Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.5, Oct. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.5.795

SP F-함수를 갖는 4-브랜치 GFN-2 구조에 대한 기지키 공격*

홍 득 조†‡
전북대학교 (교수)

Known-Key Attacks on 4-Branch GFN-2 Structures with SP F-Functions*

Deukjo Hong†‡
Jeonbuk National University (Professor)

요 약

본 논문에서는 SP 구조의 F-함수를 가진 4-브랜치 GFN-2 구조에 대한 기지키 구별 공격(Known-Key Distinguishing Attack) 및 부분 충돌 공격(Partial-Collision Attack)을 연구한다. 첫 번째로, 이 구조에 대해 기지키 구별 공격이 15 라운드까지 가능함이 밝혀진다. 두 번째로, 마지막 라운드에 셔플 연산이 있는 경우, 부분 충돌 공격이 14 라운드까지 가능함이 밝혀진다. 마지막으로, 마지막 라운드에 셔플 연산이 없는 경우, 부분 충돌 공격이 15 라운드까지 가능함이 밝혀진다.

ABSTRACT

In this paper, we study known-key distinguishing and partial-collision attacks on GFN-2 structures with SP F-functions and various block lengths. Firstly, we show the known-key distinguishing attack is possible up to 15 rounds. Secondly, for the case that the last round function has the shuffle operation, we show that the partial-collision attack is possible up to 14 rounds. Finally, for the case that the last round function has no shuffle operation, we show that the partial-collision attacks are possible up to 11 rounds.

Keywords: Generalized Feistel Network, Known-Key Attack, Partial-Collision Attack

I. 서 론

기지키 공격(Known-Key Attack)의 개념은 2007년에 Rijmen과 Knudsen이 처음으로 제시하였다[1]. 이 공격은 공격자가 블록암호의 키 입력값을 알고 있을 때, 어떤 평문-암호문 블록 집합에 대해 균등(Uniform) 확률 보다 높은 확률로 성립할 수 있는 기지키 구별자(Known-Key Distinguisher)를 이용한다. 2011년에 Sasaki와 Yasuda는 F-함수가 암호학적으로 강한 S-box와 MDS 행렬로 구성된 Feistel 네트워크에 대하여 리바운드 기법[9]을 이용해 기지키 구별자들을 구성하였고, 이를 사용하여 블록암호의 해시모드에 대한 충돌 공격(Collision Attack) 또는 부분 충돌 공격(Partial-Collision Attack)이 가능함을 보였다[2] 부분 충돌(Partial Collision 또는 Near Collision)이란, 해시값의 일부에서만 충돌이 발생하는 현상을 의미함). 이 후, 그들의 결과는

Received(09. 15. 2020), Modified(10. 08. 2020), Accepted(10. 08. 2020)
* This work was supported by Institute of Information communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (No. B0722-16-0006, Cross layer design of cryptography and physical layer security for IoT networks)
† 주저자, deukjo.hong@jbnu.ac.kr
‡ 교신저자, deukjo.hong@jbnu.ac.kr(Corresponding author)

Feistel 네트워크의 변종 구조들에 응용되었다[3,4,5].

Feistel 구조는 블록암호 DES[6]나 SEED[8]에 적용된 것으로 잘 알려져있으며, 안정적이고 효율적인 블록암호 설계를 위한 주요 연구 대상이 되어왔다. Feistel 변종 구조 중 하나인 GFN-2(Type-2 Generalized Feistel Network)는 같은 블록 길이를 갖는 Feistel 구조에 비해 라운드 함수를 경량으로 설계할 수 있고, 다른 타입에 비해 균형감 있는 암호화 연산을 수행하기 때문에 Feistel 네트워크의 대안 중 하나로서 많이 연구되어 왔고 블록암호 개발 시 자주 고려되는 구조이다. 실제로 블록암호 CLEFIA[7]에 적용되었으며, S-box가 사용되지 않고 ARX 연산으로 구성된 블록암호 HIGHT[10]에도 매우 유사한 구조가 설계에 응용되었다. 그러므로, 블록암호의 다양한 응용성을 고려할 때, GFN-2 구조의 안전성에 대한 연구는 중요하다.

본 논문에서는 매개변수 (t,a,b)를 이용하여 SP 구조의 F-함수를 가진 GFN-2 구조를 정의한다. t는 브랜치의 개수, a는 F-함수를 구성하는 S-box의 개수, b는 S-box의 입출력 크기이다. 바이트의 길이는 b 비트, 워드의 길이는 ab 비트로 정의한다. 매개변수 값 (t,a,b)를 갖는 GFN-2 구조 블록암호의 블록 길이는 abt 비트이다. 본 논문에서는 (a,b)의 값을 (4,4), (4,8), (8,4), (8,8)로, t의 값을 4로 제한하는데, 이것들은 실제 블록암호 또는 해시함수 설계에서 주로 사용되거나 고려되는 것들이다.

2012년, [3]에서는 GFN-1, GFN-2, GFN-3에 대한 기지키 공격을 연구하였는데, 모두 SP 구조 F-함수와 4개의 브랜치로 구성되어 있으며, 마지막 라운드에 셔플 연산이 없다고 가정하였다. 이러한 가정에서 GFN-2에 대한 13-라운드 기지키 구별 공격을 제시하였다. 또한, Matyas-Meyer-Oseas(MMO) 또는 Miyaguchi-Preneel(MP) 해시 운영 모드가 적용되었다는 가정에서 9 라운드에 대해 1-워드 부분 충돌 공격 및 2-워드 부분 충돌 공격을 제시하였다. 2015년에는 Dong 등이 동일한 가정이 적용된 4-브랜치 GFN-2 구조에 대해 15-라운드 기지키 구별 공격 및 11-라운드 부분 충돌 공격을 발표했으나, 해당 연구 결과는 공격 복잡도 계산에 오류가 있다. 이것에 대해서는 본 논문의 2.4절에서 설명된다.

