J Korea Industr Inf Syst Res Volume 17 Number 6
http://dx.doi.org/10.9723/jksiis.2012.17.6.011

이동 Ad-Hoc 네트워크 환경에 적합한 스트림 암호 HC-128의 부채널 안전성 분석†,‡

(Side-Channel Cryptanalysis on Stream Cipher HC-128 for Mobile Ad-Hoc Network Environments)

배 기 석*, 박 영 호**, 문 상 재***
(KiSeok Bae, YoungHo Park, and SangJae Moon)

요 약 최근 eSTREAM 공모사업에서 소프트웨어 분야로 최종 선정된 HC-128 알고리듬은 제한된 메모리 환경에서 고속 암호화가 가능하여 이동 ad-hoc 네트워크에 적합한 스트림 암호이다. 본 논문은 실제 구현되었을 때 발생할 수 있는 부채널 분석 공격에 대한 스트림 암호 HC-128 알고리듬의 안전성을 분석한다. 먼저 부채널 분석 공격에 대한 안전도가 낮은 것으로 판정하였던 기존 분석 방법의 누락된 부분을 밝히고, 올바른 분석 과정에서 필요한 계산 복잡도를 계산하여 HC-128 알고리듬의 부채널 분석 공격에 대한 안전성을 재평가하였다. 그 결과, 비밀키를 알아내기 위해서는 타 스트림 암호에 비해 훨씬 큰 약 2^{64}만큼의 복잡도가 필요하므로 스트림 암호 HC-128는 부채널 분석 공격에 안전한 것으로 평가된다.

핵심주제어 : 스트림 암호 HC-128, 부채널 분석 공격, 이동 ad-hoc 네트워크

Abstract The HC-128 stram cipher which selected for the final eSTREAM portfolio is suitable for mobile Ad-Hoc network environments because of the ability of high-speed encryption in restricted memory space. In this paper, we analyzed the vulnerability of side channel analysis attack on HC-128 stream cipher. At the first, we explain a flaw of previous theoretical analysis result which defined the complexity of side-channel attack of HC-128 stream cipher as 'low' and then re-evaluate the security against side-channel attack by estimating the concrete complexity for recovering the secret key. As a result, HC-128 stream cipher is relatively secure against side-channel attack since recovering the secret key have 2^{65} computation complexity which is higher than other stream cipher's one.

Key Words : HC-128 stream cipher, Side Channel Cryptanalysis, Mobile Ad-Hoc Network

† 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2012R1A1A4A01002603)
‡ 이 논문은 2012년도 경북대학교 학술연구비에 의하여 연구되었음
* 경북대학교 전자전기컴퓨터학과
** 경북대학교 산업전자공학과(교신저자, parkyh@knu.ac.kr)
*** 경북대학교 IT대학 전자공학부

1. 서 론

정보화 기술 및 정보화 기기의 발달로 언제라도 원하는 정보를 실시간으로 이용자 요구에 맞춘 형태로 제공할 수 있는 유비쿼터스 시대가 도래하고 있으며, 그에 따라 유비쿼터스 핵심기술 중 하나인 이동 ad-hoc 네트워크에 대한 관심이 높아지고 있다 [1,2]. 이동 ad-hoc 네트워크를 구성하는 노드는 특성상 제한된 메모리와 계산 능력을 고려하여 설계되는데, 이때 도청, 해킹, 가입자 비밀정보 유출, 서비스 도착상태(마비) 등과 같은 여러 형태의 공격에 대해 안전해야 한다. 따라서 이동 ad-hoc 네트워크 환경에 알맞은 고속의 암/복호화가 가능한 스트림 암호가 주목받고 있다.

