Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.3, Jun. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.3.467

비밀 S-box를 사용한 GFN에 대한 안전성 분석*

이 용 성,[1†] 강 형 철,[1] 홍 득 조,[2‡] 성 재 철,[3] 홍 석 희[1]
[1]고려대학교 정보보호연구원, [2]전북대학교, [3]서울시립대학교

Security Analysis on GFN with Secret S-box*

Yongseong Lee,[1†] HyungChul Kang,[1]
Deukjo Hong,[2‡] Jaechul Sung,[3] Seokhie Hong[1]
[1]Center for Information Security Technologies(CIST), Korea University,
[2]Chonbuk National Universitry, [3]University of Seoul

요 약

본 논문에서는 라운드 함수의 업데이트 함수로 SP 구조를 사용하고 비밀 S-box가 적용된 GFN(Generalized Feistel Networks) Type I, Type II, Type III에 대한 안전성을 분석한다. 이 환경에서 공격자는 S-box에 대한 정보를 갖지 못한다. 인테그랄 공격 기법(Integral attack) 기반의 선택 평문 공격으로 9 라운드(Type I), 6 라운드(Type II), 6라운드(Type III)에 대한 비밀 S-box 정보를 복구할 수 있다. 선택 암호문 공격으로 전환할 경우 GFN Type I의 16 라운드까지 비밀 S-box의 정보가 복구된다. 결론적으로 m비트 비밀 S-box와 $k \times k$ MDS 행렬이 라운드 함수로 사용된 GFN Type I, Type II, Type III에 대하여 비밀 S-box를 복구하는데 $\frac{2^{3m}}{32k}, \frac{2^{3m}}{24k}, \frac{2^{3m}}{36k}$만큼의 시간복잡도가 필요하다.

ABSTRACT

In this paper, we analyze Generalized Feistel Network(GFN) Type I, Type II, Type III that round function use SP update function, secret S-box and $k \times k$ MDS matirx. In this case an attacker has no advantage about S-box. For each type of GFN, we analyze and restore secret S-box in 9, 6, 6 round using the basis of integral cryptanalysis with chosen plaintext attack. Also we restore secret S-box in 16 round of GFN Type I with chosen ciphertext attack. In conclusion, we need 2^{2m} data complexity and $\frac{2^{3m}}{32k}, \frac{2^{3m}}{24k}, \frac{2^{3m}}{36k}$ time complexity to restore m bit secret S-box in GFN Type I, Type II, Type III.

Keywords: Secret S-box, Integral cryptanalysis, Structural cryptanalysis, Generalized feistel networks

I. 서 론

대부분의 블록 암호는 페이스텔(Feistel) 구조와 SP(Substitution Permutation) 구조로 크게 분류할 수 있다. $2n$비트의 입력값을 갖는 페이스텔 구조는 매 i번째 라운드 평문 블록을 n비트값 X_L^i과 X_R^i로 나누고 $(X_L^{i+1}, X_R^{i+1}) = (X_R^i \oplus F^i(X_L^i), X_L^i)$의 방법으로 매 라운드 업데이트한다($i \geq 1$, Fig 1. 참고). 이 때 나누어진 n비트 입력값인 X_L과 X_R을브랜치(Branch)라 부르며, F^i는 i번째 라운드의 업데이트 함수라 부른다. 대표적인 페이스텔 구조의 암호는 DES [5], SEED [10]가 존재한다. SP 구조

Received(03. 31. 2017), Modified(1st: 05. 15. 2017, 2nd: 05. 25. 2017), Accepted(05. 26. 2017)
* 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.B0722-16-0006, (창조씨앗-2단계)암호와 물리계층보안을 결합한 IoT 네트워크 보안 기술 개발)
† 주저자, yslee.cist@gmail.com
‡ 교신저자, deukjo.hong@gmail.com(Corresponding author)

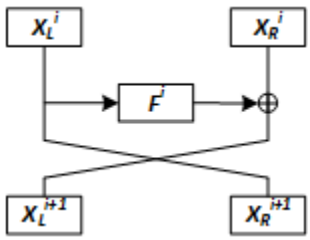

Fig. 1. Feistel structure

는 매 라운드가 치환(Substitution)과 순열(Permutation)을 반복하여 사용하는 형태이며, 매 라운드 $R^i(X,K)=P(S(X\oplus K))$와 같이 데이터를 업데이트 한다. SP 구조의 대표적인 암호는 AES[4], ARIA [13]등이 존재한다.

브랜치가 2개만 존재하는 페이스텔 구조를 확장하여 일반화 시킨 GFN(Generalized Feistel Networks)이 Y. Zheng 등에 의해 제안되었다[6]. 이 연구진은 브랜치의 개수가 임의의 t개로 확장되었을 경우에 대하여, 각 라운드별로 브랜치가 업데이트되는 방법에 따라 Type I, Type II, Type III로 구분하여 정의하였다. GFN은 페이스텔 구조보다 다양한 라운드 함수를 구성할 수 있는 장점으로 인해 많은 암호 설계에 적용되었으며, 대표적으로 GFN Type II를 사용한 CLEFIA [15]가 존재한다. GFN 구조와 유사하게 여러 브랜치를 사용하여 구현한 암호로 HIGHT [11]와 LEA [12]등이 있다.

