논문 2013-50-2-9

이동 Ad-Hoc 네트워크 환경에서 페어링 연산의 밀러 알고리듬에 대한 데이터 오류 공격

(A Data Fault Attack on the Miller Algorithm for Pairing Computation in Mobile Ad-Hoc Network Environments)

배 기 석*, 손 교 용**, 박 영 호***, 문 상 재****

(KiSeok Bae, GyoYong Sohn, YoungHo Park, and SangJae Moon)

요 약

최근 이동 ad hoc 네트워크에 적합한 ID 기반의 암호시스템 구현을 위해서 다양한 페어링 연산들이 사용되고 있으며, 밀러 알고리듬은 Weil, Tate, Ate 페어링 연산에서 가장 많이 사용되는 알고리듬이다. 본 논문에서는 Whelan과 Scott에 의해 제안된 밀러 알고리듬의 중간 값에 대한 오류 공격을 구체화하여 라운드 위치와 상관없이 적용할 수 있는 데이터 오류 주입 공격의 가능성을 분석하였다. 시뮬레이션 결과, 제안하는 공격 방법이 라운드 위치나 사용하는 좌표계와 관계없이 적용 가능하여 효과적이며 강력한 오류 주입 공격 방법임을 확인하였다.

Abstract

Recently, there has been introduced various types of pairing computations to implement ID based cryptosystem for mobile ad hoc network. The Miller algorithm is the most popular algorithm for the typical pairing computation such as Weil, Tate and Ate. In this paper, we analyze the feasibility of concrete data fault injection attack, which was proposed by Whelan and Scott, in terms of regardless of round positions during the execution of the Miller algorithm. As the simulation results, the proposed attack that can be employed to regardless of round positions and coordinate systems is effective and powerful.

Keywords : 페어링 연산, 오류 주입 공격, 데이터 오류, 밀러 알고리듬

Ⅰ. 서 론

개인 정보의 중요성이 높아짐에 따라 이동 ad hoc 네트워크에서 개인의 프라이버시 보호에 대한 연구 필요성이 대두되고 있다. 이동 ad hoc 네트워크를 안전하게 구현하기 위한 보안 분석, 보안 프로토콜 등과 같은 보안 체계와 관련된 연구가 진행되고 있으며 네트워크를 구성하는 단말장치의 낮은 연산 능력과 저장 공간의 부족 등의 한계성을 고려한 효율적인 보안 알고리듬에 관한 연구도 같이 수행되고 있다[1~2]. 이를 위해 최근 이론적으로 높은 안전도를 지닌 페어링 기반 암호 시스템의 사용이 각광받으면서 실제 구현 단계에서 발생할 수 있는 부채널 기반 분석도 함께 요구되어지고 있다[3~6]. 페어링 연산의 경우, 타 암호 시스템과 달리 입력 값에 비밀 값이 포함되기 때문에 내부의 비밀 값을 대상으로한 기존의 공격 방법들은 그대로 적용할 수 없

* 정회원, **** 평생회원, 경북대학교 전자전기컴퓨터학부
(Kyungpook National University)
** 정회원, 대구교육대학
(Daegu National Unversity of Education)
*** 정회원, 경북대학교 산업전자공학과
(Kyungpook National University)
※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2012R1A1A4A01002603)
접수일자: 2012년8월13일, 수정완료일: 2013년2월6일

다. 따라서 페어링 연산에 대한 오류 공격들은 주로 오류 출력 값과 정상적인 연산 결과를 비교하여 비밀 값과 관련된 중간 값의 정보를 통해 비밀 값을 찾아내는 형태를 가졌다. 이때 주입되는 오류 주입 기법들은 알고리듬의 반복 횟수를 변화하여 오류 연산을 수행하게 하거나 알고리듬 동작 중의 중간 연산 값을 변형할 수 있다고 가정한다[7~8].

먼저 Page와 Vercauteren은 밀러 알고리듬에서의 오류 주입 공격을 처음으로 소개하였다[7]. 오류주입을 통해 반복문의 라운드 수행 횟수가 변화하였을 때의 오류 출력 값들을 비교하여 비밀 값을 추출하는 방식을 사용한다. 다음으로 Whelan과 Scott의 공격에서는 Weil 페어링과 Eta 페어링 상의 오류 주입 공격을 제안하고 있다[8]. Whelan과 Scott의 공격[8]에서는 알고리듬 마지막 라운드 연산 도중에 주입된 오류에 의해서 내부의 중간 값이 변형된 오류 출력 값과 정상 출력 값을 비교하여 비밀 값을 찾아낼 수 있음을 예제로 기술하였다. Page와 Vercauteren의 경우[7] 오류 주입으로 인해 라운드 횟수가 랜덤하게 수행될 때, 연속적인 라운드 횟수만큼 수행한 오류 결과문 쌍 $(e_m(P,Q), e_{m+1}(P,Q))$을 얻기 위한 오류 주입을 반복적으로 시도해야하나, Whelan과 Scott의 공격은 중간 값의 일부를 변형하는 한 번의 오류 주입으로 충분하다. 그러나 Weil 페어링의 경우, 최상위 비트인 부호 비트를 변경할 수 있는 매우 정밀한 오류 주입을 가정하고 있으며 마지막 라운드라는 오류주입 상의 위치 제한도 포함하고 있어 실제 구현이 용이하지 않다.

본 논문은 밀러 알고리듬에 대한 데이터 오류 주입 공격을 일반화하여 특정 라운드 위치에 구애받지 않고 적용 가능한 공격 방법을 구체적으로 제시한다. 제안하는 공격 방법은 알고리듬 수행 중의 중간 값에 대해서 사용하는 기저 중의 일부만을 대상으로 하기 때문에 구현이 용이하다. 또한, 컴퓨터 시뮬레이션을 통해 실제로 동작함을 확인하고, 이동 ad hoc 네트워크의 다양한 구성환경에 맞춰 달리 적용될 때를 가정하여 여러 좌표계에 대한 공격 기법의 적용을 설명한다.

