情報保護學會論文誌
第 21 卷 第 2 號, 2011. 4

RVA 기반의 페어링 부채널 대응법에 대한 안전성 분석*

서 석 충,[1†] 한 동 국,[2‡] 홍 석 희[1]
[1]고려대학교, [2]국민대학교

Security Analysis against RVA-based DPA Countermeasure Applied to Eta_T Pairing Algorithm

Seog Chung SEO,[1†] Dong-Guk Han,[2‡] Seokhie Hong[1]
[1]Korea University, [2]Kookmin University

요 약

최근 타원곡선 상에서의 페어링은 ID 기반의 암호/서명/인증/키공유 기법, 등 다양한 프로토콜에 적용되고 있으며 효율적인 계산을 위하여 $GF(P^n)$ (P = 2, 3) 상에서 정의되는 Eta_T 페어링 알고리즘이 개발되었다. 하지만 Eta_T 페어링 알고리즘의 대칭적인 연산 구조 때문에 Tate, Ate 페어링과 비교하여 부채널 분석에 취약하다. Eta_T 페어링 알고리즘의 부채널 분석에 대한 안전성을 위하여 다양한 대응 방법들이 제안되었다. 특히, 2008년에 Masaaki Shirase 외가 제안한 랜덤값 덧셈 (Random value addition: RVA)을 이용한 대응방법의 경우에는 효율성 측면에서는 뛰어나지만 안전성 측면에서는 취약점이 있다. 본 논문에서는 Eta_T 페어링 알고리즘에서 제안된 RVA 기반의 부채널 대응방법의 취약성에 대하여 실제 구현 관점에서 면밀히 분석한다.

ABSTRACT

Recently, pairings over elliptic curve have been applied for various ID-based encryption/signature/authentication/key agreement schemes. For efficiency, the Eta_T pairings over $GF(P^n)$ (P = 2, 3) were invented, however, they are vulnerable to side channel attacks such as DPA because of their symmetric computation structure compared to other pairings such as Tate, Ate pairings. Several countermeasures have been proposed to prevent side channel attacks. Especially, Masaaki Shirase's method is very efficient with regard to computational efficiency, however, it has security flaws. This paper examines closely the security flaws of RVA-based countermeasure on Eta_T Pairing algorithm from the implementation point of view.

Keywords: Eta_T Pairing, Efficient Countermeasure, DPA

I. 서 론

타원곡선 상의 페어링은 타원곡선 상에서 정의되는 두 점을 유한체 상의 원소로 매핑하는 함수로서 bilinearity와 non-degeneracy를 특징으로 한다. 최근 페어링 연산의 bilinearity와 non-degeneracy 성질을 이용하는 ID 기반의 암호/서명/인증/키공유 기법 등 다양한 프로토콜들이 개발되고 있다. 하지만 페어링 암호 연산은 기존의 타원곡선 암호에 비하여 상대적으로 큰 부하를 발생시키기 때문에 실질적인 프로토콜에의 적용을 위하여 효율적인 페어링 암호에 대한 연구가 진행되어왔다. 초기에는 Tate 페어링 연산을 최적화하는 것에 대한 연구가 진행되었다. 즉, Barreto[2] 외와 Galbraith[3] 외는 Miller 루

접수일(2010년 5월 20일), 수정일(2010년 8월 13일), 게재확정일(2010년 9월 13일)
* 이 연구에 참여한 연구자(의 일부)는 '2단계 BK21 사업'의 지원비를 받았음.
"이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (20100024870)"
† 주저자, ssc2007@korea.ac.kr
‡ 교신저자, hsh@cist.korea.ac.kr

Algorithm 1. $GF(3^n)$상에서의 Eta_T 페어링 계산 알고리즘 $(n \equiv 1 \pmod{12})$[9][12]

INPUT: $P=(x_P, y_P), Q=(x_Q, y_Q) \in E^b(GF(3^n))$
OUTPUT: $Eta_T(P,Q)^M \in GF(3^{6n})$

1. **if** $b'=1$ **then** $y_P \leftarrow -y_P$.
2. $V \leftarrow -y_P(x_P+x_Q+b)+y_Q\sigma+y_P\rho$. (1M)
3. $d \leftarrow b$.
4. **for** $i \leftarrow 0$ to $(n-1)/2$ do
5. $t \leftarrow x_P+x_Q+d$.
6. $R \leftarrow -t^2+y_Py_Q\sigma-t\rho-\rho^2$. (2M)
7. $V \leftarrow R \cdot V$. (13M)
8. $y_P \leftarrow -y_P$.
9. $x_Q \leftarrow x_Q^9, y_Q \leftarrow -y_Q^9$. (4C)
10. $V \leftarrow V^3$. (6C)
11. $d \leftarrow d-b \pmod 3$.
12. **end for**
13. **Return**
$(V^M,\ M=(3^{3n}-1)(3^n+1)(3^n+1-b'3^{(n+1)/2}))$.

