情報保護學會論文誌
第19卷 第1號, 2009. 2

블록 암호 KeeLoq에 대한 취약키 공간에 관한 연구*

이 유 섭,† 김 종 성, 홍 석 희‡

고려대학교 정보보호기술 연구센터

Study on Weak-Key Classes for KeeLoq*

Yuseop Lee,† Jongsung Kim, Seokhie Hong‡

Center for Information Security Technologies, Korea University

요 약

KeeLoq는 32-비트 블록과 64-비트 마스터 키를 사용하는 블록 암호이다. 또한, 매우 단순한 구조로 무선 응용분야에 적합하게 설계되었다. 이런 특성으로 인해 크라이슬러, GM, 혼다, 도요타 등 여러 차량 제조 회사에서 차량을 보호하기 위한 무선 키리스 엔트리 시스템(keyless entry system)이나 차량 경보 시스템 등에 사용하고 있다. 본 논문에서는 $2^1 \sim 2^{32}$개의 키를 가지는 여러 취약키 공간을 소개하고, 마스터 키가 취약키 공간에 속한 경우, 슬라이드 공격 기법에 의해 마스터 키가 쉽게 노출됨을 보인다.

ABSTRACT

KeeLoq is a very light block cipher with a 32-bit block and a 64-bit key. It is suitable for the wireless applications, and thus multiple automotive OEMs as Chrysler, GM, Honda, Toyota have used remote keyless entry systems and alarm systems in order to protect the their cars. In this paper, we introduce various weak-key classes that include $2^1 \sim 2^{32}$ keys and exploit the slide attack to propose key-recovery attacks under these weak-key classes.

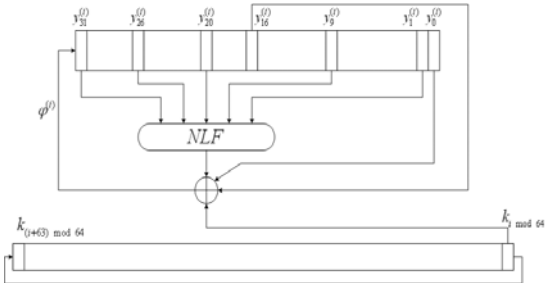
Keywords : BLock cipher, KeeLoq, weak-key attack, Cryptanalysis

Ⅰ. 서 론

1980년대에 발표된 Microchip Technology 사의 KeeLoq 기술은 KeeLoq 블록 암호와 여러 인증 프로토콜로 구성된다[1]. KeeLoq 블록 암호는 매우 단순한 구조로 설계되어 적은 전력과 초경량 하드웨어로 구현 가능하여 무선 환경에 사용하기 적합하다. 실제로 크라이슬러, GM, 혼다, 도요타 등 여러 차량 제조 회사에서 차량을 보호하기 위한 무선 키리스 엔트리 시스템(Keyless entry system)이나 차량 경보 시스템 등에 사용하고 있다.

KeeLoq은 일반적인 블록 암호의 구조적인 특징에서 벗어나 스트림 암호의 형태를 가지고 있기 때문에 현재까지는 블로 암호의 대표적인 분석 방법인 차분분석[2]과 선형분석[3,4]에 대한 취약점은 발견되지 않았다. 2007년 Bogdanov은 KeeLoq 블록 암호를 최초로 분석하였다[5]. [5]에서 소개하는 공격은 코드북이 주어지면, 슬라이드 공격 기법과 부울 함수의 선형 근사식을 이용

접수일(2008년 8월 4일), 수정일(1차: 2008년 11월 3일, 2차: 2008년 11월 23일), 게재확정일(2008년 12월 12일)
* 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임. (No. R01-2008-000-11879-0)
† 주저자, yusubi@cist.korea.ac.kr
‡ 교신저자, hsh@cist.korea.ac.kr



[그림 1] KeeLoq의 i 라운드 함수

하여 2^{52}의 KeeLoq 암호화 연산으로 키를 복구한다. 이후 Bogdanov는 시간 복잡도를 $2^{50.6}$ KeeLoq 암호화 연산으로 공격을 개선하였다[6]. 이 후, Courtois 등은 코드북이 주어진 경우 고정점을 이용하여 2^{27} KeeLoq 암호화 연산으로 키를 복구하는 공격을 제안하였다[7,8]. 또한, 2^{16}개의 기지 평문을 이용하여 슬라이드 공격과 대수적 공격 방법[9,10]을 통하여 $2^{51.4}$ KeeLoq 암호화 연산으로 키를 복구할 수 있음을 보였다. 최근 Sebastiaan Indesteege 등이 [5]과 [8]에서 제안한 공격 방법과 중간 일치 공격 (meet-in- the-middle attack)을 접목하여 2^{16}개의 기지 평문과 $2^{44.5}$ KeeLoq 암호화 연산으로 키를 복구하는 공격을 제안하였다[11].

