情報保護學會論文誌
第12券 第5號, 2002. 10

# SHACAL의 축소 라운드에 대한 확장된 부메랑 공격

김 종 성*, 문 덕 재*, 이 원 일*, 홍 석 희*, 이 상 진*

# Amplified Boomerang Attack against Reduced-Round SHACAL

JongSung Kim*, Dukjae Moon*, Wonil Lee*, Seokhie Hong*, Sangjin Lee*

요 약

SHACAL은 NESSIE 프로젝트에 발표된 블록 암호로서 국제 해쉬 표준인 SHA-1에 기반한다. SHACAL은 XOR 연산, 덧셈에 대한 modular 연산 및 비트별 계산 가능한 부울 함수를 사용한다. 이러한 연산들과 부울 함수의 사용은 차분 공격을 어렵게 만든다. 즉, 비교적 높은 확률을 가지는 긴 라운드의 차분 특성식을 찾기 힘들게 한다. 그러나 SHACAL은 높은 확률의 짧은 차분 특성식들을 가지고 있으며, 이를 이용하여 36-step 부메랑 distinguisher를 꾸밀 수 있다. 본 논문에서는 36-step 부메랑 distinguisher를 이용하여 다양한 키 길이를 가지는 SHACAL의 축소된 라운드에 대한 확장된 부메랑 공격을 소개한다. 공격 결과를 요약하면 256 비트 키를 사용하는 39-step SHACAL과 512 비트 키를 사용하는 47-step SHACAL은 확장된 부메랑 공격이 가능하다.

ABSTRACT

SHACAL is based on the hash standard SHA-1 used in encryption mode, as a submission to NESSIE. SHACAL uses the XOR, modular addition operation and the functions of bit-by-bit manner. These operations and functions make the differential cryptanalysis difficult, i.e, we hardly find a long differential with high probability. But, we can find short differentials with high probability. Using this fact, we discuss the security of SHACAL against the amplified boomerang attack. We find a 36-step boomerang-distinguisher and present attacks on reduced-round SHACAL with various key sizes. We can attack 39-step with 256-bit key, and 47-step with 512-bit key.

**Keyword** : *SHACAL, 확장된 부메랑 공격, 부메랑 distinguisher*

## Ⅰ. 서 론

SHACAL은 해쉬 함수 SHA-1을 기반으로 한 블록 암호로서 H. Handschuh와 D. Naccache에 의하여 2000년도에 개발되었다[5]. 또한 2001년도에 그들은 SHACAL의 두 버전으로 SHACAL-1와 SHACAL-2를 소개하였다. SHACAL-1은 2000년도에 소개된 SHACAL와 동일한 160비트 블록 암호이며, SHACAL-2는 해쉬 함수 SHA-2에 기반한 256비트 블록 암호이다. 본 논문은 SHACAL-1에 대한 공격 결과이며, SHACAL-1을 SHACAL이라 부르기로 하자. 이제까지 SHACAL에 대한 주요 분석 결과는 알고리즘 설계자에 의한 차분 및 선형 공격[5]으로서, 그들은 $2^{80}$미만의 기지 평문을 이용한 선형 공격과 $2^{116}$미만의 선택 평문을 이용한 차분 공격은 SHACAL에 적용 불가능함을 보였다.

본 논문에서는 36-step 부메랑 distinguisher를 소개하고, 이 distinguisher를 이용한 확장된 부메랑 공격을 다양한 키 길이를 가지는 SHACAL에 적용해 본다. 공격 결과, 256 비트 키를 사용하는

* 고려대학교 정보보호기술연구센터(CIST) ({joshep, djmoon, nice, sangjin, hsh}@cist.korea.ac.kr)

39-step SHACAL과 512 비트 키를 사용하는 47-step SHACAL은 확장된 부메랑 공격이 가능하다.

## II. 배경 지식

### 2.1 SHACAL 알고리즘

SHA는 1993년도에 NIST에 의해 소개된 해쉬 함수로서 SHA-0으로 알려졌다. 또한 1995년도에 SHA-0의 변형인 국제 표준 해쉬 함수 SHA-1이 소개되었다. SHACAL은 SHA-1에 기반을 둔 160 비트 블록 암호이다. 다음은 SHACAL에 사용되는 연산의 표기이다.

- + : $2^{32}$을 법으로 한 32비트 워드들의 덧셈
- $ROT_i(X)$ : 32비트 워드 $X$에 대한 좌측으로 I 비트 위치 순환 이동
- $\oplus$ : 배타적 논리합(XOR)
- & : 비트별 and
- | : 비트별 or

메시지에 대한 암호화 과정은 다음과 같다.

