情報保護學會論文誌
第17卷 第6號, 2007. 12

블록 기반 스트림 암호 TWOPRIME에 대한 연관키 차분 공격*

김 구 일[1†], 김 종 성[1‡], 성 재 철[2], 홍 석 희[1], 임 종 인[1]
[1]고려대학교 정보보호기술연구센터, [2]서울시립대학교 수학과

Related-Key Differential Attacks on the Block-wise Stream Cipher TWOPRIME

Guil Kim[1†], Jongsung Kim[1‡], Jaechul Sung[2], Seokhie Hong[1], Jongin Lim[1]
[1]CIST, Korea University, [2]Department of Mathematics, University of Seoul,

요 약

본 논문에서는 블록 기반 스트림 암호 TWOPRIME에 대한 연관키 차분 공격을 소개한다. 먼저, TWOPRIME에 대한 여러가지 연관키 차분 특성을 발표한 후, 그들을 이용하여 TWOPRIME에 사용된 연관키를 2^{14} 기지 평문과 2^{38} 8-비트 테이블 lookup의 계산량으로 복구할 수 있음을 보인다.

ABSTRACT

In this paper we present related-key differential attacks on the block-wise stream cipher TWOPRIME. We construct various related-key differentials of TWOPRIME and use them to show that recovering related keys of TWOPRIME can be performed with a data complexity of 2^{14} known plaintext blocks and a time complexity of 2^{38} 8-bit table lookups.

Keywords : Stream Ciphers, Related-key differential attacks, TWOPRIME

Ⅰ. 서 론

현재까지 스트림 암호 설계 방법은 크게 두 가지로 요약될 수 있다. 비선형 부울 함수를 포함하는 LFSR(Linear Feedback Shift Register)에 기반한 스트림 암호, 블록 암호에 기반한 OFB(Output Feedback), CFB(Cipher Feedback), CTR(Counter)모드가 있다. LFSR에 기반한 스트림 암호는 하드웨어에서 효율적이고, 수학적 평가를 할 수 있다는 장점이 있으나, 대수적 공격 방법에 취약하다고 알려져 있다. 또한 블록 암호 기반 스트림 암호는 블록 암호의 안전성에 기반하여 쉽게 설계할 수 있다는 장점이 있으나, 효율성에 대한 문제가 있다. 최근 소프트웨어 기반 스트림 암호의 필요성에 따라 다양한 블록 기반 스트림 암호가 제안되었다. 블록 기반 스트림 암호는 대체로 블록 암호의 설

접수일: 2007년 6월 19일; 채택일: 2007년 9월 17일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (IITA-2006-(C1090-0603-0025)).
† 주저자: okim912@cist.korea.ac.kr
‡ 교신저자: joshep@cist.korea.ac.kr

계 방법을 기반으로 한다. 하지만 블록 암호의 안전성 평가 방법은 AES 프로젝트[4] 등을 통해 지속적인 발전을 해온 반면 최근에 제안 되고 있는 블록 기반 스트림 암호의 안전성 평가는 초기 수준에 머무르고 있으며, 학계에서 커다란 관심을 가지고 있다. 본 논문에서는 블록 기반 스트림 암호의 안전성을 평가할 수 있는 연관키 차분 공격에 대한 일반적인 개념을 소개한다. 연관키 차분 공격은 공격자가 다르지만 연관된 키를 이용하여 평문/암호문 쌍을 얻을 수 있다는 가정 하에 출발한다. 이 공격법에서는 공격자가 연관된 키에 대한 관계식을 획득할 수 있다. 이러한 공격 유형은 현재까지 대칭키 암호, 특히, 블록 암호에 강력하게 적용되어 왔다[1,2,7,8,9]. 블록 암호와 스트림 암호에 대한 연관키 공격 사이의 차이는 블록 암호의 연관키 공격은 선택 평문 공격 시나리오를 사용하는 반면 스트림 암호의 연관키 공격은 기지 평문 시나리오를 사용한다는데 있다.

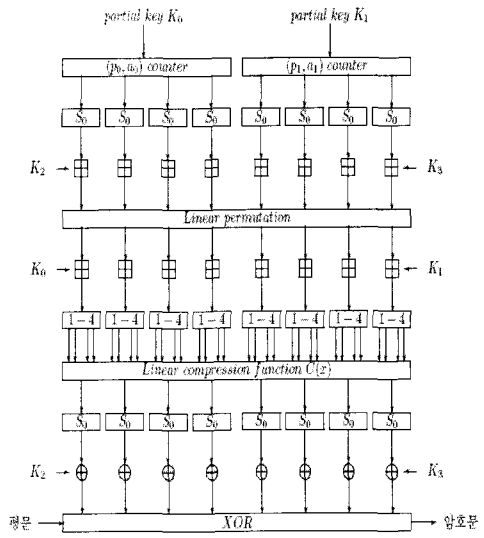
본 논문에서는 블록 기반 스트림 암호에 대한 연관키 공격의 예로, TWOPRIME에 대한 연관키 차분 공격을 소개한다. TWOPRIME의 연관키 차분 특성을 사용하여 TWOPRIME의 구조적인 문제를 이용한 기존의 TWOPRIME 공격[3]과는 다른 관점으로 마스터키를 복구하는 방법을 제시한다.

