

# 블록 암호 KT-64에 대한 안전성 분석

강진건<sup>†</sup> · 정기태<sup>\*\*</sup> · 이창훈<sup>\*\*\*</sup>

## 요약

64-비트 블록 암호 KT-64는 CSPNs (Controlled Substitution-Permutation Networks)를 사용하여 FPGA와 같은 하드웨어 구현에 적합하도록 설계된 블록 암호이다. 본 논문에서는 블록 암호 KT-64의 전체 라운드에 대한 확장된 연관키 부메랑 공격을 제안한다. 본 논문에서 소개하는 공격은 KT-64에 대한 최초의 공격이며,  $2^{45.5}$ 개의 연관키 선택 평문을 이용하여  $2^{65.17}$ 의 KT-64 암호화 연산을 수행하여 KT-64의 비밀키를 복구한다.

키워드 : 블록 암호, 확장된 연관키 부메랑 공격, 암호 분석

## Security Analysis of Block Cipher KT-64

Jinkeon Kang<sup>†</sup> · Kitae Jeong<sup>\*\*</sup> · Changhoon Lee<sup>\*\*\*</sup>

## ABSTRACT

KT-64 is a 64-bit block cipher which use CSPNs suitable for the efficient FPGA implementation. In this paper, we propose a related-key amplified boomerang attack on the full-round KT-64. The attack on the full-round KT-64 requires  $2^{45.5}$  related-key chosen plaintexts and  $2^{65.17}$  KT-64 encryptions. This work is the first known cryptanalytic result on KT-64.

Keywords : Block Cipher, Related-Key Amplified Boomerang Attack, Cryptanalysis

## 1. 서론

최근 하드웨어 환경에서 효율적으로 구현 가능한 데이터 의존 치환 함수인 DDP (Data Dependent Permutation)에 기반을 둔 블록 암호에 대한 연구가 활발히 진행되고 있다. 그 결과로서 SPECTR-H64 [1], CIKS-family (CIKS-1 [2], CIKS-128 [3]), Cobra-family (Cobra-S128 [4], Cobra-F64a [4], Cobra-F64b [4], Cobra-H64 [5], Cobra-H128 [5]), Eagle-64 [6] Eagle-128 [7], SCO-family [8] 등의 블록 암호 알고리즘이 제안되었다. 이 블록 암호들은 별다른 키 스케줄 연산 없이 키를 반복적으로 순서만 달리하여 사용하기 때문에 비밀키가 빈번하게 변경되는 환경에서 장점을 갖는다. 그러나 DDP의 선형성과 단순한 키 스케줄은 암호학적으로 취약하며, 그 결과 대부분의 알고리즘들이 분석되었다 [9]-[17].

블록 암호 KT-64 [18]는 암호학적으로 취약한 DDP 대신에 데이터에 의존하는 CSPNs (Controlled Substitution-Permutation Networks)를 기반으로 설계되었으며, FPGA와 같은 하드웨어 구현에 적합하다. 하지만 본 논문에서는 전체 라운드 KT-64에 대한 확장된 연관키 부메랑 공격을 제안함으로써, 기존에 제안된 DDP-기반 블록 암호와 마찬가지로 KT-64도 여전히 연관키 공격에 취약함을 보인다. 본 논문에서 제안하는 공격은  $2^{45.5}$ 개의 연관키 선택 평문을 필요로 하며  $2^{65.17}$ 의 KT-64 암호화 연산을 수행하여 KT-64의 128-비트 비밀키를 복구한다. 이는 블록 암호 KT-64에 대한 첫 번째 공격이다.

본 논문의 구성은 다음과 같다. 2장에서는 블록 암호 KT-64를 간략히 소개하고, 3장에서는 KT-64에 대한 확장된 연관키 부메랑 공격을 소개한다. 마지막으로 4장에서 결론을 맺는다.

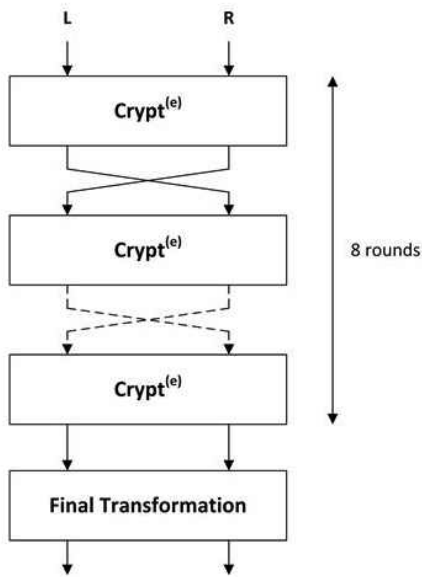
## 2. 블록 암호 KT-64 소개

본 장에서는 64-비트 블록 암호 KT-64를 소개한다. 이에 앞서, 본 논문에서는 다음과 같은 표기를 사용한다. 비트

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(No. 2011-0005648).  
† 준 회원: 고려대학교 정보보호대학원 석·박사통합과정  
\*\* 준 회원: 고려대학교 정보보호연구원 박사후연구원  
\*\*\* 종신회원: 한신대학교 컴퓨터공학부 조교수(교신저자)  
논문접수: 2011년 10월 21일  
심사완료: 2011년 11월 29일

표기는 왼쪽에서 오른쪽으로 1부터 표기된다. 예를 들어,  $P = (p_1, p_2, \dots, p_n)$  이면  $P$ 의 최상위 비트는  $p_1$ 이고,  $P$ 의 최하위 비트는  $p_n$ 이다. 그리고  $e_i$ 는  $i$ 번째 비트만 1이고, 나머지 비트는 0인 이진 수열이다. 예를 들어,  $e_1 = (1, 0, \dots, 0)$ 이다.

