情報保護學會論文誌
第 22 卷 第 1 號, 2012. 2

스트림 암호 A5/3에 대한 오류 주입 공격*

정 기 태,[1†] 이 유 섭,[1] 성 재 철,[2‡] 홍 석 희[1]
[1]고려대학교 정보보호연구원, [2]서울시립대학교 수학과

A Fault Injection Attack on Stream Cipher A5/3*

Kitae Jeong,[1†] Yuseop Lee,[1] Jaechul Sung,[2‡] Seokhie Hong[1]
[1]Center for Information Security Technologies, Korea University
[2]Department of Mathematics, University of Seoul

요 약

본 논문에서는 GSM에서 사용되는 스트림 암호 A5/3에 대한 오류 주입 공격을 제안한다. 이 공격의 오류 주입 가정은 FDTC'05와 CISC-W'10에서 제안된 오류 주입 공격에 기반을 둔다. 본 논문에서 제안하는 오류 주입 공격은 64/128-비트 세션키를 사용하는 A5/3에 모두 적용 가능하며, 적은 수의 오류 주입을 이용하여 세션키를 복구할 수 있다. 이 공격 결과는 A5/3에 대한 첫 번째 키 복구 공격 결과이다.

ABSTRACT

In this paper, we propose a fault injection attack on stream cipher A5/3 used in GSM. The fault assumption of this attack is based on that of fault injection attacks proposed in FDTC'05 and CISC-W'10. This attack is applicable to A5/3 supporting 64/128-bit session key, respectively, and can recover the session key by using a small number of fault injections. These works are the first known key recovery attack results on A5/3.

Keywords: Side channel analysis, Fault injection attack, Cryptanalysis, A5/3, Stream cipher

I. 서 론

GSM(Global System for Mobile communication)[5]은 유럽 국가를 기준으로 하여 중국, 러시아, 인도 등 전 세계적으로 널리 채택되어 2004년 세계 이동통신 가입자 중 70% 이상이 사용하는 이동통신 방식이다. GSM에서는 보안을 위한 여러 가지 암호 알고리즘이 사용되고 있는데, 크게 데이터를 암호화할 때 사용되는 A5 암호 알고리즘, A3 인증 알고리즘, A8 세션키 생성 알고리즘의 세 부분으로 구성된다. 데이터를 암호화하는데 사용되는 A5 알고리즘은 스트림 암호 A5/1, A5/2, A5/3가 있다. A5/1은 OECD에 가입된 유럽 국가에만 사용되는 알고리즘이고, A5/2는 OECD에 가입되지 않은 국가에서 사용되는 알고리즘이다. 현재까지, 다양한 분석 기법을 이용한 A5/1, A5/2에 대한 분석 논문들이 발표되면서, 보다 강한 암호 알고리즘의 필요성이 대두되었다[2,3]. 이에 블록 암호 KASUMI[7]를 기반으로 하는 스트림 암호 A5/3가 개발되어 A5/1, A5/2를 대신하여 사용되고 있다.

최근 블록 암호의 안전성 분석 기법 중 하나로서, 암호 시스템의 실질적인 구현 과정에서 얻어지는 정보들을 이용하는 기법인 부채널 공격(side channel attack)에 대한 연구가 활발히 진행 중이다. 부채널 공격은 암호 알고리즘을 구현하였을 때 발생하는 연산

접수일(2011년 1월 25일), 게재확정일(2011년 12월 29일)
* 이 연구에 참여한 연구자(의 일부)는 '2단계BK21사업'의 지원비를 받았음
† 주저자, kite.jeong@gmail.com
‡ 교신저자, jcsung@uos.ac.kr

시간, 전력, 전자기파, 오류 등의 부가적인 정보를 이용하는 공격 방법으로서, 오류 주입 공격(fault attack), 시차 공격(timing attack), 전력 분석 공격(power attack) 등이 있다. 최초의 부채널 공격은 시차 공격으로서 Kocher에 의해 공개키 암호의 분석 방법으로 제안되었다[6]. 이 공격이 제안된 이후 전력 분석 공격, 오류 주입 공격 등의 다양한 기법들이 제안되었다.

