情報保護學會論文誌
第15券 第1號, 2005. 2

가변 라운드 수를 갖는 블록 암호에 대한 차분 연관 암호 공격

성 재 철,[1†] 김 종 성,[2] 이 창 훈[2‡]
[1]서울시립대학교 수학과, [2]고려대학교 정보보호기술연구센터

Differential Related-Cipher Attacks on Block Ciphers with Flexible Number of Rounds

Jaechul Sung,[1†] Jongsung Kim,[2] Changhoon Lee[2‡]
[1]University of Seoul, [2]Korea University

요 약

연관 암호 공격은 키의 길이에 따라 가변 라운드 수를 갖는 블록 암호알고리즘의 키 스케쥴의 약점을 이용한 공격으로 2002년 Hongjun Wu에 의해 소개되었다.[22] 이 공격은 두 개의 서로 다른 키 길이에 대해 각기 다른 라운드 수를 갖는 두 암호에서 사용되는 키 쌍이 유사-동치키일 경우에 적용되는 공격이다. 본 논문에서는 이러한 연관 암호 공격을 블록 암호알고리즘의 가장 강력한 공격 방법인 차분 공격과 결합한 차분 연관 암호 공격의 개념을 제시한다. 차분 연관 암호 공격은 기존의 연관 암호 공격에서 라운드 수의 차이가 커짐에 따라 공격 적용의 어려움을 극복한 방법이다. 이 공격법을 이용하여 블록 암호알고리즘 ARIA v.0.9와 SC2000을 분석한다.[14,21] 또한, 차분 공격 뿐 아니라 선형 공격, 고계 차분 공격 등과 같은 기존의 블록 암호알고리즘 공격들과 연관 암호 공격이 결합 가능함을 이용하여, 가변 라운드 수를 갖는 블록 암호알고리즘인 SAFER++와 CAST-128을 분석한다.[16,1]

ABSTRACT

Related-Cipher attack was introduced by Hongjun Wu in 2002.[22] We can consider related ciphers as block ciphers with the same round function but different round number and their key schedules do not depend on the total round number. This attack can be applied to block ciphers when one uses some semi-equivalent keys in related ciphers. In this paper we introduce differential related-cipher attacks on block ciphers, which combine related-cipher attacks with differential cryptanalysis. We apply this attack to the block cipher ARIA and SC2000.[14,21] Furthermore, related-cipher attack can be combined with other block cipher attacks such as linear cryptanalysis, higher-order differential cryptanalysis, and so on. In this point of view we also analyze some other block ciphers which use flexible number of rounds, SAFER++ and CAST-128.

Keywords : *Block cipher, Related-cipher attack, Differential cryptanalysis, ARIA, SC2000, SAFER++, CAST-128*

접수일 : 2004년 11월 4일 ; 채택일 : 2005년 1월 24일
* 이 연구는 2004년도 서울시립대학교 학술연구조성비에 의하여 연구되었음
† 주저자, jcsung@uos.ac.kr
‡ 교신저자, crypto77@cist.korea.ac.kr

Ⅰ. 서 론

연관 키 공격과 연관 암호 공격은 마스터 키를 이용하여 라운드 키를 생성하는 키 스케줄 알고리즘 약점을 이용하는 공격방법이다. 연관 키 공격(Related-Key Attack)은 어떠한 연관 관계에 있는 서로 다른 키를 이용하여 평문과 암호문의 관계를 이용하여 공격하는 블록 암호알고리즘의 분석법으로, 블록 암호알고리즘의 키 스케줄의 약점을 이용한 공격법이다.[5] 이 연관키 공격에서는 고정된 암호화 방식(일반적으로 같은 라운드 함수와 같은 라운드 수를 이용함)을 이용하여 두 키의 연관성을 이용하여 공격하는 반면, 2002년 H. Wu에 의해 소개된 연관 암호 공격(Related-Cipher Attack)은 연관성이 있는 서로 다른 암호화 방식(일반적으로 같은 라운드 함수를 같지만 라운드 수만 다른 경우)에 두 암호에 사용되는 키의 유사 동치성(semi-equivalent)을 이용한 공격 방법이다.[22] 이 공격은 AES, ARIA, CAST-128 등과 같은 키 길이에 따라 가변 라운드를 사용하는 블록 암호알고리즘에 주로 적용될 수 있는 공격법이다. 특히, 블록 암호알고리즘이 키 스케줄의 약점을 갖고 두 라운드의 수가 작은 경우에 용이하게 적용가능한 방법이다.

차분 분석법(Differential Cryptanalysis)[4]은 블록 암호알고리즘의 가장 강력한 분석법으로 하나로, 입력 차분에 대한 출력 차분의 비균일성을 이용한 방법이다. 블록 암호알고리즘이 이러한 차분 분석에 안전하기 위해서는 차분 특성 확률이 낮은 라운드 함수를 이용하여 충분히 안전한 라운드를 반복하여 설계되어야 한다.

연관 암호 공격은 두 연관된 암호에서 라운드 수의 차이가 작은 경우에는 쉽게 적용 가능하나, 라운드 수가 늘어나면 공격 적용이 쉽지 않다. 본 논문에서는 차분 분석법의 개념을 이용하여 연관 암호 공격을 확장시키는 방법인 차분 연관 암호 공격(Differential Related-Cipher Attack)을 소개한다. 이 공격은 차분 공격 과 연관 암호 공격을 결합한 공격법으로, 라운드 수의 차이가 큰 경우에도 적용가능하다. 이러한 차분 연관 암호 공격법을 이용하여 가변 라운드 수를 갖는 블록 암호알고리즘인 ARIA v.0.9[14]와 SC2000[21]에 대한 공격을 소개한다. 또한, 차분 분석 뿐 아니라 선형 공격, 고계 차분 공격, 스퀘어 공격 등의 기존의 다른 블록 암호알고리즘의 분석법을 연관 암호 공격과 결합할 수 있음을 보인다. 이러한 공격 결합을 이용하여 SAFER++[16]과 CAST-128[1]에 대한 연관 암호 분석 관점에서의 안전성을 살펴본다.