Ⅱ. TWOPRIME의 소개

1997년 핀란드의 C. Ding, V. Niemi, A. Renvall, A. Salomaa에 의해 소개된 TWOPRIME[5]은 128 비트 키 길이를 가지는 블록 기반 스트림 암호이다. Nonce를 사용하지 않으며, 한 라운드 과정을 통해 64 비트 키 수열을 생성한다. [그림 1]은 TWOPRIME의 전체 구조를 설명한다.

키 수열 생성은 총 9개의 과정으로 구성되어 있으며, 128 비트 마스터키를 $K=k_0k_1...k_{15}$라고 가정할 때, 각 라운드에 사용 되는 작업 키 K_0, K_1, K_2, K_3은 다음과 같다.

$$K_0 = k_8 + k_9 2^8 + k_{10} 2^{16} + k_{11} 2^{24}$$
$$K_1 = k_{12} + k_{13} 2^8 + k_{14} 2^{16} + k_{15} 2^{24}$$
$$K_2 = (k_0, k_1, k_2, k_3)$$
$$K_3 = (k_4, k_5, k_6, k_7).$$



〔그림 1〕 TWOPRIME의 전체 구조

평문에 대한 암호화 과정은 다음 9 단계로 이루어진다 ([그림 1] 참조).

1. 두 개의 (p,a) counter로 구성 된다. 고정된 소수 p와 상수 a를 사용하여 두 번째 라운드의 입력 값을 결정한다. 첫 번째 과정의 입력 값을 k라 할 때, i 번째 라운드의 출력 값은 $r_i = a*i + k \bmod p$이다. 단, gcd(a, p)=1를 만족하는 상수 a를 사용하여 주기 p를 갖는 출력 값을 얻을 수 있다. (p_0, a_0) counter는 $p_0=4294967279=2^{32}-17$, $a_0=2345986071$의 값을 (p_1, a_1) counter는 $p_1=4294967291=2^{32}-5$, $a_1=3124567807$의 값을 사용한다.

2. 64비트 입/출력을 갖는 비선형 함수로 8개의 S0 함수로 구성되어 있다. 64비트 입력 값은 8개의 바이트로 나누어지며, 각 바이트는 S0 함수의 입력 값이다.

$$S0(x) = [(x^{255} mod\ 257) mod\ 256]$$

3. 작업 키 K_2, K_3를 사용하여 바이트 단위 덧셈 (mod 256)을 수행한다.

4. 64비트 입력 값을 $X_0,...,X_7$라 가정할 때, 다음 과

정을 수행하여 64비트 $Y_0,...,Y_7$를 출력한다.

$$Y_j = (\sum_{i=0}^{7} X_i) - X_j, \quad j = 0, 1, ..., 7$$

5. 작업 키 K_0, K_1를 사용하여 바이트 단위 덧셈 (mod 256)을 수행한다.

6. 1-4는 비선형 확장 함수이며, 각 바이트는 다음과 같은 과정으로 4개의 바이트로 확장한다.

$$\begin{aligned} S1(x) &= [x^3 mod\ 257] mod\ 256 \\ S2(x) &= [x^{171} mod\ 257] mod\ 256 \\ S3(x) &= [45^x mod\ 257] mod\ 256 \\ S4(x) &= \begin{cases} [log_{45} x\ mod\ 257] mod\ 256, & if\ x \neq 0 \\ 128, & if\ x = 0 \end{cases} \end{aligned}$$

7. 이 과정은 압축 함수이다. 32바이트 $X_0,...,X_{31}$은 다음과 같은 과정을 수행하여 8바이트 $Y_0,...,Y_7$을 출력한다.

8. 두 번째 과정과 동일하다.

9. 작업 키 K_2, K_3를 사용하여 비트 단위 덧셈을 수행하여 64비트 키 수열을 출력한다. 생성된 64비트 키 수열은 64비트 평문과 비트 단위 덧셈을 수행하여 암호문을 생성한다.

