情報保護學會論文誌
第19卷 第1號, 2009. 2

블록 암호 Crypton, mCrypton에 대한 충돌 공격*

김 태 웅,[1†] 김 종 성,[1] 정 기 태,[1] 성 재 철,[2‡] 이 상 진[1]

[1]고려대학교 정보보호기술연구센터, [2]서울시립대학교 수학과

Collision Attacks on Crypton and mCrypton*

Taewoong Kim,[1†] Jongsung Kim,[1] Kitae Jeong,[1] Jaechul Sung,[2‡] Sangjin Lee[1]

[1]Center for Information of Security of Technologies, Korea University, [2]Department of Mathematics, University of Seoul

요 약

H. Gilbert 등은 [5]에서 7-라운드 Rijndael-192/256에 대한 충돌 공격을 제안하였다. 이 공격을 이용하여, 본 논문에서는 2^{96}개의 선택 평문과 $2^{161.6}$의 시간 복잡도로 8-라운드 Crypton의 부분키를 복구할 수 있음을 보인다. 또한 8-라운드 mCrypton에 대하여, 2^{48}개의 선택 평문과 $2^{81.6}$의 시간 복잡도로 부분키를 복구할 수 있음을 보인다. 본 논문의 공격 결과는 기제안된 Crypton과 mCrypton에 대한 공격 중 최대 라운드에 대한 결과이다.

ABSTRACT

H. Gilbert et al. proposed a collision attack on 7-round reduced Rijndael[5]. Applying this attack, we propose collision attacks on 8-round reduced Crypton, 8-round reduced mCrypton in this paper. Attacks on Crypton requires $2^{161.6}$ time complexity with 2^{96} chosen plaintexts, respectively. The attack on mCrypton requires $2^{81.6}$ time complexity with 2^{48} chosen plaintexts. These results are the best attacks on Crypton and mCrypton in published literatures.

Keywords : collision attack, birthday paradox, Crypton, mCrypton, SPN

Ⅰ. 서 론

Crypton[2]은 NIST에서 공모한 AES proposal에 제안된 128 비트 블록 암호로서 총 12-라운드 SPN 구조이며 최대 256 비트 비밀 키를 사용한다. 또한 involution 함수를 이용하며 키 스케줄 과정을 제외하고 암호화 과정과 복호화 과정이 동일하도록 설계되었다. 현재까지 제안된 Crypton에 대한 공격 결과는 [1]에서 제안된 6-라운드 square 공격과 5/6-라운드 Crypton에 대한 불능 차분 공격[4,7], 8-라운드 Crypton에 대한 stochastic 공격[12] 등이 있다. mCrypton[13]은 WISA 2005에 제안된 64 비트 블록 암호로서, 유비쿼터스 환경에 적합하도록 Crypton의 축소 버전으로 설계되었다. 총 12-라운드로 구성되어 있으며 96/128 비트 비밀키를 사용한다. 현재까지 mCrypton에 대한 공격 결과가 제안되지 않았다.

[5]에서 7-라운드 Rijndael-192/256에 대한 충돌 공격을 제안하였는데 본 논문에서는 [5]과 [10]에서 제안된 공격 기법을 Crypton과 mCrypton에 적용하여 7/8-라운드 Crypton과 mCrypton을 제안한다. 본 논문에서 제안하는 공격을 요약하면 [표 1]과 같다.

접수일(2008년 9월 20일), 게재확정일(2008년 12월 30일)
* 이 연구에 참여한 연구자(의 일부)는 '2단계 BK21사업'의 지원비를 받았음
† 주저자, woongchiki@nate.com
‡ 교신저자, jcsung@uos.ac.kr

[표 1] Crypton, mCrypton에 대한 공격 결과

알고리즘	공격 방법	라운드 수	데이터 복잡도	시간 복잡도
Crypton (12-라운드)	불능 차분 공격[4]	5	$2^{83.4}$	2^{43}
	불능 차분 공격[7]	6	2^{91}	2^{124}
	포화 공격[1]	6	2^{32}	2^{56}
	부정 차분 공격[10]	7	2^{97}(2^{100}메모리)	$2^{97.2}$
	부정 차분 공격[10]	8	2^{126}(2^{100}메모리)	$2^{126.2}$
	stochastic 공격[12]	8	2^{112}	2^{112}
	충돌 공격 (본 논문)	7	2^{32}(2^{74}메모리)	$2^{69.2}$
	충돌 공격 (본 논문)	8	2^{96}(2^{138}메모리)	$2^{161.6}$
mCrypton (12-라운드)	충돌 공격 (본 논문)	7	2^{16}(2^{38}메모리)	$2^{33.2}$
	충돌 공격 (본 논문)	8	2^{48}(2^{65}메모리)	$2^{81.6}$

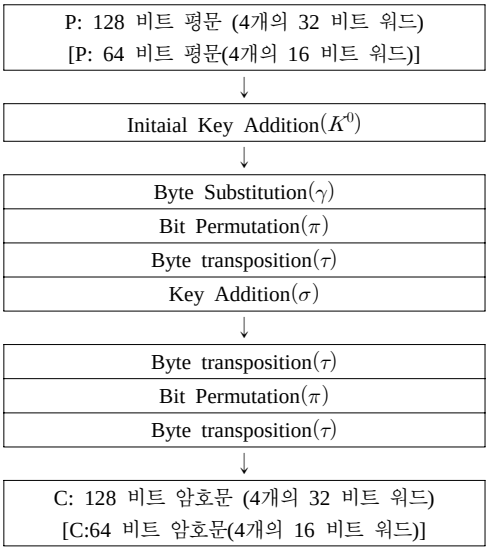
Ⅱ. Crypton, mCrypton 소개

본 장에서는 블록 암호 Crypton과 mCrypton을 소개한다. mCrypton은 Crypton의 축소 버전으로서, 바이트 단위 함수를 사용하는 Crypton에 비해, 4 비트 단위 함수를 사용하지만 전체 구조는 Crypton과 매우 유사하다. 따라서 본 장에서는 Crypton을 중심으로 소개한다. 키 스케줄은 본 논문에서 제안하는 공격에 사용되지 않으므로 생략한다[2,13].

Crypton은 128 비트 평문 P를 입력 받아, 최대 256 비트 비밀키를 이용하여 [그림 1]과 같은 과정을 통해 128 비트 암호문 C를 출력한다. 본 논문에서는 128 비트 데이터를 다음과 같은 4×4 바이트 단위 행렬로 표기한다.

$$A=(A[3],A[2],A[1],A[0])^t=\begin{pmatrix}A[0]\\A[1]\\A[2]\\A[3]\end{pmatrix}=\begin{pmatrix}a_{30}\,a_{20}\,a_{10}\,a_{00}\\a_{31}\,a_{21}\,a_{11}\,a_{01}\\a_{32}\,a_{22}\,a_{12}\,a_{02}\\a_{33}\,a_{23}\,a_{13}\,a_{03}\end{pmatrix}$$

Crypton의 라운드 함수 ρ는 네 개의 함수 γ, π, τ, σ로 구성된다([그림 1] 참조). 그리고 ρ는 홀수 라운드에서 사용되는 라운드 함수 ρ_o와 짝수 라운드에서 사용되는 ρ_e로 나뉜다. ρ_o와 ρ_e는 다음과 같이 정의된다. 여기서 K는 라운드 키를 의미한다: $\rho_o=\sigma_K\circ\tau\circ\pi_o\circ\gamma_o$, $\rho_e=\sigma_K\circ\tau\circ\pi_e\circ\gamma_e$. 마지막 라운드 $\phi=\tau\circ\pi\circ\tau$이다.