본 논문에서는 SP F-함수로 구성되는 4-브랜치 GFN-2 구조에 대하여 개선된 기지키 구별 공격 및 부분 충돌 공격을 제시한다. 인바운드 구조를 구성하는 일반적인 방법을 설명하고, 해당 방법을 이용하여 5-라운드 인바운드 구조를 구성하였다. 이 5-라운드 인바운드 구조를 기반으로, 공격 복잡도를 정밀하게 계산함으로써 아래와 같은 연구 결과들을 도출하였다.

- 15-라운드 기지키 구별 공격 제시
- 마지막 셔플 연산이 없을 때 11-라운드 3-워드 부분 충돌 공격이 가능하며, a = 8 이면 15-라운드 1-워드 부분 충돌 공격이 가능함을 밝힘
- 마지막 셔플 연산이 있을 때 10-라운드 3-워드 부분 충돌 공격이 가능하며, a = 8 이면 14-라운드 1-워드 부분 충돌 공격이 가능함을 밝힘

논문의 구성은 다음과 같다. 2절에서는 기존의 GFN-2, MMO와 MP 해시 운영 모드의 동작 방식, F-함수에 대한 인바운드 구조, 기존 연구 결과들을 설명한다. 3절에서는 GFN-2의 인바운드 구조에 필요한 차분 경로를 구성하는 방법을 설명한다. 4절에서는 4-브랜치 GFN-2에 대한 인바운드 구조, 기지키 구별자, 부분 충돌 공격들을 제시한다.

II. 배경 지식

2.1 SP 구조 F-함수로 구성된 GFN-2 구조

S: $\{0,1\}^b \rightarrow \{0,1\}^b$ 를 b비트의 입력과 출력을 갖는 비선형 전단사(Bijective) S-box라고 하자. 어떤 b비트 입력값 X에 대해 S-box S가 b비트 출력값 Y를 가질 때 Y = S(X)로 표시한다. 선형함수P: $(\{0,1\}^b)^a \rightarrow (\{0,1\}^b)^a$ 를 유한체 $GF(2^b)$ 상의 a×a MDS 행렬곱으로 정의되는 함수라고 하자. 어떤 입력 벡터 (X[0], X[1], ..., X[a-1])에 대해 P가 출력 벡터 (Y[0], Y[1], ..., Y[a-1])을 가질 때 (Y[0], Y[1], ..., Y[a-1]) = P(X[0], X[1], ..., X[a-1])로 표시한다. 이러한 S와 P에 대해, F-함수 F: $(\{0,1\}^b)^a \times (\{0,1\}^b)^a \rightarrow (\{0,1\}^b)^a$를 다음과 같이 정의할 수 있다: 입력값 X = (X[0], X[1], ..., X[a-1]) $\in (\{0,1\}^b)^a$과 RK = (RK[0], RK[1], ..., RK[a-1]) $\in (\{0,1\}^b)^a$에 대해, Y = F(X,RK) = P(S(X[0]⊕RK[0]), S(X[1]⊕RK[1]), ..., S(X[a-1]⊕RK[a-1])). 여기서, ⊕는 XOR (eXclusive OR) 연산을 의미한다. Fig.1에서 a = 4인 경우의 F-함수 연산 과정을 그림으로 이해할

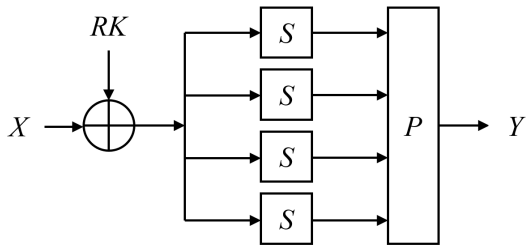

Fig. 1. Structure of F-function with a = 4

수 있다.

t를 어떤 짝수라고 하고, r을 어떤 양의 정수라고 하자. RK = ($RK_{0,0}$, …, $RK_{0,t/2-1}$, $RK_{1,0}$, …, $RK_{r-1,t/2-1}$)를 비밀키 K에 대해 블록암호의 키 스케줄 알고리즘이 생성하는 부분키들(Subkeys)의 벡터이며, 각 $RK_{i,j}$ = ($RK_{i,j}[0]$, $RK_{i,j}[1]$, …, $RK_{i,j}[a-1]$)는 길이가 a인 벡터이고, 각 $RK_{i,j}[u]$는 b비트 값이라고 하자. 셔플 연산 σ는 σ = (σ(0), σ(1), …, σ(t-1)) = (t-1, 0, …, t-2)로 정의된다. n = abt 비트 입력 블록 ($X_{0,0}$, $X_{0,1}$, …, $X_{0,t-1}$)에 대해, t개의 브랜치를 갖는 r-라운드 GFN-2 구조의 블록암호는 다음과 같이 출력 블록 ($X_{r,0}$, $X_{r,1}$, …, $X_{r,t-1}$)을 계산한다:

for i = 0, … r-1 **do**:
  for j = 0, …, t-1 **do**:
    if j is even:
      $Y_j = X_{i,j}$
    else:
      $Y_j = X_{i,j} \oplus F(X_{i,j-1}, RK_{i,(j-1)/2})$
  for j = 0, …, t-1 **do**:
    $X_{(i+1),\sigma(j)} = Y_j$

위의 의사코드에서 변수 i는 라운드 번호를 의미한다. Fig. 2는 GFN-2의 i번째 라운드를 보여준다. 본 논문에서는 키가 어떤 값으로 고정되어 있고 알려져있다고 가정하며, 공격의 설명에 영향을 미치지 않으므로 글의 간결성을 위해 부분키에 관한 기술을 생략한다. 예를 들어, $F(X_{i,j-1},RK_{i,(j-1)/2})$를 간단히 $F(X_{i,j-1})$로 표시한다.