KT-64는 128-비트 비밀키  $K = (K_1, K_2, K_3, K_4)$ 를 사용하는 64-비트 블록 암호이다. KT-64는 8 라운드로 구성되며 전체 구조는 (그림 1)과 같다.



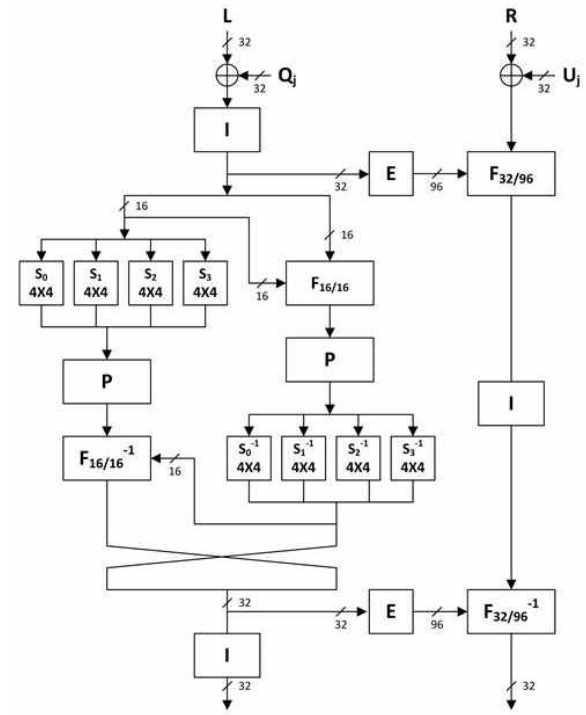
(그림 1) KT-64의 전체 구조

<표 1>은 KT-64의 8-라운드 암호화 과정이다.

<표 1> KT-64의 암호화 과정

1. A 64-bit input block  $P$  is divided into two 32-bit subblocks  $P_L$  and  $P_R$ .
2.  $(L, R) \leftarrow (P_L, P_R)$ ;
3. For  $r=1$  to 7 do:  
 $(L, R) \leftarrow \text{Crypt}^{(0)}(L, R, Q_r, U_r)$ ;  
 $(L, R) \leftarrow (R, L)$ ;
4. Perform transformation:  
 $(L, R) \leftarrow \text{Crypt}^{(0)}(L, R, Q_8, U_8)$ ;
5. Perform final transformation:  
 $(L, R) \leftarrow (L \oplus Q_9, R \oplus U_9)$ ;
6.  $(C_L, C_R) \leftarrow (L, R)$ ;
7. Return the ciphertext block  $C = (C_L, C_R)$ .

라운드 함수인  $\text{Crypt}^{(e)}$ 는 (그림 2)와 같다. 여기서  $e = 0(1)$ 은 암호화(복호화) 과정을 의미한다.