Ⅱ. 페어링 연산과 밀러 알고리듬

1. 페어링 연산

먼저 페어링 연산을 구성하는 F_q를 q개 원소를 가지는 유한체라고 가정한다. 여기서 q는 표수(characteristic) p의 멱승이며, Jacobian 좌표를 사용하는 경우에는 E는 유한체 F_q 위에 정의된 수식 (1)과 같은 Weierstrass 방정식을 따르게 된다. 이때, O는 타원곡선 E상의 무한점이라고 한다.

$$E: Y^2 = X^3 + aXZ^4 + bZ^6,\ a,b \in F_q \tag{1}$$

l은 $l|E(F_q)$를 가지는 q와 서로소인 양의 정수라고 할 때, 묻기 차원(embedding degree) k는 $l|q^k-1$를 만족하는 가장 작은 정수이다. 그러면 Tate 페어링은 비퇴화성(none-degeneracy)와 겹선형성(bilinearity)를 만족하는 다음과 같은 map으로 정의된다.

$$\langle \bullet, \bullet \rangle : E(F_q)[l] \times E(F_{q^k})/lE(F_{q^k}) \rightarrow {F_{q^k}}^*/({F_{q^k}}^*)^l \tag{2}$$
$$\langle P,Q \rangle = f_{l,P}(D_Q)$$

여기서 $f_{l,P}$는 E의 유리함수이고, $f_{l,P}$의 divisor인 $\mathrm{div}(f_{l,P})$는 $l(D)-l(O)$와 동치인 특성을 가진다. 즉, $\mathrm{div}(f_{l,P}) = l(P)-([l]P)-(l-1)O$을 만족하게 된다. 또한, D_Q는 $(Q)-(O)$와 동치인 0 차수의 divisor이며, $\mathrm{div}(f_{l,P})$와 D_Q는 서로 다른 support를 갖는다. Divisor에 관해서는 [9]에서 자세히 설명하고 있다. 또한 수식 (2) 상의 $f_{l,P}$는 $\prod_i f_{l,P}(P_i)^{a_i}$로 표현되는 누적 곱셈의 형태이며, 여기서 $D = \sum_i a_i P_i$이다.

Tate 페어링의 결과 값은 유일하지 않으며, l번 거듭제곱까지 정의되어 있는 잉여류 $(F_{q^k}^*)/(F_{q^k}^*)^l$ 상에 존재하는 하나의 값이다. 따라서 페어링 연산을 위해서는 유일한 값을 얻을 수 있는 다음의 reduced Tate 페어링을 정의하여 사용하게 된다.

$$e_l(P,Q) = \langle P,Q \rangle^{(q^k-1)/l} \tag{3}$$

여기서, $(q^k-1)/l$은 마지막 멱승법으로 알려져 있다.

2. 밀러 알고리듬

타원곡선 상의 $[l]P$를 연산하는 동안 생성되는 중간 점들 간의 접선의 함수를 루프에 대해 반복하며 누적 곱셈하여 계산한다. Jacobian 좌표계에서 사용되는 실제 알고리듬은 그림 1과 같다.

본 논문에서는 Brier와 Joye에 의해 제안된 $a=-3$을 고려한 밀러 알고리듬을 사용한다[10]. 또한, 묻기 차원 $k=4$일 때를 고려하여 F_{q^k}의 기저를 $B = \{1, \xi, \sqrt{\nu}, \xi\sqrt{\nu}\}$로 둔다[11]. 이때 $\{1,\xi\}$는 F_q위

INPUT : $P \in E(F_q)$, $Q \in E(F_q)$, torsion l
OUTPUT : $f \in F_{q^k}$
1. $T \leftarrow P$, $f \leftarrow 1$
2. for $i = \lfloor \lg(l) \rfloor - 1$ to 0 do
3. $f \leftarrow f^2 \cdot h_1$ (tanget line at T)
4. $T \leftarrow 2T$
5. if $l_i = 1$ then
6. $f \leftarrow f \cdot h_2$ (line of P and T)
7. $T \leftarrow T + P$
8. end if
9. end for
10. return f

그림 1. 밀러 알고리듬
Fig. 1. Miller Algorithm.

의 확장체 $F_{q^{k/2}}$의 기저이며, $\{1, \sqrt{\nu}\}$는 $F_{q^{k/2}}$위의 확장체 F_{q^k}의 기저이다. 페어링 연산을 위해서 E상의 두 점 $P = (X_P, Y_P, Z_P) \in E(F_q)$와 아핀 좌표의 점 $Q \in E(F_{q^k})$을 사용하는데, k가 짝수이므로 뒤틀린(twisted) 타원곡선을 이용하여 $Q = (x, y\sqrt{\nu})$으로도 사용가능하다. 여기서 x, y, $\nu \in F_{q^{k/2}}$이고 $\sqrt{\nu} \in F_{q^k}$이다. 주어진 임의의 두 점 $Q = (x, y\sqrt{\nu})$, $T = (X, Y, Z)$에 대해서 그림 1의 밀러 알고리듬 상에서 계산되는 함수 h_1과 h_2는 다음과 같이 정의된다.

$$h_1(x, y\sqrt{\nu}) = Z_3 Z^2 y\sqrt{\nu} - 2Y^2 - 3(X^2 - Z^4)(xZ^2 - X) \tag{4}$$

여기서, $Z_3 = 2YZ$이다.

$$h_2(x, y\sqrt{\nu}) = Z_3 y\sqrt{\nu} - (Y_P Z^3 - YZ_P^3)x - (X_P Y Z_P - X Y_P Z) \tag{5}$$

여기서 $Z_3 = ZZ_P(X_P Z^2 - XZ_P^2)$이다.

Ⅲ. 밀러 알고리듬에 대한 오류공격 기법 제안

페어링 연산에 대한 오류 공격에는 Counter 오류 기법[7, 12]과 Whelan과 Scott이 제안한 데이터에 오류를 주입하는 기법, 부호를 변경하는 기법[8] 등이 대표적이다. Counter 오류 기법은 알고리듬의 라운드를 반복 수행할 때, 반복 횟수(counter)가 주입된 오류에 의해 변형되어 알고리듬이 정상적인 반복 횟수를 수행하지 못하게 만든다. 이후 비정상적인 반복 횟수의 오류 결과 값 하나와 그보다 하나의 라운드가 더 수행된 경우의 다른 오류 결과 값들($e_m(P,Q), e_{m+1}(P,Q)$)의 차이를 이용하여 비밀 값을 추출한다. 데이터에 오류를 주입하는 기법은 공격 대상을 임의의 값으로 변형시키는 경우와 정밀한 위치 조정을 통해서 하나의 비트만 뒤집는 경우로 나눌 수 있다.

데이터 오류 공격[8]에서는 알고리듬의 수행 중 특정 중간 값의 기저 중에서 하나를 대상으로 데이터 오류를 주입하는 기법을 제안하였는데, Weil 페어링의 경우 중간 값의 y좌표의 부호에 해당하는 하나의 비트를 바꿔야하는 정밀도가 요구되며, 마지막 라운드에 한해서만 동작된다는 한계로 인해 실제 구현상의 어려움이 존재하였다. Counter 오류 기법의 경우에는 반복 횟수가 연속적인 오류 결과문 쌍을 얻을 때까지 수차례의 오류 주입 시도가 요구되며, 임의의 반복 횟수로 변형되기 때문에 오류 주입 횟수에 대한 확률적인 분석이 필요하였다[12]. 따라서 데이터 오류 기법은 구현의 어려움을 제외하고는 오류 결과 값과 정상적으로 수행된 결과 값 한 쌍의 비율만을 사용하므로 오류 주입 횟수에 대해서 보다 효율적인 공격 방법이라고 볼 수 있다. 따라서 구현상의 한계를 개선한 데이터 오류공격 기법을 이 장에서 설명한다.

