell and Smith [9] introduced a first known format-preserving encryption which was termed as *data-type preserving encryption* in 1997. They wanted to encrypt an existing database to let all the applications access encrypted data just as they access non-encrypted data. Their solution for this was reduced to preserve the particular datatype of entries in the databases. The term *format-preserving encryption* is due to Terence Spies from Voltage Security [21]. Though FPE dates back to late 90's, the demand to make FPE based databases has created an active area of research during last few years. There have been many techniques proposed to build FPE schemes such as prefix cipher, cycle walking, Feistel network, Feistel modes [2, 4, 5, 7, 16, 20, 21]. The complete list of FPE schemes for small domain size along with their description and their security level can be found in a synopsis by Rogaway [18, p. 6,7]. In his list, Rogaway considers the schemes that are built with pseudorandom functions (that itself might be constructed from block ciphers).

Probably, it is natural to build FPE schemes based on a Feistel network (FN) since it can be used with already existing conventional block ciphers, such as AES. Indeed, the National Institute of Standards and Technology (NIST) published an FPE standard [1] (finalized in March 2016) that includes two-approved Feistel-based FPE schemes: FF1 [5] and FF3 [8]. Both are expected to offer a 128-bit security. In this work, we are particularly interested in the attacks for breaking the FN-based standard FF3 [1] and attacks against Feistel network. The former attack utilizes the latter that is designed as a generic round-function-recovery attack.

The FF3 construction is an 8-round FN that uses a tweak XORed with a round counter as an input to the block cipher. The XOR operation guarantees that round functions are pairwise different. This is usually called "domain separation". The security of FF3 asserts that it achieves several cryptographic goals including chosen-plaintext security or even PRP-security against an adaptive chosen-ciphertext attack under the assumption that the underlying round function is a good pseudorandom function (PRF). Our work shows that its security goal has not met even when the round functions are replaced by secure PRFs and gives a round-function-recovery attack on FF3.

Our Contributions. Our work covers three significant contributions. **(a).** We give a total practical break to 8-round Feistel network based FF3 FPE standard over a small domain. Our attack exploits the "bad domain separation" in FF3. Namely, the specific design choice of FF3 allows us permuting the round functions by changing the tweak and it leads us to develop a slide attack (using only two tweaks). The attack works with chosen plaintexts and tweaks when the message domain is small. It requires $\mathsf{O}(\mathsf{N}^{\frac{7}{4}+\frac{1}{4\mathsf{L}}})$ chosen plaintexts and two tweaks, with time complexity $\mathsf{O}(\mathsf{N}^5)$, where N^2 is input domain size to the Feistel network and L is a parameter in our attack which is typically set to $\mathsf{L} = 3$ in experimental results. Luckily, the fix to prevent FF3 against our attack is quick and easy to maintain without changing the main structure of the scheme. **(b).** While we form our slide attack to break FF3, we develop a new generic known-plaintext attack on 4-round Feistel networks and we insert it in our slide attack. Our techniques to develop a 4-round attack is novel and different than previously known attacks on Feistel networks. In our attack, we compute the full recovery of round functions with $\mathsf{N}^{\frac{3}{2}}\left(\frac{\mathsf{N}}{2}\right)^{\frac{1}{2\mathsf{L}}}$ known plaintext and time complexity $\mathsf{O}(\mathsf{N}^{2+\frac{3}{\mathsf{L}}})$ for four rounds. **(c).** We utilize our 4-round FN attack to extend the round function recovery on more rounds. Due to the generic and known plaintext nature of our 4-round FN attack, we easily adapt it to a chosen-plaintext attack to apply it on 5 and more rounds Feistel structures. Our attack shows that neither FF1 with $\mathsf{N} = 7$ nor FF3 with $7 \leqslant \mathsf{N} \leqslant 10$ (even with our fix) offer a 128-bit security.

Overview Of Previous Works. A security for message recovery in FPE constructions along with many other notions for FPE was first defined by Bellare et. al. [4]. A recent work by Bellare et. al. [3] gives a practical message recovery attack on NIST standard Feistel-based FPE schemes (both FF1 and FF3) on small domain sizes. In their work, however, the security definition they consider is under the new message recovery security that they define in the same work. Briefly, consider two messages X and X' which share the same right (or left) half of the messages. In their attack, the adversary is given X' together with the encryption of X and X' under q tweaks. The adversary wins if it can fully recover X, in particular, its unknown half. The attack by Bellare et. al. uses a data complexity that exceeds the message space size. Clearly stating, their work shows that Feistel-based FPE with the standardized number of rounds does not achieve good enough security on small domain sizes.

The attack by Bellare et. al. works using $\mathsf{O}(\mathsf{N}^5 \log \mathsf{N})$ data and time complexity with many tweaks on eight rounds. This is quite interesting when the amount of data is limited for each tweak. It is a decryption attack. Our attack herein is more traditional. It uses only two tweaks, but $\mathsf{O}(\mathsf{N}^{\frac{11}{6}})$ chosen plaintexts with $\mathsf{O}(\mathsf{N}^5)$ time complexity. We recover the entire codebook (for both tweaks).

To apply the slide attack to recover the entire round functions of Feistel networks, we develop a generic known-plaintext attack on 4-rounds.

Since its invention, Feistel networks have created active research areas for cryptographers (both in theory and in practice) due to its applications and influence on the development of major constructions such as DES. The security for Feistel networks has been investigated for very long time and there already exist interesting results for cryptanalysis. The security of Feistel schemes aims either to distinguish a Feistel scheme from a random permutation or to recover the round functions. In their famous work [15], Luby and Rackoff proved the indistinguishability of 3-round Feistel network against chosen-plaintext attacks and 4-rounds against chosen-plaintext and ciphertext attacks for the number of queries $\mathsf{q} \ll \sqrt{\mathsf{N}}$, where N^2 is the size of the input domain. The directions derived from this result tried to improve the security bounds until $\mathsf{q} \ll \mathsf{N}$ (that is called the "birthday bound") which was a natural bound from information theory.[3] A work by Patarin [17], using the mirror theory, showed improved proofs and stronger security bounds for four, five, and six rounds Feistel networks. Namely, for $q \ll N$, four rounds are secure against known-plaintext attacks, five rounds are secure against chosen-plaintext attacks, and six rounds are secure against chosen-plaintext and ciphertext attacks.

From an information theory viewpoint, we could recover all functions in time $N^{O(N)}$ by exhaustive search. As far as we know, there is no efficient generic attack which is polynomial in N on the Feistel scheme with $q \sim N$. Our attack uses $q \sim N^{\frac{3}{2}}$ and is polynomial in N with known plaintexts up to four rounds.

A recent work by Dinur et. al. [11] gives a new attack on Feistel structures for more than four rounds to recover the round keys with a few known plaintext/ciphertext pairs when the i^{th} round uses $x \mapsto F_i(x \oplus k_i)$, where F_i is public whereas k_i is being kept secret. Here, we focus on the case where each round function is secret in a balanced 2-branch Feistel scheme. Furthermore, we do not restrict to the XOR addition. Our results also apply to Feistel schemes with modular addition. The new cryptanalysis results against Feistel networks with modular addition for four and five rounds are presented in a recent work by Biryukov et. al. [6]. For four rounds, they achieve the full recovery of round functions with data complexity $O(N^{\frac{3}{2}})$ with a guess and determine technique. However, their attack uses chosen plaintexts and ciphertexts. We summarize their results and ours on Table 1.

# rounds	**mode**	**time**	**data**	**ref**
3	known-plaintext	N	N	Section 4.1
4	chosen-plaintext and ciphertext	$N^{\frac{3}{2}}$	$N^{\frac{3}{2}}$	[6]
4	known-plaintext (tested for $L = 3$)	$N^{2+\frac{3}{L}}$	$N^{\frac{3}{2}+\frac{1}{2}L}$	Section 4.2
5	chosen-plaintext and ciphertext	$N^{N^{\frac{3}{4}}}$	N^2	[6]
5	chosen-plaintext	$N^{O(N^{\frac{1}{2}})}$	$N^{\frac{3}{2}+\frac{1}{2}L}$	Section 4.3
$r \geqslant 6$	chosen-plaintext	$N^{(r-5)N}$	$N^{\frac{3}{2}+\frac{1}{2}L}$	Section 4.3

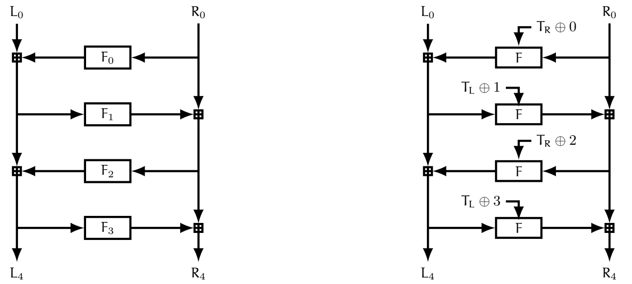
Table 1. Round-function-recovery attacks against balanced Feistel schemes with two branches of $\log_2 N$ bits and any addition rule (we omitted polynomial terms in $\log N$)

Structure of the Paper. In Section 2 and Section 3, we give the details of FF3 construction and Tweakable Encryption, respectively. In Section 4, we develop our new generic attack for Feistel structure on specifically 4-rounds and extend it on 5 and more rounds. In Section 5, we give our complete slide attack to a NIST standard FF3 scheme.

[3] In an r-round FN, q samples give $2q \log_2 N$ bits of information but functions are defined by a table of $rN \log_2 N$ bits. Thus, $q = \frac{r}{2}N$ queries is enough to reconstruct the round functions, in theory.

2 The FF3 Scheme

A Tweakable Format-Preserving Encryption (TFPE) scheme is a block cipher that preserves the format of the domain in the output. A TFPE function $\mathsf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \mapsto \mathcal{X}$ is defined from a key space $\mathcal{K}$, a tweak space $\mathcal{T}$, and a domain $\mathcal{X}$ to the same domain $\mathcal{X}$. We are particularly interested in a TFPE scheme by Brier, Peyrin, and Stern (depicted in Fig. 1 (b)) [8] whose design is based on Feistel network depicted in Fig. 1 (a). It is named as FF3 in the NIST standards.



(a) Feistel Network (b) FF3 Encryption

Fig. 1. 4-round Feistel Network and FF3 Encryption

We use the following notations for the rest of the paper. The domain $\mathcal{X}$ consists of strings of characters; s represents the cardinality of the set S of characters and b represents the length of the messages in the domain $\mathcal{X}$. For example, the credit card numbers (CCNs) consists of 16 digits of decimal numerals with $\mathsf{S} = \{0, 1, \ldots, 9\}$, $\mathsf{s} = 10$ and $\mathsf{b} = 16$ where we have $10^{16} \cong 2^{54}$ possible distinct numeral strings. We set the minimum length of the message block $\mathtt{minlen} = 2$ and the maximum length of the message block to $\mathtt{maxlen} = \lfloor \log_{\mathsf{s}}(2^{\mathsf{f}-32}) \rfloor$, where f is the input/output size of the round function used in Feistel scheme in FF3. [4] We represent the number of rounds in the scheme with w.

Unlike standard Feistel schemes which use the exclusive or (XOR) (denoted by $\oplus$), FF3 uses the modular addition that is denoted by $\boxplus$.

We define the following notations for three functions:

$\mathbf{STR}_{\mathsf{s}}^{\mathsf{b}}$ **:** a function that maps an integer x where $0 \leqslant \mathsf{x} < \mathsf{s}^{\mathsf{b}}$ to a string of length b in base s with most significant character first, e.g. $\mathsf{STR}_{12}^{4}(554) = 03A2$.

$\mathbf{NUM}_{\mathsf{s}}$ **:** a function that maps a string X to an integer x such that $\mathsf{STR}_{\mathsf{s}}^{\mathsf{b}}(\mathsf{x}) = \mathsf{X}$. For instance, $\mathsf{NUM}_2(00011010) = 26$.

$\mathbf{REV}(\mathbf{X})$ **:** a function that reverses the order of the characters of string X.

[4] We consider here the FF3 block cipher. However, there is a mode of operation for FF3 allowing variable-length messages in the original paper [8].

The length of string X is denoted by $|X|$. The concatenation of strings is denoted by $||$. The first (left-most) character of string X is $X[0]$. The i^{th} one is $X[i-1]$. We denote $X[a \cdots b]$ the substring of X formed with $X[a]X[a+1] \cdots X[b]$.

The FF3 uses a tweakable block cipher as a round function, $F_K(T, X) = Y$ with $X, Y \in \{0, 1, \ldots, 2^f - 1\}$ and $T \in \{0, 1\}^{32}$, where K is a key and T is one half of the FF3 tweak with an offset.