$$\begin{aligned} Y_0 &= X_0 + X_5 + X_{10} + X_{15} + X_{16} + X_{22} + X_{24} + X_{30} \\ Y_1 &= X_1 + X_6 + X_{11} + X_{12} + X_{17} + X_{23} + X_{25} + X_{31} \\ Y_2 &= X_2 + X_7 + X_8 + X_{13} + X_{18} + X_{20} + X_{26} + X_{28} \\ Y_3 &= X_3 + X_4 + X_9 + X_{14} + X_{19} + X_{21} + X_{27} + X_{29} \\ Y_4 &= X_{16} + X_{21} + X_{26} + X_{31} + X_0 + X_6 + X_8 + X_{14} \\ Y_5 &= X_{17} + X_{22} + X_{27} + X_{28} + X_5 + X_{11} + X_{13} + X_3 \\ Y_6 &= X_{18} + X_{23} + X_{24} + X_{29} + X_{10} + X_{12} + X_2 + X_4 \\ Y_7 &= X_{19} + X_{20} + X_{25} + X_{30} + X_{15} + X_1 + X_7 + X_9 \end{aligned}$$

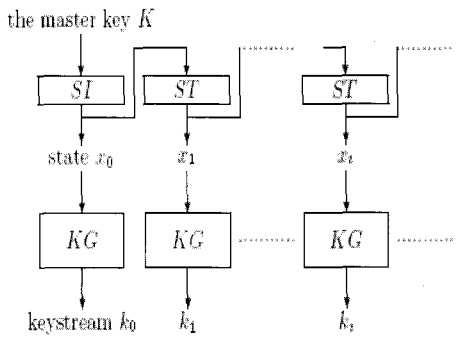
Ⅲ. 블록 기반 스트림 암호의 연관키 차분 공격

본 장에서는 블록 기반 스트림 암호의 연관키 차분 공격의 일반적인 형태를 살펴보고, 연관키 차분 공격에 안전한 블록 기반 스트림 암호의 설계 방법을 제시한다.

블록 기반 스트림 암호에 대한 연관키 차분 공격에 대한 설명에 앞서 블록 기반 스트림 암호의 일반적인 구성 방법에 대해 살펴보자. 일반적으로 블록 기반 스트림 암호는 블록 암호와 몇 가지 관점에서 다르다. 블록 기반 스트림 암호는 각 t에 대해 키 수열을 생성하는 동안 업데이트 되는 비밀 상태 값 또는 메모리를 가지고 있다. 특정 t에 대해서 비밀 상태 값은 블록 기반 스트림 암호의 키 수열을 생성하기 위한 초기 값이 되어 진다. 따라서 일반적으로 블록 기반 스트림 암호는 크게 세 가지로 구성할 수 있다. 상태 값 초기화 과정과 상태 값 업데이트 과정, 키 수열 생성과정이 그것이다. 마스터 키 또는 seed를 K로 표기하고, K에 의해 수행되는 상태 값 초기화 과정, i번째 상태 값 x_i에 의한 상태 값 업데이트 과정, i번째 상태 값 x_i에 의한 키 수열 생성 과정을 각각 SI(K), ST(x_i), KG(x_i)로 표기하자. 실제로 ST와 KG는 가령 HELIX[6]와 같이 동시에 수행될 수 있다. 또한 TWOPRIME[5]과 같이 전 상태 값과 독립적으로 수행될 수도 있으며, ST와KG 과정에 K를 사용할 수도 있다. 블록 기반 스트림 암호의 일반적인 구성 방법은 [그림 2]에 나타나 있다.

블록 기반 스트림 암호의 연관키 차분 공격은 SI와 KG의 차분 특성 확률이 높은 연관키를 선택하여 구성할 수 있다. 블록 기반 스트림 암호의 연관키 차분 공격은 다음과 같이 묘사할 수 있다.

SI에 대해 확률 p로 만족하는 연관키 차분 특성 $\alpha \rightarrow \beta$이 존재한다고 가정하자. 즉, $Pr_K[SI(K) \oplus SI(K^*) = \beta | K \oplus K^* = \alpha] = p$ (단, K와 K^*는 다르지만, 연관된 키이다). 그리고 KG에 대해 확률 q로 만족하는 연관키 차분 특성 $\beta \rightarrow \gamma$가 존재한다고 가정하자.



[그림 2] 블록 기반 스트림 암호의 일반적인 구성 방법

즉, $Pr_{xi}[KG(x_i)\oplus KG(x_i^*)= \gamma|x_i\oplus x_i^*= \beta]=q$. KG에 마스터키가 사용되는 경우에, $KG(x_i)$와 $KG(x_i^*)$는 연관키 K와 K^*를 사용하여 생성한다. 스트림 암호의 한 블록 당 생성하는 키 수열의 비트수를 n이라고 가정하자(즉, k_i는 n 비트 스트링). 그리고 $pq > 2^{-n}$를 만족한다고 가정하자. 이러한 조건이 성립하면, 2^{-n}보다 큰 확률을 갖는 연관키 K^*에 대한 첫 번째 n비트 키 수열을 예측할 수 있다. 따라서 알고리즘 설계자는 블록 기반 스트림 암호의 설계 시 예측 성질의 존재 여부를 고려해야 한다.