본 논문에서는 최초로 KeeLoq에 대한 취약키 공간을 소개하고, 이러한 취약키에 대한 키 복구 공격을 제안한다. 취약키 공간은 임의의 마스터 키가 그 집합에 포함될 경우, 상대적으로 적은 연산만으로 마스터 키를 복구할 수 있는 마스터 키의 집합을 의미한다. 본 논문에서 소개하는 KeeLoq 블록 암호에 대한 취약키 공간은 2^{32}개의 마스터 키를 가지는 취약키 공간으로 32 비트가 대칭되는 형태의 마스터 키로 이루어진다. 이러한 취약키 공간에 속하는 마스터 키에 대하여 $2^{16}+1$개의 선택 평문과 $2^{17.08}$ KeeLoq 암호화 연산으로 키를 복구한다(3.2 절). 또한, 이 이외에 $2^1 \sim 2^{16}$개의 취약키를 가지는 취약키 공간을 소개하고, $2^4 \sim 2^{16}$개의 마스터 키로 이루어진 취약키 공간은 전수조사보다 적은 복잡도로 키를 복구 할 수 있음을 보인다.(3.1 절).

본 논문은 다음과 같이 구성된다. 2장에서는 KeeLoq 블록 암호에 대한 설명과 공격에 사용되는 KeeLoq의 주요 특성에 대해 소개한다. 3장에서는 다양한 형태의 취약키 공간을 정의하고 이에 대한 공격 방법을 제안한다. 4장에서는 기존의 공격과 제안한 공격을 비교한다. 그리고 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 블록 암호 KeeLoq 및 주요 특성

2.1 블록 암호 KeeLoq

블록 암호 KeeLoq는 32-비트 블록과 64-비트 마스터 키를 사용하는 블록 암호로 NFSR(Nonlinear Feedback Shift Register)을 기반으로 설계되었다. 각 라운드는 마스터 키의 1 비트를 이용하고, 총 528 라운드 연산을 수행한다. i 라운드의 입력값은 $Y^{(i)}=(y_{31}^{(i)},\cdots,y_0^{(i)}) \in \{0,1\}^{32}$ $(0 \le i \le 527)$로 표시하고, 마스터 키 $K=(k_{63},\cdots,k_0) \in \{0,1\}^{64}$로 표시한다. 여기서, $y_{31}^{(i)}$은 최상위 비트이고, $y_0^{(i)}$은 최하위 비트이다. 즉, 0 라운드의 입력값인 평문은 $P=(y_{31}^{(0)},\cdots,y_0^{(0)})$이고, 528 라운드의 입력값(527라운드 출력값)인 암호문은 $C=(y_{31}^{(528)},\cdots,y_0^{(528)})$이다. 라운드 함수는 NFSR의 1회의 클럭과 1 비트 마스터키의 배타적 논리합으로 계산된다. [그림 1]은 i 라운드 함수를 나타내며 식 (1)과 같이 표현된다.

$$\begin{aligned} \phi^{(i+1)} &= NLF(y_{31}^{(i)}, y_{26}^{(i)}, y_{20}^{(i)}, y_9^{(i)}, y_1^{(i)}) \\ &\quad \oplus y_{16}^{(i)} \oplus y_0^{(i)} \oplus k_{(i) \bmod 64}. \\ y_{31}^{(i+1)} &= \phi^{(i+1)}, y_j^{(i+1)} = y_{j+1}^{(i)},\ 0 \le j \le 30. \end{aligned} \tag{1}$$

비선형 함수 NLF(Non-Linear Function)은 5-비트 입력을 받아 1-비트 출력을 하는 3차 부울 함수로 식 (2)와 같다.

$$\begin{aligned} &NLF(x_4,x_3,x_2,x_1,x_0) = \\ &x_4x_3x_2 \oplus x_4x_3x_1 \oplus x_4x_2x_0 \oplus x_4x_1x_0 \oplus x_4x_2 \oplus \\ &x_4x_0 \oplus x_3x_2 \oplus x_3x_0 \oplus x_2x_1 \oplus x_1x_0 \oplus x_1 \oplus x_0. \end{aligned} \tag{2}$$

$k \le j$일 때, i 라운드에서 레지스터의 j번째 비트는 $i+k$ 라운드에서 $j-k$번째 비트가 된다 $(y_j^{(i)} = y_{j-k}^{(i+k)})$. 그러므로 식 (1)을 식 (4)의 형태로 변형할 수 있다.

$$\begin{aligned} &k_{(i) \bmod 64} = \\ &NLF(y_0^{(i+31)}, y_0^{(i+26)}, y_0^{(i+20)}, y_0^{(i+9)}, y_0^{(i+1)}) \\ &\oplus y_0^{(i+16)} \oplus y_0^{(i)} \oplus y_0^{(i+32)}. \end{aligned} \tag{3}$$