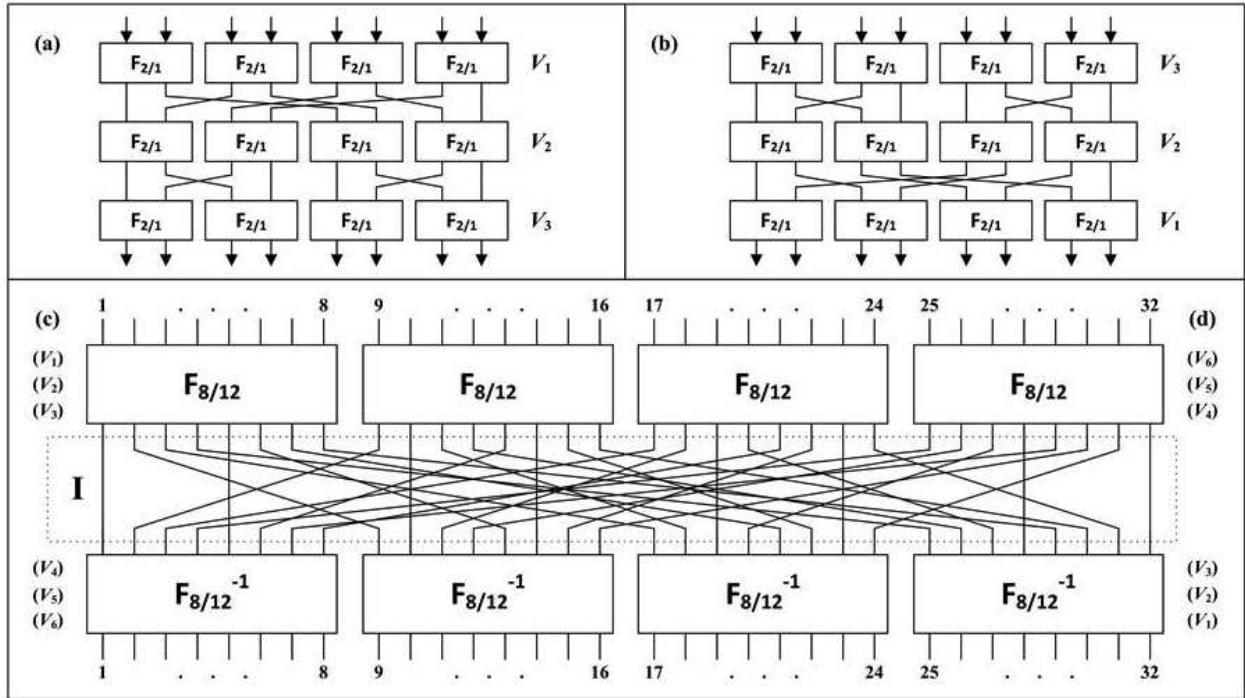


(그림 2) 라운드 함수:  $\text{Crypt}^{(e)}$

$F_{32/96}$ 와  $F_{32/96}^{-1}$ 의 96-비트 제어 비트를 생성하는데 사용되는 확장 함수  $E$ 는 32-비트  $X = (v_1, v_2, \dots, v_{32})$ 을 입력받아 제어 비트  $V = (V_1, V_2, V_3, V_4, V_5, V_6)$ 을 출력한다.

- $V_1 = (v_7, v_8, v_1, v_2, v_{16}, v_{15}, v_{10}, v_9, v_5, v_6, v_3, v_4, v_{11}, v_{12}, v_{13}, v_{14})$
- $V_2 = (v_9, v_{10}, v_{11}, v_{12}, v_1, v_2, v_7, v_8, v_{13}, v_{14}, v_{15}, v_{16}, v_5, v_6, v_3, v_4)$
- $V_3 = (v_{13}, v_{14}, v_{15}, v_{16}, v_5, v_6, v_3, v_4, v_1, v_2, v_7, v_8, v_9, v_{10}, v_{11}, v_{12})$
- $V_4 = (v_{21}, v_{22}, v_{29}, v_{30}, v_{25}, v_{26}, v_{23}, v_{24}, v_{31}, v_{32}, v_{27}, v_{28}, v_{17}, v_{18}, v_{19}, v_{20})$
- $V_5 = (v_{31}, v_{32}, v_{27}, v_{28}, v_{17}, v_{18}, v_{19}, v_{20}, v_{29}, v_{30}, v_{25}, v_{26}, v_{21}, v_{22}, v_{23}, v_{24})$
- $V_6 = (v_{19}, v_{20}, v_{23}, v_{24}, v_{27}, v_{28}, v_{29}, v_{30}, v_{21}, v_{22}, v_{17}, v_{18}, v_{32}, v_{31}, v_{25}, v_{26})$

(그림 3)과 같이,  $F_{32/96}$ 는 다음과 같이 정의되는  $F_{2/1}$ 을 결합하여 구성된다.



(그림 3) (a)  $F_{8/12}$  (b)  $F_{8/12}^{-1}$  (c)  $F_{32/96}$  (d)  $F_{32/96}^{-1}$

- $F_{2/1}(x_1, x_2, v) = (y_1, y_2)$   
 $y_1 = vx_2 \oplus x_1 \oplus x_2, y_2 = vx_1 \oplus x_2$

$F_{32/96}$ 와 역함수인  $F_{32/96}^{-1}$ 은  $F_{2/1}$ 에 대한 제어 비트의 입력 순서만 다르고 나머지는 동일하다. 여기서  $F_{32/96}$ 과  $Crypt^{(e)}$ 에 사용되는 치환 함수  $I$ 은 다음과 같다.

$$I = (1)(2,9)(3,17)(4,25)(5)(6,13)(7,21) \\ (8,29)(10)(11,18)(12,26)(14)(15,22) \\ (16,30)(19)(20,27)(23)(24,31)(28)(32)$$

(그림 4)는  $Crypt^{(e)}$ 에서 사용되는  $F_{16/16}$ 와  $F_{16/16}^{-1}$ 을 나타낸 것이다.

$Crypt^{(e)}$ 에 사용되는 S-box는 4개의 S-box로 구성되어 있는데, 자세한 내용은 [18]을 참조하라. 치환 함수  $P$ 는 다음과 같다.

$$P = (1)(2,5)(3,9)(4,13)(6)(7,10)(8,14) \\ (11)(12,15)(16)$$

마지막으로 KT-64의 키 스케줄은 매우 간단하다. 128-비트 비밀키  $K = (K_1, K_2, K_3, K_4)$ 가 <표 2>와 같이 별도의 키 스케줄 연산 없이 순서만 달리하여 바로  $Crypt^{(e)}$ 에 사용된다.