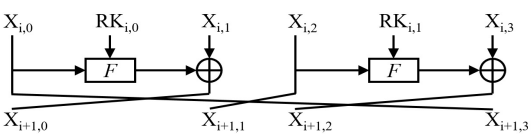

Fig. 2. i-th round of GFN-2 with t = 4

2.2 F-함수의 인바운드 구조 ISF

차분(Difference)은 특정한 위치에서 서로 다른 두 값의 XOR이며, 차분 경로(Differential Trail)는 블록암호의 어떤 위치에서 또 다른 위치로 차분 값이 변화되는 과정을 기술한 것이다.

인바운드 구조(Inbound Structure)는 리바운드 공격에서 핵심적인 부분이다[9]. 이것은 블록암호의 특정 부분에서 설정된 어떤 차분 경로들을 만족시키는 입력쌍들의 집합이다. 인바운드 구조를 설명하기 위해 ab-비트 워드의 차분에 대한 표현 몇 가지를 다음과 같이 정의할 필요가 있다.

- 0: 모든 바이트가 0 차분을 가짐
- Δ_1: 한 바이트만 0이 아니고 나머지 바이트는 모두 0인 차분
- $\Delta_{P(1)}$: Δ_1 형태의 차분이 P 함수를 통과한 이후의 차분. 즉, $P(\Delta_1) = \Delta_{P(1)}$
- $\Delta_{P^{-1}(1)}$: Δ_1 형태의 차분이 P 함수의 역함수 P^{-1}를 통과한 이후의 차분. 즉, $P^{-1}(\Delta_1) = \Delta_{P^{-1}(1)}$

여기서 부분키들은 모두 어떤 값들로 고정되어있으며, S-box의 차분분포표(Difference Distribution Table)에서 0인 성분과 0이 아닌 성분의 비율이 거의 같다고 가정한다. 2.1 절에서 정의된 F-함수에 대해, 입력 차분은 $\Delta_{P(1)}$이고 출력 차분은 Δ_1로 설정되어있다고 가정하자. 그러면 모든 가능한 $\Delta_{P(1)}$와 Δ_1의 각 조합에 대해, F-함수의 모든 S-box는 0이 아닌 입력 차분과 출력 차분을 갖게 된다. 그러나, 차분분포표에 대한 가정에 의해 각 S-box의 입력 차분과 출력 차분의 조합은 확률 1/2로 유효하다. S-box의 유효한 입력 차분과 출력 차분 조합을 만족하는 입력쌍은 평균적으로 1개 존재한다. 따라서, 이와 같은 F-함수의 인바운드 구조 ISF(Inbound Structure of F-function)가 포함할 수 있는 입력쌍의 개수는 $(2^b-1)^2\times(2^{-1})^a\times 2^a = (2^b-1)^2$이다.

예를 들어, Fig.3에서와 같이 a = 4인 경우의 ISF를 살펴보자. F 함수의 출력 차분이 첫 번째 바이트에서만 0이 아니고 나머지 바이트에서는 모두 0이라고 하자. 즉, F 함수의 출력 차분은 Δ_1의 형태를 갖는다. F 함수의 입력 차분은 $\Delta_{P(1)} = P(\Delta_1)$이라고 가정한다. 본질적으로는 F 함수의 입력 차분과

출력 차분에서 한 바이트만 0이 아니고 나머지 바이트는 모두 0인 형태가 고려되는 것이다. 그러한 F 함수의 입력 차분과 출력 차분의 조합은 모두 $(2^b-1)^2 \times 2^{-4}$ 개가 가능하다. 첫 번째 S-box의 입력쌍을 $(x_{0,0}, x_{0,1})$, 그것에 대응되는 출력쌍을 $(y_{0,0}, y_{0,1})$이라고 하자. 그러면 첫 번째 S-box의 입력 차분은 $x_{0,0} \oplus x_{0,1} = \alpha_0$ 이고, 출력 차분은 $y_{0,0} \oplus y_{0,1} = \beta_0$ 이다. 마찬가지 방법으로 두 번째, 세 번째, 네 번째 S-box의 입력쌍과 출력쌍을 $\{(x_{1,0}, x_{1,1}), (y_{1,0}, y_{1,1})\}$, $\{(x_{2,0}, x_{2,1}), (y_{2,0}, y_{2,1})\}$, $\{(x_{3,0}, x_{3,1}), (y_{3,0}, y_{3,1})\}$ 이라고 하자. 그러므로, 네 S-box의 입력 차분이 α_0, α_1, α_2, α_3이고 출력 차분이 β_0, β_1, β_2, β_3일 때,

$$x_{i,0} \oplus x_{i,1} = \alpha_i \text{ and } y_{i,0} \oplus y_{i,1} = \beta_i \quad \text{for } i = 0, 1, 2, 3$$

이다. 그러면, 가능한 F-함수의 입력쌍의 조합은 2^4 가지이다. 그런데, F-함수의 $(\Delta_{P(1)}, \Delta_1)$ 형태의 입력차분과 출력차분의 약 $(2^b-1)^2 \times 2^{-4}$ 개 조합이 가능하므로, ISF는 약 $(2^b-1)^2$ 개의 입력쌍을 포함하게 된다.

S-box의 차분분포표는 미리 주어지며, 각 입력 차분과 출력 차분의 조합에 대해 어떤 입력 쌍이 가능한지도 포함한다고 가정하자. 그러면, ISF 생성에 소요되는 계산복잡도는 차분조합 유효성 대조 작업이 거의 대부분을 차지하고 F-함수는 a개의 S-box로 구성되므로 약 $a \times 2^{2b}$ 번의 테이블 참조 즉, 2^{2b} 번의 F-함수 계산량과 거의 같다.

