Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.6, Dec. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.6.1261

일반화된 Feistel 구조에 대한 중간 일치 공격*

성 재 철†‡
서울시립대학교 수학과

Meet-in-the-Middle Attacks on Generalized Feistel Networks*

Jaechul Sung†‡
Department of Mathematics, University of Seoul

요 약

블록 암호 설계에서 Feistel 구조는 가장 널리 사용되는 구조 중의 하나이다. 또한 Feisel 구조를 확장하여 일반화한 Feistel 구조 역시 블록 암호 뿐 아니라 해쉬 함수에서도 널리 사용되는 구조이다. Feistel 구조의 구조적 안전성에 대한 다양한 분석 및 많은 연구가 진행되었다. 이 중 최근 Feistel 구조에 대한 중간 일치 공격은 Feistel 구조의 구조적 안전성을 가장 효과적으로 분석하는 방법 중 하나이다. 본 논문에서는 일반화된 Feistel 구조에 대한 중간 일치 공격에 대한 안전성을 분석한다.

ABSTRACT

Feistel Networks are one of the most well-known schemes to design block ciphers. Generalized Feistel Networks are used to construct only block ciphers but also hash functions. Many generic attacks on Feistel schemes have been studied. Among these attacks, recently proposed meet-in-the-middle attacks are one of the most effective attacks. In this paper, we analyze the security of meet-in-the-middle attacks on generalized Feistel Networks.

Keywords: Block Ciphers, Generalized Feistel Networks, Meet-In-The-Middle, Generic Attacks

I. 서 론

DES 구조로 사용된 Feistel 구조는 가장 널리 사용되는 블록 암호 설계 구조이다[1]. Feistel 구조는 $n/2$-비트 의사랜덤함수(PRF)를 이용하여 n-비트 의사랜덤순열(PRP)를 설계하는 아주 효율적이고 유용한 방법으로 이 설계 방법에 대한 구조적 안전성도 증명되었다[2,3].

Feistel 구조를 변형 및 일반화한 구조가 일반화된 Feistel 구조이다[4]. Feistel 구조는 2개의 branch를 이용하는 구조라고 한다면 일반화된 Feistel 구조는 branch의 개수를 n으로 확장하고 세 가지 Type으로 분류할 수 있다[5]. 이러한 일반화된 Feistel 구조를 GFN라 한다. 이러한 GFN 구조는 블록 암호 뿐 아니라 해쉬 함수에서도 구조적 설계 논리로 널리 사용되고 있다. 예를 들자면 GFN Type-I은 CAST-256, Type-II는 RC6, Type-III MARS 블록 암호에 사용되었다.

블록 암호 분석 시 암호 알고리즘의 세부 논리를 모두 분석하여 분석하는 방법과 달리 블록 암호의 전체 구조를 분석하는 방법이 generic attack이다. Feistel 구조에 대한 차분 및 선형 분석에 대한 구조적 안전성 및 구조적 안전성에 대한 다양한 분석이 연구되었다[6,7,8,9,10]. 최근 J.Guo 등은 이러한 Feistel 구조에 대한 구조적 안전성을 차분의 개념을 활용한 MITM 공격 기법을 이용하여 6-라운드

Received(09. 08. 2017), Modified(10. 18. 2017), Accepted(10. 21. 2017)
* 이 논문은 2017년도 서울시립대학교 연구년교수 연구비에 의하여 연구되었음.
† 주저자, jcsung@uos.ac.kr
‡ 교신저자, jcsung@uos.ac.kr(Corresponding author)

Feistel 구조를 분석하였다[11]. 이 공격은 ASIACRYPT 2014의 Feistel 구조에 대한 MITM 공격 논문[12]을 수정 및 보완한 논문으로 최근까지 제안된 Feistel 구조에 대한 generic attack 중 가장 효과적인 분석 방법이다.

Feistel 구조와 더불어 GFN 구조에 대한 다양한 구조적 분석 방법들도 다양하게 연구되었다[13,14,15]. 하지만 대부분의 분석은 차분 분석에 대한 구조적 안전성에 중점을 두고 있다. 본 논문에서는 GFN 구조에 대하여 MITM 기법을 적용하여 새로운 공격 기법을 제시하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 Feistel 구조와 GFN 구조를 정의한다. 3장에서는 Feistel 구조에 대한 MITM 기법을 소개한다. 4장과 5장에서는 GFN-I-4와 GFN-II-4에 대한 새로운 MITM에 대하여 안전성을 분석한다. 끝으로 6장에서 결과를 정리하고 향후 과제를 제시한다.

II. 공격 모델

본 절에서는 논문에 사용된 표기법과 블록 암호의 구조를 소개한다. 특히 공격의 모델이 되는 Feistel 구조와 일반화된 Feistel 구조를 정의한다.

DES와 같은 블록 암호의 구조로 가장 널리 사용되는 방법이 Feistel 구조이다. 이 구조에서 가장 널리 사용되는 방법은 라운드 함수 입력 전에 라운드 키와 XOR 연산 후 라운드 함수를 거치는 방법으로 설계된다.