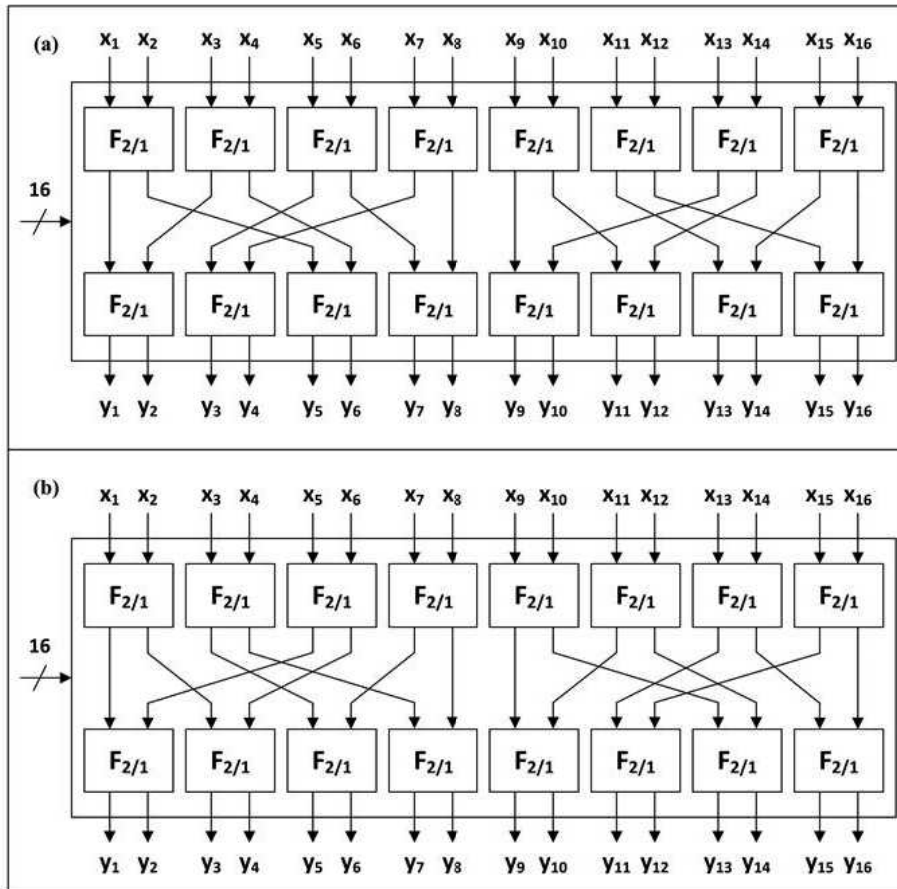
<표 2> KT-64의 키 스케줄링

Round $r$	암호화		복호화	
	$Q_r$	$U_r$	$Q_r$	$U_r$
1	$K_1$	$K_2$	$K_1$	$K_3$
2	$K_3$	$K_4$	$K_3$	$K_4$
3	$K_3$	$K_1$	$K_2$	$K_1$
4	$K_4$	$K_1$	$K_4$	$K_3$
5	$K_2$	$K_3$	$K_3$	$K_2$
6	$K_3$	$K_4$	$K_1$	$K_4$
7	$K_1$	$K_2$	$K_1$	$K_3$
8	$K_4$	$K_3$	$K_4$	$K_3$
FT	$K_1$	$K_3$	$K_1$	$K_2$

### 3. KT-64에 대한 확장된 연관키 부메랑 공격

본 장에서는 KT-64에 대한 확장된 연관키 부메랑 공격을 제안한다. 평문  $P, P^*, P', P'^*$ 를 비밀키  $K, K^*, K', K'^*$ 로 각각 암호화한다고 가정한다. 이때 각각의 평문과 비밀키는 다음 조건을 만족한다.

- $\alpha = P \oplus P^* = P' \oplus P'^* = (0, e_1)$
- $\Delta K = K \oplus K^* = K' \oplus K'^* = (0, e_1, 0, 0)$
- $\Delta K' = K \oplus K' = K^* \oplus K'^* = (0, 0, e_1, 0)$