현대에 사용하는 블록암호의 구조는 이외에도 몇 가지가 있지만 블록암호의 구조적인 모습은 그 가짓수가 많지 않다. 이러한 일반적인 구조의 안전성을 분석하는 방법으로 구조적 암호 분석 기법(Structural Cryptanalysis)이 존재한다. 구조적 암호 분석 기법은 특정한 함수의 취약점으로 안전성을 분석하는 다른 분석 방법과 달리, 알고리즘의 구조적 형태에 의존하여 안전성을 분석한다. 이 때문에 구조적 암호 분석 기법은 특정된 암호 알고리즘을 분석하는데 효과적이지는 못하지만, 특정한 구조를 갖는 암호 알고리즘에서 안전성의 하한을 제시하는데 효과적이다. Biryukov와 Shamir는 S(Substitution) 계층과 A(Affine) 계층으로 나누어진 SASAS 구조에 대해 S 계층과 A 계층을 복원하는 방법을 소개하였다 [2]. 이러한 분석으로 2.5 라운드로 구성된 SP 구조는 전수조사보다 효과적으로 내부 상태를 완벽하게 복원할 수 있으므로 안전하지 않음을 알 수 있다.

[2]의 결과를 이용하여 Tiessen 등은 비밀 S-box를 사용한 AES의 안전성을 분석하였다 [1]. 비밀 S-box는 S-box에 대한 정보가 공격자에게 전혀 없는 환경을 의미한다. 비밀 S-box를 사용하는 중요한 이유 중 하나는 추가적인 안전성을 확보하는데 있다. 공격자는 비밀키를 찾기 위해 비밀 S-box를 찾아야 하며, m비트 전단사 S-box의 경우 모든 가능한 S-box의 가짓수는 $2^m!$개가 된다. 하지만 [1]에서는 AES에서 비밀 S-box를 사용하더라도, 인테그랄(Integral) 공격 기법과 동일한 라운드가 분석되면서 추가로 비밀 S-box까지 복원할 수 있어, 인테그랄 공격에 대하여 안전성의 향상이 크지 않음을 밝혀냈다. 이 결과로 다른 암호 시스템에서도 비밀 S-box를 사용하게 될 경우 안전성이 기대한 만큼 확보되지 않을 가능성이 있다.

비밀 S-box에 대한 연구는 암호 알고리즘에 적용된 화이트박스(White-box)를 복원하는 연구의 중요한 흐름 중 하나로 볼 수 있다. 화이트박스는 암호 알고리즘의 안전성을 높이기 위한 방법으로 암호 연산 중 생기는 모든 정보가 사용자에게 노출 되더라도 안전한 암호 시스템을 만드는 것을 목표로 한다. 비밀 S-box는 공격자에게 S-box에 대한 정보가 노출되지 않기 때문에, 공격자가 연산 중간값을 알더라도 비밀값을 복원하는데 어려움이 존재한다. 화이트박스 알고리즘은 알고리즘 연산의 중간값이 노출되더라도 안전성을 확보할 수 있도록 라운드키 연산을 포함한 매 라운드의 입출력에 임의의 가역함수 연산을 덧붙여 커다란 참조표를 생성하고 이를 암호화 연산에 이용한다. 즉, 매 라운드의 입출력크기를 갖는 비밀 S-box를 가정한다면 화이트박스 알고리즘에 가까운 형태로 볼 수 있다. 때문에 이러한 비밀 S-box의 복원에 관련된 연구흐름은 화이트박스의 안전성을 파악하는데 도움이 된다.

본 논문에서는 비밀 S-box가 적용된 GFN Type I, Type II, Type III에 대해 비밀 S-box를 복원하는 방법을 소개한다. 이때 라운드 함수의 업데이트 함수는 SP구조로 가정하고, S로 사용되는 S-box는 공격자에게 정보가 알려지지 않은 m비트 전단사 S-box(Secret bijective S-box)를 사용하고, P로는 확산(Diffusion)의 영향이 가장 큰 MDS(Maximum Distance Seperable) 코드를 사용한다고 가정한다. GFN이 특정된 라운드함수를 사용하지 않기 때문에 구조적 암호 분석 기법으로 분석이 될 수 있으며, 인테그랄 공격을 적용하여 비밀

S-box를 복원할 수 있다.

본 논문의 2절에서는 공격에 필요한 배경지식으로 본 논문에서 사용할 표기법을 정의하고, 공격대상인 GFN에 대한 설명을 한다. 또한 비밀 S-box를 복원하는데 필요한 인테그랄 암호 분석 기법에 대하여 설명하며, [2] 에서 S계층을 복원한 방법에 대하여 간략히 설명한다. 3절에서는 비밀 S-box를 복원하는 구체적인 방법을 설명한다. 이를 이용하여 선택평문 공격의 결과로 9라운드(Type I), 6라운드(Type II), 6라운드(Type III)에서 비밀 S-box가 복원됨을 보이고, 각각 $\frac{2^{3m}}{18k}$(Type I), $\frac{2^{3m}}{24k}$(Type II), $\frac{2^{3m}}{36k}$(Type III)의 복잡도가 필요함을 보인다. 또한 선택 암호문 공격으로 16라운드로 구성된 GFN Type I에서 비밀 S-box 정보를 $\frac{2^{3m}}{32k}$의 복잡도로 복원할 수 있음을 보인다. 마지막으로 4절에서 본 논문의 결과를 정리하고, 향후 연구 방향에 대하여 서술한다.

II. 배경지식

2.1 표기법

본 논문의 기본적인 표기법은 다음과 같다.