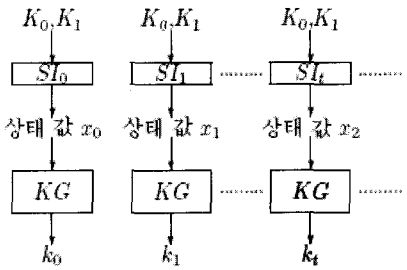
블록 기반 스트림 암호의 연관키 차분 공격은 연관키 차분 취약키 집합의 개념으로 해석할 수 있다. $q>2^{-n}$을 만족하는 $Pr_{x0}[KG(x_0)\oplus KG(x_0^*)= \gamma|x_0\oplus x^{0*}= \beta]=q$을 가정한다. 만약 공격자가 차분 β를 갖는 x0, x_0^*에 대응하는 q^{-1}개의 마스터 키를 선택할 수 있다면(즉, $SI^{-1}(x_0)$, $SI^{-1}(x_0^*)$), 이는 연관키 차분 취약키 집합이 된다. 다시 말해서 이 연관키 취약키 집합 중 적어도 하나의 키 쌍은 키 수열 쌍 $k_0\oplus k_0^*= \gamma$을 생성할 것이다.

스트림 암호에 대한 가장 잘 알려진 분석 방법은 선형 공격의 일종인 correlation 공격[10]이다. 본 장에서 표현하는 공격의 주요 특징은 블록 암호의 차분 분석을 블록 기반 스트림 암호에 적용한 것이다.

연관키 차분 공격의 관찰을 통해 본 연관키 차분 공격에 안전한 블록 기반 스트림 암호의 설계 원리는 다음과 같다. 먼저 블록 기반 스트림 암호에 사용하는 키는 빈번한 rekeying 과정을 통해 공격자로 하여금 키를 제어할 수 없도록 해야 한다. 각 t에 대해 n비트 키 수열을 생성하는 블록 기반 스트림 암호의 SI와 KG의 차분 특성 확률은 2^{-n}에 근접해야 한다. 또한 공격자로 하여금 초기 상태 값 x_0로부터 마스터키를 복구하지 못하도록 SI는 일방향 함수를 사용해야 한다.

Ⅳ. TWOPRIME의 연관키 차분 공격

본 장에서는 블록 기반 스트림 암호 TWOPRIME에 대한 연관키 차분 공격을 소개한다. TWOPRIME의 연관키 차분 공격의 표현을 용이하게 하기 위해서 TWOPRIME 구조를 세분한다. 64비트 키 수열을 생성하는 각 단계 t에 대해서 9단계 과정을 SIt(State Initialization)와 KG(Keystream Generation)으로 세분한다. 즉 ,SI_t는 t번 째 64비트키 수열 생성 과정 중 첫 번째와 두 번째 단계를, KG는 세 번째 부터 아홉 번째 단계를 표현한다. 암호문 생성 과정인 열 번째 단계는 KG에 포함하지 않는다. [그림 3]은 TWOPRIME의 키 수열 $k_0,k_1,...,k_t$ 생성과정을 SIt, KG로 세분하여 표현한다.



〔그림 3〕 TWOPRIME의 일반적 표현

본 장에서 사용하는 표기법은 다음과 같다. $x_i^{(j)}$ $(0\le i\le 7,\ 1\le j\le 10)$은 j번째 단계의 i번째 바이트 출력값을 표현한다. $X^{(j)}$은 j번째 단계의 8 바이트 출력 값을 표현한다. t번째 64비트 키 수열 생성에 대해서는 $x_i^{(j,t)}$와 $X^{(j,t)}$을 사용한다.

먼저 TWOPRIME의 KG에 대해 확률 1로 만족하는 연관키 차분 특성을 소개하고, TWOPRIME의 연관키 차분 취약 키 집합을 구성하는 방법을 소개한다. 다음 과정은 이러한 TWOPRIME의 연관키 차분 특성을 이용하여 마스터키를 복구하는 과정을 설명한다.

4.1. TWOPRIME의 연관키 차분 특성

TWOPRIME의 차분 성질은 XOR과 법 28 덧셈으로부터 유도된다. $Z=X+Y$, $Z^*=X^*+Y^*$를 고려하자(단, X, Y와 X^*, Y^*은 8-비트 스트링이다). e_i은 i번째 비트를 제외하고 모든 자리에 0을 갖는 바이트를 표현한다. 단, 최하위 비트는 0으로 왼쪽으로 갈수록 증가한다. 따라서 XOR과 법 2^8 덧셈으로부터 유도되는 차분 성질은 다음과 같이 요약할 수 있다.

- 만약 $X\oplus X^*=e_7$와 $Y=Y^*$라면, 확률 1로 $Z\oplus Z^*=e_7$을 만족한다.
- 만약 $X\oplus X^*=e_7$와 $Y\oplus Y^*=e_7$라면, 확률 1로 $Z\oplus Z^*=0$을 만족한다.

