情報保護學會論文誌
第 21 卷 第 6 號, 2011. 12

블록 암호 PP-1/64-128에 대한 부정 차분 공격*

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

Truncated Differential Cryptanalysis on PP-1/64-128*

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

요 약

블록 암호 PP-1은 다양한 길이의 데이터 블록과 비밀키를 지원하는 SPN구조의 블록 암호이다. 또한, 암호화 과정과 복호화 과정이 동일하도록 설계되어 하드웨어에서 효율적으로 구현 가능하며 오류 탐지 기법을 적용하기에 유리하다. 본 논문에서는 PP-1/64-128에 대한 부정 차분 공격을 소개한다. 본 논문에서 제안하는 공격은 $2^{50.16}$의 선택평문과 $2^{46.16}$ 바이트 메모리를 이용하여 $2^{50.45}$의 PP-1/64-128의 암호화 연산을 통해 비밀키를 복구한다. 본 논문에서 제안 하는 PP-1/64-128에 대한 공격 결과는 현재까지 알려진 공격 결과 중 가장 좋은 결과이다.

ABSTRACT

The PP-1/64-128 block cipher support variety data block and secret key size. Also, it is suitable for hardware implementation and can much easier to apply Concurrent Error Detection(CED) for cryptographic chips compared to other block ciphers, because it has same encryption and decryption process. In this paper, we proposed truncated differential cryptanalysis of PP-1/64-128. the attack on PP-1/64-128 block cipher requires $2^{50.16}$ chosen plaintexts, $2^{46.16}$ bytes memory spaces and $2^{50.45}$ PP-1/64-128 encryption to retrieve secret key. This is the best result of currently known PP-1/64-128 differential cryptanalysis.

Keywords: Block Cipher, PP-1/64-128, Truncated Differential Cryptanalysis

I. 서 론

블록 암호 기술은 1980년대부터 DES에 대한 분석 기술 연구를 통해 급격하게 발전하였다. 이후, 미국 국립 표준원에서 주최한 AES 공모 사업을 통하여 블록 암호 개발에 관한 방법과 안전성 분석 기법이 새롭게 정립되면서, 높은 안전성과 효율성을 가지는 블록 암호에 대한 연구가 활발히 진행되었다. 대표적으로, 미국 표준 블록 암호인 AES[1], 국내 표준인 SEED[2], ARIA[3] 등의 다양한 블록 암호가 개발되었다. 하지만, 이러한 블록 암호들은 RFID나 센서 네트워크, 모바일 환경 등의 제한된 컴퓨팅 환경에 적용하는데 어려움이 발생하였다. 그리하여 2000년대 중반부터 하드웨어의 자원이 제한된 환경에 적합한 경량 블록 암호 개발에 대한 연구가 시작되었다. 이러한 연구 결과 PP-1[4] , PRESENT[5], HIGHT[6], KATAN / KTAN -TAN[7] 등이 제안되었다.

접수일(2011년 3월 15일), 수정일(2011년 7월 8일), 게재확정일(2011년 10월 31일)
* 본 연구는 지식경제부 IT R&D 사업의 일환으로 수행하였음(유비쿼터스 환경에서의 정보보호 서비스를 위한 프라이버시 강화 암호 기술 개발)
† 주저자, yphong@cist.korea.ac.kr
‡ 교신저자, hsh@cist.korea.ac.kr

블록 암호 PP-1은 다양한 길이의 데이터 블록과 비밀키를 지원하도록 설계되었다, 그리고 암호화 과정과 복호화 과정이 동일한 알고리즘으로 작동하는 인볼루션 SPN 구조를 이용하였으며, S-박스, 법덧셈, 법뺄셈, XOR의 가벼운 연산과 비트 단위 치환 함수를 사용함으로서 경량으로 하드웨어 구현이 가능하다. 이후, PP-1의 제안자는 인볼루션 구조를 가지는 블록 암호에 대한 오류 탐지 기법을 소개함으로서 다른 블록 암호에 비해 효율적인 오류 탐지가 가능함을 보였다[8].

부정 차분 공격은 1994년 Lars R. Knudsen이 제안한 공격방법으로 차분 특성을 구성할 때, 차분 특성의 출력차분을 모두 예측하지 않고 공격에 필요한 비트 정보만을 이용하여 공격하는 방법이다[9]. 기존의 차분 공격은 차분 특성을 구성할 때, 하나의 입력차분과 하나의 출력차분만 고려하지만, 부정 차분 공격에서는 하나의 입력차분에 대해, 여러 개의 출력차분을 동시에 고려함으로서 보다 높은 확률을 가지는 차분 특성을 구성한다.

블록 암호 PP-1에 대한 안전성 분석 결과로는 64-비트 데이터 블록과 128-비트 비밀키를 사용하는 PP-1/64-128에 대한 차분 공격이 유일하다[10]. 이 공격 방법에서는 $2^{51.00}$의 데이터 복잡도와 $2^{112.54}$의 계산 복잡도로 1 라운드 키의 8 비트와 11 라운드 키의 72 비트를 복구한다.

본 논문에서는 PP-1/64-128에 대한 부정 차분 공격을 제안한다. 이를 위해 $2^{-44.91}$의 확률을 가지는 9개의 8-라운드 차분 특성을 구성하였다. 그리고 이 차분 특성에 6개의 1-라운드 부정 차분 특성을 연결하여 18개의 부정 차분 특성을 구성하였다. 제안하는 공격은 구성한 18개의 부정 차분 특성을 이용하여 1 라운드 키 8 비트와 11 라운드 키 72 비트를 복구한 후, 복구한 비밀키의 정보를 통해 가능한 비밀키를 전수조사한다. 본 공격은 $2^{50.16}$개 선택 평문, $2^{45.16}$바이트 메모리를 이용하여 $2^{50.45}$번 PP-1/64-128 암호화 연산을 통하여 비밀키를 복구한다. 본 논문의 분석 결과는 현재까지 알려진 PP-1/64-128에 대한 분석 결과 중 가장 좋은 분석 결과이다. [표 1]은 PP-1/64-128에 대한 분석 결과를 비교한 결과이다.

