Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.5, Oct. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.5.1099

HyMES에 대한 결합 확률 분포 기반 단일 파형 분석*

박 병 규,† 김 수 리, 김 한 빛, 진 성 현, 김 희 석,‡ 홍 석 희
고려대학교

Single Trace Analysis against HyMES by Exploitation of Joint Distributions of Leakages*

ByeongGyu Park,† Suhri Kim, Hanbit Kim, Sunghyun Jin, HeeSeok Kim,‡
Seokhie Hong
Korea University

요 약

미래에 양자컴퓨터가 상용화될 것을 대비하여 전 세계적으로 양자컴퓨터에도 안전한 후 양자 암호(post quantum cryptography)에 대한 연구가 활발히 진행되고 있다. 그중 빠른 속도와 높은 안전성을 제공하는 부호 기반 암호시스템에 대하여 다양한 부채널 분석에 대한 취약점이 발견되고 있으며, 이에 따라 부채널 분석에 안전한 암호시스템 설계를 위한 연구가 진행되고 있다. 본 논문에서는 HyMES(Hybrid McEliece Scheme)에 대해 단 하나의 파형만을 이용하여 비밀키를 복원하는 방법을 제안한다. HyMES는 기존에 제안되었던 McEliece에 비해 키 크기가 작고, 암복호화 속도 또한 기존보다 빠르게 설계된 부호 기반 공개키 암호시스템이다. HyMES 복호화 알고리즘에는 신드롬 값 계산에 필요한 패리티 검사 행렬(parity-check matrix)을 연산하는 과정이 있다. 본 논문에서는 이 과정에서 사용되는 비선형 함수에 대한 결합 확률 분포가 비밀키 값에 따라 달라짐을 이용하여 HyMES를 분석하였다. 공개키 암호를 대상으로 한 결합 확률 분포 기반 분석은 본 논문에서 처음으로 제안되었다.

ABSTRACT

The field of post-quantum cryptography (PQC) is an active area of research as cryptographers look for public-key cryptosystems that can resist quantum adversaries. Among those categories in PQC, code-based cryptosystem provides high security along with efficiency. Recent works on code-based cryptosystems focus on the side-channel resistant implementation since previous works have indicated the possible side-channel vulnerabilities on existing algorithms. In this paper, we recovered the secret key in HyMES(Hybrid McEliece Scheme) using a single power consumption trace. HyMES is a variant of McEliece cryptosystem that provides smaller keys and faster encryption and decryption speed. During the decryption, the algorithm computes the parity-check matrix which is required when computing the syndrome. We analyzed HyMES using the fact that the joint distributions of nonlinear functions used in this process depend on the secret key. To the best of our knowledge, we were the first to propose the side-channel analysis based on joint distributions of leakages on public-key cryptosystem.

Keywords: HyMES, McEliece, Code based cryptosystem, Side-channel analysis, Joint distribution

Received(10. 1. 2018), Accepted(10. 10. 2018)
* 이 성과는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구 재단의 지원을 받아 수행된 연구임(No. NRF-2017R1C1B2004583).
† 주저자, darkadooa@naver.com
‡ 교신저자, 80khs@kore.ac.kr(Corresponding author)

I. 서 론

현재의 공인인증서나 전자서명 등에 사용되는 공개키 암호시스템인 RSA와 ECC는 각각 인수분해와 타원곡선 위에서의 이산 대수 문제의 어려움에 기반을 두고 있다. 하지만, 양자컴퓨터가 실용화되면 Shor 알고리즘에 의해 이 문제는 해결이 가능하게 되어 기존에 사용하던 RSA와 ECC는 더 이상 안전성을 보장할 수 없게 된다[18]. 이에 대한 대응책으로 양자컴퓨터로도 분석이 불가능한 후 양자 암호에 대한 연구가 전 세계적으로 활발히 진행되고 있다. 후 양자 암호는 다변수 함수 기반 암호, 래티스 기반 암호, 부호 기반 암호, 아이소제니 기반 암호 그리고 해시 기반 전자서명 알고리즘 등이 있다. 이러한 후 양자 암호들 중 부호 기반 암호는 암복호화 속도가 빠르기 때문에 활발한 연구가 진행되고 있으며, NIST(National Institute of Standards and Technology) 표준 공모에서도 래티스 암호와 더불어 가장 많은 암호시스템이 제안되었다.

1978년에 McEliece가 최초의 부호 기반 암호시스템인 McEliece를 제안하였다[2]. McEliece 암호시스템은 암호화 과정이 행렬 곱셈을 수행하는 단순한 구조로 구성되어 있으며, RSA와 래티스 기반 후 양자 암호인 NTRU에 비해 암복호화 속도가 빠르지만[14] 비교적 키 크기가 크다는 단점이 있다. 예를 들어 Goppa 부호를 기반으로 한 McEliece는 80-비트의 안전성을 갖기 위해 약 437KB의 매우 큰 공개키를 사용한다[16].

HyMES는 2008년 B.Biswas에 의해 제안된 McEliece 암호시스템의 변형이다[4]. HyMES는 암호화에 사용하는 공개키인 생성행렬(generator matrix)을 가우스 소거법이 적용된 기약 사다리꼴(reduced row echelon form)로 사용하기 때문에 기존의 McEliece에 비해 키 크기가 작고 암복호화 속도도 빠르다는 장점이 있다. Goppa 부호를 기반으로 한 HyMES는 80-비트의 안전성을 갖기 위해 약 63KB의 공개키를 사용하고 있어, 기존의 McEliece와 비교했을 때 약 7배 정도 크기를 감소할 수 있다.

McEliece 암호시스템이 수학적으로 안전하다고 하더라도 암호 장비에서 암호 알고리즘이 연산 되는 동안 발생하는 부가적인 정보인 시간, 소비전력, 전자기파 등을 이용하여 비밀키를 찾는 부채널 분석에 대해서는 취약성이 존재한다. Goppa 부호 기반 McEliece 암호시스템에 대한 부채널 분석으로 생성되는 오류비트에 의존해 복호화 과정에 소요되는 시간차 정보를 활용하는 시간차공격기법[11, 12]과 복호화 과정의 소비전력 파형을 이용한 SPA(Simple Power Analysis) 기법[5]이 제안되었다. 시간차 공격의 경우 [17]방법에 의해 대응이 가능하며, SPA 공격은 HyMES에는 적용이 불가능하다.

한편, 2013년에 Y.Linge가 비선형성을 띠는 함수의 입력과 출력의 결합 확률 분포가 비밀키 값에 따라 달라짐을 이용하여 비밀키를 찾는 부채널 분석 기법을 제안하였다[8]. 결합 확률 분포란 두 개 이상의 확률변수를 고려한 확률 분포이다. 결합 확률 분포를 이용한 분석은 대칭키 암호에서만 국한되어 연구되었으며, 여러 개의 파형을 이용하여 적용 가능하다[8, 10].

본 논문에서는 HyMES에 대하여 결합 확률 분포를 이용한 부채널 분석을 통해 비밀키를 찾는 방법을 제안한다. 제안하는 방법은 하나의 파형만을 이용하여 분석 가능하며, 최초의 공개키 암호를 대상으로 한 결합 확률 분포 기반 분석이다. 제안하는 분석방법을 통해 HyMES의 비밀키인 Goppa 다항식 $g(z)$와 서포트 L_{sec}를 모두 찾을 수 있다. 분석방법은 $g(z)$를 찾을 때 결합 확률 분포를 이용하며, L_{sec}을 찾을 때 수평 상관계수 전력분석(horizontal correlation analysis)을 이용한다[9]. 본 논문에서는 시뮬레이션 파형을 이용한 실험을 통해 제안하는 분석방법이 올바르게 동작함을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서 분석을 이해하기 위한 관련 지식들을 설명하고, 3장에서 제안하는 분석방법에 대해 자세히 설명한다. 4장에서는 3장에서 설명한 분석방법을 적용한 시뮬레이션 결과를 보이고 마지막으로 5장에서 결론을 맺는다.