$l(<32)$ 라운드 사이의 입력값과 출력값을 아는 경우, 식 (3)를 이용하면 사용된 키를 계산할 수 있다, 식 (1)과 식(3)에 필요한 연산량이 동일하므로 l 라운드 키를

계산하는데 필요한 연산량은 l 라운드 함수의 연산량과 동일하다[5]. 예를 들어, 0 라운드 입력값과 15 라운드 출력값(16 라운드 입력값)은 각각 $(y_0^{(0)}, y_1^{(0)}, \cdots, y_{31}^{(0)})$과 $(y_0^{(17)}, y_1^{(17)}, \cdots, y_{31}^{(17)})$으로 표시한다. $y_j^{(i)} = y_{j-k}^{(i+k)}$이므로 $(y_0^{(0)}, y_0^{(1)}, \cdots, y_0^{(48)})$를 계산 가능하여 식 (3)에 대입하면 $k_0, k_1, \cdots, k_{15}$를 계산 가능하다. 그리고 이 과정에 필요한 연산량은 16 라운드 연산량과 동일하다.

2.2 KeeLoq 블록 암호에 대한 슬라이드 공격

슬라이드 공격은 1999년 Biryukov 와 Wagner에 의해 제안되었다[12]. 슬라이드 공격은 키에 의해 동일한 치환이 반복적으로 나타나는 블록 암호에 적용된다. 즉, 라운드 키가 동일하게 반복된다면, 전체 r 라운드 암호는 다음과 같이 동일한 치환이 r번 반복되는 형태로 표현 된다.

$$C = \underbrace{F(F(\cdots(F(P))))}_{r} = F^r(P) \tag{4}$$

여기서, 치환과 라운드 함수가 반드시 일치하지 않아도 된다. 즉, F는 일정한 여러 라운드의 합성이어도 무관하다. 이러한 슬라이드 공격의 목적은 전체 라운드의 수에 상관없이 F의 입력값과 출력값을 이용하여 분석을 가능하게 하는 것이다. 이를 위해 식 (6)을 만족하는 F의 입력값과 출력값이 되는 평문 쌍 (P_1, P_2)를 슬라이드 쌍으로 정의한다.

$$P_2 = F(P_1) \tag{5}$$

슬라이드 쌍에 대응하는 암호문쌍 (C_1, C_2)도 동일한 특성을 만족한다. 즉, $C_2 = F(C_1)$이다. 이를 이용하면 슬라이드 쌍을 반복적으로 암호화하면 공격에 필요한 많은 슬라이드 쌍을 얻을 수 있다[13,14]. 이러한 슬라이드 쌍은 모두 F의 입력값과 출력값이 된다. 그러므로 슬라이드 쌍을 이용하면 전체 라운드의 수에 상관없이 F에 대한 분석으로 공격을 가능하게 한다.

KeeLoq 블록 암호는 각 라운드에서 64 비트 마스터 키 중 1 비트가 사용되는 528 라운드로 구성된다. 그러므로 64 라운드가 지나면 동일한 키가 다시 반복되어 사용된다. 그러므로 F를 64 라운드 함수에 해당하는 치환에 대응시키면 512 라운드는 F를 8번 반복하여 적용시킨 형태가 되어 슬라이드 공격을 적용할 수 있다. 하지만 전체 라운드 수가 $528(=64\times8+16)$ 라운드로 64의 배수가 아니기 때문에 일반적인 슬라이드 공격을 바로 적용하기 어렵다. 이를 해결하기 위해 마지막 16 라운드에 사용되는 16-비트 키를 추측하고, 남은 512 라운드에 대해서 슬라이드 공격을 적용하는 공격 방법이 소개되었다[5,7,8,11]. [11]에서는 중간 일치 공격을 통하여 슬라이드 쌍이 될 수 없는 평문쌍을 필터링한다. 이 공격은 2^{16}개의 기지 평문과 슬라이드 쌍을 필터링하기 위해 $2MB$의 테이블을 구성하고, $2^{44.5}$의 KeeLoq 암호화 연산으로 키를 복구한다. 2^{16}개의 평문을 이용하면 2^{32}개의 평문쌍을 조합할 수 있다. 이렇게 조합된 평문쌍 중 1개 이상의 슬라이드 평문쌍이 존재하는 경우에, 이 공격이 성공한다. 임의이 평문쌍이 될 확률은 2^{-32}이므로, 2^{32}의 평문쌍이 모두 슬라이드 쌍이 아닐 확률인 $(1-2^{-32})^{2^{32}}$ $(\approx 1/e)$이다. 그러므로 이 공격의 성공 확률은 약 $63(\approx 1-1/e)\%$이다.