1. 제안 공격 기법의 개요

기존 공격 방법[8]과 같이 공격자는 오류가 주입된 라운드에서 연산된 함수만을 도출하기 위해서 오류 페어링 결과 값 $e(P,Q)'$과 정상 페어링 값 $e(P,Q)$의 비율을 계산한다.

여기서 밀러 루프란 반복되는 알고리듬의 라운드 전체 과정을 말하며, 밀러 루프 이후 마지막 멱승과정을 거쳐 페어링 값이 출력된다. 기존 공격 방법[7, 12]과 같이 공격자는 오류가 주입된 라운드에서 연산된 함수만을 도출하기 위해서 두 페어링 결과 값 $e(P,Q)'$과 $e(P,Q)$의 비율을 계산한다. 예를 들어 밀러 알고리듬의 임의의 d 라운드의 h_1함수에 오류가 주입되었다고 가정하고 두 페어링 결과 값의 비율은 다음과 같이 된다.

$$\frac{e(P,Q)'}{e(P,Q)} = \left(\frac{(((h_{1,1}h_{1,2})^2 h_{2,1}h_{2,2})^2 \ldots)^2 h'_{d,1}h_{d,2})^2 \ldots)^2 h_{m,1}h_{m,2}}{(((h_{1,1}h_{1,2})^2 h_{2,1}h_{2,2})^2 \ldots)^2 h_{d,1}h_{d,2})^2 \ldots)^2 h_{m,1}h_{m,2}} \right)^W = \left(\frac{h_{d,j}'}{h_{d,j}} \right)^{(2^{m-d})W} \tag{6}$$

여기서, $m = \lfloor \lg(l) \rfloor$, $h_{i,j}$는 i는 밀러 루프의 순서이고 $j \in \{1,2\}$는 그림 1의 4번째 또는 7번째 함수이며 타원곡선의 점 P의 꼬임수 l의 이진표현에 의존한다. 이는 h_1과 h_2의 값을 도출하기 위해서 오류가 주입되어야 할 위치이기 때문이다. 또한, 페어링의 마지막 멱승과정에서 사용되는 멱승수는 $W = (q^k - 1)/l$이며, 정상적인 페어링 결과 값이나 오류 결과 값이나 동일하게 적용되고 있다. 마지막 멱승의 경우 [7]에서 소개된 역환 방법이나 [12]에서의 생략 방법들로 무시할 수 있다고 가정하므로 본 논문에서는 밀러 루프의 출력 값만을 고려한다. 수식 (6)의 비율은 멱승을 벗겨낸 값이 되며, 오류가 주입된 라운드의 위치 d에 따라 수식 (7)로 정리할 수 있다[8]. 오류가 주입된 라운드를 제외한 나머지 연산 값 h_i들은 동일하므로 소거되었으며, 수식 (7)에 대해서 $1/2^{m-d}$의 멱승을 수행하면 $h_{d,j}'/h_{d,j}$의 유일한 근을 찾아낼 수 있다.

$$\frac{e(P,Q)'}{e(P,Q)} = \left(\frac{h_{d,j}'}{h_{d,j}}\right)^{2^{m-d}} \tag{7}$$

마지막으로 묻기 차원 $k = 4$인 타원곡선 방정식 (3)을 고려하는 경우, 유한체 F_{q^k}의 원소는 네 개의 원소 $a_i \in F_q$로 표현이 되며, 이들 기저에 해당되는 값들은 4개의 서로 다른 배열로 구현된다. 우리는 이러한 유한 확장체 원소의 저장형태를 $[a_0][a_1][a_2][a_3]$로 표현되며, 주입하는 데이터 오류는 이들 배열 중에 하나를 변형하는 것으로 가정한다. 또한 마지막 라운드라는 특정 위치가 아닌 임의의 라운드에서의 오류 주입을 가정한다. 이러한 연산 도중의 임의의 데이터 오류 주입 방식은 기존의 CRT-RSA 암호 시스템에 대한 공격 기법으로 수차례 소개된 형태이며, 실제 실험적으로 가능함이 밝혀졌다[13~14]. 본 논문은 이와 같은 데이터 오류를 사용하여 마지막 라운드를 포함한 모든 라운드를 고려하는 데이터 오류 주입 공격을 구체적으로 제시한다.

2. h_1 함수 연산에 대한 오류

공격자는 밀러 알고리듬 그림 1의 임의의 d 라운드에서 연산되는 h_1 함수를 대상으로, 임의의 좌표에 오류를 주입한다고 가정한다. 그림 1의 $l_d = 0$ 또는 1인 경우에 상관없이 앞 절에서 설명한 정상 페어링의 값과 오류 주입 페어링 값 간의 비율은 아래의 수식으로 표현된다.

$$\frac{e(P,Q)'}{e(P,Q)} = \frac{h_{d,1}'}{h_{d,1}} \tag{8}$$

따라서 $l_d = 0$ 또는 1에 상관없이 공격자는 h_1함수에 오류를 주입할 수 있으므로, 알고리듬의 라운드의 위치에 의존하지 않는다. 오류가 주입된 임의의 d 라운드에 입력되는 타원곡선의 점을 $[j]P = (X_j, Y_j, Z_j)$라고 할 때, 이는 $d-1$ 라운드에 저장되어진 점 T의 값이다. 그림 1의 4 번째 단계의 $h_1(Q)$의 형태는 다음의 수식의 형태를 취한다.

$$h_1(x, y\sqrt{\nu}) = 2Y_jZ_j^3 y\sqrt{\nu} - 2Y_j^2 - 3(X_j^2 - Z_j^4)(xZ_j^2 - X_j) \tag{9}$$

묻기 차원 $k = 4$에 의해서 유한체 F_{q^k}상의 원소를 $B = \{1, \xi, \sqrt{\nu}, \xi\sqrt{\nu}\}$와 같은 네 개의 기저로 표현 할 때, 수식 (8)의 비율 값은 아래의 수식으로도 표현 가능하다.

$$\frac{h_1(Q)'}{h_1(Q)} = R_0 + R_1\xi + R_2\sqrt{\nu} + R_3\xi\sqrt{\nu} \tag{10}$$

여기서 R_0, R_1, R_2, $R_3 \in F_q$ 이며, 공격자가 알고 있는 값들이다. 앞서 유한 확장체 원소의 저장형태를 $[a_0][a_1][a_2][a_3]$으로 가정하였으며, 주입한 오류의 위치를 $h_1(Q)$의 첫 번째 자리로 가정한다면, 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned}\frac{h_1(Q)'}{h_1(Q)} &= \frac{[a_0]'[a_1][a_2][a_3]}{[a_0][a_1][a_2][a_3]} \\ &= [R_0][R_1][R_2][R_3]\end{aligned} \tag{11}$$