프 연산에서의 denominator 제거 방법 및 $GF(3^n)$상에서의 효율적인 타원곡선 Tripling 연산 및 페어링 연산을 제안하였다. Duursma and Lee 외는 위수 3인 유한체 상에서 페어링 연산을 구성하는 타원곡선 연산과 divisor 계산을 효율적인 closed formula로 구성하였다[4]. Barreto 외는 supersingular 커브 상에서 Duursma-Lee 방법을 이진체에 적용하여 유사한 결과를 도출하였으며 뿐만 아니라 기존의 Duursma-Lee 방법에서의 Miller 루프 횟수를 절반으로 줄인 Eta_T 페어링 방법을 일반화하여 제시하였다[9]. Hess 외는 supersingular 커브에서 제안된 Eta_T 페어링 방법을 일반적인 커브로 확장한 Ate 페어링을 제안하였다[8].

부채널 분석 (Side Channel Analysis)은 암호 알고리즘이 특정 장치에서 동작할 때 의도하지 않게 발생되는 내부 데이터 및 키와 연관된 부가적인 정보 (시간 정보, 전력 소모량, 전자기파 등)를 이용하여 암호 시스템의 비밀 정보를 알아내는 기법이다[1]. 페어링 알고리즘에 대한 부채널 분석에 대한 연구는 Page 외[5], Scott[6], Whelan 외[7], Kim 외[10], Choi 외[11] 등에 의하여 수행되어왔다. 타원곡선 암호와는 달리 페어링 알고리즘의 Miller 루프에서 사용되는 스칼라는 비밀 정보가 아니기 때문에 페어링 암호에서의 부채널 분석은 비밀키 역할을 하는 타원곡선 점을 알아내는 것을 목표로 한다. 즉, 페어링 연산에서의 유한체 연산에 단순전력분석 (SPA: Simple Power Analysis), 차분전력분석 (DPA: Differential Power Analysis) 혹은 상관계수전력 분석 (CPA: Correlation Power Analysis)[14]을 적용하여 연산되는 타원곡선 점의 값을 알아낸다. 현재까지 제안된 페어링 암호에서의 부채널 분석 대응방법들은 크게 세 가지로 분류할 수 있다. 첫 번째는, 페어링 연산의 bilinearity를 이용하는 방법으로서, Page외는 타원곡선 점을 랜덤점으로 blinding하기 위하여 bilinearity 성질을 이용하였다[5]. 두 번째는 페어링 연산의 중간 변수들에 랜덤한 값을 곱하는 방법으로서 가장 많이 연구된 방법이다. Scott은 Tate 페어링 계산 시에 SPA와 DPA를 방어하기 위한 방법으로 Miller 루프의 연산에 사용되는 변수들에 랜덤변수를 곱하는 방법을 제안하였으며[6], Kim 외[10]와 Whelan 외 [7]는 $GF(2^n)$상의 Eta_T 페어링의 Miller 루프의 연산에 사용되는 모든 중간 변수들에 랜덤 값을 곱하는 방법을 제안하였다. Kim 외는 Miller 루프의 중간 변수들을 랜덤화하기 위하여 randomized projective coordinate (RPC)방법을 적용하였다[11]. Choi 외는 Kim의 방법을 개량하여 좀 더 적은 연산량으로 동일한 방법의 대응방법을 제시하였다. 마지막 세 번째 방법은 Shirase 외가 제안한 방법으로서 $GF(3^n)$상의 Eta_T 페어링 알고리즘의 부채널 대응방법으로서 랜덤 값을 더하는 방법 (Random Value Addition: RVA)이다[12]. RVA 기반의 대응방법은 $GF(2^n)$상의 Eta_T 페어링 알고리즘에 쉽게 적용될 수 있다. 하지만, 이 방법은 효율성 측면에서는 매우 뛰어나지만 Miller 루프의 연산 과정에서 랜덤값이 상쇄되어 중간 결과값이 노출되는 취약성이 있다.

본 논문에서는 Shirase가 제안한 RVA 기반의 대응방법에 대하여 소개하고 이 대응방법의 부채널 분석에의 취약성에 대하여 분석한다. 알고리즘 내에서 분석의 대상이 되는 위치를 식별하고 어떤 이유에서 분석이 되는지를 실제 구현 관점에서 기술한다.

본 논문의 나머지는 다음과 같이 구성된다. 2장에서 Eta_T 알고리즘에서 RVA 기반의 부채널 대응방법에 대하여 알아보고 기존에 제안된 다른 대응방법들과 연산량 관점에서 비교한다. 3장에서는 RVA 기반의 대응 방법의 보안 취약성을 실제 구현관점에서 기술하고 안전한 RVA 기반의 대응방법의 설계에 대한 가이드라인을 제시한다.