한편, FDTC'05[4]와 CISC-W'10[1]에서는 오류 주입을 통하여 타깃 알고리즘의 라운드 수를 감소시킴으로써 AES와 Triple-DES의 비밀키를 각각 찾을 수 있음을 보였다. 이 공격의 오류 주입 가정은 "for"문과 같은 반복문에 오류를 주입하여 타깃 알고리즘의 라운드 수를 1(AES), 15(Triple-DES)로 감소시킨다는 것이다. [4]에서는 "for"문과 같은 반복문에 오류를 주입하여 라운드 수를 감소시킴으로써 AES의 비밀키를 복구하는 공격을 제안하였다. 이 공격에서는 오류 주입을 통하여 AES를 1 라운드만을 수행시킨 뒤, 라운드 1의 출력값을 암호문으로 출력한다. 이를 통해 라운드 1의 라운드 키를 복구함으로써, AES의 비밀키를 복구할 수 있다. [1]에서는 [4]에서 제안된 공격 아이디어를 Triple-DES에 적용하였다. 오류 주입을 이용하여 Triple-DES에서 동작하는 세 개의 DES 중 타깃 DES를 15 라운드만을 수행하도록 한 후, 라운드 16의 라운드 키를 복구하였다. 그 결과, 9개의 오류를 주입하여 2^{24}의 계산 복잡도로 Triple-DES의 192-비트 비밀키를 복구할 수 있다.

본 논문에서는 [1,4]의 공격 아이디어를 스트림 암호 A5/3에 적용하여 세션키를 복구할 수 있음을 보인다. 본 논문에서 제안하는 공격은 기지 평문 공격 가정 하에서 수행되며, 오류 주입을 통하여 A5/3를 구성하고 있는 블록 암호 KASUMI의 라운드 수를 감소시킨다고 가정한다. A5/3에서 사용되는 세션키의 길이는 64 ~ 128 비트이며, 본 논문에서는 64-비트 세션키를 사용하는 A5/3(A5/3-64)와 128-비트 세션키를 사용하는 A5/3(A5/3-128)에 대한 오류 주입 공격을 소개한다. 본 논문에서 제안하는 공격의 공격 결과는 [표 1]과 같다. 이 공격은 적은 수의 오류 주입을 이용하여 A5/3의 세션키를 복구할 수 있다. 본 공격 결과는 A5/3에 대한 첫 번째 키 복구 공격 결과이다.

본 논문은 다음과 같이 구성되어 있다. 먼저, 2장에서는 스트림 암호 A5/3에 대해 간략히 소개한다. 64/128-비트 세션키를 사용하는 A5/3에 대한 오류 주입 공격은 각각 3장과 4장에서 제안된다. 마지막으로 5장에서 결론을 맺는다.

[표 1] A5/3에 대한 오류 주입 공격 결과

세션키 길이	오류 주입 수	공격 복잡도
64 비트	1	$2^{45.44}$ KASUMI 암호화 연산
128 비트	2	$2^{62.89}$ KASUMI 암호화 연산

II. 스트림 암호 A5/3

GSM에서는 사용자 데이터의 기밀성을 보장하기 위하여 A5 암호 알고리즘이 사용되고 있다. GSM 초기 버전에의 A5 암호 알고리즘으로 A5/1 스트림 암호, A5/2 스트림 암호가 포함되어 있었다. 하지만 A5/1, A5/2에 대한 많은 분석 결과들이 발표되면서 GSM에 사용될 강화된 암호 알고리즘의 필요성이 대두되었다. 이에 ETSI의 SAGE(Security Algorithm Group of Experts)는 새로운 표준 암호화 알고리즘 A5/3를 개발하였다. 스트림 암호 A5/3는 블록 암호 KASUMI와 이를 기반으로 구성되는 스트림 암호 KGCORE로 구성된다.