Algorithm 1: FF3 Encryption

```
   Input  : string $X$ in base $s$ of length $b$ such that
            $b \in [minlen \cdots maxlen]$, a tweak bit string $T$ such that
            $|T| = 64$.
   Output: string $Y$ such that $|Y| = b$
1  Let $\ell = \lceil \frac{b}{2} \rceil$; $r = b - \ell$.
2  Let $L_0 = NUM_s(REV[X[1 \cdots \ell]])$ and $R_0 = NUM_s(REV[X[\ell + 1 \cdots b]])$
3  Let $T_L = T[0 \cdots 31]$ and $T_R = T[32 \cdots 63]$
4  foreach $i = 0 \cdots w - 1$ do
5      if $i$ is even then
6          $L_{i+1} = L_i \boxplus F_K(T_R \oplus STR_2^{32}(i), R_i) \pmod{s^\ell}$
7          $R_{i+1} = R_i$
8      end
9      else
10         $R_{i+1} = R_i \boxplus F_K(T_L \oplus STR_2^{32}(i), L_i) \pmod{s^r}$
11         $L_{i+1} = L_i$
12     end
13 end
14 return $REV[STR_s^\ell(L_w)||REV[STR_s^r(R_w)]$
```

In lines 1-2, the encryption algorithm splits the input X into two substrings L_0 and R_0. In lines 5-8 (respectively in lines 10-12), the algorithm first takes the tweak T_R (respectively T_L) XORed with the encoded round index i and R_i (respectively L_i) to input tweakable PRF F_K. Second, it applies modular addition of the output of F_K to L_i (respectively R_i).

For simplicity and by abuse of notations, we say that FF3 encrypts the plaintext (L_0, R_0) into the ciphertext (L_w, R_w) with tweak (T_L, T_R), so that we only concentrate on lines 4-14. We illustrate the 4-round FF3 scheme in Fig. 1 (b).

In concrete proposal, $w = 8$, $f = 128$ and

$$F_K(T, X) = NUM_2(AES_K(T \mid\mid STR_2^{f-32}(X)))$$

where AES maps an f-bit bitstring to an f-bit bitstring [1].

3 Tweakable Encryption

A tweakable block cipher (TBC) is a tuple $(\mathcal{K}, \mathcal{E}_K(\cdot, \cdot), \mathcal{D}_K(\cdot, \cdot))$ formed of three algorithms for key generation, encryption, and decryption with a key K; all efficiently computable algorithms. We follow the notion of security from [13] as chosen-plaintext-secure (CPA) tweakable block cipher.

Definition 1. *A TBC is a* $(\mathsf{q}, \mathsf{t}, \epsilon)$*-CPA-secure cipher if for any probabilistic time adversary* $\mathcal{A}$ *limited to* t *steps and* q *oracle queries, the advantage of distinguishing TBC from* Π *is bounded by* ϵ*:*

$$\mathsf{Adv}^{\mathsf{TBC}}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\mathcal{E}_{\mathsf{K}}(\cdot,\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\Pi(\cdot,\cdot)} = 1\right]\right| \leqslant \epsilon$$

where $\mathsf{K} \in \mathcal{K}$ *is selected at random and* $\Pi(\mathsf{T}, \cdot)$ *is defined as a random permutation for every* T*.*

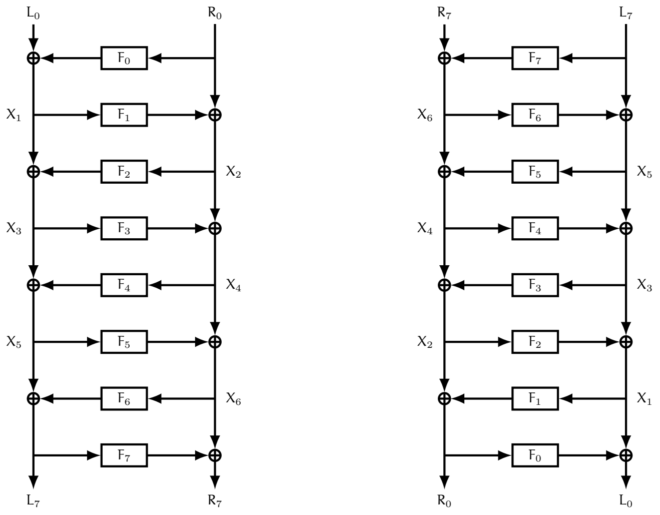
In the standard model, the tweakable block ciphers [4, 14] are used to construct tweakable format-preserving encryption schemes since tweakable encryptions provide better security bounds for tweakable FPE in terms of the number of chosen plaintext/ciphertext to attack the system [4].

It is underlined in [8] that using the same round function F twice during an encryption process can introduce some security vulnerability to the system. So, the domain of the tweaks in different rounds must be separated. For this, the scheme in [8] XORs tweaks with a round counters. However, this way to separate domains is not fully effective. Indeed, the tweaks are known to the adversary and are under adversary's control in chosen-tweak attacks. Consider two 4-round Feistel networks with tweaks $\mathsf{T_R}$ and $\mathsf{T_L} = \mathsf{T_R} \oplus \mathsf{STR}_2^{32}(1)$. For the first round, we have the tweak $\mathsf{T_R} \oplus \mathsf{STR}_2^{32}(0) = \mathsf{T_R}$ and the second round we have $\mathsf{T_L} \oplus \mathsf{STR}_2^{32}(1) = \mathsf{T_R}$. Then, for the third round $\mathsf{T_R} \oplus \mathsf{STR}_2^{32}(2)$ and fourth round $\mathsf{T_L} \oplus \mathsf{STR}_2^{32}(3) = \mathsf{T_R} \oplus \mathsf{STR}_2^{32}(2)$. We observe the following behavior: round $2i$ and $2i+1$ uses the same function $\mathsf{F_i} = \mathsf{F_K}(\mathsf{T_R} \oplus \mathsf{STR}_2^{32}(2i), \cdot)$.

For a variant of FF3 with $\oplus$ instead of $\boxplus$, we present a trivial attack: Consider an FF3 encryption with a key $\mathsf{K} \in \mathcal{K}$, a tweak $\mathsf{T} = \mathsf{T_L}||\mathsf{T_R} \in \mathcal{T}$ and domain $\mathcal{X}$. Each round i defines a random function $\mathsf{F}_i = \mathsf{F_K}(\mathsf{T_R} \oplus \mathsf{STR}_2^{32}(i), \cdot)$ for i even ($\mathsf{F}_i = \mathsf{F_K}(\mathsf{T_L} \oplus \mathsf{STR}_2^{32}(i), \cdot)$ for i odd). We use the encryption with an input message $\mathsf{X} = (\mathsf{L}_0, \mathsf{R}_0)$ and output ciphertext $\mathsf{Y} = (\mathsf{L}_w, \mathsf{R}_w)$ with output X_i from each round in Fig. 2 (a). We assume that b is even so that $\ell = r$. Now, we take the ciphertext Y from Fig. 2 (a) and reverse it into $(\mathsf{L}'_0, \mathsf{R}'_0) = (\mathsf{R}_w, \mathsf{L}_w)$ to encrypt it with a new tweak $\mathsf{T}' = \mathsf{T_R} \oplus \mathsf{STR}_2^{32}(w-1)||\mathsf{T_L} \oplus \mathsf{STR}_2^{32}(w-1) \in \mathcal{T}$. We show this encryption in Fig. 2 (b). We assume that w is a power of two (Fig. 2 uses $w = 8$). With given encryption, we obtain the round functions $\mathsf{F}'_i = \mathsf{F}_{w-1-i}$ as shown on Fig. 2 (a). More precisely, the attack works as follows:

- ○ Encrypt $(\mathsf{L}_0, \mathsf{R}_0)$ with the tweak T to get $(\mathsf{L}_w, \mathsf{R}_w)$.
- ○ Encrypt $(\mathsf{R}_w, \mathsf{L}_w)$ with the tweak T' to get $(\mathsf{L}', \mathsf{R}')$.
- ○ If $\mathsf{L}' = \mathsf{R}_0$ and $\mathsf{R}' = \mathsf{L}_0$, output 1. Otherwise, output 0.

The adversary always outputs 1 with $\mathcal{E}_\mathsf{K}$. It outputs 1 with $\Pi(\cdot,\cdot)$ with probability $\frac{1}{\mathsf{s}^\mathsf{b}}$. Therefore, the advantage is $1 - \frac{1}{\mathsf{s}^\mathsf{b}}$.



(a) (b)

Fig. 2. Trivial Attack on 8-round FF3 Encryption with $\oplus$ instead of modular addition $\boxplus$.

4 Known-Plaintext Round-Function-Recovery Attack on Feistel Scheme

In this section, we define the Feistel network over a group of order N. Typically, this group is $\mathbb{Z}_{\mathsf{N}}$. Later in Section 5, we assume b is even and $\mathsf{N} = \mathsf{s}^{\frac{\mathsf{b}}{2}}$.

First of all, we observe that the round functions are not uniquely defined by the codebook. Namely, if $(\mathsf{F}_0, \ldots, \mathsf{F}_{r-1})$ is a solution to map given sample plaintexts to the corresponding ciphertexts, then we can construct many other solutions. Indeed, for any set of values $\alpha_0, \ldots, \alpha_{r-1}$ such that $\alpha_1 + \alpha_3 + \alpha_5 + \cdots = \alpha_0 + \alpha_2 + \alpha_4 + \cdots = 0$, we can define

$$\mathsf{F}'_j(u) = \mathsf{F}_j(u - \alpha_{j-1} - \alpha_{j-3} - \alpha_{j-5} - \cdots) + \alpha_j$$

for all j and u to obtain another solution. Therefore, we can fix one point arbitrarily in $\mathsf{F}_0, \ldots, \mathsf{F}_{r-3}$ when looking for a solution. All the other solutions are obtained by the above transformation of the round functions.

The rest of the section is organized as follows: in Section 4.1, we give a heuristic attack for 3-round FN and analyze its time complexity. We report the ratio of success recovery in Fig. 3 with the parameters the attack takes. In Section 4.2, we give an attack for 4-round FN that leverage our 3-round attack. The correctness and further analysis is presented with formally stated lemmas. In Section 4.3, we expand our attack for five rounds and more and derived the time complexities.

4.1 Round-Function-Recovery on 3-Round Feistel Scheme

Consider a 3-round Feistel Scheme with three round functions F_0, F_1, F_2 and modular addition. Given x and y in $\mathcal{X}$, we define:

$$\begin{aligned} c &= x + F_0(y), \\ t &= y + F_1(c), \\ z &= c + F_2(t). \end{aligned} \tag{1}$$

Due to the symmetry of the set of solutions (F_0, F_1, F_2) (as already observed), we can fix F_0 on one point arbitrarily. The idea of our attack is to concentrate on data for which we know how to evaluate F_0 so that we can deduce the output for the round function F_2. Then, we concentrate on data for which we know how to evaluate F_2 and we deduce more points in F_0. We continue by alternating the deduction between F_0 and F_2 until we recover them all. When we continue iterating as described, we can fully recover the tables for all three round functions (F_0, F_1, F_2). Our attack is presented in Algorithm 2 in more detail.

Algorithm 2: (F_0, F_1, F_2) Recovery Attack

```
Collect a set $S$ of tuples $(xyzt)$ of size $\theta N$.
Take a subset $S_1 \subseteq S$ of size $\theta$ such that $y$ is constant in $S_1$.
Fix $F_0(y) = 0$ arbitrarily and deduce $\theta$ tuples $(cyzt)$ in $S_1$ by
  $c = x + F_0(y)$. We collect $\theta$ equations of the form $F_2(t) = z - c$.
Take the subset $S_2 \subseteq S$ of all $(xyzt) \in S$ such that $\exists(x'y'z't') \in S_1$ with
  $t = t'$. The expected size of $S_2$ is $\theta^2$.
Using the $\theta$ points of $F_2$, we deduce $\theta^2$ tuples $(xyct)$ by $c = z - F_2(t)$.
  From these tuples, we obtain $\theta^2$ equations of the form $F_0(y) = c - x$.
Take the subset $S_3 \subseteq S$ of all $(xyzt) \in S$ such that $\exists(x'y'z't') \in S_2$ with
  $y = y'$. The expected size of $S_3$ is $\theta^3$.
Using the $\theta^2$ points of $F_0$, we deduce from $\theta^3$ tuples $(cyzt)$...
We iterate through $S_1 \subseteq S_3 \subseteq S_5 \subseteq \cdots \subseteq S$ and $S_2 \subseteq S_4 \subseteq \cdots \subseteq S$ to
  complete the tables of $F_0$ and $F_2$.
```

We model our set S as a bipartite graph with two parties of N vertices (one for the y's and the other for the t's) and edges for each (y, t) pair represented by tuples from S. What our algorithm does is just to look for a connected component of a random starting point y with complexity $O(\theta N)$. Following the theory of random graphs [19], we have θN random edges so that the graph is likely to be fully connected when $\theta \approx \ln(N)$. For a constant $\theta \geqslant 1$, it is likely to have a giant connected component. This component corresponds to a constant fraction of the tables of F_0 and F_2. Therefore, after $\log_\theta N$ iterations, we can reconstruct F_0 and F_2 which allow us to reconstruct F_1. For any y, we can see that it does not appear in S with probability $\left(1 - \frac{1}{N}\right)^{\theta N} \approx 1 - e^{-\theta}$. Thus, we can only hope to recover a fraction $1 - e^{-\theta}$ of the table of F_0. The same holds for F_1 and F_2. **Therefore, with data and time complexity N, we recover a good fraction of all tables. With data and time complexity $N \ln N$, we recover the full tables with good probability.**

We implemented our attack. On Fig. 3, we plot the average fraction of recovered F_0 values depending on θ for several values of N. For this, we computed an average over 10,000 independent runs. For $\theta = 1$, the fraction is about 40%. We also plot the fraction of the trials which fully recovered all functions. These two values can be taken as an approximation of the expected fraction of recovered table for F_0 and the probability to fully recover all functions, respectively. As we can see, the first value does not depend so much on N (we have a giant connected component for θ around 1), but the second one jumps for θ proportional to $\ln \mathsf{N}$ (the graph becomes fully connected). For $\theta = \ln \mathsf{N}$, the probability is roughly $\frac{1}{3}$.