1. $X_i$가 각각 32비트 워드일 때 160비트 메시 지 $X(=X_1\|X_2\|X_3\|X_4\|X_5)$를 $A_0, B_0, C_0, D_0, E_0$에 입력한다.

$$A_0=X_1,\ B_0=X_2,\ C_0=X_3,\ D_0=X_4,\ E_0=X_5$$

2. $A_0, B_0, C_0, D_0, E_0$을 총 80-step 암호화 하여 암호문 $A_{80}, B_{80}, C_{80}, D_{80}, E_{80}$을 얻는다. 각step은 다음과 같은 암호화 과정을 거친다.
$i=1,\cdots,80$에 대해서 $i$번째 step의 암호화 과정은

$$A_i=K_i+ROT_5(A_{i-1})+f_i(B_{i-1},C_{i-1},D_{i-1})+E_{i-1}+y_i$$
$$B_i=A_{i-1}$$
$$C_i=ROT_{30}(B_{i-1})$$
$$D_i=C_{i-1}$$
$$E_i=D_{i-1}$$

이 되며, $f_i$와 $y_i$는 다음과 같다.

$$f_i(B,C,D)=(B\ \&\ C)|(\neg B\ \&\ D)\quad (1\le i\le 20)$$
$$f_i(B,C,D)=B\oplus C\oplus D\quad (21\le i\le 40, 61\le i\le 80)$$
$$f_i(B,C,D)=(B\ \&\ C)|(B\ \&\ D)|(C\ \&\ D)\quad (41\le i\le 60)$$

위의 함수 $f_i$들을 각각 $f_{if}$, $f_{xor}$, $f_{maj}$라고 표기하자. 라운드별 상수값 $y_i$은 다음과 같다.

$$y_i=5A827999_x,\quad (1\le i\le 20)$$
$$y_i=6ED9EBA1_x,\quad (21\le i\le 40)$$
$$y_i=8F1BBCDC_x,\quad (41\le i\le 60)$$
$$y_i=CA62C1D6_x,\quad (61\le i\le 80)$$

$K_i$은 $i$번째 step의 32비트 부분키이다. SHACAL의 키 스케줄은 다음과 같다. SHACAL은 최소 128 비트의 키를, 최대 512비트의 키를 받아들인다. 만약 키 길이가 512비트 미만이면 512비트의 길이가 되도록 입력된 키 다음 비트들에 모두 0으로 패딩시킨다. 512비트 키 스트링을 $K=[K_1|K_2|\cdots|K_{16}]$ 이라고 한다면, $K_i$, $(17\le i\le 80)$은 다음과 같이 생성된다.

$$K_i=ROT_1(K_{i-3}\oplus K_{i-8}\oplus K_{i-14}\oplus K_{i-16}).$$

### 2.2 확장된 부메랑 공격

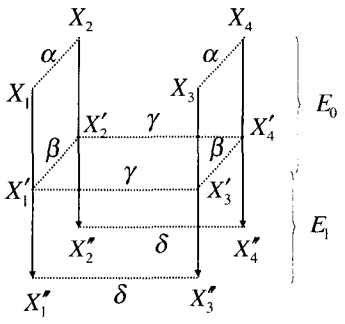
확장된 부메랑 공격은[4] 향상된 부메랑 공격으로서 선택 평문 공격이다. 확장된 부메랑 공격의 핵심 아이디어는 낮은 확률을 가지는 긴 라운드의 차분 특성식 대신에 높은 확률을 가지는 두 개의 짧은 차분 특성식들을 이용한다는데 있다.

블록 암호 $E:\{0,1\}^n\times\{0,1\}^k\to\{0,1\}^n$ 가 $E=E_1\circ E_0$으로 구성된다고 하자. $E_0$에 대해서 확률 $p$를 가지는 차분 특성식 $\alpha\to\beta$, $E_1$에 대해서 확률 $q$를 가지는 차분 특성식 $\gamma\to\delta$일 때, $pq\gg 2^{-n/2}$ 라 가정하자.