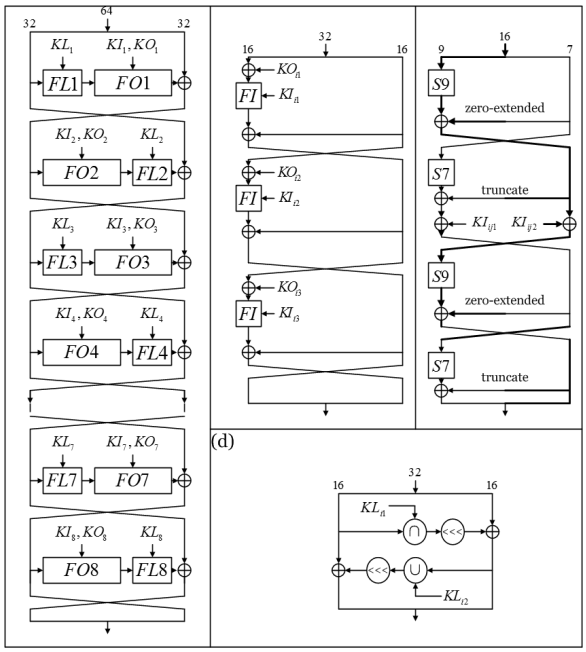
2.1 블록 암호 KASUMI

SAGE는 KASUMI를 개발하기 위하여, 아주 새로운 블록 암호를 개발하기 보다 기제안된 블록 암호인 MISTY를 이용하여 하드웨어 구현에 적합하도록 약간의 수정을 하는 방법을 사용하였다. KASUMI와 MISTY는 매우 유사한 구조를 가지기 때문에 상호간의 분석 결과를 쉽게 적용 가능한 특성을 보인다.

KASUMI는 [그림 1]-(a)와 같이 64-비트 데이터와 128-비트 비밀키를 사용하는 블록 암호로서, 8-라운드 Feistel 구조를 갖는다. 즉, 64-비트 평문 $P = L_0 \| R_0$를 입력 받아 다음과 같이 8번의 라운드 함수를 반복적으로 수행한다. 여기서, f_i는 라운드 i의 라운드 함수이고, RK_i는 라운드 i의 라운드 키 (KL_i, KO_i, KI_i)로 구성된다.

- $R_i = L_{i-1}$.
- $L_i = f_i(L_i, RK_i) \oplus R_{i-1}$ $(1 \le i \le 8)$.

라운드 함수 f_i는 32-비트 입력값 I와 128-비트



〔그림 1〕 (a) KASUMI (b) *FO* 함수 (c) *FI* 함수 (d) *FL* 함수

라운드 키를 입력 받아 32-비트 값을 출력하며, 다음과 같이 비선형 함수인 *FO* 함수와 선형 함수인 *FL* 함수로 구성된다 ($i=1,3,5,7,\ j=2,4,6,8$).

- $f_i(I,RK_i)=FO(FL(I,KL_i),KO_i,KI_i)$.
- $f_j(I,RK_j)=FL(FO(I,KO_j,KI_j),KL_j)$

FL 함수는 32-비트 입력값 $I=L\|R$과 32-비트 서브 라운드 키 $KL_i=KL_{i,1}\|KL_{i,2}$를 입력 받아 32-비트 값을 출력하는 선형 함수로서, 다음과 같이 정의된다 (〔그림 1〕-(d) 참조). 여기서, *ROL*()은 1-비트 좌측 순환 이동 연산이고, '∩'은 비트별 논리곱이며 '∪'은 비트별 논리합이다. 최종적으로, *FL* 함수는 $L'\|R'$을 출력한다.

- $R'=R\oplus ROL(L\cap KL_{i,1})$.
- $L'=L\oplus ROL(R'\cup KL_{i,2})$.

FO 함수는 32-비트 입력값 $I=L_0\|R_0$와 48-비트 서브 라운드 키 $KO_i=KO_{i,1}\|KO_{i,2}\|KO_{i,3}$와 48-비트 서브 라운드 키 $KI_i=KI_{i,1}\|KI_{i,2}\|KI_{i,3}$를 입력 받아 32-비트 값을 출력하는 비선형 함수로서, 16-비트 입·출력을 갖는 비선형 함수 *FI*로 구성된다 (〔그림 1〕-(b) 참조). $1\le j\le 3$에 대하여, *FO* 함수는 다음을 계산한 후 $L_3\|R_3$을 출력한다.