부채널 분석 공격(side channel cryptanalysis)은 암호 알고리듬을 실제 암호 칩이나 암호 장치에 설계할 때 고려되지 못한 부가적인 정보의 누출에 의해 비밀정보를 알아내는 방식이다. 따라서 노드에 탑재되는 스트림 암호 역시 부채널 분석 공격의 대상이 될 수 있다 [3]. 최근 유럽연합(EU)에서 수행한 eSTREAM 공모사업에서 최종 선정된 알고리듬들도 부채널 분석 공격 가능성이 언급되었다 [4]. 이들 알고리듬들은 이론적 뿐만 아니라 실제 부채널 공격 실험을 수행한 결과도 소개되었다.

소프트웨어 분야에서 선정된 알고리듬중의 하나인 HC-128 암호 알고리듬은 고속 연산이 가능하여 암·복호의 실시간 처리가 필요한 이동 ad-hoc 네트워크 환경에 적합하고, DRM(digital rights management) 분야 등에서 주로 활용되고 있으나 [5], 부채널 분석 공격에 대한 안전성이 낮은 수준으로 평가되었다. 그러나 기 발표된 분석 결과[4]에서는 부채널 분석으로 추출한 값에서 실제 비밀 키 성분을 추출하는 과정이 누락되어 있다. 즉, 비밀 키 성분을 찾아내기 위한 모듈러 연산과 메시지 다이제스트 함수에 관해서 추가적으로 고려해야한다. 따라서 기존의 누락된 비밀 키를 찾기 위한 실질적인 계산 복잡도를 분석해야만 HC-128 알고리듬의 안전성을 정확히 판별할 수 있다.

본 논문에서는 스트림 암호 HC-128 알고리듬에 대한 기존 안전성 분석 방법에서 드러난 비밀 키 조사 과정에서의 문제점을 밝히고, 실질적인 계산 복잡도를 도출한다. 복잡도를 기반으로 비밀 키를 추출하는 시뮬레이션을 구현하여 검증한다. 도출된 계산 복잡도는 타 스트림 암호의 부채널 분석에 필요한 계산 복잡도와 비교하여 크고, 이론적인 공격의 복잡도와 비슷하여 부채널 공격의 실용성이 낮다. 이러한 분석을 통해 스트림 암호 HC-128의 부채널 분석 공격에 대한 견고성을 확인한다.

2. 스트림 암호와 부채널 분석

2.1 최종 선정된 스트림 암호

2008년 유럽연합은 안전한 스트림 암호의 분석과 설계능력을 향상하기 위해 ECRYPT의 프로젝트인 eSTREAM 공모사업을 수행하여 우수한 스트림 암호를 선정하였다. 구현 환경에 따라 소프트웨어와 하드웨어 분야로 선정하였으며 <표 1>은 최종 선정된 알고리듬을 보여주고 있다.

<표 1> eSTREAM의 최종 선정 알고리듬들

Profile 분야	알고리듬	비밀키	초기벡터
소프트웨어	Rabbit	128	64
	HC-128	128	128
	Salsa20/12	128/256	65
	Sosemanuk	128/256	64/128
하드웨어	Grain v1	128	96
	Mickey v2	128	64
	Trivium	80	80/64

2.2 스트림 암호 HC-128

스트림 암호 HC-128은 HC-256 버전의 단순화 버전으로 고속 처리가 가능한 소프트웨어용 암호 알고리듬이다. 유럽 연합에 의해 진행된 eSTREAM 공모전에서 소프트웨어부분에 최종 선정되었으며, 확장성과 고속의 암·복호화의 특성에 따라 이동 ad-hoc 네트워크 환경에 적합하고, DRM과 같은 다양한 분야에서 충분히 활용가능하다[5]. 128비트의 비밀 키 K와 초기벡터 IV(initial vector)를 이용하여 뛰어난 보안성을 가지는 랜덤 키 스트림(keystream)을 생성한다. 또한, 낮은 연산 복잡도에 의해서 저가의 칩에서도 활용가능하며, 키스트림 생성과정의 전반적인 연산이 병렬처리가 가능한 것이 특징이다. 사용된 연산은 아래와

같다.