- o m : word 비트 길이
- o t : 브랜치 개수
- o k : 하나의 브랜치를 구성하는 word의 개수
- o n : 하나의 브랜치의 비트 길이
- o N : GFN의 입·출력 블록 길이 ($N=tn$)
- o c_i : i번째 암호문($c_i=(c_{i,0},c_{i,1},\cdots,c_{i,k-1})$)
- o $c_{i,j}$: i번째 암호문의 j번째 m비트 word
- o X_j^i : i번째 라운드의 j번째 n비트 브랜치
- o F_j^i: GFN에서 i번째 라운드의 j번째 n비트 브랜치를 입력받아 업데이트하는 함수
- o K_j^i : F_j^i에서 사용되는 라운드 키
- o $K_{j,k}^i$: K_j^i의 k번째 m비트 word
- o R^i : i번째 라운드 함수
- o S : m비트 비밀 S-box
- o M : $k\times k$ MDS 행렬

원소의 중복을 허용하여 집합을 구성하는 것을 멀티셋(Multiset)이라 한다. 일반적인 집합에서 $\{a,a,b\}$라는 집합과 $\{a,b\}$라는 집합이 존재하면, 두 집합을 같은 집합으로 분류하지만, 멀티셋의 관점에서는 다른 집합이 된다. 암호 분석에서 멀티셋의 원소는 같은 비밀키를 사용하는 알고리즘의 평문, 암호문 또는 특정 위치의 중간값이 될 수 있다. 예를 들어 8비트 암호 알고리즘이 있을 때, 5개의 평문 집합 $\{0,0,5,7,7\}$은 하나의 멀티셋으로 볼 수 있다. 이렇게 멀티셋을 구성하였을 때, 암호 분석에 사용가능한 특정한 성질이 존재하며 분석에 사용 가능한 멀티셋의 성질과 표기는 다음과 같다.

- o A: all set
- o B: balanced set
- o C: constant set
- o D: dual set
- o ? : unknown set

A는 멀티셋의 가능한 모든 원소가 한 번씩 나오는 집합을 의미하고, B는 멀티셋 내의 모든 원소를 XOR(eXclusive-OR)하였을 경우 0이 되는 성질을 갖는 집합이다. C는 멀티셋 내의 모든 원소가 동일한 집합이며, D는 A의 성질을 갖거나 모든 원소가 짝수 번 등장하는 성질을 갖는 집합이다. ?는 A, B, C, D 의 성질 중 어느 것도 만족하지 않는 집합이다. 3비트 멀티셋 성질의 예시는 다음과 같다.

- o A = {0, 1, 2, 3, 4, 5, 6, 7}
- o B = {1, 1, 1, 2, 2, 3, 4, 6}
- o C = {1, 1, 1, 1, 1, 1, 1, 1}
- o D = {2, 2, 5, 5, 6, 6, 6, 6}
- o ? = {0, 1, 2, 3, 3, 5, 6, 7}

멀티셋이 특정 길이별로 성질이 존재한다면 한번에 C^kAB^t와 같은 형식으로 표기한다. 앞의 C^kAB^t는 멀티셋을 특정한 길이별로 보았을 때, C의 성질을 갖는 부분이 k개가 존재하고, A의 성질을 갖는 부분이 하나 존재하며, B의 성질을 갖는 부분이 t개 존재하는 멀티셋임을 의미한다.

2.2 GFN 구조

N비트의 평문블록을 갖는 GFN Type I, Type

II, Type III구조는 평문블록을 t개로 나누어 각각의 블록을 n비트 값 $(X_1^i, X_2^i, \cdots, X_t^i)$으로 둔다. 이때 X_j^i는 i번째 라운드 입력값의 n비트 브랜치가 된다. GFN Type I, Type II, Type III는 매 라운드 다음과 같이 업데이트 된다.

o GFN Type Ⅰ

$$(X_1^{i+1}, X_2^{i+1}, \cdots, X_t^{i+1}) = (X_2^i \oplus F_1^i(X_1^i, K_1^i), X_3^i, X_4^i, \cdots, X_t^i, X_1^i)$$

o GFN Type Ⅱ

$$(X_1^{i+1}, X_2^{i+1}, \cdots, X_t^{i+1}) = (X_2^i \oplus F_1^i(X_1^i, K_1^i), X_3^i, X_4^i \oplus F_3^i(X_3^i, K_3^i), \cdots, X_t^i \oplus F_{t-1}^i(X_{t-1}^i, K_{t-1}^i), X_1^i)$$

o GFN Type Ⅲ

$$(X_1^{i+1}, X_2^{i+1}, \cdots, X_t^{i+1}) = (X_2^i \oplus F_1^i(X_1^i, K_1^i), X_3^i \oplus F_2^i(X_2^i, K_2^i), X_4^i \oplus F_3^i(X_3^i, K_3^i), \cdots, X_t^i \oplus F_{t-1}^i(X_{t-1}^i, K_{t-1}^i), X_1^i)$$