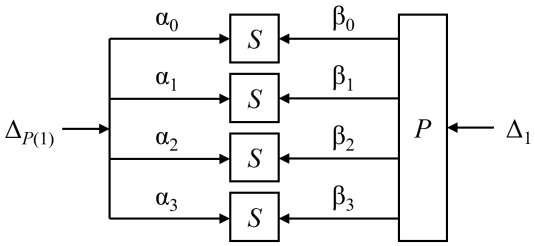

Fig. 3. Differences in the inbound structure of F-function with a = 4

2.3 Matyas-Meyer-Oseas 모드와 Miyaguchi-Preneel 모드

Matyas-Meyer-Oseas(MMO) 모드와 Miyaguchi-Preneel(MP) 모드는 블록암호를 1회 호출하여 Merkle-Damgard 해쉬 함수의 압축함수를 구성하는 안전한 12가지 PGV 모드에 속한다[12]. Fig.4에서 보듯이 이 두 모드는 공격자가 컨트롤할 수 없지만 알 수 있는 입력 연쇄변수(Input chaining variable) 값이 블록암호의 키가 되고 공격자가 컨트롤할 수 있는 메시지 블록(Message block) 값이 블록암호의 평문 블록(Plaintext block)이 되며, 블록암호의 암호문 블록(Ciphertext block)이 평문 블록 또는 키(Key)와 XOR되어 압축함수의 출력 연쇄변수(Output chaining variable) 값을 만들어낸다는 특징이 있다. 본 논문에서 설명하는 모든 부분 충돌 공격은 GFN-2 구조의 블록암호가 MMO 또는 MP 모드와 결합된 상황을 가정한다.

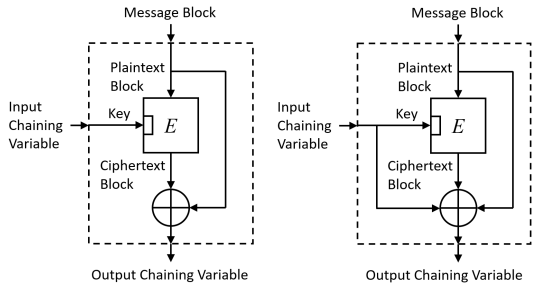

Fig. 4. Matyas-Meyer-Oseas(MMO) mode (Left) and Miyaguchi-Preneel(MP) mode (Right)

2.4 Dong 등의 공격

2015년에 Dong 등이 본 논문의 연구 대상인, SP구조 F-함수를 갖는 4-브랜치 GFN-2에 대한 15-라운드 기지키 구별자와 11-라운드에 대해 부분 충돌 공격을 제시하였다[11]. 또한, 독립적으로 연구된 것이지만 본 논문과 동일한 5-라운드 인바운드 구조를 구성하였다.

본 논문에서는 [11] 보다 GFN-2에 대한 더 다양한 공격 결과를 제시한다. [11]에서는 마지막 라운드 함수에 셔플 연산이 없는 경우에 대해서만 연구하였는데, 본 논문에서는 마지막 라운드에 셔플 연산이 없는 경우 F 함수가 8개의 S-box를 사용할 때에는 부분 충돌 공격도 15 라운드까지 가능함을 보인다. 또한, 본 논문에서는 마지막 라운드에 셔플 연산이 있는 경우에 대해서도 연구하였다. 해시 함수는 복호화 과정이 필요하지 않기 때문에 마지막 라운드 함수가 다른 라운드 함수와 동일하다는 점에서 본 논

문의 결과가 해시 함수 설계에 더 참고가 될 만하다.

본 논문에서는 [11]에 비해 공격 복잡도를 매우 정밀하게 계산하였기에 더 많은 결과를 제시할 수 있었던 것으로 보인다. 게다가, [11]에서는 2.2에서 설명되는 F-함수의 인바운드 구조를 구성하는 과정을 2^{2b}가 아닌 2^{b}로 계산하였고, 4절에서 설명되는 인바운드 구조의 구성 과정에서 파생되는 많은 복잡도를 무시함으로써 잘못 계산된 공격 복잡도를 제시하고 있다 ([11]의 Table 3). 이것은 공격의 가능성이 왜곡될 수 있는 중요한 문제이므로 실수 없이 정확하게 계산될 필요가 있다.

III. GFN-2의 인바운드 구조 ISG2에 대한 차분 경로 구성 방법

4-브랜치 GFN-2의 각 라운드 입력 또는 출력은 네 개의 ab-비트 워드들로 구성되는 벡터이다. 이 벡터의 차분은 각 구성 요소에서 0, Δ_1, $\Delta_{P(1)}$, ? 중 하나의 형태를 갖게 된다. 이 논문에서는 입력 차분과 출력 차분이 가능한 한 최소의 Δ_1 차분을 갖고 최대의 0 차분을 갖는 GFN-2의 인바운드 구조 ISG2 (Inbound Structure of GFN-2)만 다루는데, 그러한 차분 경로가 ISG2의 입력 차분으로부터 역방향으로, 출력 차분으로부터 정방향으로 차분 전파를 통해 가장 긴 구별 공격 및 부분 충돌 공격들을 만들어내기 때문이다. 그러한 ISG2에 적용될 수 있는 차분 경로를 구성하는 일반적인 방법은 다음과 같다.

① 구성하려는 ISG2의 라운드 수 R을 설정한다.
② ISF의 개수와 그것들을 적용할 F 함수 위치를 랜덤하게 선택하고, 각 선택된 F 함수의 입력 차분과 출력 차분을 각각 $\Delta_{P(1)}$과 Δ_1로 설정한다.
③ ISF들로부터의 전파되는 0이 아닌 차분들을 0, Δ_1, $\Delta_{P(1)}$ 차분들을 이용하여 정방향과 역방향으로 최소화하도록 조정한다.
④ ISG2의 입력 차분과 출력 차분이 최소의 Δ_1과 최대의 0을 갖는지 확인한다. 만약 그렇다면, 구성된 차분 경로를 출력한다; 그렇지 않다면, 단계 ②에서 다시 시작한다.