$+ : x+y \bmod 2^{32}$
$\ominus : x-y \bmod 512$
$\oplus$: 배타논리합
$\gg$: 우측방향비트시프트
$\ll$: 좌측방향비트시프트
$\ggg$: 우측방향순환연산, $((x \gg n) \oplus (x \ll (32-n)))$
$\lll$: 좌측방향순환연산, $((x \ll n) \oplus (x \gg (32-n)))$

키스트림을 생성하기 위해서 P, Q 두 개의 테이블을 사용하며, 두 테이블은 초기화 과정을 통해서 생성된다. 초기화 과정에서는 128비트의 비밀 키 K와 초기 벡터 IV를 입력으로 하여 여섯 가지의 함수를 사용하여 P와 Q를 생성한다. 그 중 초기화 단계에서 사용하는 함수 $f_1(x)$와 $f_2(x)$는 랜덤성과 보안을 위해 SHA-256에서 사용하는 메시지 스케줄 함수이다[7]. 아래는 $f_1(x)$, $f_2(x)$를 포함한 여섯 함수이다.

$$f_1(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3),$$
$$f_2(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10)$$
$$g_1(x,y,z) = (x \ggg 10) \oplus (z \ggg 23) \oplus (y \ggg 8),$$
$$g_2(x,y,z) = (x \lll 10) \oplus (x \lll 23) \oplus (x \lll 8)$$
$$h_1(x) = Q[x_0] + Q[256 + x_2],$$
$$h_2(x) = P[x_0] + P[256 + x_2]$$

비밀 키와 IV를 사용하여 두 테이블을 만드는 실제 초기화 과정은 아래의 순서로 구성된다.

① 행렬 W 생성.

$$W_i = \begin{cases} K_i & 0 \le i \le 7 \\ IV_{i-8} & 8 \le i \le 15 \\ f_2(W_{i-2}) + W_{i-7} + f_1(W_{i-15}) + W_{i-16} + i & 16 \le i \le 1279 \end{cases} \quad (1)$$

② W를 통해 P, Q 생성

$$P[i] = W_{i+256} \quad \text{for } 0 \le i \le 512 \quad (2)$$
$$Q[i] = W_{i+768} \quad \text{for } 0 \le i \le 512$$

이후 P와 Q는 키스트림 생성 과정의 입력으로 사용된다. 총 1024 단계의 암호화를 통과한 출력 값들은 다시 P, Q 테이블로 인가되어 재사용된다.

for i=0 to 511

$$P[i] = (P[i] + g_1(P[i \bullet 3], P[i \bullet 10], P[i \bullet 511])) \oplus h_1(P[i \bullet 12]),$$
$$Q[i] = (Q[i] + g_2(Q[i \bullet 3], Q[i \bullet 10], Q[i \bullet 511])) \oplus h_2(Q[i \bullet 12])$$

본 논문에서는 비밀 키가 직접 사용되고 있는 키스트림 생성 이전의 초기화 과정에 대해서만 분석하므로 남은 과정은 생략한다.

2.3 스트림 암호에 대한 부채널 분석 결과

유비쿼터스 컴퓨팅을 구성하는 핵심 요소인 휴대용 기기들의 사용이 빈번해 짐에 따라 보안적인 요소가 더욱더 필요해지고 있다. 이때 암호 시스템의 탑재 과정에서 발생하는 취약점을 이용하는 부채널 분석이 실제 환경에서 발생 가능한 가장 큰 위협으로 떠오르고 있다. 최근 들어 고속 처리와 낮은 오버헤드의 특성에 따라 휴대용 기기의 보안성을 위한 스트림 암호의 사용이 대두되고 있다. 따라서 스트림 암호의 구현 시 부채널 분석 공격에 대한 안전성을 확보해야 한다[8].

부채널 분석은 1996년에 처음으로 DES에 대한 차분전력분석 [3]과 시차분석 공격으로 소개되었으며, 그 후 오류주입 공격 및 전자파분석 공격 등이 추가적으로 도입되었다. 대표적인 스트림 암호인 RC4에 대한 공격 [9]을 시작으로 유럽의 GSM 표준인 A5/1와 블루투스용 E0 알고리듬에 대한 부채널 분석 공격 [10]도 수행되었다.