확장된 부메랑 공격은 몇 가지 차분 조건들을 만족하는 평문 quartets을 만드는데 기반을 둔다. 평문 $X_1, X_2, X_3, X_4$에 대해서 $X_1\oplus X_2=X_3\oplus X_4=\alpha$을 만족한다고 가정하자. 또한 $X_1, X_2, X_3, X_4$의 $E_0$을 통한 암호문 값들을 각각 $X_1', X_2', X_3', X_4'$

라 하고, $X_1', X_2', X_3', X_4'$의 $E_1$을 통한 암호문 값들을 각각 $X_1'', X_2'', X_3'', X_4''$라 가정하자. 만약 차분 조건 $X_1' \oplus X_2' = X_3' \oplus X_4' = \beta$와 $X_1' \oplus X_3' = \gamma$ (또는 $X_1' \oplus X_4' = \gamma$)을 만족한다면, $X_2' \oplus X_4' = (X_1' \oplus \beta) \oplus (X_3' \oplus \beta) = \gamma$ (또는 $X_2' \oplus X_3' = \gamma$)이 성립한다. 또한, $E_1$에 대한 입력 차분 $\gamma$에 대해서 출력 차분이 $\delta$가 된다면, 즉 $X_1'' \oplus X_3'' = X_2'' \oplus X_4 = \delta$ (또는 $X_1'' \oplus X_4'' = X_2'' \oplus X_3'' = \delta$)을 만족한다면, 위의 모든 차분 조건들을 만족하는 quartet를 올바른 quartet로 부른다. 올바른 quartet에 대한 묘사는 [그림 1]과 같다.

입력 차분 $\alpha$를 만족하는 평문쌍 $m$가 주어졌을 때, 올바른 quartet가 발생하는 기대치를 계산해 보자. 평문쌍 $m$개에 대해서 $E_0$를 통과하여 출력 차분이 $\beta$가 될 기대 개수는 대략 $mp$개 이며, $E_0$에 대한 차분 조건을 만족할 quartets의 기대 개수는 $\binom{mp}{2} \simeq \frac{(mp)^2}{2}$이 된다. 또한 $E_1$의 입력 차분 $X_1 \oplus X_3 = \gamma$ 또는 $X_1 \oplus X_4 = \gamma$가 될 확률은 $2^{1-n}$이 된다. 따라서 $E_1$에 대한 차분 특성식 $\gamma \rightarrow \delta$일 확률이 $q$이므로, 입력 차분 $\alpha$를 만족하는 평문쌍 $m$개에 대한 올바른 quartet가 될 기대 개수는 $\binom{mp}{2} \cdot 2^{1-n} \cdot q^2 \simeq m^2 \cdot 2^{-n} \cdot (pq)^2$이 된다. 한편, 랜덤한 permutation에 대한 올바른 quartet의 기대되는 개수는 대략 $m^2 \cdot 2^{-2n}$가 되므로, $pq \gg 2^{-n/2}$일때 블록 암호 $E$와 랜덤 permutation을 구별 할 수 있는 부메랑 distinguisher를 가진다.




[그림 1] 부메랑 distinguisher

## Ⅲ. SHACAL에 대한 확장된 부메랑 공격

본 장에서는 SHACAL에서 사용된 두 개의 연산과 세 개의 불함수에 대한 차분 성질들을 살펴본다. 그리고 이러한 차분 성질들을 이용한 36-step 부메랑 distinguisher를 소개하고 SHACAL에 대한 확장된 부메랑 공격을 적용한다.

### 3.1 SHACAL에 대한 차분 성질들

SHACAL에 대한 차분 확률을 발생시키는 요인은 첫 번째로 XOR과 modular 덧셈을 동시에 사용하는 것과 두 번째로 세 가지 불함수 $f_{if}, f_{xor}, f_{maj}$의 사용에 기인한다.

첫 번째로, XOR과 modular 덧셈의 사용으로 인한 차분 확률에 대해 고려해 보자. $X$, $Y$, $Z$, $X^*, Y^*, Z^*$가 32 비트 워드일때, $Z = X + Y$, $Z^* = X^* + Y^*$라고 하자. X와 Y가 단지 $i^{th}$ 비트($0 \le i \le 31$)위치에서만 다를때, $X \oplus Y = e_i$로 표기한다. 단, msb 비트는 $31^{th}$ 위치의 비트를 의미한다. 다음은 XOR 차분과 modular 덧셈 사이의 4가지 관계를 나타낸다.

1. $X \oplus X^* = e_{31}$이고 $Y = Y^*$일 때, 확률 1로 $Z \oplus Z^* = e_{31}$이 성립한다.
2. $X \oplus X^* = e_{31}$이고 $Y \oplus Y^* = e_{31}$일 때, 확률 1로 $Z = Z^*$이 성립한다.
3. $X \oplus X^* = e_j$이고 $Y = Y^*$일 때, 확률 1/2로 $Z \oplus Z^* = e_j$로 성립한다. (단, $0 \le j \le 30$)
4. $X \oplus X^* = e_j$이고 $Y \oplus Y^* = e_j$일 때, 확률 1/2로 $Z = Z^*$이 성립한다. (단, $0 \le j \le 30$)

두 번째로, 부울 함수 $f_{if}, f_{xor}, f_{maj}$에 대한 차분 성질들을 살펴보자. 이러한 함수들은 3 비트를 입력값으로 받아 1 비트를 출력하는 함수로 볼 수 있다. [표 1]은 위의 함수들의 XOR 차분 분포를 나타낸다.