[그림 1] Crypton(대괄호안은 mCrypton)

γ_o와 γ_e는 4개의 8 비트 S-박스 S_1, S_2, S_3, S_4를 사용한다. 이 S-박스들은 $S_2=S_0^{-1}$, $S_3=S_1^{-1}$, $\gamma_o(\gamma_e(A))=\gamma_e(\gamma_o(A))=A$을 만족한다. (mCrypton의 γ_o, γ_e는 Crypton와 동일한 구조를 가지며, 4 비트 4개와 S-박스를 사용하며 인덱스는 다르다.)

S_3	S_2	S_1	S_0
S_0	S_3	S_2	S_1
S_1	S_0	S_3	S_2
S_2	S_1	S_0	S_3

γ_o

S_1	S_0	S_3	S_2
S_2	S_1	S_0	S_3
S_3	S_2	S_1	S_0
S_0	S_3	S_2	S_1

γ_e

π_o와 π_e는 다음과 같은 4개의 선형 함수 π_0, π_1, π_2, π_3로 구성된다. 각각의 π_i는 $\pi_i^{-1}=\pi_i$를 만족하며, 전체적으로 $\pi_o^{-1}=\pi_o$, $\pi_e^{-1}=\pi_e$이 성립한다(단,$m_0=fc_x$,$m_1=f3_x$, $m_2=cf_x$,$m_3=3f_x$, $(i=0,1,2,3)$). mCrypton의 π는 매 라운드 같은 값을 사용하며, 4 비트 m_i상수를 사용한다.(단, $m_0=1110_2$, $m_1=1101_2$, $m_2=1011_2$,$m_3=0111_2$이다.)

$$\begin{pmatrix}b_0\\b_1\\b_2\\b_3\end{pmatrix}=\pi_0\begin{pmatrix}a_0\\a_1\\a_2\\a_3\end{pmatrix},\quad \begin{pmatrix}b_1\\b_2\\b_3\\b_0\end{pmatrix}=\pi_1\begin{pmatrix}a_0\\a_1\\a_2\\a_3\end{pmatrix},\quad \begin{pmatrix}b_2\\b_3\\b_0\\b_1\end{pmatrix}=\pi_2\begin{pmatrix}a_0\\a_1\\a_2\\a_3\end{pmatrix},\quad \begin{pmatrix}b_3\\b_0\\b_1\\b_2\end{pmatrix}=\pi_3\begin{pmatrix}a_0\\a_1\\a_2\\a_3\end{pmatrix}.$$

$$b_0\leftarrow(a_3\wedge m_3)\oplus(a_2\wedge m_2)\oplus(a_1\wedge m_1)\oplus(a_0\wedge m_0),$$
$$b_1\leftarrow(a_3\wedge m_0)\oplus(a_2\wedge m_3)\oplus(a_1\wedge m_2)\oplus(a_0\wedge m_1),$$
$$b_2\leftarrow(a_3\wedge m_1)\oplus(a_2\wedge m_0)\oplus(a_1\wedge m_3)\oplus(a_0\wedge m_2),$$
$$b_3\leftarrow(a_3\wedge m_2)\oplus(a_2\wedge m_1)\oplus(a_1\wedge m_0)\oplus(a_0\wedge m_3).$$

π_3	π_2	π_1	π_0

π_o ,

π_1	π_0	π_3	π_2

π_e

τ는 다음과 같은 선형 함수로서, a_{ij}에서 a_{ji}로 변환하는 전치 함수이고 $\tau^{-1}=\tau$을 만족한다.

a_{30}	a_{20}	a_{10}	a_{00}
a_{31}	a_{21}	a_{11}	a_{01}
a_{32}	a_{22}	a_{12}	a_{02}
a_{33}	a_{23}	a_{13}	a_{03}

$\Rightarrow$

a_{03}	a_{02}	a_{01}	a_{00}
a_{13}	a_{12}	a_{11}	a_{10}
a_{23}	a_{22}	a_{21}	a_{20}
a_{33}	a_{32}	a_{31}	a_{30}

라운드 키 덧셈 함수 σ_K는 다음과 같이 정의된다: $B=\sigma_K(A)$, $(B=A\oplus K)$.

Ⅲ. 4-라운드 Crypton, mCrypton의 충돌 특성

본 장에서는 [5]에서 제안된 공격 기법을 Crypton, mCrypton에 적용하여 얻은 4-라운드 충돌 특성을 소개한다. 블록 암호에서 동일한 입력 값은 동일한 출력 값을 가지며, 서로 다른 입력 값은 상이한 출력 값을 갖는다. 하지만 서로 다른 입력 값에 대하여 특정 부분을 관찰할 때, 출력 값이 같은 경우가 존재할 수 있다. 만약 출력 값의 특정 부분이 동일하다면, 이를 충돌이라 한다.

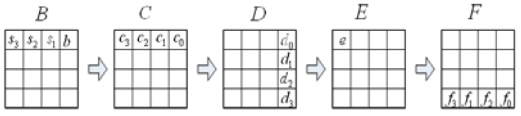
3.1 표기법

A,B,C,D,E는 각 1, 2, 3, 4, 5번째 라운드의 입력 상태 값을 의미하며, F는 5번째 라운드의 출력 상태 값을 의미한다. 입력 상태 값 B는 $b=b_{00}$를 제외한 15개의 바이트 위치에서 상수 값을 갖는다. 즉, b_{00}을 제외한 b_{ij}는 상수 값을 갖는다. 특히 24 비트(3개의 워드 s값) 상수는 변하는 상수 값을 의미하며 b_{ij}는 고정된 상수 값이다 $(0\le i,j\le 3)$. 또한 상수 s와 $b=\{0,1,\cdots,255\}$와 관련된 c_{ij}, d_{ij}, e_{ij}, f_{ij} 바이트를 $c_{ij}^s[b]$,$d_{ij}^s[b]$,$e_{ij}^s[b]$$f_{ij}^s[b]$로 표기한다 $(0\le i,j\le 3)$. [그림 3]의 연속된 4-라운드의 상태 값 중 충돌 특성에 관여하는 바이트들을 위의 표기법에 의해 다음과 같이 표기한다.

$$c_0=c_{00}^s[b],\ c_1=c_{10}^s[b],\ c_2=c_{20}^s[b],\ c_3=c_{30}^s[b],$$
$$d_0=d_{00}^s[b],\ d_1=d_{01}^s[b],\ d_2=d_{02}^s[b],\ d_3=d_{03}^s[b],$$
$$e^s[b]=e_{30}^s[b],$$
$$f_0=f_{03}^s[b],\ f_1=f_{13}^s[b],\ f_2=f_{23}^s[b],\ f_3=f_{33}^s[b].$$

3.2 두 알고리즘의 4-라운드 특성 및 7-라운드 공격

본 절에서는 Crypton과 mCrypton의 암호화 과정을 함수로 표현하고, 암호화 과정 중 16개의 바이트 상태 값의 상호 의존성에 대해 살펴본다. 특히 c_i, d_i, e, f_i의 상태 값을 관찰한다 $(0\le i\le 3)$. 먼저, Crypton에 대한 충돌 특성을 다룬 후에 mCrypton 충돌 특성을 살펴본다.