블록 암호의 블록 길이를 n이라 하자. 그러면 Feistel 구조 설계에서의 i-라운드 입력을 (v_i, v_{i-1})를 입력 값, K_i를 라운드 키, F_i를 라운드 함수라고 하자. 그러면 i-라운드 출력 값 (v_{i+1}, v_i)는 다음과 같이 정의할 수 있다. 여기서 v_i는 $n/2$-비트의 값이다.

$$v_{i+1} = F_i(v_i \oplus K_i) \oplus v_{i-1}$$

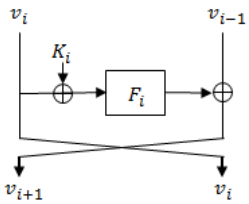

Fig. 1. Feistel Structure

일반화된 Feistel 구조(GFN)는 Feistel 구조의 branch의 수를 m개로 확장시킨 구조로 다음과 같은 세 가지 Type으로 분류한다[5]. GFN 구조에서 Type I, II, III와 branch의 수가 m인 경우를 각각 GFN-I-m, GFN-II-m, GFN-III-m으로 표기한다.

본 논문에서는 $m=4$인 경우, 즉 branch의 수가 4인 경우와 Type I, Type II를 중점적으로 살펴본다. Type III인 경우는 Type I을 3-라운드 변형시킨 구조로 볼 수 있기 때문에 Type I의 분석 방법을 이용하면 쉽게 적용할 수 있을 것이다.

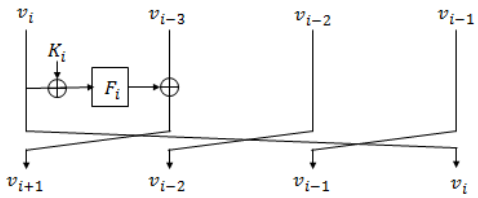

Fig. 2. GFN-I-4

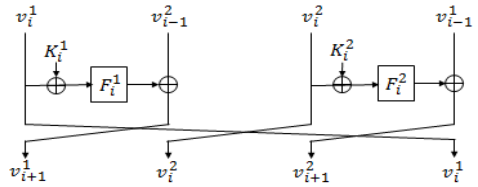

Fig. 3. GFN-II-4

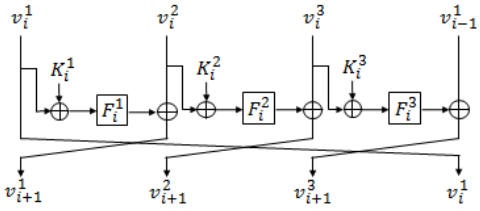

Fig. 4. GFN-III-4

III. Feisel 구조에 대한 중간 일치 공격[11]

본 장에서는 J. Guo et al.이 [11]에서 제안한 중간 일치 공격 기법을 소개한다. 블록 암호뿐 아니라 해쉬 함수 등에서 중간 일치 공격(Meet-In-The-Middle, MITM)은 암호 알고리즘의 분석에 널리 사용되는 기법 중의 하나이다.

차분 분석은 블록 암호의 대표적 공격 기법이다[6]. 주어진 입력 차분에 대해 원하는 출력 차분을

만족하는 평문 및 암호문 쌍을 이용하여 distinguisher를 구성하거나 distinguisher 전후로 몇 라운드 키를 추측하여 라운드 키를 복구하는 공격이다.

앞 장에서 살펴본 Feistel 구조에서 라운드 함수 F_i의 입력 차분을 α, 출력 차분을 β라 하자. 주어진 (α,β)에 대해 다음 식을 살펴보자.

$$F_i(x)\oplus F_i(x\oplus\alpha)=\beta \quad (1)$$

일반적으로 식 (1)을 만족하는 해의 개수는 라운드 함수 F_i에 의존한다. 만약 라운드 함수가 비선형 함수라면 (1)의 식을 만족하는 해의 수가 많은 경우도 있고 해가 존재하지 않을 수도 있다. 평균적으로 (1)의 식을 만족하는 해의 수는 1이다. 또한 입력 차분과 출력 차분만 주어진 경우 (1)을 만족하는 해를 찾기 위해서는 라운드 함수에 대한 전수조사가 필요하다. 만약 라운드 함수의 차분 분포표(Differential Distribution Table, DDT)가 미리 선 계산 되어질 수 있다면 쉽게 해를 찾을 수도 있다. 하지만, 라운드 함수의 비트가 $n/2$비트라면 DDT를 구성하는데 2^n의 시간 복잡도와 메모리 복잡도가 필요하다.

Feistel 구조의 중간 일치 공격을 이해하기 위해 스퀘어 공격 혹은 Integral 공격에 사용되는 $\delta-set$을 다음과 같이 정의하자.

〔Definition 1〕 ($b-\delta-set$)

상태 값 중 특정 b-비트만 다르고 나머지 비트는 다 같은 2^b개의 집합을 $b-\delta-set$이라 정의한다.