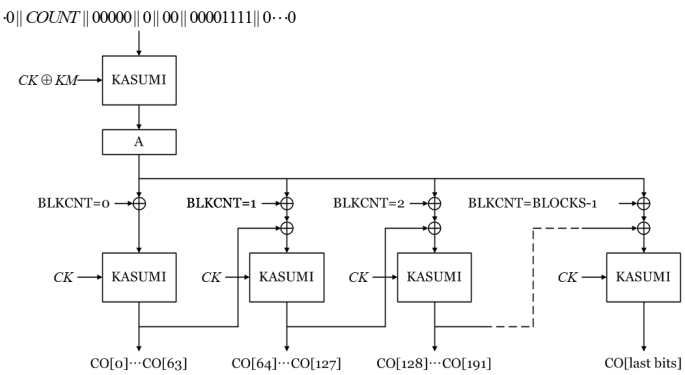
- $R_j=FI(L_{j-1}\oplus KO_{i,j},KI_{i,j})\oplus R_{j-1}$.
- $L_j=R_{j-1}$.

FI 함수는 16-비트 입력값 $I=L_0\|R_0$와 16-비트 서브 라운드 키 $KI_{i,j}=KI_{i,j,1}\|KI_{i,j,2}$를 입력 받아 16-비트 값을 출력한다. 이 함수는 두 개의 S-box *S*7,*S*9를 이용한 4-라운드 구조를 가지며, 다음과 같은 과정을 수행한다 (〔그림 1〕-(c) 참조). 여기서, *TR*()은 9-비트 입력값 중 최상위 2 비트를 버리는 함수이고, *ZE*()는 7-비트 입력값의 최상위 2 비트에 0을 추가하여 9 비트로 확장하는 함수이다. 최종적으로, *FI* 함수는 $L_4\|R_4$를 출력한다.

- $L_1=R_0$.
- $L_2=R_1\oplus KI_{i,j,2}$.
- $L_3=R_2$.
- $L_4=S7[L_3]\oplus TR(R_3)$.
- $R_1=S9[L_0]\oplus ZE(R_0)$.

〔표 2〕 KASUMI 라운드 키

i	1	2	3	4	5	6	7	8
$KL_{i,1}$	$K1 \lll 1$	$K2 \lll 1$	$K3 \lll 1$	$K4 \lll 1$	$K5 \lll 1$	$K6 \lll 1$	$K7 \lll 1$	$K8 \lll 1$
$KL_{i,2}$	$K3'$	$K4'$	$K5'$	$K6'$	$K7'$	$K8'$	$K1'$	$K2'$
$KO_{i,1}$	$K2 \lll 5$	$K3 \lll 5$	$K4 \lll 5$	$K5 \lll 5$	$K6 \lll 5$	$K7 \lll 5$	$K8 \lll 5$	$K1 \lll 5$
$KO_{i,2}$	$K6 \lll 8$	$K7 \lll 8$	$K8 \lll 8$	$K1 \lll 8$	$K2 \lll 8$	$K3 \lll 8$	$K4 \lll 8$	$K5 \lll 8$
$KO_{i,3}$	$K7 \lll 13$	$K8 \lll 13$	$K1 \lll 13$	$K2 \lll 13$	$K3 \lll 13$	$K4 \lll 13$	$K5 \lll 13$	$K6 \lll 13$
$KI_{i,1}$	$K5'$	$K6'$	$K7'$	$K8'$	$K1'$	$K2'$	$K3'$	$K4'$
$KI_{i,2}$	$K4'$	$K5'$	$K6'$	$K7'$	$K8'$	$K1'$	$K2'$	$K3'$
$KI_{i,3}$	$K8'$	$K1'$	$K2'$	$K3'$	$K4'$	$K5'$	$K6'$	$K7'$



〔그림 2〕 스트림 암호 KGCORE

- $R_2 = s7[L_1] \oplus TR(R_1) \oplus KI_{i,j,1}$.
- $R_3 = S9[L_2] \oplus ZE(R_2)$.
- $R_4 = R_3$.

KASUMI의 키스케줄은 128-비트 비밀키 $K = K1 \| K2 \| \cdots \| K8$을 입력 받아 $(128 \cdot 8)$-비트 라운드 키를 생성한다. 이를 위해 $K1', \cdots, K8'$을 다음과 같이 계산한다: $Ki' = Ki \oplus C_i$ $(i = 1, \cdots, 8)$. 여기서, 상수 C_i는 다음과 같다.

- $C_1 = 0x0123$, $C_2 = 0x4567$.
- $C_3 = 0x89AB$, $C_4 = 0xCDEF$.
- $C_5 = 0xFEDC$, $C_6 = 0xBA98$.
- $C_7 = 0x7654$, $C_8 = 0x3210$.