[그림 3] Crypton, mCrypton의 연속된 4-라운드의 상태 값

다음은 2번째 라운드를 b_{ij}라고 할 때 3번째 라운드의 입력 상태 값, c_i에 대해 설명한다.$(0\le i,j\le 3)$ [그림 3]에서 $c_0^s[b]$는 한 바이트 라운드 키 K_{30}^2 에 의해 결정되며, 고정된 상수 $s=(s_1,s_2,s_3)$는 $c_0^s[b]$ 값에 영향을 미치지 않는다. 그리고 c_0는 일대일 대응 함수 $b\rightarrow c_0^s[b]$으로 표현 가능하다. 또한 $b\rightarrow c_1^s[b]$, $b\rightarrow c_2^s[b]$, $b\rightarrow c_3^s[b]$에 대해서도 이와 같은 성질을 만족함을 확인할 수 있다. 따라서 c_i는 키 의존-상수임을 알 수 있다 $(0\le i\le 3)$.

다음은 3번째 라운드를 c_{ij}라고 할 때 4번째 라운드의 입력 상태 값, d_i에 대해 설명한다.$(0\le i,j\le 3)$.

$d_0^s[b]$는 한 바이트 라운드 키 K_{00}^3 와 c_{01},c_{02},c_{03}에 의해 결정된다. c_{01}은 s_1과 K_{01}^2 , c_{02}는 s_2과 K_{02}^2 , c_{03}는 s_3와 K_{03}^2 에 의해 결정된다. 하나의 고정된 상수 s에 대해서 $c_0^s[b]\rightarrow d_0^s[b]$는 일대일 대응 함수이며, 결론적으로, 합성 함수 $b\rightarrow c_0^s[b]\rightarrow d_0^s[b]$은 일대일 대응 관계를 만족한다. 또

한 $b \rightarrow c_1^s[b] \rightarrow d_1^s[b]$, $b \rightarrow c_2^s[b] \rightarrow d_2^s[b]$, $b \rightarrow c_3^s[b] \rightarrow d_3^s[b]$에 대해서도 같은 성질을 만족한다. 따라서 d_i는 하나의 키-의존 상수, 세 개의 상수 s와 키-의존 상수, 하나의 키에 의해 결정된다($b_{00} = b = \{0,1,\cdots,255\}$을 제외하고 b_{ij}는 상수 값을 갖는 포화 집합의 두 라운드 후의 출력 상태 값을 관찰해 보면 일대일 대응 관계를 이용하여 모든 d_{ij}는 포화 집합이 됨을 알 수 있다).

다음 식은 4번째 라운드 입력 상태 값을 d_{ij}라 할 때, 5번째 라운드 입력 상태 값 e를 나타낸다 .

$$
\begin{aligned}
e = {} & (S_1(S_3(S_1(b) \wedge m_3) \oplus (S_2(b_{01}) \wedge m_0) \oplus (S_3(b_{02}) \wedge m_1) \oplus (S_0(b_{03}) \wedge m_2) \oplus K_{00}^2) \wedge m_3) \oplus \\
& \boxed{\begin{array}{l}(S_0(S_0(s_1) \wedge m_2) \oplus (S_1(b_{11}) \wedge m_3) \oplus (S_2(b_{12}) \wedge m_0) \oplus (S_3(b_{13}) \wedge m_1) \oplus K_{01}^2) \wedge m_0) \oplus \\ (S_1(S_3(s_2) \wedge m_1) \oplus (S_0(b_{21}) \wedge m_2) \oplus (S_1(b_{22}) \wedge m_3) \oplus (S_2(b_{23}) \wedge m_0) \oplus K_{02}^2) \wedge m_1) \oplus \\ (S_2(S_2(s_3) \wedge m_0) \oplus (S_3(b_{31}) \wedge m_1) \oplus (S_0(b_{32}) \wedge m_2) \oplus (S_1(b_{33}) \wedge m_3) \oplus K_{03}^2) \wedge m_2) \oplus K_{00}^3) \wedge m_2) \oplus \\
& (S_2(S_2(S_1(b) \wedge m_0) \oplus (S_2(b_{01}) \wedge m_1) \oplus (S_3(b_{02}) \wedge m_2) \oplus (S_0(b_{03}) \wedge m_3) \oplus K_{10}^2) \wedge m_2) \oplus \\
& \boxed{\begin{array}{l}(S_3(S_0(s_1) \wedge m_3) \oplus (S_1(b_{11}) \wedge m_0) \oplus (S_2(b_{12}) \wedge m_1) \oplus (S_3(b_{13}) \wedge m_2) \oplus K_{11}^2) \wedge m_3) \oplus \\ (S_0(S_3(s_2) \wedge m_2) \oplus (S_0(b_{21}) \wedge m_3) \oplus (S_1(b_{22}) \wedge m_0) \oplus (S_2(b_{23}) \wedge m_1) \oplus K_{12}^2) \wedge m_0) \oplus \\ (S_1(S_2(s_3) \wedge m_1) \oplus (S_3(b_{31}) \wedge m_2) \oplus (S_0(b_{32}) \wedge m_3) \oplus (S_1(b_{33}) \wedge m_0) \oplus K_{13}^2) \wedge m_1) \oplus K_{01}^3) \wedge m_3) \oplus \\
& (S_3(S_3(S_1(b) \wedge m_1) \oplus (S_2(b_{01}) \wedge m_2) \oplus (S_3(b_{02}) \wedge m_3) \oplus (S_0(b_{03}) \wedge m_0) \oplus K_{20}^2) \wedge m_1) \oplus \\
& \boxed{\begin{array}{l}(S_2(S_0(s_1) \wedge m_0) \oplus (S_1(b_{11}) \wedge m_1) \oplus (S_2(b_{12}) \wedge m_2) \oplus (S_3(b_{13}) \wedge m_3) \oplus K_{21}^2) \wedge m_2) \oplus \\ (S_3(S_3(s_2) \wedge m_3) \oplus (S_0(b_{21}) \wedge m_0) \oplus (S_1(b_{22}) \wedge m_1) \oplus (S_2(b_{23}) \wedge m_2) \oplus K_{22}^2) \wedge m_3) \oplus \\ (S_0(S_2(s_3) \wedge m_2) \oplus (S_3(b_{31}) \wedge m_3) \oplus (S_0(b_{32}) \wedge m_0) \oplus (S_1(b_{33}) \wedge m_1) \oplus K_{23}^2) \wedge m_0) \oplus K_{02}^3) \wedge m_0) \oplus \\
& (S_0(S_0(S_1(b) \wedge m_2) \oplus (S_2(b_{01}) \wedge m_3) \oplus (S_3(b_{02}) \wedge m_0) \oplus (S_0(b_{03}) \wedge m_1) \oplus K_{30}^2) \wedge m_0) \oplus \\
& \boxed{\begin{array}{l}(S_1(S_0(s_1) \wedge m_1) \oplus (S_1(b_{11}) \wedge m_2) \oplus (S_2(b_{12}) \wedge m_3) \oplus (S_3(b_{13}) \wedge m_0) \oplus K_{31}^2) \wedge m_1) \oplus \\ (S_2(S_3(s_2) \wedge m_0) \oplus (S_0(b_{21}) \wedge m_1) \oplus (S_1(b_{22}) \wedge m_2) \oplus (S_2(b_{23}) \wedge m_3) \oplus K_{32}^2) \wedge m_2) \oplus \\ (S_3(S_2(s_3) \wedge m_3) \oplus (S_3(b_{31}) \wedge m_0) \oplus (S_0(b_{32}) \wedge m_1) \oplus (S_1(b_{33}) \wedge m_2) \oplus K_{33}^2) \wedge m_3) \oplus K_{03}^3) \wedge m_1) \\
& \oplus K_{10}^4
\end{aligned}
$$