위의 정의를 이용하면 스퀘어 공격에 주로 사용되는 하나의 S-박스의 값만 active하게 만들고 나머지는 모두 같게 설정하는 집합은 $8-\delta-set$으로 2^8의 원소로 구성된 집합이 된다. 또한

어떤 주어진 상태 값 v에 대에 대해 $b-\delta-set$을 구성한다고 가정하자. 또한 특정 b-비트를 최하위비트라고 한다면 v에 대한 $b-\delta-set$는 다음과 같다.

$$\{v, v\oplus 1, \mathrm{v}\oplus 2, \cdots, \mathrm{v}\oplus 2^{\mathrm{b}}-1\}$$

또한 위 $b-\delta-set$을 라운드 함수 F_i를 계산한 후의 값은 다음과 같이 표현된다.

$$\{F_i(v), F_i(v\oplus 1), F_i(v\oplus 2), \cdots, F_i(v\oplus 2^b-1)\}$$

우리는 앞으로 위의 라운드 함수 출력 값에서 출력 차분만을 고려한 2^b-1 길이의 다음 수열의 변화를 살펴볼 것이다.

$$\begin{Bmatrix} F_i(v)\oplus F_i(v\oplus 1), \\ F_i(v)\oplus F_i(v\oplus 2), \\ \vdots \\ F_i(v)\oplus F_i(v\oplus 2^b-1) \end{Bmatrix},$$

3.1 Feistel 구조에 대한 5-라운드 distinguisher

우선 Feistel 구조에 대한 새로운 5-라운드 distinguisher를 살펴보자. Fig. 1의 Feistel 구조를 5개의 라운드 함수 F_i와 라운드 K_i가 적용된 구조이다. 또한 마지막 swap은 없는 것으로 한다. 그러면 5-라운드 distinguisher는 다음과 같다.

〔Lemma 1〕〔11〕

0이 아닌 차분 $X, Y(\in\{0,1\}^{n/2})$에 대하여 $X\neq Y$이고 입력 $(m, m\oplus(0,X))$에 대한 Feistel 구조 5-라운드 암호화 후의 출력 차분이 $(0, Y)$라 하자. 그러면 중간 3 라운드(2, 3, 4 라운드)의 가능한 내부 상태의 값의 개수는 평균적으로 $2^{n/2}$이다.

$X, Y(\neq X)$인 고정된 입력 차분이 $(0, X)$이고 출력 차분이 $(0, Y)$이다. 그러면 1 라운드와 5 라운드의 입력 차분이 0이므로 확률 1로 출력 차분도 0이다. 2-라운드의 차분을 Δ라 하면 2,3,4 라운드 함수의 입출력 차분에 관한 식은 다음과 같다.

$$2\text{ 라운드 : } F_2(x)\oplus F_2(x\oplus X)=\Delta \quad (2)$$

$$3\text{ 라운드 : } F_3(x)\oplus F_3(x\oplus\Delta)=X\oplus Y \quad (3)$$

$$4\text{ 라운드 : } F_4(x)\oplus F_4(x\oplus Y)=\Delta \quad (4)$$

고정된 Δ에 대하여, 위의 식을 만족하는 가능한 해의 개수는 평균적으로 하나이다. 또한 가능한 Δ의 개수는 $2^{n/2}$이므로 2,3,4 라운드의 가능한 상태 값의 개수는 평균적으로 $2^{n/2}$이다.

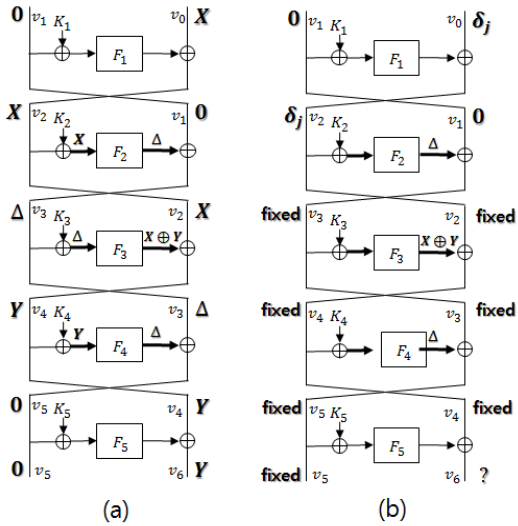

Fig. 5. 5-Round Distinguisher and $b-\delta-set$ Sequence of Feistel Structure

평문 $(m, m\oplus(0,X))$ 및 이에 대응하는 5-라운드 암호문 쌍의 차분이 Fig. 5의 5-라운드 distinguisher를 만족한다고 가정하자. 하나의 고정된 Δ에 대하여 그러면 식 (2), (3), (4)를 만족하는 내부 상태의 값을 평균적으로 하나 구할 수 있다.

평문 m과 고정된 Δ에 대하여 (2), (3), (4)를 만족하는 내부 상태의 값이 각각 s_2, s_3, s_4라 하자. 그러면 각 차분 δ_j에 대하여 평문 쌍 $(m, m\oplus(0,\delta_j))$에 대한 5-라운드 암호문의 Δv_5의 값을 구할 수 있다. $\Delta v_2=\delta_j$이고 (2)의 해가 s_2이므로 평문 m에 대한 F_2 함수의 입력 값은 s_2이고 $m\oplus\delta_j$에 대한 F_2 함수의 입력 차분이 δ_j이므로 F_2 함수의 입력은 $s_2\oplus\delta_j$이므로 출력 값은 $F(s_2\oplus\delta_j)$가 된다. 따라서 2 라운드 F_2 함수의 출력 차분은 $F_2(s_2)\oplus F_2(s_2\oplus\delta_j)$이 된다. 그러므로 3 라운드 입력 차분 Δv_3을 구할 수 있다.