2008년 eSTREAM 공모사업 중에는 후보군이었던 소프트웨어 분야 및 하드웨어 분야의 알고리듬에 대해서 부채널 분석 공격의 안전성이 Gierlichs 등에 의해서 평가되었다 [4]. 그 중 소프트웨어 분야의 최종 선정된 스트림 암호 알고리듬들의 부채널 안전성 평가 결과는 아래의 <표 2>와 같다. 이들 중 타 알고리듬에 비해 상대적으로 높게(medium) 평가된 Rabbit 알고리듬과 낮음(low)으로 판정된 Salsa20/12 알고리듬의 경우에는 데이터의 해밍무게(hamming weight)를 기반으로 하는 실제적인 부채널 분석 공격 실험이 수행된 결과도 최근 소개되었다 [11,12].

<표 2> 부채널 안전성 평가 결과

알고리듬 / 평가항목	Rabbit	HC-128	Salsa2 0/12	Sose-manuk
부채널 분석 공격 취약성	Yes	Yes	Yes	Yes
부채널 분석의 복잡성	**Medium**	**Low**	**Low**	**Low**
대응방법의 비용	High	High	High	High

3. 스트림 암호 HC-128에 대한 부채널 분석

3.1 기존 분석 방법

스트림 암호 HC-128는 기존 분석 [4]에 따르면 데이터의 해밍무게 유출에 의한 단순전력 분석과 차분전력 분석 공격이 가능한 것으로 나타났다. 공격의 대상이 되는 부분은 비밀 키 성분이 드러나는 초기화 과정이며 비밀 키를 찾아내는 부채널 분석의 과정은 다음과 같다.

먼저 수식 (1)에서 i=16인 경우의 W값은 아래와 같이 표현될 수 있다.

$$\begin{aligned} W_{16} &= f_2(W_{14}) + W_9 + f_1(W_1) + W_0 + 16 \quad (3)\\ &= f_2(IV_2) + IV_1 + f_1(K_1) + K_0 + 16\\ &= \Phi(IV) + \alpha \end{aligned}$$

여기서 $\Phi(IV)$와 α는 다음과 같다.

$$\Phi(IV) = f_2(IV_2) + IV_1 + 16, \quad (4)$$

$$\alpha = f_1(K_1) + K_0 . \quad (5)$$

초기 벡터인 IV를 재동기화 과정을 통해 고정된 값 또는 알고 있는 값으로 가정하였을 때 [13], 공격자는 알고 있는 $\Phi(IV)$를 제외한 나머지 비밀 키 성분 K_1과 K_0에 관한 추정으로 W_{16}의 해밍무게를 예측할 수 있다. 이 때 하나의 W_{16}에 대해서 가능한 키 조합 (K_1, K_0)은 $2^{32} \times 2^{32}$가지이므로 계산 복잡도가 높아 비효율적이다. 따라서 [4]에서는 이를 줄이기 위해서 추가적으로 다음의 방법을 소개하고 있다.

맨 처음 $\Phi(IV)$에 대해서 32비트가 모두 0이 되도록 초기벡터 값을 조작하였을 때, W_{16}의 해밍무게 값은 α의 해밍무게와 같게 된다. 이후 $\Phi(IV)$의 32비트 중 하나만 1이 되도록 IV를 조작하여 W_{16}의 해밍무게가 어느 위치에서 변화하는지 확인한다. 이 때 발생한 변화를 통해 해당 비트 위치의 α값을 확인할 수 있다. 이후 비트의 위치 변화에 따른 W_{16}의 해밍무게 변화를 확인하여 α의 전체 해밍무게 값을 찾아낸다.