TWOPRIME의 연관키 차분 공격에 대해 연관키에

따른 두 가지 형태의 차분 특성을 사용할 수 있다. 차분 특성의 첫 번째 형태는 128-비트 연관키 $K=(K_0,K_1,K_2,K_3)$와 $K^*=(K_0^*,K_1^*,K_2^*,K_3^*)$에서 (K_0,K_1)와 (K_0^*,K_1^*)은 다르지만 연관된 키이고 $(K_2,K_3)=(K_2^*,K_3^*)$인 형태이다. 이 경우에 모든 t에 대해서 같은 키 수열을 생성하는 연관키를 구성할 수 있다. 한편, 두 번째 형태는 128비트 연관키 $K=(K_0,K_1,K_2,K_3)$와 $K^*=(K_0^*,K_1^*,K_2^*,K_3^*)$에서 $K=(K_0,K_1,K_2,K_3)$와 $K^*=(K_0^*,K_1^*,K_2^*,K_3^*)$은 각 워드가 모두 다르지만, 연관된 키 형태이다. 이 경우에는 모든 t에 대해서 차분 $(\triangle K2,\triangle K3)$을 갖는 키 스트림 쌍을 생성하는 연관키를 구성할 수 있다.

키 생성 부분 KG에 대해 첫 번째 형태인 연관키 차분 특성을 살펴보자. 만약 연관키 차분 $(\triangle K_0,\triangle K_1,\triangle K_2,\triangle K_3)=(e_7,0,0,0,e_7,0,0,0,0,0,0,0,0,0,0,0)$와 두 번째 단계 후의 값 $X^{(2)}=(x_0^{(2)},...,x_7^{(2)})$와 $X^{*(2)}=(x_0^{*(2)},...,x_7^{*(2)})$이 차분 $(e_7,0,0,0,e_7,0,0,0)$을 만족한다면, 확률 1로 연관키에 의해 생성되는 키 수열은 동일하다. [표 1]은 단계별 차분 경로를 설명한다. [표 1]에 대한 차분 특성 경로는 앞서 설명한 XOR과 법 2^{32}에 대한 차분 성질로 부터 쉽게 유도할 수 있다. 첫 번째 형태의 연관키를 갖고 두 번째 단계의 출력 차분을 고려할 때, 동일한 키 수열을 출력하는 240가지 경우가 있다.

두 번째 형태의 연관키 차분 특성에 대해 살펴보자. 만약 연관키 차분 형태를 $(\triangle K_0,\triangle K_1,\triangle K_2,\triangle K_3)=(e_7,e_7,0,0,e_7,0,0,0,e_7,0,e_7,e_7,e_7,e_7,e_7,e_7)$으로 선택하고, $X^{(2)}=(x_0^{(2)},...,x_7^{(2)})$와 $X^{*(2)}=(x_0^{*(2)},...,x_7^{*(2)})$이 차분 $(e_7,0,0,0,e_7,0,0,0)$을 만족한다면, 연관키에 의해 생성되는 키 수열은 확률 1로 차분 $(e_7,0,e_7,e_7,e_7,e_7,e_7,e_7)$을 만족한다. [표 2]은 단계별 차분 경로를 설명한다. 두 번째 형태의 연관키를 갖고 두 번째 단계의 출력 차분을 고려할 때, 특정 차분을 갖는 키 수열을 출력하는 57826가지 경우가 있다.

4.2. TWOPRIME의 연관키 차분 취약키 집합

지금부터 전 절에서 논의한 연관키 차분 특성을 만족하는 연관키 차분 취약키 집합을 구성하는 방법을 소개한다. 먼저 차분 $(e_7,0,0,0,e_7,0,0,0,0,0,0,0,0,0,0,0)$을 갖는 연관키 (K,K^*)에 대한 연관키 차분 취약키 집합을 구성하는 방법을 소개한다.