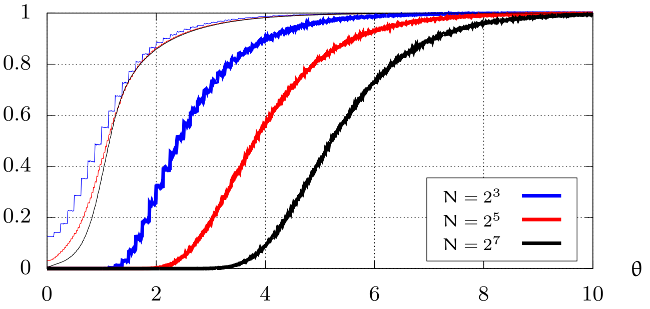


Fig. 3. Fraction of recovered F_0 depending on θ in the 3-round attack (in thin) and fraction of experiments which fully recovered all functions (in bold) over 10,000 trials.

4.2 Round-Function-Recovery on 4-Round Feistel Scheme

In this section, we give an attack to fully recover the round functions of a 4-round Feistel scheme.

Consider a 4-round Feistel scheme with round functions $\mathsf{F}_0, \mathsf{F}_1, \mathsf{F}_2, \mathsf{F}_3$. Given x and y in $\mathcal{X}$, we define the following equations (see Fig. 4 (a)):

$$\begin{aligned} c &= x + \mathsf{F}_0(y), \\ d &= y + \mathsf{F}_1(c), \\ z &= c + \mathsf{F}_2(d), \\ t &= d + \mathsf{F}_3(z). \end{aligned}$$

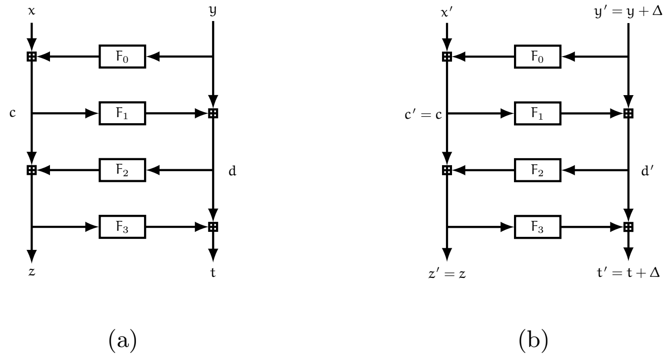
Assume that we collected M random pairwise different plaintext messages (xy). We collect the pairs:

$$V = \{(xy, x'y') \mid z' = z, t' - y' = t - y, xy \neq x'y'\}$$

and,

$$V_{good} = \{(xy, x'y') \mid z' = z, c' = c, xy \neq x'y'\}$$

where c, d, z, t (respectively c', d', z', t') are defined from (xy) (respectively form $(x'y')$) as above. We define $\mathsf{Label}(xy, x'y') = x - x'$.



(a) (b)

Fig. 4. 4-round Feistel Scheme Attack

We form a directed graph $G = (V, E)$ with the vertex set V as defined above. We take $(x_1y_1x_1'y_1', x_2y_2x_2'y_2') \in E$ if $y_1' = y_2$ (i.e. a pair of tuples $x_1y_1x_1'y_1'$ is connected to a pair $x_2y_2x_2'y_2'$ if the y_2 in the second message in former tuple is same as in the first message in latter tuple). Furthermore, we let $E_{good} = (V_{good} \times V_{good}) \cap E$ and define the sub-graph $G_{good} = (V_{good}, E_{good})$.

Then, we have the following Lemma with four properties:

Lemma 1. *Given a graph* G *with a vertex set* V *defined as above:*

1. $V_{good} \subseteq V$.
2. *If* $(xy, x'y') \in V$, *then* $y \neq y'$.
3. *If* $(xy, x'y') \in V_{good}$, *then* $F_0(y') - F_0(y) = \mathsf{Label}(xy, x'y')$.
4. *For all cycles* $v_1v_2 \cdots v_Lv_1$ *of* G_{good}, $\sum_{i=1}^{L} \mathsf{Label}(v_i) = 0$.[5]

Proof. The proofs are straightforward:

1. Clearly, $z' = z$ and $c' = c$ imply that $t' - y' = t - y$, hence $V_{good} \subseteq V$.
2. If $t' - y' = t - y$ and $y' = y$, then $t' = t$. If we further have $z' = z$, then we deduce $c' = c$. If $c' = c$, then $x' = x$, thus $xy = x'y'$. Hence, we cannot have $(xy, x'y') \in V$.
3. If $c' = c$ then $F_0(y') - F_0(y) = x - x' = \mathsf{Label}(xy, x'y')$.

[5] Note that the cycle length notation L should not be confused with the subscript L indicating the left part of a plaintext or a ciphertext.

4. Let $\nu_i = (x_i y_i, x'_i y'_i)$. If $\nu_i \in V_{\mathsf{good}}$ then $F_0(y'_i) - F_0(y_i) = \mathsf{Label}(\nu_i)$. If we have a cycle then $y'_i = y_{i+1}$ with $y_{L+1} = y_1$. Hence, $\sum_i \mathsf{Label}(\nu_i) = 0$.

□

The principle of our attack is as follows: if we get vertices in V_{good}, the property 3 from Lemma 1 gives equations to characterize F_0. One problem is that we can identify vertices in V, but we cannot tell apart good and non-good (bad) ones. One way to recognize good vertices is to use property 4 in Lemma 1: to find cycles with zero sum of labels. For this, we will prove in Lemma 4 that this is a characteristic property of good cycles, meaning that all the vertices in these cycles are good vertices. First, we estimate the number of vertices and edges with the following two Lemma.

Lemma 2. *For x, y, x', y' random and F_0, F_1, F_2, F_3 random,*

$$\Pr[(xy, x'y') \in V_{\mathsf{good}} \mid (xy, x'y') \in V] = \tfrac{1}{2-\frac{1}{N}} \approx \tfrac{1}{2}.$$

Proof. We compute the following probabilities:

$$\begin{aligned}
\Pr[xy, x'y' \in V_{\mathsf{good}}] &= \Pr[z' = z, c' = c, x'y' \neq xy] \\
&= \Pr[z' = z, c' = c, y' \neq y] \\
&= \Pr[y' \neq y] \Pr[c' = c \mid y' \neq y] \Pr[z' = z \mid c' = c, y' \neq y] \\
&= \left(1 - \frac{1}{N}\right) \frac{1}{N^2}. \qquad (2)
\end{aligned}$$

$$\begin{aligned}
\Pr[xy, x'y' \in V \setminus V_{\mathsf{good}}] &= \Pr[z' = z, t' - y' = t - y, c' \neq c, xy \neq x'y'] \\
&= \Pr[z' = z, d' - y' = d - y, c' \neq c, y' \neq y] \\
&= \Pr[y' \neq y] \Pr[c' \neq c \mid y' \neq y] \\
&\quad \Pr[d' - y' = d - y \mid y' \neq y, c' \neq c] \\
&\quad \Pr[z' = z \mid d' - y' = d - y, y' \neq y, c' \neq c] \\
&= \left(1 - \frac{1}{N}\right)\left(1 - \frac{1}{N}\right)\left(\frac{1}{N}\right)\left(\frac{1}{N}\right).
\end{aligned}$$

Hence,

$$\begin{aligned}
\Pr[xy, x'y' \in V_{\mathsf{good}} \mid xyx'y' \in V] &= \frac{\Pr[xy, x'y' \in V_{\mathsf{good}}]}{\Pr[xy, x'y' \in V]} \\
&= \frac{1}{1 + \frac{\Pr[xy, x'y' \in V \setminus V_{\mathsf{good}}]}{\Pr[xy, x'y' \in V_{\mathsf{good}}]}} \\
&= \frac{1}{2 - \frac{1}{N}} \approx \frac{1}{2}.
\end{aligned}$$

□

Lemma 3. *The expected number of elements in* V_{good} *is* $\frac{M(M-1)\left(1-\frac{1}{N}\right)}{N^2} \approx \frac{M^2}{N^2}$.

Proof. We have $M(M-1)$ possible pair of tuples $xy, x'y'$ with $xy \neq x'y'$ to construct V_{good}. From Eq. (2), the probability of each vertex in V_{good} is $\frac{1}{N^2}\left(1-\frac{1}{N}\right)$. Thus, we expect to have $\frac{M(M-1)\left(1-\frac{1}{N}\right)}{N^2} \approx \frac{M^2}{N^2}$ elements in V_{good}. □

We have the property that for each cycle $v_1 v_2 \cdots v_L v_1 \in G$, if $v_1, \ldots, v_L$ are all in V_{good}, then the sum of $\mathsf{Label}(v_i)$ is zero due to Lemma 1, property 4. If one vertex is not good, the sum may be random. This suggests a way to find good vertices in V that is to look for long cycles in G with a zero sum of labels.

Lemma 4. *(*$L = 2$ *case) If* $v_1 = (x_1 y_1, x_1' y_1')$ *we say that* v_1 *and* v_2 *are permuting if* $v_2 = (x_1' y_1', x_1 y_1)$. *If* $v_1 v_2 v_1$ *is a cycle in* G *with zero sum of labels, and* v_1, v_2 *are not permuting, then* v_1 *and* v_2 *are likely to be good. More precisely, for* $v_1 = (x_1 y_1 x_1' y_1')$ *and* $v_2 = (x_2 y_2 x_2' y_2')$ *random, we have* $\Pr[v_1, v_2 \in V_{good} \mid v_1 v_2 v_1$ *is a cycle,* v_1, v_2 *not permuting,* $\sum_{i=1}^{2} \mathsf{Label}(v_i) = 0] \geqslant \frac{1}{1+\frac{10}{N-5}}$.

The proof for Lemma 4 is in Appendix A.1. We believe that Lemma 4 remains true for valid cycles of small length except in trivial cases. In Appendix A.2, we extend to $L > 2$ for cycles satisfying some special non-repeating condition [$\neg$repeat] on the c and d values to rule out many trivial cases. However, this condition [$\neg$repeat] cannot be checked by the adversary. Instead, we could just avoid repetitions of any message throughout the cycle (as repeating messages induce repeating c's or d's). We use the following conjecture (which is supported by experiment for $L = 3$).

Conjecture 1 *If* $v_1 v_2 \cdots v_L v_1$ *is a cycle of length* L *in* G *with zero sum of labels and the vertices use no messages in common, then* $v_1 \cdots v_L$ *are all good with probability close to 1.*

For M known plaintexts, the expected number of valid cycles in G_{good} of a given length L is $\frac{M^{2L}}{N^{3L}}$.

The aim of our attack is to collect as many F_0 outputs as possible to reconstruct a table of this function. Thus, we are interested in vertices whose labels are defined as $\mathsf{Label}(v_i) = F_0(y) - F_0(y'), \forall i \in \{0, 1, \ldots, |V|\}$ and we generate another graph to represent the collection of many independent equations for F_0.

We have a valid cycle $v_1 v_2 \cdots v_L v_1$ of length L in G when $v_i \in V$,

$$\sum_{i=1}^{L} \mathsf{Label}(v_i) = 0$$

and vertices use no messages in common. Now, let us define an undirected graph $G' = (V', E')$, where $V' = \{0, 1, \ldots, N-1\}$ and E' is defined as follows: for each vertex $v_i = (xy, x'y')$ in a valid cycle $v_1 v_2 \cdots v_L v_1$ of length L, add $\{y_i, y_i'\}$ as an edge in E' with label set to $\mathsf{Label}(v_i)$. The purpose of such a graph G' is to put y values which are dependent on each other in a single connected component and put apart with independent y values in separate connected components.

When we model G' as a random graph, we can adjust M so that we can have a large connected component in G'. Given the vertex set size $|V'| = N$ and the edge size $|E'| = m$, $m = \frac{N(N-1)}{2}p$, where p is the probability that G' has an edge between two vertices. From Erdős-Rényi model [12] on random graphs, we want $Np \geqslant 1$. We know that $Np \sim 2\frac{m}{N}$. So, we want $m \geqslant \frac{N}{2}$. We have $\frac{M^{2L}}{L \cdot N^{3L}}$ expected good cycles (counted without repetition of their L circular rotations) of length L, thus $m \sim \frac{M^{2L}}{N^{3L}}$. Therefore, we need to set $M = \lambda N^{\frac{3}{2}} \left(\frac{N}{2}\right)^{\frac{1}{2L}}$ for a constant $\lambda \geqslant 1$ to have a large connected component in G'. Our attack works with $M = N^{\frac{3}{2}+\epsilon}$ for $\epsilon > 0$ small, with complexity $O(2^L N^{(1+2\epsilon)L})$ and a constant probability of success. If our attack recovers at least $\sqrt{N}$ points in F_0 correctly (which is the case when we have a large connected component in G'), we obtain $M \times \frac{\sqrt{N}}{N} \gg N$ samples to apply the attack on 3-rounds so that it recovers a good fraction of F_1, F_2, F_3. It is enough to bootstrap a yoyo attack (Steps 9–18 of Algorithm 3). And, our attack succeeds.