또 다른 입력 값인 $Q = (x, y\sqrt{\nu})$는 공개 값이므로, 공격자는 알고 있는 좌표 값 $x = x_0 + x_1\xi$와 $y = y_0 + y_1\xi$를 대입하여 위의 식을 정리하면 아래와 같다. 이때 결합된 기저들은 $\xi^2 = \gamma \in F_q$, $\nu = \alpha + \beta\xi \in F_{q^{k/2}}$으로 가정하였다.

$$a_0' = R_0a_0 + R_1a_1\gamma + R_2a_2\alpha + R_2a_3\beta\gamma + R_3a_2\beta\gamma + R_3a_3\alpha\gamma \tag{12}$$

$$a_1 = R_0a_1 + R_1a_0 + R_2a_2\beta + R_2a_3\alpha + R_3a_2\alpha + R_3a_3\gamma\beta \tag{13}$$

$$a_2 = R_0a_2 + R_1a_3\gamma + R_2a_0 + R_3a_1\gamma \tag{14}$$

$$a_3 = R_0a_3 + R_1a_2 + R_2a_1 + R_3a_0 \tag{15}$$

공격자는 위의 방정식을 풀기 위해 최소의 미지수가 포함된 3개의 방정식 (13), (14), (15)를 선택한다. 이는 수식 (12)는 오류로 인해 임의의 값으로 변형된 a_0'가 포함되어 있어 또 하나의 미지수가 발생하고, 수식 (12)에 대한 정보가 없더라도 남은 3개의 수식을 통해서 수식 풀이가 가능하기 때문이다. 수식 (13), (14), (15)에 수식 (9)의 $h_1(Q)$의 각각의 좌표 값에 해당하는 값들을 치환한 후 다시 정리하였을 때 공격자는 다음의 3개의 비선형 방정식을 얻을 수 있다.

$$\begin{cases} 3R_1X_j^3-3R_1X_jZ_j^4-2R_1Y_j^2 \\ \quad +A_{00}X_j^2Z_j^2+A_{01}Z_j^6+A_{02}Y_jZ_j^3=0\ , \\ 3R_2X_j^3-3R_2X_jZ_j^4-2R_2Y_j^2 \\ \quad +A_{10}X_j^2Z_j^2+A_{11}Z_j^6+A_{12}Y_jZ_j^3=0\ , \\ 3R_3X_j^3-3R_3X_jZ_j^4-2R_3Y_j^2 \\ \quad +A_{20}X_j^2Z_j^2+A_{21}Z_j^6+A_{22}Y_jZ_j^3=0\ , \end{cases} \tag{16}$$

여기서, $A_{00}=3((1-R_0)x_1-x_0R_1)$,
$A_{01}=3(R_1x_0+(R_0-1)x_1)$,
$A_{02}=2(R_2(y_0\beta+y_1\alpha)+R_3(y_0\alpha+y_1\beta\gamma))$,
$A_{10}=-3(R_2x_0+R_3\gamma x_1)$,
$A_{11}=3(R_2x_0+R_3\gamma x_1)$,
$A_{12}=2(y_0(R_0-1)+y_1R_1\gamma)$,
$A_{20}=-3(R_3x_0+R_2x_1)$,
$A_{21}=3(R_3x_0+R_2x_1)$,
$A_{22}=2(y_0R_1+(R_0-1)y_1)$ 이다.

공격자는 유한체 F_q의 계수를 가지는 세 개의 미지수 X_j, Y_j, Z_j로 이루진 세 개의 비선형 방정식으로부터 $[j]P=(X_j,Y_j,Z_j)$를 얻을 수 있다. 이를 위해서 두 방정식의 공동 근을 찾아주는 resultant방법을 사용한다[15].

먼저 수식 (16)의 세 방정식과 주어진 타원곡선 방정식을 이용하여 F_q에서 계수로 가지는 세 개의 비선형 방정식을 얻을 수 있다. 여기서 $F_i(X_j,Y_j)_{Z_j}$, $i=1,2,3$를 $F_q[X_j,Y_j]$에서 계수를 가지는 Z_j의 방정식으로 둔다. 그럼 F_i, $i=1,2,3$의 차수는 각각 4가 된다. 다음으로 두 방정식의 쌍 F_1, F_2 그리고 F_2, F_3에 대해 각각 resultant를 사용하여 $Res_{Z_j}(F_1,F_2)$와 $Res_{Z_j}(F_2,F_3)$를 구한다. 여기서 $Res_{Z_j}(F_1,F_2)$는 $F_q[X_j,Y_j]$안의 다항식으로써 근들은 두 다항식 F_1과 F_2의 공동 X_j, Y_j 좌표들이다. $Res_{Z_j}(F_2,F_3)$도 마찬가지다. 앞의 두 resultant 값을 각각 $S_1(X_j,Y_j)$와 $S_2(X_j,Y_j)$로 두었을 때, 각각을 $F_q[X_j]$에서 계수를 가지는 Y_j의 방정식으로 고려할 수 있다. 이 때 S_1과 S_2의 차수는 8이다. 마지막으로 두 방정식의 resultant 값, $Res_{Y_j}(S_1,S_2)$, 을 구하면 X_j의 다항식으로써 그들의 근은 S_1과 S_2의 공동된 X_j 좌표들이 된다. 이때의 $Res_{Y_j}(S_1,S_2)$의 X_j의 차수는 160이며, Resultant의 성질에 의해 X_j의 좌표 값이 0인 20개의 후보군과 공통근을 제외하면 총 75개의 X_j 후보군을 구할 수 있다. 따라서 (X_j,Y_j,Z_j)좌표 값의 최종 후보군의 개수는 150(=75×2×1)이다.

이후 각 후보군에 대해서 j^{-1}의 스칼라 곱셈을 수행한 후, P좌표의 후보군들 중에서 정상적인 밀러 루프 결과를 출력하는지 여부를 판단하여 올바른 비밀 키 값을 검증한다.