Ⅱ. 연관 키 공격과 연관 암호 공격

블록 암호알고리즘의 키 스케줄의 약점을 이용한 공격법으로는 취약 키 공격[10,11], 슬라이드 공격[7,8], 연관 키 공격[12,5], 연관 암호 공격[22] 등이 있다.

취약 키 공격은 기존의 차분 공격, 선형 공격 등의 블록 암호알고리즘의 분석에 대해 일반적인 키보다 쉽게 분석 가능한 키(취약 키)들에 대해 분석하는 방법이다. 단순한 키 스케줄을 사용하는 블록 암호알고리즘 IDEA[15]의 경우, 이러한 취약 키 공격에 대한 많은 분석이 소개되었다.[6,10,11] IDEA의 경우, $2^{16}+1$ 에서의 곱셈 연산을 사용하기 때문에, 라운드 키가 0이거나 1인 경우 차분 공격이나 선형 공격에 쉽게 노출될 수 있다.

슬라이드 공격은 블록암호의 자체 연관성을 이용한 공격법으로, 반복적인 구조를 사용하는 블록 암호알고리즘에서 라운드 키의 주기성을 이용한다. 이 공격 역시 블록 암호알고리즘의 키 스케줄의 약점을 이용한 공격법이다.

연관 키 공격은 선택 키 공격법으로 서로 다른 두 키 사이의 연관성을 이용하여 기존의 블록 암호알고리즘의 분석법을 적용하여 공격하는 방법이다. 이 공격에서는 라운드 수는 고정되고 키와 연관되어 있다.

연관 암호 공격은 라운드 수가 가변이고 키가 고정된 블록 암호알고리즘을 분석하는 방법으로 2002년H. Wu에 의해 소개되었다. 이 공격을 블록 암호알고리즘 SQUARE와 ACISP 2002에서 제안된 변형된 AES[17]에 적용하여 분석하였다. 연관 암호 공격은 블록 암호알고리즘에서의 라운드 수에는 무관하나, 두 연관 암호의 라운드 수의 차이에 의존한다.

연관 키 공격에서는 사용되는 두 키는 서로 다르나 연관되어 있음을 이용하여 분석하는 방법인 반면, 연관 암호 공격에서는 가변 라운드를 사용하는 블록 암호알고리즘에서 라운드 수의 차이를 이용하는 분석법이다. 두 라운드 수의 차이가 작다면 쉽게 연관 암호 공격이 가능하다. 그러나 두 라운드 수의 차이가 점점 커질수록 기존의 연관 암호 공격을 그대로 적용하기는 쉽지 않다. 이러한 문제점을 극복하기 위해, 본 논문에서는 연관 암호 공격과 기존의 차분 분석의 기법을 결합한 차분 연관 암호 공격

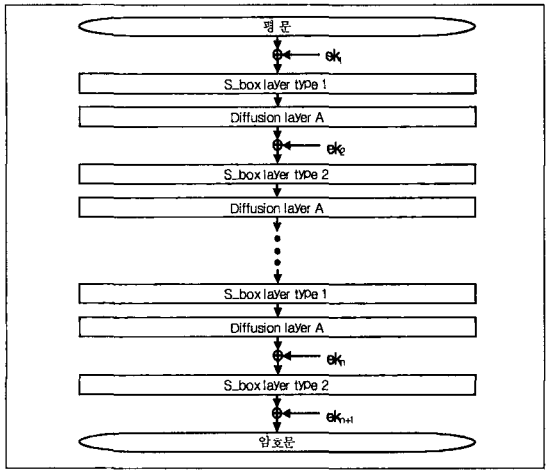


그림 1. 블록 암호알고리즘 ARIA의 구조도

(Differential Related-Cipher Attack)이라는 새로운 분석 개념을 도입한다. 또한, 기존의 블록 암호알고리즘의 분석법과 연관 암호 공격을 결합하여, 선형 연관 암호 공격, 고계 차분 연관 암호 공격, 포화 연관 암호 공격 등으로 개념을 확장시킬 수 있다.

Ⅲ. ARIA v.0.9에 대한 차분 연관 암호 공격

3.1 블록 암호알고리즘 ARIA의 소개

ARIA v.0.9는 2003년 국내에서 개발된 AES와 같은 SPN 구조의 128-비트 블록 암호알고리즘이다. ARIA는 키 길이에 따라 가변 라운드를 갖는다. 즉, 키가 128비트인 경우 10라운드, 192비트인 경우 12라운드, 256비트인 경우 14라운드를 사용한다. 2004년 v.0.9의 키 스케줄을 수정과 라운드 수를 각 2 라운드 증가하여 v.1.0을 개발하였다. 본 논문에서는 ARIA v.0.9를 ARIA로 표기하고, 이 ARIA v.0.9에 대한 분석한다.

암호화 함수와 복호화 함수가 동일한 구조를 같은 Khazad[2]와 Anubis[3]와 같은 involution($x^{-1}=x$) 암호는 작은 칩의 사이즈와 짧은 코드 길이로 하드웨어 구현 가능하다는 장점을 가지고 있다. 그러나 구조의 자기 연관성으로 인한 안전성의 문제가 제기되었다.[9] ARIA는 전체적으로는 구현상의 효율성을 고려하여 involution 구조를 사용하였으나, 완전한 involution 구조는 아니다. 특히, substitution layer는 involution이 아니다.

ARIA의 라운드는 round key addition layer, substitution layer, diffusion layer로 구성되어 있다. Substitution layer는 S_1, S_2, S_1^{-1}, S_2^{-1} 4개의 S-box를 사용한다. Substitution layer는 두 종류의 type 1과 type 2를 순차적으로 사용한다. 이것은 ARIA의 전체적인 구조적 특징인 involution의 성질에도 부합한다. S-box layer type 1은 다음과 같이 구성된다.

$$SL1 : (S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1})$$

S-box layer type 2는 type 1의 역함수 꼴로 다음과 같이 구성되어 있다.

$$SL2 : (S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2)$$

Diffusion layer는 involution 성질($A^{-1}=A$)을 만족하는 행렬 A를 사용한다. A의 원소의 값은 0 혹은 1의 값을 갖는 이진 행렬(binary matrix)이다. 이 행렬은 역행렬을 갖는 이진 행렬 중 확산 효과가 가장 좋은 행렬로써, branch number의 수가 8이다. 다음은 행렬 A를 표현한 것이다.