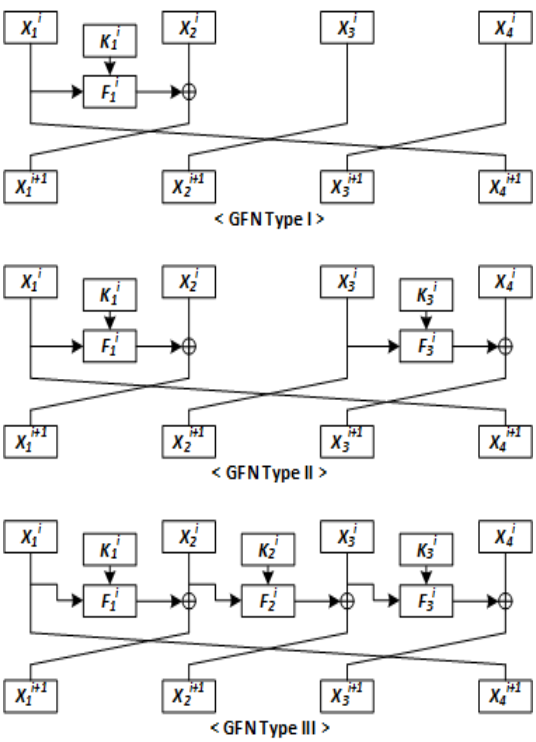

Fig. 2. GFN with 4 branches

본 논문에서는 브랜치의 개수가 4개인 GFN Type I, Type II, Type III에 대하여 분석을 진행할 것이다(Fig 2. 참고). 각 브랜치를 업데이트하는 $F_{i,j}$ 함수는 동일한 m비트 비밀 S-box k개와 동일한 $k \times k$ MDS 행렬을 사용하는 SP 구조로 가정한다. 또한, 각 브랜치의 크기는 n비트로 가정하므로, $n = k \times m$을 만족한다. 따라서 업데이트 함수 F_j^i 함수는 라운드 키를 XOR하는 key XOR, m비트 S-box를 통과하는 치환(substitution), 그리고 MDS 행렬 연산을 하는 확산(diffusion) 순으로 구성된다.

2.3 인테그랄 암호 분석 기법

인테그랄 암호 분석 기법은 Knudsen 등에 의하여 제안된 블록 암호 공격 방법이다 [3]. 블록 암호 Square 제안 논문 [7]에서 Square의 안전성을 보이기 위한 특화 공격으로 Square 공격이 소개되었는데, 이 공격 방법을 일반화하여 인테그랄 암호 분석 기법으로 완성되었다. 이 공격 방법의 핵심은 특정한 성질을 만족하도록 평문 또는 암호문을 묶어 멀티셋을 구성하고, 암호화 또는 복호화 과정에서 멀티셋의 성질이 어떻게 변화하는지를 계산하여 비밀키를

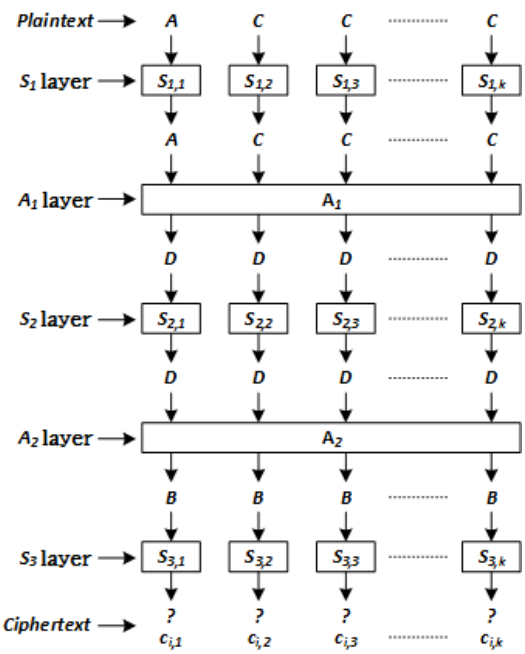

Fig. 3. Attack path on SASAS

복구하는 공격 방법이다. 인테그랄 암호 분석 기법은 블록 암호를 공격하는 가장 강력한 공격 중 하나이며, Y.Todo [8]는 인테그랄 공격을 개선한 Division Property를 이용하여 블록 암호 MISTY1 [9]의 전체 라운드를 분석하였다.

인테그랄 암호 분석 기법에서 A의 성질을 갖는 멀티셋은 전단사 S-box를 지나도 A의 성질을 유지한다. S-box의 입력으로 모든 값이 한 번씩 입력되기 때문에 출력에서도 모든 값이 한 번씩 나오기 때문이다. A의 성질을 갖는 두 개 이상의 멀티셋이 선형연산으로 서로 결합이 되는 경우 A의 성질을 더 이상 보장할 수 없게 된다. 하지만 A의 성질을 갖는 멀티셋은 각각 모든 원소를 XOR하였을 때 0이 됨을 알고 있다. 두 개 이상의 A의 성질을 갖는 멀티셋이 선형연산으로 서로 결합이 되어도 이 성질을 변하지 않는다. 하지만 A의 성질이 유지될 수 있는 보장이 존재하지 않는다. 따라서 이 경우 A의 성질이 B의 성질로 변하게 된다. B의 성질을 갖는 멀티셋은 모든 원소를 XOR하였을 때 0이 됨을 알고 있지만, 비선형 함수인 S-box를 지났을 때 그 성질을 더 이상 보장할 수 없다. 이 경우 ?의 성질을 갖는 멀티셋이 된다.

2.4 SASAS의 구조적 분석

[2]에서 소개되었던 SASAS 구조에서 S 계층과 A 계층을 복구하는 방법은 인테그랄 공격방법과 유사하다. 하지만 공격 경로를 탐색할 때 추가적인 성질을 갖는 D(dual) 멀티셋을 이용하였다. D는 A의 성질을 갖거나 모든 원소가 짝수 번 등장하는 성질을 갖는 집합이다.

각각의 계층은 순서에 따라 $S_1A_1S_2A_2S_3$계층으로 구분되며, S_i계층의 j번째 S-box는 $S_{i,j}$로 표기한다 (Fig 3. 참고).

각 S-box는 m비트로 가정하고, k개의 S-box가 S 계층을 구성할 때 Fig 3.과 같이 AC^{k-1}의 성질을 갖는 멀티셋을 이용하여 공격 경로를 구성한다. S_1 계층을 지나도 이 성질은 변하지 않으며, A_1 계층을 지날 때 D^k로 멀티셋의 성질이 변하게 된다 ([2]의 Lemma 2. 참고). S_2 계층을 지날 때 D^k의 성질은 그대로 유지되며, A_2 계층을 지날 때, B^k로 성질이 변하게 된다([2]의 Lemma 2. 참고). S 계층과 A 계층의 복원은 위의 공격경로를 갖도록 2^{2m}개의 평문으로 2^m개의 멀티셋을 구성한 다음 아래와 같이 진행된다.