3. h_2함수 연산에 대한 오류

다음으로 밀러알고리듬의 임의의 라운드 d상에서 연산되는 함수 h_2에 오류를 주입 했을 때를 고려한다. 이는 그림 1의 5번째 줄의 $l_d=1$인 경우이며, 타원곡선상의 add에 해당된다. d 번째 라운드에서 입력되는 타원곡선의 점을 $[j]P=(X_j,Y_j,Z_j)$라고 두었을 때, 그림 1의 4번째와 5번째 연산의 결과 점들을 각각 $[2j]=(X_{2j},Y_{2j},Z_{2j})$, $[2j+1]P=(X_{2j+1},Y_{2j+1},Z_{2j+1})$으로 표현한다. $[j]P$와 $[2j]P$간의 add 기법에 의해서 계산되는 $h_2(Q)$는 아래의 수식으로 표현된다.

$$\begin{aligned} h_2(x,y\sqrt{\nu})=Z_{2j+1}y\sqrt{\nu}-(Y_PZ_{2j}^3-Y_{2j}Z_P^3)x \\ -(X_PY_{2j}Z_P-X_{2j}Y_PZ_{2j}). \end{aligned} \tag{17}$$

여기서 $Z_{2j+1}=Z_PZ_{2j}(X_PZ_{2j}^2-X_{2j}Z_P^2)$이다. 우리는 첫 번째 배열(기저)에 부호 오류를 주입했다고 가정하자. 이는 공격자가 얻고자 하는 미지수의 개수만큼 방정식을 얻기 위해서이다. 앞서 설명한 방법과 같이 오류 결과 값과 정상 결과 값의 비율인 $e(P,Q)'/e(P,Q)$은 아래의 수식과 같이 표현된다.

$$\frac{h_2(Q)'}{h_2(Q)}=\frac{[-b_0]'[b_1][b_2][b_3]}{[b_0][b_1][b_2][b_3]}=[R_0][R_1][R_2][R_3] \tag{18}$$

마찬가지로 공격자는 다음의 6개의 미지수와 타원곡선을 이용한 총 6개의 비선형 방정식을 얻을 수 있다.

$$\begin{aligned} &(R_0+1)Y_PX_{2j}Z_{2j}-(R_0+1)X_PZ_PY_{2j}+B_{00}Z_P^3Y_{2j} \\ &\quad -B_{01}Y_PZ_{2j}^3+B_{02}X_PZ_PZ_{2j}^3-B_{02}Z_P^3X_{2j}Z_{2j}=0, \\ &R_1Y_PX_{2j}Z_{2j}-R_1X_PZ_PY_{2j}+B_{10}Z_P^3Y_{2j} \\ &\quad -B_{11}Y_PZ_{2j}^3+B_{12}X_PZ_PZ_{2j}^3-B_{12}Z_P^3X_{2j}Z_{2j}=0, \end{aligned}$$

$$
\begin{aligned}
&R_2 Y_P X_{2j} Z_{2j} - R_2 X_P Z_P Y_{2j} + B_{20} Z_P^3 Y_{2j} \\
&\quad - B_{21} Y_P Z_{2j}^3 + B_{22} X_P Z_P Z_{2j}^3 - B_{22} Z_P^3 X_{2j} Z_{2j} = 0, \\
&R_3 Y_P X_{2j} Z_{2j} - R_3 X_P Z_P Y_{2j} + B_{30} Z_P^3 Y_{2j} \\
&\quad - B_{31} Y_P Z_{2j}^3 + B_{32} X_P Z_P Z_{2j}^3 - B_{32} Z_P^3 X_{2j} Z_{2j} = 0, \\
&Y_P^2 - X_P^3 + 3 X_P Z_P^4 - b Z_P^6 = 0, \\
&Y_{2j}^2 - X_{2j}^3 + 3 X_{2j} Z_{2j}^4 - b Z_{2j}^6 = 0.
\end{aligned} \tag{19}
$$

여기서, $B_{00} = x_0(R_0+1) + \gamma x_1 R_1$,

$B_{01} = x_0(R_0+1) + x_1 \gamma R_1$,

$B_{02} = (\alpha y_0 + \beta\gamma y_1) R_2 + (\beta\gamma y_0 + \alpha\gamma y_1) R_3$,

$B_{10} = x_0 R_1 + x_1(R_0 - 1)$,

$B_{11} = x_0 R_1 + x_1(R_0 - 1)$,

$B_{12} = (y_0\beta + y_1\alpha) R_2 + (y_0\alpha + y_1\gamma\beta) R_3$,

$B_{20} = x_0 R_2 + \gamma R_3$,

$B_{21} = x_0 R_2 + \gamma R_3$,

$B_{22} = (R_0 - 1) + \gamma R_1$,

$B_{30} = x_0 R_3 + R_2$,

$B_{31} = x_0 R_3 + R_2$.

$B_{32} = R_1 + R_0 - 1$ 이다.

앞서 2절과 동일한 방식으로 resultant 기법을 적용하게 되면, 점 P의 좌표를 바로 찾아낼 수 있기 때문에 별도의 검증과정이 불필요하다.

제안된 공격 기법과 기존 공격[7, 12]간의 차이점은 h_2에 오류가 주입되는 경우에 있다. 논문[12]의 $l_{d+1} = 1$인 경우 $d+1$과 d번째의 두 페어링의 비율의 결과는 h_1함수와 h_2함수의 곱, $h_1 h_2$,로 나타나고, 미지수는 X_P, Y_P, Z_P 그리고 X_j, Y_j, Z_j이다. 따라서 수식 (17)의 h_2함수의 $[2j]P$ 좌표를 $[j]P$로 표현하여 방정식을 풀어야하나 제안하는 공격에서는 $[2j]P$좌표에서의 방정식만 고려하므로 수식이 단순하여 기존 방법에 비해서 풀이가 보다 용이하다고 할 수 있다.

또 하나 고려해야 될 부분은 밀러 알고리듬의 마지막 라운드의 h_2 함수에 오류를 주입할 경우이다. 이 경우 $h_2(Q)$는 $h_2(x, y\sqrt{\nu}) = Z_P^2 x - X_P$의 형태가 되며, 첫 번째 좌표의 부호에 오류를 주입한 후 두 페어링 값의 비율을 앞서와 동일하게 실행한다. 그러면 우리는 다음의 2개의 비선형방정식과 타원곡선 방정식을 이용하여 타원곡선의 점 P를 얻을 수 있다.

$$
\begin{cases}
((R_0+1)x_0 + R_1\gamma x_1) Z_P^2 - (R_0+1) X_P = 0, \\
(R_1 x_0 + (R_0 - 1) x_1) Z_P^2 - R_1 X_P = 0, \\
Y_P^2 - X_P^3 + 3 X_P Z_P^4 + b Z_P^6 = 0,
\end{cases}
$$

여기서 b는 타원곡선 방정식의 매개변수이다.

위의 방정식들을 연립하여 풀면 Z_P에 대한 6개의 후보군들이 존재한다는 사실을 쉽게 증명할 수 있다.

Ⅳ. 기타 좌표계에 대한 오류공격 기법

이 장에서는 Jacobian 좌표계 외에 아핀좌표, 사영(projective)좌표, 그리고 최근에 연구되고 있는 Edward 좌표계 등에서 정의된 타원곡선을 사용하는 경우에 대해서 제안하는 오류공격이 적용가능한지를 살펴본다. 제안하는 공격의 유용성은 doubling의 경우에 대해서 적용하여 검증한다.