$$A = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix}$$

ARIA에 대한 차분 분석, 선형 분석, 불능 차분 분석, 부정 차분 특성 등에 대한 안전성 분석도 알고리즘과 함께 제시되었다.[13]

■ ARIA 암호 표기

$\sigma[ek_i]$를 i번째 라운드 key addition layer 함수로 표기하고 A를 diffusion layer 함수라고 하자. 또한, γ_i를 substitution layer 함수로 놓자

(γ_i는 i가 홀수이면 $SL1$이고 i가 짝수이면 $SL2$가 된다). 그러면, ARIA의 i번째 라운드 함수는 $\phi[ek_i]$는 다음과 같이 표현할 수 있다.

$$\phi[ek_i] = A \circ \gamma_i \circ \sigma[ek_i]$$

이러한 표현을 이용하여 키 길이 128/192/256-비트에 따른 t-라운드(t=10, 12, 14) ARIA는 다음과 같이 표기한다.

$$ARIA^t(MK) = \sigma[ek_{t+1}] \circ \gamma_t \circ \sigma[ek_{t+1}] \circ \phi[ek_{i-1}] \circ \cdots \circ \phi[ek_1]$$

■ ARIA v.0.9의 키 스케줄

ARIA는 세 종류의 키 길이(128/192/256)에 따라 가변 라운드를 갖는 블록 암호알고리즘이다. ARIA의 키 스케줄은 초기화 과정과 라운드 키 생성 과정의 두 과정으로 구성되어 있다.

초기화 과정은 최초의 마스터 키 MK로부터 3-라운드 Feistel 구조를 이용하여 4개의 128-비트 값 W_0, W_1, W_2, W_3을 생성한다. 여기서 사용되는 F_o와 F_e는 각각 ARIA의 홀수 라운드 함수와 짝수 라운드 함수를 나타낸다. 그리고 각 라운드 함수에는 고정된 상수 CK_i를 키로 사용한다. ARIA는 가변 키 길이를 사용하므로 최초의 마스터 키 MK는 128, 192, 256 비트가 될 수 있다. 하지만, 초기화 과정에 최초 입력되는 $(KL\|KR)$는 256비트이므로 마스터 키가 256 비트보다 작은 경우, 단순히 0으로 패딩(padding)하여 $(KL\|KR)$의 값을 얻는다.

$$(KL\|KR) = MK\|00 \cdots 0$$

이러한 단순한 패딩으로 $(KL\|KR)$의 값을 생성하여 서로 다른 키 길이를 같은 두 개의 키가 같은 $(KL\|KR)$을 생성하여, 동일한 4개의 128-비트 값 W_0, W_1, W_2, W_3을 생성하는 약점이 존재한다.

$(KL\|KR)$로부터 W_i들을 생성하는 초기화 과정을 거친 후, 이 W_i값들을 적당히 조합하여 각 라운드에 사용되는 라운드 키를 생성하는 라운드 키 생성 과정은 다음과 같다. 여기서, $\ll a$은 왼쪽으로 a-비트만큼의 쉬프트 로테이션을 의미하고, $\gg a$은 오른쪽으로 a-비트만큼의 쉬프트 로테이션을 의미한다.

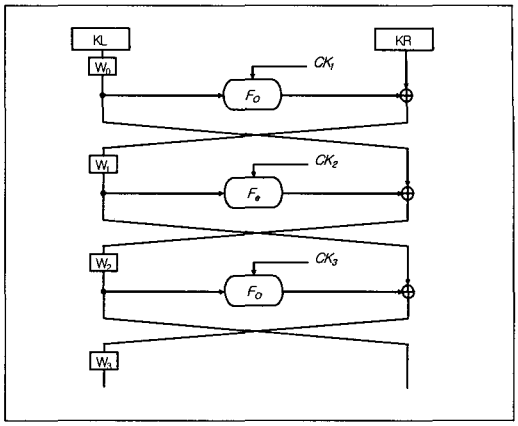


그림 2. ARIA의 키 스케줄의 초기화 과정

$$ek_1 = (W_0^{\gg 7}) \oplus (W_1^{\ll 11}),$$
$$ek_2 = (W_1^{\ll 22}) \oplus (W_2),$$
$$ek_3 = (W_2^{\gg 17}) \oplus (W_3^{\ll 16}),$$
$$ek_4 = (W_0^{\gg 14}) \oplus (W_3^{\ll 22}),$$
$$ek_5 = (W_0^{\gg 21}) \oplus (W_2^{\gg 34}),$$

$$ek_6 = (W_1^{\ll 33}) \oplus (W_3^{\ll 48}),$$
$$ek_7 = (W_1^{\ll 44}) \oplus (W_2^{\gg 51}),$$
$$ek_8 = (W_0^{\gg 28}) \oplus (W_3^{\ll 64}),$$
$$ek_9 = (W_1^{\ll 55}) \oplus (W_3^{\ll 80}),$$
$$ek_{10} = (W_0^{\gg 35}) \oplus (W_2^{\gg 68}),$$
$$ek_{11} = (W_0^{\gg 42}) \oplus (W_1^{\ll 66}),$$
$$ek_{12} = (W_1^{\ll 77}) \oplus (W_2^{\ll 85}) \oplus (W_3^{\ll 96}),$$
$$ek_{13} = (W_0^{\gg 49}) \oplus (W_2^{\gg 102}),$$
$$ek_{14} = (W_2^{\ll 119}) \oplus (W_3^{\ll 112}) \oplus (KR^{\ll 64}),$$
$$ek_{15} = (W_0^{\gg 56}) \oplus (W_1^{\ll 88}) \oplus (KR).$$

마스터 키의 길이가 각각 128/192/256-비트일 때, 각 라운드 수는 10/12/14이다. 따라서 마스터 키 길이에 따라 각각 11/13/15개의 라운드 키를 사용한다.