같은 방법으로 3-라운드 입력 차분과 m에 대한 상태 값 s_3를 알 수 있으므로 4-라운드 차분 Δv_4를 구할 수 있다. 또한 4-라운드 입력 차분과 입력 차분과 m에 대한 상태 값 s_4를 이용하면 Δv_5의 차분을 구할 수 있게 된다. 모든 δ_j에 다하여 모두 할 수 있으므로 Δv_5의 2^b-1개의 차분 수열을 값을 완벽하게 계산할 수 있다.

5-라운드 Feistel 구조 암호화 함수를 F라 하면 m, δ_j, Δ를 이용한 다음의 함수를 정의하자.

$$F^{\Delta}(m,\delta_j)=msb_{n/2}(F(m)\oplus F(m\oplus(0,\delta_j)))$$

여기서 $msb_{n/2}$는 n-비트 중 최상위 $n/2$비트만을 취하는 함수이다. 이 함수를 이용하면 2^b-1개로 구성된 $F^{\Delta}(m,\delta_j)$의 수열의 값을 정확하게 계산가능하게 된다.

그런데, Δ의 가능한 수의 개수의 최대 $2^{n/2}$개 이므로 2^b-1개로 구성된 Δv_5의 차분 수열을 $2^{n/2}$개 계산할 수 있게 된다. 이를 정리하면 다음과 같다.

〔Proposition 1〕〔11〕

0이 아닌 차분 $X, Y(\in\{0,1\}^{n/2})$에 대하여 $X\neq Y$이고 입력 $(m, m\oplus(0,X))$에 대한 Feistel 구조 5-라운드 암호화 후의 출력 차분이 $(0, Y)$라 하자. 최하위 b-비트만 변화하여 만들어진 $b-\delta-set$에 대한 길이 2^b-1로 구성된 $F^{\Delta}(m,\delta_j)$의 수열은 평균적으로 $2^{n/2}$개 이다.

이 5-라운드 distinguisher는 아주 특이한 특성이다. 만약 이 5-라운드 distinguisher를 메모리를 이용하여 미리 계산할 수만 있다면 쉽게 6-라운드 공격을 할 수 있다.

3.2 Feistel 구조에 대한 6-라운드 공격

앞에서 구성된 5-Round distinguisher를 이용하여 차분 분석 등에 많이 사용되는 1R attack 기법을 활용하여 6-라운드 Feistel 구조를 공격할 수 있다.

6-라운드 공격은 Precomputation Phase와 Online Phase의 두 부분으로 나눌 수 있다. Precomputation Phase에서는 우선 Algorithm 1을 이용하여 가능한 내부 상태의 값에 대하여 테이블을 저장하고 Algorithm 2에서는 이 내부 상태의 값을 이용하여 모든 가능한 Δv_5 수열을 T_δ에 저장한다.

주어진 X에 대한 2^y개의 Y에 대하여, Algorithm 1과 2를 사용하여 T_δ를 선계산한다면 Precomputation Phase에서는 $2^{n/2+y}$번의 연산과 메모리가 필요하다.

Online Phase에서는 우선 Algorithm 3를 이

Algorithm 1. [11] Construction the sequence of the table T_2, T_3, T_4 of Feistel strucure

Input: X, Y and F_2, F_3, F_4
Output: Table T_2, T_3, T_4

1. for $i = 0, 1, ..., 2^{n/2} - 1$ do
2. compute $\Delta F_2^O = F_2(i) \oplus F_2(i \oplus X)$
3. store $(i, \Delta F_2^O)$ in T_2 indexed by ΔF_2^O
4. compute $\Delta F_4^O = F_4(i) \oplus F_4(i \oplus Y)$
5. store $(i, \Delta F_4^O)$ in T_4 indexed by ΔF_4^O
6. for $i = 0, 1, ..., 2^{n/2} - 1$ do
7. store $(i, F_3(i))$ in a temp indexed by $F_3(i)$
8. for $i = 0, 1, ..., 2^{n/2} - 1$ do
9. compute $F_3(i) \oplus (X \oplus Y)$
10. find j s.t. $F_3(j) = F_3(i) \oplus (X \oplus Y)$ in temp
11. store $(i, i \oplus j)$ in T_3 indexed by $i \oplus j$

Algorithm 2. [11] Construction the sequence of the table T_δ (Δv_5 sequence) of Feistel strucure

Input: T_2, T_3, T_4
Output: T_δ (Δv_5 sequence)

1. for $i = 0, 1, ..., 2^{n/2} - 1$ do
2. obtain internal state values F_2^I, F_3^I, F_4^I by looking up T_2, T_3, T_4
3. for all b active bits of $b-\delta-set$ do
4. modify Δv_0 and compute Δv_5
5. compute the sequence of Δv_5 and add it to T_δ

Algorithm 3. [11] Data collection phase of the 6-round attack of Feistel strucure

Input: T_δ
Output: Possible (P, C) pairs such that satisfies 5-round distinguisher