[표 1]에 따라, 모든 t에 대해서 $Pr_X^{(2,0)},_K[KG(X^{(2,0)}) \oplus KG(X^{(2,0)*})=0|\triangle X^{(2,0)}=(\triangle K_0,\triangle K_1)=(e_7,0,0,0,e_7,0,0,0),\triangle K_3=\triangle K_4=0]=1$을 만족한다. 그러므로 만약 차분 $(e_7,0,0,0,e_7,0,0,0)$을 만족하는 연관키 (K_0,K_1)와 (K_0^*,K_1^*)이 SI_t를 통해 $X^{(2,0)}\oplus X^{*(2,0)}=(e_7,0,0,0,e_7,0,0,0)$을 만족한다면, 항상 같은 키 수열을 생성한다. 2^{64}개의 연관키 가운데 취약 연관키의 비율을 알아보기 위해 $0\le t\le 2^{13}-1$에 대해 두 부분 (K_0,K_0^*)와 (K_1,K_1^*)으로 나누어 구현을 하였다. 즉, 차분 특성 $\triangle K_0=(e_7,0,0,0)\rightarrow(\triangle x_0^{(2,t)},\triangle x_1^{(2,t)},\triangle x_2^{(2,t)},\triangle x_3^{(2,t)})=(e_7,0,0,0)$와 $\triangle K_1=(e_7,0,0,0)\rightarrow(\triangle x_4^{(2,t)},\triangle x_5^{(2,t)},\triangle x_6^{(2,t)},\triangle x_7^{(2,t)})=(e_7,0,0,0)$을 만족하는 (K_0,K_0^*)와 (K_1,K_1^*)의 개수를 평가하였다. 구현 결과 각각의 $t(0\le t\le 2^{13}-1)$에 대해 취약 연관키의 개수는 약 2^{27}개의 쌍으로 평가할 수 있었다. (K_1,K_1^*)에 대한 평가 결과 역시 동일하였다. 따라서 각각의 $t(0\le t\le 2^{13}-1)$에 대해서 $Pr_K[KG(SI_t(K)) = KG(SI_t(K^*)) \mid (\triangle K_0,\triangle K_1,\triangle K_2,\triangle K_3)=(e_7,0,0,0,e_7,0,0,0,0,0,0,0,0,0,0,0)]\approx(2^{27}/2^{32})^2=2^{-10}$을 만족한다.

유사하게, 전 절에서 논의 했던 다양한 차분 경로에 대해서 연관키 차분 취약키 집합을 구성할 수 있다.

〔표 1〕 TWOPRIME의 KG에 대한 첫 번째 형태의 연관키 차분 경로

단계 (r)	$\Delta x_0^{(r)}$	$\Delta x_1^{(r)}$	$\Delta x_2^{(r)}$	$\Delta x_3^{(r)}$	$\Delta x_4^{(r)}$	$\Delta x_5^{(r)}$	$\Delta x_6^{(r)}$	$\Delta x_7^{(r)}$	확률
2	e_7	0	0	0	e_7	0	0	0	
3	e_7	0	0	0	e_7	0	0	0	1
4	e_7	0	0	0	e_7	0	0	0	1
5	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
8	0	0	0	0	0	0	0	0	1
키 수열 $(r=9)$	0	0	0	0	0	0	0	0	1

〔표 2〕 TWOPRIME의 KG에 대한 두 번째 형태의 연관키 차분 경로

단계 (r)	$\Delta x_0^{(r)}$	$\Delta x_1^{(r)}$	$\Delta x_2^{(r)}$	$\Delta x_3^{(r)}$	$\Delta x_4^{(r)}$	$\Delta x_5^{(r)}$	$\Delta x_6^{(r)}$	$\Delta x_7^{(r)}$	확률
2	e_7	0	0	0	e_7	0	0	0	
3	0	0	e_7	e_7	0	e_7	e_7	e_7	1
4	e_7	e_7	0	0	e_7	0	0	0	1
5	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
8	0	0	0	0	0	0	0	0	1
키 수열 $(r=9)$	e_7	0	e_7	e_7	e_7	e_7	e_7	e_7	1

4.3. TWOPRIME의 연관키 차분 공격

본 절에서는 TWOPRIME의 구별공격과 키 복구 공격을 소개한다. 먼저 차분(e_7,0,0,0,e_7,0,0,0,0,0,0, 0,0,0,0)을 만족하는 연관키에 대한 구별 공격을 소개한다. TWOPRIME의 구별 공격의 단계별 과정은 다음과 같다.

1. 연관키를 선택한다.
($(\triangle K_0, \triangle K_1, \triangle K_2, \triangle K_3)=(e_7,0,0,0,e_7,0,0,0,0,$ 0,0,0,0,0,0,0)을 갖는 K와 K^*)

2. 키 $K(K^*)$의 2^{10} 블록(t에 대응)에 대한 기지 평문 $P(P^*)$에 대응하는 암호문 $C(C^*)$를 가지고, $(X^{(9,0)}, X^{(9,1)},...,X^{(9,1023)})=P\oplus C$와 $(X^{*(9,0)},X^{*(9,1)},...,X^{*(9,1023)})=P^* \oplus C^*$을 계산한다.

3. 만약 어떤 $t(0\le t\le 2^{10}-1)$에 대해서 $X^{(9,t)}= X^{*(9,t)}$을 만족한다면, TWOPRIME에 의해 생성된 키 수열이라고 출력한다. 만약 그렇지 않다면 랜덤 함수에 의해 출력된 키 수열이라고 출력한다.