[표 1] PP-164/128에 대한 공격 결과 비교

	[10]	본 논문
데이터 복잡도(개)	$2^{51.00}$	$2^{50.16}$
메모리 복잡도(바이트)	$2^{46.00}$	$2^{46.16}$
계산 복잡도 (PP-1/64-128 복호화 연산)	$2^{112.54}$	$2^{50.45}$

본 논문은 다음과 같이 구성된다. 2장에서는 논문에서 사용하는 표기법과 블록 암호 PP-1/64-128, 부정 차분 분석 기법을 설명한다. 그리고 공격에 사용하는 부정 차분 특성과 이를 이용한 블록 암호 PP-1/64-128에 대한 부정 차분 공격을 3장과 4장에서 소개한 후, 5장에서 결론을 맺는다.

II. 표기법 및 관련 연구

본 장에서는 논문에서 사용하는 전반적인 표기법을 정의 하고, 블록 암호 PP-1/64-128와 부정 차분 분석 기법에 대하여 소개한다.

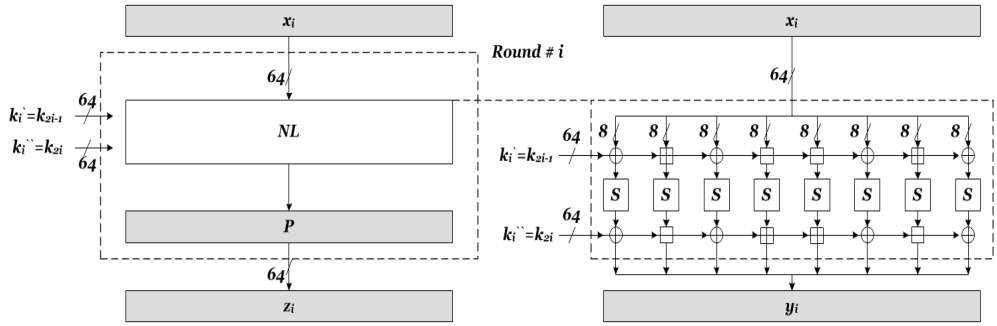
2.1 표기법

본 논문에서는 다음과 같은 표기법을 사용한다.

- $k_i^{'}$: i 번째 128-비트 라운드 키 중 첫 번째 64-비트 라운드 키, $k_i^{'} = k_{i,1}^{'} \| k_{i,2}^{'} \| \cdots \| k_{i,8}^{'}$.
- $k_i^{''}$: i 번째 128-비트 라운드 키 중의 두 번째 64-비트 라운드 키, $k_i^{''} = k_{i,1}^{''} \| k_{i,2}^{''} \| \cdots \| k_{i,8}^{''}$.
- $??_x$: 임의의 8-비트 정수의 16진수 표현.
- $(a, ??_x)$: a 번째 바이트의 값이 $??_x$이고, 나머지 바이트는 0인 64-비트 값, $a = 1,2,\cdots,8$.
- $P((a, ??_x))$: P 함수에 $(a, ??_x)$가 입력되었을 때 출력값.
- $(\alpha \rightarrow \beta)_t$: 입력차분이 α이고 출력차분이 β인 t-라운드 차분 특성.
- $[\alpha \rightarrow \Delta_\beta]_t$: 입력차분이 α이고 출력차분의 집합이 Δ_β인 t-라운드 부정 차분 특성.

2.2 블록 암호 PP-1/64-128

블록 암호 PP-1은 Buholc 등이 제안한 블록 암호로서 다양한 길이의 데이터 블록과 비밀 키를 사용한다. 여기서, 64-비트 데이터 블록과 128-비트 비밀키를 사용하는 PP-1을 PP-1/64-128로 표기한다. PP-1/64-128은 암호화 과정과 복호화 과정이 동일한 11-라운드 인볼루션 SPN 구조로 구성된다. 각 라운드 함수는 비선형 함수 NL(Nonlinear function)과 비트 기반 치환 함수 P로 구성된다. 비선형



〔그림 1〕 블록 암호 PP-1/64-128 라운드 함수 및 NL 함수

함수 NL은 8×8 S-박스와 XOR, 법 2^8에서의 덧셈과 뺄셈 연산으로 구성되고, 두 개의 64-비트 라운드 키가 사용 된다 (〔그림 1〕 참조).

비트 기반 치환 함수 P는 64-비트 입력값을 64-비트 출력값으로 치환한다. 가장 왼쪽에 있는 비트의 인덱스를 '1', 가장 오른쪽에 있는 비트의 인덱스를 '64'라고 하면, 각 비트가 치환되는 규칙은 〔표 2〕와 같다. 예를 들어, '1' 번째 비트는 '10' 번째 비트로 치환되고, 비트 '64' 번째 비트는 비트 '61' 번째 비트로 치환 된다.

키 스케줄은 공격 과정에 영향을 주지 않기 때문에 생략한다. PP-1/64-128에 대한 보다 자세한 사항은 〔4〕을 참고하기 바란다.

2.3 부정 차분 공격 (Truncated Differential Cryptanalysis)

부정 차분 공격은 1994년 Lars R. Knudsen이 제안한 공격방법으로 차분 특성을 구성할 때, 차분 특성의 출력차분을 모두 예측하지 않고 공격에 필요한 비트 정보만을 이용하여 차분 공격하는 방법이다. 다시 말해서, 기존의 차분 공격은 하나의 입력차분과 하나의 출력차분을 고려하지만, 부정 차분 공격에서는 여러 개의 출력차분을 동시에 고려하여 높은 확률을 가지는 차분 특성을 구성한다. 이를 통해 공격에 필요한 선택 평문쌍의 수를 감소시킬 수 있지만, 공격 과정에서 여러 개의 출력차분을 고려하여야 하기 때문에 옳은 암호문쌍을 걸러내기 위한 필터링 과정에서의 효율성이 감소하는 단점이 존재한다. 예를 들어, 3개의 t-라운드 차분 특성 $(\alpha\rightarrow\beta_1)_t$, $(\alpha\rightarrow\beta_2)_t$, $(\alpha\rightarrow\beta_3)_t$의 확률이 각각 p_1, p_2, p_3이면, t-라운드 부정 차분 특성 $[\alpha\rightarrow\{\beta_1,\beta_2,\beta_3\}]_t$의 확률은 $p_1+p_2+p_3$로 계산된다. 하지만, 필터링 단계에서는 β_1,β_2,β_3이 확산되는 것을 모두 고려한다.