먼저 아핀좌표계의 경우 $h_1 = t/v$가 된다. 여기서, t는 직선방정식(tangent line)이고, v는 수직방정식(vertical line)이다. 묻기 차원 $k=4$인 경우를 고려하여 임의의 라운드에서 오류가 주입된다면, Ⅲ장에서 설명한 내용대로 정상 결과 값과 오류 결과 값의 비율을 구하여 다음과 같은 방정식을 유도할 수 있다.

$$
\begin{aligned}
&a_{k,0} x_i^3 + a_{k,1} x_i^2 + a_{k,2} x_i \\
&+ a_{k,3} y_i^2 + a_{k,4} y_i + a_{k,5} = 0
\end{aligned} \tag{20}
$$

여기에서 $a_{k,j}$, $k \in \{0,1,2\}$, $j \in \{0,1,...,5\}$는 알고 있는 상수이다. 세 개의 방정식으로부터 $[j]P=(x_j, y_j)$를 쉽게 복원할 수 있다. 묻기 차원 $k=2$인 경우에는 데이터 오류의 방법과 상관없이 항상 두 개의 방정식을 얻을 수 있으며 이를 이용하여 공격자는 비밀 값을 복구할 수 있다.

두 번째로 사영좌표계의 경우 위의 Jacobian 좌표계의 공격 방법과 동일하게 임의의 d번째 Miller 루프의 첫 번째 원소에서 오류를 주입했다고 가정한다. 이때 그림 1]의 h_1 함수는 다음과 같다.

$$
\begin{aligned}
h_1(x, y\sqrt{\nu}) = 2YZ^3 y\sqrt{\nu} - Z(3X^2 + aZ^2)x \\
- (2bZ^3 - X^3 + aXZ^2)
\end{aligned} \tag{21}
$$

여기서 a, b는 사영공간에 정의된 타원곡선 방정식의 매개변수이다. 두 함수 h_1과 h_1'에 대한 비율을 3.1절의 (11)처럼 고려한다면 우리는 다음과 같은 3개의 비선형 방정식을 유도할 수 있다.

$$
\begin{cases}
R_1 X_j^3 - a R_1 X_j Z_j^2 + P_{00} Z_j^3 + P_{01} X_j^2 Z_j + P_{02} Y_j Z_j^2 = 0, \\
R_2 X_j^3 - a R_2 X_j Z_j^2 + P_{10} Z_j^3 + P_{11} X_j^2 Z_j + P_{12} Y_j Z_j^2 = 0, \\
R_3 X_j^3 - a R_3 X_j Z_j^2 + P_{20} Z_j^3 + P_{21} X_j^2 Z_j + P_{22} Y_j Z_j^2 = 0,
\end{cases} \tag{22}
$$

여기서, $P_{00}=a(x_1R_0-x_0R_1-x_1)-2bR_1$,
$P_{01}=3(x_1(1-R_0)-x_0R_1)$,
$P_{02}=2(R_2(y_0\beta+y_1\alpha)+R_3(y_0\alpha+y_1\gamma\beta))$,
$P_{10}=-a(x_0R_2+x_1\gamma R_3)-2bR_2$,
$P_{11}=-3(x_0R_2+\gamma x_1R_3)$,
$P_{12}=2(y_0(R_0-1)+y_1R_1\gamma)$,
$P_{20}=-a(x_0R_3+x_1R_2)-2bR_3$,
$P_{21}=-3(R_3x_0+R_2x_1)$,
$P_{22}=2(y_0R_1+(R_0-1)y_1)$ 이다.

3장에서 적용한 방법과 동일한 과정을 거치면 최종 $Res(S_1(X_j)_{Y_j},S_2(X_j)_{Y_j})$는 X_j만의 함수로 나타나며 차수가 25임을 확인 할 수 있다. 따라서 $[j]P$의 후보군의 개수는 50개가 되며 비밀 값을 복구할 수 있다. 만약 묻기 차원 $k=2$인 경우에는 정의된 타원곡선 방정식을 이용하여야 하며 $[j]P$의 후보군이 42개가 된다.

세 번째로 Edward 좌표계는 최근 군의 빠른 연산속도로 인해 각광받는 타원곡선의 좌표계로 부채널 공격에 대한 내성을 가지고 있다고 알려져 있다[16]. Edward 좌표계의 효율적인 연산 구현[17]에 따라 두 점의 doubling에 대한 함수식은 다음과 같다.

$$g_d(Q)=c_{Z^2}\eta\sqrt{\nu}+c_{XY}y_0+c_{XZ} \tag{23}$$

여기서

$$\eta=(Z_Q+Y_Q)/X_Q,\ y_0=Y_Q/Z_Q\ \in F_{q^{k/2}}$$

이고,

$$c_{Z^2}=X_j(2Y_j^2-2Y_jZ_j),\ c_{XY}=2Z_j(Z_j^2-aX_j^2-Y_jZ_j),$$
$$c_{XZ}=Y_j(2aX_j^2-2X_jZ_j)\ \in F_q$$

이다.

공격자는 점 Q에 대한 정보를 가지고 있으며, 확장체 $F_{q^{k/2}}$의 기저 $\{1,\xi\}$에 따라 Q의 좌표 값인 η와 y_0값을 각기 $\eta=\eta_0+\eta_1\xi$, $y_0=y_{00}+y_{01}\xi$로 표기한다. Edward 좌표계를 사용하는 타원곡선에 대해서 제안하는 공격 방법을 적용하면 다음과 같은 관계식을 도출할 수 있다.

$$\begin{cases}((R_0-1)y_{01}+R_1y_{00})c_{XY}+R_1c_{XZ}\\ \quad+((R_2+R_3\alpha)\eta_0+(R_2\alpha+R_3\beta)\eta_1)c_{Z^2}=0\ ,\\ ((R_0-1)\eta_0+R_1\gamma\eta_1)c_{Z^2}+R_2c_{XZ}+R_2y_{00}c_{XY}=0,\\ (R_0\eta_1+R_1\eta_0-\eta_1)c_{Z^2}\\ \quad+(R_2y_{01}+R_3y_{00})c_{XY}+R_3c_{XZ}=0.\end{cases} \tag{24}$$

위의 3개의 방정식에서 c_{XY}, c_{Z^2} 그리고 c_{XZ}의 값을 각각 유도 할 수 있다. 유도된 값들을 c_{XY}, c_{XZ}, c_{Z^2}로 두고, Y_j, Z_j의 변수로 가지는 다음 두 개의 방정식으로 재차 유도할 수 있다.

$$4Z_j^2Y_j^5+(2c_{XY}-12Z_j^3)Y_j^4+(12Z_j^4-4c_{XY}Z_j)Y_j^3$$
$$+(4Z_j^2+2Z_j^2c_{XY})Y^2+ac_{Z^2}^2Z_j=0 \tag{25}$$

$$4Z_jY_j^4-8Z_jY_j^3+(4Z_j^3+2c_{XZ}-4c_{XZ}Z_j)Y_j^2$$
$$-2c_{XZ}Z_j^2Y_j-ac_{Z^2}^2=0 \tag{26}$$

Y_j, Z_j에 대한 두 방정식을 resultant 방법을 사용하여 획득한 Y_j 좌표를 기준으로 나머지 좌표 값들을 찾아낼 수 있다. 찾아낸 좌표 값들을 이용한다면 원래의 비밀 값을 복원할 수 있게 된다.