전 절에서 살펴본 것과 같이 연관키 K와 K^*이 각 t에 대해서 $X^{(9,t)}=X^{*(9,t)}$을 만족할 확률은 약 2^{-10}이다. 연관키 K와 K^*이 모든 t에 대해서 $X^{(9,t)}=X^{*(9,t)}$을 만족하지 않을 확률은 $(1-2^{-10})^{1024}\approx 0.37$이다. 랜덤 함수인 경우에 이 확률은 $(1-2^{-64})^{1024}\approx 1$이다. 따라서 위의 구별 공격은 약 $0.63(\approx(1-(1-2^{-10})^{1024})^*(1-2^{-64})^{1024})$의 확률로 성공한다. 만약 공격자가 2^{11} 블록에 대한 연관키 평문쌍을 이용한다면, 공격의 성공 확률은 약 $0.86(\approx(1-(1-2^{-10})^{2048})^*(1-2^{-64})^{2048})$이다.

다음 공격 과정은 TWOPRIME의 차분 특성을 이용하여 부분키를 복구하는 방법을 소개한다. 본 소절에서는 [표 1]에 나타난 차분 특성을 예로 설명한다. TWOPRIME의 키 복구 과정은 다음과 같다.

1. 연관키를 선택한다.
($(\triangle K_0, \triangle K_1, \triangle K_2, \triangle K_3)=(e_7,0,0,0,e_7,0,0,0,0,$ 0,0,0,0,0,0,0)을 갖는 K와 K^*)

2. 키 $K(K^*)$의 2^{13} 블록(t에 대응)에 대한 기지 평문 $P(P^*)$에 대응하는 암호문 $C(C^*)$를 가지고, $(X^{(9,0)}, X^{(9,1)},...,X^{(9,8191)})=P\oplus C$와 $(X^{*(9,0)},X^{*(9,1)},...,X^{*(9,8191)})=P^* \oplus C^*$을 계산한다.

3. 모든 $t(0\le t\le 2^{13}-1)$에 대해서 $X^{(9,t)}=X^{*(9,t)}$을 만족하는 t를 저장한다. 저장된 k개의 t값을 $i_1,i_2,...,i_k$으로 표기한다.

4. 차분 (e_7,0,0,0)을 만족하는 K_0와 K_0^*를 추측하여 모든 $i_j(1\le j\le k)$에 대해 $(\triangle x_0^{(2,ij)},\triangle x_1^{(2,ij)},\triangle x_2^{(2,ij)},\triangle x_3^{(2,ij)})=(e_7,0,0,0)$을 만족하는지 평가한다. 만약 그렇다면, K_0와 K_0^*를 TWOPRIME의 올바른 부분키로 출력한다. 만약 그렇지 않다면, 단계 4로 돌아간다.

5. 차분 (e_7,0,0,0)을 만족하는 K_1와 K_1^*를 추측하여 모든 $i_j(1\le j\le k)$에 대해 $(\triangle x_4^{(2,ij)},\triangle x_5^{(2,ij)},\triangle x_6^{(2,ij)},\triangle x_7^{(2,ij)})=(e^7,0,0,0)$을 만족하는지 평가한다. 만약 그렇다면, K_1와 K_1^*을 TWOPRIME의 올바른 부분키로 출력한다. 만약 그렇지 않다면, 단계 5로 돌아간다.

위의 키 복구 공격의 데이터 복잡도는 2^{14} 블록의 연관키 기지 평문쌍이다. $X^{(9,t)}=X^{*(9,t)}$을 만족하는 k의 기댓값은 약 $2^{-10}*2^{13}=8$이기 때문에 위의 공격은 약 $2^{38}(=2*4*8*2^{32})$번의 S_0-박스 테이블 lookup을 요구한다. 더구나, k가 7보다 크거나 같을 확률은 약 $0.986(\approx 1-\sum_{i=0}^{6} {}_6C_i \cdot (2^{-10})^i \cdot (1-2^{-10})^{6-i})$이다. $k\ge 7$일 때, 옳지 않은 연관키 (K_0, K_0^*) 또는 (K_1, K_1^*)는 기껏해야 $(2^{-10})^7=2^{-70}$의 확률로 제시된다. 따라서 옳지 않은 연관키 K_0,K_1,K_0^*,K_1^*의 기댓값은 기껏해야 $2^{-70}*2^{64}=2^{-6}$개이다. 따라서 위의 공격의 성공확률은 약 0.986이다.

V. 결 론

본 논문에서는 블록 기반 스트림 암호의 연관키 차분 공격에 대한 일반적인 개념을 소개한 후, 그의 적용 사례로 블록 기반 스트림 암호 TWOPRIME에 대한 연관키 차분 공격을 발표하였다. TWOPRIME에 대한 연관키 차분 공격은 2^{14} 기지 평문과 2^{38} 8비트 테이블 lookup의 계산량으로 연관키를 복구할 수 있다. 본 논

문에서 제시한 블록 기반 스트림 암호의 일반화된 연관키 차분 공격이 스트림 암호 안전성 분석에 유용한 도구로 사용되기를 기대한다.

참고문헌

[1] E. Biham, "New Types of Cryptanalytic Attack Using Related Keys", *Journal of Cryptology*, Vol. 7, No. 4, pp. 156-171, 1994.