II. 관련 연구

2.1 Goppa 부호 기반 Classical McEliece

부호 이론은 고전 통신 채널에서 메시지에 오류가 발생하였을 때 수신자가 메시지의 오류를 정정하여 올바른 메시지를 복구할 수 있도록 만든 방법이다. 송신자가 원래의 메시지를 부호화하여 전달하면, 수신자는 부호화된 메시지에 발생한 오류를 정정할 수 있다. McEliece 암호시스템은 1978년 McEliece

에 의해 제안된 Goppa 부호 기반 공개키 암호시스템이다[2]. McEliece의 암호화, 복호화, 키 생성과정은 알고리즘 1., 2., 3.과 같다. 알고리즘 3. Classical McEliece 키 생성과정의 단계 1에서 Goppa 다항식 $g(z)$는 Goppa 부호의 최소 거리가 $d_c \le 2t+1$를 만족하도록 선택되어야 한다. 단계 2의 부호 C와 대응하는 생성행렬 G는 C를 생성하는 기저(basis)들로 이루어진 행렬이다. 따라서, k는 부호 C의 차원(dimension)을 의미한다. 생성행렬 G와 패리티 검사 행렬 H는 정의 1.과 같다.

정의 1. $k\times n$ 행렬 G의 행벡터 공간이 유한체 F위에서의 C를 생성하고, 행벡터 공간의 차원이 k라면, 행렬 G를 F위에서의 $[n,k]$-C의 생성행렬이라고 한다. $(n-k)\times n$ 행렬 H에 대해서 H^T의 열벡터들이 G의 영 공간의 기저라면, H를 부호 C의 패리티 검사 행렬이라고 한다.

McEliece는 Goppa 부호를 기반으로 하고 있기 때문에 복호화 과정에서 디코딩 알고리즘으로 Patterson 알고리즘이 일반적으로 사용된다[3]. 본 논문에서는 유한체 F_2 위에서의 이진(binary) Goppa 부호에 관한 기본적인 내용만 언급한다.

정의 2. m,t가 양의 정수일 때, Goppa 다항식 $g(z)$와 서포트 L는 다음과 같다.

$$g(z)=\sum_{i=0}^{t} g_i z^i \in F_{2^m}[z] \tag{1}$$

$$L=\{\alpha_0,..,\alpha_{n-1}\}\in F_{2^m}^n, g(\alpha_j)\neq 0, \quad \forall 0\le j\le n-1 \tag{2}$$

Goppa 부호 디코딩에 필요한 신드롬 $S_{\hat{c}}$은 정의 3.와 같으며, 이진 $[n,k,d_c]$-Goppa 부호 C를 식 (4)와 같이 신드롬 $S_{\hat{c}}$가 0이 되는 모든 $\hat{c}$의 집합으로 정의한다. 정의 1.과 같이 패리티 검사 행렬 H는 C의 영 공간(null space)을 생성하므로, 신드롬 $S_{\hat{c}}$로부터 식 (5)의 H를 이끌어낼 수 있다.

정의 3. Goppa 다항식이 $g(z)$이고, 서포트가 $L=\{\alpha_0,..,\alpha_{n-1}\}$일 때, 벡터 $\hat{c}=(c_0,..,c_{n-1})\in F_{2^m}^n$에 대한 신드롬 $S_{\hat{c}}(z)$은 다음과 같다.

$$\begin{aligned} S_{\hat{c}}(z) &= -\sum_{i=0}^{n-1}\frac{\hat{c}_i}{g(\alpha_i)}\frac{g(z)-g(\alpha_i)}{z-\alpha_i} \\ &\equiv \sum_{i=0}^{n-1}\frac{\hat{c}_i}{z-\alpha_i} \bmod g(z) \end{aligned} \tag{3}$$

$$Goppa(L,g(z)) := \left\{\hat{c}\in F_{2^m}^n \mid S_{\hat{c}}(z)=\sum_{i=0}^{n-1}\frac{\hat{c}_i}{z-\alpha_i}\equiv 0 \bmod g(z)\right\} \tag{4}$$

알고리즘 1. Classical McEliece : 암호화

입력: 공개키 $K_{pub}=(\hat{G})$, 메시지 M
출력: 암호문 c
1. 메시지 M을 k-비트 벡터 m으로 표현한다.
2. 랜덤한 n-비트 에러 벡터 e를 선택한다. $HW(e)\le t$
3. return $c=m\times\hat{G}+e$

알고리즘 2. Classical McEliece : 복호화

입력: n-비트 암호문 c, 개인키 $K_{\text{sec}}=(G,S^{-1},P^{-1})$
출력: 메시지 M
1. $\hat{c}=c\times P^{-1}$를 계산한다.
2. $\hat{c}$에 대한 신드롬 $S_{\hat{c}}$를 계산한다.
3. 디코딩 알고리즘 $D_{Goppa}(\hat{c})$를 이용하여 $S_{\hat{c}}$로부터 k-비트 $\hat{m}=m\times S$를 얻는다.
4. $m=\hat{m}\times S^{-1}$를 계산한다.
5. m을 메시지 M으로 표현한다.
6. return M

알고리즘 3. Classical McEliece : 키 생성

입력: 고정된 파라미터 t,n,m
출력: 개인키 K_{sec}, 공개키 K_{pub}
1. t개의 에러를 정정할 수 있는 이진 $[n,k,d_c]$-Goppa 부호 C를 선택한다.$(g(z),L$ 선택$)$
2. 부호 C와 대응하는 $k\times n$ 생성행렬 G를 계산한다.
3. 랜덤한 $k\times k$ 가역행렬 S를 선택한다.
4. 랜덤한 $n\times n$ 순열행렬 P를 선택한다.
5. $k\times n$ 행렬 $\hat{G}=S\times G\times P$를 계산한다.
6. S와 P의 역행렬을 계산한다.
7. return $K_{\text{sec}}=(G,S^{-1},P^{-1})$, $K_{pub}=(\hat{G})$

$$H=\begin{pmatrix} \frac{g_t}{g(\alpha_0)} & \frac{g_t}{g(\alpha_1)} & \cdots & \frac{g_t}{g(\alpha_{n-1})} \\ \frac{g_t\alpha_0+g_{t-1}}{g(\alpha_0)} & \frac{g_t\alpha_0+g_{t-1}}{g(\alpha_1)} & \cdots & \frac{g_t\alpha_0+g_{t-1}}{g(\alpha_{n-1})} \\ \cdots & \cdots & & \cdots \\ \frac{g_t\alpha_0^{t-1}+\ldots+g_2\alpha+g_1}{g(\alpha_0)} & \frac{g_t\alpha_0^{t-1}+\ldots+g_2\alpha+g_1}{g(\alpha_1)} & \cdots & \frac{g_t\alpha_0^{t-1}+\ldots+g_2\alpha+g_1}{g(\alpha_{n-1})} \end{pmatrix} \quad (5)$$

식 (5)의 H 값을 식 (6)과 같이 간소화할 수 있으며, H_g의 행렬식이 0이 아니기 때문에 패리티 검사 행렬로 $\hat{H}$를 사용할 수 있다.

$$H=\begin{pmatrix} g_t & 0 \ldots 0 \\ g_{t-1} & g_t \ldots 0 \\ \ldots & \ldots \ldots \\ g_1 & g_2 \ldots g_t \end{pmatrix} * \begin{pmatrix} \frac{1}{g(\alpha_0)} & \frac{1}{g(\alpha_1)} & \cdots & \frac{1}{g(\alpha_{n-1})} \\ \frac{\alpha_0}{g(\alpha_0)} & \frac{\alpha_1}{g(\alpha_1)} & \cdots & \frac{\alpha_{n-1}}{g(\alpha_{n-1})} \\ \cdots & \cdots & & \cdots \\ \frac{\alpha_0^{t-1}}{g(\alpha_0)} & \frac{\alpha_1^{t-1}}{g(\alpha_1)} & \cdots & \frac{\alpha_{n-1}^{t-1}}{g(\alpha_{n-1})} \end{pmatrix} = H_g \times \hat{H} \quad (6)$$