[표 1] f-함수의 차분 분포

| $x$ | $y$ | $z$ | $f_{xor}$ | $f_{if}$ | $f_{maj}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0/1 | 0/1 |
| 0 | 1 | 0 | 1 | 0/1 | 0/1 |
| 1 | 0 | 0 | 1 | 0/1 | 0/1 |
| 0 | 1 | 1 | 0 | 1 | 0/1 |
| 1 | 0 | 1 | 0 | 0/1 | 0/1 |
| 1 | 1 | 0 | 0 | 0/1 | 0/1 |
| 1 | 1 | 1 | 1 | 0/1 | 1 |

첫 번째 세 개의 열은 $x, y, z$ 각각이 한 비트로서 8가지 가능한 입력 차분을 나타낸다. 다음의 세 개의 열은 입력 차분에 대응하는 출력 차분의 분포를 나타낸다. 마지막 세 개의 열에서 1(또는 0)은 입력 차분에 대응하는 출력 차분이 항상 1(또는 0)이 됨을 의미하며, 0/1은 입력 차분에 대응하는 출력 차분이 0과 1이 각각 반씩 된다는 의미이다.

### 3.2 36-step 부메랑 distiguisher

앞 절의 차분 성질들을 이용하여 SHACAL에 대한 부메랑 distingusher를 이루는 두 개의 차분 특성식을 꾸밀 수 있다. 즉, 첫 번째 차분 특성식 $\alpha \rightarrow \beta$는 $\alpha=(0, e_{22}, e_{15}, e_{10}, e_5)$, $\beta=(e_{2,7,14,24,29}, e_{19}, e_{12}, e_7, e_2)$ 이며, 확률 $p(=2^{-45})$인 $1^{th}$부터 $21^{th}$ step까지의 특성식이다. 두 번째 차분 특성식 $\gamma \rightarrow \delta$는 $\gamma=(e_{1,5,8}, e_{1,3,5}, e_{3,13}, e_{1,5,13,31}, e_{6,10,13,31})$, $\delta=(e_{9,19,29,31}, e_{14,29}, e_{7,29}, e_2, e_{29})$이며, 확률 $q(=2^{-31})$인 $22^{th}$부터 $36^{th}$ step까지의 특성식이다. 여기서 차분 기호 $e_{i_1,\cdots,i_k}$는 $e_{i_1} \oplus \cdots \oplus e_{i_k}$을 의미한다. [표 2]는 첫 번째 차분 특성식을 나타낸다. [표 2]의 첫 번째 행은 첫 번째 step의 입력 차분을 나타내며, $i^{th}$ step에 해당하는 두 번째 열부터 여섯 번째 열까지는 $i^{th}$ step의 출력 차분을 나타내며, 일곱 번째 열은 $(i-1)^{th}$ step의 출력 차분이 $i^{th}$ step의 출력 차분이 될 확률을 나타낸다. $1^{th}$부터 $20^{th}$ step까지는 부울 함수 $f_{if}$를 $21^{th}$ step에서는 부울 함수 $f_{xor}$ 가 사용되며, [표 2]의 확률 값은 앞절에서 살펴본 차분 성질들을 이용하면 쉽게 계산 할 수 있다. 그리고 [표 3]은 두 번째 차분 특성식을 나타낸다.

### 3.3 공격 과정

본 절에서는 앞 절에서 제시된 36-step 부메랑 distinguisher를 이용하여 다양한 키 길이를 가지는 SHACAL의 축소된 라운드에 대한 확장된 부메랑 공격을 소개한다.

$S=E_f \circ E=E_f \circ E_1 \circ E_0$을 축소된 라운드의 SHACAL로 볼 경우, $E_0$가 $1^{th}$부터 $21^{th}$ step까지, $E_1$이 $22^{th}$부터 $36^{th}$ step까지 나타낸다고 하자. 찾고자 하는 키 요소는 $E_f$의 부분키이다. $E_0$에 사용된 첫 번째 차분 특성식 $\alpha \rightarrow \beta$는 확률 $p(=2^{-45})$을 가지며, $E_1$에 사용된 두 번째 차분 특성식 $\gamma \rightarrow \delta$는 확률 $q(=2^{-31})$를 가진다. $\alpha, \beta, \gamma$와 $\delta$는 앞 절에서 설명되었다. 따라서 확률 $pq \gg 2^{-75}$인 36-step 부메랑 distinguisher를 만들 수 있다.