2.3 KeeLoq 블록 암호의 취약키 특성

블록 암호의 취약키[15]는 마스터 키 또는 마스터 키로부터 생성되는 라운드 키가 특정 조건을 만족할 경우 알고리즘 분석에 유용한 성질을 나타내는 것을 의미한다. 대부분의 암호 알고리즘은 전체 키 공간에 비해 상당히 작은 부분의 취약키 공간을 갖고 있으나 특정한 경우엔 알고리즘 분석에 치명적인 약점이 되는 경우도 있다.

KeeLoq 블록 암호의 경우, 마스터 키의 형태가 $t \in \{1,2,4,8,16\}$ 비트씩 반복되면 t 라운드 함수를 치환 F에 대응시키면 마지막 16 라운드에 사용되는 키에 대한 추측 없이 슬라이드 공격이 적용 가능하다. 또한, 마스터 키의 형태가 $(k_{63}, \cdots, k_{32}) = (k_0, \cdots, k_{31})$인 경우 64 라운드에 해당하는 암호화 함수와 복호화 함수가 동일한 형태가 되어 이를 이용한 슬라이드 공격이 가능하다. 그러므로 이러한 형태의 마스터 키는 취약키 된다. 이러한 취약키에 대한 슬라이드 공격은 3 장에서 자세히 소개한다.

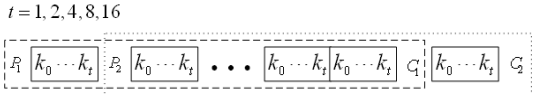
Ⅲ. 블록 암호 KeeLoq의 취약키 공간

본 장에서는 KeeLoq에 대한 다양한 취약키 공간을 소개하고, 마스터 키가 이런 취약키 공간에 속하는 경우 슬라이드 공격 기법을 이용한 키 복구 공격을 제안한다.

3.1 동일한 키 비트가 반복하는 형태의 취약키 공간

본 절에서는 동일한 $t \in \{1,2,4,8,16\}$ 비트가 반복되는 마스터 키에 대해 두 가지 슬라이드 공격 방법을 제안한다. 첫 번째 공격인 슬라이드 공격 1은 성공확률이 1인 공격 방법이고, 슬라이드 공격 2는 필요한 선택 평문의 양은 적지만 성공확률은 1보다 작은 공격 방법이다.

마스터 키가 t 비트씩 반복되는 형태인 경우, Keeloq 블록 암호는 t-라운드 함수를 $528/t$번 반복하는 형태로 표현된다. 라운드 함수는 최상위 1비트만 갱신하고 나머지 비트는 단순히 옆으로 이동하기 때문에 슬라이드 쌍 P_1의 하위 $(32-t)$ 비트와 P_2의 상위 $(32-t)$ 비트는 동일해야 한다(상위 l 비트는 레지스터의 왼쪽 l 비트를 의미함). 또한, 슬라이드 특성에 의해 대응하는 암호문 C_1과 C_2도 하위 $(32-t)$ 비트와 P_2의 상위 $(32-t)$ 비트가 동일해야 한다. 이 특성을 이용하면 슬라이드 쌍이 아닌 평문쌍을 필터링할 수 있다. [그림 2]는 마스터 키의 형태가 t 비트씩 반복되는 형태일 때, 슬라이드 쌍이 가지는 특성을 나타낸다.



[그림 2] 마스터 키가 t 비트씩 반복하는 형태인 경우, 슬라이드 쌍의 특성

3.1.1 슬라이드 공격 1 (2^t 취약키, $t \in \{1,2,4,8,16\}$)

$t \in \{1,2,4,8,16\}$인 경우, 임의의 평문 $P=(p_{31},\cdots,p_0)$에 대해 t 라운드 후에 가능한 내부 상태값인 하위 $(32-t)$ 비트가 $(p_{31},\cdots p_t)$로 고정된 2^t개의 평문 $P_i (0 \le i < 2^t)$를 선택하면 이러한 평문쌍의 집합에는 P의 슬라이드 쌍이 반드시 1 개 존재한다.