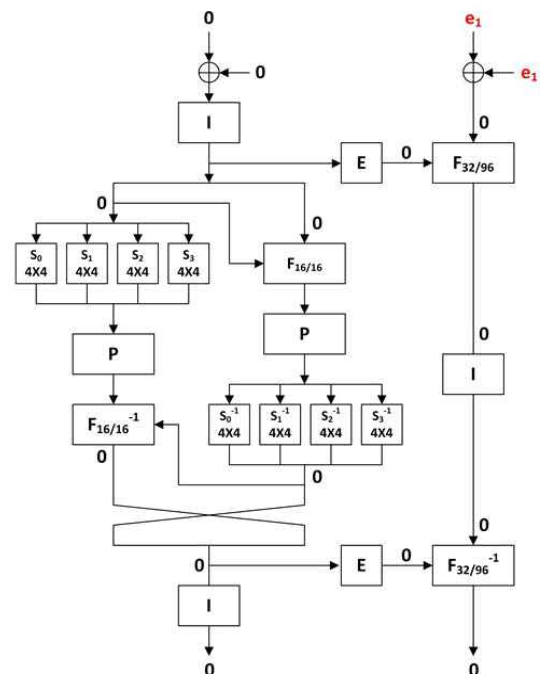


(그림 4) (a)  $F_{16/16}$  (b)  $F_{16/16}^{-1}$

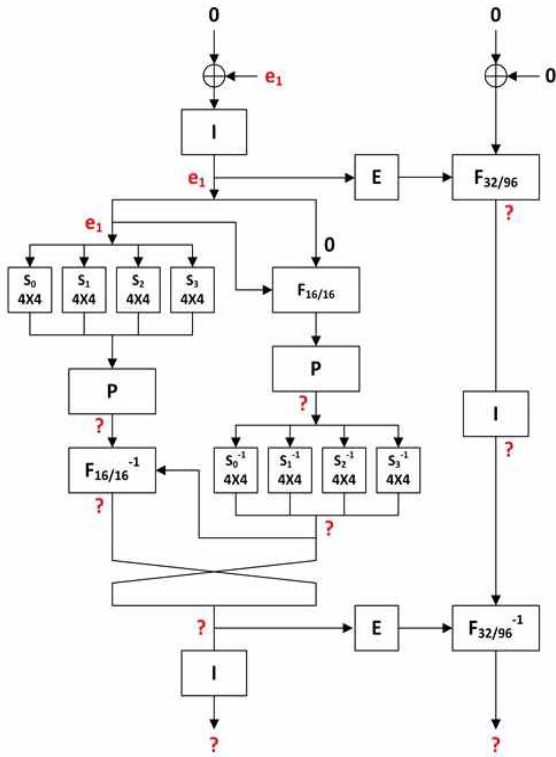
그러면 <표 3>과 같이, 라운드 1 ~ 5에 대한 확률 1의 5-라운드 연관키 차분 특성  $\alpha \rightarrow \beta = (?, ?)$ 을 구성할 수 있다. 표에서  $\Delta I_r$ 은 라운드  $r$ 의 입력 차분을 의미한다. (그림 5)와 (그림 6)은 라운드 1과 라운드 5에서의 차분 특성을 나타낸 것이다.

<표 3> KT-64에 대한 연관키 차분 특성

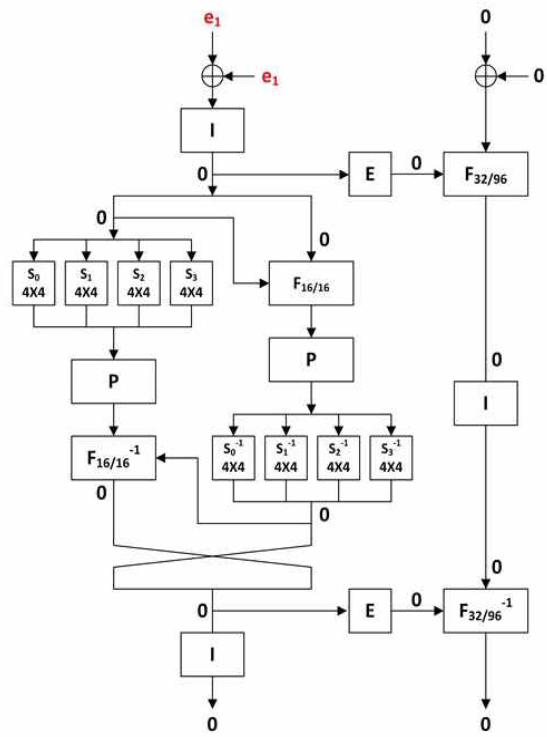
Round $r$	$\Delta I_r$	$(\Delta Q_r, \Delta U_r)$	확률
1	$(0, e_1) = \alpha$	$(0, e_1)$	1
2	$(0, 0)$	$(0, 0)$	1
3	$(0, 0)$	$(0, 0)$	1
4	$(0, 0)$	$(0, 0)$	1
5	$(0, 0)$	$(e_1, 0)$	1
Output	$(?, ?) = \beta$	.	.
6	$(e_1, 0) = \gamma$	$(0, e_1)$	1
7	$(0, 0)$	$(0, 0)$	1
8	$(0, 0)$	$(0, e_1)$	$2^{-12}$
Output	$(0, e_1) = \delta$	.	.
FT	$(0, e_1)$	$(0, e_1)$	1
Output	$(0, 0)$	.	.



(그림 5) 라운드 1의 차분 특성



(그림 6) 라운드 5의 차분 특성



(그림 7) 라운드 6의 차분 특성

이와 유사하게 라운드 6 ~ 8에 대한 3-라운드 연관키 차분 특성도 구성할 수 있다. 중간 상태값  $I, I^*, I', I'^*$ 를 비밀키  $K, K^*, K', K'^*$ 로 각각 암호화한다고 가정한다. 이때 각각의 중간값은 다음 조건을 만족한다:  $\gamma = I \oplus I' = I^* \oplus I'^* = (e_1, 0)$ . 그러면 <표 3>과 같이, 라운드 6 ~ 8에 대한 확률  $2^{-12}$ 의 3-라운드 연관키 차분 특성  $\gamma \rightarrow \delta = (0, e_1)$ 를 구성할 수 있다. (그림 7)은 라운드 6의 차분 특성을, (그림 8)은 라운드 8과 FT의 차분 특성을 나타낸 것이다. 먼저,  $F_{2/1}$ 는 다음과 같은 특성을 만족한다.