위의 식을 통해 e는 네 개의 키-의존 상수, 네 개의 상수 s와 키-의존 상수 (네 개의 상자로 표시하며 각각은 8 비트이다.), 하나의 키에 의해 결정됨을 알 수 있다. 결론적으로 $e^s[b]$는 $s=(s_1,s_2,s_3)$에 의존하고 키에 의존하는 네 개의 상수와 $s=(s_1,s_2,s_3)$에 독립이고 키에 의존적인 다섯 개의 상수로 구성된다는 것을 알 수 있다. ($b_{00} = b = \{0,1,\cdots,255\}$을 제외하고 b_{ij}는 상수 값을 갖는 포화 집합의 3번째 라운드 후의 출력 상태 값을 관찰해 보면 e_{ij}는 균일 집합이 됨을 알 수 있다.) 만약 서로 다른 상수 $s' = (s_1', s_2', s_3')$와 $s'' = (s_1'', s_2'', s_3'')$에 대해서 네 개의 키와 상수 $s=(s_1,s_2,s_3)$에 의존하는 32비트 (네 개의 네모상자) 값이 동일하다면 전체적으로 e의 값이 동일하게 된다. 즉, 모든 $b=\{0,1,\cdots,255\}$에 대해서 $e^{s'}[b] = e^{s''}[b]$이 성립하는 $s' = (s_1', s_2', s_3')$와 $s'' = (s_1'', s_2'', s_3'')$는 생일 공격을 이용하여 2^{16}개의 서로 다른 $s=(s_1,s_2,s_3)$에 대해 50% 이상의 확률로 충돌 특성을 찾을 수 있다. 다음 과정은 f_{ij}를 이용하여 e를 표현하며 5번째 라운드의 입력 값을 E라 할 때, 5번째 라운드 과정은 다음과 같이 표현할 수 있다 $(0 \le i,j \le 3)$

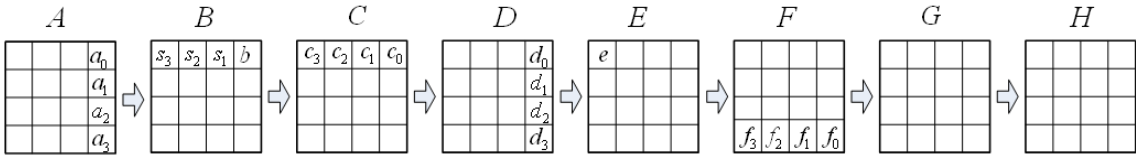
$$
\begin{aligned}
F &= \sigma_{K^5} \circ \tau \circ \pi_o \circ \gamma_o(E) \\
&= \tau \circ \pi_o \circ \sigma_{\pi_o^{-1} \circ \tau^{-1}(K^5)} \circ \gamma_o(E)
\end{aligned}
$$

따라서 $\sigma_{\pi_o^{-1} \circ \tau^{-1}(K^5)} \circ \pi_o^{-1} \circ \tau^{-1}(F) = \gamma_o(E)$이 성립한다. 또한 $\tau^{-1} = \tau$, $\pi_o^{-1} = \pi_o$이 성립하므로 $\sigma_{\pi_o \circ \tau(K^5)} \circ \pi_o \circ \tau(F) = \gamma_o(E)$임을 알 수 있다. 결론적으로, $S_3(e) = (f_0 \wedge m_2) \oplus (f_1 \wedge m_1) \oplus (f_2 \wedge m_0) \oplus (f_3 \wedge m_3) \oplus (\pi_o \circ \tau(K^5))_{30}$이 성립함을 알 수 있다. $e^{s'}[b] = e^{s''}[b]$을 만족하는 $s' = (s_1', s_2', s_3')$와 $s'' = (s_1'', s_2'', s_3'')$가 존재한다면 $S_3(e^{s'}[b]) = (f_0^{s'}[b] \wedge m_2) \oplus (f_1^{s'}[b] \wedge m_1) \oplus (f_2^{s'}[b] \wedge m_0) \oplus (f_3^{s'}[b] \wedge m_3) \oplus (\pi_o \circ \tau(K^5))_{30}$와 $S_3(e^{s''}[b]) = (f_0^{s''}[b] \wedge m_2) \oplus (f_1^{s''}[b] \wedge m_1) \oplus (f_2^{s''}[b] \wedge m_0) \oplus (f_3^{s''}[b] \wedge m_3) \oplus (\pi_o \circ \tau(K^5))_{30}$을 비교해 볼 때 다음을 만족한다.

$$
\begin{aligned}
& (f_0^{s'}[b] \wedge m_2) \oplus (f_1^{s'}[b] \wedge m_1) \oplus (f_2^{s'}[b] \wedge m_0) \oplus (f_3^{s'}[b] \wedge m_3) \\
&= (f_0^{s''}[b] \wedge m_2) \oplus (f_1^{s''}[b] \wedge m_1) \oplus (f_2^{s''}[b] \wedge m_0) \\
&\quad \oplus (f_3^{s''}[b] \wedge m_3) \qquad (1)
\end{aligned}
$$

모든 $b=\{0,1,\cdots,255\}$에 대해서 $e^{s'}[b] = e^{s''}[b]$의 성립 여부는 식 (1)의 성립 여부를 테스트함으로서 확인 가능하다. 따라서 $b=\{0,1,\cdots,255\}$에 대해 2^{16}개의 서로 다른 상수 $s=(s_1,s_2,s_3)$를 이용하여 50%의 확률로 모든 $b=\{0,1,\cdots,255\}$에 대하여 식 (1)이 성립할 s', s''을 찾을 수 있다.

Crypton과 마찬가지로 mCrypton은 5-라운드 입력 값(e_{30})인 e값은 네 개의 키-의존 상수, 네 개의 상수 s와 키-의존 상수 그리고 하나의 키에 의해 결정됨을 알 수 있으며 mCrypton의 5번째 라운드의 $e(=e_{00})$값은 Crypton의 상수값(m_0, m_1, m_2, m_3)과 S-박스만 다를 뿐 구조는 동일하므로 생략한다. 아래의 식 (2)는 mCrypton의 $e^{s'}[b] = e^{s''}[b]$를 만족하는 s', s''쌍을 찾기 위한 식으로 [알고리즘 2]의 과정 (4)-(ㄱ)에서 다시 언급한다.



[그림 4] 7-라운드 Crypton, mCrypton의 상태 값

$$(f_3^{s'}[b]\wedge m_0)\oplus(f_2^{s'}[b]\wedge m_1)\oplus(f_1^{s'}[b]\wedge m_2)\oplus(f_0^{s'}[b]\wedge m_3)$$
$$=(f_3^{s''}[b]\wedge m_0)\oplus(f_2^{s''}[b]\wedge m_1)\oplus(f_1^{s''}[b]\wedge m_2)$$
$$\oplus(f_0^{s''}[b]\wedge m_3) \quad (3)$$

Crypton, mCrypton의 4-라운드 충돌 특성은 식(1)과 (2)를 이용하여 다음 알고리즘을 통해 구할 수 있다.