3.2 ARIA v.0.9에 대한 공격

ARIA는 가변 라운드 수를 갖는 알고리즘이다. 하지만, 키 스케줄 알고리즘이 다른 블록 암호알고리즘에 비해 상대적 간단히 설계되어, 마스터 키의 사

이즈에 의존하지 않는다. 따라서 서로 다른 키 사이즈를 사용하는 경우, 마지막 차이가 나는 라운드를 제외하면 같은 라운드 키를 갖는 서로 다른 마스터 키 쌍을 쉽게 찾을 수 있다. 우리는 이러한 키 쌍을 유사-동치키(semi-equivalent keys)라 하겠다.

ARIA의 키 스케쥴 초기화 과정은 마스터 키 MK를 이용하여 256 비트인 $(KL\|KR)$을 생성한다. K를 128 비트의 마스터 키라 하자. 그러면, 키 스케쥴에 의해 $(KL\|KR) = K\|0^{128}$이 된다. 또한, 키의 길이가 192 비트의 마스터 키 $MK = K\|0^{64}$이라하면, $(KL\|KR) = K\|0^{64}\|0^{64}$이다. 따라서 키 길이가 서로 다른 두 키($K$와 $K\|0^{64}$)는 같은 $(KL\|KR)$을 생성한다. 따라서 라운드 키 생성과정에 의해 마지막 두 라운드 키(ek_{12}와 ek_{13})을 제외한 다른 라운드 키(ek_1, ek_2, $\cdots$, ek_{11})은 같은 값을 갖는다. 이 때, 사용된 두 키(K와 $K\|0^{64}$)는 서로 유사-동치키가 된다.

이러한 성질을 이용하여 서로 다른 키 사이즈를 사용하여 같은 평문 P를 암호화한 경우에 대한 차분 연관 암호 공격에 대해 살펴보자. 두 개의 서로 다른 두 키, 128-비트의 키(K)와 192-비트의 키($K\|0^{64}$)가 서로 유사-동치키라고 가정한다. 그리고 128-비트의 키에 의해 평문 P를 암호화하여 얻은 암호문을 C라 놓고, 192-비트의 키에 의해 같은 평문 P를 암호화하여 얻은 암호문을 C'라 하자. 그러면, C와 C'는 다음의 식을 만족한다.

$$\begin{aligned} C &= ARIA^{10}(K) \\ &= \sigma[ek_{11}] \circ \gamma_{10} \circ \sigma[ek_{10}] \circ \phi[ek_9] \circ \cdots \circ \phi[ek_1](P) \\ C' &= ARIA^{12}(K\|0^{64}) \\ &= \sigma[ek_{13}] \circ \gamma_{12} \circ \sigma[ek_{12}] \circ \phi[ek_{11}] \circ \cdots \circ \phi[ek_1](P) \end{aligned}$$

ARIA에서 사용된 diffusion 행렬 A는 선형이고 involution이므로, 위의 두 식은 C와 C'의 관계식으로 다음과 같이 표현된다.

$$\begin{aligned} C' &= \sigma[ek_{13}] \circ SL2 \circ A \circ \sigma[A(ek_{12})] \circ SL1 \\ &\quad \circ \sigma[ek_{11} \oplus A(ek_{11})](A(C)) \\ &\updownarrow \\ A(C) &= \sigma[ek_{11} \oplus A(ek_{11})] \circ SL1^{-1} \circ \sigma[A(ek_{12})] \\ &\quad \circ A \circ SL2^{-1} \circ \sigma[ek_{13}](C') \end{aligned}$$

위의 C와 C'는 그림 3의 관계에 있다. 즉, 유사-동치키의 관계에 의해 같은 평문을 암호화한 두 암호문은 서로 ARIA의 2-라운드의 관계식으로 표현된다. 따라서 두 암호문을 이용하면 마지막 3개의 라운드 키(ek_{11}, ek_{12}, ek_{13})를 찾을 수 있다. 이 세 개의 라운드 키를 찾는 과정을 자세히 살펴보자.

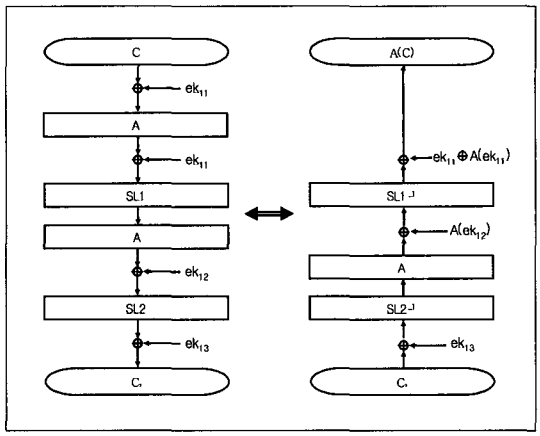


그림 3. 10-라운드와 12-라운드 ARIA의 연관 암호

임의의 128-비트의 값 $X = (X_1, X_2, \cdots, X_{16})c$로 표기한다. 여기서 각 X_i는 8-비트 값이다. 평문 P_1의 128-비트의 키(K)에 의한 암호문과 192-비트의 키($K\|0^{64}$)에 의한 각 암호문의 쌍을 (C_1, C_1')라 하자. 마찬가지로, 평문 P_2에 대한 암호문의 쌍을 (C_2, C_2')라 하자. 이 두개의 쌍을 가지고 차분 분석[17]의 기법을 이용하여 각 라운드 키를 찾는다. 그림 3의 오른쪽 부분의 연관 암호 쌍을 이용한다.

일반적으로 S-박스가 주어지면 S-박스에 대한 입출력 차분 분포 표를 구성할 수 있다. 만약 S-박스 이전에 키 XOR 연산이 있을 경우, XOR 연산 이전의 입력 차분과 S-박스 연산 이후의 출력 차분을 알면 옳은 키를 찾을 수 있다.

그림 3의 $SL1^{-1}$의 첫 번째 S-박스의 입출력 차분 값을 이용하여 ek_{12}와 ek_{13}의 값을 찾는다. 우선, $SL1^{-1}$의 첫 번째 S-박스에 영향을 주는 키 값인 ek_{13}의 처음 7개 바이트(ek_{13}^4, ek_{13}^5, ek_{13}^7, ek_{13}^9, ek_{13}^{10}, ek_{13}^{14}, ek_{13}^{15})과 $A(ek_{12})$의 첫 번째 바이트(ek_{12}^1)을 추측한다. 그러면, $SL1^{-1}$의 첫 번째 S-박스의 입력 차분을 알 수 있고, 출력 차분은 $A(C)$와 $A(C')$의 첫 번째 바이트의 차분이므로, S-박스 차분 분포 표를 이용하면 추측한 키 바이트들이 옳은 값인지 여

부를 판별할 수 있다. 옳은 키는 항상 이러한 과정을 통과하고, 랜덤한 키의 경우 확률 2^{-8}으로 통과한다. 따라서 추측하는 키는 64 비트이므로, 약 9개의 연관 암호 쌍이 있으면 옳은 키를 거의 유일하게 찾을 수 있다 ($2^{64} \times e^{-8 \cdot 9} < 1$).