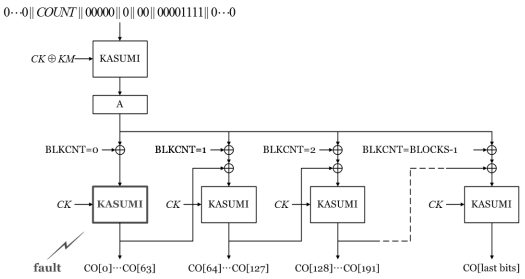
각 라운드 키는 〔표 2〕와 같이 계산된다.

2.2 스트림 암호 KGCORE

A5/3의 데이터 프레임의 암호화를 위해 사용되는 스트림 암호 KGCORE는 〔그림 2〕와 같이 KASUMI를 기반으로 64-비트 피드백을 가지는 변형된 OFB 모드로 구성된다. KGCORE는 22-비트 *COUNT*와 세션키 *CK*를 입력 받아 64 비트 단위의 키스트림 수열을 출력한다. 여기서, 키 변형 상수 *KM*은 $0x555 \cdots 5$이다. 세션키의 경우, GSM 표준에서 A8 알고리즘을 통해 생성된 64-비트 세션키를 사용하지만, A5/3를 이용하기 위해 128 비트까지의 세션키를 사용하는 것도 허용된다.

III. A5/3-64에 대한 오류 주입 공격

본 절에서는 64-비트 세션키를 사용하는 A5/3-64에 대한 오류 주입 공격을 소개한다. A5/3-64의 경우, 64-비트 세션키를 128-비트 비밀키를 사용하는 KASUMI에 적용하기 위해 64-비트 세션키를 반복 적용하여 128-비트 값으로 확장시킨다. 즉, KASUMI의 128-비트 비밀키 $K(= K1 \| K2 \| \cdots \| K8)$는 $(K1 \| \cdots \| K4 \| K1 \| \cdots \| K4)$가 된다. 본 논문에서 제안하는 공격은 기지 평문 공격 가정 하에서 수행된다. 즉, 공



〔그림 3〕 A5/3-64에 대한 오류 주입 공격 모델

격자는 임의의 평문에 대한 암호문을 얻을 수 있어서 공격자가 원하는 만큼의 길이의 키스트림 수열을 얻을 수 있다고 가정한다.

3.1 오류 주입 가정

A5/3-64에 대한 오류 주입 공격은 〔1,4〕의 오류 주입 가정을 A5/3-64를 구성하는 블록 암호 KASUMI에 적용하여 A5/3-64의 세션키를 복구한다. 즉, 첫 64-비트 키스트림 수열 $(CO[0],\cdots,CO[63])$을 생성하는 KASUMI에 오류를 주입하여, 이 블록 암호가 6 라운드만 수행하도록 한다 (〔그림 3〕 참조). 그러면 공격자는 기지 평문 공격 가정 하에서 KASUMI의 라운드 7, 8의 입·출력값을 알 수 있다 (〔그림 1〕 참조). 이 값들을 이용하여, 64-비트 세션키 CK를 복구한다.

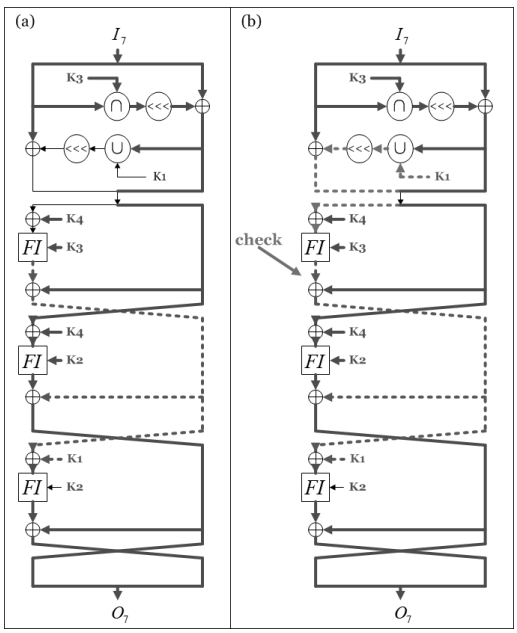
3.2 A5/3-64에 대한 오류 주입 공격

본 절에서 소개하는 공격은 1개의 오류 주입만을 이용하여 64-비트 세션키를 복구할 수 있다. 본 공격의 공격 과정은 다음과 같다.