III. PP-1/64-128에 대한 부정 차분 특성

본 장에서는 PP-1/64-128에 대한 9-라운드 부정 차분 특성을 소개한다. 이를 위해, 9개의 8-라운드 차분 특성을 구성하고, 6개의 1-라운드 부정 차분 특성을 연결하여 18개의 9-라운드 부정 차분 특성을 구성한다.

3.1 PP-1/64-128에 대한 8-라운드 차분 특성

PP-1/64-128은 S-박스, 법덧셈, 법뺄셈, XOR만을 사용한다. 그러므로 높은 확률을 갖는 차분 특성을 구성하기 위해서 비선형 연산인 법덧셈, 법뺄셈, S-박스를 최소한으로 포함하는 차분 특성을 구성하여야한다. 이를 위해 한 라운드에 하나의 S-박스를 지나고, XOR만을 포함하는 차분 특성만을 고려한다. 이러한 기준에 따라, 다음과 같은 9개의 1-라운드 차분 특성 $(\alpha\rightarrow\beta)_1$을 구성하였다 ($(\alpha,\beta)\in\{(8,01_x),(8,08_x),(8,09_x)\}$). 이 차분 특성은 8 번째 S-박스에만 차분이 입력되고 법덧셈과 법뺄셈을 통과하지 않는다. 그러므로 S-박스에 대한 차분 확률만으로 차분 특성의 확률을 계산한다. 또한, 치환 함수 P는 $P((8,01_x))=(8,08_x)$, $P((8,08_x))=(8,01_x)$, $P((8,09_x))=(8,09_x)$을 만족하므로, $((8,01_x)\rightarrow(8,08_x))$의 확률은 S-박스에서 입력차분이 01_x일 때 출력차분이 01_x일 확률과 같다. S-박스의 차분 분포표를 이용하여 9개의 1-라운드 차분 특성의 확률이 모두 2^{-7}임을 쉽게 확인할 수 있다.

위에서 소개한 1-라운드 차분 특성을 확장하여 다음과 같은 2-라운드 차분 특성 $(\alpha\rightarrow\beta)_2$을 고려하면 이 차분 특성의 확률은 다음과 같이 계산된다.

〔표 2〕 P 함수

입력	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
출력	10	15	18	31	26	47	34	63	42	1	50	17	58	33	2	49
입력	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
출력	12	3	20	19	28	35	36	51	44	5	52	51	60	37	4	53
입력	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
출력	14	7	22	23	30	39	38	55	46	9	54	25	62	41	6	57
입력	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
출력	16	11	24	27	32	43	40	59	48	13	56	29	64	45	8	61

$(\alpha, \beta \in \{01_x, 08_x, 09_x\})$

$$\sum_{\gamma \in \{(8,01_x),(8,08_x),(8,09_x)\}} \Pr[(\alpha \rightarrow \gamma)_1] \cdot \Pr[(\gamma \rightarrow \beta)_1].$$

1-라운드 차분 특성의 확률이 2^{-7}이므로, 2-라운드 차분 특성의 확률은 $3 \cdot 2^{-7 \cdot 2}$이다. 이를 확장하여 t-라운드 차분특성 $(\alpha \rightarrow \beta)_t$의 확률을 다음과 같은 재귀식을 이용하여 계산 할 수 있다.

$$\sum_{\gamma \in \{(8,01_x),(8,08_x),(8,09_x)\}} \Pr[(\alpha \rightarrow \gamma)_{t-1}] \cdot \Pr[(\gamma \rightarrow \beta)_1]. \quad (1)$$

식 (1)에 의해, t-라운드 차분 특성 $(\alpha \rightarrow \beta)_t$의 차분 특성의 확률은 각각 $3^{t-1} \cdot 2^{-7 \cdot t}$이다. 따라서 $\Pr[(\alpha \rightarrow \beta)_8] \approx 2^{44.91}$이다.

3.2 9-라운드 부정 차분 특성

본 절에서는 18개의 1-라운드 부정 차분 특성을 구성하고, 3.1절에서 구성한 8-라운드 차분 특성과 연결하여 18개의 9-라운드 부정 차분 특성을 구성한다. 여기서, 고려하는 1-라운드 부정 차분 특성은 입력차분이 각각 $(8,01_x),(8,08_x),(8,09_x)$이고, 출력차분은 특정한 성질을 만족하는 6개의 차분 집합으로 구성된다. 첫 번째 출력차분 집합은 $\{(8,01_x),(8,08_x),(8,09_x)\}$로 8 번째 바이트에만 차분이 존재하고, 두 번째 출력차분 집합은 2번째와 8 번째 바이트에만 차분이 존재하거나 2 번째 바이트에만 차분이 존재한다. 이러한 출력차분 집합을 정의하기 위해 〔표 3〕과 같은 차분 집합 X_i를 고려한다. $P((8,x_{1,j}))(x_{1,j} \in X_1)$는 $(8,01_x),(8,08_x),(8,09_x)$로 8 번째 바이트에만 출력차분이 존재하고, $P((8,x_{2,j}))(x_{2,j} \in X_2)$는 2 번째와 8 번째 바이트에만 출력차분이 존재하거나 2 번째 바이트에만 차분이 존재한다.