Δ_1, $P^{-1}(\Delta_{P(1)})$ 형태에서 고려되는 0이 아닌 차분 바이트의 위치는 동일한 것으로 가정한다. 또한, F 함수에 입력되는 모든 서브키 값들은 알려져있고 고정되어 있다고 가정한다. 그래서 모든 설명에서 서브키 및 서브키 XOR의 표기를 생략한다.

Table 1. Hexadecimal representation for two consecutive words

	0	Δ_1	$\Delta_{P(1)}$	?
0	0x0	0x1	0x2	0x3
Δ_1	0x4	0x5	0x6	0x7
$\Delta_{P(1)}$	0x8	0x9	0xA	0xB
?	0xC	0xD	0xE	0xF

본 논문에서 한 워드(ab 비트) 차분의 형태를 표시하기 위해 사용하는 기호는 '0', 'Δ_1', '$\Delta_{P(1)}$', '?' 네 가지 뿐이다. '?'는 차분이 어떤 형태인지 알 수 없음을 의미한다. 더 간결한 표현을 위해 이 기호들을 각각 두 비트의 이진수 00_2, 01_2, 10_2, 11_2에 대응시킨다. 그러면 연속된 두 워드의 차분 형태는 Table 1과 같이 한 자리 16진수에 대응시킬 수 있다.

IV. 4-브랜치 GFN-2에 대한 공격

4.1 5-라운드 인바운드 구조

4-브랜치 GFN-2에 대해 발견된 ISG2의 라운드 수를 R = 5로 설정하고, 3절에서 설명된 차분 경로 구성 방법을 적용하면 Fig.5와 같은 차분 경로를 구할 수 있다. 이것은 Table 1에 의해 16진수 벡터 (0x40, 0x81, 0x46, 0x91, 0x06, 0x10)로 표현될 수 있다. ISG2의 입력값은 $X_0 = (X_{0,0}, X_{0,1}, X_{0,2}, X_{0,3})$이고, i번째 라운드의 출력값은 $X_{i+1} = (X_{i+1,0}, X_{i+1,1}, X_{i+1,2}, X_{i+1,3})$이다 (i = 0, 1, 2, 3, 4). 여기서, ISF가 고려되는 위치는 $X_{1,0}$와 $X_{3,0}$을 입력으로 갖는 F 함수이다. $X_{i,j}$의 차분을 $\Delta X_{i,j}$라고 하자. 다음과 같은 과정을 통해 Fig.5의 차분 경로를 만족하는 쌍들을 찾아 ISG2를 구성할 수 있다.

① $X_{1,0}$을 입력으로 갖는 F 함수에 대하여 ISF를 고려한다. 이 F 함수에 대해 $\Delta_{P(1)}$ 형태의 입력 차분과 Δ_1 형태의 출력 차분을 만족하는 $(2^b-1)^2 \cong 2^{2b}$ 개의 쌍이 설정된다. 이것을 ISF-1로 표시한다.
② 단계 ①과는 독립적으로 $X_{3,0}$을 입력으로 갖는 F 함수에 대하여 ISF를 고려한다. 이 F 함수에 대해 $\Delta_{P(1)}$ 형태의 입력 차분과 Δ_1 형태의 출력 차

분을 만족하는 $(2^b-1)^2 \cong 2^{2b}$ 개의 쌍이 설정된다. 이것을 ISF-2로 표시한다.

③ $X_{0,2}$를 랜덤하게 선택하여 $F(X_{0,2})$를 계산하고, ISF-1의 모든 $F(X_{1,0})$ 값에 대하여 $X_{2,0}$ 및 $F(X_{2,0})$을 계산한다.

④ $X_{4,0}(=X_{5,3})$을 랜덤하게 선택하여 $F(X_{4,0})$을 계산하고, ISF-2의 모든 $F(X_{3,0})$ 값에 대하여 $X_{2,2}$ 및 $F(X_{2,2})$를 계산한다.

⑤ ISF-1의 쌍 (x_1,x_2)로부터 계산되는 $\Delta F(X_{2,0})$ 및 $\Delta X_{1,0}$에 대해, 그리고 ISF-2의 쌍 (y_1,y_2)로부터 계산되는 $\Delta X_{3,0}$ 및 $\Delta F(X_{2,2})$에 대해, $\Delta F(X_{2,0}) = \Delta X_{3,0}$ 이고 $\Delta X_{1,0} = \Delta F(X_{2,2})$이면 그 쌍들을 $\{(x_1,y_1), (x_2,y_2)\}$와 같이 연결한다. 이렇게 연결된 쌍들을 새로운 테이블 ISF-(1,2)에 저장한다. 평균적으로, ISF-(1,2)는 2^{2b}개 쌍들을 포함한다.

⑥ ISF-(1,2)에 포함된 모든 쌍들에 대해 $F(X_{1,2})$와 $F(X_{0,0})$을 계산하여, $\Delta F(X_{0,0}) \neq \Delta X_{1,0}$인 쌍은 버린다. 평균적으로 2^b개의 쌍이 살아남는다.

⑦ 살아남은 모든 쌍들에 대해 $F(X_{3,2})$와 $F(X_{4,2})$를 계산하여, $\Delta F(X_{4,2}) \neq \Delta X_{3,0}$인 쌍은 버린다. 평균적으로 1개의 쌍이 살아남는다.

⑧ 살아남은 쌍들에 대해 $X_{0,1}$, $X_{0,3}$, $X_{5,0}$, $X_{5,2}$를 계산하고 ISG2에 포함시킨다.

고정된 $X_{0,1}$, $X_{5,3}$ 값에 대해 위 5-라운드 ISG2는 평균적으로 1개의 쌍을 갖는다. 5-라운드 ISG2의 구성에 소요되는 계산 복잡도 T는 9×2^{2b} 번의 F-함수 연산으로 추정된다. 이것을 $T \cong 9\cdot2^{2b}F$ 로 표시하자. 이 값의 근거는 다음과 같다.