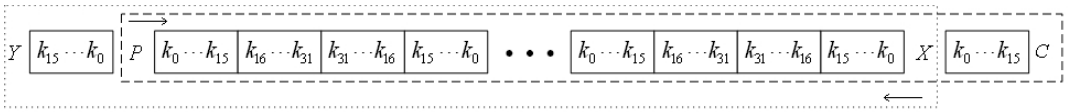
$$
\begin{aligned}
P_0 &= (0,\cdots,0,p_{31},\cdots p_t)\\
P_1 &= (0,\cdots,1,p_{31},\cdots p_t)\\
&\vdots\\
P_{2^t-1} &= (1,\cdots,1,p_{31},\cdots p_t)
\end{aligned}
$$

P의 슬라이드 쌍을 구하기 위해, 앞에서 설명한 P의 암호문인 C의 상위 $(32-t)$ 비트와 P의 슬라이드 평문쌍 P_j에 대응하는 암호문 C_j의 하위 $(32-t)$ 비트가 동일해야 한다는 특성을 이용한다. 이를 이용하여 슬라이드 쌍이 될 수 없는 P_j를 $(32-t)$ 비트의 비율만큼 필터링 한다. 이 필터링 과정에서 슬라이드 쌍은 확률 1로 통과할 것이고, 슬라이드 쌍이 아닌 경우는 랜덤하게 통과할 것이므로 평균 $1+2^{2t-32}$개의 P_j가 필터링은 통과한다. 다음으로 필터링을 통과한 $1+2^{2t-32}$개의 P_j와 P를 이용하여 t 비틀 키를 계산한다. 이 키가 C_j와 C를 이용하여 계산한 t 비트 키가 동일한지 검사한다. P_j와 P가 슬라이드 쌍이라면 올바른 키가 계산되어 확률 1로 이 검사를 통과하고, 슬라이드 쌍이 아니라면 이 검사를 통과할 확률은 2^{-t}이므로 틀린 키의 기댓값은 $2^{(2t-32)-t} \ll 1$이 되어 높은 확률로 올바른 키만이 남게 된다. 만약 두 개 이상의 키가 남는다면, 남은 키에 대한 평문, 암호문쌍을 이용하여 틀린 키를 걸러낸다.

본 공격의 데이터 복잡도는 2^t+1개의 선택 평문이고, 2^t+1개의 평문과 암호문을 저장하기 위해서 $(2^t+1)\cdot 8$ 바이트의 메모리가 필요하다(평문과 암호문은 각각 32 비트). 암호문을 이용한 첫 번째 필터링 과정을 거치면 $(2^{2t-32}+1)$개의 평문쌍이 남는다. 이렇게 남은 평문쌍에 대해 평문과 암호문을 이용하여 t 비트 키를 계산하므로 $2\cdot t\cdot(2^{2t-32}+1)$ Keeloq 라운드 함수 연산이 필요하다. 전체 라운드 수가 528이므로 $2\cdot t\cdot(2^{2t-32}+1)/528<1$ KeeLoq 암호화 연산이 필요하다. 선택한 2^t+1개의 평문에 대하여 암호문을 계산하는 시간을 포함하면 전체 시간 복잡도는 2^t+1로 취약키 공간의 크기인 2^t를 넘어가므로 전수조사의 시간 복잡도보다 커지게 된다.

3.1.2 슬라이드 공격 2 (2^t 취약키, $t \in \{1,2,4,8,16\}$)

슬라이드 공격 1은 공격에 필요한 평문의 수가 취약키 공간의 크기보다 크기 때문에 공격에 전수조사보다 더 많은 연산이 들었다. 그러므로 필요한 평문의 수를 줄이기 위해서 슬라이드 공격 2는 상위 $(32-t)$ 비트를



[그림 3] 마스터 키가 대칭 형태인 취약키의 특성

일정한 값으로 고정한 $2^{t'}$개의 평문의 집합 P_i $(0 \le i < 2^{t'})$와 하위 $(32-t)$ 비트를 동일한 값으로 고정한 $2^{t'}$개의 평문의 집합 $P_j^*(0 \le j < 2^{t'})$를 선택하고, 각 집합에서 하나씩 선택하여 조합한 $2^{2t'}$개의 평문쌍을 이용한다. 이러한 평문쌍이 슬라이드 쌍이 될 확률이 2^{-t}이므로, $t' = t/2$으로 정하면 조합한 2^t개의 평문쌍 중 평균 1개의 슬라이드 쌍이 존재한다. 슬라이드 공격 1과 유사하게, 슬라이드 쌍이 되기 위해서 만족해야 하는 P_i에 대응하는 암호문인 C_i의 상위 $(32-t)$ 비트와 P_j^*에 대응하는 암호문 C_j^*의 하위 $(32-t)$ 비트가 동일한 특성을 이용하여 슬라이드 쌍이 될 수 없는 평문쌍을 필터링한다. 이 후, 필터링을 통과한 평문쌍을 이용하여 계산한 t 비트 키와 대응하는 암호문쌍을 이용하여 계산한 t 비트 키가 동일한지 검사한다.