이렇게 구성한 1라운드 부정 차분 특성의 확률은 S-박스의 차분 분포표를 이용하여 계산할 수 있다. 예를 들면, $[01_x \rightarrow \{(8,01_x),(8,08_x),(8,09_x)\}]$인 1라운드 차분 특성의 확률은 $3 \cdot 2^{-7}$ $(\approx \Pr[(8,01_x) \rightarrow (8,01_x)] + \Pr[(8,01_x) \rightarrow (8,08_x)] + \Pr[(8,01_x) \rightarrow (8,09_x)])$ 이다. 이를 통해 1절에서 소개한 8-라운드 차분 특성 $(\alpha \rightarrow \beta)_8$과 1 라운드 부정 차분 특성 $[\beta \rightarrow X_i]_1$을 연결하여 18개의 9라운드 부정 차분 특성 $[\alpha \rightarrow X_i]_9$을 구성한다. 각각의 부정 차분 특성의 확률은 다음과 같이 계산된다. 〔표 4〕는 이 결과를 나타낸다.

$$\sum_{\gamma \in \{(8,01_x),(8,08_x),(8,09_x)\}} \Pr[(\alpha \rightarrow \gamma)_8] \cdot \Pr[[\gamma \rightarrow X_i]_1]$$

IV. PP-1/64-128에 대한 부정 차분 공격

본 장에서는 블록 암호 PP-1/64-128에 대한 부정 차분 공격을 소개한다. 제안하는 공격은 3장에서 소개한 9-라운드 부정 차분 특성을 2~10 라운드에 적용하고, 2 라운드 입력차분이 $(8,01_x)$, $(8,08_x)$, $(8,09_x)$이 되는 평문쌍이 존재하도록 평문 스트럭처를 구성한다. 그리고 18개의 부정 차분 특성을 이용하여, 11 라운드 키의 72 비트와 1 라운드 키의 8 비트를 복구한다. 그리고 복구된 정보를 이용하여 가능한 비밀 키에 대해 전수 조사한다.

4.1 스트럭쳐 구성

데이터 복잡도를 낮추기 위해서 입력차분이 다른 부정 차분 특성을 동시에 고려한다. 이를 위해 2 라운드 입력차분이 $(8,01_x)$, $(8,08_x)$, $(8,09_x)$이 되는 평문쌍이 존재하도록 다음과 같은 평문 스트럭쳐를 구성한다.

임의의 56-비트값 P_i를 선택하고, 여기에 가능한 모든 8 비트 값을 하위 비트에 연접하여 2^8개의 64-

〔표 3〕 8번째 S-박스에서의 출력차분 집합 ($X_0 = \{00000000\}$)

출력차분 집합
$X_1 = \{x_1x_2x_3x_4x_5x_6x_7x_8 \mid x_5, x_8 \in \{0,1\},\ x_1 = x_2 = x_3 = x_4 = x_6 = x_7 = 0\} - X_0 = \{01_x,\ 08_x,\ 09_x\}$
$X_2 = \{x_1x_2x_3x_4x_5x_6x_7x_8 \mid x_2,\ x_5,\ x_8 \in \{0,1\},\ x_1 = x_3 = x_4 = x_6 = x_7 = 0\} - \bigsqcup_{i=0}^{1} X_i = \{40_x, 41_x, 48_x, 49_x\}$
$X_3 = \{x_1x_2x_3x_4x_5x_6x_7x_8 \mid x_1,\ x_2,\ x_5,\ x_6,\ x_8 \in \{0,1\},\ x_3 = x_4 = x_7 = 0\} - \bigsqcup_{i=0}^{2} X_i$
$X_4 = \{x_1x_2x_3x_4x_5x_6x_7x_8 \mid x_1, x_2,\ x_5,\ x_6,\ x_7,\ x_8 \in \{0,1\},\ x_3 = x_4 = 0\} - \bigsqcup_{i=0}^{3} X_i$
$X_5 = \{x_1x_2x_3x_4x_5x_6x_7x_8 \mid x_1,\ x_2,\ x_3,\ x_5,\ x_6,\ x_7,\ x_8 \in \{0,1\},\ x_4 = 0\} - \bigsqcup_{i=0}^{4} X_i$
$X_6 = \{x_1x_2x_3x_4x_5x_6x_7x_8 \mid x_1,\ x_2,\ x_3,\ x_4,\ x_5, x_6,\ x_7, x_8 \in \{0,1\}\} - \bigsqcup_{i=0}^{5} X_i$

비트 $P_{i,j}$를 하나의 스트럭쳐로 정의한다. 즉, $S_i = \{(P_{i,j} = P_i \| j) \mid 0 \leqq j \leqq 255\}$. 하나의 스트럭쳐에서 2개의 평문을 선택하면 차분의 형태가 $(8,??_x)$인 2^{15}개의 평문쌍을 구성할 수 있으며, 이 중, 라운드 2의 입력차분이 $(8,01_x), (8,08_x), (8,09_x)$인 평문쌍이 각각 2^7개 존재한다. 그러므로 하나의 스트럭쳐에서 라운드 10의 출력차분이 $P((8,X_i))$에 포함되는 평문쌍의 기댓값은 다음과 같이 계산된다 ($P((8,X_i)) = \{P((8,x_{i,j})) \mid x_{i,j} \in X_i\}$).

$$\sum_{\alpha \in \{(8,01_x),(8,08_x),(8,09_x)\}} 2^7 \cdot \Pr[[\alpha \rightarrow X_i]_9].$$

즉, 하나의 스트럭쳐에 대해 라운드 10_x의 출력차분이 $P((8,X_1))$에 포함되는 평문쌍의 기댓값은 $2^{-40.14} (\simeq 3 \cdot 2^7 \cdot 2^{-48.74})$이다. 〔표 5〕는 하나의 스트럭쳐에서 각각의 X_i에 대한 라운드 10의 입력차분이 $P((8,X_i))$에 포함되는 평문쌍의 기댓값을 의미한다.