$SL1^{-1}$의 두 번째 S-박스의 살펴보면, ek_{13}의 5개 바이트와 $A(ek_{12})$의 두 번째 바이트의 값을 위와 같은 방법으로 구할 수 있다. 마찬가지의 방법으로 나머지 ek_{13}의 3 바이트를 구할 수 있다. ek_{13}을 모두 찾고 $A(ek_{12})$의 4 바이트를 찾으면, 나머지 $A(ek_{12})$의 12 바이트는 차분 분포 표를 이용하면 쉽게 구할 수 있다. $A(ek_{12})$을 찾으면 ek_{12}의 값은 쉽게 구한다.

이러한 과정으로 ek_{12}와 ek_{13}의 값의 모두 구하면, $A(C)$와 $ek_{11} \oplus A(ek_{11})$의 XOR 값을 얻을 수 있다. 이 값을 X라 하면 다음 식을 얻는다.

$$X = A(C) \oplus (ek_{11} \oplus A(ek_{11}))$$
$$\leftrightarrow (A+I)(ek_{11}) = A(C) \oplus X$$

여기서 I는 16×16 항등 행렬이고, $A+I$의 덧셈 연산은 각 행렬 원소의 XOR 연산이다(A와 I의 행렬 원소의 값은 0 또는 1이다). 행렬 $A+I$의 위수(rank)를 구하면 14이다. 따라서, 우리는 ek_{11}을 유일하게 결정할 수는 없지만, 2^{16}개의 동치키를 찾을 수 있다. 하지만, 다른 방법을 이용하거나, 전수조사 방법을 이용하면 ek_{11}의 옳은 키 값을 정확히 찾을 수 있다.

이 공격에는 약 9개의 연관 암호 쌍이 필요하며, 총 2^{64}번의 S-박스 연산이 필요하다. 만약 이러한 공격으로 마지막 3개의 라운드 키(ek_{11}, ek_{12}, ek_{13})을 찾았다면 이것을 이용하여 마스터 키를 찾는 문제를 살펴보자.

두 연관 암호에 사용된 키는 (K와 $K \| 0^{64}$)에서 K는 ARIA의 키 스케줄 초기화 과정에 의하면 W_0의 값과 같다. 또한, $ek_{11} = (W_0^{\gg 42}) \oplus (W_1^{\ll 66})$이므로, $W_1 = (W_0^{\ll 20}) \oplus (ek_{11}^{\ll 62})$이다. 또한, W_0과 W_1는 $W_1 = F_o(W_0)$을 만족한다. 여기서, W_0의 값을 x라 놓고 $ek_{11}^{\ll 62}$을 c라 한다면, 두 관계 식 $W_1 = (W_0^{\ll 20}) \oplus (ek_{11}^{\ll 62})$과 $W_1 = F_o(W_0)$을 이용하여 다음 식을 얻는다.

$$A \circ SL1(x \oplus CK_1) = x^{\ll 20} \oplus c$$

행렬 A는 선형이고, $SL1$에 사용되는 S-박스는 $GF(2^8)$위에서의 x^{-1} 함수의 아핀 변환이다. 따라서, 키의 값 x를 $(x_1, x_2, \cdots, x_{128})$으로 표현하면, 위의 식에서 x의 값을 구하는 문제는 128개의 변수와 128개의 이차 방정식을 푸는 문제로 귀결된다. 하지만, 아직까지 이러한 방정식을 푸는 효과적인 방법은 알려져 있지 않다.

이상에서 살펴본 공격은 128-비트 키를 이용한 10-라운드 ARIA와 192-비트 키를 이용한 12-라운드 ARIA 암호에 대한 차분 연관 암호 공격이다. 이와 같은 방법으로 192-비트를 사용하는 12-라운드 ARIA와 256-비트를 사용하는 14-라운드 ARIA에 대해서도 같은 공격이 가능하다. 하지만 각 라운드에 사용되는 라운드 키의 길이가 128 비트이므로, 차분 분석의 방법을 사용하지 않고 기존의 연관 암호 공격만으로는 효과적으로 공격할 수 없다. 즉, 두 연관 암호의 라운드 수의 차가 클수록 기존 연관 암호 분석만으로는 공격할 수 없다. 따라서, 연관 암호 분석의 개념에 차분 분석법을 결합한 차분 연관 암호 공격은 기존의 연관 암호 분석을 보다 효과적으로 확장할 수 있는 좋은 분석법이다.

ARIA는 키 스케줄 과정이 라운드 수와 관계없이 라운드 키를 생성하여, 유사-동치키가 발생하는 취약점이 발생하였다. ARIA와 같이 키 길이에 따라 가변 라운드 수를 사용하는 블록 암호 알고리즘의 경우, 키 스케줄의 설계 시 이러한 성질이 발생하지 않도록 주의하여야 한다.

실제로 ARIA v.1.0의 경우에는 이러한 연관 암호 공격이 적용되지 않도록 키 길이에 따른 서로 다른 상수를 사용하여 설계하였다.