공격에 필요한 데이터 복잡도는 $2 \cdot 2^{t/2}$의 선택 평문이고, $2 \cdot 2^{t/2}$의 평문과 암호문을 저장하기 위해 $2 \cdot 2^{t/2} \cdot 8$ 바이트의 메모리가 필요하다. 이 공격의 성공 확률은 선택한 평문의 집합을 이용하여 조합한 평문쌍 중에 슬라이드 쌍이 최소 1개 이상 존재할 확률이 된다. 이 공격에서 선택한 평문쌍이 슬라이드 쌍이 될 확률은 2^{-t}이므로, 성공 확률은 $1-(1-2^{-t})^{2^t}$이다. 필터링을 통과하는 평문쌍의 개수는 평균적으로 2^{2t-32}개이므로 슬라이드 공격 1과 동일하게 $2 \cdot t \cdot 2^{2t-32}/528 < 1$ KeeLoq 암호화 연산이 필요하다. 선택한 평문에 대한 암호문을 얻는 시간 복잡도를 합하면 전체 시간 복잡도는 $2 \cdot 2^{t/2}$ KeeLoq 암호화 연산이다. 그러므로 $t = 4, 8, 16$인 경우, 전수조사보다 적은 복잡도로 키를 복구할 수 있다. $t = 16$인 경우, 2^9개의 선택 평문과 $2K$ 바이트의 메모리를 이용하여 2^9의 KeeLoq 암호화 연산으로 16 비트가 반복되는 형태의 취약키 공간에 존재하는 키를 복구한다.

3.2. 대칭 형태의 취약키 공간에 대한 슬라이드 공격 (2^{32} 취약키)

본 절에서는 마스터 키 K가 $(k_{63},\cdots,k_{32}) = (k_0,\cdots,k_{31})$의 형태일 때, 64 라운드 암호화 함수와 64 라운드 복호화 함수가 동일한 특성을 이용한 키 복구 공격을 소개한다. 이러한 형태의 마스터 키에 대해, 평문 P에 대응하는 암호문 C에 대하여 16라운드 복호화를 한 값을 X로 표시한다. 그리고 X에 대응하는 암호문을 Y로 표시하면 Y를 16라운드 복호화한 값이 P가 된다. 그러므로 Y의 하위 16 비트와 P의 상위 16 비트는 동일해야 하고, X의 하위 16 비트와 C의 상위 16 비트가 동일해야 한다. [그림 3]은 이러한 대칭 형태의 취약키가 가지는 특성을 나타낸다.

우선, 임의의 평문 P를 선택하고 이에 대응하는 암호문 C를 얻는다. C를 16 라운드 복호화 하여 발생할 수 있는 가능한 모든 값 $X_i(0 \le i < 2^{16})$을 선택하고, 각각의 X_i에 대응하는 암호문 Y_i를 얻는다. 이 중에는 P를 올바른 $k_0, k_1, \cdots, k_{15}$을 이용하여 16 라운드 복호화한 값과 일치하는 Y_i가 반드시 존재한다. 이 Y_i의 상위 16 비트와 P의 하위 16 비트가 일치하여야 하므로 이를 이용하여 틀린 Y_i를 필터링한다. 올바른 Y_i는 필터링을 확률 1로 통과하고, 틀린 Y_i는 2^{-16}의 확률로 필터링을 통과하므로 평균 $2^{16} \cdot 2^{-16} + 1 = 2$개의 Y_i가 필터링을 통과한다. 필터링을 통과한 Y_i와 P를 이용하여 계산한 16 비트 키와 이에 대응하는 X_i와 C를 이용하여 계산한 16 비트 키가 동일한지 검사한다. 옳은 키는 확률 1로 이 검사를 통과하고, 틀린 키는 랜덤하게 통과하므로 이 과정을 마치면 높은 확률로 옳은 키가 남게 된다. 이렇게 복구된 $k_0, k_1, \cdots, k_{15}$에 대해 남은 키 $(k_{16},\cdots,k_{31})$를 전수조사한다.

본 공격의 데이터 복잡도는 $2^{16}+1$개의 선택 평문이다. 첫 필터링을 통과한 평균 2개의 Y_i에 대해 평문과 암호문을 이용하여 16 라운드 키를 계산하므로 $64(=2 \cdot 16 \cdot 2)$ 라운드 연산량으로 틀린 Y_i를 필터링한다. 필터링 과정을 마치고 남은 평균 $1+2^{-16} \approx 1$개의 16 비트 키에 대해 남은 16 비트를 전수 조사한다 (2개의 후보중 1개는 올바른 Y_i에 대응하는 값이므로 항상 1개 존재). 2^{16}개의 가능한 키에 대해 두 쌍의 평문과 암호문을 이용하여 전수 조사를 하므로 필요한 연산량은