이 과정에서 부채널 분석의 일종인 상관도 또는 차분전력분석 기법은 IV의 변화에 따른 W_{16}의 해밍무게를 찾아내는 데 사용된다. i개의 초기 벡터 IV_i를 입력으로 하여 HC-128 알고리듬을 수행할 때의 각 파형(P_i)를 수집하고, IV_i에 따른 W_{16}의 해밍무게 값들의 리스트(L_i)와의 상관도를 확인한다. 상관도 파형에서의 피크 성분의 존재 유무를 통해 추정한 α의 해밍무게가 올바른 것인지 아닌지를 확인할 수 있다. 최종적으로 공격자는 부채널 분석을 통해 α의 값을 찾아낼 수 있다.

기존의 공격 기법은 위의 방식으로 총 32번의 $\Phi(IV)$ 입력을 통해 두 개의 비밀 키 성분에 대한 후보 값들을 찾을 수 있다고 설명한다. 또한 그 후보 값들을 W_{17}, W_{18}, W_{19} 에 대해서 적용하게 되면 4개의 비밀 키 성분에 대한 수식 (5)와 아래의 3개 수식을 포함한 총 4개의 수식으로 가능한 비밀 키의 값을 찾을 수 있음을 설명하고 있다.

$$\beta = K_1 + f_1(K_2) \quad (6)$$

$$\gamma = K_2 + f_1(K_3) \quad (7)$$

$$\delta = K_3 + f_1(K_0) \quad (8)$$

여기서 β, γ, δ는 공격자가 $\Phi(IV)$를 조작한 후 부채널 분석 공격으로 검증한 값이며, 각 수식에 해당하는 키 쌍(k_i, k_{i+1})의 후보군을 추출하여 다음 수식에 인가하여 만족하는 후보군만 추려내는 방법으로 진행된다. 이 과정에서의 계산 복잡도는 $3*2^{32}$이며 최종적인 부채널 분석 공격의 계산 복잡도는 약 $32 + 2^{32} + 3*2^{32} (\geq 2^{34})$가 되므로 기존의 분석에서는 스트림 암호 HC-128의 안전도를 '낮다'로 판정하였다.

3.2 개선된 부채널 분석

기존 분석 결과 [4]의 문제점은 부채널 분석을 통해 획득한 값에서 비밀 키를 찾아내는 과정을 누락한 것이다. 부채널 분석 과정에서 $\Phi(IV)$의 조작을 통해서 찾아낸 해밍무게 값은 비밀 키 자체의 해밍무게가 아

닌 α의 값이다. 이 α값은 수식 (5)와 같이 두 비밀키 K_0와 K_1의 메시지 스케줄 함수와 모듈러 덧셈의 결과 값이다. 기존 분석 결과는 α의 추출 이후의 추가적인 설명 없이 비밀 키의 후보 값을 찾을 수 있다고 판단하여 α를 찾아내기 위한 계산 복잡도를 기준으로 부채널 분석의 계산 복잡도를 평가하였다.

실제로 비밀 키 쌍(K_i, K_{i+1})을 추출하기 위해서는 비트 순환과 비트 이동을 수행하는 함수 f_1의 역연산과 모듈러 덧셈을 함께 고려해야한다. 먼저 함수 f_1은 메시지 스케줄 함수로 그 역 연산이 어렵기 때문에

```
f1(x) : 메시지 스케줄 함수
List_k : 키 후보군의 리스트
Cand_k : 최종 키 후보군의 리스트
for k_a=0 to 2^32(=4,294,967,296)
  for k_b=0 to 2^32(=4,294,967,296)
   if( k_a+f1(k_b)==alpha)
     List_k0[n0++] =k_a
     List_k1[n1++] =k_b
   if( k_a+f1(k_b)==beta)
     List_k1_2[n2++] =k_a
     List_k2[n3++] =k_b
   if( k_a+f1(k_b)==gamma)
     List_k2_2[n4++] =k_a
     List_k3[n5++] =k_b
   if( k_a+f1(k_b)==alpha)
     List_k3_2[n6++] =k_a
     List_k0_2[n7++] =k_b
for i=0 to n0
  for j=0 to n7
   if(List_k0[i] == List_k0_2[j])
     cand_k0[c0++]=k0
for i=0 to n1
  for j=0 to n2
   if(List_k1[i] == List_k1_2[j])
     cand_k1[c1++]=k1
for i=0 to n3
  for j=0 to n4
   if(List_k2[i] == List_k2_2[j])
     cand_k2[c2++]=k2
for i=0 to n5
  for j=0 to n6
   if(List_k3[i] == List_k3_2[j])
     cand_k3[c3++]=k3
```