1. 기지 평문 공격 가정 하에서, 임의의 22-비트 $COUNT$에 대해 오류를 주입하지 않은 상태에서의 64-비트 키스트림 수열 $(CO[0],\cdots,CO[63])$을 얻는다.
2. 오류를 주입한 KASUMI를 이용하여, 단계 1에서 선택된 $COUNT$에 대해 라운드 6의 출력값 O_6를 계산한다. O_6과 $(CO[0],\cdots,CO[63])$를 이용하여, 라운드 7, 8의 입·출력값 $((I_7,O_7),(I_8,O_8))$을 계산한다.
3. 48-비트 라운드 키 $(K2,K3,K4)$를 추측한 후, 각각의 추측한 라운드 키에 대해 (I_7,O_7)을 이용하여 $K1$을 계산한다.
4. 후보 라운드 키 $(K1,K2,K3,K4)$와 (I_7,O_7)를 이용하여 라운드 7에서 첫 번째 FI 함수의 출력값이 옳게 계산되는지 확인한다. 틀리게 계산되는 후보 라운드 키는 버린다.
5. 단계 4를 통과한 후보 라운드 키 $(K1,K2,K3,K4)$와 I_8을 이용하여 O_8이 옳게 계산되는지 확인한다. 옳게 계산되는 후보 $(K1,K2,K3,K4)$를 옳은 64-비트 세션키로 출력한다.

단계 1과 단계 2에서는, 64-비트 키스트림 수열을 얻기 위해 약 2번의 KASUMI 암호화 연산이 각각 필요하다. 따라서 단계 1과 단계 2의 계산 복잡도는 각각 약 2, $1.75(\approx 1+6/8)$ KASUMI 암호화 연산이다.

단계 3에서는, 추측한 $(K2,K3,K4)$와 라운드 7의 입·출력값 (I_7,O_7)을 이용하여 세 번째 FI 함수 연산 부분에서 사용되는 $K1$을 계산한다. 〔그림 4〕-(a)는 단계 3을 나타낸 것이다. 그림에서 굵은 선으로 표기된 부분은 공격자가 라운드 키 추측 등을 통해 알고 있는 값을 나타낸 것이고, 점선으로 표기된 부분은 알고 있는 값들을 이용해서 추가적으로 공격자가 알 수 있는 값을 나타낸 것이다. $K1$은 세 번째 FI 함수 연산 부분에서 XOR 연산을 통해 사용되므로, 공격자는 추측한 라운드 키 $(K2,K3,K4)$마다 1개의 $K1$을 얻을



〔그림 4〕 A5/3-64에 대한 오류 주입 공격 (a) 단계 3 (b) 단계 4

수 있다. 따라서 단계 3을 수행한 후의 후보 라운드 키의 수는 2^{48}이다. 이 단계의 계산 복잡도는 약 $2^{44.36}$ $(\approx 2^{48}\cdot 1/8\cdot 9/14)$ KASUMI 암호화 연산이다.

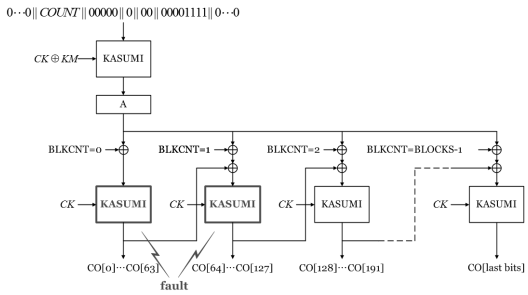
〔그림 4〕-(b)는 단계 4를 나타낸 것이다. 단계 3으로부터 얻은 2^{48}개의 후보 라운드 키와 (I_7, O_7)를 이용하여 첫 번째 FI 함수 연산 부분의 출력값을 계산할 수 있다. 틀린 후보 라운드 키를 사용했다면, 이 값이 옳게 계산될 확률은 2^{-16}이다. 따라서 단계 4를 통과하는 후보 라운드 키의 수는 $2^{32}(\approx 2^{48}\cdot 2^{-16})$이다. 이 단계의 계산 복잡도는 약 $2^{44.51}(\approx 2^{48}\cdot 1/8\cdot 10/14)$ KASUMI 암호화 연산이다.