Now, we give the full algorithm of our attack to 4-round Feistel scheme.

Algorithm 3: (F_0, F_1, F_2, F_3) Recovery Attack (Strategy S_2)

```
1  Pick $M$ known plaintexts and retrieve their ciphertext.
2  Create $G = (V, E)$.
3  Find valid cycles of length $2, 3, \ldots, L$ and collect the vertices in these
     cycles.
4  Create $G'$ from $\{y, y'\}$ from the collected vertices.
5  Find the largest connected component in $G'$.
6  Assign one $F_0(y)$ value arbitrarily and deduce $F_0$ on the connected
     component.
7  For all known plaintexts using $y$ in the connected component, evaluate
     and deduce a tuple for the 3-round Feistel scheme based on $(F_1, F_2, F_3)$.
8  Apply the attack on 3-round Feistel scheme from Section 4.1 to recover a
     constant fraction of $(F_1, F_2, F_3)$.
9  while nothing more revealed do
10     foreach of the M plaintext/ciphertext pairs do
11         if $F_0$ and $F_1$ are known for this plaintext then
12             deduce one point for $F_2$ and $F_3$
13         end
14         if $F_2$ and $F_3$ are known for this ciphertext then
15             deduce one point for $F_0$ and $F_1$
16         end
17     end
18 end
```

Experimentally, we noticed that $\lambda = 0.8$ is too small to obtain a large enough connected component for $L = 3$. Conversely, for $\lambda = 2$, G' is more connected but the giant component contains many bad edges that we want to avoid.

Let E_j be the event that the sizes of the j largest connected components sum to greater than $\sqrt{N}$ with no bad edges in G'. Let $E_{\leqslant j}$ be the event that either of $E_1, E_2, \ldots, E_j$ occurs. We simulated the attack for various N values and $\lambda = 1, 2, 3$ and report the numbers for $E_{\leqslant 1}, E_{\leqslant 2}, E_{\leqslant 3}$ on Table 2. When we read the table, by taking $\lambda = 1$ and $j = 3$, our attack recovers $\sqrt{N}$ points of F_0 with probability at least 23 %. In our attack, if we look at j connected components, we need to multiply the complexity by N^{j-1} (We can fix F_0 on one point for free, then all values in its connected components are inferred, but for each additional connected component, we must guess one value of F_0). It is likely that we can mitigate this N^{j-1} factor by early abort during the attack on 3-rounds.

In our experiments, we observe better success probability of our attack with $\lambda = 1$. With λ larger, the attack hardly ever succeeds. It may look paradoxical to say that if λ is too large, then the attack fails, but this is due to higher chances to collect bad edges. However, when G' is heavily connected, we could propose algorithms to eliminate inconsistencies in labels and get rid of bad edges. It means that we would have a successful attack for any $\lambda \geqslant 2$. We let it as future work.

Therefore, we have a double phase transition. The first phase transition occurs when we have enough data to be able to make the graph and find cycles. Our attack quickly succeeds after this phase transition. The second phase transition occurs when we start having bad edges in the collected cycles. Then, our attack must be enriched to be able to work any longer. We did not do it on purpose as we noticed there is a sufficient window in between these two phase transitions to break the scheme with good probability of success and without caring about possible bad edges.

In Table 3, we show the experimental results of success probability of the entire attack for various strategies. Let S_j be an event with strategy j. In S_1, we accumulate the three largest connected components and abort unless the accumulated size is at least $\sqrt{N}$ and they have no bad edges. I.e., S_1 is exactly $E_{\leqslant 3}$. In S_2, we just look at the largest connected component and fail unless it has no bad edges in G' (we remove the condition on size of the connected component that is greater than $\sqrt{N}$). In S_3 (and S_4 resp.), we look at the two largest (three largest resp.) connected components that have no bad edges. What we report in Table 3 includes the success probability $\Pr_{\mathsf{succ}}$ of S_i and we recover the entire tables for each round function. These various strategies considered for experimental purpose even though we have the theory results that suggests to condition on the size of the connected component.

The data complexity of our attack in Algorithm 3 is $M = O(N^{\frac{3}{2}+\frac{1}{2L}})$. We compute the time complexity for the algorithm based on the step 2, 3, 4, and 5, since the other steps are much shorter. In step 2, creating our graph G is defined as forming the vertices in G. This can be done in $M \log(M)$ time with collision detection for M known plaintext/ciphertext pairs. In step 3, we look

N	$\mathsf{M}(\lambda)$	#trials	$\Pr[\mathsf{E}_{\leqslant 1}]$	$\Pr[\mathsf{E}_{\leqslant 2}]$	$\Pr[\mathsf{E}_{\leqslant 3}]$
2	2(0.71)	5022	0.00 %	0.00 %	0.00 %
4	5(0.56)	7098	1.51 %	1.51 %	1.51 %
8	15(0.53)	7010	0.36 %	4.07 %	4.07 %
16	46(0.51)	6665	0.05 %	1.23 %	1.23 %
32	144(0.50)	6103	0.02 %	0.03 %	0.16 %
64	457(0.50)	7986	0.00 %	0.00 %	0.01 %
128	1449(0.50)	7460	0.00 %	0.00 %	0.00 %
256	4598(0.50)	6879	0.00 %	0.00 %	0.00 %
512	14597(0.50)	6094	0.00 %	0.00 %	0.00 %
2	3(1.06)	4316	0.00 %	0.00 %	0.00 %
4	8(0.89)	4153	15.19 %	15.19 %	15.19 %
8	23(0.81)	6703	5.83 %	18.54 %	18.54 %
16	73(0.81)	6886	4.57 %	13.87 %	13.87 %
32	230(0.80)	6952	2.52 %	7.12 %	10.98 %
64	730(0.80)	6568	1.40 %	5.65 %	9.18 %
128	2318(0.80)	6189	0.29 %	1.13 %	2.83 %
256	7357(0.80)	8054	0.02 %	0.30 %	0.86 %
512	23355(0.80)	752	0.00 %	0.00 %	0.00 %
2	3(1.06)	4352	0.00 %	0.00 %	0.00 %
4	9(1.00)	3864	23.08 %	23.08 %	23.08 %
8	29(1.02)	5791	15.59 %	35.02 %	35.02 %
16	91(1.01)	6585	16.20 %	29.90 %	29.90 %
32	288(1.00)	6814	14.66 %	27.09 %	31.67 %
64	913(1.00)	6981	18.16 %	34.69 %	40.87 %
128	2897(1.00)	6609	16.31 %	33.53 %	40.73 %
256	9196(1.00)	6176	16.27 %	36.90 %	46.50 %
512	29193(1.00)	544	12.87 %	33.64 %	45.40 %
8	58(2.03)	988	22.77 %	23.99 %	23.99 %
16	182(2.01)	2504	6.71 %	6.79 %	6.79 %
32	575(2.00)	3425	0.53 %	0.55 %	0.55 %
64	1825(2.00)	5727	0.02 %	0.02 %	0.02 %
128	5793(2.00)	2197	0.00 %	0.00 %	0.00 %
256	18391(2.00)	139	0.00 %	0.00 %	0.00 %
512	58386(2.00)	8	0.00 %	0.00 %	0.00 %
32	863(3.00)	1389	0.00 %	0.00 %	0.00 %
64	2737(3.00)	2786	0.00 %	0.00 %	0.00 %
128	8689(3.00)	188	0.00 %	0.00 %	0.00 %
256	27586(3.00)	10	0.00 %	0.00 %	0.00 %

Table 2. Experimental $\Pr[\mathsf{E}_{\leqslant \mathsf{j}}]$ over several trials for various N, λ, and j; the number of trials correspond to the successful runs of the whole attack on FF3 in the first step out of 10 000 using $\mathsf{L} = 3$.

N	$M(\lambda)$	#trials	Pr[succ, S_1]–(Pr[S_1])	Pr[succ, S_2]–(Pr[S_2])	Pr[succ, S_3]–(Pr[S_3])	Pr[succ, S_4]–(Pr[S_4])
2	2(0.71)	5022	0.00 %–(0.00 %)	0.00 %–(100.00 %)	0.00 %–(49.70 %)	0.00 %–(49.70 %)
4	5(0.56)	7098	0.00 %–(1.51 %)	0.00 %–(99.42 %)	0.00 %–(36.97 %)	0.00 %–(36.97 %)
8	15(0.53)	7010	0.00 %–(4.07 %)	0.00 %–(98.49 %)	0.00 %–(36.01 %)	0.00 %–(36.01 %)
16	46(0.51)	6665	0.00 %–(1.23 %)	0.00 %–(97.99 %)	0.00 %–(38.86 %)	0.00 %–(38.84 %)
32	144(0.50)	6103	0.05 %–(0.16 %)	0.77 %–(98.33 %)	2.24 %–(45.55 %)	2.24 %–(45.53 %)
64	457(0.50)	7986	0.01 %–(0.01 %)	2.02 %–(98.32 %)	6.36 %–(53.72 %)	6.41 %–(53.72 %)
128	1449(0.50)	7460	0.00 %–(0.00 %)	2.01 %–(98.75 %)	7.02 %–(67.63 %)	7.67 %–(67.57 %)
256	4598(0.50)	6879	0.00 %–(0.00 %)	0.74 %–(98.92 %)	5.16 %–(80.23 %)	6.67 %–(80.20 %)
512	14597(0.50)	6094	0.00 %–(0.00 %)	0.31 %–(99.38 %)	3.02 %–(92.25 %)	5.07 %–(92.16 %)
2	3(1.06)	4316	0.00 %–(0.00 %)	0.00 %–(100.00 %)	0.00 %–(76.90 %)	0.00 %–(76.90 %)
4	8(0.89)	4153	0.07 %–(15.19 %)	0.07 %–(93.74 %)	1.13 %–(59.64 %)	1.13 %–(59.64 %)
8	23(0.81)	6703	3.88 %–(18.54 %)	2.27 %–(90.23 %)	4.83 %–(57.72 %)	4.85 %–(57.69 %)
16	73(0.81)	6886	10.30 %–(13.87 %)	21.71 %–(87.71 %)	29.65 %–(67.25 %)	29.67 %–(67.14 %)
32	230(0.80)	6952	10.34 %–(10.98 %)	43.18 %–(88.62 %)	57.44 %–(79.67 %)	57.44 %–(78.88 %)
64	730(0.80)	6568	8.82 %–(9.18 %)	59.10 %–(91.21 %)	75.29 %–(88.78 %)	75.21 %–(87.62 %)
128	2318(0.80)	6189	2.70 %–(2.83 %)	65.89 %–(93.89 %)	84.15 %–(93.75 %)	84.15 %–(92.39 %)
256	7357(0.80)	8054	0.84 %–(0.86 %)	67.21 %–(96.52 %)	87.79 %–(96.52 %)	88.33 %–(95.49 %)
512	23355(0.80)	752	0.00 %–(0.00 %)	67.15 %–(98.27 %)	90.82 %–(98.27 %)	91.36 %–(97.87 %)
2	3(1.06)	4352	0.00 %–(0.00 %)	0.00 %–(100.00 %)	0.00 %–(75.30 %)	0.00 %–(75.30 %)
4	9(1.00)	3864	3.03 %–(23.08 %)	3.60 %–(88.69 %)	7.27 %–(64.65 %)	7.27 %–(64.65 %)
8	29(1.02)	5791	27.65 %–(35.02 %)	29.11 %–(78.62 %)	34.31 %–(65.88 %)	34.31 %–(65.76 %)
16	91(1.01)	6585	28.44 %–(29.90 %)	49.83 %–(73.27 %)	54.08 %–(68.37 %)	54.08 %–(67.84 %)
32	288(1.00)	6814	30.69 %–(31.67 %)	62.91 %–(71.79 %)	65.17 %–(70.75 %)	65.10 %–(68.80 %)
64	913(1.00)	6981	39.52 %–(40.87 %)	73.80 %–(77.14 %)	73.24 %–(77.14 %)	72.87 %–(74.03 %)
128	2897(1.00)	6609	39.17 %–(40.73 %)	83.10 %–(83.83 %)	79.77 %–(83.83 %)	79.03 %–(79.89 %)
256	9196(1.00)	6176	45.16 %–(46.50 %)	88.52 %–(88.76 %)	85.80 %–(88.76 %)	85.01 %–(85.82 %)
512	29193(1.00)	544	44.49 %–(45.40 %)	93.57 %–(93.57 %)	91.36 %–(93.57 %)	90.44 %–(91.36 %)
8	58(2.03)	988	23.99 %–(23.99 %)	25.40 %–(25.40 %)	25.40 %–(25.40 %)	25.40 %–(25.40 %)
16	182(2.01)	2504	6.79 %–(6.79 %)	6.79 %–(6.79 %)	6.79 %–(6.79 %)	6.79 %–(6.79 %)
32	575(2.00)	3425	0.55 %–(0.55 %)	0.55 %–(0.55 %)	0.55 %–(0.55 %)	0.55 %–(0.55 %)
64	1825(2.00)	5727	0.02 %–(0.02 %)	0.02 %–(0.02 %)	0.02 %–(0.02 %)	0.02 %–(0.02 %)
128	5793(2.00)	2197	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)
256	18391(2.00)	139	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)
512	58386(2.00)	8	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)
32	863(3.00)	1389	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)
64	2737(3.00)	2786	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)
128	8689(3.00)	188	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)
256	27586(3.00)	10	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)	0.00 %–(0.00 %)