2.2 HyMES (Hybrid McEliece Scheme)

2008년에 프랑스 INRIA 연구소의 SECRET 팀 소속인 B.Biswas와 N.Sendrier가 McEliece 암호시스템의 변형인 HyMES를 제안하고 구현하였다[4]. HyMES는 McEliece에 비해 키 크기가 작고, 암복화 속도도 빠르다는 장점이 있다. 일반적인 HyMES의 암호화, 복호화, 키 생성 알고리즘은 각각 알고리즘 4., 알고리즘 5., 알고리즘 6.과 같다. 알고리즘 6.의 단계 1에서 Goppa 다항식을 기약다항식으로 선택하는데, 이 경우에 부호의 최소 거리가 $d_c \le 2t+1$를 만족하기 때문에 정당한 사용자는 최대 t개의 오류를 정정할 수 있다. 단계 2에서는 서포트의 개수 n을 $GF(2^m)$의 모든 원소의 개수인 2^m으로 사용하며, 서포트를 선택할 때 0부터 2^m-1까지 순서대로 선택한 후에 랜덤하게 섞어서 사용한다. HyMES의 공개키 크기는 $k\times(n-k)$로, 공개키 크기가 $k\times n$인 기존의 McEliece에 비해 $k\times k$만큼 감소한다. 알고리즘 4.의 단계 3에서 m과 G_{sys}의 행렬 곱셈 연산을 할 때 단위행렬 I_k와의 곱셈이 필요 없으므로, HyMES의 암호화 속도 또한 기존의 McEliece보다 빠르다.

2010년에 기존의 McEliece 복호화 과정에서 신드롬 $S_{\hat{c}}$를 계산하는 부분을 대상으로 SPA를 하여 P^{-1}을 찾아내고 순차적으로 S^{-1}, G를 찾아내는 방법이 제안되었다[5]. 위 논문에서 제안된 분석방법은 신드롬 $S_{\hat{c}}$을 계산하는 단계 이전에 P^{-1}를 곱하는 과정으로 인해 발생하는 취약점을 이용하였다. 그러나 HyMES의 복호화 과정인 알고리즘 5.에서는 신드롬을 계산하기 전에 P^{-1}를 곱해주지 않기 때문에, [5]에서 제안된 분석방법은 HyMES의 복호화 과정에 적용되기 어렵다.

본 논문에서는 알고리즘 5.의 단계 1에서 신드롬 $S_c=(H\hat{P})c^T$ 연산에 필요한 $H\hat{P}$를 계산하는 알고리즘의 구현상의 취약점을 이용하여 HyMES를 분석하였다. HyMES 구현의 $H\hat{P}$의 열벡터들을 구하는 과정은 알고리즘 7.과 같다[4]. 분석방법에 대한 설명은 3장에서 서술한다.

알고리즘 7.은 순열 행렬 $\hat{P}$로 치환된 L_{sec}를 이용하여 $H\hat{P}$를 구하는 과정으로, 길이가 mt-비트인 h_i를 $GF[2^m]$위에서의 $t-1$차 다항식으로 표현하면 식 (7)과 같다.

$$h_i = h_i[t-1]z^{t-1}+h_i[t-2]z^{t-2}+..+h_i[0] \quad (7)$$

알고리즘 4. HyMES : 암호화

입력: 공개키 $K_{pub}= G_{sys}=(I_k|Q)$, 메시지 M
출력: 암호문 c
1. 메시지 M을 k-비트 벡터 m으로 표현한다.
2. 랜덤한 n-비트 에러 벡터 e를 선택한다. $HW(e) \le t$
3. return $c= m\times G_{sys}+e$

알고리즘 5. HyMES : 복호화

입력: n-비트 암호문 c, 개인키 $K_{\text{sec}}=(g(z), L_{\text{sec}})$
출력: 메시지 M
1. c에 대한 신드롬 S_c를 계산한다. $S_c= (H\hat{P})c^T$
2. 디코딩 알고리즘 $D_{Goppa}(c)$를 이용하여 S_c로부터 k-비트 $\hat{m}= m\times S\times G\times P= mG_{sys}= m(I_k|Q)=(m|mQ)$를 얻는다.
3. $\hat{m}$의 상위 k-비트로부터 m을 얻는다.
4. m을 메시지 M으로 표현한다.
5. return M

알고리즘 6. HyMES : 키 생성

입력: 고정된 파라미터 t,n,m

출력: 개인키 K_{sec}, 공개키 K_{pub}

1. 랜덤한 일 계수 기약다항식 $g(z)$를 Goppa 다항식으로 선택한다.
   $g(z)=z^t+c_{t-1}z^{t-1}+..+c_1z+c_0$
   $\deg(g(z))=t,\ c_i\in GF(2^m)$
2. n개의 서포트 α_i를 선택한다.
   $L=\{\alpha_0,..,\alpha_{n-1}\},\ \alpha_i\in GF(2^m),\ g(\alpha_i)\neq 0$
3. 패리티 검사 행렬 $\hat{H}$를 계산한다.
4. 패리티 검사 행렬을 $\hat{H}_{sys}=\hat{S}\hat{H}\hat{P}=(Q^T|I_{n-k})$의 형태로 변환한다.
5. 패리티 검사 행렬 $\hat{H}$가 $(Q^T|I_k)$의 형태로 변환되지 않는다면 단계 1로 돌아간다.
6. 순열 행렬 $\hat{P}$가 단위행렬이 아니라면 $L_{\mathrm{sec}}=L\hat{P}$ 다음과 같이 순열 행렬 $\hat{P}$로 서포트 L을 전치 시켜준다.
7. $G_{sys}=(I_k|Q)=SGP$ 를 계산한다.
8. return
   $K_{\mathrm{sec}}=(g(z),L_{\mathrm{sec}}),\ K_{pub}=G_{sys}=(I_k|Q)$

알고리즘 7. HyMES 구현 : $H\hat{P}$ 계산[4]

입력: $g(z)=z^t+g_{t-1}z^{t-1}+..+g_0$
, $L_{\mathrm{sec}}=\{\alpha_0,..,\alpha_{n-1}\},\ n=2^m$

출력: $H\hat{P}$의 열벡터 $h_i,\ i=0,..,n-1$

```
for i = 0 : n-1
  h_i[t-1] = 1
    for j = t-2 : 0
      h_i[j] = g_{j+1} ⊕ (α_i × h_i[j+1])
    end
  a = g_0 ⊕ (α_i × h_i[0])
    for j = 0 : t-1
      h_i[j] = h_i[j] / a
    end
end
return h_i, i = 0,..,n-1
```

HyMES를 구현할 때, 연산 시간 단축을 위해 $H\hat{P}$의 열벡터 $h_i,\ i=0,...,n-1$를 키 생성 과정에서 사전 연산하여 비밀키로 저장하여 사용하거나 이를 복호화 과정에서 직접 계산할 수 있다. [4]에서는 HyMES의 $H\hat{P}$의 열벡터들을 키 생성 과정에서 미리 계산하여 비밀키의 일부로 저장하여 사용하는 방식으로 구현하였다.

2.3 결합 확률 분포 기반 부채널 분석

소비전력을 이용한 부채널 분석은 1999년 Kocher에 의해 제안된 차분 전력분석(differential power analysis)[1]을 시작으로, 2003년 S.Chari이 제안한 템플릿 공격(template attack)[6], 2004년 K.Schramm이 제안한 충돌 공격(collision attack)[7]등 다양한 공격기법이 제안되었으며 지속적으로 연구가 진행되고 있다. 2014년에 Y.Linge가 처음으로 AES의 비선형 함수인 S-box에 대하여 키 값에 따라 S-box 입력과 출력 해밍 웨이트의 결합 확률 분포 차이가 발생함을 이용하여 AES의 비밀키를 찾는 방법을 제안하였다[8]. 해당 분석방법을 이용하면, 공격자가 평문 쌍들의 값을 모른다는 강력한 가정하에 비밀키를 찾을 수 있으며, 차분 전력분석의 대응 기법인 마스킹이 적용된 대칭키 암호 대하여 모든 라운드의 키 값을 찾을 수 있다는 장점이 있다.