4.2 PP-1/64-128에 대한 부정 차분 공격

본 논문에서 제안하는 공격은 3장에서 소개한 18개의 부정 차분 특성을 이용한다. 2 라운드의 입력차분이 부정 차분 특성의 입력차분이 되도록 스트럭쳐를 이용하여 평문쌍을 구성한다. 그리고 이를 이용하여 11 라운드 키를 바이트별로 나누어 복구한다. 첫 번째로, 출력차분 집합이 $P((8,X_1))$인 3개의 부정 차분 특성을 따르는 평문쌍을 이용하여 8-비트 $k''_{11,8}$을 복구한다. 다음으로 출력차분 집합이 $P((8,X_2))$인 3개의 부정 차분 특성을 따르는 평문쌍을 이용하여 16-비트 $(k'_{11,2}, k''_{11,2})$를 복구한다. 이 때 이전 단계에서 복구한 $k''_{11,8}$을 이용하여 부정 차분 특성을 따르지 않는 평문쌍은 1 라운드 키 $k'_{1,8}$를 복구하는데 사용 하므로 '테이블 2'에 저장한다. 이러한 과정을 반복하여 11 라운드 키의 72 비트를 복구하고, 공격 단계에서 각각의 부정 차분 특성을 따르는 암호문쌍에 대응하는 평문쌍을 이용하여 8-비트 $k'_{1,8}$을 복구한다. 마지막으로 복구한 라운드 키 정보를 이용하여 가능한 비밀키에 대한 전수조사를 수행한다. 제안하는 부정 차분 공격의 공격 과정은 다음과 같다.

1. $2^{42.16}$개의 스트럭쳐를 $S_i (1 \leq i \leq 2^{42.16})$ 구성하고, 이에 대응되는 암호문을 얻는다. 각각의 스트럭쳐에서 2^{15}개의 암호문쌍을 구성할 수 있으므로, $2^{57.16}$개의 암호문쌍을 구성한다.
2. 단계 1에서 구성한 암호문쌍에 대해, 3 번째 바이트와 5 번째 바이트의 차분이 00_x인 암호문쌍과 이에 대응되는 평문쌍을 '테이블 1'에 저장한다.

〔표 4〕 9-라운드 부정차분 특성 $[\alpha \rightarrow X_i]_9$의 확률

	X_1	X_2	X_3	X_4	X_5	X_6
01_x	$2^{-48.74}$	$2^{-48.74}$	$2^{-46.78}$	$2^{-46.41}$	$2^{-45.24}$	$2^{-44.38}$
08_x	$2^{-48.74}$	$2^{-48.74}$	$2^{-46.78}$	$2^{-46.41}$	$2^{-45.24}$	$2^{-44.38}$
09_x	$2^{-48.74}$	$2^{-48.74}$	$2^{-46.78}$	$2^{-46.41}$	$2^{-45.24}$	$2^{-44.38}$

3. 단계 2에서 저장한 '테이블 1'의 암호문쌍에서 차분의 형태가 $(00\,0000\,0000\,00\,00\,??_x)$인 암호문쌍을 선택하여 다음을 수행한다.
   3.1. $k''_{11,8}$을 추측하여 라운드 10의 출력값의 차분이 $P((8,X_1))$에 포함되면 이에 대응되는 키의 카운트 값을 1씩 증가시킨다.
   3.2. 카운트가 '3'이상 추측된 키를 옳은 $k''_{11,8}$로 결정한다.
   3.3. 단계 3에서 사용한 암호문쌍을 '테이블 1'에서 제거하고, '테이블 2'에 저장한다.
   3.4. '테이블 1'의 암호문쌍에 대해 복구한 $k''_{11,8}$을 이용하여 부정 차분 특성을 만족하지 않는 암호문쌍을 '테이블 1'에서 제거한다.
4. '테이블 1'의 암호문쌍에 대해 차분의 형태가 $(00\,??\,00\,0000\,00\,00\,??_x)$인 암호문쌍을 선택하여 다음을 수행한다.
   4.1. 단계 4를 통과한 암호문쌍에 대해, $k''_{11,2}$, $k'_{11,2}$를 추측하여 라운드 10의 출력값의 차분이 $P((8,X_2))$에 포함되면 이에 대응되는 키의 카운트 값을 '1'씩 증가시킨다.
   4.2 카운트가 '3'이상 추측된 키를 옳은 $k''_{11,2}$, $k'_{11,2}$로 결정한다.
   4.3. 단계 4에서 사용한 암호문쌍을 '테이블 1'에서 제거하고, '테이블 2'에 저장한다.
   4.4. '테이블 1'의 암호문쌍에 대해 복구한 $k''_{11,2}$, $k'_{11,2}$을 이용하여 부정 차분 특성을 만족하지 않는 암호문쌍을 '테이블 1'에서 제거한다.
5. '테이블 1'의 암호문쌍에 대해, 차분의 형태가 $(00\,??\,00\,0000\,??\,00\,??_x)$인 암호문쌍을 선택하여 다음을 수행한다.
   5.1. 단계 5를 통과한 암호문쌍에 대해, $k''_{11,6}$를 추측하여 라운드 10의 출력값의 차분이 $P((8,X_3))$에 포함되면 이에 대응되는 키의 카운트 값을 '1'씩 증가시킨다.
   5.2 카운트가 '7'이상 추측된 키를 옳은 $k''_{11,6}$로 결정한다.
   5.3. 단계 5에서 저장한 암호문쌍을 '테이블 1'에서 제거하고, '테이블 2'에 저장한다.
   5.4. '테이블 1'의 암호문쌍에 대해 복구한 $k''_{11,6}$을 추측하여 부정 차분 특성을 만족하지 않는 암호문쌍을 '테이블 1'에서 제거한다.
6. '테이블 1'의 암호문쌍에 대해, 차분의 형태가 $(??\,??\,00\,0000\,??\,00\,??_x)$인 암호문쌍을 선택하여 다음을 수행한다.
   6.1. 단계 6을 통과한 암호문쌍에 대해 $k''_{11,1}$을 추측하여 라운드 10의 출력값의 차분이 $P((8,X_4))$에 포함되면 이에 대응되는 키의 카운트 값을 '1'씩 증가시킨다.
   6.2 카운트가 '7'이상 추측된 키를 옳은 $k''_{11,1}$로 결정한다.
   6.3. 단계 6에서 저장한 암호문쌍을 '테이블 1'에서 제거하고, '테이블 2'에 저장한다.
   6.4. '테이블 1'의 암호문쌍에 대해 복구한 $k''_{11,1}$을 추측하여 부정 차분 특성을 만족하지 않는 암호문쌍을 '테이블 1'에서 제거한다.
7. '테이블 1'의 암호문쌍에 대해, 차분의 형태가 $(??\,??\,00\,0000\,??\,??\,??_x)$인 암호문쌍을 선택하여 다음을 수행한다.
   7.1. 단계 7을 통과한 암호문쌍에 대해, $k''_{11,7}$, $k'_{11,7}$를 추측하여 라운드 10의 출력값의 차분이 $P((8,X_5))$에 포함되면 이에 대응되는 키의 카운트 값을 '1'씩 증가시킨다.
   7.2 카운트가 '7'이상 추측된 키를 옳은 $k''_{11,7}$, $k'_{11,7}$로 결정한다.
   7.3. 단계 7에서 저장한 암호문쌍을 '테이블 1'에서 제거하고, '테이블 2'에 저장한다.
   7.4. '테이블 1'의 암호문쌍에 대해 복구한 $k''_{11,7}$, $k'_{11,7}$를 이용하여 부정 차분 특성을 만족하지 않는 암호문쌍을 '테이블 1'에서 제거한다.
   7.5. 단계 7.4에서 저장한 '테이블 1'에 대해, $k''_{11,4}$, $k'_{11,4}$를 추측하여 라운드 10의 출력값의 차분이 $P((8,X_6))$에 포함되면 이에 대응되는 키의 카운트 값을 '1'씩 증가시킨다.
   7.5 카운트가 '12'이상 추측된 키를 옳은 $k''_{11,4}$, $k'_{11,4}$로 결정한다.