Ⅴ. 시뮬레이션 결과

제안하는 오류주입 공격의 유용성을 검증하기 위해서, [18]에서 사용한 수학 소프트웨어 Mathemetica[19]를 이용하여 Jacobian 좌표계에서의 알고리듬을 구현하였다. 시뮬레이션을 위한 환경은 다음과 같다.

-CPU : Intel Quad CPU 3.8GHz
-RAM : 4GB
-테스트 환경 : Window 7 32bit
-개발 환경 : Wolfram Mathemetica 7

구현한 밀러 알고리듬의 정상 동작 여부를 판별하기 위해서 16비트 크기의 예제 파라미터들을 사용하였다.

타원곡선이 $Y^2=X^3-3XZ+122$로 정의 될 때, 타원곡선이 위치한 유한체 F_p의 파라미터들은 아래와 같다.

$p=7549$
$l=3831=111011110111_2$

밀러 루프에서 사용하는 비밀 값 점 P의 좌표와 공개 값 점 Q의 좌표는 아래와 같다.

$P=[4425,5458,5128]$
$Q=[1826+4304\xi,778\sqrt{v}+5646\xi\sqrt{v}]$

이 때 확장된 유한체 F_{q^k}에서 사용된 묻기 차원의 정도는 $k=4$이며, 기저를 구성하는 기약다항식(irreducible polynomial)은 $a^4=2$으로 a는 F_{q^k}의 한

원소이다. 기약다항식에 의해서 $\xi^2=\gamma=2$, $\nu=\alpha+\beta\xi=0+1\xi$가 된다.

공격 기법의 시뮬레이션에서는 먼저 오류 주입 위치를 가정한다. 본 논문에서는 임의로 루프 횟수가 14일 때, $h_1(Q)$의 연산이 끝난 후, 유한체의 원소를 저장하는 $[a_0][a_1][a_2][a_3]$의 배열 중에 a_0에 오류가 주입된다고 가정하였다. 주입되는 오류의 값은 임의의 값으로 공격자가 알 수 없도록 시뮬레이션 되었다. 시뮬레이션 결과 정상적으로 수행한 출력 값과 연산 중도에 오류가 주입된 경우의 출력 값을 나눈 비율 R을 구하면 아래와 같다.

$$1849+3933\xi+5579\sqrt{\nu}+6032\xi\sqrt{\nu}$$

수식 (13), (14), (15)에 따라 a_1, a_2, a_3를 이용하여 식 (16)의 계수들을 계산 한 후, resultant 방법으로 150개의 좌표 후보군들을 계산 할 수 있다. 그림 2는 계산된 좌표 후보군들의 일부이다.

도출된 [14]P의 좌표 후보군을 이용하여 모듈러스 l 상에서 14의 역수인 4853만큼의 스칼라 곱셈을 통해서 비밀 값의 후보군을 확인한다. 이는 PC상에 구현된 Mathematica로 구현된 페어링 기법을 사용하는데, 계산된 비밀 값의 후보군들에 대해서 l 크기의 스칼라 곱셈을 수행한 후, 그 결과 값이 O인지 여부를 판별하는 방법을 사용하여 후보군들에 대한 진위여부를 판별한다. 판별 과정에서 남은 후보 값들에 대해서는 밀러 알고리듬의 수행을 통해 정상 결과 값과 동일한 값을 출력하는 후보를 찾는 방법을 수행하여 하나의 올바른 비밀 값을 찾게 된다. 따라서 시뮬레이션 결과, 제안된 공격 기법을 적용하게 되면 공격자는 비밀 값을 찾을 수 있음을 확인할 수 있었다.

VI. 결 론

본 논문에서는 이동 ad hoc 네트워크의 보안을 위해 사용되는 페어링 연산에서 가장 중요한 단계인 밀러 알고리듬에 대한 데이터 오류 주입 공격을 제안하였다. 공격을 위해서 유한체의 원소가 기저에 따라 표현되며, 각 기저가 구현시 독립적인 배열로 저장되는 점을 이용하여, 임의의 라운드에서 연산되는 함수의 기저 중 하나인 배열에 오류를 주입하는 방식을 사용한다. 제안하는 공격 방법은 기존의 카운터 오류 주입 공격과 비교하여 1번의 오류 주입을 통해 획득한 오류 결과문을 분석하여 비밀 값인 P의 좌표를 복원할 수 있으며, 기존의 데이터 오류 공격과 달리 특정 라운드에 구애받지 않고 제한 없이 동작하므로 오류주입 공격의 실제 구현이 용이하다. 또한, 컴퓨터 시뮬레이션을 통한 검증을

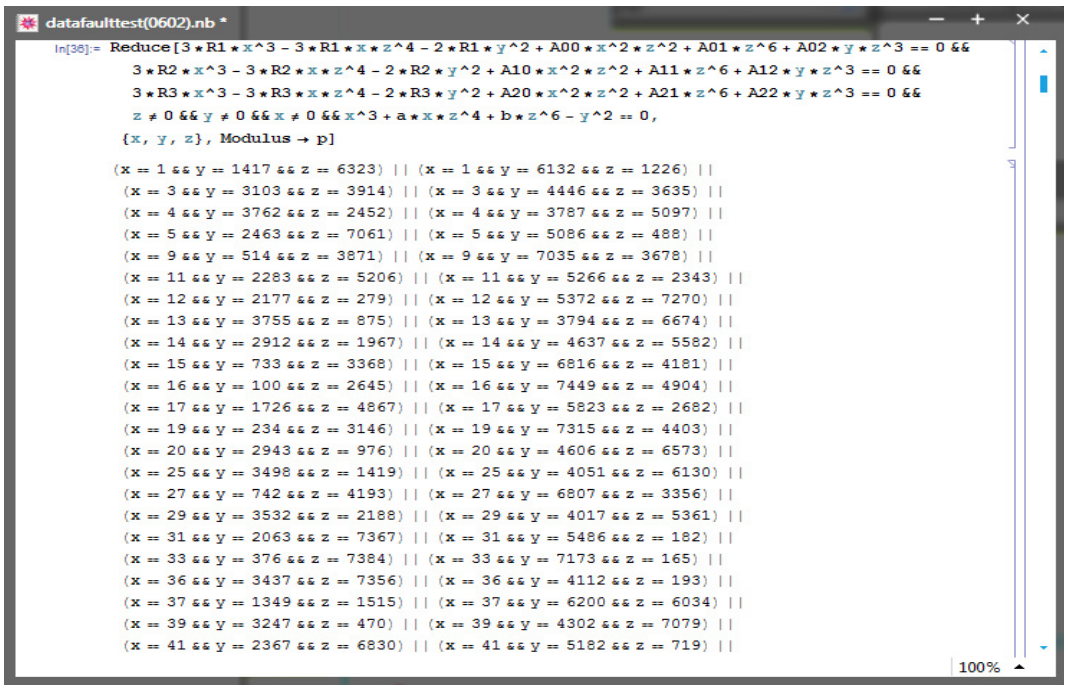


그림 2. 시뮬레이션 결과
Fig. 2. Simulation Result.