마지막으로 단계 5에서는, 단계 4를 통과한 2^{32}개의 후보 라운드 키와 라운드 8의 입력값 I_8을 이용하여 라운드 8의 출력값 O_8이 옳게 계산되는지 확인한다. 후보 라운드 키가 단계 5를 통과할 확률은 2^{-64}이므로, 단계 5를 통과하는 후보 라운드 키의 개수의 기댓값은 $2^{-32}(\approx 2^{32}\cdot 2^{-64})$이다. 이는 틀린 후보 라운드 키가 단계 5를 통과할 확률은 매우 적음을 의미한다. 이 단계의 계산 복잡도는 약 $2^{29}(\approx 2^{32}\cdot 1/8)$ KASUMI 암호화 연산이다. 그러므로 본 절에서 소개한 A5/3-64에 대한 오류 주입 공격은 1개의 오류 주입을 이용하여 약 $2^{45.44}$ KASUMI 암호화 연산의 계산 복잡도로 64-비트 세션키를 복구할 수 있다.

IV. A5/3-128에 대한 오류 주입 공격

A5/3-128에 대한 오류 주입 공격은 A5/3-64에 대한 공격과 유사하다. 64-비트 세션키를 사용하는 A5/3-64는 세션키를 반복 적용하여 128-비트 값으로 확장시키지만, A5/3-128은 128-비트 세션키를 그대로 KASUMI의 128-비트 비밀키로 사용한다. 따라서 앞 절에서 소개한 공격 보다 더 많은 길이의 라운드 키를 추측해야하므로, 계산 복잡도가 증가한다.

A5/3-128에 대한 오류 주입 공격에서는, 〔그림 5〕와 같이 첫 128-비트 키스트림 수열 $(CO[0],\cdots, CO[127])$을 생성하는 두 개의 KASUMI에 오류를 주입하여, 각각의 블록 암호가 6 라운드만 수행하도록 한다. 이를 이용하여 기지 평문 공격 가정 하에서 라운드 7, 8의 입·출력값 $((I_7^1, O_7^1),(I_7^2, O_7^2),(I_8^1, O_8^1),(I_8^2, O_8^2))$을 각각 얻을 수 있다. 이 값들을 이용하여 A5/3-128의 128-비트 세션키를 복구한다.



〔그림 5〕 A5/3-128에 대한 오류 주입 공격 모델

앞에서 언급한 것처럼, 본 절에서 소개하는 공격은 앞 절에서 소개한 공격과 유사하기 때문에, 공격 과정과 복잡도만을 간략히 소개한다. A5/3-128에 대한 오류 주입 공격의 공격 과정은 다음과 같다.

1. 기지 평문 공격 가정 하에서, 임의의 22-비트 $COUNT$에 대해 오류를 주입하지 않은 상태에서의 128-비트 키스트림 수열 $(CO[0],\cdots, CO[127])$을 얻는다.
2. 첫 번째 KASUMI에 오류를 주입하여, 단계 1에서 선택된 $COUNT$에 대해 라운드 6의 출력값 O_6를 계산한다. 이를 이용하여, 라운드 7, 8의 입·출력값 $((I_7^1, O_7^1),(I_8^1, O_8^1))$을 계산한다.
3. 두 번째 KASUMI에 오류를 주입하여, 단계 1에서 선택된 $COUNT$에 대해 라운드 6의 출력값 O_6를 계산한다. 이를 이용하여, 라운드 7, 8의 입·출력값 $((I_7^2, O_7^2),(I_8^2, O_8^2))$을 계산한다.
4. 64-비트 라운드 키 $(K2, K4, K5, K7)$을 추측한 후, 각각의 추측한 라운드 키에 대해 (I_7^1, O_7^1)을 이용하여 $K6$을 계산한다.
5. 후보 라운드 키 $(K2, K4, K5, K6, K7)$과 (I_7^2, O_7^2)를 이용하여 단계 4에서 계산된 $K6$이 옳게 추측되었는지 확인한다. 틀리게 계산되는 후보 라운드 키는 버린다.
6. 추가적으로 $K8$을 추측한 후, 단계 5를 통과한 후보 라운드 키와 (I_8^1, O_8^1)을 이용하여 $(K1, K3)$을 계산한다.
7. $((I_7^1, O_7^1),(I_7^2, O_7^2))$를 이용하여 단계 6으로부터 계산된 후보 라운드 키 $(K1, K2, K3, K4, K5, K6, K7, K8)$을 검사한다. 틀린 후보 라운드 키는 버린다.
8. 단계 7을 통과한 후보 라운드 키와 I_8^2을 이용하