〔표 5〕 하나의 스트럭쳐에서 라운드 11의 입력차분이 $P((8,X_i))$인 평문쌍의 기댓값

X_1	X_2	X_3	X_4	X_5	X_6
$2^{-40.16}$	$2^{-40.16}$	$2^{-38.20}$	$2^{-37.83}$	$2^{-36.66}$	$2^{-35.80}$

〔표 6〕 각 단계별 공격 복잡도 및 성공확률

	필터링 후 남는 암호문쌍	키 추측 계산 복잡도	테이블 계산 복잡도	테이블 저장 암호문쌍	성공 확률 (%)
단계 3	$4+2^{1.14}<7$	$2^{4.34}(\approx \frac{7\cdot 2^8}{88})$	$2^{35.68}(\approx \frac{2^{41.14}}{88})$	$2^{34.73}(\approx 2^{41.14}\cdot 3\cdot 2^{-8})$	98.62
단계 4	$4+2^{2.73}<11$	$2^{13.00}(\approx \frac{11\cdot 2^{16}}{88})$	$2^{28.27}(\approx \frac{2^{35.68}}{88})$	$2^{26.73}(\approx 2^{34.73}\cdot 2^{-8})$	99.75
단계 5	$4\cdot 2^{1.91}+2^{2.73}<22$	$2^{6.00}(\approx \frac{22\cdot 2^8}{88})$	$2^{20.27}(\approx \frac{2^{26.73}}{88})$	$2^{20.32}(\approx 2^{26.72}\cdot 3\cdot 2^{-8})$	99.99
단계 6	$4\cdot 2^{2.32}+2^{4.32}<47$	$2^{7.34}(\approx \frac{47\cdot 2^8}{88})$	$2^{13.86}(\approx \frac{2^{20.32}}{88})$	$2^{12.32}(\approx 2^{20.32}\cdot 2^{-8})$	99.99
단계 7	$4\cdot 2^{3.5}+2^{4.32}<66$	$2^{15.59}(\approx \frac{66\cdot 2^{16}}{88})$	$2^{5.86}(\approx \frac{2^{12.32}}{88})$	$2^{4.32}(\approx 2^{12.32}\cdot 2^{-8})$	99.99
	$4\cdot 2^{4.35}+2^{4.32}<101$	$2^{16.20}(\approx \frac{101\cdot 2^{16}}{88})$	·	·	99.99
단계 8	$169.83+75.39<246$	$2^{9.48}(\approx \frac{246\cdot 2^8}{88})$	·	·	99.99

8. '테이블 2'에 저장된 암호문쌍에 대한 평문쌍에 대해 다음을 수행한다.
   8.1. $k'_{1,8}$을 추측하여 라운드 1의 S-박스의 출력 차분이 $\{01_x, 08_x 09_x\}$에 포함되면 대응하는 키의 카운트 값을 '1'씩 증가시킨다.
   8.2 카운트가 '13'이상 추측된 키를 옳은 $k'_{1,8}$로 결정한다.
9. 복구한 80 비트 정보를 이용하여, 가능한 비밀키를 전수조사 하여 비밀키를 복구한다.

4.3 공격 복잡도 및 성공확률

4.3.1 데이터 복잡도 및 메모리 복잡도

제안하는 공격 알고리즘의 데이터 복잡도는 2^8개의 평문으로 구성된 $2^{42.16}$개의 스트럭처를 사용하므로 $2^{50.16}(\approx 2^{42.16}\cdot 2^8)$개의 선택 평문이다.