Y.Linge가 제안한 분석방법은 알고리즘 8.과 같다. g 함수는 키 k와 입력 데이터 a를 인자로 받아 b를 출력하는 함수이고 ϕ는 입력의 해밍 웨이트를 출력하는 함수이다. [8]에서는 함수 g를 AES S-box로 선택하였다. 본 예시에서는 AES의 1라운드 SubBytes에서 사용하는 S-box를 기준으로 고격을 설명하도록 한다.

먼저, 알고리즘 8.의 단계 2-7은 크기가 1-바이트인 키 $k'=0\sim255$에 대하여 이론적 결합 확률 분포표를 만드는 과정으로 평문 $a=0\sim255$와 S-box 출력 b의 해밍 웨이트 값을 확률변수로 사용한다. $S(g,k)$는 함수 g에 대하여 키가 k일 때 $\phi(a)$와 $\phi(b)$의 이론적 결합 확률 분포 표이다. 1-바이트 평문 a와 S-box 출력 b가 가질 수 있는 해밍 웨이트의 범위는 0~8이기 때문에 $S(g,k)$의 크기는 9×9이다. 다음으로, 단계 9-11은 실제 사용한 평문 a와 출력 b의 해밍 웨이트 추측 값인 (a_i,b_i)쌍을 이용해 추정된 결합 확률 분포표를 구성하는 과정이다. 마지막으로, 단계 13-18은 추정된 분포와 각 키에 대한 이론적 분포 간의 유사도를 구하여 유사도가 가장 높은 키를 실제 키로 추측하는 과정이다.

추정된 결합 확률 분포표를 구성하기 위해 수집된 소비전력 파형으로부터 평문 a와 출력 b의 해밍 웨이트 값을 추측해야 한다. 분석 성능을 위해 해밍 웨이트 값을 정확히 추측해야 하지만, 실제로는 노이즈 신호로 인해 해밍 웨이트 값을 정확히 추측하는 것은

알고리즘 8. 결합 확률 분포를 이용한 키 추측 〔8〕

입력: 추측된 해밍 웨이트 쌍
(a_i,b_i), $i=0,..,M-1$
출력: 키 key
1. $g: K\times A\rightarrow B$, $N=|K|$,
$S(g,k),S_d\in\{0,..,n\}\times\{0,..,m\}$
2. for $k\in K$ ←이론적 분포 계산
3. $S(g,k)\leftarrow 0$
4. for $a\in A$
5. $S(g,k)(\phi(a),\phi(g(a,k)))$
$\leftarrow S(g,k)(\phi(a),\phi(g(a,k)))+\frac{1}{|A|}$
6. end
7. end
8. $S_d\leftarrow 0$
9. for $i=0:M-1$ ←추정된 분포 계산
10. $S_d(a_i,b_i)\leftarrow S_d(a_i,b_i)+\frac{1}{M}$
11. end
12. $key\leftarrow 0$
13. for $k\in K$ ←이론적 분포와 추정된 분포 간의 비교
14. if $d(S(g,k),S_d)<d(S(g,key),S_d)$
15. $key\leftarrow k$
16. end
17. end
18. return key

매우 어렵다. 2017년에 노이즈의 영향을 확률로 반영하는 최대 가능도(maximum likelihood) 방법을 이용하여 이전보다 개선된 방식의 논문이 제안되었다〔10〕.

결합 확률 분포를 이용한 분석은 AES의 S-box가 충분히 비선형성을 띠고 있기 때문에 키에 따른 결합 확률 분포의 차이가 발생하여 가능한 공격이며, 이 분석은 대칭키 암호에 국한되어 연구되어왔다.

III. 결합 확률 분포를 이용한 HyMES 분석

본 장에서는 〔4〕에서 제안된 HyMES 구현에 대해 결합 확률 분포를 이용한 분석방법을 서술한다. 제안하는 분석방법은 복호화 과정에서 신드롬 값 계산을 위해 필요한 $\hat{HP}$의 열벡터 h_i, $i=0,...,n-1$를 계산하는 알고리즘 7.를 대상으로 분석을 하였다. 〔4〕에서 제안된 HyMES의 구현은 신드롬 연산에 필요한 h_i를 키 생성 시에 사전연산하여 사용하기 때문에 키 생성 알고리즘에 h_i를 구하는 과정이 포함되어있다. 만약, 사전 연산을 하지 않는 경우라면 복호화 알고리즘에 h_i를 구하는 과정이 따로 필요하다. 따라서 사전 연산 여부와 관계없이, 알고리즘 7.과 같은 방식으로 h_i를 계산하는 구현은 본 논문에서 제안하는 분석방법에 취약하다.

제안하는 분석방법은 크게 두 가지 단계로 나뉜다. 먼저 결합 확률 분포를 이용하여 비밀키 중 하나인 Goppa 다항식 $g(z)$를 찾고, 알아낸 $g(z)$의 값을 이용한 수평 상관계수 전력분석으로 나머지 비밀키인 서포트 L_{sec}를 찾는다. 3.1, 3.2절에서 $g(z)$와 L_{sec}를 찾는 방법을 자세히 설명하도록 한다.

3.1 $g(z)$ 분석

알고리즘 7.에서 사용되는 곱셈과 나눗셈은 $GF(2^m)$위에서의 연산이다. 일반적으로 $GF(2^m)$위에서의 곱셈과 나눗셈 연산은 속도의 효율성을 위해 지수연산 테이블과 로그 테이블을 이용하여 연산한다. 지수연산 테이블은 $GF(2^m)$를 구성하는 원시다항식 $p(x)$의 근 α에 대해서 양의 정수 i의 값을 받아 α^i을 반환하도록 구성하고, 로그 테이블은 $GF(2^m)$의 원소를 α^i으로 표현하였을 때 i를 반환하도록 구성한다. 〔4〕에서는 곱셈 또는 나눗셈을 두 개의 피연산자에 대해 로그 테이블을 참조하여 덧셈 또는 뺄셈을 하고 감산 연산한 결과에 대해 지수연산 테이블을 참조하는 방식으로 구현하였다. 이는 피연산자끼리 직접 곱셈 또는 나눗셈을 하여 $p(x)$로 감산하는 방식보다 연산속도 면에서 훨씬 효율적이다. HyMES의 $GF(2^m)$위에서 곱셈과 나눗셈 구현은 알고리즘 9., 10.과 같다.

본 절에서는 결합 확률 분포를 이용하여 비밀키 $g(z)$를 찾는 분석방법을 설명한다. $g(z)$ 전체를 찾는 과정은 알고리즘 11.과 같다. 함수 $l(k,a)$은 $GF(2^m)$의 원소 k,a를 입력으로 받아 $\log[k\oplus a]$를 반환하는 함수이고, 함수 $\phi(v)$는 v를 입력으로 받아 v의 해밍 웨이트를 반환한다. a_i,b_i는 함수 $l(k,a)$의 입력 a와 출력 b의 해밍 웨이트 추측 값이다. $g(z)$를 찾기 위해서는 $g(z)$의 모든 차항의 계수 $g_{t-1},g_{t-2},..,g_0$를 찾아야 하며, 각 차항의 계수를 분석하는 방법이 g_{t-1}을 찾는 방법과 동일하기 때문에 본 논문에서는 g_{t-1}을 찾는 방법에 대해서만 설

명하도록 한다.