1. Choose 2^y differences Y so that the $n/2 - y$ MSBs of Y are 0 for all Y
2. Choose X such that $X \neq Y$
3. for $2^{n/2-y}$ different values of v_0 do
4. for all $2^{n/2}$ choices of v_{-1} do
5. query (v_0, v_{-1}) and store it in L_0
6. query $(v_0 \oplus X, v_{-1})$ and store in L_1
7. pick up the elements of $L_0 \times L_1$ whose ciphertext match in the $n-y$ MSBs

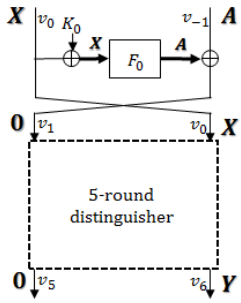

Fig. 6. 6-Round Feistel Structure Attack

용하여 6-라운드 암호화된 암호문의 차분이 $(0, Y)$가 되는 평문-암호문 쌍을 찾아낸다. 이 단계에서는 고정된 v_0에 대한 평문쌍 2^n개 중 $(0, Y)$를 만족하는 것의 개수는 평균적으로 2^y이다. v_0의 개수가 $2^{n/2-y}$이므로 가능한 쌍은 $2^{n/2}$개가 된다.

Online Phase 후 남은 $2^{n/2}$개의 쌍 중 1-라운드 후 차분이 $(0, X)$를 만족하는 K_0는 평균적으로 1개가 존재한다. 이를 만족하는 평문쌍과 키에 대해 v_0 값에서 $b-\delta-set$를 계산하고 키를 이용하여 v_{-1}를 계산하여 평문 입력 $b-\delta-set$에 대한 Δv_5를 계산한다. 이 계산된 Δv_5 값이 미리 계산된 테이블 T_δ에 매치한다. 만약 T_δ와 매치가 되면 K_0의 값이 복구된다.

Online Phase에서 $2^{n/2}$개를 T_δ개를 체크한다. 잘못된 K_0가 이 T_δ의 수열과 매치가 되기 위한 확률은 다음과 같다.

$$\frac{2^{n/2}}{2^{n2^b/2}} = 2^{n(1-2^b)/2}$$

여기서 $b \geq 2$가 되면 잘못된 키는 거를 수 있게 된다. 이 6-라운드 공격에서는 데이터 복잡도는 $2^{n/2-y+1}$개의 선택평문, 시간 복잡도는 $2^{n/2+y}$이고, 메모리 복잡도는 2^{n-y+1}이다.

$y = n/4$로 택하면 복잡도를 최적화할 수 있다. 이 때 데이터 복잡도는 $2^{3n/4}$, 시간 복잡도는 $2^{3n/4}$, 메모리 복잡도 $2^{3n/4}$이 된다. 즉, 라운드 함수의 구조에 상관없이 6-라운드 Feistel 구조는 분석 가능하다.

IV. GFN-I-4 구조에 대한 중간 일치 공격공격

본 장에서는 4 branch를 갖는 일반화된 Feistel

구조의 Type I 구조인 GFN-I-4에 대한 중간 일치 공격을 살펴본다.

4.1 GFN-I-4에 대한 9-라운드 distinguisher

GFN-I-4에 대한 새로운 distinguisher를 구성하기 위해 차분 특성을 먼저 구성한다. 다음 Fig. 7은 GFN-I-4에 대한 9-라운드 차분 특성이다. 입력 차분 $(0,0,0,X)$에 대하여 출력 차분 $(0,Y,Z,W)$이다. 여기서 X,Y,Z,W는 0이 아닌 차분이고 $W \neq X$이다.

〔Lemma 2〕

0이 아닌 차분 $X,Y,Z,W(\in\{0,1\}^{n/4})$에 대하여 $X \neq W$이고 입력 $(m, m\oplus(0,0,0,X))$에 대한 GFN-I-4 구조 9-라운드 암호화 후의 출력 차분이 $(0,Y,Z,W)$라 하자. 그러면 중간 5 라운드(4~8 라운드)의 라운드 함수의 가능한 내부 상태의 값의 개수는 평균적으로 $2^{n/4}$이다.

(증명) 입력 차분은 $(0,0,0,X)$, 출력 차분은 $(0,Y,Z,W)$으로 고정되었다고 하자. 그러면 1, 2, 3 라운드의 라운드 함수의 입력 차분은 0이므로 출력 차분 또한 확률 1로 0이 된다. 또한 4-라운드에서 라운드 함수의 차분을 Δ라고 하자. 그러면 4~8라운드 함수의 입력과 출력 차분은 다음과 같다.

- 4 라운드 : $F_4(x)\oplus F_4(x\oplus X)=\Delta$
- 5 라운드 : $F_5(x)\oplus F_5(x\oplus\Delta)=Y$
- 6 라운드 : $F_6(x)\oplus F_6(x\oplus Y)=Z$
- 7 라운드 : $F_7(x)\oplus F_7(x\oplus Z)=W\oplus X$
- 8 라운드 : $F_8(x)\oplus F_8(x\oplus W)=\Delta$

이때 X,Y,Z,W는 고정된 값이다. 하나의 고정된 Δ에 대하여 4~8 라운드의 라운드 함수에 대한 입력 차분과 출력 차분은 하나의 값으로 정해진다. 또한 라운드 함수가 선형 함수가 아니라면 각 라운드 함수에 대해 평균적으로 하나의 해가 존재한다. 따라서 4~8 라운드 내부 상태의 값은 평균적으로 하나의 해가 존재하게 된다. 그런데 가능한 Δ의 개수의 상한이 $2^{n/4}$이므로 가능한 상태의 값은 $2^{n/4}$보다 작거나 같다. ▣

이제 GFN-I-4에 대한 9-라운드 distinguisher를 구성할 수 있다. Feistel 구조에서와 비슷한 방법으로 9-라운드 GFN-I-4 구조 암호화 함수를 F라 하면 m,δ_j,Δ를 이용한 다음의 함수를 정의하자.