거쳐 정상 동작함을 확인하였고, 이동 ad hoc 네트워크의 구성 환경에 따라 Jacobian 좌표계만이 아닌 기타 좌표계를 사용하더라도 공격의 적용이 가능함을 검증하였다.

참 고 문 헌

[1] L. Zhou, and Z. J. Haas, "Securing ad hoc networks," *IEEE Network megazine*, vol.13, no.6, pp. 24 - 30, November/December 1999.

[2] A. Khalili, J. Katz, and W.A. Arbaugh, "Toward secure key distribution in truly ad-hoc networks," In IEEE Workshop : Security and Assurance in Ad hoc Networks, pp. 342-346, 2003.

[3] Tae Hyun KIM, Tsuyoshi Takagi, Dong-Guk Han, Ho Won Kim, and Jongin Lim, "Power Analysis Attacks and Countermeasures on ηT Pairing over Binary Fields," *ETRI Journal*, vol.30, no.1, pp. 68-80, Feb. 2009.

[4] N.E. Mrabet, M.L. Flottes, and G. D. Natale, "A practical Differential Power Analysis attack against the Miller algorithm," Research in Microelectronics and Electronics, PRIME 2009. Ph.D., pp.308-311, July 2009.

[5] N.E. Mrabet, "Fault Attacks against the Miller's Algorithm in Edwards Coordinates," In Information Security and Assurance(ISA 2010), pp. 72-85, Miyazaki, Japan, June 2010.

[6] S. Ghosh, D. Mukhopadhyay, and D. R. Chowdhury, "Fault Attack and Countermeasures on Pairing Based Cryptography," *Internation Journal of Network Security*, vol.12, no.1, pp. 26-33, Jan. 2011.

[7] D. Page and F. Vercauteren, "A Fault Attack on Pairing Based Cryptography," *IEEE Transactions on Computers*, vol.55, no.9, pp. 1075-1080, 2006.

[8] C. Whelan and M. Scott, "The Importance of the Final exponentiation in Pairing when considering Fault Attacks," Proc. of Pairing 2007, pp.225-246, Tokyo, Japan, July 2007.

[9] J. Siverman, "*The Arithmetic of Elliptic Curves*," Springer-Verlag, 1991.

[10] E. Brier, M. Joye, Point multiplication on elliptic curves through isogenies, Proc. of AAECC 2003, LNCS 2643, pp.43-50, 2003.

[11] J.C.Bajard and N.E. Marbet, "Pairing in Cryptography: an arithmetic point of view," Advanced Signal Processing Algorithms, Architectures, and Implementations XVI, part of SPIE, August 2007

[12] N. E. Mrabet, "What about Vulnerability to a Fault Attack of the Miller's Algorithm During an Identity Based Protocol?," In Information Security and Assurance(ISA 2009), pp. 122-134, Seoul, Korea, June 2009.

[13] J.-M. Schmidt and M. Hutter, "Optical and EM Fault-Attacks on CRTbased RSA: Concrete Results," in Austrochip 2007, pp. 61 - 67, Graz, Austria, Oct. 2007.

[14] E. Trichina and R. Korkikyan. "Multi fault laser attacks on protected CRT-RSA," In FDTC 2010, pp. 75~86, Santa Barbara, USA, Aug. 2010.

[15] S. Lang, Algebra, Rev. *3rd Ed., Graduate Texts in Mathematics*, Springer-Verlag, 2002.

[16] D. J. Bernstein and T. Lange. "Faster addition and doubling on elliptic curves," Advanced in Cryptology - ASIACRYPT 2007, pp. 29-50, Kuching, Malaysia, Dec. 2007.

[17] C. Arene, T. Lange, M. Naehrig and C. Ritzenthaler, "Faster computation of the Tate pairing," *Journal of Number Theory*, vol.131, no.5, pp. 842-857, May 2011.

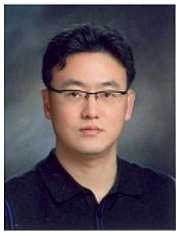
[18] M. Maas, *Pairing-Based Cryptography*, Master Thesis, Technische Universiteit Eindhoven, 2004.

[19] Mathematica, http://www.wolfram.com/products/mathematica/index.html

저 자 소 개

배 기 석(정회원)
2006년 경북대학교 전자전기 컴퓨터학부 학사
2008년 경북대학교 전자전기 컴퓨터학부 석사
2009년～현재 경북대학교 전자 전기컴퓨터학부 박사과정
<주관심분야 : 정보보호, 네트워크 보안, 스마트 카드 보안>

박 영 호(정회원)
1989년 경북대학교 전자공학과 학사
1991년 경북대학교 전자공학과 석사
1995년 경북대학교 전자공학과 박사
1996년～2008년 상주대학교 전자전기공학부 교수
2003년～2004년 Oregon State Univ. 방문교수
2008년～현재 경북대학교 산업전자공학과 교수
<주관심분야 : 정보보호, 네트워크보안, 모바일 컴퓨팅>



손 교 용(정회원)-교신저자
2000년 영남대학교 수학과 이학사
2002년 경북대학교 산업응용 수학과 이학석사
2008년 경북대학교 산업응용 수학과 이학박사
2008년～2010년 충북대학교 전자 정보대학 박사후 연구원
2010년～2012년 경북대학교 전자전기컴퓨터학부 박사후 연구원
2012년～현재 대구교육대학교 수학교육과 조교수
<주관심분야 : (초)타원곡선 암호 이론, 페어링 기반 암호 이론, 정보보호>

문 상 재(평생회원)
1972년 서울대학교 공업교육과 (전자전공) 학사
1974년 서울대학교 전자공학과 공학석사
1984년 미국 UCLA 전기공학과 공학박사
1984년～1985년 미국 UCLA 포스트닥터
1984년～1985년 미국 OMNET 회사 컨설턴트
1974년～현재 경북대학교 IT대학 전자공학부 교수
2002년 2월～현재: 한국정보보호학회 명예회장
<주관심분야 : 무선통신, 네트워크 보안, 암호학>