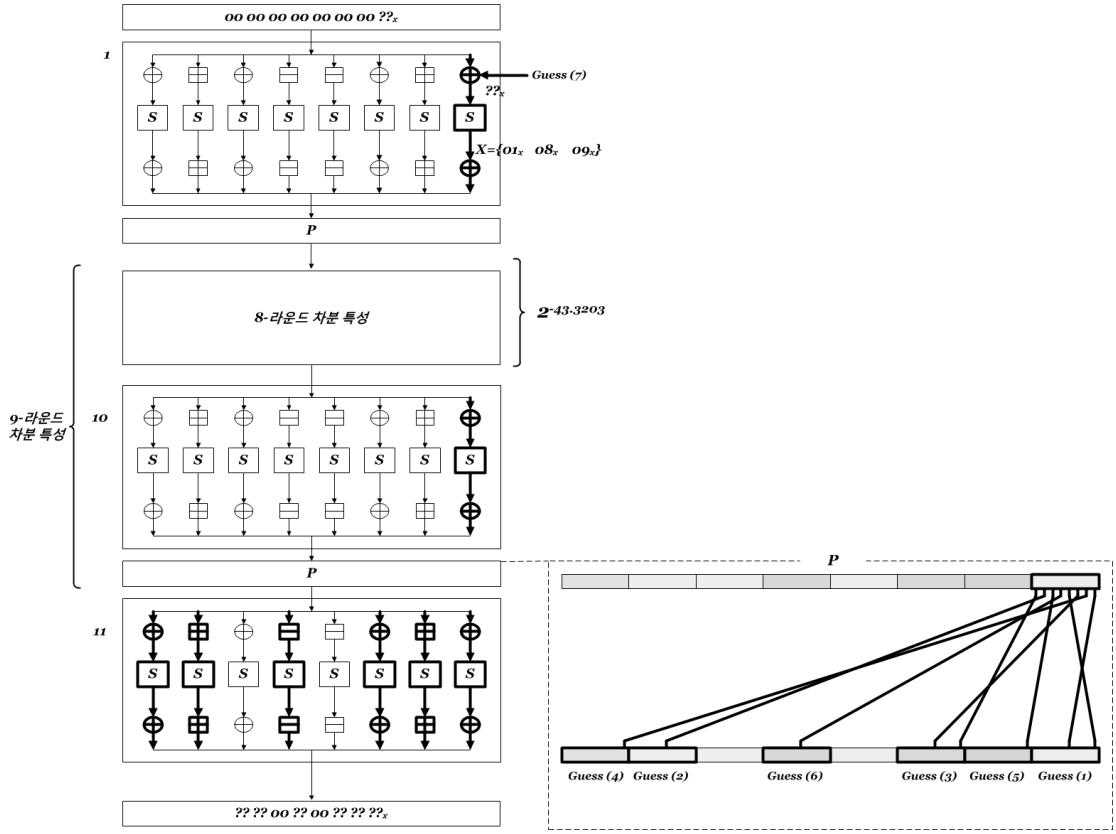
공격자는 하나의 스트럭처에 있는 2^8개의 평문에 대하여 $2^{15}(\approx 2^8 C_2)$개의 암호문쌍을 생성할 수 있으므로 $2^{42.16}$개 스트럭처에서는 $2^{57.16}(\approx 2^{42.16}\cdot 2^{15})$개의 암호문쌍을 구성 할 수 있다. 구성한 암호문쌍에 대하여 단계 2의 '16-비트 필터링 과정'을 통과하는 평문쌍·암호문쌍만 저장하므로 $2^{41.16}(\approx 2^{57.16}\cdot 2^{-16})$개의 평문쌍·암호문쌍을 '테이블 1'에 저장한다. 따라서, 메모리 복잡도는 $2^{46.16}(=2^{41.16}\cdot 2^3\cdot 2\cdot 2)$ 바이트가 된다.

'테이블 2'에는 '테이블 1'에서 사용한 암호문쌍을 옮겨 저장하므로 $2^{33.17}(\approx 2^{1.16}+2^{9.16}+2^{17.16}+2^{25.16}+2^{33.16})$개의 평문쌍·암호문쌍을 저장할 공간이 필요하다. 따라서 '테이블 2'의 메모리 복잡도는 $2^{38.17}(\approx 2^{33.17}\cdot 2^3\cdot 2\cdot 2)$바이트가 된다. 따라서 메모리 복잡도는 '테이블 1'과 '테이블 2'의 저장공간을 합한 $2^{46.16}(\approx 2^{46.16}+2^{38.17})$이 된다.

4.3.2 시간 복잡도

단계 1에서 $2^{50.16}(\approx 2^{42.16}\cdot 2^8)$개의 평문을 암호화하는 과정이 필요하고, 단계 3에서 필터링 과정을 통과한 암호문쌍의 개수가 약 $7(\approx 4+2^{1.16})$개 이므로, 이 암호문쌍에 대하여 $k''_{11,8}$을 추측하면 $2^{4.34}(\approx 7\cdot 2^8\cdot\frac{1}{11}\cdot\frac{1}{8})$의 PP-1/64-128의 복호화 연산이 된다. 단계 3.4에서는 $2^{41.16}$개의 암호문쌍을 복호화 하여 '테이블 1'을 갱신하는 과정이 필요하므로 $2^{35.70}(\approx 2\cdot 2^{41.16}\cdot\frac{1}{11}\cdot\frac{1}{8})$의 PP-1/64- 128의 복호화 연산이 필요하다. 이 과정을 통과하는 암호문쌍의 수는 $2^{34.75}(\approx 2^{41.16}\cdot\frac{3}{256})$개가 된다. 유사한 방법으로 각 단계의 복잡도를 계산하면 〔표 6〕과 같다.

그러므로 단계 2~8 계산 복잡도는 키 추측 계산 복잡도와 테이블 계산 복잡도를 합한 $2^{35.69}$번 PP-1/64-128 복호화 연산이 필요하다. 단계 9에서 8-비트 $k'_{1,8}$와 72-비트 라운드 키 복구 이후, 비밀 키



〔그림 2〕 PP-1/64-128에 대한 부정 차분 공격 모델

복구를 위하여 2^{48}번의 키 스케줄 연산이 필요하므로 비밀 키 복구에 필요한 전체 계산 복잡도는 2^{48}의 암호화 연산이 필요하다. 따라서 비밀 키 복구에 필요한 전체 계산 복잡도는 $2^{50.45}(\approx 2^{50.16}+2^{48.00}+2^{35.69})$가 된다.