알고리즘 7.의 단계 3 반복문에서 $j=t-2, t-3$ 인 경우의 연산을 $i=0:2^m-1$ 에 대해 모두 나열하면 다음과 같다.

$$\begin{aligned}
&h_0[t-2]=g_{t-1}\oplus(\alpha_0\times h_0[t-1])\\
&h_0[t-3]=g_{t-2}\oplus(\alpha_0\times h_0[t-2])\\
&h_1[t-2]=g_{t-1}\oplus(\alpha_1\times h_1[t-1])\\
&h_1[t-3]=g_{t-2}\oplus(\alpha_1\times h_1[t-2])\\
&\cdots\\
&h_{2^m-1}[t-2]=g_{t-1}\oplus(\alpha_{2^m-1}\times h_{2^m-1}[t-1])\\
&h_{2^m-1}[t-3]=g_{t-2}\oplus(\alpha_{2^m-1}\times h_{2^m-1}[t-2])
\end{aligned} \tag{8}$$

식 (8)에서 $GF(2^m)$위에서의 곱셈 연산 $\alpha_i\times h_i[t-2]$는 실제로 식 (9)와 같이 사전 연산 테이블을 이용한 방법으로 이루어진다.

$$\exp[\mathrm{mod}(\log[\alpha_i]+\log[g_{t-1}\oplus(\alpha_i\times 1)])] \tag{9}$$

g_{t-1}을 찾기 위해서 식 (9)의 두 번째 로그 테이블 연산부인 $\log[g_{t-1}\oplus(\alpha_i\times 1)]$를 결합 확률 분포 분석에 이용한다. 먼저, α_i와 $\log[g_{t-1}\oplus\alpha_i]$의 해밍 웨이트를 확률변수로 두고, 2^m개의 키 후보군 g_{t-1}' 각각에 대해서 이론적 결합 확률 분포표를 구성한다. $GF(2^m)$의 원소인 α_i와 $\log[g_{t-1}\oplus\alpha_i]$가 가질 수 있는 해밍 웨이트 값의 범위는 0~m이며, 이론적 결합 확률 분포표 $S(l,g_{t-1}')$의 크기는 $(m+1)\times(m+1)$이다. 다음으로, 알고리즘 7.의 소비전력 파형에서 α_i와 $\log[g_{t-1}\oplus\alpha_i]$이 연산 되는 부분의 시점을 찾는다. 본 논문에서 소비전력 모델은 식 (10)과 같은 해밍 웨이트 모델을 따르며, 모든 시점에서의 α, β는 같다고 가정한다.

$$P_v=\alpha HW(v)+\beta+w \tag{10}$$

두 시점으로부터 α_i가 처리될 때 발생하는 소비전력 값 P_{α_i}과 $\log[g_{t-1}\oplus\alpha_i]$가 처리될 때 발생하는 소비전력 값 $P_{\log[g_{t-1}\oplus\alpha_i]}$를 얻을 수 있고, 각 시점에 대한 소비전력 값으로부터 α_i과 $\log[g_{t-1}\oplus\alpha_i]$의 해밍 웨이트 값을 추측하여 추정된 결합 확률 분포 표를 만든다. α_i와 $\log[g_{t-1}\oplus\alpha_i]$의 해밍 웨이트 값을 추측하기 위해 사용한 방법은 Linge's slice method 이다[8]. Slice method는 해밍 웨이트 값과 소비전력 값이 비례관계임을 이용한다. 예를 들면 균등 분포를 따르는 N-비트 데이터 M개 중 해밍 웨이트가 p인 데이터의 개수는 약 $\frac{M\times C_N^p}{2^N}$임을 알 수 있고, M개의 데이터에 대한 소비전력 값들을 오름차순으로 정렬하였을 때 각 소비전력 값의 위치에 따라 해당 데이터의 해밍 웨이트 추측이 가능하다. Slice method의 성능은 실제로 사용된 데이터들의 값이 균일하게 분포되었을 때 높아지기 때문에 최대한 많은 데이터를 이용해야 한다. 마지막으로, 이론적 확률 분포와 추정된 확률 분포 간의 유사도를 비교하였을 때 가장 유사도가 높은 g_{t-1}'을 실제 키 g_{t-1}로 선택한다. 본 논문에서는 분포 간의 유사도 비교방법으로 Harmonic Mean Distance, Inner

알고리즘 9. HyMES 구현 : $GF(2^m)$위에서의 곱셈[4]

입력: $a,b\in GF(2^m)$, $\exp[i]=\alpha^i$, $\log[\alpha^i]=i$, $i=0,..,2^m-1$
$\mathrm{mod}(d)=((d)\ \&(2^m-1))+((d)\gg m)$
출력: $a\times b\in GF(2^m)$
1. if a or $b=0$
2. $a\times b\leftarrow 0$
3. else
4. $a\times b\leftarrow\exp[\mathrm{mod}(\log[a]+\log[b])]$
5. end
6. return $a\times b$

알고리즘 10. HyMES 구현 : $GF(2^m)$위에서의 나눗셈[4]

데이터: $a,b\in GF(2^m)$, $\exp[i]=\alpha^i$, $\log[\alpha^i]=i$, $i=0,..,2^m-1$
$\mathrm{mod}(d)=((d)\ \&(2^m-1))+((d)\gg m)$
결과: $a/b\in GF(2^m)$
1. if $a=0$
2. $a/b\leftarrow 0$
3. else
4. $a/b\leftarrow\exp[\mathrm{mod}(\log[a]-\log[b])]$
5. end
6. return a/b

알고리즘 11. $g(z)$ 복구

입력: HyMES 키 생성 연산에 대한 소비전력 파형 P, 원시다항식 $p(x)$로 생성한 테이블 $\log[], \exp[]$

출력: $g(z)=z^t+g_{t-1}z^{t-1}+..+g_0$

1. $l: K\times A\rightarrow B,\ S(l,g_i')$, $S_d\in\{0,..,m\}\times\{0,..,m\}$
2. for $j=t-1:1$
3. for $g_j'=0:2^m-1$
4. $S(l,g_j')\leftarrow 0$
5. for $a=0:2^m-1$
6. $S(l,g_j')(\phi(a),\phi(l(a,g_j')))$ $\leftarrow S(l,g_j')(\phi(a),\phi(l(a,g_j')))+\frac{1}{2^m}$
7. end
8. end
9. for $i=0:2^m-1$
10. $\alpha_i\times h_i[j],\log[h_i[j-1]]$의 POI를 찾는다.
11. end
12. slice method를 이용하여 각 데이터의 해밍 웨이트를 추측한다.
13. for $i=0:2^m-1$
14. $S_d(a_i,b_i)\leftarrow S_d(a_i,b_i)+\frac{1}{2^m}$
15. end
16. $g_j\leftarrow 0$
17. for $g_j'=0:2^m-1$
18. if $d(S(l,g_j'),S_d)<d(S(l,g_j),S_d)$
19. $g_j\leftarrow g_j'$
20. end
21. end
22. end
23. return $g(z)=z^t+g_{t-1}z^{t-1}+..+g_0$

Product Distance, χ-square Pearson Distance를 사용하였다.

3.2 L_{sec} 분석

본 절에서는 3.1 절에서 찾아낸 $g(z)$를 이용하여 나머지 비밀키 L_{sec}를 찾아내는 방법에 대하여 설명한다. L_{sec}의 값을 찾는 과정은 알고리즘 11.과 같으며 본 논문에서는 구체적인 예시로 첫 번째 서포트 값 α_0를 찾는 과정을 설명한다. 나머지 서포트 값들도 α_0를 찾는 방법과 같은 논리로 분석이 가능하다.

알고리즘 7.에서 $i=0$ 일 때 처리되는 연산식은 식 (11)와 같다. α_0를 찾기 위해 식 (11)에서 α_0가 포함된 값이 처리될 때 발생하는 실제 소비전력 값들과 α_0의 값을 추측하여 계산한 α_0가 포함된 값들의 해밍 웨이트 간의 수평 상관계수 전력분석을 한다.