• [알고리즘 1] 4-라운드 충돌 특성 찾기 (단, 대괄호 안은 mCrypton의 경우이다).

① 임의의 25개의 b를 갖는 집합 $\Lambda\subset\{0,1,\cdots,255\}$와 $2^{16}[2^8]$개의 서로 다른 상수 $s=(s_1,s_2,s_3)$를 갖는 25개로 이루어진 $2^{16}[2^8]$개의 평문 집합을 선택한다 (b와 $s=(s_1,s_2,s_3)$를 제외한 모든 부분은 고정된 상수 값을 갖는다). 7-라운드 충돌 공격에 사용되는 집합 Λ는 전체 추측하는 키 공간이 $32+40\times4=192$[96]비트이다. 한 쌍의 s값에 대해 충돌 할 확률은 $2^{-8}[2^{-4}]$이며 25개의 b값에 대해서 $(2^{-8})^{25}=2^{-200}[(2^{-4})^{25}=2^{-100}]$이므로 $2^{-200}\times 2^{192}\ll 1[2^{-100}\times 2^{96}\ll 1]$이다. 따라서 25개의 원소만으로 충분하다.

② 각 평문 집합은 2번째 라운드 입력 값을 나타내며 각 평문 집합의 5번째 라운드의 출력값을 얻는다.

③ 각 암호문 집합에 대해 $T^s[b]=(f_0^s[b]\wedge m_2)\oplus(f_1^s[b]\wedge m_1)\oplus(f_2^s[b]\wedge m_0)\oplus(f_3^s[b]\wedge m_3)$를 계산하여 저장한다 [mCrypton의 경우 $T^s[b]=(f_3^s(b)\wedge m_0)\oplus(f_2^s(b)\wedge m_1)\oplus(f_1^s(b)\wedge m_2)\oplus(f_0^s(b)\wedge m_3)$].

④ 임의의 $b\in\Lambda$에 대해서 $(T^{s'}[b]=T^{s''}[b])_{b\in\Lambda}$이 성립하는 s', s''를 50% 이상의 확률로 찾을 수 있다.

3.3 7-라운드 Crypton, mCrypton의 충돌 공격

본 절에서는 앞 절에서 구성한 4-라운드 충돌 특성을 이용하여 7-라운드 충돌 공격을 소개한다. [그림 4]는 7-라운드 Crypton과 mCrypton의 상태 값을 나타낸다. B, C, D, E, F는 앞 절과 동일하며, A는 첫 번째 라운드 입력 평문 상태 값, F는 5번째 라운드, G는 6번째 라운드 그리고 H는 7번째 라운드 출력 상태 값을 나타낸다.

다음은 각 알고리즘의 7번째 라운드 출력값(H)을 이용해 5번째 라운드 출력값(F)을 구하는 식이다. 각 알고리즘의 H는 위의 각각의 과정들을 통해 F를 복구할 수 있다.

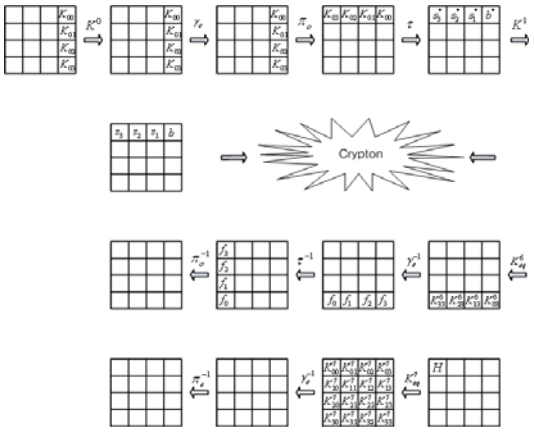
- Crypton (5−7 라운드):

$$H=\tau\circ\pi_o\circ\tau\circ\sigma_{K^7}\circ\tau\circ\pi_o\circ\gamma_o\circ\sigma_{K^6}\circ\tau\circ\pi_e\circ\gamma_e(F)$$
$$=\tau\circ\sigma_{\pi_o^{-1}\circ\tau^{-1}(K^7)}\circ\gamma_o\circ\tau\circ\pi_e\circ\sigma_{\pi_e^{-1}\circ\tau^{-1}(K^6)}\circ\gamma_e(F)$$
$$=\sigma_{\tau\circ\pi_o^{-1}\circ\tau^{-1}(K^7)}\circ\tau\circ\gamma_o\circ\tau\circ\pi_e\circ\sigma_{\pi_e^{-1}\circ\tau^{-1}(K^6)}\circ\gamma_e(F)$$
$$=\sigma_{\tau\circ\pi_o^{-1}\circ\tau^{-1}(K^7)}\circ\gamma_o\circ\pi_e\circ\sigma_{\pi_e^{-1}\circ\tau^{-1}(K^6)}\circ\gamma_e(F)$$

- mCrypton (5−7 라운드):

$$H=\tau\circ\pi\circ\tau\circ\sigma_{K^7}\circ\tau\circ\pi\circ\gamma_o\circ\sigma_{K^6}\circ\tau\circ\pi\circ\gamma_e(F)$$
$$=\tau\circ\sigma_{\pi^{-1}\circ\tau^{-1}(K^7)}\circ\gamma_o\circ\tau\circ\pi\circ\sigma_{\pi^{-1}\circ\tau^{-1}(K^6)}\circ\gamma_e(F)$$
$$=\sigma_{\tau\circ\pi^{-1}\circ\tau^{-1}(K^7)}\circ\tau\circ\gamma_o\circ\tau\circ\pi\circ\sigma_{\pi^{-1}\circ\tau^{-1}(K^6)}\circ\gamma_e(F)$$
$$=\sigma_{\tau\circ\pi^{-1}\circ\tau^{-1}(K^7)}\circ\gamma_o\circ\pi\circ\sigma_{\pi^{-1}\circ\tau^{-1}(K^6)}\circ\gamma_e(F)$$

[그림 5]는 [알고리즘 2]를 설명하고 있으며, 다음은 복호화 할 때 쓰이는 변환식이다.



[그림 5] Crypton, mCrytpon의 7-라운드 충돌 공격

Crypton: $K_{eq}^{6} = \pi_e^{-1} \circ \tau^{-1}(K^6)$,
$K_{eq}^{7} = \tau \circ \pi_e^{-1} \circ \tau^{-1}(K^7)$
mCrypton: $K_{eq}^{6} = \pi^{-1} \circ \tau^{-1}(K^6)$,
$K_{eq}^{7} = \tau \circ \pi^{-1} \circ \tau^{-1}(K^7)$).