II. $GF(3^n)$상에서 정의되는 Eta_T 페어링 및 RVA 기반의 대응 방법

$GF(3^n)$상에서의 Eta_T 페어링 계산 방법에 대하여

간략히 알아본 후 Eta_T 페어링 계산에서 부채널 분석을 방어하기 위하여 제안된 RVA 기반의 대응방법에 대하여 알아본다. 또한, 기존에 제안된 Eta_T 페어링에서의 부채널 대응방법들과 연산량 관점에서 비교한다.

2.1 GF(3^n)상에서 정의되는 Eta_T 페어링

Algorithm 1은 ($n \equiv 1 \pmod{12}$경우) Eta_T 페어링 계산 알고리즘을 나타낸 것이며 각 단계 당 연산량을 나타내었다. GF(2^n)상에서의 Eta_T 페어링 계산 알고리즘 역시 Algorithm 1과 유사한 divisor 계산, 확장체 곱셈, 포인트 갱신의 구조를 가진다 (좀 더 자세한 설명을 위하여 [9]를 참조하라). M과 C는 각각 $GF(3^n)$상에서의 곱셈과 cubing 연산을 의미하여 덧셈 (뺄셈)의 경우에는 연산부하가 곱셈과 cubing에 비하여 매우 작기 때문에 제외하였다. 따라서, 과정 15의 final powering을 제외한 Algorithm 1의 연산량은 약 (7.5n+8.5)M+(5n+5)C가 된다. 특히, 과정 9는 $GF(3^{6n})$상의 두 원소인 $R=(-t^2, y_P/y_Q, -t, 0, -1, 0)$과 $V=(v_0, v_1, v_2, v_3, v_4, v_5)$의 곱셈이기 때문에 13M의 연산량이 소모된다.

2.2 Eta_T 페어링 알고리즘에서의 부채널 분석

Eta_T 페어링 알고리즘에서의 부채널 분석 대상은 덧셈, 곱셈과 같은 유한체 연산이다. 먼저 $Eta_T(P, Q)$의 계산에서 $P=(x_P, y_P)$를 비밀키 역할을 하는 고정된 타원곡선 점으로, $Q=(x_Q, y_Q)$를 공개키 역할을 하는 타원곡선 점으로 가정한다. 위의 가정을 기반으로 Algorithm 1의 과정 2와 5의 (x_P+x_Q+d) 연산과 과정 6의 (y_P/y_Q) 연산은 (고정된 비밀정보 + 입력 가능한 공개된 정보) 혹은 (고정된 비밀정보 × 입력 가능한 공개된 정보)의 형태이기 때문에 DPA/CPA 분석 방법을 이용하여 x_P 혹은 y_P의 값을 알아낼 수 있다. 즉, (x_P+x_Q+d)의 연산에서 x_P는 $x_P=\sum_{i=0}^{n-1} 3^i x_p^i$와 같이 표현된다. GF($3^n$)상에서의 덧셈 연산의 경우에 하위 워드에서의 연산 결과 상위 워드로 캐리가 전파되는 경우가 존재하지 않기 때문에, x_P의 최하위 워드부터 DPA/CPA 분석을 수행할 수 있다. 자세한 분석 방법과 유한체 곱셈 연산에 대한 전력분석 방법에 대한 시나리오가 [7]과 [10] 논문에 기술되어있다.

x_P와 y_P의 값 중 하나의 값이라도 노출되면, 타원곡선 방정식을 풀어서 나머지 값을 복원해내는 것이 가능하다. 따라서, Eta_T 페어링 계산에서 DPA를 방어하기 위해서는 공격자가 알고리즘 내부에서 계산되는 중간 결과값을 예측하지 못하도록 중간 내부 변수를 랜덤화해야 한다.

2.3 RVA 기반의 부채널 대응 방법

기존에 제안되었던 RPC 기반의 대응방법은 Miller 루프에서의 중간 변수들을 랜덤화하기 위하여 $Eta_T(P, Q)$의 입력인 P 점 혹은 Q 점을 randomized projective coordinate 방법을 이용하여 랜덤화한다. 즉, Kim 외 및 Choi 외는 아핀 좌표계로 표현된 $Q=(x_Q, y_Q)$점을 $(X_Q, Y_Q, Z_Q)=(\lambda x_Q, \lambda y_Q, \lambda)$ (λ: 랜덤값)와 같이 사영좌표계로 표현하여 GF(2^n)상에서의 Eta_T 페어링 계산식을 제안하였다[10, 11]. 입력으로 들어온 Q점을 랜덤화시키기 때문에 공격자는 Eta_T 페어링의 중간 결과값을 올바르게 예측할 수 없을 뿐만 아니라, Miller 루프의 모든 중간 변수가 랜덤값인 λ에 의하여 blinding되기 때문에 공격자는 중간 결과값에 연관된 정보를 얻을 수 없다. 하지만, 모든 중간 결과가 랜덤값 λ가 곱해진 형태가 되기 때문에, 연산의 결과가 항상 1로 고정되어있던 항까지 랜덤한 값으로 대체 된다. 따라서, 페어링 결과의