1) S_3 계층의 입력값이 B의 성질을 만족함을 이용하여 $\bigoplus_{i=0}^{2^m-1} S^{-1}(c_{i,j}) = 0, (1 \le j \le k)$ 을 만족하는 방정식을 구성한다.
2) 충분한 방정식을 획득하면 이를 이용하여 S-box를 복원할 연립방정식을 구성한다. 가능한 S-box의 진리표를 $2^m \times m$ 행렬로 구성하면, S-box가 전단사함수이기 때문에 m개의 열들이 선형독립을 이룬다. 열들의 임의의 선형결합은 마찬가지로 연립방정식의 해가 되고, 모든 해의 보수 역시 연립방정식의 해가 된다. 따라서 연립방정식은 최소 $m+1$ 차원의 커널(kernel)을 가지고 있다. 그러므로 구성된 연립방정식은 최대 $2^m - m - 1$개의 선형독립식을 구성한다.
3) 획득한 연립방정식의 해로부터 정확한 S-box를 복원할 수는 없지만, 연립방정식 해의 아핀 동치(affine equivalent) 중 하나는 정확한 S-box를 표현하고 있다. 그렇기 때문에 연립방정식으로부터 얻어진 임의의 S-box를 실제 S-box로 가정하면, 임의의 아핀 함수와 역함수가 A_2 계층에 추가로 적용된 형태로 생각할 수 있다. 아핀 함수 역시 선형 함수이기 때문에 실제 A_2 계층에 아핀 함수가 추가로 연산이 되더라도 여전히 선형 함수이다. 따라서 연립방정식으로부터 획득한 S-box를 실제 S-box로 가정하면 S_3 계층이 사라지며 $S_1A_1S_2A_2'$형태로 변하게 된다.
4) 암호문으로부터 S_1 계층을 같은 방법으로 제거한 후 남은 A_1' 계층과 A_2' 계층의 복원은 Biham's low rank detection technique [14]을 이용하여 해결할 수 있다.

A_2 계층을 복원하는 것과 A_2' 계층을 복원하는 것은 본질적으로 임의의 아핀 함수를 복원하는 것이므로 이후 공격에 큰 영향을 끼치지 않는다. A 계층까지 성공적으로 복원하면, 남은 S_2 계층은 자동으로 복원되어 공격이 완료된다.

III. GFN의 비밀 S-box 복원

GFN의 비밀 S-box 복원은 AC^{k-1} 의 성질을

갖는 멀티셋으로부터 공격이 시작된다. AC^{k-1}의 멀티셋이 업데이트 함수 F를 통과하게 되면, A^k의 성질을 갖는 멀티셋으로 변화하게 되는데 이는 MDS 행렬에 의하여 A의 성질이 확산되기 때문이다. A^k의 성질을 갖는 멀티셋이 다시 한 번 업데이트 함수 F를 통과하게 되면 B^k의 성질을 갖는 멀티셋으로 변하게 되며, B^k의 성질을 갖는 멀티셋이 라운드 함수 F를 통과하면 성질을 알 수 없는 ?의 성질로 변하게 된다.

본 장에서는 GFN Type I, Type II, Type III에서 인테그랄 암호 분석 기법을 이용하여 비밀 S-box를 복원하는 공격 방법을 설명한다.

3.1 비밀 S-box 복원

비밀 S-box의 복원은 인테그랄 공격으로 멀티셋이 B^k의 성질을 갖는 부분에서 이루어진다(Fig. 4. 회색 부분 참고). GFN Type I, Type II, Type III에 대하여 비밀 S-box의 복원 과정은 큰 차이가 없으며, 본 장에서는 GFN Type I에 대한 복원 과정을 예로 들어 설명한다.

1, 2, 3번째 브랜치에는 C^k의 성질을 갖고, 4번째 브랜치에 AC^{k-1}의 성질을 갖도록 2^m개의 평문을 구성하여 암호화한다. 얻어진 암호문들에 대하여 마지막 라운드를 복호화하면 X_2^8에서 B의 성질을 갖는 멀티셋이 구성되므로 아래의 식이 성립함을 알 수 있다.

$$\bigoplus_{i=0}^{2^m-1}\left(M\begin{bmatrix} S(c_{i,0}\oplus K_{1,0}^8)\\ S(c_{i,1}\oplus K_{1,1}^8)\\ \vdots \\ S(c_{i,k-1}\oplus K_{1,k-1}^8)\end{bmatrix}\oplus\begin{bmatrix} c_{i,k}\\ c_{i,k+1}\\ \vdots \\ c_{i,2k-1}\end{bmatrix}\right) = \begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix} \quad \cdots(1)$$

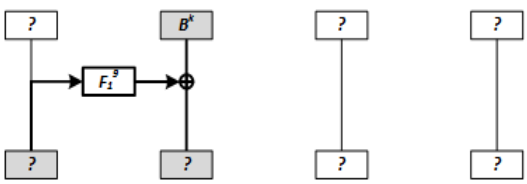

Fig. 4. Attack round of GFN Type I

식(1)로부터 실제적인 공격에 사용할 아래의 식을 유도할 수 있다.

$$\bigoplus_{i=0}^{2^m-1}\begin{bmatrix} S(c_{i,0}\oplus K_{1,0}^8)\\ S(c_{i,1}\oplus K_{1,1}^8)\\ \vdots \\ S(c_{i,k-1}\oplus K_{1,k-1}^8)\end{bmatrix} = \bigoplus_{i=0}^{2^m-1} M^{-1}\bullet\begin{bmatrix} c_{i,k}\\ c_{i,k+1}\\ \vdots \\ c_{i,2k-1}\end{bmatrix} \quad \cdots(2)$$