Table 3. Experimental Pr[S_j] and success probability over many trials for various N and j using $L = 3$.

for the cycles of length L. The cycles of length L in our graph can be found with multiplication on adjacency matrix (which is sparse). Matrix multiplication can be done in $\mathsf{O}(|\mathsf{V}|^2\mathsf{d})$ where $\mathsf{d} = \frac{|\mathsf{E}|}{|\mathsf{V}|}$ is the average degree of a vertex. Therefore, the complexity is $\mathsf{O}(|\mathsf{V}||\mathsf{E}|)$. With the Floyd-Warshall algorithm, we need $(\mathsf{L}-1)$ multiplications by the adjacency matrix in the max-plus algebra that leads us to a complexity $\mathsf{O}(\mathsf{L}|\mathsf{V}||\mathsf{E}|)$. With $|\mathsf{E}| \sim \frac{|\mathsf{V}|^2}{\mathsf{N}}$, where $|\mathsf{V}| = 2\frac{\mathsf{M}^2}{\mathsf{N}^2} = 2^{3-\frac{1}{\mathsf{L}}}\mathsf{N}^{1+\frac{1}{\mathsf{L}}}$ and L constant, we have $\mathsf{O}(\frac{|\mathsf{V}|^3}{\mathsf{N}})$ which is equal to $\mathsf{O}(\mathsf{N}^{2+\frac{3}{\mathsf{L}}})$. Another method to find cycles is to enumerate all L-tuples of vertices in $\mathsf{O}(|\mathsf{V}|^{\mathsf{L}})$ which is $\mathsf{O}(\mathsf{N}^{\mathsf{L}+1})$. Therefore, we compute the minimum between the two methods which is $\mathsf{O}(\mathsf{N}^3)$ for any L and it is the complexity of step 3. (It can even be lower for $\mathsf{L} > 3$.) Step 4 takes N time and finally step 5 takes $\frac{\mathsf{M}^{2\mathsf{L}}}{\mathsf{N}^{3\mathsf{L}}} = \frac{\mathsf{N}}{2}$. Since the complexity is weighted by step 3, **we have time complexity of our algorithm as $\mathsf{O}(\mathsf{N}^3)$ for $\mathsf{L} = 3$ and a smaller $\mathsf{O}(\mathsf{N}^{2+\frac{3}{\mathsf{L}}})$ for $\mathsf{L} > 3$.** Instead of $\mathsf{L}-1$ multiplications to a sparse matrix in the max-plus algebra, we could also use $\mathsf{O}(\log \mathsf{L})$ general purpose matrix multiplications over the integer with the Coppersmith-Winograd algorithm [10]. We would reach a complexity of $\mathsf{O}(|\mathsf{V}|^{2.38}\log \mathsf{L})$ which is not better.

4.3 Round-Function-Recovery on 5-Round Feistel Scheme and More

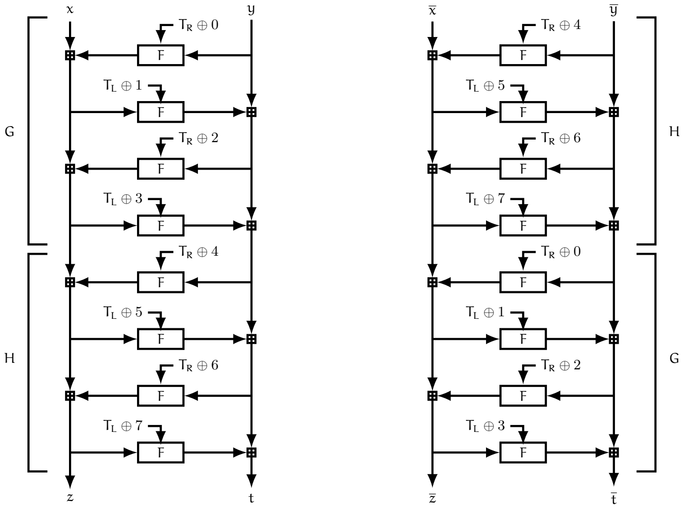
Given the 4-round full recovery attack from Section 4.2, we can extend it to attack 5-round Feistel network. The attack for 5-round Feistel network is straightforward; it uses chosen plaintexts and guess strategies. First of all, consider our 4-round attack and the known plaintexts from this attack. We choose plaintexts for the 5-round so that the right half of the messages have as little different values as possible then guess the corresponding images through F_0. It means that for the right halves of the messages, we generate all the possible partial tables of the first round function for these right values. Then, we guess which table is consistent after running the attack on the next 4-round. The data complexity of our 4-round attack is $\lambda\mathsf{N}^{\frac{3}{2}+\epsilon}$, **hence our time complexity for 5-round recovery with chosen plaintexts is $\mathsf{O}(\mathsf{N}^{\lambda\mathsf{N}^{\frac{1}{2}+\epsilon+3}})$.** The data complexity is unchanged.

We can attack $\mathsf{r}-$rounds similarly with complexity $\mathsf{O}(\mathsf{N}^{(\mathsf{r}-5)\mathsf{N}+\sqrt{\mathsf{N}}+3})$ by guessing the round functions on the last $(\mathsf{r}-5)$ rounds. The data complexity is unchanged. We can apply this to FF1 ($\mathsf{r} = 10$) and FF3 ($\mathsf{r} = 8$). We obtain a complexity lower than 2^{128} for FF1 with $\mathsf{N} = 7$ and for FF3 with $7 \leqslant \mathsf{N} \leqslant 10$. (For lower N, exhaustive search on either the codebook or the round functions reaches the same conclusion.) Hence, **these instances of FF1 and FF3 do not offer a 128-bit security**.

5 Slide Attack on FF3

We develop an attack on 4-round Feistel network in Section 4 and we deploy it as a building block for our chosen-plaintext and chosen-tweak attack to FF3 scheme. Our FF3 attack aims to reconstruct the entire codebook for a challenge tweak for a number of queries which is lower than the size of the brute force codebook attack. The main idea of the designed FF3 attack takes advantage of the flexibility to change the tweak to permute the round functions.

Consider two functions G and H, where G is a 4-round Feistel scheme using tweakable block cipher F with tweaks $(\mathsf{T_R} \oplus \mathsf{STR}_2^{32}(0), \mathsf{T_L} \oplus \mathsf{STR}_2^{32}(1), \mathsf{T_R} \oplus \mathsf{STR}_2^{32}(2), \mathsf{T_L} \oplus \mathsf{STR}_2^{32}(3))$ and H is a 4-round Feistel scheme using tweakable block cipher F with tweaks $(\mathsf{T_R} \oplus \mathsf{STR}_2^{32}(4), \mathsf{T_L} \oplus \mathsf{STR}_2^{32}(5), \mathsf{T_R} \oplus \mathsf{STR}_2^{32}(6), \mathsf{T_L} \oplus \mathsf{STR}_2^{32}(7))$. In Fig. 5, we show two runs of FF3 encryption with tweak $\mathsf{T} = \mathsf{T_L} \| \mathsf{T_R}$ in (a) and tweak $\mathsf{T}' = \mathsf{T_L} \oplus \mathsf{STR}_2^{32}(4) \| \mathsf{T_R} \oplus \mathsf{STR}_2^{32}(4)$ in (b) on two distinct plaintext. We observe that $\mathsf{FF3.E}(\mathsf{K}, \mathsf{T}, \cdot) = \mathsf{H} \circ \mathsf{G}$ and $\mathsf{FF3.E}(\mathsf{K}, \mathsf{T}', \cdot) = \mathsf{G} \circ \mathsf{H}$. For simplicity, we do not explicitly write $\mathsf{STR}_2^{32}(\cdot)$ any longer. Given this permuting ability by setting the tweaks XORed with round functions, we desire to form a "cyclic" behavior of plaintext/ciphertext pairs under two FF3 encryption with sliding G and H.



(a) FF3 Encryption (b) Slided Encryption

Fig. 5. FF3 encryption with sliding round functions

We pick at random two sets of messages $\mathsf{X} = \{\mathsf{xy}_0^1, \ldots, \mathsf{xy}_0^i, \ldots, \mathsf{xy}_0^A\}$ and $\overline{\mathsf{X}} = \{\overline{\mathsf{xy}}_0^1, \ldots, \overline{\mathsf{xy}}_0^i, \ldots, \overline{\mathsf{xy}}_0^A\}$ of size A. For each message xy_0^i in X, set $\mathsf{xy}_{j+1}^i = \mathsf{Enc}(\mathsf{K}, \mathsf{T}, \mathsf{xy}_j^i)$ with a fixed tweak $\mathsf{T} \in \mathcal{T}$ and a fixed key $\mathsf{K} \in \mathcal{K}$. We repeat the chain encryption of outputs B times for each message in X. Let XC be the set of chain encryption of elements of X. It contains segments of length B of cycles of $\mathsf{H} \circ \mathsf{G}$. Similarly, for each message $\overline{\mathsf{xy}}_0^i$ in $\overline{\mathsf{X}}$, set $\overline{\mathsf{xy}}_{j+1}^i = \mathsf{Enc}(\mathsf{K}, \mathsf{T}', \overline{\mathsf{xy}}_j^i)$ with the fixed tweak $\mathsf{T}' \in \mathcal{T}$ under the same key K. Let $\overline{\mathsf{XC}}$ be the set of chain encryption of elements of $\overline{\mathsf{X}}$. Apparently, we have $|\mathsf{XC}| = \mathsf{AB}$ and $|\overline{\mathsf{XC}}| = \mathsf{AB}$. Given these 2 sets XC and $\overline{\mathsf{XC}}$, we attempt to find a collision between XC and $\overline{\mathsf{XC}}$ such that $\mathsf{G}(\mathsf{xy}^{\mathsf{i}}_{\mathsf{j}}) = \overline{\mathsf{xy}}^{\mathsf{i}'}_0$ or $\mathsf{G}(\mathsf{xy}^{\mathsf{i}}_0) = \overline{\mathsf{xy}}^{\mathsf{i}'}_{\mathsf{j}'}$ for $1 \leqslant \mathsf{i}, \mathsf{i}' \leqslant \mathsf{A}$ and $1 \leqslant \mathsf{j}, \mathsf{j}' \leqslant \mathsf{B}$. (See Fig. 6.) Upon having a table with inputs to G and H, we can apply the known-plaintext recovery attack on 4-round Feistel networks. The concrete algorithm to collect plaintext/ciphertext pairs is given in Algorithm 4.

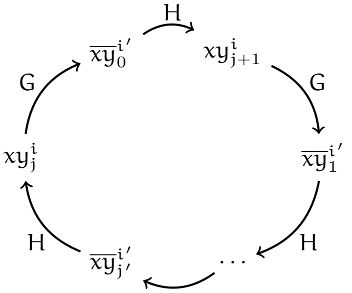

H
$\overline{\mathsf{xy}}^{\mathsf{i}'}_0$
$\mathsf{xy}^{\mathsf{i}}_{\mathsf{j}+1}$
G
G
$\mathsf{xy}^{\mathsf{i}}_{\mathsf{j}}$
$\overline{\mathsf{xy}}^{\mathsf{i}'}_1$
H
H
$\overline{\mathsf{xy}}^{\mathsf{i}'}_{\mathsf{j}'}$
...

Fig. 6. Circular behavior of plaintext/ciphertext pairs.

We, now, formally prove useful results for the analysis and success probability of the attack in Algorithm 4.

Let Π be a random permutation on $\{0, \ldots, \mathsf{N}^2 - 1\}$. Let c_k be the number of cycles of length k in Π. The total number of elements in a cycle of length k (for all k) is equal to N^2, meaning that $\sum_{\mathsf{k}=1}^{\mathsf{N}^2}(\mathsf{k}\mathsf{c}_\mathsf{k}) = \mathsf{N}^2$. It is well-known that the expected number of cycles of length k over a random Π is $\mathbb{E}_\Pi(\mathsf{c}_\mathsf{k}) = \frac{1}{\mathsf{k}}$. [6]

In what follows we show two useful results.

Lemma 5. *For a message* xy^i *picked at random, let* $\mathsf{length}(\mathsf{xy}^\mathsf{i})$ *be the length of the cycle that contains* xy^i. *For two messages* xy^i *and* $\overline{\mathsf{xy}}^{\mathsf{i}'}$ *picked at random, let* E_0 *be an event that* xy^i *and* $\overline{\mathsf{xy}}^{\mathsf{i}'}$ *are in the same cycle. The expected value of* $\mathsf{length}(\mathsf{xy}^\mathsf{i})$ *is* $\mathbb{E}_{\mathsf{xy}^\mathsf{i},\Pi}[\mathsf{length}(\mathsf{xy}^\mathsf{i})] = \frac{\mathsf{N}^2+1}{2}$ *and the expected value of* $\mathsf{length}(\mathsf{xy}^\mathsf{i})$ *given* E_0 *is* $\mathbb{E}[\mathsf{length}(\mathsf{xy}^\mathsf{i})|\mathsf{E}_0] = \frac{2\mathsf{N}^2+1}{3}$.