Algorithm 2. GF(3^n)상에서의 RVA 기반의 Eta_T 페어링 계산 알고리즘 ($n \equiv 1 \pmod{12}$)[12]

INPUT: $P=(x_P, y_P), Q=(x_Q, y_Q) \in E^b(GF(3^n))$
OUTPUT: $Eta_T(P, Q)^M \in GF(3^{6n})$

1. **if** $b'=1$ **then** $y_P \leftarrow -y_P$.
2. **Prepare random** $\lambda, \lambda' \in GF(3^n)$.
3. $Y_P \leftarrow \lambda' y_P$, $Y_Q \leftarrow \lambda' y_Q$. **(2M)**
4. $x_P' \leftarrow x_P+\lambda, y_P' \leftarrow y_P+\lambda$, .
$x_Q' \leftarrow x_Q-\lambda, y_Q' \leftarrow y_Q-\lambda, \lambda'' \leftarrow \lambda^2$ **(1M)**
5. $V \leftarrow -Y_P(x_P'+x_Q'+b)+Y_Q\sigma+Y_P\rho$. **(1M)**
6. $d \leftarrow b$.
7. **for** $i \leftarrow 0$ **to** $(n-1)/2$ **do**
8. $t \leftarrow x_P'+x_Q'+d$.
9. $R \leftarrow -(t+\lambda)(t-\lambda)-\lambda''+$.
$(y_P'y_Q'+\lambda(y_P'-y_Q'-\lambda))\sigma-t\rho-\rho^2$. **(3M)**
10. $V \leftarrow R \cdot V$. **(13M)**
11. $x_P' \leftarrow x_P'-\lambda+\lambda^9, y_P' \leftarrow -y_P'+\lambda+\lambda^9$. **(2C)**
12. $x_Q' \leftarrow x_Q'^9, y_Q' \leftarrow y_Q'^9, \lambda \leftarrow \lambda^9, \lambda'' \leftarrow \lambda''^9$. **(6C)**
13. $V \leftarrow V^3$. **(6C)**
14. $d \leftarrow d-b \pmod 3$.
15. **end for**
16. **Return**
$(V^M,\ M=(3^{3n}-1)(3^n+1)(3^n+1-b'3^{(n+1)/2}))$.

sparse한 성질을 이용한 기존의 효율적인 확장체 곱셈 방법을 더 이상 적용할 수 없게 된다. 즉, 기존의 연산인 $R=(y_P+\theta+u, x_P+v+1, 1, 0)$가 랜덤한 값인 Z_Q로 blinding되기 때문에 기존의 1이었던 항이 Z_Q가 된다. 따라서, 확장체 곱셈인 $R \bullet V$의 연산량이 6M에서 8M으로 증가하게 된다. RPC 기반의 대응방법은 R의 모든 항을 Z_Q로 blinding함으로써 중간 연산값을 공격자가 조작하지 못하도록 하고 final powering의 결과 Z_Q의 영향이 1이 되도록 한다. 따라서, RPC 기반의 대응방법은 $R \bullet V$의 연산량을 필연적으로 증가시킨다.