식(2)의 우항은 암호문으로부터 계산하여 얻을 수 있기 때문에,

$$z_{i,j} := S(i\oplus K_{1,j}^8),$$
$$\begin{bmatrix} c'_k\\ c'_{k+1}\\ \vdots \\ c'_{2k-1}\end{bmatrix} := \bigoplus_{i=0}^{2^m-1} M^{-1}\begin{bmatrix} c_{i,k}\\ c_{i,k+1}\\ \vdots \\ c_{i,2k-1}\end{bmatrix}$$

로 정의하고 각각의 식을 정리하여 다음의 방정식을 얻는다.

$$\bigoplus_{i=0}^{2^m-1}\begin{bmatrix} z_{c_{i,0},0}\\ z_{c_{i,1},1}\\ \vdots \\ z_{c_{i,k-1},k-1}\end{bmatrix} = \begin{bmatrix} c'_k\\ c'_{k+1}\\ \vdots \\ c'_{2k-1}\end{bmatrix} \quad \cdots(3)$$

(3)의 식에서 각 행에 해당하는 방정식은 2^m개의 변수를 갖기 때문에, 인테그랄 공격을 2^m번 반복하여 2^m개의 서로 독립인 방정식을 얻는다면 연립방정식을 해결할 수 있을 것이다. 이때, 좌변의 변수 $z_{i,j}$는 암호문의 실제값을 기준으로 정의되어있고, 우변은 암호문으로부터 계산할 수 있기 때문에 획득한 방정식이 서로 독립임을 쉽게 확인할 수 있다.

[1,2]의 S-box 복구과정에서는 SP 구조로 인하여 생성된 연립방정식의 우항이 전부 0인 동차(homogeneous) 형태지만, GFN에서 생성된 연립방정식은 우항이 0이 아닌 비동차(nonhomogeneous) 형태이다. 따라서 서로 독립인 방정식을 충분히 모았다고 할지라도

$$x_1\oplus x_2\oplus\cdots\oplus x_{2^m}=0$$

과 같은 동차식이 참이라고 가정하면, 임의의 아핀 변환(affine transformation) T에 대하여

$$T(x_1)\oplus T(x_2)\oplus\cdots\oplus T(x_{2^m})=0$$

역시 참이 된다. 앞 절에서 설명했듯이 이 경우 연립방정식은 $m+1$차원의 커널을 갖는다. 하지만 비동차식의 경우 임의의 상수값 b에 대하여 $T: x\rightarrow x\oplus b$ 와 같은 아핀 변환으로만 참인 식이 생성된다. 이 경우 생성되는 연립방정식은 최대 2^m-1개의 서로 독립인 방정식이 되며, 커널의 차원은 1차원이 된다. 따라서 [1,2]에서 발생하는 아핀 동치(affine equivalent)의 개수보다 그 수가 2^m으로 현저히 감소한다.

식(3)으로 구성된 연립방정식을 풀었을 때 우리가 결정할 수 있는 S-box의 형태는 임의의 m비트 상수 a와 b에 대한 $S(a\oplus x)\oplus b$의 형태로 볼 수 있다. 이는 S-box의 입·출력에 XOR되는 상수값이 비동차 방정식의 해에 영향을 주지 않기 때문이다. 따라서 2^{2m}가지의 가능한 a,b쌍을 전수조사 하면, S-box를 복원할 수 있게 된다. 이 수치는 가능한 S-box의 모든 가능한 아핀 동치인 2^{m^2+m}보다 훨씬 적은 가짓수이며, 충분히 전수조사가 가능한 수치이다.

3.2 선택 평문 공격(Chosen Plaintext Attack; CPA)

GFN Type I, Type II, Type III에 대한 인테그랄 공격 경로는 각각 Table 1, 2, 3. 과 같다. 표에서는 각 라운드의 출력값을 표기하였고 라운드 0의 출력값은 평문(plaintext)을 의미한다.

Table 1. CPA Integral path(GFN Type Ⅰ)

Round	X_1	X_2	X_3	X_4
0	C^k	C^k	C^k	AC^{k-1}
1	C^k	C^k	AC^{k-1}	C^k
2	C^k	AC^{k-1}	C^k	C^k
3	AC^{k-1}	C^k	C^k	C^k
4	A^k	C^k	C^k	AC^{k-1}
5	B^k	C^k	AC^{k-1}	A^k
6	?	AC^{k-1}	A^k	B^k
7	?	A^k	B^k	?
8	?	B^k	?	?
9	?	?	?	?

전체적인 공격과정은 첫 라운드에 m비트의 A의 성질을 갖는 멀티셋을 구성하여 시작한다. 나머지 $(n-m)$비트는 m비트 단위로 C 성질을 갖는 멀티셋을 구성하여 최대한 많은 라운드를 갈 수 있는 경로를 구성하였다.