식 (11)에서 α_0가 포함된 식의 개수는 $t-1+1+t=2t$ 개로 수평 상관계수 전력분석에 이용 가능한 데이터의 개수는 $2t$개이다. 일반적으로 88-비트

알고리즘 12. L_{sec} 복구

입력: HyMES POI에 대한 소비전력 $\boldsymbol{P}\in Mat_{1\times 8t}$, 원시다항식 $p(x)$로 생성한 테이블 $\log[], \exp[]$, Goppa 다항식 $g(z)=z^t+g_{t-1}z^{t-1}+..+g_0$

출력: $L_{\text{sec}}=\{\alpha_0,..,\alpha_{n-1}\}$

1. for $i=0:n-1$
2. for $j=0:n-1$
3. $pm_1\leftarrow hw(j)$ $\boldsymbol{PM}\in Mat_{1\times 8t}$
4. $pm_2\leftarrow hw(\log[j])$
5. $pm_3\leftarrow hw(\log[j]+\log[h_i[t-1]])$
6. $pm_4\leftarrow hw(\text{mod}(\log[j]+\log[h_i[t-1]]))$
7. $pm_5\leftarrow hw(\exp[\text{mod}(\log[j]+\log[h_i[t-1]])])$
8. for $k=t-2:0$
9. $pm_{5(t-2-k)+6}\leftarrow hw(h_i[k])$
10. $pm_{5(t-2-k)+7}\leftarrow hw(\log[h_i[k]])$
11. $pm_{5(t-2-k)+8}\leftarrow hw(\log[j]+\log[h_i[k]])$
12. $pm_{5(t-2-k)+9}\leftarrow hw(\text{mod}(\log[j]+\log[h_i[k]]))$
13. $pm_{5(t-2-k)+10}\leftarrow hw(\exp[\text{mod}(\log[j]+\log[h_i[k]])])$
14. end
15. for $k=0:t-1$
16. $pm_{3k+5t+1}\leftarrow hw(\log[h_i[k]]-\log[a])$
17. $pm_{3k+5t+2}\leftarrow hw(\text{mod}(\log[h_i[k]]-\log[a]))$
18. $pm_{3k+5t+3}\leftarrow hw(\exp[\text{mod}(\log[h_i[k]]-\log[a])])$
19. end
20. $ct_j\leftarrow\rho(\boldsymbol{PM},\boldsymbol{T})$
21. end
22. $\alpha_i\leftarrow\arg max_j(\boldsymbol{CT})$
23. end
24. return $L_{\text{sec}}=\{\alpha_0,..,\alpha_{n-1}\}$

의 안전성을 갖는 HyMES는 파라미터 $m=11$, $t=32$ 를 사용한다[4]. 해당 파라미터를 사용했을 때 수평 상관계수 전력분석에 이용 가능한 데이터는 64개로, 실제로 수평 상관계수 전력분석을 하기에는 적은 개수이다. 하지만 알고리즘 11.의 단계 2-21과 같이 실제 곱셈과 나눗셈 연산에 사용되는 α_0가 포함된 값 모두를 이용하면, 수평 상관계수 전력분석에 총 $8t=256$ 개의 데이터를 사용할 수 있다. α_0를 0부터 $2^{11}-1$까지 추측하면서 계산한 $8t$개의 소비전력 모델과 실제 소비전력 파형 간의 상관계수를 구하였을 때 가장 높은 상관계수를 갖는 값을 α_0로 찾을 수 있다.

$$
\begin{aligned}
&h_0[t-1]=1\\
&h_0[t-2]=g_{t-1}+(\alpha_0\times h_0[t-1])\\
&h_0[t-3]=g_{t-2}+(\alpha_0\times h_0[t-2])\\
&\qquad\vdots\\
&h_0[0]=g_1+(\alpha_0\times h_0[1])\\
&a=g_0+(\alpha_0\times h_0[0])\\
&h_0[0]=h_0[0]/a\\
&h_0[1]=h_0[1]/a\\
&\qquad\vdots\\
&h_0[t-1]=h_0[t-1]/a
\end{aligned}
\tag{11}
$$

IV. 실 험

본 장에서는 3장에서 설명한 분석기법을 시뮬레이션 파형을 이용하여 실험한 결과를 설명한다. 시뮬레이션 파형의 소비전력 모델은 식 (10)과 같으며 모든 시점에서 α, β를 각각 1, 0으로 동일하게 가정하였다. 분석 대상 알고리즘은 파라미터 $m=11$, $t=32$ 를 사용하는 HyMES[4]를 대상으로 하였으며, 다른 파라미터를 사용하여도 동일한 방법으로 분석이 가능하다.

4.1 시뮬레이션 파형을 이용한 실험

본 절에서는 비밀키 $g(z)$의 32번째 차항의 계수 g_{31}와 L_{sec}의 첫 번째 서포트 값 α_0에 대한 분석 실험결과를 설명한다. 본 논문에서는 g_{31}의 값이 1499일 때와 α_0가 999일 때의 실험결과만을 담았지만 다른 비밀키 값에 대해서도 유사한 실험결과를 확인하였다.

Fig. 1., Fig. 2., Fig. 3.은 g_{31}을 찾기 위해 2개의 확률변수인 α_i와 $\log[h_i[t-2]]=\log[g_{t-1}\oplus\alpha_i]$의 결합 확률 분포를 이용한 분석 결과이며, 각각 추측 키 g_{31}' 값에 따른 이론적 분포와 시뮬레이션 파형으로부터 추정한 분포 간의 유사도 비교방법 3종을 의미한다. 소비전력 모델은 $\alpha=1, \beta=0$, $\sigma=0.1$로 설정하였으며, σ는 노이즈 표준편차를 의미한다. Fig. 1.은 IPD를 이용하여 이론적 분포와 추정한 분포 간의 유사도를 구한 결과로, $g_{31}'=1535$일 때 가장 높은 유사도를 보인다. 따라서 IPD를 이용해 유사도를 비교하는 방식으로는 노이즈가 거의 없는 환경에서도 분석이 어려움을 알 수 있다. 하지만, Fig. 2., Fig. 3.과 같이 각각 HMD와 χ^2PD를 이용하여 비교한 결과는 올바른 비밀키 값인 $g_{31}'=1499$일 때 가장 높은 유사도를 보인다. Fig. 2.는 χ^2PD를 이용하여 이론적 분포와 추정한 분포 간의 유사도를 구한 결과로, $g_{31}'=1499$일 때 분포 간의 거리가 0.00112로 가장 높은 유사도를 보이며, $g_{31}'=1848$ 일 때의 거리

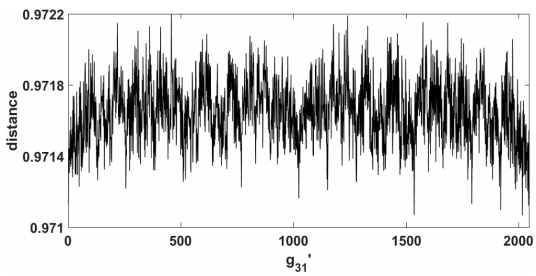


Fig. 1. Inner Product Distance using 2 probability variable $\alpha_i\times h_i[t-1], \log[h_i[t-2]]$

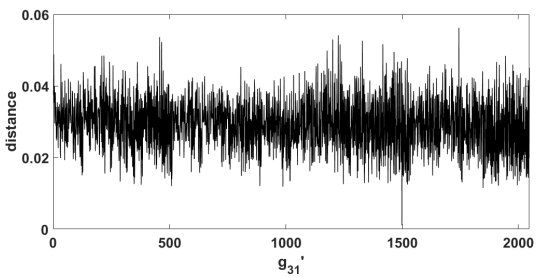


Fig. 2. χ-square Pearson Distance using 2 probability variable $\alpha_i\times h_i[t-1], \log[h_i[t-2]]$