Ⅳ. SC2000에 대한 차분 연관 암호 공격

SC2000[21]은 일본에서 개발한 알고리즘으로, 유럽의 NESSIE 프로젝트와 일본의 CRYPTREC 프로젝트에 제안된 128-비트 블록 암호 알고리즘이다. CRYPTREC 프로젝트에서는 최종 후보 알고리즘으로 선택되기도 하였다. AES나 ARIA와 마찬가지 SC2000도 키 길이에 따라 가변 라운드 수를

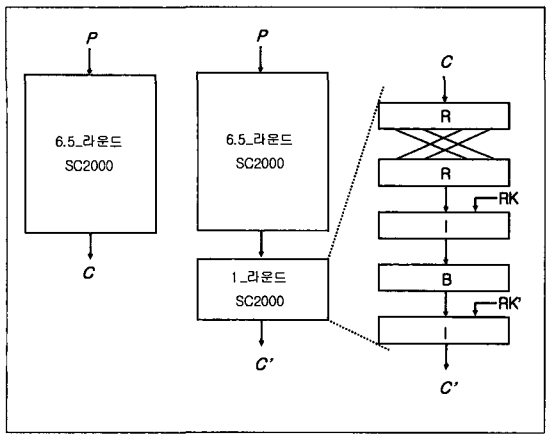


그림 4. 6.5-라운드와 7.5-라운드 SC2000의 연관 암호

갖는 알고리즘이다. 키 길이가 128/192/256-비트인 경우 각각 6.5/7.5/7.5 라운드를 갖는다.

SC2000의 각 라운드는 $(R\times R)\circ I\circ B\circ I$로 정의된다. 여기서, I는 비트별 키 XOR 함수이고,B는 S-box 대치 부분으로 128-비트를 32개의 4-비트로 나누어서 4-비트 입출력을 갖는 S-박스의 값으로 대치시키는 것이다. $(R\times R)$에서 R은 데이터 치환 함수 구조를 갖는 함수이고, $\times$는 4개의 32-비트의 값의 순서를 바꾸는 연산과정이다. 즉, $\times$은 (a_0,a_1,a_2,a_3)를 (a_2,a_3,a_0,a_1)으로 바꾼다. 그리고 앞의 $(R\times R)$을 0.5 라운드, $I\circ B\circ I$를 0.5 라운드라고 표기한다.

SC2000의 키 스케줄 과정도 ARIA의 경우와 마찬가지로 라운드 수과 관계없이 설계되었다. 따라서 128-비트의 키와 192-비트의 키가 유사-동치키 관계가 되도록 만들 수 있다. 마찬가지로, 128-비트와 256-비트 키 쌍도 유사-동치키 관계가 되도록 만들 수 있다.

SC2000은 마스터 키는 각 키 길이에 따라 다음과 같은 방법으로 8개의 32-비트를 생성한 후, 이 생성된 값을 이용하여 키 스케줄 과정을 수행한다. 여기서, K_1는 128-비트, K_2는 192-비트, K_3는 256-비트 키이다.

$$K_1=(K_1^1,K_1^2,K_1^3,K_1^4)\rightarrow(K_1^1,K_1^2,K_1^3,K_1^4,K_1^1,K_1^2,K_1^3,K_1^4)$$

$$K_2=(K_2^1,K_2^2,\ldots,K_2^6)\rightarrow(K_2^1,K_2^2,K_2^3,K_2^4,K_2^5,K_2^6,K_2^1,K_2^2)$$

$$K_3=(K_3^1,K_3^2,\ldots,K_3^8)\rightarrow(K_3^1,K_3^2,K_3^3,K_3^4,K_3^5,K_3^6,K_3^7,K_3^8)$$

만약 128-비트의 키 $K_1=(K_1^1,K_1^2,K_1^1,K_1^2)$이고, 192-비트 키 $K_2=(K_1^1,K_1^2,K_1^1,K_1^2,K_1^1,K_1^2)$가 이러한 형태를 갖는다면 두 키에 의해 생성된 8개의 32-비트 값은 같게 되고, 두 키에 의해 생성된 앞의 6.5-라운드 키는 동일한 값이 된다. 즉, 두 키는 유사-동치키 관계가 된다. 같은 논리를 이용하여, 128-비트 키 $K_1=(K_1^1,K_1^2,K_1^3,K_1^4)$와 256-비트 키 $K_3=(K_1^1,K_1^2,K_1^3,K_1^4,K_1^1,K_1^2,K_1^3,K_1^4)$도 유사-동치키 관계가 된다.

유사 동치키 쌍 (K_1,K_2)혹은 (K_1,K_3)이 사용된 연관 암호가 사용된 경우에 마지막 7.5-라운드의 라운드 키 RK와 RK'를 복구하여 보자. P를 평문이라 하고 (C,C')를 각 키 사이즈에 사용하여 얻은 암호문 쌍이라 하자. 또한, 암호문 C가 $(R\times R)$의 과정을 거친 후의 값을 D라 하자. 즉, $C=(R\times R)(D)$이다.

차분 연관 암호 공격을 위해 평문 P_1과 P_2을 연관 암호에 의해 얻은 두 암호문 쌍을 (C_1,C_1')과 (C_2,C_2')라 하자. 그러면 $(R\times R)$의 과정에는 키에 의한 연산이 없으므로 $C_1=(R\times R)(D_1)$과 $C_2=(R\times R)(D_2)$의 값을 각각 구할 수 있다. 따라서, S-박스 대치 연산인 B 함수의 각 입력 차분과 출력 차분을 얻을 수 있으므로, 4-비트 입출력을 갖는 S-박스에 대한 차분 분포 표를 이용하면 쉽게 RK의 값을 쉽게 구할 수 있다. RK의 올바른 값을 구하기 위해서는 약 2개의 차분 쌍이 있으면 된다. RK의 값을 찾으면, RK'은 쉽게 얻을 수 있다. 따라서 7.5-라운드에 사용된 두 개의 키 값을 모두 찾을 수 있다.

SC2000에 대한 차분 연관 암호 공격은 약 2^6번 정도의 S-박스 연산과 약 4~6개의 연관 암호 쌍이 필요하다. SC2000의 경우도 라운드 키 생성하는 과정에서 256-비트보다 작은 경우 256-비트로 확장하는 단계에서 키 길이가 다른 두 개의 키가 유사-동치키가 발생하여 이러한 공격이 가능하게 되었다.

SC2000의 경우의 경우, 연관 암호 공격만으로는 정확한 키의 값을 구하기 어렵다. 같은 논리로 만약 두 연관 암호의 라운드 차가 더 커진다면 연관 암호 공격은 불가능하다. 이 경우, 차분 분석과 연관 암호 공격을 결합한 차분 연관 암호 공격을 적용한다면 보다 기존의 공격보다 적용 범위를 확장시킬 수 있는 강력한 공격이 될 수 있다.