- [알고리즘 2] 7-라운드 충돌 공격 알고리즘(단, 대괄호 안의 값은 mCrypton의 경우이다.)
  ① (a_0, a_1, a_2, a_3)를 제외한 모든 바이트가 상수인 2^{32}개[2^{16}개]의 text로 구성된 평문 집합 $A = \{A\}^{i}_{(0 \le i \le 2^{32}-1)}$을 획득한다. 평문 A^i에 대응하는 암호문은 H^i이며 i는 32 비트[16 비트] 워드 $a_0 \| a_1 \| a_2 \| a_3$에 대응된다.
  ② 40 비트 키[20 비트 키] $(K_{eq}^{6})_{j3}$, $(K_{eq}^{7})_{j0}$, $(K_{eq}^{7})_{j1}$, $(K_{eq}^{7})_{j2}$, $(K_{eq}^{7})_{j3}$을 추측하여 2^{32}개[2^{16}개]의 암호문 H^i를 f_j 위치까지 복호화한다 ($j = 0,1,2,3$, $4 \times 2^{40} \times 2^{32} (= 2^{74})$[$4 \times 2^{20} \times 2^{16} (= 2^{38})$]) 각각에 대해 실행). 복호화된 값을 추측한 네 개의 40 비트[20 비트]의 키, 인덱스 i, j에 따라 테이블에 저장한다. 위의 과정을 모든 네 개의 40 비트[20 비트]의 키에 대해 실행하므로 테이블의 크기는 $4 \times 2^{40} \times 2^{32} (= 2^{74})$[$2^{38} (= 4 \times 2^{16} \times 2^{20})$]메모리 바이트를 요구한다.
  ③ 첫 번째 라운드 π단계 후의 상태 $(b^*, s_1^*, s_2^*, s_3^*)$ 위치에서 25개의 원소로 구성된 2^{16}개[2^8개]의 집합을 선택한다. 각각의 집합은 동일한 상수 $s^* = (s_1^*, s_2^*, s_3^*)$에 대해 서로 다른 25개의 값 b^*를 사용한다. 집합과 집합 사이에 사용된 $s^* = (s_1^*, s_2^*, s_3^*)$는 모두 서로 다르며, 25개의 b^*는 서로 동일하도록 선택한다.
  ④ 32 비트[16] 비트 키 $(K^0)_{00}$, $(K^0)_{01}$, $(K^0)_{02}$, $(K^0)_{03}$을 추측하여 과정 ③ 에서 선택한 2^{16}개[2^8개]의 집합을 (a_1, a_2, a_3, a_4) 위치까지 복호화한다.
    (ㄱ) 복호화한 2^{16}개(2^8개)의 집합 중 서로 다른 두 집합과 앞서 추측한 $40 \times 4 = 160$ [$20 \times 4 = 80$] 비트 키 각각에 대해 과정 ②에서 저장한 테이블을 이용하여, 식 (1)[식 (2)]을 체크한다. 만약 25개의 원소 각각에 대해서 식 (1)[식 (2)]을 모두 만족한다면 192 비트[96 비트] 키 $(K^0)_{00}$, $(K^0)_{01}$, $(K^0)_{02}$, $(K^0)_{03}$, $(K_{eq}^{6})_{j3}$, $(K_{eq}^{7})_{j0}$, $(K_{eq}^{7})_{j1}$, $(K_{eq}^{7})_{j2}$, $(K_{eq}^{7})_{j3}$을 올바른 키로 출력한다. 모든 서로 다른 두 집합과 모든 160 비트 키[80 비트]에 대해서 위의 테스트를 통과하지 못한다면, 다시 과정 ④로 돌아간다.

본 7-라운드 공격은 [알고리즘 2]의 과정 ①에 의해, Crypton은 2^{32}개의 선택 평문을 요구하며 mCrypton은 2^{16}개의 선택 평문을 요구한다. 시간 복잡도와 메모리 복잡도는 [알고리즘 2]의 과정 ②의 테이블에 의해 좌우된다. 따라서 Crypton의 경우, 테이블에 저장하면 $2^{69.2} (= 2^{32} \times 2^{40} \times \frac{1}{7} \times \frac{1}{4} \times 4)$의 시간 복잡도와 $2^{74} (= 4 \times 2^{32} \times 2^{40})$의 메모리 바이트가 요구되며 mCrypton의 경우는 $2^{33.2} (= 2^{16} \times 2^{20} \times \frac{1}{7} \times \frac{1}{4} \times 4)$의 시간 복잡도와 $2^{38} (= 4 \times 2^{16} \times 2^{20})$의 메모리 바이트가 요구된다.

Ⅳ. 5-라운드 특성 및 8-라운드 공격

본 장에서는 각 알고리즘의 5-라운드 충돌 특성을 이용한 8-라운드 충돌 공격을 소개한다. 8-라운드 충돌 공격은 7-라운드 충돌 공격 방법과 유사하며 7-라운드 충돌 공격보다 시간 복잡도는 많이 요구되나 복구하는 부분 키의 비트 수는 크게 향상된다. 앞서 살펴본 것과 같이 mCrypton 공격은 Crypton의 공격과 유사하므로, 공격 과정은 생략한다.

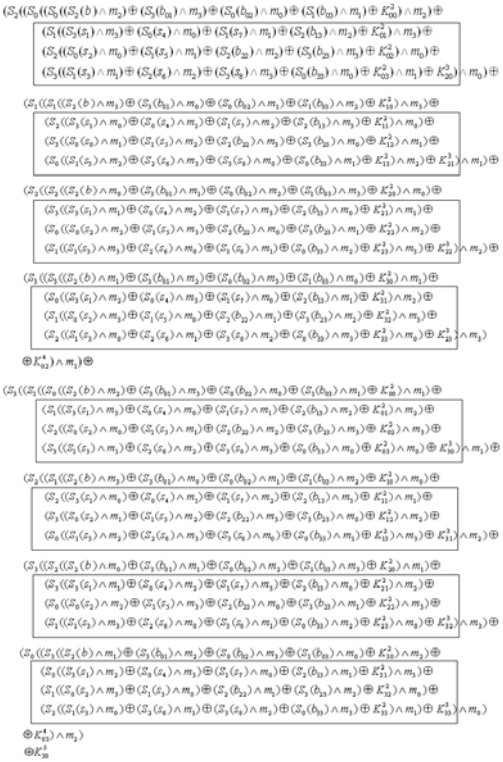
4.1 5-라운드 Crypton, mCrypton의 충돌 특성

Crypton 5-라운드 충돌 특성은 4-라운드 충돌 특성과 표기 방식이 동일하나, 7-라운드 공격에서 2번째 라운드의 $s (= s_1, s_2, s_3)$값을 8-라운드 공격에서는 8개의 바이트를 사용하며, 나머지 7개의 바이트들은 일정한 상수로 놓는다 ($s = (s_1 = b_{10},\ s_2 = b_{20},\ s_3 = b_{30},\ s_4 = b_{11},\ s_5 = b_{21},\ s_6 = b_{31},\ s_7 = b_{12},\ s_8 = b_{22})$). 아래 $f (= f_{30})$값은 Crypton의 6-라운드 입력 상태 값을 나타낸다. 이 식을 통해 f 값은 스물 한 개의 키-의존 상수, 열여섯 개의 상수 s와 키-의존 상수에 의해 결정됨을 알 수 있다. 즉, $f^s[b]$는 $s = (s_1, s_2, \cdots, s_8)$에 의존하고 키에 의존하는 열여섯 개의 상수와 $s = (s_1, s_2, \cdots, s_8)$에 독립이고 키에 의존적인 열여섯 개의 상수 그리고 키만 존재하는 다섯 개의 상수로

구성된다는 것을 알 수 있다. 만약 서로 다른 상수 s'와 s''에 대해서 열여섯 개의 키와 상수 $s=(s_1,s_2,\cdots,s_8)$에 의존하는 128 비트(16개의 네모상자) 값이 동일하다면 전체적으로 f값이 동일하게 된다. 따라서 2^{64}개의 서로 다른 s를 갖는다면 50% 이상의 확률로 s의 충돌 쌍을 찾을 수 있으며 아래의 식 (3)을 만족한다.