Proof. We use the same notation for c_k as above.

$$\mathbb{E}_{\mathsf{xy}^\mathsf{i},\Pi}[\mathsf{length}(\mathsf{xy})] = \mathbb{E}_{\mathsf{xy}^\mathsf{i},\Pi}\left[\sum_{\mathsf{k}=1}^{\mathsf{N}^2} \mathsf{k}\mathsf{c}_\mathsf{k}\frac{\mathsf{k}}{\mathsf{N}^2}\right] = \sum_{\mathsf{k}=1}^{\mathsf{N}^2} \mathbb{E}[\mathsf{c}_\mathsf{k}]\frac{\mathsf{k}^2}{\mathsf{N}^2} = \sum_{\mathsf{k}=1}^{\mathsf{N}^2}\frac{\mathsf{k}}{\mathsf{N}^2} = \frac{\mathsf{N}^2+1}{2}$$

[6] The probability that a given point is in a cycle of length exactly k is $\frac{(\mathsf{N}^2-1)\cdots(\mathsf{N}^2-\mathsf{k}+1)}{\mathsf{N}^2(\mathsf{N}^2-1)\mathsf{cldots}(\mathsf{N}^2-\mathsf{k}+1)} = \frac{1}{\mathsf{N}^2}$. Hence, the expected number of points in a cycle of length k is $1 = \mathbb{E}_\Pi(\mathsf{k}\mathsf{c}_\mathsf{k})$.

Algorithm 4: FF3 Attack

```
Input : a tweak bit string $T$ such that $|T| = 64$, a key $K$
$T_L||T_R \leftarrow T$
$T' \leftarrow T_L \oplus 4||T_R \oplus 4$
foreach $i = 1 \cdots A$ do
    pick $xy_0^i$ and $\overline{xy}_0^i$
    foreach $j = 1 \cdots B$ do
        $xy_j^i = \mathsf{FF3.E}(K, T, xy_{j-1}^i)$
        $\overline{xy}_j^i = \mathsf{FF3.E}(K, T', \overline{xy}_{j-1}^i)$
    end
end
foreach $i, i' = 1 \cdots A$ do
    foreach $j = 0 \cdots B - M - 1$ do
        // assume that $G(xy_j^i) = \overline{xy}_0^{i'}$
        run attack on $G$ with samples $G(xy_{j+k}^i) = \overline{xy}_k^{i'}$ for $k = 0 \cdots B - j$
        if succeeded, run attack on $H$ with samples $H(G(xy_k^i)) = xy_{k+1}^i$ for
          $k = 0 \cdots B - 1$
    end
    foreach $j = 0 \cdots B - M - 1$ do
        // assume that $G(xy_0^i) = \overline{xy}_{j'}^{i'}$
        run attack on $G$ with samples $G(xy_k^i) = \overline{xy}_{j+k}^{i'}$ for $k = 0 \cdots B - j$
        if succeeded, run attack on $H$ with samples $H(G(xy_k^i)) = xy_{k+1}^i$ for
          $k = 0 \cdots B - 1$
    end
end
```

We first, observe that for any messages xy^i and $\overline{xy}^{i'}$, being in the same cycle of every possible length occurs with probability $\frac{1}{2}$. Then,

$$\Pr[E_0] = \mathbb{E}_\Pi \left[\sum_{k=1}^{N^2} c_k \left(\frac{k}{N^2} \right)^2 \right] = \sum_{k=1}^{N^2} \frac{k}{N^4} = \frac{1}{2} + \frac{1}{2N^2} \approx \frac{1}{2}$$

$$\begin{aligned} \mathbb{E}[\mathsf{length}(xy^i)|E_0] &= \mathbb{E}_\Pi \left[\sum_{k=1}^{N^2} kc_k \left(\frac{k^2}{N^4} \right) \frac{1}{\Pr(E_0)} \right] = \frac{\sum_{k=1}^{N^2} \frac{k^2}{N^4}}{\Pr(E_0)} \\ &= \frac{2N^2}{N^2+1} \times \frac{(N^2+1)(2N^2+1)}{6N^2} = \frac{2N^2+1}{3} \end{aligned}$$

□

This means that if we pick xy^i and $\overline{xy}^{i'}$ at random and let $xy^j = G^{-1}(\overline{xy}^{i'})$ then xy^i and $\overline{xy}^{i'}$ are in the same cycle with probability close to $\frac{1}{2}$ and we will observe Fig. 6. One problem is that the cycle is typically long, i.e. $\frac{2N^2}{3}$ as shown in Lemma 5, but we want that two segments of length B starting from xy^i and $\overline{xy}^{i'}$ intersect on at least M points. Therefore, we need the probability of two segments overlapping in a cycle of length k on at least M points.

Lemma 6. *Let two segments* $xy^i - \Pi(xy^i) - \Pi^2(xy^i) - \cdots - \Pi^B(xy^i)$ *and* $\overline{xy}^{i'} - \Pi(\overline{xy}^{i'}) - \Pi^2(\overline{xy}^{i'}) - \cdots - \Pi^B(\overline{xy}^{i'})$ *overlap in a given cycle of length* k *on at least* M *points be the event* E_1^k*. Let* E_1 *be the union of all* E_1^k *for every possible length of* k*. The probability that* E_1 *occurs is equivalent to* $\frac{2(B-M)}{N^2}$ *for* $M = o(N^2)$*.*

Proof. We use the same notation for c_k as above.

$$\Pr[E_1] = \mathbb{E}_\Pi\left[\sum_{k=M}^{N^2} c_k \Pr[E_1^k]\right] = \mathbb{E}_\Pi\left[\sum_{k=M}^{N^2} c_k \frac{k}{N^2}\frac{\min\{k, 2(B-M)+1\}}{N^2}\right]$$
$$\sim \frac{2(B-M)}{N^2} \text{ for } M = o(N^2)$$

□

The probability of success of our FF3 attack depends on $\Pr[E_1]$ and on the success probability of our 4-round recovery attack on Feistel network. More clearly,

$$p_{success} = \left(1 - (1-\Pr[E_1])^{A^2}\right) p_{success}^{Feistel}$$

which is equivalent to $\left(1 - e^{\frac{-2(B-M)A^2}{N^2}}\right) p_{success}^{Feistel}$. Thus, we need $A^2(B-M) \approx N^2$ to obtain a constant $p_{success}$. We can neglect the cost of the attack on H as we have plenty of samples and we only run it once G is recovered.

Our attack has $2AB$ data complexity. The time complexity is A^2B times the complexity of 4-round recovery attack on Feistel network. To minimize the data complexity $2AB$ with $A^2(B-M) = N^2$ and $B \geqslant M$, we set $B = 2M$, then $A = \frac{N}{\sqrt{M}}$. Therefore, **we have data complexity of FF3 attack as $4N\sqrt{M}$ and time complexity as $2N^2$ times the complexity of 4-round recovery attack on Feistel network and $p_{success} \approx 1 - e^{-p_{success}^{Feistel}}$.**

We fully implemented the attack but to test its success probability we could skip some parts of the running time we knew the attack would fail. Namely, in Algorithm 4 we can identify directly which segments overlap (using the key) and proceed directly to the 4-round Feistel attack on the right pair of segments. We show on Table 4 the experimental probability of success of the whole attack following the strategies S_j, $j = 1, \ldots, 4$. The probability was computed for 10,000 executions. [7] We also took the executions collecting less than M samples, as long

[7] Executions of the attack on the 4-round Feistel scheme which we used to fill our previous tables are precisely those getting the M samples in this experiment. For some rows with M too large, no experiments collected M pairwise different messages so they are not reported in the previous table. Nevertheless, our attack may still work even though we collect less than M samples. This is why they appear on Table 4.

as they succeed to recover all tables. Curiously, the $\mathsf{N} \leqslant 4$ and $\lambda = 1$ cases seem to take M too low to be able to find cycles. As we can see, the success probability is pretty good (18%–77% for $8 \leqslant \mathsf{N} \leqslant 512$) for $\lambda = 1$ and the strategy S_2 collecting the largest connected components in G'.

We conclude that the full attack succeeds with good probability.

6 Repairing FF3

As a quick fix, we can propose to change the length of the tweak in FF3 so that the adversary has no longer control on what is XORed to the round index. The same should hold if some other part of the tweak is XORed to a counter in a CBC mode, as proposed by the authors of the construction [8]. We obtain a scheme with a shorter tweak, to which we concatenate the round index instead of XORing it.

The original Luby-Rackoff results [15] was extended following this idea by Black and Rogaway [7], but the obtained security result is quite weak as we can only prove that for a number of queries $\mathsf{q} \ll \sqrt{\mathsf{N}}$, the cipher resists to chosen-plaintext attacks, even with only three rounds. By similarly extending the results by Patarin [17], we can obtain that for $\mathsf{q} \ll \mathsf{N}$, the cipher resists to chosen-plaintext and ciphertext attacks, even with only six rounds. However, this says nothing in the case $\mathsf{q} \sim \mathsf{N}^{\frac{3}{2}}$ which is the case of our 4-round attack.[8]

7 Conclusion

We took the NIST standard FF3 and investigated its security on small domain sizes. We started exploiting that we can permute the round functions due to a bad domain separation in the tweak scheme which uses an XOR with the round index. This permutation leads us to develop a slide attack on FF3 based on our own design for 4-round Feistel schemes attack that works with known plaintexts/ciphertexts. Our FF3 attack works with chosen plaintexts and two tweaks. It improves the recent results from Bellare et. al. [3] on data and time complexity to break FF3. Our 4-round Feistel network attack is a full round-function-recovery attack that works with known plaintexts instead of chosen plaintexts and ciphertexts unlike the recent results from Biryukov et. al. [6].

Acknowledgments. The work was done while the first author was visiting EPFL. It was supported by NSF grant CNS-1453132. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center, Pacific (SSC Pacific) under contract No. N66001-15-C-4070.

We thank Adi Shamir for the useful comments and Stefano Tessaro for the discussions.

[8] In reaction to this attack, NIST released the following announcement: `https://beta.csrc.nist.gov/News/2017/Recent-Cryptanalysis-of-FF3`.

N	M	λ	A	B	#run	$\Pr[\mathsf{succ}, S_1]$	$\Pr[\mathsf{succ}, S_2]$	$\Pr[\mathsf{succ}, S_3]$	$\Pr[\mathsf{succ}, S_4]$
2	2	0.71	1	4	10000	0.00 %	0.00 %	0.00 %	0.00 %
4	5	0.56	2	10	10000	0.00 %	0.00 %	0.00 %	0.00 %
8	15	0.53	2	30	10000	0.00 %	0.00 %	0.00 %	0.00 %
16	46	0.51	2	92	10000	0.00 %	0.00 %	0.00 %	0.00 %
32	144	0.50	2	288	10000	0.03 %	0.47 %	1.38 %	1.38 %
64	457	0.50	3	914	10000	0.01 %	1.61 %	5.08 %	5.12 %
128	1449	0.50	3	2898	10000	0.00 %	1.51 %	5.25 %	5.73 %
256	4598	0.50	3	9196	10000	0.00 %	0.52 %	3.55 %	4.59 %
512	14597	0.50	3	29194	9996	0.00 %	0.19 %	1.85 %	3.10 %
2	3	1.06	1	6	10000	0.00 %	0.00 %	0.00 %	0.00 %
4	8	0.89	1	16	10000	0.03 %	0.03 %	0.48 %	0.48 %
8	23	0.81	2	46	10000	2.64 %	1.54 %	3.29 %	3.30 %
16	73	0.81	2	146	10000	7.32 %	15.34 %	21.04 %	21.05 %
32	230	0.80	2	460	10000	7.38 %	30.84 %	41.19 %	41.19 %
64	730	0.80	2	1460	10000	5.90 %	39.58 %	50.78 %	50.73 %
128	2318	0.80	2	4636	10000	1.69 %	41.36 %	53.14 %	53.16 %
256	7357	0.80	3	14714	10016	0.68 %	54.52 %	71.68 %	72.12 %
512	23355	0.80	3	46710	1006	0.00 %	50.40 %	68.59 %	69.18 %
2	3	1.06	1	6	10000	0.00 %	0.00 %	0.00 %	0.00 %
4	9	1.00	1	18	10000	1.18 %	1.40 %	2.84 %	2.84 %
8	29	1.02	2	58	10000	17.24 %	17.99 %	21.46 %	21.46 %
16	91	1.01	2	182	10000	20.15 %	35.35 %	38.85 %	38.85 %
32	288	1.00	2	576	10000	22.01 %	45.89 %	48.29 %	48.24 %
64	913	1.00	2	1826	10000	28.20 %	54.14 %	54.41 %	54.15 %
128	2897	1.00	2	5794	10000	26.24 %	56.85 %	55.14 %	54.65 %
256	9196	1.00	2	18392	10000	28.09 %	55.87 %	54.64 %	54.14 %
512	29193	1.00	3	58386	682	35.48 %	76.10 %	74.93 %	74.05 %
2	6	2.12	1	12	10000	12.20 %	12.20 %	12.20 %	12.20 %
4	18	2.00	1	36	10000	14.15 %	15.62 %	16.48 %	16.48 %
8	58	2.03	1	116	10000	12.96 %	13.92 %	14.40 %	14.40 %
16	182	2.01	1	364	10000	6.10 %	7.37 %	7.65 %	7.65 %
32	575	2.00	1	1150	10000	2.20 %	3.62 %	3.80 %	3.80 %
64	1825	2.00	2	3650	10000	2.80 %	5.59 %	6.34 %	6.32 %
128	5793	2.00	2	11586	3384	2.28 %	4.14 %	4.37 %	4.34 %
256	18391	2.00	2	36782	210	1.43 %	4.29 %	4.29 %	4.29 %
512	58386	2.00	2	116772	13	7.69 %	7.69 %	7.69 %	7.69 %
2	9	3.18	1	18	10000	12.38 %	12.38 %	12.38 %	12.38 %
4	27	3.01	1	54	10000	13.92 %	15.62 %	16.46 %	16.46 %
8	86	3.02	1	172	10000	12.79 %	13.95 %	14.31 %	14.31 %
16	272	3.01	1	544	10000	5.13 %	6.56 %	6.91 %	6.91 %
32	863	3.00	1	1726	10000	2.04 %	3.25 %	3.47 %	3.46 %
64	2737	3.00	1	5474	10000	1.23 %	2.18 %	2.38 %	2.39 %
128	8689	3.00	1	17378	508	0.20 %	0.98 %	1.38 %	1.38 %
256	27586	3.00	2	55172	12	0.00 %	0.00 %	0.00 %	0.00 %
512	87579	3.00	2	175158	2	0.00 %	0.00 %	0.00 %	0.00 %