2008년 Shirase는 기존의 Eta_T 페어링 알고리즘에서의 부채널 대응방법의 문제점인 과중한 연산 부하를 해결하기 위하여 중간 변수에 랜덤값을 곱하는 방법이 아닌 부채널 분석의 대상이 되는 위치에 랜덤값을 더하는 방식을 이용한 대응방법을 제시하였다[12]. Shirase 외는 $GF(3^n)$ 상에서 정의되는 Eta_T 페어링 연산에서 RVA 기반의 대응방법을 제안하였다. RVA 기반의 대응방법은 입력 점들인 $P=(x_P, y_P)$와 $Q=(x_Q, y_Q)$에 대하여 $GF(3^n)$상에서의 랜덤값인 λ을 더하여 $P'=(x_P+\lambda, y_P+\lambda)$와 $Q'=(x_Q-\lambda, y_Q-\lambda)$로 변환한다. Algorithm 2는 Shirase 외가 제안한 $GF(3^n)$상에서 정의되는 Eta_T 페어링 알고리즘에 RVA 기반의 대응방법을 적용한 것이다. 과정 3에서 y_P와 y_Q에 λ'를 곱하는 이유는 매 루프마다 갱신되는 페어링 계산값인 V를 과정 5에서 λ'값으로 blinding하기 위해서이다. 즉, 과정 5의 결과 V의 모든 항은 랜덤값인 λ'가 곱해져있는 형태가 된다. blinding된 λ'값은 과정 16에서 final powering의 결과 1이 되기 때문에 연산 결과에 영향을 미치지 않는다. 과정 5, 8, 9에서 사용되는 중간 변수들은 모두 랜덤값으로 마스킹되어 있다. RVA 기반의 대응방법은 중간 변수에 랜덤값이 곱해진 형태가 아니라 더해진 형태이기 때문에 보정과정을 거치지 않고 과정 10를 수행할 경우 올바른 결과를 얻을 수 없다. 즉, RPC 기반의 대응방법의 경우 λ가 곱해져 있기 때문에 확장체 곱셈 연산을 수행 시 $\lambda(R \bullet V)$가 되지만, RVA 기반의 대응방법은 R에 랜덤값이 더해진 형태이기 때문에 확장체 곱셈을 수행하였을 때 올바른 값을 유지하지 못한다. 따라서, Shirase는 과정 9에서 보정과정을 수행하여 마스킹 값을 상쇄시킨다. 보정 과정을 거친 연산 결과값은 과정 10의 입력으로 사용되어 올바른 결과값으로 계산된다. Shirase가 제안한 RVA 기반의 대응방법의 주된 부하는 과정 9에서의 보정 연산에서 발생한다. 즉, 과정 9의 $y_P'y_Q'$의 연산의 결과는 실제로 $(y_P+\lambda)(y_Q-\lambda)$이기 때문에 추가적으로 $\lambda(y_P'-y_Q'-\lambda)$의 결과를 더하여 올바른 결과인 y_Py_Q를 계산해낸다. Shirase 외가 제안한 방법의 연산량은 약 $(8n+12)M+(7n+7)C$가 된다.

Shirase 외가 제안한 $GF(3^n)$ 상에서 제안한 RVA 기반의 대응방법은 $GF(2^n)$상에서의 Eta_T 페어링 계산에도 쉽게 적용 가능하다. $GF(2^n)$상의 Eta_T 페어링 계산에 적용될 경우 추가적인 연산 부하가 약 0.5nM 정도 되기 때문에 성능측면에서 기존에 제안된 RPC 기반의 대응방법들 보다 뛰어나다. 〔표 1〕은 Eta_T 페어링 알고리즘에서 기존에 제안된 대응방법들과 RVA 기반의 대응방법을 연산량 측면에서 비교한다. Shirase가 제안한 RVA 기반의 대응방법의 부하는 $(0.5n+3.5)M+(2n+2)C$로서 기존의 RPC 기반의 대응방법들에 비하여 약 75%의 곱셈 수를 줄인 것을 확인할 수 있다.

〔표 1〕 Eta_T Pairing에서 부채널 대응 방법 비교(M, S, I는 각각 유한체 곱셈, 제곱, 역원 연산을 의미한다).

	추가적인 연산 부하
Page-Vercauteren 〔5〕	(2n)M+(9n)S+(I)n
Page-Vercauteren 〔5〕	(3.5n+72.5)M+(4n+4)C
Scott 〔6〕	(3.5n+7.5)M
Kim 외 〔10〕	(2.5n+6.5)M+(0.5n+0.5)S
Choi 외 〔11〕	(2n+6)M+(0.5n+0.5)S
Shirase 외 〔12〕	(0.5n+3.5)M+(2n+2)C

III. RVA 기반의 부채널 대응방법의 보안 취약성

$GF(3^n)$상에서의 Eta_T 페어링 계산 방법에 대하여 간략히 알아본 후 Eta_T 페어링 계산에서 부채널 분석을 방어하기 위하여 제안된 RVA 기반의 대응방법에 대하여 알아본다. 또한, 기존에 제안된 Eta_T 페어링에서의 부채널 대응방법들과 연산량 관점에서 비교한다.

3.1 RVA 기반의 대응 방법의 취약성

Shirase 외가 제안한 RVA 기반의 대응방법은 Algorithm 1의 과정 2, 5, 6, 7을 부채널 분석의 대상으로 간주하고 위의 연산 과정에서 중간값들의 노출을 방지하는 것을 목표로 하였다. Algorithm 1의 과

정 2, 5, 6, 7은 RVA 방법을 적용한 Algorithm 2의 과정 5, 8, 9, 10로 대응된다. 과정 5, 8, 9에서의 연산을 자세히 살펴보면 연산을 수행하는 순간에는 중간 결과값이 노출되지 않으나, 각 과정에서의 연산을 끝낸 후의 결과값은 랜덤값이 상쇄된 형태라는 것을 확인할 수 있다. 과정 5, 8, 9에서의 취약성을 기술한다.