V. SAFER++와 CAST-128에 대한 연관 암호 분석

연관 암호 공격은 두 연관 암호의 라운드 수가 작은 경우에 적용되는 공격법이다. 하지만, 두 라운드 수의 차이가 커지면 연관 암호 공격만으로는 공격하기 어렵다. 그래서 본 논문에서는 차분 분석의 개념을 연관 암호 공격과 결합한 차분 연관 암호 분석을 도입하여 ARIA v.0.9와 SC2000을 분석하였다. 이 개념을 보다 일반적으로 확장하면, 차분 분석 뿐 아니라 선형 분석, 고계 차분 분석 등을 연관 암호 공격과 결합할 수 있다.

본 절에서는 가변 라운드 수를 갖는 다른 블록 암호 알고리즘인 SAFER++[16]과 CAST-128[1]에 대해 기존의 선형 분석, 고계 차분 분석 등의 결과를 활용하여 연관 암호 공격을 적용하여 본다. 연관 암호 공격은 마스터 키를 이용하여 각 라운드 키를 생성하는 과정에서의 유사-동치키 관계가 발생하는 약점을 이용한 공격이므로 SAFER++와 CAST-128 알고리즘에 대한 자세한 설명은 생략한다.

SAFER++는 128-비트 블록 암호 알고리즘이고, 키 길이 128/256-비트에 따라 각각 7/10-라운드를 사용한다. ARIA와 SC2000의 라운드 키 생성의 취약점과 마찬가지로 두 개의 서로 다른 키 사이즈에 대해 유사-동치키를 생성할 수 있다. 라운드 수의 차가 3이므로 기존 연관 암호 공격은 효과적으로 적용할 수 없다.

지금까지 알려진 SAFER++의 3 라운드에 대한 가장 효과적인 공격은 선형 분석이다.[19] 이 선형 분석에는 2^{101}의 암호화 연산과 2^{81}의 기지 평문이 필요하다. 이러한 분석 결과를 그대로 연관 암호 공격에 적용할 수 있다. 이 공격은 선형 분석과 연관 암호 분석을 결합한 선형 연관 암호 공격이 되는 것이다.

128-비트 블록 암호 CAST-128은 키 길이가 40~80 비트인 경우 12 라운드를 사용하고, 키 길이가 81~128 비트인 경우에는 16 라운드를 사용한다. 마스터 키 K의 길이가 128-비트보다 작은 경우 뒤에 0을 첨가하여 128-비트가 되도록 $K' = K \| 0 \cdots 0$ 생성한 후, K'를 이용하여 라운드 키를 만든다. 따라서 길이가 다른 두 키에 대해 유사-동치키를 만들 수 있다. 두 키가 모두 40~80 비트 사이 혹은 81~128 비트 사이에 있는 경우에는 동치키가 되고, 한 키가 40~80 비트 사이에 있고 다른 키가 81~128 비트에 있는 경우 유사-동치키를 이용하여 연관 암호 공격을 할 수 있다.

두 개의 연관 암호의 라운드 수의 차가 4이다. 따라서, 연관 암호 공격을 그대로 적용해서는 마지막 4 라운드에 사용된 라운드 키들을 찾을 수는 없다. 따라서, 기존에 알려진 4-라운드 CAST-128에 대한 공격과 결합하여 분석한다. CAST-128에 대해 가장 효과적인 공격은 5-라운드 CAST-128에 대한 고계 차분 공격이다. 이 경우, 2^{17}개의 선택 평문과 2^{40}번의 암호화가 필요하다. 이를 이용하면, 4-라운드 CAST-128의 경우 5-라운드 계산 복잡도와 선택 평문 보다 작게 공격 가능하다. 이 공격은 고계 차분 분석과 연관 암호 공격을 결합한 고계 차분 연관 암호 공격이 된다.

VI. 결 론

연관 암호 공격은 두 연관 암호의 라운드 수의 차이가 커질수록 그대로 적용하기는 어렵다. 이러한 문제를 극복하기 위해, 본 논문에서는 차분 분석의 개념을 연관 암호 공격과 결합한 차분 연관 암호 공격을 소개하였다. 이 공격법을 가변 라운드 수를 갖는 블록 암호 알고리즘인 ARIA, SC2000, SAFER++ 등에 적용하여 분석하였다. 다음의 표는 본 논문의 결과를 요약한 것이다(ARIA v.1.0의 경우에는 본 공격은 적용되지 않는다).

표 1. 본 논문의 공격 결과

연관 암호[1] (암호(a)/암호(b))	라운드수[2] (c/d)	유사-동치 키 수	공격 복잡도[3]	공격유형[4] (암호(a)/암호(b))
ARIA(128) / ARIA(192)	10/12	2^{64}	$9RC/2^{64}T$	KP/CP
ARIA(192) / ARIA(256)	12/14	2^{64}	$9RC/2^{64}T$	KP/CP
SC2000(128) / SC2000(192)	6.5/7.5	2^{64}	$2RC/2^{33}T$	KP/CP
SC2000(128) / SC2000(256)	6.5/7.5	2^{128}	$2RC/2^{33}T$	KP/CP
SAFER++(128) / SAFER++(256)	7/10	2^{128}	$2^{81}RC/2^{101}E$	KP/CP
CAST-128(m) / CAST-128(n)	12/16	2^m	$2^{17}RC/2^{40}E$	CC/ACP

[1] a, b : 키 길이 ($40 \le m \le 80, 81 \le n \le 128$)
[2] c, d : 각 Cipher(a)와 Cipher(b)의 라운드 수
[3] RC : 연관 암호 쌍, T : 테이블 참조 연산, E: 암호화 연산
[4] KP : 기지 평문, (A)CP : (능동적) 선택 평문, CC : 기지 암호문

연관 암호 공격은 가변 라운드 수를 갖는 블록 암호 알고리즘의 키 스케줄 과정의 설계의 안전성에

커다란 의미가 있다. 아무리 암호화 및 복호화 알고리즘을 안전하게 설계한다 할지라도, 키 스케쥴 과정을 잘못 설계한다면 그 블록 암호는 취약한 알고리즘은 간주되어 실제적인 사용에 제약을 줄 수 있다. 따라서 암호화 및 복호화 알고리즘과 마찬가지로 키 스케쥴 과정도 동등한 안전성을 유지할 수 있도록 설계 시 주의를 기울여야 한다.