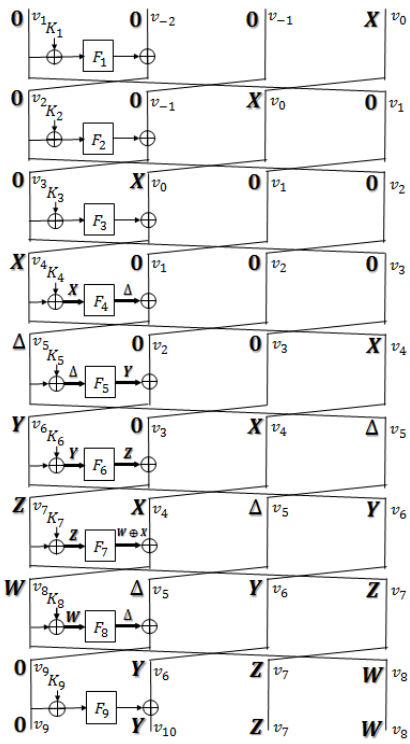

Fig. 7. 9-Round Distinguisher of GFN-I-4

$$F^{\Delta}(m,\delta_j)=msb_{n/4}(F(m)\oplus F(m\oplus(0,0,0,\delta_j)))$$

여기서 $b-\delta-set$는 입력 차분의 최하위 b-비트를 변화하여 만든 집합이다. 고정된 Δ에 대해 그러면 2^b-1개로 구성된 $F^{\Delta}(m,\delta_j)$의 수열의 값을 정확하게 계산가능하게 된다. 가능한 Δ가 $2^{n/4}$보다 작기 때문에 다음을 얻을 수 있다.

〔Proposition 2〕

0이 아닌 차분 $X,Y,Z,W(\in\{0,1\}^{n/4})$에 대하여 $X \neq W$이고 입력 $(m, m\oplus(0,0,0,X))$에 대한 GFN-I-4 구조 9-라운드 암호화 후의 출력 차분이 $(0,Y,Z,W)$라 하자. 그러면 최하위 b-비트만 변화하여 만들어진 $b-\delta-set$에 대한 길이 2^b-1로 구성된 $F^{\Delta}(m,\delta_j)$의 수열은 평균적으로 $2^{n/4}$개 이다.

4.2 GFN-I-4에 대한 10-라운드 공격

10-라운드 GFN-I-4에 대한 공격은 9-라운드 distinguisher를 이용하여 앞에 1-라운드를 덧붙이는 공격으로 첫 라운드의 키를 복구하는 공격이다. 이 공격 역시 Precomputation Phase와 Online Phase 두 과정으로 구성된다.

우선 9-라운드 distinguisher의 Δv_9 수열의 값을 계산한다. Feistel 구조의 Algorithm 1과 2를 한꺼번에 수행한 Algorithm 4는 4~8 라운드의 입출력 차분에 대한 테이블을 구성한 후 Δv_9 수열의 T_δ를 계산한다.

Algorithm 4. Construction the sequence of the table T_δ (Δv_9 sequence) of GFN-I-4

Input: X, Y, Z, W and F_4, F_5, F_6, F_7, F_8
Output: T_δ (Δv_9 sequence)

1. for $i = 0, 1, ..., 2^{n/4}-1$ do
2. store $(i, F_4(i) \oplus F_4(i \oplus X))$ in T_4 indexed by $F_4(i) \oplus F_4(i \oplus X)$
3. store $(i, F_8(i) \oplus F_8(i \oplus X))$ in T_8 indexed by $F_4(i) \oplus F_4(i \oplus X)$
4. for $i = 0, 1, ..., 2^{n/4}-1$ do
5. store $(i, F_5(i))$ in a t5 indexed by $F_5(i)$
6. store $(i, F_6(i))$ in a t6 indexed by $F_6(i)$
7. store $(i, F_7(i))$ in a t7 indexed by $F_7(i)$
8. for $i = 0, 1, ..., 2^{n/4}-1$ do
9. compute $F_5(i) \oplus Y$
10. find j s.t. $F_5(j) = F_5(i) \oplus Y$ in t5
11. store $(i, i \oplus j)$ in T_5 indexed by $i \oplus j$
12. for $i = 0, 1, ..., 2^{n/4}-1$ do
13. compute $F_6(i) \oplus Z$
14. find j s.t. $F_6(j) = F_6(i) \oplus Z$ in t6
15. store $(i, i \oplus j)$ in T_6 indexed by $i \oplus j$
16. for $i = 0, 1, ..., 2^{n/4}-1$ do
17. compute $F_7(i) \oplus (W \oplus X)$
18. find j s.t. $F_7(j) = F_7(i) \oplus (W \oplus X)$ in t7
19. store $(i, i \oplus j)$ in T_7 indexed by $i \oplus j$
20. for $i = 0, 1, ..., 2^{n/4}-1$ do
21. obtain internal state values $F_4^I \sim F_8^I$ by looking up $T_4 \sim T_8$
22. for all b active bits of $b-\delta-set$ do
23. modify Δv_0 and compute Δv_9
24. compute the sequence of Δv_9 and add it to T_δ