과정 5 ($V \leftarrow\!- Y_P(x_P' + x_Q' + b) + Y_Q\sigma + Y_P\rho$)의 $(x_P' + x_Q')$의 연산은 아래와 같은 어셈블리 코드를 사용하여 구현될 수 있다. 아래의 코드에서 $x_P' = (x_P^{H'} \| x_P^{L'})$, $x_Q' = (x_Q^{H'} \| x_Q^{L'})$로 구성되어 있다. 즉, GF(3)상에서 하나의 원소는 컴퓨터 상에서 두 비트로 표현되기 때문에, 이를 효율적으로 표현하기 위하여 상위 비트는 H 부분에, 하위 비트는 L 부분에 저장하는 방법을 사용한다[13].

```
1. Loop:
2. tst      R2
3. breq     Exit
4. dec      R2
5. mv       X,          R15
6. ld       R4,         X+ // R4 = x_P^{L'}[i]
7. mv   R15,            X
8. mv   X, R14
9. ld       R5,         X+ // R5 = x_P^{H'}[i]
10. mv R14,             X
11. mv      Y,          R17
12. ld      R6,         Y+// R6 = x_Q^{L'}[i]
13. mv R17,             Y
14. mv      Y,          R16
15. ld      R7,         Y+ // R7 = x_Q^{H'}[i]
16. mv R16,             Y
17. mv      R10,        R4
18. or      R10,        R6 // R10 = (x_P^{L'}[i] | x_Q^{L'}[i])
19. mv R11,             R5
20. or      R11,        R7 // R11 = (x_P^{H'}[i] | x_Q^{H'}[i])
21. or      R5,         R6 // R5 = (x_P^{H'}[i] | x_Q^{L'}[i])
22. or      R7,         R4 // R7 = (x_Q^{H'}[i] | x_P^{L'}[i])
23. eor R5,             R7
    // R5 = T = (x_P^{H'}[i] | x_Q^{L'}[i]) ^ (x_Q^{H'}[i] | x_P^{L'}[i])
24. eor R10,            R5
    // R10 = T ^ (x_P^{L'}[i] | x_Q^{L'}[i])
25. eor R11,            R5
    // R11 = T ^ (x_P^{H'}[i] | x_Q^{H'}[i])
26. mv Z, R18
27. st      Z+,         R10
    // CH[i] = R10 = T ^ (x_P^{L'}[i] | x_Q^{L'}[i])
28. mv R18,             Z
29. mv      Z,          R19
30. st      Z+,         R11
    // CL[i] = R11 = T ^ (x_P^{H'}[i] | x_Q^{H'}[i])
31. mv R19              Z
32. Exit:
```

위의 코드에서 X, Y, Z는 각각 메모리에서 x_P', x_Q' 그리고 임시적으로 $(x_P' + x_Q')$의 결과값이 저장되는 주소를 저장하는 주소 레지스터이다. x_P', x_Q'가 $GF(3^n)$의 원소이기 때문에 $x_P' = (x_P^{H'} \| x_P^{L'})$, $x_Q' = (x_Q^{H'} \| x_Q^{L'})$로 표현되며, $x_P^{H'}$, $x_P^{L'}$, $x_Q^{H'}$, $x_Q^{L'}$의 각각은 메모리 공간에서 $t = \lceil n/WORDSIZE \rceil$의 공간을 차지한다. 따라서, 위의 코드는 [13]에서 기술된 GF(3)에서의 덧셈연산 방법을 통하여 x_P', x_Q'의 하위 워드에서부터 덧셈을 수행하여 그 결과값을 메모리에 저장한다. 이 과정을 순차적으로 주소의 변위를 변화시키면서 t번 수행한다. 위의 코드 중 과정 6, 9, 12, 15에서 메모리로부터 버스를 거쳐 레지스터에 데이터를 로딩하는 과정에서는 x_P와 x_Q의 값이 랜덤값인 λ로 masking 되었기 때문에 이때 소모되는 전력은 공격자에게 유용한 정보가 되지 못한다. 하지만 과정 27, 30에서는 masking 값이 상쇄된 $(x_P + x_Q)$가 레지스터에서 버스를 거쳐 메모리에 저장된다. 이때 소모되는 전력은 $(x_P + x_Q)$의 값과 상관성을 가지게 된다. 따라서 공격자는 많은 수의 Q 점을 $Eta_T(P, Q)$의 입력으로 넣어 과정 27, 30에서 x_P의 값을 추측하는 시뮬레이션을 수행할 수 있게 된다.