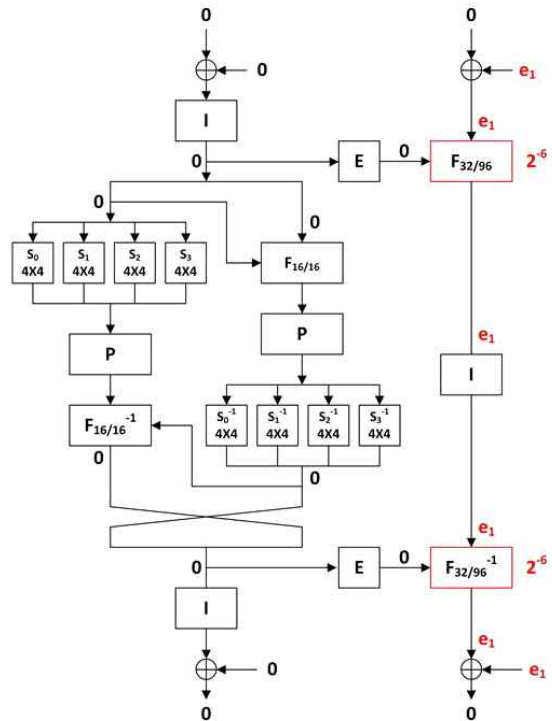
$$\Pr[F_{2/1}(x_1, x_2, v) \oplus F_{2/1}(x_1 \oplus 1, x_2, v) = (1, 0)] = 2^{-1}$$

이를 이용하여  $F_{32/96}$ 와  $F_{32/96}^{-1}$ 는 다음을 만족함을 쉽게 알 수 있다.

- $\Pr[F_{32/96}(X) \oplus F_{32/96}(X \oplus e_1) = e_1] = 2^{-6}$
- $\Pr[F_{32/96}^{-1}(X) \oplus F_{32/96}^{-1}(X \oplus e_1) = e_1] = 2^{-6}$

따라서 라운드 8에 대한 확률  $2^{-12}$ 의 연관키 차분 특성  $(0, 0) \rightarrow (0, e_1)$ 을 구성할 수 있다.

8-라운드 확장된 연관키 부메랑 차분 특성을 이용하여 전체 라운드 KT-64에 대한 확장된 연관키 부메랑 공격을



(그림 8) 라운드 8과 FT의 차분 특성

수행한다. KT-64가  $\Delta K = K \oplus K^* = K' \oplus K'^* = (0, e_1, 0, 0)$ ,  $\Delta K' = K \oplus K' = K^* \oplus K'^* = (0, 0, e_1, 0)$ 을 만족하는 비밀키  $K$ 와 연관키  $K^*, K', K'^*$ 를 사용한

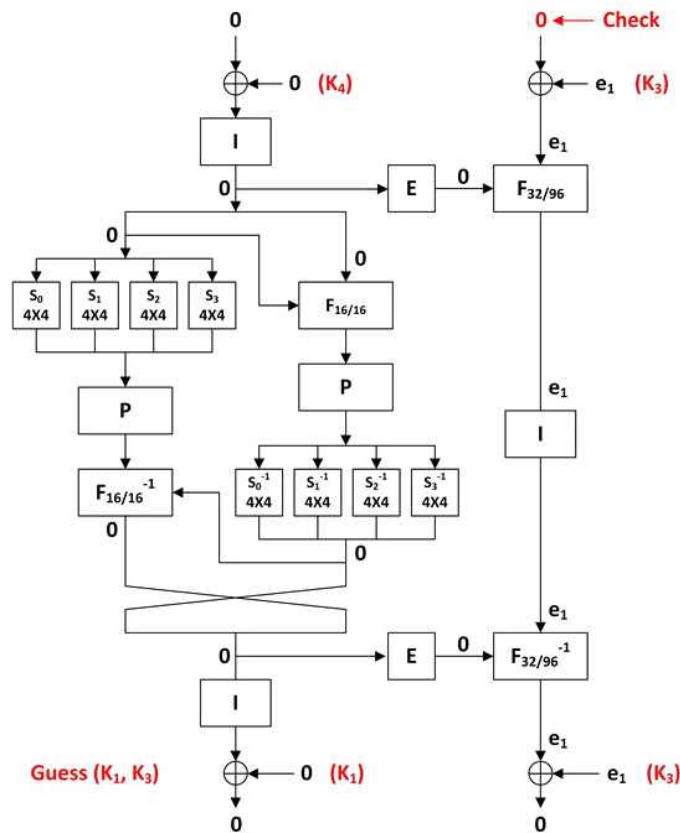
다고 가정할 때, 전체 라운드 KT-64에 대한 확장된 연관키 부메랑 공격은 다음과 같은 과정을 수행한다.