- 단계 ①, ②에서 두 개의 독립적인 ISF 구성. 그러나, 본질적으로 동일한 집합이 적용되므로 2.2절에서 설명되었듯이, 2^{2b} 번의 F-함수 연산
- 단계 ③ $\cong 2^{2b+1}F$ ($\because F(X_{2,0})$ 2^{2b+1} 번 계산)
- 단계 ④ $\cong 2^{2b+1}F$ ($\because F(X_{2,2})$ 2^{2b+1} 번 계산)
- 단계 ⑥ $\cong 2^{2b+2}F$ ($\because F(X_{1,2})$ 2^{2b+1} 번 계산, $F(X_{0,0})$ 2^{2b+1} 번 계산)
- 단계 ⑦ $\cong 2^{b+2}F$ ($\because F(X_{3,2})$ 2^{b+1} 번 계산, $F(X_{4,2})$ 2^{b+1} 번 계산)

즉, Fig.5를 만족시키는 한 개의 쌍을 얻기 위해 T 만큼의 계산 복잡도가 소요된다. 그러므로, $(X_{0,1}, X_{5,3})$를 N개 선택한다면, 이 5-라운드 ISG2는 N개의 쌍을 포함하며, 이것을 갖기 위한 계산 복잡도는 NT가 된다.

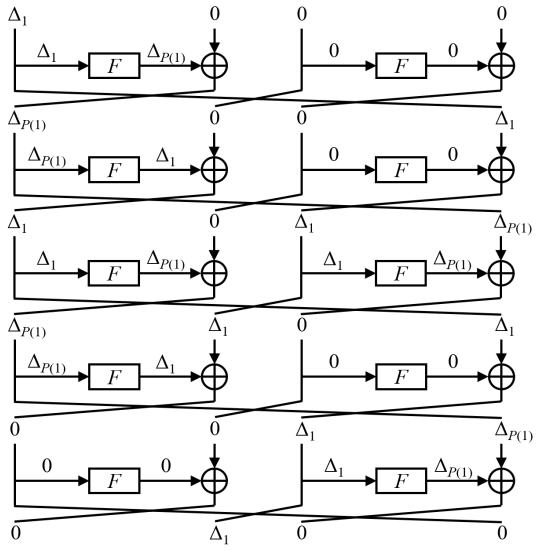

Fig. 5. Differential trail for the 5-round inbound structure of 4-branch GFN-2

4.2 기지키 구별자

4.1절에서 설명된 ISG2를 중심으로 정방향과 역방향으로 차분을 전파시켜 Table 3과 같은 차분 트레일을 얻을 수 있다. 이 차분 트레일의 상태 변수 차분들은 $\Delta X_i = (\Delta X_{i,0}, \ldots, \Delta X_{i,3})$으로 고려되는데, ISG2는 ΔX_0부터 ΔX_5까지, ISG2의 입력 차분으로부터 역방향 차분 전파는 ΔX_{-1}부터 ΔX_{-5}까지, 그리고 ISG2의 출력 차분으로부터 정방향 차분 전파는 ΔX_{+1}부터 ΔX_{+5}까지 표시된다. 리바운드 공격(Rebound Attack)에서는 이것을 아웃바운드 페이즈(Outbound Phase)라고 한다. 아웃바운드 페이즈에서는 F-함수의 입력 차분 형태에 대해 출력 차분 형태가 Table 2와 같이 결정된다.

Table 3의 차분 트레일은 0xFB → 0xEF 와 같이 표현될 수 있다. Table 3에서 각 $X_{i,j}$의 차분 형태는 $\Delta X_{i,j}$로 표시된다. 이상적인 암호(Ideal Cipher)의 경우, 키 입력을 알고 있는 공격자가 이와 같은 형태의 차분 트레일을 만족하는 쌍을 적어도 하나 찾는데 필요한 계산 복잡도는 다음과 같이 계산된다. 공격자는 2^b개의 입력값들로 $(?,?,\Delta_{P(1)},?)$ 형태의 차분을 갖는 쌍들을 만들 수 있다. 먼저, 입력의 $\Delta_{P(1)} = P(\Delta_1)$ 부분에서 Δ_1의 0 아닌 바이트 부분

Table 2. Transition from input difference form to output difference form of F function

Input Diff.	Output Diff.
0	0
Δ_1	$\Delta_{P(1)}$
$\Delta_{P(1)}$	?
?	?

Table 3. Difference propagation from 5-round inbound structure of 4-branch GFN-2

i	$\Delta X_{i,0}$	$\Delta X_{i,1}$	$\Delta X_{i,2}$	$\Delta X_{i,3}$
-5	?	?	$\Delta_{P(1)}$	?
-4	Δ_1	$\Delta_{P(1)}$	?	?
-3	$\Delta_{P(1)}$	?	0	Δ_1
-2	0	0	Δ_1	$\Delta_{P(1)}$
-1	0	Δ_1	0	0
0	Δ_1	0	0	0
1	$\Delta_{P(1)}$	0	0	Δ_1
2	Δ_1	0	Δ_1	$\Delta_{P(1)}$
3	$\Delta_{P(1)}$	0	0	Δ_1
4	0	0	Δ_1	$\Delta_{P(1)}$
5	0	Δ_1	0	0
+1	Δ_1	0	0	0
+2	$\Delta_{P(1)}$	0	0	Δ_1
+3	?	0	Δ_1	$\Delta_{P(1)}$
+4	?	Δ_1	$\Delta_{P(1)}$	?
+5	?	$\Delta_{P(1)}$	?	?