<그림 1> 비밀 키 추출 알고리듬

가능한 모든 입력을 시도해야하며 (2^{32}), 모듈러 덧셈의 경우에도 α를 만족하는 두 피연산자의 무수한 합동이 존재하기 때문에 모든 가능한 K_0를 시도해야 한다 (2^{32}). 따라서 α의 값으로부터 가능한 비밀 키 쌍 (K_i, K_{i+1})의 후보군을 찾기 위한 복잡도는 $2^{32} \times 2^{32}$가 된다.

즉, W_{16}에서 사용한 두 개의 비밀 키 성분을 찾기 위한 계산 복잡도는 α값을 찾기 위한 IV의 32번의 변화와 α값에서 비밀 키 쌍을 찾아내기 위한 복잡도 $2^{32} \times 2^{32}$의 합이 된다 $(32+2^{64})$. 나머지 W_{17}, W_{18}과 W_{19}의 경우에는 각 W값마다 존재하는 두 개의 비밀 키 중에서 이전 단계에서 먼저 추출한 비밀 키 후보군을 사용한다면 2^{32}의 복잡도만으로 남은 키의 후보군을 찾을 수 있다. 그 결과 128비트의 비밀 키 성분을 찾아내기 위한 부채널 분석 공격의 계산 복잡도는 기존의 결과와 달리 대략 $32+2^{64}+3*2^{32}(\geq 2^{64})$정도가 된다.

4. 부채널 안전성 분석 평가

4.1 시뮬레이션 결과

이론적으로 도출된 계산 복잡도를 확인하기 위해서 부채널 분석으로 찾아낸 α값을 이용하여 실제 비밀 키를 찾기 위한 시뮬레이션을 구현하였다. 사용된 알고리듬은 <그림 1>과 같다. 실제 시뮬레이션 결과, 최종 비밀 키의 후보군 값들은 다음의 <표 3>과 같다. 이는 $\alpha, \beta, \gamma, \delta$을 매번 랜덤한 값으로 설정하여 1000번을 수행하여 평균한 결과이다.

메시지 스케줄 함수와 모듈러 연산의 특성에 의해 추출된 비밀 키의 최종 후보군은 하나가 아닌 여러 개임을 확인할 수 있었다. 따라서 하나의 올바른 비밀 키 성분을 찾기 위한 추가적인 검증이 필요하게 된다. 검증 방식은 동일한 메시지를 사용하여 비밀 키 후보군들을 사용한 암호문과 정상 암호문의 비교로 적용한다. 이 때 최종 후보군에 대한 검증 과정 $(448*512*358*192 \geq 2^{32})$은 2^{64}보다 작기 때문에 전제 계산 복잡도는 변하지 않는다. 따라서 공격자는 부채널 공격을 통해 올바른 전체 비밀 키를 알아내기 위해서는 부채널 공격뿐만 아니라 비밀 키의 후보군을

추출하고 이를 검증하는 여러 단계를 필요로 함을 알 수 있었다.

<표 3> 시뮬레이션 결과

구분	K_0	K_1	K_2	K_3
최종 후보군수	448	512	358	192

4.2 타 스트림암호와의 안전도 비교

기존 분석 결과 [4]에서 언급된 다른 스트림 암호의 경우, Rabbit 알고리듬은 중간(medium), Salsa20/12 알고리듬은 낮음(low)의 판정을 보였다.