4.3.3 성공확률

본 논문에서 제안하는 공격의 성공확률을 각 단계별로 계산하면 〔표 6〕과 같다. 예를 들어, 단계 3.2의 경우 틀린 키가 통과할 확률은 다음과 같이 이항 분포로 계산할 수 있다.

$$\sum_{i=3}^{7}\left(\binom{7}{i}\cdot\left(\frac{3}{2^8}\right)^i\cdot\left(1-\frac{3}{2^8}\right)^{7-i}\right). \quad (2)$$

식 (2)는 '테이블 1'에 저장될 평문 · 암호문쌍의 기댓값이 '7' 이하이고, 차분 값 X_1을 만족할 확률이 $3\cdot 2^{-8}$이므로 이항분포를 이용하여 틀린 키 하나가 통과할 확률을 계산한 것이다. 그러므로 255개의 틀린 키가 통과하지 못할 확률은 다음과 같다.

$$\left(1-\sum_{i=3}^{7}\left(\binom{7}{i}\cdot\left(\frac{3}{2^8}\right)^i\cdot\left(1-\frac{3}{2^8}\right)^{7-i}\right)\right)^{255}\approx 0.9862.$$

위와 동일한 방법으로 각 단계별 성공 확률을 정리하면 〔표 6〕과 같고, 단계 3을 통과 하면 이후 단계의 성공확률은 99.99%에 근사 하므로 전체 공격의 성공확률은 단계 3을 통과하는 성공확률인 98.62% 가 된다.

V. 결 론

블록 암호 PP-1은 다양한 길의 데이터 블록과 비밀 키 길이를 지원하는 블록 암호이다. 본 논문에서는 PP-1/64-128에 대한 부정 차분 공격을 제안했다. 본 논문에서 제안하는 공격은 $2^{50.16}$의 선택 평문과 $2^{46.16}$ 바이트 메모리를 이용하여, $2^{50.45}$의 PP-1/64 -128의

암호화 연산을 통해 비밀키를 복구한다. 본 논문에서 제안 하는 PP-1/64-128에 대한 공격 결과는 현재까지 알려진 공격 결과 중 가장 좋은 결과이다.

다른 길이의 데이터 블록과 비밀키 길이를 사용하는 블록 암호 PP-1에 대한 연구는 향후 과제로 남긴다.

참고문헌

〔1〕 J. Daemen and V. Rijmen, The design of Rijndael: AES - the advanced encryption standard, Springer, Mar. 2002.

〔2〕 H. J. Lee, S. J. Lee, J. H. Yoon, D. H. Cheon, J. I. Lee, "The SEED Encryption Algorithm," RFC 4269, Dec 2005.

〔3〕 D. Kwon, J. Kim, S. Park, S. H. Sung, Y. Sohn, J. H. Song, Y. Yeom, E-J. Yoon, S. Lee and J. Lee,, "New Block Cipher: ARIA," ICISC 2003, LNCS 2971, pp. 432-445, 2004.

〔4〕 K. Buholc, K. Chmiel, A. Grocholewska-Czurlo, E. Idzikowska, I. Janicka-lipska, J. Stoklosa, "Scalable PP-1 block cipher," Mathematics Computer Science 2010, Vol 20, No.2, pp. 401-411, 월 2010.

〔5〕 A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," CHES 2007, LNCS 4727, pp. 450-466, 2007.

〔6〕 D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B-S. Koo, C. Lee, D. Chang, J. Lee and K. Jeong, "HIGHT : A New Block Cipher Suitable for Low-Resource Device ," CHES 2006, LNCS 4249, pp. 46-59, 2006.

〔7〕 C.D. Canniere, O. Dunkelman and M. Knezevic, "KATAN and KTANTAN : A Family of Small and Efficient Hardware-Oriented Block Ciphers" CHESS '09, LNCS 5747, pp. 272-288, 2009.

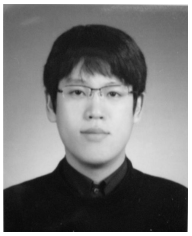
〔8〕 E. Idzikowska, "CED for Involutional Functions of PP-1 Cipher," Future Information Technology (FutureTech) 2010, pp. 1-5, May. 2010.

〔9〕 L. R. Knudsen, "Truncated and High Order Differentials", FSE'06, LNCS 1039, pp. 99-211, 1996.

〔10〕 홍용표, 이유섭, 정기태, 홍석희, "블록 암호 PP-1/64-128에 대한 차분 공격", KoreaCrypt '11, 10(1), pp. 1-7, 2011년 1월.

〈著者紹介〉

홍 용 표 (Yongpyo Hong) 학생회원
2009년 2월: 숭실대학교 수학과 학사
2011년 8월: 고려대학교 정보경영공학전문대학원 석사
2011년 9월~현재: LG전자
〈관심분야〉 블록 암호 및 해쉬 함수 분석 및 설계



이 유 섭 (Yuseop Lee) 학생회원
2007년 2월: 서울시립대학교 수학과 학사
2007년 3월~현재: 고려대학교 정보경영공학전문대학원 석박사 통합과정
〈관심분야〉 스트림 암호 및 해쉬 함수의 분석 및 설계

정 기 태 (Kitae Jeong) 학생회원
2004년 2월: 고려대학교 수학과 학사
2006년 2월: 고려대학교 정보보호대학원 석사
2011년 8월: 고려대학교 정보경영공학전문대학원 박사
2011년 8월~현재: 정보보호기술센터 연구원
〈관심분야〉 블록 암호, 스트림 암호 및 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 2월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2004년 4월~2005년 2월: K.U. Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~2008년 8월: 고려대학교 정보보호대학원 조교수
2008년 9월~현재: 고려대학교 정보보호대학원 부교수
〈관심분야〉 암호 알고리즘 설계 및 분석, 컴퓨터 포렌식