여 O_8^2이 옳게 계산되는지 확인한다. 옳게 계산되는 후보 $(K1, K2, K3, K4, K5, K6, K7, K8)$을 옳은 128-비트 세션키로 출력한다.

이 공격은 2번의 오류 주입을 이용하여 약 $2^{62.89}$ KASUMI 암호화 연산의 계산 복잡도로 128-비트 세션키를 복구할 수 있다. 틀린 후보 세션키가 단계 8을 통과할 확률은 2^{-32}이다. 이는 틀린 후보 세션키가 A5/3-128에 대한 오류 주입 공격을 통과할 확률이 매우 적음을 의미한다.

V. 결 론

본 논문에서는 A5/3에 대한 오류 주입 공격을 제안하였다. 본 논문에서는 64/128-비트 세션키를 사용하는 A5/3에 대해 각각 적용하였으며, 다른 길이의 세션키를 사용했을 경우에도 유사한 방식으로 분석이 될 것으로 예상된다. 본 논문에서 제안한 공격은 매우 적은 수의 오류 주입을 이용하여 A5/3의 세션키를 복구할 수 있으며, 본 논문에서 소개한 공격 결과는 A5/3에 대한 첫 번째 공격 결과이다.

참고문헌

〔1〕 최두식, 오두환, 배기석, 문상재, 하재철, "반복문 오류 주입을 이용한 Triple DES 차분 오류 공격", 2010년도 한국정보보호학회 동계학술대회, pp. 308-312, 2010년 12월.

〔2〕 E. Barkan, E. Biham, and N. Keller, "Instant ciphertext-only cryptanalysis of GSM encrypted communications", CRYPTO'03, LNCS 2729, pp. 600-616, 2003.

〔3〕 E. Biham and O. Dunkelman, "Cryptanalysis of the A5/1 GSM Stream Cipher", Indocrypt'00, LNCS 1977, pp.43-51, 2000.

〔4〕 H. Choukri and M. Tunstall, "Round Reduction Using Faults", Proceedings of the 2nd workshop on fault diagnosis and tolerance in cryptography - FDTC'05, pp. 13-24, Sept. 2005.

〔5〕 European Telecommunications Standards Institute (ETSI), "Digital cellular telecommunications system (Phase 2+)". Available at http://www.etsi.org.

〔6〕 P. Kocher, "Timing attacks on implementation of Diffie-Hellman", Crypto'96, LNCS 1109, pp. 104-113, 1996.

〔7〕 3GPP, "KASUMI Specification", http://www.3gpp.org/tb/other/algorithms.htm.

〈著者紹介〉

정 기 태 (Kitae Jeong) 학생회원
2004년 2월: 고려대학교 수학과 학사
2006년 2월: 고려대학교 정보보호대학원 석사
2011년 8월: 고려대학교 정보경영공학전문대학원 박사
2011년 9월~현재: 고려대학교 정보보호연구원 박사후연구원
〈관심분야〉 대칭키 암호의 분석 및 설계

이 유 섭 (Yuseop Lee) 학생회원
2007년 2월: 서울시립대학교 수학과 학사
2007년 3월~현재: 고려대학교 정보보호대학원 석박사 통합과정
〈관심분야〉 스트림 암호, 해쉬 함수의 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 8월~2004년 1월: 한국정보보호진흥원 선임연구원
2004년 2월~현재: 서울시립대학교 수학과 부교수
〈관심분야〉 암호 알고리즘 설계 및 분석

홍 석 희 (Seokhie Hong) 종신회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 8월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2003년 8월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U. Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~현재: 고려대학교 정보경영공학전문대학원 부교수
〈관심분야〉 대칭키·공개키 암호 분석 및 설계, 컴퓨터 포렌식