1. 차분  $\alpha = (0, e_1)$ 을 만족하는  $2^{44.5}$ 개의 평문 쌍  $(P_j, P_j^*)$ 를 선택한다 ( $j = 1, \dots, 2^{44.5}$ ). 이를 이용하여  $2^{88}$ 개의 평문 quartet  $(P_i, P_i^*, P_i', P_i'^*)$ 를 구성한 후, 비밀키  $K$ 와 연관키  $K^*, K', K'^*$ 로 각각 암호화하여 대응되는 암호문 quartet  $(C_i, C_i^*, C_i', C_i'^*)$ 를 계산하고 테이블에 저장한다.
2. 각각의  $i$ 에 대해,  $C_i \oplus C_i' = C_i^* \oplus C_i'^* = (0, 0)$ 을 만족하는지 체크하고 이를 만족하는 암호문 quartet에 대해 단계 3을 수행한다.
3. 64-비트 부분키  $(K_1, K_3)$ 을 추측한 후, 다음을 수행한다.
  - (1) 추측한  $(K_1, K_3)$ 을 이용하여  $(K_1^*, K_3^*), (K_1', K_3')$ 를 계산한다.
  - (2) 추측한 부분키 quartet을 이용하여 라운드 8의 오른쪽 입력값 32-비트를 각각 계산한다 ((그림 9) 참조). 이 32-비트 값을  $(T_i, T_i^*, T_i', T_i'^*)$ 라 할 때, 각각의  $i$ 에 대해  $T_i \oplus T_i' = T_i^* \oplus T_i'^* = 0$ 을 만족하는지 검사한다. 이를 만족하는 부분키 quartet에 대해, 단계 4를 수행한다.

4. 단계 3을 통과한 부분키 quartet에 대해, 나머지 64-비트 부분키  $(K_2, K_4)$ 를 전수조사하고 2개의 평문/암호문 쌍을 이용하여 검사한다. 이를 만족하는 128-비트 비밀키를 KT-64의 옴은 128-비트 비밀키로 출력하고 연관키  $K^*, K', K'^*$ 도 출력한다. 그렇지 않으면 단계 3으로 간다.

이 공격을 수행하기 위해  $2^{44.5}$ 개의 선택 평문 쌍이 필요하므로 이 공격의 데이터 복잡도는  $2^{45.5}$ 개의 연관키 선택 평문이다. 그리고 이 공격에 필요한 메모리는  $2^{48.5} (= 2^{44.5} \cdot 2 \cdot 8)$  메모리 바이트이다.

단계 1의 계산 복잡도는  $2^{45.5}$  KT-64 암호화 연산이다. 각각의 암호문 quartet이 단계 2를 통과할 확률은  $2^{-128} (= (2^{-64})^2)$ 이다. 그래서 단계 2를 통과할 틀린 quartet의 개수의 기댓값은  $2^{-40} (= 2^{88} \cdot 2^{-128})$ 이다. 이는 옴은 quartet만이 단계 2를 통과함을 의미한다. 단계 3의 계산 복잡도는 평균  $2^{62} (= 2^{64} \cdot 4 \cdot 1/8 \cdot 1/2)$  KT-64 암호화 연산이다. 틀린 부분키 quartet이 단계 3을 통과할 확률은  $2^{-64} (= (2^{-32})^2)$ 이다. 그래서 1개의 후보 부분키 quartet이 단계 3을 통과한다. 단계 4의 계산 복잡도는  $2^{65} (= 2^{64} \cdot 1 \cdot 2)$  KT-64 암호화 연산이다. 그러므로 이



(그림 9) KT-64에 대한 공격 과정

공격 알고리즘의 계산 복잡도는 약  $2^{65.17}$  KT-64 암호화 연산이다 ( $2^{65.17} (\approx 2^{45.5} + 2^{62} + 2^{65})$ ).

한편, 각각의 128-비트 틀린 비밀키가 단계 4를 통과할 확률은  $2^{-128} (= (2^{-64})^2)$ 이므로,  $2^{-64} (= 2^{64} \cdot 2^{-128})$  개의 틀린 비밀키가 단계 4를 통과한다. 이는 본 논문에서 제안하는 공격 알고리즘이 틀린 비밀키 quartet을 출력할 확률이 매우 낮음을 의미한다. 그러므로 본 논문에서 제안하는 확장된 연관키 부메랑 공격은 KT-64의 128-비트 비밀키를 복구할 수 있다.

#### 4. 결 론

본 논문에서는 확장된 연관키 부메랑 공격을 이용하여 64-비트 블록 암호 KT-64에 대한 첫 번째 안전성 분석 결과를 제안하였다. 본 논문에서 소개한 공격은 전수조사보다 효율적인  $2^{65.17}$ 의 계산 복잡도를 필요로 한다. 이는 KT-64가 기제안된 DDP-기반 블록 암호와 마찬가지로 연관키 공격에 매우 취약함을 의미한다.