[2] M. Blunden, A. Escott, "Related Key Attacks on Reduced Round KASUMI", *The 8th Fast Software Encryption Workshop(FSE 2001)*, LNCS 2355, Springer-Verlag, pp. 277-285, 2001.

[3] D. Coppersmith, D. Wagner, B. Schneier, J. Kelsey, "Cryptanalysis of TWOPRIME", *The 5th Fast Software Encryption Workshop(FSE 1998)*, LNCS 1372, Springer-Verlag, pp. 32-48, 1998.

[4] J. Daemen, V. Rijndael, "The Rijndael block cipher", AES proposal, 1998.

[5] C. Ding, V. Niemi, A. Renvall, A. Salomaa, "TWOPRIME: A fast stream ciphering algorithm", *The 4th Fast Software Encryption Workshop(FSE 1997)*, LNCS 1267, Springer-Verlag, pp. 88-102, 1997.

[6] N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, S. Lucks, T. Kohno, "Helix: Fast encryption and authentication in a single cryptographic primitive", *The 10th Fast Software Encryption Workshop(FSE 2003)*, LNCS 2887, Springer-Verlag, pp. 330-346, 2003.

[7] G. Jakimoski, Y. Desmedt, "Related-Key Differential Cryptanalysis of 192-bit Key AES Variants", *SAC'03*, LNCS 3006, Springer-Verlag, pp. 208-221, 2004.

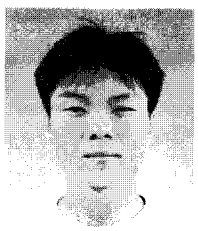
[8] J. Kelsey, B. Schneier, D. Wagner, "Key-schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES", *Advances in Cryptology - CRYPTO'96*, LNCS 1109, Springer-Verlag, pp. 237-251, 1996.

[9] J. Kelsey, B. Schneir, D. Wagner, "Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA", *Advances in Cryptology - ICICS'97*, LNCS 1334, pp. 233-246, Springer-Verlag, 1997.

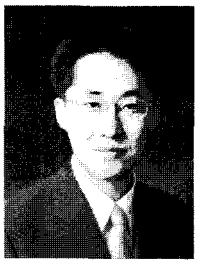
[10] W. Meier, O. Staffelbach, "Fast correlation attacks on certain stream ciphers", Journal of Cryptology, 1(3), pp. 159-176, 1989.

<著者紹介>

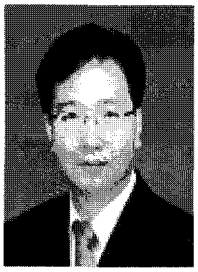
김 구 일 (Gu-il Kim) 정회원
2002년 2월 : 고려대학교 수학과 학사
2002년 9월 : 고려대학교 정보보호대학원 석사 과정
2004년 9월~현재 : 고려대학교 정보보호기술연구센터 연구원
<관심분야> 대칭키 암호의 분석 및 설계



김 종 성 (Jongsung Kim) 정회원
2000년 8월 : 고려대학교 수학과 학사
2002년 8월 : 고려대학교 수학과 석사
2006년 11월 : K.U.Leuven, ESAT/SCD-COSIC 박사
2007년 2월 : 고려대학교 정보보호대학원 박사
2007년 3월~현재 : 고려대학교 정보경영공학전문대학원 연구전임강사
<관심분야> 대칭키 암호의 분석 및 설계



성 재 철 (Jaechul Sung) 종신회원
1997년 8월 : 고려대학교 수학과 학사
1999년 8월 : 고려대학교 수학과 석사
2002년 8월 : 고려대학교 수학과 박사
2002년 7월~2004년 1월 : 한국정보보호진흥원 선임연구원
2004년 2월~현재 : 서울시립대학교 수학과 조교수
<관심분야> 대칭키 암호의 분석 및 설계



홍 석 희 (Seok-Hie Hong) 종신회원
1995년 2월 : 고려대학교 수학과 학사
1997년 2월 : 고려대학교 수학과 석사
2001년 2월 : 고려대학교 수학과 박사
1999년 8월~2004년 2월 : (주) 시큐리티 테크놀로지스 선임연구원
2003년 2월~2004년 2월 : 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월 : K.U.Leuven 박사후연구원
2005년 3월~현재 : 고려대학교 정보보호대학원 조교수
<관심분야> 대칭키 암호의 분석 및 설계, 컴퓨터 포렌식

임 종 인 (Jongin Lim) 정회원
1980년 2월 : 고려대학교 수학과 학사
1982년 2월 : 고려대학교 수학과 석사
1986년 2월 : 고려대학교 수학과 박사
1999년 2월~현재 : 고려대학교 정보보호대학원 원장, CIST 센터장
<관심분야> 암호이론, 정보보호정책