GFN Type I, Type II, Type III에 대하여 각각 9, 6, 6 라운드의 공격 경로가 구성된다. 각각의 공격에서 필요한 데이터 복잡도는 연립방정식을 구성하기 위해 필요한 데이터이다. 연립방정식은 커널이 1차원이고 변수가 2^m개이기 때문에 최대 2^m-1차원을 갖는다. 매번 선택되는 연립방정식은 이전에 선택한 연립방정식과 독립이어야 하므로, 이전에 선택한 연립방정식이 l차원을 이룰 때 새로 선택한 연립방정식이 독립일 확률은 $1-\frac{2^l}{2^{2^m-1}}=1-\frac{1}{2^{2^m-1-l}}$이 된다. 이 때 기댓값은 확률의 역수인 $\frac{2^{2^m-1-l}}{2^{2^m-1-l}-1}=1+\frac{1}{2^{2^m-1-l}-1}$이 된다. 따라서 총 기댓값은

$$\sum_{l=0}^{2^m-2}\left(1+\frac{1}{2^{2^m-1-l}-1}\right)$$
$$=(2^m-1)+\left(\frac{1}{2^{2^m-1}-1}+\frac{1}{2^{2^m-2}-1}+\cdots+\frac{1}{2-1}\right)$$
$$<(2^m-1)+2$$

이 되므로 2^m+1개의 연립방정식을 획득하면 2^m-1개의 연립방정식은 서로 독립일 것으로 기대할 수 있다. 따라서 2^m+1개의 연립방정식을 획득하기 위해 $2^m(2^m+1)\approx 2^{2m}$의 데이터 복잡도가 필요하다.

시간복잡도를 살펴보면 데이터를 암호화 하는데 2^{2m} 암호화 복잡도가 들어간다. 가우스 소거법(Gaussian elimination)으로 연립방정식을 푸는

Table 2. CPA Integral path(GFN Type Ⅱ)

Round	X_1	X_2	X_3	X_4
0	C^k	AC^{k-1}	C^k	C^k
1	AC^{k-1}	C^k	C^k	C^k
2	A^n	C^k	C^k	AC^{k-1}
3	B^k	C^k	AC^{k-1}	A^n
4	?	AC^{k-1}	B^k	B^k
5	?	B^k	?	?
6	?	?	?	?

Table 3. CPA Integral path(GFN Type III)

Round	X_1	X_2	X_3	X_4
0	C^k	C^k	C^k	AC^{k-1}
1	C^k	C^k	AC^{k-1}	C^k
2	C^k	AC^{k-1}	A^n	C^k
3	AC^{k-1}	B^k	B^k	C^k
4	B^k	?	?	AC^{k-1}
5	?	?	?	B^k
6	?	?	?	?

데 필요한 시간 복잡도는 $O(n^3)$으로, 2^m개의 변수를 갖는 연립방정식을 푸는데 필요한 시간 복잡도는 2^{3m}번의 단계가 필요하다. [1]에서 가우스 소거법 한 번의 단계가 한 번의 테이블 룩업(table lookup)에 대응되는 점을 보았을 때, 2^{3m}번의 표 검색에 해당되는 시간 복잡도임을 짐작할 수 있다. GFN Type I, Type II, Type III 에서 각각 매 라운드 $2k, 4k, 6k$ 번의 테이블 룩업이 사용되므로(F 함수당 k번의 S-box 참조, k번의 라운드키 참조), 연립방정식을 푸는 복잡도는 각각 $\frac{2^{3m}}{18k}, \frac{2^{3m}}{24k}, \frac{2^{3m}}{36k}$의 암호화에 해당된다. 또한 S-box를 결정하는데 2^{2m}번의 암호화가 필요하다.

연립방정식의 해를 찾는 과정이 가장 오래걸리기 때문에, 전체적인 시간 복잡도는 GFN Type I, Type II, Type III 에 따라 연립방정식을 해결하는데 필요한 $\frac{2^{3m}}{18k}, \frac{2^{3m}}{24k}, \frac{2^{3m}}{36k}$이 된다.

3.3 선택 암호문 공격(Chosen Ciphertext Attack; CCA)

선택 암호문 공격의 공격 과정은 선택 평문 공격과 크게 다르지 않다. 암호문으로부터 구성된 인테그랄 공격 경로를 따라 복호화한 뒤, 첫 번째 라운드에서 한 라운드를 암호화하여 생성된 식으로부터 비밀 S-box를 복원한다. 선택 암호문 공격으로 인테그랄 공격 경로를 구성하면, GFN Type I의 공격 경로가 9 라운드에서 16 라운드로 증가하게 된다. 하지만 GFN Type II에서는 공격 라운드의 변화가 없으며, GFN Type III에서는 오히려 공격 라운드가 줄어들어 더 효과적인 공격이 불가능하다. GFN Type I에 대한 선택 암호문 공격 경로는 Table 4. 와 같고 공격 복잡도는 $\frac{2^{3m}}{32k}$이다.

Table 4. CCA Integral path(GFN Type I)

Round	X_1	X_2	X_3	X_4
16	C^k	AC^{k-1}	C^k	C^k
15	C^k	AC^{k-1}	C^k	C^k
14	C^k	C^k	AC^{k-1}	C^k
13	C^k	C^k	C^k	AC^{k-1}
12	AC^{k-1}	A^n	C^k	C^k
11	C^k	AC^{k-1}	A^n	C^k
10	C^k	C^k	AC^{k-1}	A^n
9	A^n	B^k	C^k	AC^{k-1}
8	AC^{k-1}	B^k	B^k	C^k
7	C^k	AC^{k-1}	B^k	B^k
6	B^k	?	AC^{k-1}	B^k
5	B^k	?	?	AC^{k-1}
4	AC^{k-1}	B^k	?	?
3	?	?	B^k	?
2	?	?	?	B^k
1	B^k	?	?	?
0	?	?	?	?