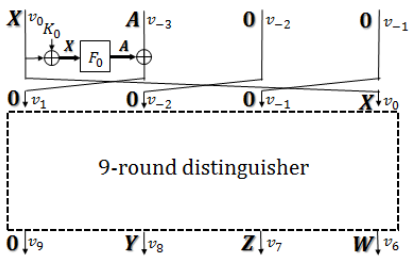

Fig. 8. 10-Round GFN-I-4 Attack

하나의 고정된 X, Y, Z, W에 대하여 Algorithm 4를 계산하는데 필요한 계산 복잡도 및 메모리 복잡도는 $2^{n/4}$이다. 출력 차분 $(0, Y, Z, W)$의 개수를 2^m개를 선택한다면 Precomputation Phase에 필요한 복잡도는 $2^{n/4+m}$이 된다.

Online Phase에서는 우선 Algorithm 5를 이용하여 9-라운드 암호화된 암호문의 차분이 $(0, Y, Z, W)$가 되는 평문-암호문 쌍을 찾아낸다. 이 단계에서는 고정된 (v_0, v_{-2}, v_{-1})에 대한 평문쌍 $2^{n/2}$ 중 2^m개의 $(0, Y, Z, W)$를 만족하는 것의 개수는 평균적으로 $2^{-n/2+m}$이다. (v_0, v_{-2}, v_{-1})를 $2^{3n/4-m}$개 선택하면 가능한 쌍은 $2^{n/4}$개가 된다.

Online Phase 후 남은 $2^{n/4}$개의 쌍 중 1-라운드 후 차분이 $(0, 0, 0, X)$를 만족하는 K_0는 평균적으로 1개가 존재한다. 이를 만족하는 평문쌍과 키에 대해 v_0 값에서 $b-\delta-set$를 계산하고 키를 이용하여 v_{-3}를 계산하여 평문 입력 $b-\delta-set$에 대한 Δv_9를 계산한다. 이 계산된 Δv_9 값이 미리 계산된 테이블 T_δ에 매치한다. 만약 T_δ와 매치가 되면 K_0

Algorithm 5. Data collection phase of the 6-round attack of GFN-I-4

Input: T_δ
Output: Possible (P, C) pairs such that satisfies 9-round distinguisher

1. Choose 2^m differences $(0, Y, Z, W)$
2. Choose X such that $X \neq W$
3. for $2^{3n/4-m}$ different values of (v_1, v_{-2}, v_{-1}) do
4. for all $2^{n/4}$ choices of v_{-3} do
5. query $(v_0, v_{-3}, v_{-2}, v_{-1})$ and store it in L_0
6. query $(v_0 \oplus X, v_{-3}, v_{-2}, v_{-1})$ and store in L_1
7. pick up the elements of $L_0 \times L_1$ whose ciphertext mathch in the 2^m $(0, Y, Z, W)$ choices

의 값이 복구된다.

데이터 복잡도는 대략 2^{n-m}개의 선택평문, 시간 복잡도는 $2^{n/4+m}$이고, 메모리 복잡도는 $2^{n/4+m}$이다. m을 조절하면 필요한 데이터와 시간 및 메모리 복잡도 사이의 trade-off를 할 수 있다. 만약 $m=n/4$ 이면 이때 데이터 복잡도 대략 $2^{3n/4}$, 시간 복잡도 $2^{n/2}$, 메모리 복잡도 $2^{n/2}$이 된다. 만약 $m=3n/8$ 이면 데이터, 시간, 메모리 복잡도 모두 $2^{5n/8}$이 된다.

V. GFN-II-4 구조에 대한 중간 일치 공격

본 장에서는 4 branch를 갖는 일반화된 Feistel 구조의 Type II 구조인 GFN-II-4에 대한 중간 일치 공격을 살펴본다.

5.1 GFN-II-4에 대한 5-라운드 distinguisher

GFN-II-4 공격 방법은 앞 장에서 살펴본 GFN-I-4에 대한 방법과 거의 유사하므로 간단하게 distinguisher만 소개한다. Fig. 9는 GFN-II-4에 대한 5-라운드 distinguisher이다.

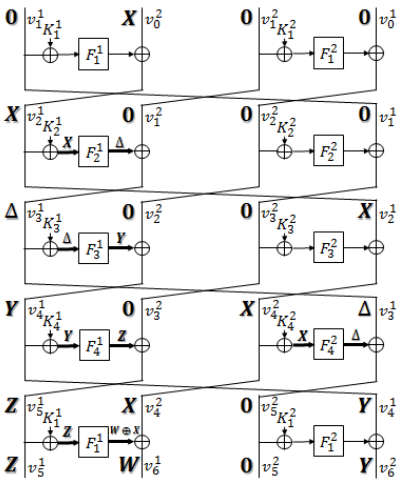

Fig. 9. 5-Round Distinguisher of GFN-II-4

5.2 GFN-II-4에 대한 6-라운드 공격

Fig. 5의 5-라운드 distinguisher를 이용하면 6-라운드에 대한 공격을 GFN-I-4와 거의 유사하고 복잡도 역시 GFN-I-4과 같다.