참 고 문 헌

[1] C. M. Adams, "The CAST-128 Encryption Algorithm," *Request for Comments (RFC) 2144*, Network Working Group, Internet Engineering Task Force, May 1997.

[2] P. S. L. M. Barreto and V. Rijmen, "The Khazad Legacy-level Block Cipher," *Primitive Submitted to NESSIE*, 2000.

[3] P. S. L. M. Barreto and V. Rijmen, "Anubis Block Cipher," *Primitive Submitted to NESSIE*, 2000.

[4] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," *Advances in Cryptology - CRYPTO'90*, LNCS 537, Springer-Verlag, pp. 2-21, 1991.

[5] E. Biham, "New Types of Cryptanalytic Attack Using Related Keys," *Journal of Cryptology*, Vol. 7, No. 4, pp. 156-171, 1994.

[6] A. Biryukov, J. Nakahara Jr, B. Preneel, and J. Vandewalle, "New Weak-Key Classes of IDEA," *Information and Communication Security: 4th International Conference (ICICS 2002)*, LNCS 2513, Springer-Verlag, pp. 315-326, 2002.

[7] A. Biryukov and D. Wagner, "Slide Attacks," *The 6th Fast Software Encryption(FSE 1999)*, LNCS 1636, Springer-Verlag, pp. 245-259, 1999.

[8] A. Biryukov and D. Wagner, "Advanced Slide Attacks," *Advances in Cryptology - EUROCRYPT 2000*, LNCS 1807, Springer-Verlag, pp. 589-606, 2000.

[9] A. Biryukov, "Analysis of Involutional Ciphers : Khazad and Anubis," *The 10th Fast Software Encryption(FSE 2003)*, LNCS 2887, Springer-Verlag, pp. 45-53, 2003.

[10] J. Daemen, R. Govaerts, and J. Vandewalle, "Weak Keys for IDEA," *Advances in Cryptoloty - CRYPTO'93*, LNCS 773, Springer-Verlag, pp. 224-231, 1994.

[11] P. Hawkes, "Differential-Linear Weak Key Classes of IDEA," *Advances in Cryptology - EUROCRYPT'98*, LNCS 1403, Springer-Verlag, pp. 112-126, 1998.

[12] J. Kelsey, B. Schneier, and D. Wagner, "Key-schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES," *Advances in Cryptology - CRYPTO'96*, LNCS 1109, Springer-Verlag, pp. 237-251, 1996.

[13] B. W. Koo, H. S. Jang, and J. H. Song, "Constructing and Cryptanalysis of a 16×16 Binary Matrix as a Diffusion Layer," *The 4th International Workshop on Information Security Applications (WISA 2003)*, LNCS 2908, Springer-Verlag, pp. 489-503, 2003.

[14] D. Kwon et al. "New Block Cipher : ARIA," *Pre-Proceedings of the 6th International Conference on Information Security and Cryptography (ICISC 2003)*, pp. 443-456, 2003.

[15] X. Lai, J. Massey, and S. Murphy, "Markov Ciphers and Differential Cryptanalysis," *Advances in Cryptology - EUROCRYPT'91*, LNCS 547, Springer-Verlag, pp. 17-38, 1991.

[16] J. L. Massey, G. H. Khachatrian, and M. K. Kuregian, "Nomination of SAFER++ as Candidate Algorithm for the NESSIE," *Primitive Submitted*

to NESSIE, 2000.

[17] L. May, M. Henricksen, W. Millian, G. Carter, and E. Dawson, "Strengthening the Key Schedule of the AES," *The 7th Australasian Conference on Information Security and Privacy (ACISP 2002)*, LNCS 2384, Springer-Verlag, pp. 226-240, 2002.

[18] S. Moriai, T. Shimoyama, and T. Kaneko, "Higher Order Differential Attack of a CAST Cipher," *The 5th Fast Software Encryption Workshop (FSE 1998)*, LNCS 1372, Springer-Verlag, pp. 17-31, 1998.

[19] J. Nakahara Jr, "Cryptanalysis and Design of Block Ciphers," *PhD thesis*, Katholieke Universiteit, Leuven, June 2003.

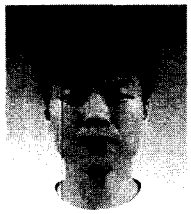
[20] National Institute of Standards and Technology, "Advanced Encryption Standard," *FIPS PUB 197*, 2001.

[21] T. Shimoyama, H. Yanami, K. Yokoyama, K. Ioth, J. Yajima, N. Toril, and H. Tanaka, "The Block Cipher SC2000," *The 8th Fast Software Encryption(FSE 2001)*, LNCS 2355, Springer-Verlag, pp. 312-327, 2001.

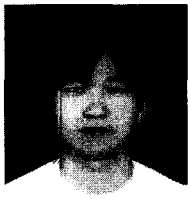
[22] H. Wu, "Related-Cipher Attacks," *Information and Communication Security : 4th International Conference(ICICS 2002)*, LNCS 2513, Springer-Verlag, pp. 447-455, 2002.

〈著者紹介〉

성 재 철 (Jaechul 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 전임강사
〈관심분야〉 대칭키 암호, 해쉬 함수, 메시지 인증 코드



김 종 성 (Jongsung Kim) 학생회원
2000년 8월 : 고려대학교 수학과 학사
2002년 8월 : 고려대학교 정보보호대학원 석사
2004년 8월 : 고려대학교 정보보호대학원 박사 수료
2002년 9월~현재 : 고려대학교 정보보호대학원 박사과정
〈관심분야〉 대칭키 암호알고리즘의 설계 및 분석



이 창 훈 (Changhoon Lee) 학생회원
2001년 2월 : 한양대학교 수학과 학사
2003년 2월 : 고려대학교 정보보호대학원 석사
2005년 2월 : 고려대학교 정보보호대학원 박사 수료
2003년 3월~현재 : 고려대학교 정보보호대학원 박사과정
〈관심분야〉 대칭키 암호알고리즘의 설계 및 분석