과정 10에서도 동일한 방법의 분석이 가능하다. 즉, 과정 10의 $(t \leftarrow x_P' + x_Q' + d)$ 연산 역시 위에서 기술한 어셈블리 코드로 구현되기 때문에 위에서와 동일한 취약성을 이용하여 분석이 가능하다.

과정 11에서는 두 곳에서 취약성이 노출된다. 첫 번째 위치는 $(t+\lambda)(t-\lambda)$를 연산하는 것으로서 중간 결과값이 노출되지 않도록 t값에 랜덤값을 더한 형태의 $(t+\lambda)(t-\lambda)$로 계산을 수행하지만, t값이 이미 과정 10에서 masking이 상쇄된 $(x_P + x_Q)$로 메모리에 저장되었기 때문에 연산을 위하여 레지스터로 로딩될 때, $(x_P + x_Q)$의 값과 연관된 전력소모를 발생시킨다. 두 번째 위치는 $(y_P' y_Q' + \lambda(y_P' - y_Q' - \lambda))$를 연산하는 것으로서 이 연산의 결과는 masking이 상쇄된 $y_P y_Q$가 되기 때문에 이 값이 버스를 통하여 메모리에 저장될 때 발생하는 전력소모 정보는 분석에 이용될 수 있다.

따라서, Shirase가 제안한 RVA 기반의 부채널 대응 방법은 연산 과정 중에는 중간 결과값을 랜덤한 값으로 masking하고 있지만, 연산을 마친 후에 저장되는 값은 masking 값이 상쇄된 형태로 저장된다. 따라서, 연산 결과가 버스를 통하여 메모리에 저장되거나 혹은 masking이 상쇄된 형태로 메모리에 저장되었던 중간 결과값이 다른 연산에서 재사용될 때 발

생하는 전력 소모의 특성을 이용하여 공격자는 분석을 수행할 수 있다. 이 취약성은 근본적으로 Shirase가 제안한 대응 방법이 원래의 Eta_T 페어링 계산 결과와 동일하도록 만들기 위하여 $P=(x_P+\lambda, y_P+\lambda)$, $Q=(x_Q-\lambda, y_Q-\lambda)$와 같이 대칭적으로 설정한 것에서 기인한다.

3.2 안전한 RVA 기반의 대응방법 설계 가이드라인

Eta_T 페어링 계산에서 RVA 기반의 부채널 대응방법을 안전하도록 설계하기 위해서는 다음의 세 시점에서 모두 마스킹 값이 유지되어야 한다.

① 연산이 적용되는 데이터를 메모리에서 레지스터로 로딩되는 시점
② 실제 CPU 상에서 연산이 수행되는 시점
③ 연산 결과값이 레지스터에서 메모리로 저장되는 시점

Shirase가 제안한 RVA 방법의 경우 2)의 조건은 보장되었으나 1)과 3)의 조건이 보장되지 않았다. 따라서, 안전한 RVA 기반의 대응방법을 설계하기 위해서는 먼저, P와 Q 점을 랜덤값을 이용하여 masking 할 때 대칭적인 값을 사용하는 것이 아니라, 비대칭적인 값을 사용하여 masking을 수행해야한다. 연산의 결과 masking이 상쇄되지 않아야 3)의 조건을 보장할 수 있으며 나아가 향후 해당 연산 값이 재사용될 때 1)번 조건을 충족시킬 수 있다. 하지만, 연산의 결과 masking 값이 상쇄되지 않아 기존의 연산 결과값에 더해져있는 경우 확장체 곱셈 연산에서 올바른 값이 계산되는 것을 보장할 수 없다. 따라서, RVA 기반 대응방법의 장점을 유지하면서 위에서 제시한 기준을 만족하도록 설계하기 위해서는 확장체 곱셈 연산에서 올바른 보정 과정을 수행하도록 해야 하며 이것은 향후 과제로 남긴다.

IV. 결 론

본 논문에서는 $GF(3^n)$상의 Eta_T 페어링 알고리즘의 부채널 대응방법으로서 랜덤 값을 더하는 방법을 사용하는 RVA 기반의 Eta_T 페어링 알고리즘의 안전성을 실제 구현관점에서 분석하였다. RVA 기반의 대응방법은 연산 측면에서는 기존에 제안된 RPC 기반의 대응방법보다는 효율적이지만, 연산의 결과가 masking이 상쇄된 형태로 메모리에 저장되거나, 메모리로부터 레지스터로 로딩되어 재사용되기 때문에 부채널 분석에 취약하다. 본 논문에서는 안전한 RVA 기반의 대응방법 설계 가이드라인을 제시하였으며 향후에는 이를 준수하는, $GF(3^n)$과 $GF(2^n)$상에의 Eta_T 페어링 알고리즘에서 실제적으로 적용할 수 있는 대응방법을 설계할 것이다.