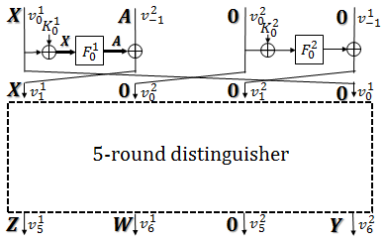

Fig. 10. 6-Round GFN-II-4 Attack

데이터, 시간, 메모리 복잡도 역시 GFN-I-4과 동일하다.

VI. 결 론

본 논문에서는 4-branch GFN 구조에 대한 중간 일치 공격에 대한 새로운 distinguisher와 이를 이용한 새로운 공격 방법을 소개하였다. 이를 요약하면 Table.1과 같다.

향후 GFN의 다른 구조 뿐 아니라 다양한 구조에 대한 중간 일치 공격의 적용과 이 중간 일치 공격을 향상시키기 위한 연구에 대한 보다 심층적인 연구가 필요할 것이다.

Table 1. Complexity of Generic MITM Attacks

	# of rounds		complexity			ref.
	disting uisher	attack	time	space	data	
Feistel	5	6	$2^{\frac{3n}{4}}$	$2^{\frac{3n}{4}}$	$2^{\frac{3n}{4}}$	[11]
GFN-I-4	9	10	$2^{\frac{5n}{8}}$	$2^{\frac{5n}{8}}$	$2^{\frac{5n}{8}}$	Ours
GFN-II-4	5	6	$2^{\frac{5n}{8}}$	$2^{\frac{5n}{8}}$	$2^{\frac{5n}{8}}$	Ours

References

[1] H. Feistel, W. Notz, and J. Smith, "Some cryptographic techniques for machine-to-machine data communications," *Proc of the IEEE*, vol. 63, pp. 1545-1554, 1975.

[2] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM Journal Computing*, vol. 17, no. 2, pp. 373-386, 1988.

〔3〕 M. Naor and O. Reingold, "On the construction of pseudorandom permutations : Luby-Rackoff revisited," *Journal of Cryptology*, vol. 12, no. 1, pp. 29-66, 1999.
〔4〕 K. Nyberg, "Generalized Feistel Networks," ASIACRYPT*'96*, LNCS 1163, pp. 491-104, 1996.
〔5〕 Y. Zheng, T. Matsumoto, and H, Imai, "On the construction of block ciphers provably secure and not relying on any unproved hypotheses," *CRYTPO'89*, LNCS 435, pp. 461-480, 1989.
〔6〕 E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystem," *Journal of Cryptology*, vol. 4, no. 1, pp. 3-72, 1991.
〔7〕 K. Nyberg and L. R. Knudsen, "Provable security against differential cryptanalysis," *Journal of Cryptology* , vol. 8, no. 1, pp. 27-37, 1995.
〔8〕 J. Daemen, L. R. Knudsen, and V. Rijmen, "The block cipher square," *FSE'97*, LNCS 1267, pp. 149-165, 1997.
〔9〕 J. Patarin, "Generic attacks on Feistel schemes," *ASIACRYPT'01*, LNCS 2248, pp. 222-238, 2001.
〔10〕 I. Dinur, O. Dunkelmann, N. Keller, and A. Shamir, "New attacks on Feistel structures with improved memory complexities," *CRYTPO'15*, LNCS 9215, pp. 433-454, 2015.
〔11〕 J. Guo, J. Jean, I. Nikolic, and Y. Sasaki, "Extended meet-in-the-middle attacks on some Feistel constructions," *Designs, Codes and Cryptography*, vol. 80, no. 3, pp. 587-618, 2016.
〔12〕 J. Guo, J. Jean, I. Nikolic, and Y. Sasaki, "Meet-in-the-middle attacks on generic Feistel constructions," *ASIACRYPT'14*, LNCS 8873, pp. 439-457, 2014.
〔13〕 V. Hoang and P. Rogaway, "On Generalized Feistel networks," *CRYTPO'10*, LNCS 6223, pp. 613-630, 2010.
〔14〕 V. Nachef, E.Volte, and J. Patarin, "Differential attacks on generalized Feistel schemes," *CANS'13*, LNCS 8257, pp. 1-19, 2013.
〔15〕 A. Bogdanov and K,Shibutani, "Generalized Feistel networks revisted," *Designs, Codes and Cryptography*, vol. 66, no. 1, pp. 75-97, 2013.

〈저자소개〉

성 재 철 (Jaechul Sung) 종신회원
1997년 8월: 고려대학교 수학과 학사
1999년 8월: 고려대학교 수학과 석사
2002년 8월: 고려대학교 수학과 박사
2002년 8월~2004년 1월: 한국정보보호진흥원 선임연구원
2004년 2월~현재: 서울시립대학교 수학과 교수
〈관심분야〉 암호 알고리즘 설계 및 분석