#### 참 고 문 헌

- [1] N. Goots, A. Moldovyan and N. Moldovyan, "Fast Encryption Algorithm Spectr-H64," MMM-ACNS'01, LNCS 2052, pp.275-286, 2001.
- [2] A. Moldovyan and N. Moldovyan, "A cipher Based on Data-Dependent Permutations," Journal of Cryptology, Vol.15, No.1, pp.61-72, 2002.
- [3] N. Goots, B. Izotov, A. Moldovyan and N. Moldovyan, "Modern cryptography: Protect Your Data with Fast Block Ciphers," Wayne, A-LIST Publish., 2003.
- [4] N. Goots, N. Moldovyan, P. Moldovyanu and D. Summerville, "Fast DDP-Based Ciphers: From Hardware to Software," 46th IEEE Midwest International Symposium on Circuits and Systems, 2003.
- [5] N. Sklavos, N. Moldovyan and O. Koufopavlou, "High Speed Networking Security: Design and Implementation of Two New DDP-Based Ciphers," Mobile Networks and Applications-MONET, Kluwer Academic Publishers, Vol.25, Issue1-2, pp.219-231, 2005.
- [6] N. Moldovyan, A. Moldovyan, M. Ereemeev and D. Summerville, "Wireless Networks Security and Cipher Design Based on Data-Dependent Operations: Classification of the FPGA Suitable Controlled Elements," CCCT'04, Vol.VII, pp.123-128, Texas, USA, 2004.
- [7] N. Moldovyan, A. Moldovyan, M. Ereemeev and N. Sklavos, "New Class of Cryptographic Primitives and Cipher Design for Networks Security," International Journal of Network Security, Vol.2, No.2, pp.114-225, 2006.
- [8] N. Moldovyan, "On Cipher Design Based on Switchable Controlled Operations," MMM-ACNS'03, LNCS 2776, pp.316-327, 2003.
- [9] Y. Ko, D. Hong, S. Hong, S. Lee and J. Lim, "Linear Cryptanalysis on SPECTR-H64 with Higher Order Differential Property," MMM-ACNS'03, LNCS 2776, pp.298-307, 2003.
- [10] Y. Ko, C. Lee, S. Hong and S. Lee, "Related Key Differential Cryptanalysis of Full-Round SPECTR-H64 and CIKS-1," ACISP'04, LNCS 3108, pp.137-148, 2004.
- [11] Y. Ko, C. Lee, S. Hong, J. Sung and S. Lee, "Related-Key Attacks on DDP based Ciphers: CIKS-128 and CIKS-128H," Indocrypt'04, LNCS 3348, pp.191-205, 2004.
- [12] C. Lee, D. Hong, S. Lee, S. Lee, H. Yang and J. Lim, "A Chosen Plaintext Linear Attack on Block Cipher CIKS-1," ICICS'02, LNCS 2513, pp.456-468, 2002.
- [13] C. Lee, J. Kim, S. Hong, J. Sung and S. Lee, "Related-Key Differential Attacks on Cobra-S128, Cobra-F64a, and Cobra-F64b," MYCRYPT'05, LNCS 3715, pp.245-263, 2005.
- [14] C. Lee, J. Kim, J. Sung, S. Hong and S. Lee, "Related-Key Differential Attacks on Cobra-H64 and Cobra-H128," CCC'05, LNCS 3796, pp.201-219, 2005.
- [15] J. Lu, C. Lee and J. Kim, "Related-Key Attacks on the Full-Round Cobra-F64a and Cobra-F64b," SCN'06, LNCS 4116, pp.95-110, 2006.
- [16] K. Jeong, C. Lee, J. Sung, S. Hong and J. Lim, "Related-Key Amplified Boomerang Attacks on the Full-Round Eagle-64 and Eagle-128," ACISP'07, LNCS 4586, pp.143-157, 2007.
- [17] K. Jeong, C. Lee, J. Kim and S. Hong, "Security analysis of the SCO-family using key schedules," Information Sciences, Vol.179, pp.4232-4242, 2009.
- [18] N. Minh, N. Luan and L. Dung, "KT-64: A New Block Cipher Suitable to Efficient FPGA Implementation," IJCSNS, Vol.19, No.1, pp.10-18, 2010.



#### 강진건

e-mail : jeangun@korea.ac.kr

2007년 8월 고려대학교 산업시스템정보  
공학과(학사)

2007년 9월~현 재 고려대학교 정보보호  
대학원 석·박사통합과정

관심분야: 대칭키 암호의 설계 및 분석



### 정 기 태

e-mail : kite.jeong@gmail.com

2004년 2월 고려대학교 수학과(학사)

2006년 2월 고려대학교 정보보호대학원  
(석사)

2011년 8월 고려대학교 정보보호대학원  
(박사)

2011년 9월~현재 고려대학교 정보보호연구원 박사후연구원  
관심분야: 대칭키 암호의 설계 및 분석



### 이 창 훈

e-mail : chlee@hs.ac.kr

2001년 2월 한양대학교 수학과(학사)

2003년 2월 고려대학교 정보보호대학원  
(석사)

2008년 2월 고려대학교 정보경영공학전문  
대학원(박사)

2009년 3월~2011년 2월 한신대학교 컴퓨터공학부 전임강사

2011년 3월~현재 한신대학교 컴퓨터공학부 조교수

관심분야: 정보보호, 암호학, 디지털포렌식