에서 모든 값이 나타나고 0 바이트 부분에서는 상수이며, '?' 부분은 랜덤하게 선택되는 2^b개의 입력값들의 집합을 구성한다. 그러면, 이 집합은 약 2^{2b-1} 개의 쌍을 가지며 모든 쌍은 (?,?,$\Delta_{P(1)}$,?)의 차분 형태를 만족한다. 그리고 출력 차분이 (?,$\Delta_{P(1)}$,?,?) 형태가 될 확률은 $2^{-(a-1)b}$ 이므로, 평균적으로 하나의 입력값 집합은 $2^{2b-1}\times 2^{-(a-1)b} = 2^{(-a+3)b-1}$ 개의 쌍을 갖는다. 그런데, 실제 블록암호 설계에서 a = 4 또는 8 이기 때문에 (-a+3)b- 1은 음수이다. 즉, $1/2^{(-a+3)b-1} = 2^{(a-3)b+1}$ 번을 반복하면 한 개의 쌍을 기대할 수 있다. 그러므로, 계산 복잡도는 $2^b\times 2^{(a-3)b+1} = 2^{(a-2)b+1}$ 이다.

반면에, 4-브랜치 GFN-2 구조의 경우, 공격자가 동일한 조건에서 하나의 쌍을 찾는데 필요한 계산 복잡도는 5-라운드 ISG2 구성에 필요한 계산이 전부이며 한 번의 15-라운드 4-브랜치 GFN-2 연산은 30번의 F-함수 연산으로 구성되므로, $9\times 2^{2b}/30$이다. a = 4 또는 8일 때, 아래 부등식과 같이 4-브랜치 GFN-2 구조에 대한 계산 복잡도가 이상적인 암호에 대한 것 보다 항상 작으므로 0xFB → 0xEF이 유효한 기지키 구별자임이 확인된다.

$$9\times 2^{2b}/30 = 2^{2b-1.73} < 2^{(a-2)b+1}$$

단, a = 4인 경우는 a = 8인 경우 보다 위 부등식의 좌변과 우변의 차이가 크지 않기 때문에 구별 공격의 어드벤티지 또한 크지 않다.

이 구별 공격의 복잡도는 Table 4에 정리되었다. 구별 공격의 유효성은 마지막 라운드에서 셔플 연산 적용 유무와 관련이 없으나, Table 4에서는 편의상 마지막 라운드에 셔플 연산이 있는 쪽에 구별 공격 결과를 기술하였다.

4.3 부분 충돌 공격

Table 3으로부터 파생되는 부분 충돌 공격이 Tabl e 4와 같이 정리된다. Table 4에서 'L'은 마지막 라운드의 셔플 연산을 의미하고, 해당 성분 값 'Y'는 마지막 셔플 연산이 있음을, 'N'은 마지막 셔플 연산이 없음을 의미한다. 즉, Table 4에서 상위 세 공격은 마지막 라운드의 셔플 연산이 있는 경우에 해당하고, 하위 두 공격에 대해서는 마지막 라운드의 셔플 연산이 없는 경우에 해당한다. 'R'은 공격되는 라운드 수를 의미한다. 'KKD'는 기지키 구별자를 의미하며 각 성분은 '(입력 차분 형태, 출력 차분 형태)'의 꼴로 표시하였다. 'w'는 부분 충돌 공격에서 충돌이 발생하는 워드의 개수를 의미한다. Table 4의 첫 번째 공격은 구별 공격이어서 이 영역을 '-'로 표시하였다. 'Comp.'는 공격에 필요한 복잡도를 의미하고 'Generic'은 해당 공격에 대응되는 이상적인 암호에 대한 구별 공격 복잡도 또는 생일 공격 복잡도를 의미한다. 마지막으로 '(a,b)'는 해당 공격을 유효하게 만드는 (a,b)값을 의미한다. 만일, (a,b) = (4,4), (4,8), (8,4), (8,8)에 대해 모두 공격이 유효하다면 해당 성분을 'all'로 표시한다. a = 8일 때에만 공격이 유효하다면 해당 성분을 '(8,*)'로 표시한다.

GFN-2 구조를 가진 블록암호에 MMO 또는 MP 해시 모드가 적용된다는 가정에서 이 부분 충돌 공격은 의미를 갖는다. 이 절에서 설명되는 공격들은 그러한 해시 함수에 대한 생일 공격 보다 더 효율적으로 부분 충돌 쌍을 찾기 때문이다.

Table 4의 두 번째 공격은 (?,?,$\Delta_{P(1)}$,?) → (?, Δ_1,$\Delta_{P(1)}$,?) 또는 (Δ_1,$\Delta_{P(1)}$,?,?) → (?,$\Delta_{P(1)}$,?,?)

Table 4. Known-key distinguishing and w-word partial-collision attacks on 4-branch GFN-2

L	R	KKDs	w	Comp.	Generic	(a,b)
Y	15	(0xFB, 0xEF)	-	$2^{2b-1.73}$	$2^{(a-2)b+1}$	all
	14	(0xFB, 0xDB), (0x6F, 0xEF)	1	$2^{3b-1.63}$	$2^{ab/2}$	(8,*)
	10	(0xB1, 0x81), (0x06, 0xC6)	3	$2^{4b-1.15}$	$2^{3ab/2}$	all
N	15	(0xFB, 0xFB)	1	$2^{3b-1.73}$	$2^{ab/2}$	(8,*)
	11	(0xB1, 0x81)	3	$2^{4b-1.29}$	$2^{3ab/2}$	all

형태의 기지키 구별자를 이용하는데, ISG2의 쌍을 2^b개 시도하면 1개의 1-워드 부분충돌쌍을 기대할 수 있다. 이 공격은 14 라운드에 대한 것이므로, 공격 복잡도는 4.1절에서 설명된 복잡도에 2^b을 곱하고 14-라운드 4-브랜치 GFN-2를 구성하는 F-함수의 개수로 나누어 계산한다. 즉, $2^b\times(9\times2^{2b})/28 = 2^{3b-1.63}$이다. 반면에, 출력값의 1개 워드에서 충돌을 발생시키는 쌍을 찾는 생일 공격의 복잡도는 $2^{ab/2}$이다. a = 8일 때에만 이 공격이 유효하다. 왜냐하면 이 경우에만 공격 복잡도가 생일 공격 복잡도 보다 작기 때문이다.