[표 1] KeeLoq에 대한 기존의 공격 결과와 본 논문에서 제안한 공격 결과의 비교

취약키 공간	공격 방법	데이터 복잡도	메모리 복잡도	시간 복잡도	성공 확률	참조
2^1	슬라이드 공격	2^1+1 CP	$<1KB$	2^1+1	1	3.1.1
	슬라이드 공격	2^1 CP	$<1KB$	2^1	0.50	3.1.2
2^2	슬라이드 공격	2^2+1 CP	$<1KB$	2^2+1	1	3.1.1
	슬라이드 공격	2^2 CP	$<1KB$	2^2	0.68	3.1.2
2^4	슬라이드 공격	2^4+1 CP	$<1KB$	2^4+1	1	3.1.1
	슬라이드 공격	2^3 CP	$<1KB$	2^3	0.64	3.1.2
2^8	슬라이드 공격	2^8+1 CP	$1KB$	2^8+1	1	3.1.1
	슬라이드 공격	2^5 CP	$<1KB$	2^5	0.63	3.1.2
2^{16}	슬라이드 공격	$2^{16}+1$ CP	$2KB$	$2^{16}+2$	1	3.1.1
	슬라이드 공격	2^9 CP	$2KB$	2^9	0.63	3.1.2
2^{32}	3.2 절	$2^{16}+1$ CP	$512KB$	$2^{17.08}$	1	3.2
2^{64}(Full)	슬라이드/Guess-and-Determine 공격	2^{32} KP	$16GB$	2^{52}	0.63	[5]
2^{64}(Full)	슬라이드/Guess-and-Determine 공격	2^{32} KP	$16GB$	$2^{50.6}$	0.63	[6]
2^{64}(Full)	슬라이드/대수적 공격	2^{16} KP	$512KB$	$2^{51.4}$	0.63	[7,8]
2^{64}(Full)	슬라이드/중간 일치 공격	2^{16} KP	$3MB$	$2^{44.5}$	0.63	[11]
2^{64}(Full)	슬라이드/중간 일치 공격	2^{16} KP	$2MB$	$2^{44.5}$	0.63	[11]

CP : 선택 평문, KP : 기지 평문
시간 복잡도 단위: KeeLoq 암호화 연산

$2^{16}\cdot 528+2^{16}\cdot 2^{-32}\cdot 528$ 라운드 연산량이다. (틀린 키가 하나의 평문과 암호문에 대하여 이 과정을 통과할 확률은 2^{-32}). 그러므로 이 두 과정에 필요한 시간 복잡도는 다음과 같다.

$$(64+(2^{16}\cdot 528+2^{16}\cdot 2^{-32}\cdot 528))/528=2^{16.08}$$

KeeLoq 암호화 연산이다. $2^{16}+1$개의 평문에 대하여 암호문을 계산하는 시간을 더하면 전체 시간 복잡도는 $2^{16.08}+2^{16}+1=2^{17.08}$ Keeloq 암호화 연산이다. 본 공격에서 선택한 평문들에는 슬라이드 쌍이 반드시 존재하므로 성공 확률은 1 이다.

Ⅳ. 공격의 효율성 비교

본 논문에서 제안하는 공격은 KeeLoq에 대한 취약키 공간에 속하는 마스터 키에 대해 적은 연산량만으로 마스터 키를 복구한다. 3.1절에서 제안한 슬라이드 공격 1은 선택한 평문의 수가 취약키 공간의 수보다 크므로 공격 자체의 의미가 적을 수 있다. 하지만 이를 슬라이드 공격 2와 3.2절에서 제안한 슬라이드 공격으로 확장하여 KeeLoq에 대한 취약키 공간이 존재함을 보이는데 의미가 있다. [표 1]은 본 논문에서 소개한 기존의 공격 결과와 본 논문에서 제안하는 공격 결과를 비교한 것이다. 여기서, 취약키 공간은 공격이 적용되는 마스터 키의 공간의 크기를 의미하고, Full은 전체 키 공간에 대한 공격을 의미한다. 데이터 복잡도는 공격에 필요한 평문의 수이고, 메모리 복잡도는 공격에 사용되는 저장 공간을 바이트 단위의 크기로 나타낸 값이며, 시간 복잡도는 KeeLoq 암호화 연산을 단위로 한다.

기존 공격은 모든 키 공간에 적용되므로, 모든 키 공간의 크기인 2^{64} 이하의 시간 복잡도로 공격을 하는 것을 목표로 하고 있다. 반면, 본 논문에서 제안하는 공격은 특정한 키 공간에 속하는 마스터 키에 대한 공격이므로, 각각의 특정 키 공간의 크기 이하의 시간 복잡도로 공격하는 것을 목표로 한다. 따라서 특정 취약키 공간에 대해서는 본 논문의 결과가 기존의 결과보다 시간 복잡도면에서 월등하나, 특정 취약키를 제외한 마스터 키에는 적용되지 않는다. 이는 KeeLoq 사용시, 본 논문에서 제안한 취약키 공간에 속하는 마스터 키에 대한

사용은 특별히 더 피해야 함을 시사한다.