$$(g_0^{s'}[b]\wedge m_0)\oplus(g_1^{s'}[b]\wedge m_3)\oplus(g_2^{s'}[b]\wedge m_2)\oplus(g_3^{s'}[b]\wedge m_1)$$
$$=(g_0^{s''}[b]\wedge m_0)\oplus(g_1^{s''}[b]\wedge m_3)\oplus(g_2^{s''}[b]\wedge m_2)$$
$$\oplus(g_3^{s''}[b]\wedge m_1) \qquad (3)$$

- [알고리즘 3] 5-라운드 충돌 특성 찾기(단, 대괄호 안은 mCrypton의 경우이다).
  ① 임의의 33개의 b를 갖는 집합 $\Lambda\subset\{0,1,\cdots,255\}$와 $2^{64}[2^{32}]$개의 서로 다른 상수 $s=(s_1,s_2,\cdots,s_8)$를 갖는 33개로 이루어진 $2^{96}[2^{48}]$개의 평문 집합을 선택한다. 8-라운드 충돌 공격에 사용되는 Λ는 33개의 원소만으로 충분하다. 8-라운드 충돌 공격에



사용되는 집합 Λ는 전체 추측하는 키 공간은 $96+40\times4=256$[128]비트이다. 한 쌍의 s값이 충돌할 확률은 2^{-8}[2^{-4}]이며 33개의 b값에 대해서 $(2^{-8})^{33}=2^{-264}$[2^{-132}]가 성립하므로 $2^{-264}\times2^{256}\ll1$[$2^{-132}\times2^{128}\ll1$]이다. 따라서 33개의 원소만으로 충분하다.

② 각 평문 집합은 2번째 라운드의 입력 값을 나타내며 각 평문 집합의 6번째 라운드의 출력 값을 얻는다.

③ 각 암호문 집합에 대해 $T^s[b]=(g_0^s[b]\wedge m_0)\oplus(g_1^s[b]\wedge m_3)\oplus(g_2^s[b]\wedge m_2)\oplus(g_3^s[b]\wedge m_1)$를 계산하여 저장한다. [mCrypton의 경우 $T^s[b]=(g_3^s(b)\wedge m_0)\oplus(g_2^s(b)\wedge m_1)\oplus(g_1^s(b)\wedge m_2)\oplus(g_0^s(b)\wedge m_3)$].

④ 임의의 $b\in\Lambda$에 대해서 $(T^{s'}[b]=T^{s''}[b])_{b\in\Lambda}$이 성립하는 s', s''를 50% 이상의 확률로 찾을 수 있다.

4.2 8-라운드 Crypton, mCrypton의 충돌 공격

7-라운드 충돌 공격과 유사한 방법으로 앞에서 구성한 5-라운드 충돌 특성을 이용하여 8-라운드 충돌 공격

을 소개한다. 7-라운드 충돌 공격과 같이 8-라운드 출력 값 I를 이용하여 다음 식을 구할 수 있다(다음 식은 Crypton의 경우이며 mCrypton의 경우는 Crypton과 모든 식이 유사하므로 생략한다).

$$I = \tau \circ \pi_e \circ \tau \circ \sigma_{K^8} \circ \tau \circ \pi_e \circ \gamma_e \circ \sigma_{K^7} \circ \tau \circ \pi_o \circ \gamma_o(G)$$
$$= \tau \circ \sigma_{\pi_e^{-1} \circ \tau^{-1}(K^8)} \circ \gamma_e \circ \tau \circ \pi_o \circ \sigma_{\pi_o^{-1} \circ \tau^{-1}(K^7)} \circ \gamma_o(G)$$
$$= \sigma_{\tau \circ \pi_e^{-1} \circ \tau^{-1}(K^8)} \circ \tau \circ \gamma_e \circ \tau \circ \pi_o \circ \sigma_{\pi_o^{-1} \circ \tau^{-1}(K^7)} \circ \gamma_o(G)$$
$$= \sigma_{\tau \circ \pi_e^{-1} \circ \tau^{-1}(K^8)} \circ \gamma_e \circ \pi_o \circ \sigma_{\pi_o^{-1} \circ \tau^{-1}(K^7)} \circ \gamma_o(G).$$

따라서 I는 과정 $\gamma_e \circ \sigma_{\pi_o^{-1} \circ \tau^{-1}(K^7)} \circ \pi_o \circ \gamma_o \circ \sigma_{\tau \circ \pi_e^{-1} \circ \tau^{-1}(K^8)}(I) = G$을 통해 G를 복구할 수 있다.

- [알고리즘 4] 8-라운드 충돌 공격 알고리즘(단, 대괄호 안은 mCrypton의 경우이다).
  ① $(a_0, a_1, a_2, \cdots, a_{11})$를 제외한 모든 바이트가 상수인 $2^{96}[2^{48}]$개의 text로 구성된 평문 집합 $A = \{A^i\}_{0 \le i \le 2^{96}-1}$을 획득한다. 평문 A^i에 대응하는 암호문은 I^i이며 i는 96 비트[48 비트] 워드 $a_0 \| a_1 \| a_2 \| \cdots \| a_{10} \| a_{11}$에 대응된다 ($a_0 = a_{00}$, $a_1 = a_{01}$, $a_2 = a_{02}$, $a_3 = a_{03}$, $a_4 = a_{01}$, $a_5 = a_{11}$, $a_6 = a_{12}$, $a_7 = a_{13}$, $a_8 = a_{20}$, $a_9 = a_{21}$, $a_{10} = a_{22}$, $a_{11} = a_{23}$).
  ② 40 비트[20 비트] 키 $(K_{eq}^7)_{j3}$, $(K_{eq}^8)_{j0}$, $(K_{eq}^8)_{j1}$, $(K_{eq}^8)_{j2}$, $(K_{eq}^8)_{j3}$을 추측하여 $2^{96}[2^{48}]$개의 암호문 I^i를 g_j위치까지 복호화한다 ($j = 0,1,2,3$ 각각에 대해 실행). 복호화된 값을 추측한 네 개의 40 비트[20 비트]의 키, 인덱스 i, j 에 따라 테이블에 저장한다. 위의 과정을 모든 네 개의 40 비트[20 비트]의 키에 대해 실행하므로 테이블의 크기는 $4 \times 2^{96} \times 2^{40} \times \frac{1}{8} \times \frac{1}{4} (= 2^{133})$ [$4 \times 2^{48} \times 2^{20} \times \frac{1}{8} \times \frac{1}{4} (= 2^{65})$]이다.
  ③ 첫 번째 라운드 π_o 후의 상태 $(b^*, s_1^*, s_2^*, \cdots, s_8^*)$ 위치에서 33개의 원소로 구성된 $2^{64}[2^{32}]$개의 집합을 선택한다. 각각의 집합은 동일한 상수 $s^* = (s_1^*, s_2^*, \cdots, s_8^*)$에 대해 서로 다른 33개의 값 b^*를 사용한다. 집합과 집합 사이에 사용된 $s^* = (s_1^*, s_2^*, \cdots, s_8^*)$는 모두 서로 다르며, 33개의 b^*는 서로 동일하도록 선택한다.
  ④ 96 비트[48 비트] 키 $(K^0)_{00}$,$(K^0)_{01}$,$(K^0)_{02}$,$(K^0)_{03}$, $(K^0)_{10}$, $(K^0)_{11}$, $(K^0)_{12}$, $(K^0)_{13}$, $(K^0)_{21}$, $(K^0)_{22}$, $(K^0)_{20}$, $(K^0)_{23}$을 추측하여 과정 ③에서 선택한 $2^{64}[2^{32}]$개의 집합을 $(a_0, a_1, \cdots, a_{11})$ 위치까지 복호화한다.
  (ㄱ) 복호화한 $2^{64}[2^{32}]$개의 집합 중 서로 다른 두 집합과 앞서 추측한 $40 \times 4 = 160$ 비트 [$20 \times 4 = 80$ 비트] 키 각각에 대해 과정 ②에서 저장한 테이블을 이용하여, 식 (3)을 체크한다. 만약 33개의 원소 각각에 대해서 식 (3)을 모두 만족한다면 256비트 키 $(K^0)_{00}$, $(K^0)_{01}$, $(K^0)_{02}$, $(K^0)_{03}$, $(K^0)_{10}$, $(K^0)_{11}$, $(K^0)_{12}$, $(K^0)_{13}$, $(K^0)_{20}$, $(K^0)_{21}$, $(K^0)_{22}$, $(K^0)_{23}$, $(K_{eq}^7)_{j3}$, $(K_{eq}^8)_{j0}$,$(K_{eq}^8)_{j1}$,$(K_{eq}^8)_{j2}$,$(K_{eq}^8)_{j3}$을 올바른 키로 출력한다. 모든 서로 다른 두 집합과 모든 160 비트[80 비트] 키에 대해서 위의 테스트를 통과하지 못하면, 과정 ④로 돌아간다.