먼저 Rabbit 알고리듬에 대한 실제 공격 결과[11]에서는 4단계의 부채널 공격을 통해 비밀 키를 찾아낸다. 32비트 계수에 대한 8번의 부채널 공격, 1비트의 올림수, 32비트 계수에 대한 8번 공격, 그리고 마지막으로 1비트의 올림수를 찾는 방식으로 총 32*8+2+32*8+2($\geq 2^9$)의 계산 복잡도가 필요하다. 여기서 32비트 계수는 공격자가 알고 있는 값과 비밀 키의 합을 추측하기 때문에 2^{32}가 아닌 32번의 비트 이동과 유사하다.

다음으로 Salsa20/12의 경우 [12], 4x4 행렬 형태의 내부변수 중 세 번째 열에 속하는 32비트 내부 변수 중 하나의 갱신 전 후의 값을 찾아낸 후, 다른 하나를 찾아낸다. 다음으로 네 번째 열의 32비트 변수 3가지를 찾아내면 비밀 키를 복원해낼 수 있게 된다. 비밀 키를 찾아내기 위해서는 공격자는 32*2+32*2+32*2+32*2+32*2($\geq 2^8$)의 계산 복잡도가 필요하다. 복원 과정은 비트순환의 역을 계산하는 것으로 별도의 계산 복잡도는 없다.

반면 본 논문에서 확인된 HC-128의 부채널 분석 공격의 복잡도는 약 2^{64}정도로 상대적으로 높다는 것을 확인할 수 있다. 일반적인 128비트 비밀 키에 대한 전수조사의 복잡도는 2^{128}이며, 부채널 분석 공격을 적용하여 그 복잡도가 감소하더라도 여전히 절반의 복잡도를 필요로 한다. 이는 일반적인 이론적 공격의 복잡도와 유사하기 때문에 구현 환경을 구축해야 적용할 수 있는 부채널 분석 공격을 수행하는 실용성이 낮다. 또한, 기존 분석에서 동등하거나 보다 안전하다고 판정받은 eSTREAM 사업의 스트림 암호들에 비해서 상대적으로 높은 복잡도가 필요하기 때문에 스트림 암호 HC-128은 부채널 분석 공격에 대해서는 안전하다고 볼 수 있다.

5. 결 론

본 논문에서는 최근 유럽연합에 의한 스트림 암호 공모 사업에서 소프트웨어 분야에 최종 선정된 고속처리가 가능한 HC-128 알고리듬에 대해 부채널 분석 공격에 대한 안전성을 평가하였다. 기존의 분석 결과에서는 부채널 분석 이후 비밀 키 추출 과정을 누락하여 계산 복잡도 측정에 문제가 있었다. 정상적인 분석 결과, 부채널 공격을 통해 비밀 키를 알아내기 위한 계산 복잡도가 약 2^{64}으로 확인되었다. 이는 타 스트림 암호의 부채널 분석을 위한 계산 복잡도에 비해 훨씬 높으며, 이론적 공격의 복잡도와 차이가 없어 실용성이 떨어진다. 따라서 스트림 암호 HC-128은 부채널 공격에 대한 충분한 안전성을 가지고 있는 것으로 판단된다. 이러한 결과는 스트림 암호분야에 대한 부채널 분석 연구가 미진한 국내에 기여할 것으로 여겨진다.

참 고 문 헌

[1] Feng, Zhao, and Leonidas Guibas, “Wireless Sensor Networks,” Elsevier, 2004.

[2] 김시관, 신윤식, 임은기, “이동 임시 무선망에서의 키관리 기법에 관한 연구,” 한국산업정보학회논문지 Vol. 9, No. 4, pp.90-98, 2004.

[3] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” Advances in Cryptology, Proc.Crypto’ 99, pp.388-397, 1999.