Ⅳ. 결 론

블록 암호 KeeLoq는 무선 환경에 적합하게 설계 되어 여러 차량 제조 회사에서 차량을 보호하기 위한 무선 키리스 엔트리 시스템이나 차량 경보 시스템등에 사용되고 있다. 본 논문에서는 블록 암호 KeeLoq에 대한 취약키 공간을 처음으로 제안했다. $2^1 \sim 2^{32}$개의 키를 가지는 여러 취약키 공간을 소개하고, $2^4 \sim 2^{16}$개의 마스터 키로 이루어진 취약키 공간은 전수조사보다 적은 복잡도로 키를 복구 할 수 있음을 보이고, 2^{32}개의 마스터 키로 이루어진 취약키 공간에 대해 $2^{16}+1$개의 선택 평문을 이용하여 $2^{17.08}$의 KeeLoq 암호화 연산량으로 마스터 키를 복구할 수 있음을 보였다. 그러므로 본 논문에서 제안하는 취약키 공간에 속하는 형태의 마스터 키는 사용하지 않아야 한다.

참고문헌

[1] Microchip Technology Inc. KeeLoqⓇ Authentication Products, http://www.microchip.com/keeloq.

[2] 김태현, 김종성, 성재철, 홍석희, “축소된 20-라운드 SMS4에 대한 차분 공격,” 정보보호학회논문지, 18(4), pp. 37-44, 2008년 8월.

[3] 홍득조, 성재철, 이사진, 홍석희, “블록암호에 대한 새로운 다중선형공격법,” 정보보호학회논문지, 17(6), pp. 11-18, 2007년 12월.

[4] 김종성, 정기태, 이상진, 홍석희, “새로운 블록 암호 구조에 대한 차분/선형 공격의 안전성 증명,” 정보보호학회논문지, 17(1), pp. 121-125, 2007년 2월.

[5] A. Bogdanov, “Cryptanalysis of the KeeLoq block cipher,” IACR ePrint 2007-055, Feb. 2007.

[6] A. Bogdanov, “Attacks on the KeeLoq Block Cipher and Authentication Systems,” Conference on RFID Security 2007(RFIDSec 2007), July 2007.

[7] N.T. Courtois and G.V. Bard, “Algebraic and Slide Attacks on KeeLoq,” IACR ePrint 2007-062, May 2007.

[8] N.T. Courtois, G.V. Bard, and D. Wagner, “Algebraic and Slide Attacks on KeeLoq,” FSE 2008, LNCS 5086, pp. 97-115, 2008.

[9] 성재철, 문덕재, 임홍수, 지성택, 이상진, “소프트웨어 구현에 적합한 스트림 암호의 대수적 공격에 대한 안전성,” 정보보호학회논문지, 15(1), pp. 29-40, 2005년 2월.

[10] 이동훈, 김재헌, 한재우, 홍진, 문덕재, “Summation Generator에 대한 대수적 공격,” 정보보호학회논문지, 14(1), pp. 71-77, 2004년 2월.

[11] S. Indesteege, N. Keller, O. Dunkelman, E. Biham, and B. Preneel, “A Practical Attack on KeeLoq,” EUROCRYPT 2008, LNCS 4965, pp. 1 – 18, 2008.

[12] A. Biryukov and D. Wagner, “Slide Attacks,” Fast Software Encryption Workshop 1999, LNCS 1636, pp. 245 – 259. 1999.

[13] A. Biryukov and D. Wagner, “Advanced Slide Attacks,” EUROCRYPT 2000, LNCS 1807, pp. 586 – 606, 2000.

[14] S. Furuya, “Slide Attacks with a Known-Plaintext Cryptanalysis,” International Conference on Information Security and Cryptology 2001, LNCS 2288, pp. 214 – 225, 2002.

[15] J. Daemen, R. Govaerts, and J. Vandewalle, “Weak Keys for IDEA,” Crypto 1993, LNCS 773, pp. 224-231, 1994.

<著者紹介>

이 유 섭 (Yuseop Lee) 학생회원
2007년 2월: 서울시립대학교 수학과 학사
2007년 3월~현재: 고려대학교 정보경영공학전문대학원 석·박사 통합과정
<관심분야> 스트림 암호 및 해쉬 함수의 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~2008년 3월: 고려대학교 정보보호기술연구센터 연구전임강사
2008년 4월~현재: 고려대학교 정보보호기술연구센터 연구교수
<관심분야> 암호 알고리즘 설계 및 분석, 부채널 공격, 유비쿼터스 시스템

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 박사후연구원
2005년 3월~2008년 8월: 고려대학교 정보경영공학전문대학원 조교수
2008년 9월~현재: 고려대학교 정보경영공학전문대학원 부교수
<관심분야> 암호 알고리즘 설계 및 분석, 컴퓨터 포렌식