는 0.01152로 두 번째로 높은 유사도를 보인다. HMD를 이용하여 유사도를 구한 결과는 $g_{31}' = 1499$일 때 거리가 0.0001016으로 가장 높은 유사도를 보이며, $g_{31}' = 508$일 때의 거리가 0.007898로 두 번째로 높은 유사도를 보인다. 이론적 결합 확률 분포를 올바른 키로 구성하였을 때의 거리와 틀린 키로 구성하였을 때의 거리 차이가 그리 크지 않기 때문에 노이즈 값이 커지게 되면 분석 실패확률이 높아지게 된다. 하지만, 위 분석은 2개의 확률변수만을 이용하여 결합 확률 분포를 구성하였으며, 실제로는 2개의 확률변수 $\alpha_i \times h_i[t-1] = A$, $\log[h_i[t-2]] = B$에 2개의 확률변수 $\mathrm{mod}(\log[\alpha_i] + \log[h_i[t-2]]) = C$, $\exp[C] = D$를 추가로 사용하여 총 4개의 확률변수를 이용한 결합 확률 분포를 구성하여 분석할 수 있다. Fig. 4.는 4개의 확률변수를 이용하여 구성한 이론적 결합 확률 분포와 추정된 결합 확률 분포 간의 HMD를 구한 결과를 나타낸다. 실험결과는 올바른 비밀키 값인 $g_{31}' =$ 1499일 때의 거리가 0.01776로 가장 높은 유사도를 보이며, $g_{31}' = 927$일 때의 거리가 0.3911로 두 번째로 높은 유사도를 보인다. 4개의 확률변수를 이용하면, 이론적 결합 확률 분포를 올바른 추측 키로 구성하였을 때의 거리와 틀린 키로 구성하였을 때의 거리 차이가 2개의 확률변수를 이용한 결과보다 큰 폭을 보임을 알 수 있다. 이는 확률변수가 많아질수록 키에 따른 결합 확률 분포 간의 차이가 커지기 때문에 발생하는 결과이다.

Fig. 5.는 2개의 변수를 이용한 결합 확률 분포를 구성하여 분석하였을 때, 노이즈 표준편차에 따른 분석 성공률을 보여준다. 분석 성공률은 노이즈 표준편차를 0부터 2.5까지 0.1 단위로 늘려가며 측정하였으며 노이즈 표준편차에 따라 각각 10000번씩 분석을 시행하였다. IPD를 이용한 실험결과는 노이즈 표준편차가 0일 경우에도 성공률이 0이다. χ^2PD를 이용한 결과는 노이즈 표준편차가 0.3일 때까지 분석 성공률이 1이며 0.4부터 분석 성공률이 급격히 감소하기 시작한다. HMD를 이용한 결과는 세 가지 방법 중에 가장 좋은 결과를 보이는데, 노이즈 표준편차가 0.4일 때 분석 성공률이 0.9814로 감소하기 시작한다. Fig. 6.은 4개의 변수를 이용한 실험결과이며 나머지 실험 조건은 Fig. 5.와 동일하다. HMD를 이용하였을 때 노이즈 표준편차에 따른 분석 성공률이 가장 높으며, 노이즈 표준편차 0.6까지 분석 성공률이 1이고 0.7부터 분석 성공률이 0.9823으로 감소하기 시작한다. Fig. 6.의 결과로

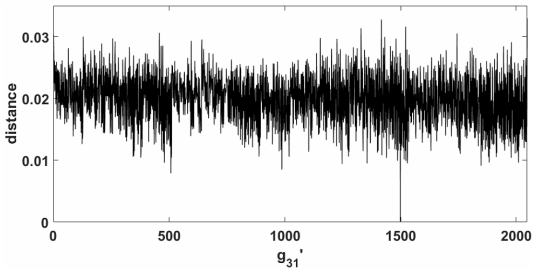


Fig. 3. Harmonic Mean Distance using 2 probability variable $\alpha_i \times h_i[t-1]$, $\log[h_i[t-2]]$

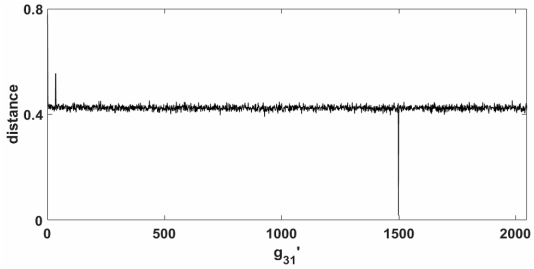


Fig. 4. Harmonic Mean Distance using 4 probability variable $\alpha_i \times h_i[t-1]$, $\log[h_i[t-2]]$, C, D

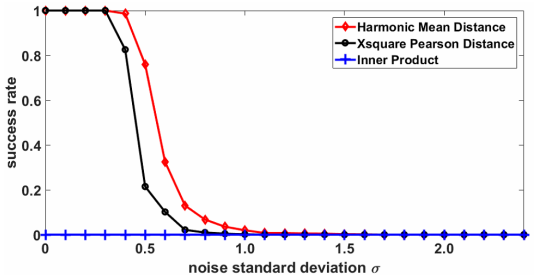


Fig. 5. Success rate according to noise standard deviation (using 2 probability variable)

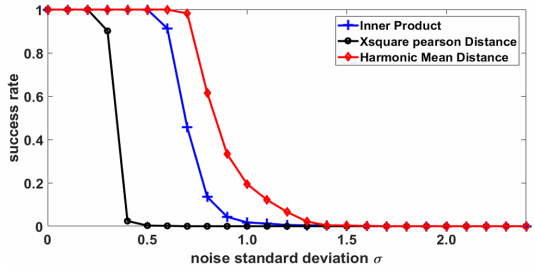


Fig. 6. Success rate according to noise standard deviation (using 4 probability variable)

확률변수 4개를 이용하면 분석 성능이 좋아진다는 것을 알 수 있다. 하지만, 실제 실험환경을 고려하였을 때 노이즈 표준편차 0.7인 환경은 극히 드물기 때문에 분석에 어려움을 겪을 수 있다. 위 실험결과처럼 노이즈 표준편차가 작은 환경에서도 분석 성공률이 높지 않은 이유는 추정된 결합 확률 분포를 구성하기 위해 확률변수의 해밍 웨이트를 추측할 때 실제 값과 오차가 발생하기 때문이다. 하나의 파형만으로 POI(Point Of Interesting)의 해밍 웨이트를 추측하게 되면 노이즈에 의한 오차를 보정하기 매우 어렵다. 하지만, 노이즈의 영향을 확률로 반영할 수 있는 최대 가능도 방법[10]을 이용하여 분석하면 분석 성능을 높일 수 있다. 최대 가능도 방법을 이용하여 분석한 결과는 Fig. 7.과 같다. 변수 2개를 사용한 경우에 노이즈 표준편차 0.4까지 분석 성공률 1로 Fig. 5.의 기존의 방법을 이용한 결과보다 근소하지만 좋은 결과를 보인다. 변수 4개를 사용한 경우에는 노이즈 표준편차 1.3까지 분석 성공률 1인 결과를 보이며, 이는 변수 4개를 이용하여 분포 간의 유사도를 비교하는 방법 중 가장 결과가 좋은 HMD를 이용한 분석 성능보다 약 2배 높은 분석 성능을 보인다.

Fig. 8.은 α_0를 찾기 위해 256개의 부분 파형으로 수평 상관계수 전력분석을 하였을 때 노이즈 표준편차에 따른 분석 성공률을 나타낸다. 분석 성공률은 노이즈 표준편차를 0부터 2.5까지 0.1 단위로 늘려가며 측정하였으며 노이즈 표준편차에 따라 각각 10000번씩 수평 상관계수 전력분석을 시행하였다. 노이즈 표준편차가 2.7까지 분석 성공률이 1이고, 노이즈 표준편차 2.8부터 감소하기 시작한다.