Crypton 8-라운드 공격에서 [알고리즘 4]의 과정 ①에 의해 2^{96}의 선택 평문이 요구되며 시간 복잡도와 메모리 복잡도는 Crypton의 경우 $2^{161.6} (= 2^{96} \times 2^{64} \times 33 \times \frac{1}{8} \times \frac{3}{4})$의 시간 복잡도와 $2^{133} (= 4 \times 2^{96} \times 2^{40} \times \frac{1}{8} \times \frac{1}{4})$의 메모리 바이트가 요구되며 mCrypton의 경우 2^{48}의 선택 평문이 요구되며 시간 복잡도는 $2^{81.6} (= 2^{48} \times 2^{32} \times 32 \times \frac{1}{8} \times \frac{3}{4})$ 그리고 메모리 복잡도는 $2^{65} (= 4 \times 2^{48} \times 2^{20} \times \frac{1}{8} \times \frac{1}{4})$이다.

Ⅴ. 결 론

본 논문은 H. Gilbert와 M. Minier에 의해 제시된 충돌 공격[5]을 Crypton, mCrypton에 적용하여 7/8-라운드 Crypton과 7/8-라운드 mCrypton의 부분키를 복구할 수 있음을 보였다. 본 논문에서 제안한 공격은 블록 암호의 처음 몇 라운드의 약한 확산 성질과 생일 공격을 이용한 충돌 성질을 이용하여 공격하므로 일정한 확산 성질을 갖는 일반적인 SPN 구조인AES, Anubis, Square[8]에 적용 가능하다. 그러나 비트 단위의 확산 함수를 사용하는 KHAZAD[14]나 강한 확산 함수를 사

용하는 ARIA[3], Q[11] 등의 블록 암호나, 한 라운드에 두 번의 키를 사용하는 Grand Cru[6]와 같은 블록 암호는 공격이 불가능하다.

참고문헌

[1] C. D'Halluin, G. Bijnens, V. Rijmen, and B. Preneel, “Attack on Six Rounds of Crypton,” Fast Sofrware Encryption Workshop 1999, LNCS 1636, pp. 46-59, 1999.

[2] C. Lim, “CRYPTON: A New 128-bit Block Cipher,” AES Proposal, Aug. 1998.

[3] D. Kwon, J. Kim, S.Park, S. Sung, Y. Sohn, J. Song, Y. Yeom, E. Yoon, S. Lee, J. Lee, S. Chee, D. Han, and J. Hong, “New Block Cipher: ARIA,” International Conference on Information Security and Cryptology 2003, LNCS 2971, pp. 443-456, 2003.

[4] H. Seki and T. Kaneko, “Cryptanalysis of Five Rounds of CRYPTON Using Impossible Differential,” ASIACRYPT'99, LNCS 1716, pp. 45-51, 1999.

[5] H. Gilbert and M. Minier, “A collision attack on 7 rounds of Rijndael,” Proceedings of the 3rd Advanced Encryption Standard Candidate Conference, pp 230-241, Apr. 2000.

[6] J. Borst, “The block cipher Grand Cru,” Primitive submitted to NESSIE, Nov. 2000.

[7] J. Cheon, M. Kim, K. Kim, and J. Lee, “Improved Impossible Differential Cryptanalysis of Rijndeal and Crypton,” International Conference on Information Security and Cryptology 2001, LNCS 2288, pp. 39-49, 2001.

[8] J. Daemen, L. Kundsen, and V. Rijmen, “The block cipher Square,” Fast Sofrware Encryption Workshop 1997, LNCS 1267, pp. 149-165, 1997.

[9] J. Daemen and V. Rijmen, “AES Proposal: Rijndael,” Proceedings of the 1st Advanced Encryption Standard Candidate Conference, Aug. 1998.

[10] J. Kim, S. Hong, S. Lee, J. Song, and H. Yang, “Truncated Differential Attacks on 8-Round CRYPTON,” International Conference on Information Security and Cryptology 2003, LNCS 2971, pp. 446-456, 2004.

[11] L. McBride, “Q: A Proposal for NESSIE V2.00,” Submission of NESSIE Project, Nov. 2000.

[12] M. Minier and H. Gilbert, “Stochastic Cryptanalysis of Crypton,” Fast Sofrware Encryption Workshop 2000, LNCS 1978, pp. 121-133, 2000.

[13] C. Lim and T. Korkishko, “mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors,” International Workshop on Information Security Applications 2005, LNCS 3786, pp. 243-258, 2006.

[14] P. Baretto and V. Rijmen, “The KHAZAD Legacy-Level Block Cipher,” Submission of NESSIE Project, 2002.

<著者紹介>

김 태 웅 (Taewoong Kim) 학생회원
2002년 2월: 강남대학교 수학과, 경영학 학사
2003년 3월~현재: 고려대학교 정보경영공학전문대학원 석사과정
<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 정보보호기술연구센터 연구교수
<관심분야> 대칭키 암호의 분석 및 설계

정 기 태 (Kitae Jeong) 학생회원
2004년 2월: 고려대학교 수학과 학사
2006년 2월: 고려대학교 정보보호대학원 석사
2006년 3월~현재 : 고래대학교 정보보호대학원 박사과정
<관심분야> 블록 암호, 스트림 암호 및 해쉬 함수의 분석 및 설계

성 재 철 (GJaechul 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

이 상 진 (Sangjin Lee) 종신회원
1987년 2월: 고려대학교 수학과 학사
1989년 2월: 고려대학교 수학과 석사
1994년 2월: 고려대학교 수학과 박사
1989년 2월~1999년 2월: 한국전자통신연구원 선임 연구원
1999년 2월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현 재: 고려대학교 정보보호대학원 부교수
<관심분야> 대칭키 암호의 분석 및 설계. 정보은닉이론, 컴퓨터 포렌식