참고문헌

〔1〕 P. Kocher, "Differential Power Analysis," CRYPTO 1999, LNCS 1666, pp. 388-397, 1999.

〔2〕 P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient Algorithms for Pairing-based Cryptosystems," CRYPTO 2002, LNCS 2442, pp. 354-368, 2002.

〔3〕 S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate Pairing," ANTS V, LNCS 2369, pp. 324-337, 2002.

〔4〕 I. Duursma and H. S. Lee, "Tate Pairing Implementation for Hyperelliptic Curves $y^2=x^p-x+d$," Asiacrypt 2003, LNCS 2894, pp. 111-123, 2003.

〔5〕 D. Page and F. Vercauteren, "Fault and Side-Channel Attacks on Pairing Based Cryptography," Cryptology ePrint Archive, Report 2004/283, 2005.

〔6〕 M. Scott, "Computing the Tate Pairing", CT-RSA 2005, LNCS 3376, pp. 293-304, 2005.

〔7〕 C. Whelan and M. Scott, "Side Channel Analysis of Practical Pairing Implementations: Which Path is More Secure?," VIETCRYPT 2006, LNCS 4341, pp. 99-114, 2006.

〔8〕 F. Hess, N. Smart, and F. Vercauteren, "The Eta Pairing Revisited," IEEE Trans. Inf. Theory. 52 no. 10 pp. 4595-4602, 2006.

〔9〕 P. S. L. M. Barreto, S. D. Galbraith, Colm Ó' hÉigeartaigh, and M. Scott, "Efficient Pairing Computation on Supersingular Abelian Varieties," Design Codes and Cryptography, 42(3), pp. 239-271, 2007.

〔10〕 TaeHyun. Kim, T. Takagi, Dongguk Han, Howon Kim and Jongin Lim, "Power Analysis Attacks and Countermeasures on Pairing over Binary Fields," ETRI Journal, Volume 30, Number 1, pp. 68-80, 2008.

〔11〕 Dooho Choi, DongGuk Han, and Howon Kim, "Construction of Efficient and Secure Pairing Algorithm and Its Application," Jounal of Communications and Networks, Vol. 10, No. 4, pp. 437-443, 2008.

〔12〕 M. Shirase, T. Takagi, and E. Okamoto "An Efficient Countermeasure against Side Channel Attacks for Pairing Computation," ISPEC 2008, LNCS 4991, pp. 290-303, 2008.

〔13〕 K. Harrison, D. Page, and N. P. Smart, "Software Implementation of Finite Fields of Characteristic Three, for Use in Pairing-Based Cryptosystems," London Mathematical Society, Journal of Computer Math, Vol. 5, pp. 181-193, 2002.

〔14〕 Eric Brier, Christophe Clavier and Francis Olivier, "Correlation Power Analysis with a Leakage Model," CHES 2004, LNCS 3156, pp. 135-152, 2004.

〈著者紹介〉

서 석 충 (Seog Chung Seo) 학생회원
2005년 2월: 아주대학교 정보 및 컴퓨터 공학과 학사
2007년 2월: 광주과학기술원 정보통신 공학과 석사
2007년 9월~현재: 고려대학교 정보경영공학전문대학원 박사 과정
〈관심분야〉 공개키 암호 고속 구현, 센서 네트워크 보안, 부채널 분석, 암호 알고리즘 병렬화 구현

한 동 국 (Dong-Guk Han) 정회원
1999년: 고려대학교 수학과 졸업(학사)
2002년: 고려대학교 수학과 석사 (이학석사)
2005년: 고려대학교 정보보호대학원 박사 (공학박사)
2004년 4월~2005년 4월: 일본 Kyushu Univ., 방문연구원
2005년 4월~2006년 4월: 일본 Future Univ.-Hakodate, Post.Doc.
2006년 6월~2009년 2월: 한국전자통신연구원 정보보호연구단 선임연구원
2009년 3월~현재: 국민대학교 수학과 조교수
〈관심분야〉 공개키 암호시스템 안전성 분석 및 고속 구현, 부채널 분석, RFID/USN 정보보호 기술

홍 석 희 (Seokhie Hong) 종신회원
1995년: 고려대학교 수학과 학사
1997년: 고려대학교 수학과 석사
2001년: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주)시큐리티 테크놀로지스 선임연구원
2003년 3월~2004년 2월: 고려대학교 시간강사
2004년 4월~2005년 2월: K.U. Leuven 박사후연구원
2005년 3월~2008년 8월: 고려대학교 정보경영전문대학원 조교수
2008년 9월~현재: 고려대학교 정보경영전문대학원 부교수
〈관심분야〉 대칭키 암호 알고리즘, 공개키 암호 알고리즘, 포렌식