[표 2] SHACAL에 대한 첫 번째 차분 특성식

| step | $\triangle A$ | $\triangle B$ | $\triangle C$ | $\triangle D$ | $\triangle E$ | Prob |
|---|---|---|---|---|---|---|
| | 0 | $e_{22}$ | $e_{15}$ | $e_{10}$ | $e_5$ | |
| 1 | $e_5$ | 0 | $e_{20}$ | $e_{15}$ | $e_{10}$ | $2^{-4}$ |
| 2 | 0 | $e_5$ | 0 | $e_{20}$ | $e_{15}$ | $2^{-3}$ |
| 3 | $e_{15}$ | 0 | $e_3$ | 0 | $e_{20}$ | $2^{-3}$ |
| 4 | 0 | $e_{15}$ | 0 | $e_3$ | 0 | $2^{-2}$ |
| 5 | 0 | 0 | $e_{13}$ | 0 | $e_3$ | $2^{-2}$ |
| 6 | $e_3$ | 0 | 0 | $e_{13}$ | 0 | $2^{-2}$ |
| 7 | $e_8$ | $e_3$ | 0 | 0 | $e_{13}$ | $2^{-2}$ |
| 8 | 0 | $e_8$ | $e_1$ | 0 | 0 | $2^{-2}$ |
| 9 | 0 | 0 | $e_6$ | $e_1$ | 0 | $2^{-2}$ |
| 10 | 0 | 0 | 0 | $e_6$ | $e_1$ | $2^{-2}$ |
| 11 | $e_1$ | 0 | 0 | 0 | $e_6$ | $2^{-2}$ |
| 12 | 0 | $e_1$ | 0 | 0 | 0 | $2^{-1}$ |
| 13 | 0 | 0 | $e_{31}$ | 0 | 0 | $2^{-1}$ |
| 14 | 0 | 0 | 0 | $e_{31}$ | 0 | $2^{-1}$ |
| 15 | 0 | 0 | 0 | 0 | $e_{31}$ | $2^{-1}$ |
| 16 | $e_{31}$ | 0 | 0 | 0 | 0 | 1 |
| 17 | $e_4$ | $e_{31}$ | 0 | 0 | 0 | $2^{-1}$ |
| 18 | $e_9$ | $e_4$ | $e_{29}$ | 0 | 0 | $2^{-2}$ |
| 19 | $e_{14}$ | $e_9$ | $e_2$ | $e_{29}$ | 0 | $2^{-3}$ |
| 20 | $e_{19}$ | $e_{14}$ | $e_7$ | $e_2$ | $e_{29}$ | $2^{-4}$ |
| 21 | $e_{2,7,14,24,29}$ | $e_{19}$ | $e_{12}$ | $e_7$ | $e_2$ | $2^{-5}$ |

입력 차분 $\alpha$를 만족하는 $m(=2^{157.5})$개의 평문쌍에 대해서 올바른 quartets의 기대되는 개수는 대략 $8(=(2^{157.5})^2 \cdot 2^{-160} \cdot (2^{-75})^2)$이다. 다음은 160 비트 이상의 키 길이를 가지는 축소 라운드의 SHACAL에 대한 공격 과정이다.

1. 입력 차분 $\alpha$를 만족하는 평문쌍 $m(=2^{157.5})$개를 선택한다. - $m$개의 평문쌍으로부터 만들 수 있는 quartets의 기대되는 개수는 대략 $m^2(=2^{315})$이다. quartet의 평문들을 $P_1 \oplus P_2 = P_3 \oplus P_4 = \alpha$을 만족하는 $(P_1, P_2, P_3, P_4)$ 로 표기하고, 각각의 $S$에 대한 암호문들을 $(C_1, C_2, C_3, C_4)$라 표기하자.