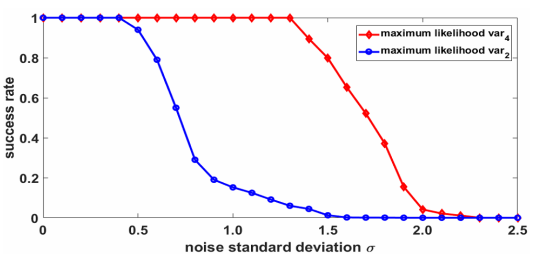


Fig. 7. Success rate according to noise standard deviation (using maximum likelihood)

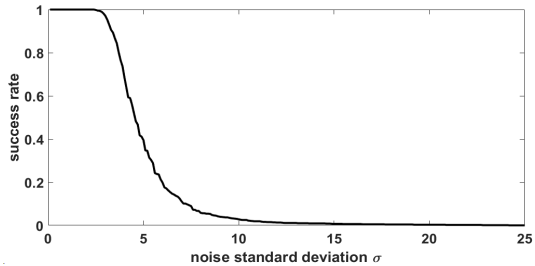


Fig. 8. Success rate according to noise standard deviation ($\alpha_0 = 999$)

V. 결 론

기존의 McEliece 암호시스템의 단점인 큰 키 크기를 보완하고자 개발된 HyMES는 암호화, 복호화 속도의 효율성이 높고 키의 크기도 매우 감소하였다. 하지만, 복호화 과정에서 신드롬 값을 계산하기 위해 패리티 검사 행렬 H의 열벡터들인 h_i 값을 계산하는 과정에서 비밀키 값에 따라 결합 확률 분포의 차이가 발생하고 본 논문에서는 이를 이용하여 HyMES를 분석하였다.

기존의 결합 확률 분포를 이용한 분석은 대칭키 암호에만 국한되어 연구되었지만, 본 논문에서 처음으로 공개키 암호시스템에 대하여 결합 확률 분포 기반 분석을 하였다. 시뮬레이션 파형을 이용한 실험결과는 노이즈 표준편차 1.3까지 100%의 분석 성공률을 보인다.

본 논문에서 제안한 분석기법은 [4]에서 제안된 방식으로 h_i를 계산하는 모든 McEliece 기반 암호시스템에 적용할 수 있다. h_i를 계산하는 방법은 [4]에서 제안된 방식 말고도 다른 방법이 있는데, $z+\alpha_i$의 역원을 확장 유클리드 알고리즘을 이용하여 구하거나 키에 따른 결합 확률 분포의 차이를 두드러지게 만드는 로그 테이블 및 지수연산 테이블을 사용하지 않는 방법이 있다. 두 가지 방법 모두 제안하는 분석기법에 안전하지만, [4]에서 사용하는 방법에 비해 연산속도에 효율성이 떨어진다. 따라서 본 분석방법에 안전하면서 h_i를 효율적으로 계산하기 위해서는 키에 따른 결합 확률 분포의 차이를 감소시킬 수 있도록 비선형성이 낮은 로그 테이블을 사용해야 한다. 이를 위해 비선형성이 낮은 로그 테이블을 생성하는 원시다항식 $p(x)$를 사용하여 유한체 $GF(2^m)$를 생성해야 한다.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," CRYPTO'99, pp. 789-789, 1999.

[2] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," Deep Space Network Progress, vol. 44, pp. 114 - 116, 1978.

[3] N. Patterson, "The Algebraic Decoding of Goppa Codes," IEEE Transactions on Information Theory, vol. 21, pp. 203 - 207, 1975.

[4] B. Biswas, N. Sendrier, "McEliece cryptosystem implementation: Theory and practice," PQCrypto 2008, vol. 5299, pp. 47 - 62, 2008.

[5] S. Heyse, A. Moradi, C. Paar, "Practical power analysis attacks on software implementations of McEliece," PQCrypto 2010, vol. 6061, pp. 165 - 181, 2010.

[6] S. Chari, J. Rao, P. Rohatgi, "Template attacks," CHES 2002, vol. 2523, pp. 13 - 28, 2003.

[7] K. Schramm, G. Leander, P. Leander, C. Paar, "A collision attack on AES: Combining side channel and differential attack," CHES 2004, vol. 3156, pp. 163 - 175, 2004.

[8] Y. Linge, C. Dumas, "Using the joint distributions of a cryptographic function in side channel analysis," COSADE 2014, vol. 8622, pp. 199 - 213, 2014.

[9] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, V. Verneuil, "Horizontal Correlation Analysis on Exponentiation," ICICS 2010, vol. 6476, pp. 46 - 61, 2010.

[10] C. Clavier, L. Reynaud, "Improved blind side-channel analysis by exploitation of joint distributions of leakages," CHES 2017, pp. 24 - 44, 2017.

[11] F. Strenzke, E. Tews, H. Molter, R. Overbeck, A. Shoufan, "Side Channels in the McEliece PKC," PQCrypto 2008, vol. 5299, pp. 216 - 229, 2008.

[12] F. Strenzke, "A timing attack against the secret permutation in the McEliece PKC," PQCrypto 2010, vol. 60, pp. 95 - 107, 2010.

[13] R. Avanzi, S. Hoerder, D. Page, M. Tunstall, "Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems," Journal of Cryptographic Engineering, vol. 1, no. 4, pp. 271 - 281, 2011.

[14] H. B. Nguyen, "An overview of the NTRU cryptographic system," M.S. thesis, 2014.

[15] S. Siim, "Study of McEliece cryptosystem," 2015.

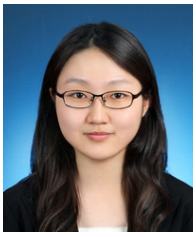
[16] H. C. Hudde, "Development and Evaluation of a Code-based Cryptography Library for Constrained Devices," M.S. thesis, 2013.

[17] D. J. Bernstein, T. Chou, and P. Schwabe, "McBits: Fast constant-time code-based cryptography," CHES 2013, vol. 8086, pp. 250 - 272, Aug. 2013.

[18] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM review, vol. 41, no. 2, pp. 303-332, 1999.

〈저자소개〉

박 병 규 (ByeongGyu Park) 학생회원
2017년 2월: 고려대학교 정보수학과 학사
2017년 3월~현재: 고려대학교 사이버국방학과 석사과정
〈관심분야〉 부채널 공격, 후양자암호



김 수 리 (Suhri Kim) 학생회원
2014년 2월: 고려대학교 수학과 학사
2016년 8월: 고려대학교 정보보호학과 석사
2016년 9월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 후양자암호

김 한 빛 (Hanbit Kim) 학생회원
2014년 2월: 고려대학교 신소재공학과 학사
2016년 2월: 고려대학교 정보보호학과 석사
2016년 3월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 부채널 공격, 부채널 대응기법, 암호시스템 안전성 분석 및 고속구현

진 성 현 (Sunghyun Jin) 학생회원
2015년 2월: 서울시립대학교 수학과 학사
2017년 2월: 고려대학교 정보보호학과 석사
2017년 3월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 부채널 공격

김 희 석 (HeeSeok Kim) 정회원
2006년: 연세대학교 수학과 학사
2008년: 고려대학교 정보보호대학원 석사
2011년: 고려대학교 정보보호대학원 박사
2011년 9월~2012년 12월: Bristor University 박사후 연구원
2013년~2016년 8월: 한국과학기술정보연구원(KISTI) 선임연구원
2015년~2016년 8월: 과학기술연합대학원대학교(UST) 조교수
2016년 9월~현재: 고려대학교 과학기술대학 사이버보안전공 조교수
〈관심분야〉 부채널 공격, 암호시스템 안전성 분석 및 고속구현, 암호칩 설계 기술, 보안관제, 네트워크 보안

홍 석 희 (Seokhie Hong) 종신회원
1995년: 고려대학교 수학과 학사
1997년: 고려대학교 수학과 석사
2001년: 고려대학교 수학과 박사
1999년 8월~2004년 2월: ㈜시큐리티 테크놀로지 선임연구원
2003년 3월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U. Leuven ESAT/SCD-COSIC 박사후 연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키 및 공개키 암호 알고리즘, 부채널 공격 및 대응기법, 디지털 포렌식