Table 4의 세 번째 공격은 $(\Delta_{P(1)},?,0,\Delta_1) \rightarrow (\Delta_{P(1)},0,0,\Delta_1)$ 또는 $(0,0,\Delta_1,\Delta_{P(1)}) \rightarrow (?,0,\Delta_1,\Delta_{P(1)})$ 형태의 기지키 구별자를 이용하는데, ISG2의 쌍을 2^{2b}개 시도하면 1개의 3-워드 부분충돌쌍을 기대할 수 있다. 이 공격은 10 라운드에 대한 것이므로, 공격 복잡도는 $2^{2b}\times(9\times2^{2b})/ 20 = 2^{3b-1.15}$ 이다. 반면에, 출력값의 3개 워드에서 충돌을 이 공격에 대응되는 생일 공격의 복잡도는 $2^{3ab/2}$이다. 이 공격은 모든 (a,b) 값에 대해 유효하다.

Table 4의 네 번째와 다섯 번째 공격은 마지막 라운드의 셔플 연산이 없는 것으로 가정한 상황에서의 공격이며, 각각 $(?,?,\Delta_{P(1)},?) \rightarrow (?,?,\Delta_{P(1)},?)$와 $(\Delta_{P(1)},?,0,\Delta_1) \rightarrow (\Delta_{P(1)},0,0,\Delta_1)$ 형태의 기지키 구별자를 이용한다. 4-브랜치 GFN-2 구조의 15 라운드가 30개의 F 함수 연산으로, 11 라운드가 22개의 F 함수 연산으로 구성된다는 점을 제외하고는 이전에 설명된 공격들의 복잡도와 유사하게 계산할 수 있다. 또한, 네 번째 공격이 a = 8일 때에만 유효하고 다섯 번째 공격은 모든 (a,b) 값에 대해 유효하다는 것도 쉽게 확인된다.

V. 결 론

GFN-2 구조는 Feistel 구조에 비해 작은 입출력 길이를 가진 F-함수를 이용하여 블록암호를 설계할 수 있다는 장점이 있다. 그러므로, 블록암호의 다양한 응용성을 고려할 때, 이러한 GFN-2 구조의 안전성을 여러 가지 가정과 공격 모델을 고려하여 연구하는 것은 유용한 결과를 만들어낼 수 있다. 본 논문에서는 [3]에서 제시한 4-브랜치 GFN-2 구조에 대한 기지키 구별 공격 및 부분 충돌 공격 결과들을 개선하였다.

향후에는, 브랜치의 개수가 4 보다 큰 GFN-2 구조에 대해서도 하여한 공격들에 대해서도 연구할 계획이다.

References

[1] Lars R. Knudsen and Vincent Rijmen, "Known-Key Distinguishers for Some Block Ciphers," ASIACRYPT 2007, LNCS 4833, pp. 315-324, Springer, 2007.

[2] Yu Sasaki and Kan Yasuda, "Known-Key Distinguishers on 11-Round Feistel and Collision Attacks on Its Hashing Modes," FSE 2011, LNCS 6733, pp. 397-415, Springer, 2011.

[3] HyungChul Kang, Deukjo Hong, Dukjae Moon, Daesung Kwon, Jaechul Sung, and Seokhie Hong, "Known-Key Attacks on Generalized Feistel Schemes with SP Round Function," IEICE Transactions on Fundamentals., Vol. 95-A, No. 9, pp. 1550-1560, 2012.

[4] Yu Sasaki, Sareh Emami, Deukjo Hong, and Ashish Kumar, "Improved Known-Key Distinguishers on Feistel-SP Ciphers and Application to Camellia," ACISP 2012, LNCS 7372, pp. 87-100, Springer, 2012.

[5] HyungChul Kang, Deukjo Hong, Jaechul Sung, and Seokhie Hong, "Known-Key Attack on SM4 Block Cipher," IEICE Transactions on Fundamentals., Vol. 100-A, No. 12, pp. 2985-2990, 2

017.
[6] FIPS 46-3, Data Encryption Standard (DES), Oct. 25, 1999.
[7] ISO/IEC 29192-2:2019, Information security – Lightweight cryptography – Part 2: Block ciphers, Nov. 2019.
[8] ISO/IEC 18033-3:2010, Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers, Dec. 2010.
[9] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen, "The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl," FSE 2009, LNCS 5665, pp. 260-276, Springer, 2009.
[10] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee, "HIGHT: A New Block Cipher Suitable for Low-Resource Device," CHES 2006, LNCS 4249, pp. 46-59, Springer, 2006.
[11] L. Dong, Y. Wang, W. Wu, and J. Zou, "Known-key distinguishers on 15-round 4-branch type-2 generalised Feistel networks with single substitution-permutation functions and near-collision attacks on its hashing modes," IET Information Security, 9(5): 277-283, 2015.
[12] Alfred Menezes, Paul Oorschot, and Scott Vanstone, *Handbook of Applied Cryptography*, CRC Press, Oct. 1996.

〈저자소개〉

홍 득 조 (Deukjo Hong) 종신회원
1999년 8월: 고려대학교 수학과 학사
2001년 8월: 고려대학교 수학과 석사
2006년 2월: 고려대학교 정보보호대학원 박사
2006년 3월~2007년 12월: 고려대학교 정보보호기술연구센터 연구교수
2007년 12월~2015년 8월: 국가보안기술연구소 선임연구원
2015년 9월~현재: 전북대학교 IT정보공학과 부교수
〈관심분야〉 암호 알고리즘 설계 및 분석