IV. 결 론

본 논문에서는 SP 함수를 업데이트 함수로 사용하는 GFN Type I, Type II, Type III에서 비밀 S-box를 사용하였을 때, S-box에 대한 정보가 전혀 없는 공격자가 S-box를 복원하는 방법에 대하여 알아보았다. 또한, [1,2]에서 보여준 SP 구조에서 비밀 S-box의 복원에서 동차방정식이 구성되는데 반해 GFN에서는 비동차방정식이 구성되어 훨씬 수월하게 비밀 S-box가 복원됨을 알 수 있다.

m비트 비밀 S-box를 사용하였을 경우 이를 복원하는데 GFN Type I, Type II, Type III 에 대하여 각각 최대 16라운드, 6라운드, 6라운드까지 공격이 가능하고, $\frac{2^{3m}}{32k}, \frac{2^{3m}}{24k}, \frac{2^{3m}}{36k}$의 복잡도가 필요하다. 이 결과로 2.5라운드 SP 구조에 해당되는 SASAS에서의 S 계층 복원 공격보다 더 많은 라운드가 분석되어 라운드 하한을 더 크게 가져가야함을 알 수 있다.

공격자가 연산의 중간값을 모두 확인할 수 있는

화이트박스 환경을 고려하면 알려지지 않은 화이트박스의 정보가 노출되는 것은 치명적이다. 이러한 관점에서 비밀 S-box에 대한 연구는 단순히 숨겨진 S-box를 복원하는데에만 의의가 있는 것이 아니라, 화이트박스가 적용된 환경에서의 안전성을 조사할 수 있는 기틀이 되는데 의의가 있다.

현재 인테그랄 분석 기법을 일반화한 Division Property가 Y.Todo에 의해 제안된 이후 몇몇 암호들이 기존에 알려진 인테그랄 공격보다 더 많은 라운드가 분석이 되었다. 본 논문의 비밀 S-box의 복원과정이 인테그랄 공격에 기반을 두고있는 방법인 만큼 Division Property를 이용하여 비밀 S-box를 복원할 수 있는 추가적인 기법의 연구가 가능할 것으로 전망한다. Division Property를 적용한 추가적인 연구가 이루어진다면, 비밀 S-box를 적용하더라도 안전성에 위협을 받지 않도록 암호알고리즘을 설계할 수 있을 것이다.

References

[1] T.Tiessen, L.R.Knudsen, S.Kolbl, and M.M.Lauridsen, Security of the AES with a Secret S-box, Fast Software Encryption 2015, August 2015.

[2] A.Biryukov, and A.Shamir, Structural Cryptanalysis of SASAS, EUROCRYPT 2001, April, 2001.

[3] L.Knudsen, and D.Wagner, Integral Cryptanalysis (Extended Abstract), Fast Software Encryption 2002, July, 2002.

[4] National Institute of Standards and Technology, Advanced Encryption Standard, Federal Information Processing Standard (FIPS), November, 2001.

[5] National Bureau of Standards, Data Encryption Standard (DES), Federal Information Processing Standard (FIPS), 1999.

[6] Y.Zheng, T.Matsumoto, and H.Imai, On the construction of block ciphers provably secure and not relying on any unproved hypotheses. CRYPTO 1989. LNCS, vol. 435, pp. 461 - 480. Springer, Heidelberg (1990)

[7] J.Daemen, L.Knudsen, and V.Rijmen, The block cipher Square, Fast Software Encryption 97, January, 1997.

[8] Y.Todo, Integral Cryptanalysis on Full MISTY1, Fast Software Encryption 2015, August, 2015.

[9] M.Matsui, New block encryption algorithm MISTY, Fast Software Encryption 97, January, 1997.

[10] J.Park, S.Lee, J.Kim, and J.Lee, Korea Internet and Security Agency, The SEED encryption algorithm, RFC 4269, 2005.

[11] D.Hong, J.Sung, S.Hong, J.Lim, S.Lee, B.Koo, C.Lee, D.Chang, J.Lee, K.Jeong, H.Kim, J.Kim, and S.Chee, HIGHT: A New Block Cipher Suitable for Low-Resource Device, International Workshop on Cryptographic Hardware and Embedded Systems, October, 2006.

[12] D.Hong, J.Lee, D.Kim, K.H.Ryu, and D.Lee, LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors, International Workshop on Information Security Applications, 2013.

[13] D.Kwon, J.Kim, S.Park, S.H.Sung, Y.Sohn, J.H.Song, Y.Yeom, E.Yoon, S.Lee, J.Lee, S.Chee, D.Han, and J.Hong, New Block Cipher: ARIA, International Conference on Information Security and Cryptology, 2003.

[14] E. Biham, Cryptanalysis of Patarin's 2-Round Public key System with S-boxes(2R), EUROCRYPT 2000, May, 2000.

[15] T.Shirai, K.Shibutani, T.Akishita, S.Moriai, and T.Iwata, The 128-bit Blockcipher CLEFIA, Fast Software Encryption 2007, March, 2007.

〈저자소개〉

이 용 성 (Yongseong Lee) 학생회원
2015년 2월: 고려대학교 수학과 학사
2016년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 암호 알고리즘 설계 및 분석

강 형 철 (Hyngchul Kang) 학생회원
2010년 2월: 고려대학교 산업시스템정보공학과 학사
2010년 3월~현재: 고려대학교 정보보호대학원 석박사통합과정
〈관심분야〉 블록 암호화 해쉬 함수 설계 및 분석, 인증 암호화 모드 설계

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 조교수
〈관심분야〉 암호 알고리즘 설계 및 분석

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 전임강사, 조교수, 부교수, 교수
〈관심분야〉 암호 알고리즘 설계 및 분석

홍 석 희 (SeokHie Hong) 종신회원
1995년: 고려대학교 수학과 학사
1997년: 고려대학교 수학과 석사
2001년: 고려대학교 수학과 박사
1999년 8월~2004년 2월: ㈜시큐리티 테크놀로지 선임연구원
2003년 3월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U. Leuven ESAT/SCD-COSIC 박사후 연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키 및 공개키 암호 알고리즘, 부채널 공격 및 대응기법, 디지털 포렌식