〔표 3〕 SHACAL에 대한 두 번째 차분 특성식

| step | $\triangle A$ | $\triangle B$ | $\triangle C$ | $\triangle D$ | $\triangle E$ | Prob |
|---|---|---|---|---|---|---|
| | $e_{1,5,8}$ | $e_{1,3,5}$ | $e_{3,13}$ | $e_{1,5,}$ 13,31 | $e_{6,10,}$ 13,31 | |
| 22 | 0 | $e_{1,5,8}$ | $e_{1,3,31}$ | $e_{3,13}$ | $e_{1,5,}$ 13,31 | $2^{-3}$ |
| 23 | $e_{1,8}$ | 0 | $e_{3,6,31}$ | $e_{1,3,31}$ | $e_{3,13}$ | $2^{-4}$ |
| 24 | $e_{1,3}$ | $e_{1,8}$ | 0 | $e_{3,6,31}$ | $e_{1,3,31}$ | $2^{-4}$ |
| 25 | 0 | $e_{1,3}$ | $e_{6,31}$ | 0 | $e_{3,6,31}$ | $2^{-4}$ |
| 26 | $e_1$ | 0 | $e_{1,31}$ | $e_{6,31}$ | 0 | $2^{-3}$ |
| 27 | $e_1$ | $e_1$ | 0 | $e_{1,31}$ | $e_{6,31}$ | $2^{-2}$ |
| 28 | 0 | $e_1$ | $e_{31}$ | 0 | $e_{1,31}$ | $2^{-1}$ |
| 29 | 0 | 0 | $e_{31}$ | $e_{31}$ | 0 | $2^{-1}$ |
| 30 | 0 | 0 | 0 | $e_{31}$ | $e_{31}$ | 1 |
| 31 | 0 | 0 | 0 | 0 | $e_{31}$ | 1 |
| 32 | $e_{31}$ | 0 | 0 | 0 | 0 | 1 |
| 33 | $e_4$ | $e_{31}$ | 0 | 0 | 0 | $2^{-1}$ |
| 34 | $e_{9,31}$ | $e_4$ | $e_{29}$ | 0 | 0 | $2^{-1}$ |
| 35 | $e_{14,29}$ | $e_{9,31}$ | $e_2$ | $e_{29}$ | 0 | $2^{-3}$ |
| 36 | $e_{9,19,}$ 29,31 | $e_{14,29}$ | $e_{7,29}$ | $e_2$ | $e_{29}$ | $2^{-4}$ |

2. counter 배열을 0으로 초기화한다. - counter 배열의 개수는 $E_f$의 부분키의 가능한 개수와 같다.
3. $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta'$ 인지 체크한다. 단, $\delta'$는 $E_f$의 입력 차분이 $\delta$일 때 나올 수 있는 출력 차분이다.
4. 3번 과정을 통과한 quartets에 대해서, $E_{f_k}^{-1}(C_1) \oplus E_{f_k}^{-1}(C_3) = E_{f_k}^{-1}(C_2) \oplus E_{f_k}^{-1}(C_4) = \delta$을 만족하는 부분키 $K_f$에 대응하는 counter를 1만큼 증가시킨다.
5. 모든 counter들을 체크하고, counter의 값이 7이상인 부분키를 출력한다.

먼저, 256 비트 키를 사용하는 39-step SHACAL에 대한 확장된 부메랑 공격이 키의 전수 조사량보다 적은 복잡도로 공격 가능함을 보이겠다. 즉, 37,38,39 steps에 해당하는 $E_f$의 96 비트 $K_f$를 키의 전수 조사량보다 적은 복잡도로 찾을 수 있다. 단계 1에서 입력 차분 $\alpha$를 만족하는 $2^{157.5}$개의 평문쌍으로부터 $2^{315}$개의 quartets를 생성할 수 있다. 이러한 quartets에 대해서 단계 3을 통하여 옳지 않은 quartets을 걸러낼 수 있다. 즉, 단계 3에서 $\delta'$의 값이 $(?,?,?,e_{7,17,27,29},e_{12,27})$을 만족하지 않는 quartets들은 올바른 quartets가 될 수 없다. (?는 임의의 차분이다.) 왜냐하면, $E_f$에 대한 입력 차분이 $\delta(=(e_{9,19,29,31},e_{14,29},e_{7,29},e_2,e_{29}))$일 때 출력 차분이 $(?,?,?,e_{7,17,27,29},e_{12,27})$이 되기 때문이다. 따라서 $2^{315}$개의 quartets들이 단계 3을 통과할 비율이 $(2^{-64})^2$이므로, $2^{187}$개의 올바른 quartets 후보들이 단계 3을 통과한다. 그리고 나서 96 비트 부분키 $K_f$를 추측하여 단계 3을 통과한 quartets에 대해서 $E_f$에 해당하는 세 steps을 복호화 한다. 복호화된 quartet가 단계 4를 통과한다면, 추측된 키의 counter를 1만큼 증가시킨다. 올바른 quartets의 기대되는 개수가 대략 8이므로, 올바른 부분키 $K_f$의 counter의 기대되는 개수는 7보다 크다. 그러나 올바르지 못한 부분키에 대해서 단계 4를 통과할 quartets의 기대되는 개수가 대략 $2^{-5}(=2^{187}\cdot(2^{-96})^2)$이므로, 올바르지 못한 부분키의 counter의 기대되는 개수는 기껏해야 1이다. 따라서, 위의 공격 알고리즘을 이용하여 256 비트 키를 사용하는 39-step SHACAL을 공격할 수 있다. 공격 과정에 필요한 선택 평문의 개수는 $2^{158.5}$개 이며, 대략 $2^{250.8}(=2^{158.5}\cdot 2^{96}\cdot\frac{3}{39})$번의 39-step SHACAL의 암호화 과정이 필요하다.