[4] B. Gierlichs, L. Batina, C. Clavier, T. Eisenbarth, A. Gouget, Helena H, T. Kasper, K .Lemkerust, S. Mangard, A. Moradi and E. Oswald, “Susceptible of eSTREAM Candidates towards Side Channel Analysis,” Proc.SASC 2008 - Candidate of the Art of Stream Ciphers, 2008.

[5] 박준철, “HC-256 스트림 암호화를 이용한 범용성 및 확장성을 가진 DRM 기법 설계,” 한국통신학회

논문지, Vol. 32, No. 9, pp. 923-930, 2009.
[6] H. Wu, “The Stream Cipher HC-128,” Proc.New Stream Cipher Desings, pp.39-47, 2008.
[7] Natioanl Institute of Standars and Technology, “Secure Hash Standard (SHS),” FIPS PUB 180-4, 2012.
[8] 이훈재, 최희봉, 이상곤, “블록 형태 암호에서의 DPA 방어기술 연구,” 한국산업정보학회논문지, Vol. 7, No. 4, pp.1-8, 2002.
[9] C. Rechberger, E. Oswald, "Stream Ciphers and Side Channel analysis," In Proceedings of SASC 2004 - The State of the Art of Stream Ciphers 2004, pp.320-326, 2004.
[10] Keke Wu, Huiyun Li, Bo Peng, and Fengqi Yu,, “Correlation Power Analysis Attack against Synchronous Stream Ciphers," Proc.ICYCS'08, pp.2067-2072, 2008.
[11] 배기석, 안만기, 박제훈, 이훈재, 문상재, “스트림 암호 Rabbit에 대한 전력분석 공격,” 정보보호학회 논문지, Vol. 21, No. 3, pp. 27-36, 2011.
[12] K. S. Bae, M. K. Ahn, H. J. Lee, S. J. Moon, "Practical Side Channel Analysis Attacks on the Stream Cipher Salsa20/12," In Proceedings of ITC-CSCC 2011, pp. 835-838, 2011.
[13] J. Lano, N. Mentens, B. Prenell and I. Verbauwhede, “Power Analysis of Synchronous Stream Ciphers with Resynchronization Mechanism," The State of the Art of Stream Cipher, Proc. SASC'04, pp.327-333, 2004.

배 기 석 (KiSeok Bae)

- 2006년 8월 : 경북대학교 전자전기공학부 공학사
- 2008년 8월 : 경북대학교 전자공학과 공학석사
- 2009년 3월~현재 : 경북대학교 전자전기컴퓨터공학부 박사과정
- 관심분야 : 정보보호, 네트워크 보안, 스마트카드 보안

박 영 호 (YoungHo Park)

- 1989년 2월 : 경북대학교 전자공학과 학사
- 1991년 2월 : 경북대학교 전자공학과 석사
- 1995년 8월 : 경북대학교 전자공학과 박사
- 1996년~2008년 : 상주대학교 전자전기공학부 교수
- 2003년~2004년 : Oregon State Univ. 방문교수
- 2008년~현재 : 경북대학교 산업전자공학과 교수
- 관심분야 : 정보보호, 네트워크보안, 모바일 컴퓨팅

문 상 재 (SangJae Moon)

- 1972년 2월 : 서울대학교 공업교육(전자전공)과 학사
- 1974년 2월 : 서울대학교 전자공학과 석사
- 1984년 6월: 미국 UCLA 전기공학과 박사
- 1984년 7월~1985년 6월 : UCLA Postdoctor 근무
- 1984년 7월~1985년 6월 : 미국 OMNET 컨설턴트
- 1974년 12월~현재 : 경북대학교 IT대학 전자공학부 교수
- 2002년 2월~현재 : 한국정보보호학회 명예회장
- 관심분야 : 정보보호, 디지털 통신, 이동 네트워크

논 문 접 수 일 : 2012년 08월 17일
1차수정완료일 : 2012년 10월 16일
2차수정완료일 : 2012년 12월 02일
게 재 확 정 일 : 2012년 12월 02일