Table 4. Experimental probability of success in the FF3 attack for various parameters using strategy S_j

References

1. *Recommendation for Block Cipher Modes of Operation: Methods for Format Preserving Encryption.* National Institute of Standards and Technology, 2016.
2. Ross Anderson and Eli Biham. Two practical and provably secure block ciphers: Bear and lion. In Dieter Gollmann, editor, *Fast Software Encryption: Third International Workshop Cambridge, UK, February 21–23 1996 Proceedings*, volume 1029, pages 113–120, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
3. Mihir Bellare, Viet Tung Hoang, and Stefano Tessaro. Message-recovery attacks on Feistel-based Format Preserving Encryption. In *23th CCS Proceedings*, 2016.
4. Mihir Bellare, Thomas Ristenpart, Phillip Rogaway, and Till Stegers. Format-preserving encryption. In Michael J. Jacobson, Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography: 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers*, volume 5867, pages 295–312. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
5. Mihir Bellare, Phillip Rogaway, and Terence Spies. The FFX mode of operation for format-preserving encryption. Draft 1.1. Submission to NIST, Feb. 2010. `http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ffx/ffx-spec.pdf`.
6. Alex Biryukov, Gaëtan Leurent, and Léo Perrin. Cryptanalysis of feistel networks with secret round functions. In Orr Dunkelman and Liam Keliher, editors, *Selected Areas in Cryptography - SAC 2015: 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers*, volume 9566, pages 102–121. Springer International Publishing, 2016.
7. John Black and Phillip Rogaway. Ciphers with arbitrary finite domains. In Bart Preneel, editor, *Topics in Cryptology — CT-RSA 2002: The Cryptographers' Track at the RSA Conference 2002 San Jose, CA, USA, February 18–22, 2002 Proceedings*, volume 2271, pages 114–130, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
8. Eric Brier, Thomas Peyrin, and Jacques Stern. BPS: a Format-Preserving Encryption Proposal. `http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/bps/bps-spec.pdf`.
9. Michael Brightwell and Harry E. Smith. Using Datatype-Preserving Encryption To Enchance Data Warehouse Security. Available at: `http://csrc.nist.gov/nissc/1997/proceedings/141.pdf`, 1997.
10. Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251 – 280, 1990.
11. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. New attacks on feistel structures with improved memory complexities. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215, pages 433–454, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
12. Paul Erdős and Alfred Renyi. *On Random Graphs I*, pages 290–297. Publicationes Mathematicae, 1959.
13. David Goldenberg, Susan Hohenberger, Moses Liskov, Elizabeth Crump Schwartz, and Hakan Seyalioglu. On tweaking luby-rackoff blockciphers. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007: 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007. Proceedings*, volume 4833, pages 342–356, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

14. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. *Journal of Cryptology*, 24(3):588–613, 2011.
15. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, April 1988.
16. Stefan Lucks. Faster luby-rackoff ciphers. In Dieter Gollmann, editor, *Fast Software Encryption: Third International Workshop Cambridge, UK, February 21–23 1996 Proceedings*, volume 1039, pages 189–203, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
17. Jacques Patarin. Security of balanced and unbalanced feistel schemes with linear non equalities. http://eprint.iacr.org/2010/293, 2010.
18. Phillip Rogaway. A Synopsis of Format Preserving Encryption. `http://web.cs.ucdavis.edu/~rogaway/papers/synopsis.pdf`.
19. A. I. Saltykov. The number of components in a random bipartite graph. *Discrete Mathematics Applications*, 5:515–523, 1995.
20. Bruce Schneier and John Kelsey. Unbalanced feistel networks and block cipher design. In *Proceedings of the Third International Workshop on Fast Software Encryption*, volume 1039, pages 121–144, London, UK, 1996. Springer-Verlag.
21. Terence Spies. Format preserving encryption. Unpublished white paper, available at: `https://www.voltage.com/wp-content/uploads/Voltage-Security-WhitePaper-Format-Preserving-Encryption.pdf`, 2008.

A Deferred Proofs

A.1 Proof of Lemma 4

Proof. Before we start computations, we let the followings:
[good]: the event that v_1 and v_2 are both in V_{good}.
[bad]: the event that v_1 and v_2 are both in V but not both in V_{good}.
[cyc]: the event that $y'_1 = y_2$ and $y'_2 = y_1$.
[perm]: the event that $x_1 y_1 = x'_2 y'_2$ and $x'_1 y'_1 = x_2 y_2$.
[$\Sigma = 0$]: the event that $\mathsf{Label}(v_1) + \mathsf{Label}(v_2) = 0$.
[$\#\{d\} = 4$]: the event that d_1, d'_1, d_2, d'_2 are pairwise different.
[$\#\{d\} = j$]: the event that there are exactly j pairwise different values among d_1, d'_1, d_2, d'_2.
Let $p_{\mathsf{good}} = \Pr[\mathsf{good}, \mathsf{cyc}, \neg\mathsf{perm}, \Sigma = 0]$.
Let $p_{\mathsf{bad}} = \Pr[\mathsf{bad}, \mathsf{cyc}, \neg\mathsf{perm}, \Sigma = 0]$.
We are interested in $\Pr[\mathsf{good} \mid \mathsf{cyc}, \neg\mathsf{perm}, \Sigma = 0] = \frac{1}{1+\frac{p_{\mathsf{bad}}}{p_{\mathsf{good}}}}$.

We want to upper bound $\frac{p_{\mathsf{bad}}}{p_{\mathsf{good}}}$. And, we start with the probability p_{good}.

Note that if [good], we have [$\Sigma = 0$] and it is equivalent to [$c_1 = c'_1, c_2 = c'_2, d_1 \neq d'_1, d_2 \neq d'_2, z_1 = z'_1, z_2 = z'_2$]. When [$c_1 = c'_1, c_2 = c'_2, \mathsf{cyc}$] holds, [$\mathsf{perm}$] is equivalent to [$c'_1 = c_2$]. When [$c_1 = c'_1, c_2 = c'_2, y'_1 = y_2$] holds, $(d_1 - y_1) - (d'_2 - y'_2) = F_1(c_1) - F_1(c'_2) = F_1(c'_1) - F_1(c_2) = (d'_1 - y'_1) - (d_2 - y_2) = d'_1 - d_2$. So, $y_1 = y'_2$ is equivalent to $d_1 - d'_2 = d'_1 - d_2$.

We let A be the event [$c_1 = c'_1 \neq c_2 = c'_2, \#\{d\} = 4, d_1 + d_2 = d'_1 + d'_2$] which consists of only the c and d. Picking the xy is equivalent to picking the cd. So, A only depends on the c, d. We have $\Pr[A] \geqslant \frac{1}{N^3}\left(1 - \frac{1}{N}\right)^2\left(1 - \frac{3}{N}\right) \geqslant \frac{1}{N3}\left(1 - \frac{5}{N}\right)$

(We first pick c_1 and d_1, then $c_2 \neq c_1$, $d_1' \neq d_1$, and $d_2 \notin \{d_1, d_1', 2d_1' - d_1\}$). When A holds, $[y_1' = y_2]$ only depends on F_1 and occurs with probability $\frac{1}{N}$. When A holds, $[z_1 = z_1', z_2 = z_2']$ only depends on F_2 and occurs with probability $\frac{1}{N^2}$. Therefore,

$$\begin{aligned}
p_{\mathsf{good}} &= \Pr[\mathsf{good}, \mathsf{cyc}, \neg\mathsf{perm}, \Sigma = 0] \\
&= \Pr[{\scriptstyle c_1=c_1'\neq c_2=c_2', d_1\neq d_1', d_2\neq d_2', d_1+d_2=d_1'+d_2', y_1'=y_2, z_1=z_1', z_2=z_2'}] \\
&\geqslant \Pr[{\scriptstyle c_1=c_1'\neq c_2=c_2', \#\{d\}=4, d_1+d_2=d_1'+d_2', y_1'=y_2, z_1=z_1', z_2=z_2'}] \\
&= \Pr_{c,d}[A] \Pr_{F_1}[y_1' = y_2 | A] \Pr_{F_2}[z_1 = z_1', z_2 = z_2' | A] \\
&\geqslant \frac{1}{N^6}\left(1 - \frac{5}{N}\right)
\end{aligned}$$

Now, we compute the probability p_{bad}.

We know that $[\mathsf{bad}]$ is equivalent to $[c_1 \neq c_1' \text{ or } c_2 \neq c_2', F_1(c_1) = F_1(c_1'), F_1(c_2) = F_1(c_2'), d_1 \neq d_1', d_2 \neq d_2', z_1 = z_1', z_2 = z_2']$. When $[\mathsf{cyc}]$ occurs, $[\neg\mathsf{perm}]$ is equivalent to $[c_1' \neq c_2 \text{ or } c_1 \neq c_2']$. When $[F_1(c_1) = F_1(c_1'), F_1(c_2) = F_1(c_2')]$ holds, $[\mathsf{cyc}]$ is equivalent to $[d_1 + d_2 = d_1' + d_2', y_1' = y_2]$. When $[\mathsf{cyc}]$ holds, $[\Sigma = 0]$ is equivalent to $[c_1 + c_2 = c_1' + c_2']$. So, when $[\mathsf{cyc}, \Sigma = 0]$ occurs, $[c_1 \neq c_1' \text{ or } c_2 \neq c_2']$ is equivalent to $[c_1 \neq c_1', c_2 \neq c_2']$.

From the symmetry, $[c_1' \neq c_2 \text{ or } c_1 \neq c_2']$ case is at most twice the $[c_1' \neq c_2]$ case. Let B be the event $[c_1 \neq c_1' \neq c_2 \neq c_2', c_1 + c_2 = c_1' + c_2', d_1 + d_2 = d_1' + d_2', d_1 \neq d_1', d_2 \neq d_2']$ which consists of only the c and d. When B holds, $[F_1(c_1) = F_1(c_1'), F_1(c_2) = F_1(c_2'), y_1' = y_2]$ only depends on F_1. Therefore,

$$\begin{aligned}
p_{\mathsf{bad}} &= \Pr[\mathsf{bad}, \mathsf{cyc}, \neg\mathsf{perm}, \Sigma = 0] \\
&= \Pr[{\scriptstyle c_1\neq c_1', c_2\neq c_2', c_1'\neq c_2 \text{ or } c_1\neq c_2', c_1+c_2=c_1'+c_2', F_1(c_1)=F_1(c_1'), F_1(c_2)=F_1(c_2'),} \\
&\qquad {\scriptstyle d_1+d_2=d_1'+d_2', d_1\neq d_1', d_2\neq d_2', y_1'=y_2, z_1=z_1', z_2=z_2'}] \\
&\leqslant 2\Pr[{\scriptstyle c_1\neq c_1'\neq c_2\neq c_2', c_1+c_2=c_1'+c_2', F_1(c_1)=F_1(c_1'), F_1(c_2)=F_1(c_2'), d_1+d_2=d_1'+d_2',} \\
&\qquad {\scriptstyle d_1\neq d_1', d_2\neq d_2', y_1'=y_2, z_1=z_1', z_2=z_2'}] \\
&= 2 \Pr_{c,d,F_2}[{\scriptstyle B, z_1=z_1', z_2=z_2'}] \Pr_{F_1}[{\scriptstyle F_1(c_1)=F_1(c_1'), F_1(c_2)=F_1(c_2'), y_1'=y_2 | B, z_1=z_1', z_2=z_2'}] \\
&= 2 \Pr_{c,d,F_2}[B, z_1 = z_1', z_2 = z_2'] \times \frac{1}{N^3}
\end{aligned}$$