또한, 위의 공격 알고리즘을 이용하여 256 비트 이상의 마스터 키를 가지는 SHACAL의 축소 라운드에 대한 확장된 부메랑 공격이 가능하다. $i=0,1,\cdots,8$에 대해서 축소된 $(39+i)$-step SHACAL이 $(256+32\cdot i)$ 비트 마스터 키를 사용한다고 가정하자. 그러면 $E_f$는 $(i+3)$ steps을 가지며, $32\cdot(i+3)$ 비트 부분키 $K_f$를 찾을 수 있다. 위와 같은 방법을 이용하여 축소된 $(39+i)$-step SHACAL을 공격하기 위해서는 $2^{158.5}$개의 선택 평문이 필요하며, 대략 $2^{252.4+32\cdot i}(\simeq 2^{158.5}\cdot 2^{32\cdot(i+3)}\cdot\frac{i+3}{39+i})$번의 $(39+i)$-step SHACAL의 암호화 과정이 필요하다. 따라서, 512 비트 마스터 키를 가지는 47-step SHACAL은 확장된 부메랑 공격이 가능하다. 또한, 128 비트를 제외한 256 비트 미만의 마스터 키를 사용하는 축소된 SHACAL에 대해서도 위의 공격 알고리즘이 적용 가능하다. 이러한 경우, 키의 전수 조사량보다 적은 복잡도로 공격하기 위해서 단계 3을 통과하는 quartets의 기대되는 개수는 $2^{156.5}$개 미만이어야 하며, 160 비트 마스터 키를 사용하는 축소된 37-step SHACAL과 192 또는 224 비트 마스터 키를 사용하는 축소된 38-step SHACAL에 대해

서 공격 가능하다. 37-step SHACAL을 공격하기 위해서는 $2^{158.5}$개의 선택 평문이 필요하며, 대략 $2^{87.8}(=2^2 \cdot 2^{315} \cdot 2^{-256} \cdot 2^{32} \cdot \frac{1}{37})$번의 37-step SHACAL의 암호화 과정이 필요하다. 그리고 38-step SHACAL을 공격하기 위해서는 $2^{158.5}$개의 선택 평문이 필요하며, 대략 $2^{184.8}(=2^2 \cdot 2^{315} \cdot 2^{-192} \cdot 2^{64} \cdot \frac{2}{38})$번의 38-step SHACAL의 암호화 과정이 필요하다.

128 비트 마스터 키를 사용하는 SHACAL에 대해서는 위의 공격 알고리즘의 36-step 부메랑 distinguisher를 사용할 수 없다. 왜냐하면, 요구할 수 있는 평문의 개수가 $2^{128}$개 미만 이어야 하기 때문이다. 따라서, 확률 $pq$가 $2^{-45.5}(=(2^3 \cdot (2^{-127})^2 \cdot 2^{160})^{1/2})$ 보다 큰 새로운 부메랑 distinguisher를 찾아야만 한다. 위의 36-step 부메랑 distinguisher를 구한 비슷한 방법으로 확률 $pq=2^{-45}$인 26-step 부메랑 distinguisher를 ($1^{st}$ step부터 $26^{th}$ step 까지) 찾을 수 있으며, 28-step SHACAL을 공격 할 수 있다. 28-step SHACAL을 공격하기 위해서는 $2^{127.5}$개의 선택 평문이 필요하며, 대략 $2^{127.2}$번의 28-step SHACAL의 암호화 과정이 필요하다.

## Ⅳ. 결 론

SHACAL은 긴 라운드에 대한 차분 특성식들은 낮은 확률을 가지며, 짧은 라운드에 대한 차분 특성식들은 비교적 높은 확률을 가진다. 이러한 사실로부터 36-step 부메랑 distinguisher을 꾸밀 수 있었으며, 36-step 부메랑 distinguisher를 이용하여 다양한 키 길이를 가지는 SHACAL의 축소된 라운드에 대한 확장된 부메랑 공격을 적용해 볼 수 있었다. 공격 결과를 요약하면 [표 4]와 같이 요약할 수 있다. [표 4]의 Time 복잡도 $n$은 주어진 축소 라운드 SHACAL에 대한 $n$번의 암호화 과정을 의미한다.

[표 4] SHACAL의 축소 라운드에 대한 확장된 부메랑 공격 결과

| 마스터 키 | Steps | Data | Time |
|---|---|---|---|
| 128 | 28 | $2^{127.5}$ | $2^{127.2}$ |
| 160 | 37 | $2^{158.8}$ | $2^{87.8}$ |
| 192, 224 | 38 | $2^{158.5}$ | $2^{148.8}$ |
| $256+32 \cdot i$ ($0 \le i \le 8$) | $39+i$ | $2^{158.5}$ | $\le 2^{252.4+32 \cdot i}$ |

## 참 고 문 헌


[1] E. Biham and A. Shamir, "Differential Cryptanalysis of the Data Encryption Standard", Springer-Verlag, 1993.

[2] David Wagner, "The boomerang Attack", proceedings of Fast Software Encryption, Lecture Notes in Computer Science 1636, pp. 156~170, Springer-Verlag, 1999.

[3] E. Biham, O. Dunkelman and N. Keller, "The Rectangle Attack-Rectangling the Serpent", Proc. of Eurocrypt' 2001, Springer-Verlag, LNCS 2045, pp. 340~357, 2001.

[4] J. Kelsey, T. Kohno, and B. Schneier, "Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent", Proc. of FSE'2000, Springer-Verlag, LNCS 1978, pp. 75~93, 2001

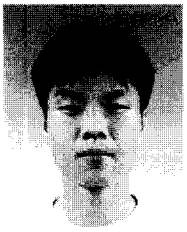
[5] H. Handschuh, D. Naccache, "SHACAL", In Proceedings of the First Open NESSIE Workshop, November 2000.

[6] H. Handschuh, D. Naccache, "SHACAL", NESSIE project, October 2001.

[7] H. Handschuh, L. R. Knudsen and M. J. Robshaw, "Analysis of SHA-1 in Encryption Mode", CT-RSA 2001, Springer-Verlag, LNCS 2020, pp. 70~83, 2001.

[8] J. Nakahara Jr, "The Statistical Evaluation of the NESSIE Submission", October 2001.

## 〈著者紹介〉

**김 종 성 (Jong-Sung Kim)**
2000년 8월 : 고려대학교 수학과 학사
2002년 8월 : 고려대학교 수학과 석사
2002년 8월~현재 : 고려대학교 정보보호대학원 박사 과정
〈관심분야〉 블록 암호 및 스트림 암호의 분석과 설계

**문 덕 재 (Duk-Jae Moon)**
2000년 2월 : 서울시립대학교 수학과 학사
2001년 3월~현재 : 고려대학교 정보보호대학원 석사과정
〈관심분야〉 블록 암호 및 스트림 암호의 분석과 설계

**이 원 일 (Won-Il Lee)**
1998년 2월 : 고려대학교 수학과 학사
2000년 2월 : 고려대학교 수학과 석사
2000년 3월~현재 : 고려대학교 수학과 박사 과정
〈관심분야〉 블록 암호 및 스트림 암호의 분석과 설계

**홍 석 희 (Seok-Hie Hong)**
1995년 2월 : 고려대학교 수학과 학사
1997년 2월 : 고려대학교 수학과 석사
2001년 2월 : 고려대학교 수학과 박사
2001년 3월~현재 : 고려대학교 정보보호기술연구센터 선임 연구원
〈관심분야〉 블록 암호 및 스트림 암호의 분석과 설계

**이 상 진 (Sang-Jin Lee)**
1987년 2월 : 고려대학교 수학과 학사
1989년 2월 : 고려대학교 수학과 석사
1994년 8월 : 고려대학교 수학과 박사
1989년 2월~1999년 2월 : 한국전자통신연구소 선임 연구원
1999년 3월~현재 : 고려대학교 자연과학대학 부교수, 고려대학교 정보보호대학원 겸임 교수, 고려대학교 정보보호기술연구센터 연구실장
〈관심분야〉 블록 암호 및 스트림 암호의 분석과 설계, 암호 프로토콜, 공개키 암호 알고리즘 분석