We split B following the $[\#\{d\} = j]$ cases for $j = 2, 3, 4$. Each case is denoted B_j. When we have $[d_1 \neq d', d_2 \neq d_2', \#\{d\} = 2, d_1 + d_2 = d_1' + d_2']$, we have either $[d_1 = d_2', d_1' = d_2]$ or $[d_1 = d_2, d_1' = d_2', d_1' = d_1 + \frac{N}{2}]$. When we have $[d_1 \neq d_1', d_2 \neq d_2', \#\{d\} = 3]$, we have $[d_1 = d_2 \text{ or } d_1' = d_2']$ (If we have $[d_1 = d_2' \text{ or } d_1' = d_2]$, then $d_1 + d_2 = d_1' + d_2'$ and $\#\{d\} = 2$ conflicts). When we have

$[d_1 \neq d_1', d_2 \neq d_2', \#\{d\} = 4]$, we have no equality of d's. For B_4,

$$\begin{aligned}
&\Pr_{c,d,F_2}[B_4, z_1 = z_1', z_2 = z_2'] \\
&= \Pr_{c,d}[B_4] \Pr_{F_2}[z_1 = z_1', z_2 = z_2' | B_4] \\
&= \Pr_{c,d}[c_1 \neq c_1' \neq c_2 \neq c_2', c_1 + c_2 = c_1' + c_2', d_1 + d_2 = d_1' + d_2', \#\{d\} = 4] \Pr_{F_2}[z_1 = z_1', z_2 = z_2' | B_4] \\
&\leqslant \Pr_{c,d}[c_1 + c_2 = c_1' + c_2', d_1 + d_2 = d_1' + d_2'] \Pr_{F_2}[z_1 = z_1', z_2 = z_2' | B_4] \\
&= \frac{1}{N^4}
\end{aligned}$$

For each of the two cases of B_3, either $z_1 = z_1'$ or $z_2 = z_2'$ occurs with probability $\frac{1}{N}$. So,

$$\begin{aligned}
&\Pr_{c,d,F_2}[B_3, z_1 = z_1', z_2 = z_2'] \\
&\leqslant 2 \Pr_{c,d}[c_1 + c_2 = c_1' + c_2', d_1 + d_2 = d_1' + d_2', d_1 = d_2] \Pr_{F_2}[z_1 = z_1']|d_1 \neq d_1'] \\
&= \frac{2}{N^4}
\end{aligned}$$

For B_2,

$$\begin{aligned}
&\Pr_{c,d,F_2}[B_2, z_1 = z_1', z_2 = z_2'] \\
&\leqslant \Pr_{c,d}[B_2] \\
&= \Pr_{c,d}[c_1 \neq c_1' \neq c_2 \neq c_2', c_1 + c_2 = c_1' + c_2', d_1 + d_2 = d_1' + d_1', d_1 \neq d_1', d_2 \neq d_2', \#\{d\} = 2] \\
&= \Pr_{c,d}[c_1 + c_2 = c_1' + c_2', d_1 + d_2 = d_1' + d_2', d_1 = d_2', d_1' = d_2'] + \\
&\quad \Pr_{c,d}[c_1 + c_2 = c_1' + c_2', d_1 = d_2, d_1' + d_2', d_1' = d_1 + \tfrac{N}{2}] \\
&= \frac{2}{N^4}
\end{aligned}$$

Therefore, $\Pr_{c,d,F_2}[B, z_1 = z_1', z_2 = z_2'] \leqslant \frac{5}{N^4}$ and $p_{\mathsf{bad}} \leqslant \frac{10}{N^7}$.

Finally, $\frac{p_{\mathsf{bad}}}{p_{\mathsf{good}}} \leqslant \frac{10}{N-5}$. We deduce

$$\Pr[\mathsf{good} \mid \mathsf{cyc}, \neg\mathsf{perm}, \Sigma = 0] \geqslant \frac{1}{1 + \frac{10}{N-5}}$$

□

A.2 Extended Lemma 4

Lemma 7. *If* $v_1 v_2 \cdots v_i \cdots v_L v_1$ *is a cycle of length* L *in* G *with zero sum of labels and the vertices use no* d_i *or* c_i *in common, then all* v_i *are likely to be good. More precisely, for* $v_i = (x_i y_i x_i' y_i')$ *random, we have* $\Pr\left[\forall i, v_i \in V_{good} \,\middle|\, v_1 \cdots v_i \cdots v_L v_1 \text{ is a cycle}, \left(\#\{c\} = \#\{c'\} = L, \forall i \neq j \;\; c_i \neq c_j'\right), \left(\#\{d\} = L, \forall i,j \;\; d_i \neq d_j'\right), \sum_{i=1}^{L} \mathsf{Label}(v_i) = 0\right] \geqslant \frac{1}{1 + \frac{2^L - 1}{N}}$.

Proof. We compute $p = \Pr[\mathtt{good} \mid \mathtt{good} \vee \mathtt{bad}, \mathtt{cyc}, \neg\mathsf{repeat}_c, \neg\mathsf{repeat}_d, \Sigma = 0]$, where we use the same notation as in Lemma 4 with new $[\neg\mathsf{repeat}_c]$ and $[\neg\mathsf{repeat}_d]$ notations. We define them as follows:

We note that when all v_i are vertices (good or bad), since $F_1(c_i') = F_1(c_i)$, $y_{i+1}' = y_i$ is equivalent to $d_i' - d_{i+1} = F_1(c_i) - F_1(c_{i+1})$. We further note that when this holds, then $\sum d_i = \sum d'$. To be able to compute the probability of $[\mathsf{cyc}]$, we introduce a condition on the non-repetition of the c and c', except for the possible equalities $c_i = c_i'$ in good vertices. Namely, we define

$$[\neg\mathsf{repeat}_c] : \left(\#\{c\} = \#\{c'\} = L \quad , \quad \forall i \neq j \quad c_i \neq c_j'\right)$$

When $[\neg\mathsf{repeat}_c, \sum d = \sum d']$ holds and all v_i are vertices, $[\mathsf{cyc}]$ occurs with probability $\frac{1}{N^{L-1}}$. Therefore, $\Pr[\mathtt{cyc} \mid \mathtt{good} \vee \mathtt{bad}, \neg\mathsf{repeat}_c, \Sigma d = \Sigma d'] = \frac{1}{N^{L-1}}$

The event $[\forall i \quad z_i = z_i']$ is equivalent to $c_i + F_2(d_i) = c_i' + F_2(d_i')$. To be able to compute its probability, we introduce a condition on the non-repetition of the d and d'. Namely, we define

$$[\neg\mathsf{repeat}_d] : \left(\#\{d\} = L \quad , \quad \forall i, j \quad d_i \neq d_j'\right)$$

Hence, when $[\neg\mathsf{repeat}_d]$ occurs, $[\forall i \quad z_i = z_i']$ occurs with probability $\frac{1}{N^L}$: $\Pr[z' = z \mid \neg\mathsf{repeat}_d] = \frac{1}{N^L}$. Finally, when $[\mathtt{cyc}]$ holds, $[\Sigma = 0]$ is equivalent to $\Sigma(c - c') = 0$, and $[\mathtt{good} \vee \mathtt{bad}]$ is equivalent to $[F_1(c) = F_1(c'), z' = z]$.

We define

$$p_{\mathsf{good}} = \Pr[{\scriptstyle c=c', \neg\mathsf{repeat}_c, \mathsf{cyc}, \neg\mathsf{repeat}_d, z'=z}]$$
$$p_{\mathsf{bad}} = \Pr\left[{\scriptstyle \neg(c=c'), F_1(c)=F_1(c'), \sum(c-c')=0, \neg\mathsf{repeat}_c, \mathsf{cyc}, \neg\mathsf{repeat}_d, z'=z}\right]$$

with obvious shorthands $[c = c']$, $[z' = z]$, $[F_1(c) = F_1(c')]$, $[\sum(c - c') = 0]$.

We upper bound $\frac{p_{\mathsf{bad}}}{p_{\mathsf{good}}}$ to compute p.

We have

$$\begin{aligned}
p_{\mathsf{good}} &= \Pr[c = c', \neg\mathsf{repeat}_c, \mathsf{cyc}, \neg\mathsf{repeat}_d, z' = z] \\
&= \Pr\left[c = c', \neg\mathsf{repeat}_c \sum d = \sum d', \mathsf{cyc}, \neg\mathsf{repeat}_d, z' = z\right] \\
&= \frac{1}{N^{2L-1}} \Pr[c = c', \neg\mathsf{repeat}_c] \Pr\left[\sum d = \sum d', \neg\mathsf{repeat}_d\right] \\
&= \frac{1}{N^{3L-1}} \frac{N(N-1)\cdots(N-L+1)}{N^L} \Pr\left[\sum d = \sum d', \neg\mathsf{repeat}_d\right]
\end{aligned}$$

$$\begin{aligned}
p_{\mathsf{bad}} &= \Pr\left[{\scriptstyle \neg(c=c'), F_1(c)=F_1(c'), \sum(c-c')=0, \neg\mathsf{repeat}_c, \mathsf{cyc}, \neg\mathsf{repeat}_d, z'=z}\right] \\
&= \Pr\left[{\scriptstyle \neg(c=c'), F_1(c)=F_1(c'), \sum(c-c')=0, \neg\mathsf{repeat}_c \sum d=\sum d', \mathsf{cyc}, \neg\mathsf{repeat}_d, z'=z}\right] \\
&= \frac{1}{N^{2L-1}} \Pr[{\scriptstyle \neg(c=c'), F_1(c)=F_1(c'), \sum(c-c')=0, \neg\mathsf{repeat}_c}] \Pr\left[{\scriptstyle \sum d=\sum d', \neg\mathsf{repeat}_d}\right]
\end{aligned}$$

So,

$$\begin{aligned}
\frac{\mathsf{p}_{\mathsf{bad}}}{\mathsf{p}_{\mathsf{good}}} &= \frac{\mathsf{N}^{2\mathsf{L}}}{\mathsf{N}(\mathsf{N}-1)\cdots(\mathsf{N}-\mathsf{L}+1)} \Pr\left[\scriptstyle \neg(c=c'),F_1(c)=F_1(c'),\sum(c-c')=0,\neg\mathsf{repeat}_c\right] \\
&= \frac{\mathsf{N}^{2\mathsf{L}}}{\mathsf{N}(\mathsf{N}-1)\cdots(\mathsf{N}-\mathsf{L}+1)} \sum_{I\neq\emptyset} \Pr\left[\begin{array}{l}\scriptstyle \neg\mathsf{repeat}_c \\ \scriptstyle \forall i\notin I \quad c_i=c'_i \\ \scriptstyle \forall i\in I \quad c_i\neq c'_i,F_1(c_i)=F_1(c'_i) \\ \scriptstyle \sum_{i\in I}(c_i-c'_i)=0\end{array}\right] \\
&\leqslant \frac{\mathsf{N}^{2\mathsf{L}}}{\mathsf{N}(\mathsf{N}-1)\cdots(\mathsf{N}-\mathsf{L}+1)} \sum_{I\neq\emptyset} \Pr\left[\begin{array}{l}\scriptstyle \neg\mathsf{repeat}_c \text{ except } c'_{\max I} \\ \scriptstyle \forall i\notin I \quad c_i=c'_i \\ \scriptstyle \forall i\in I\setminus\{\max I\} \quad c_i\neq c'_i,F_1(c_i)=F_1(c'_i) \\ \scriptstyle \sum_{i\in I}(c_i-c'_i)=0 \\ \scriptstyle F_1(c_{\max I})=F_1(c'_{\max I})\end{array}\right] \\
&= \frac{\mathsf{N}^{2\mathsf{L}}}{\mathsf{N}(\mathsf{N}-1)\cdots(\mathsf{N}-\mathsf{L}+1)} \sum_{I\neq\emptyset} \frac{\mathsf{N}(\mathsf{N}-1)\cdots(\mathsf{N}-\mathsf{L}-\#I)}{\mathsf{N}^{2\mathsf{L}+\#I}} \\
&= \sum_{I\neq\emptyset} \frac{(\mathsf{N}-\mathsf{L})(\mathsf{N}-\mathsf{L}-1)\cdots(\mathsf{N}-\mathsf{L}-\#I)}{\mathsf{N}^{\#I}} \\
&\leqslant \sum_{I\neq\emptyset} \frac{1}{\mathsf{N}} = \frac{2^{\mathsf{L}}-1}{\mathsf{N}}
\end{aligned}$$

where $[\neg\mathsf{repeat}_c$ except $c'_{\max I}]$ means

$$\begin{cases} \#\{c\} = \mathsf{L} \\ \#\{c'_1,\ldots,c'_{\max I-1},c'_{\max I+1},\ldots,c'_{\mathsf{L}}\} = \mathsf{L}-1 \\ \forall i\forall j \neq \max I \ i\neq j \implies c_i \neq c'_j \end{cases}$$

By relaxing the constraints on $c'_{\max I}$, we can compute the probability of $\Sigma(c-c')=0$ conditioned to other events about c and c'. This probability is $\frac{1}{\mathsf{N}}$.

Therefore,

$$\frac{\mathsf{p}_{\mathsf{bad}}}{\mathsf{p}_{\mathsf{good}}} \leqslant \frac{2^{\mathsf{L}}-1}{\mathsf{N}}$$

and we have

$$\frac{1}{1+\frac{\mathsf{p}_{\mathsf{bad}}}{\mathsf{p}_{\mathsf{good}}}} \geqslant \frac{1}{1+\frac{2^{\mathsf{L}}-1}{\mathsf{N}}}$$

□