情報保護學會論文誌
第19卷 第4號, 2009. 8

타원곡선암호시스템에서 Montgomery ladder 방법에 기반한 새로운 스칼라 곱셈 알고리즘*

조 성 민,[1†] 서 석 충,[1] 김 태 현,[1] 박 영 호,[2] 홍 석 희[1‡]
[1]고려대학교 정보경영공학전문대학원, [2]세종사이버대학교

New Efficient Scalar Multiplication Algorithms Based on Montgomery Ladder Method for Elliptic Curve Cryptosystems*

Sung Min Cho,[1†] Seog Chung Seo,[1] Tae Hyun Kim,[1]
Yung Ho Park,[2] Seokhie Hong[1‡]
[1]Graduate School of Information Management and Security, Korea University,
[2]School of Computer Engineering, Sejong Cyber University

요 약

본 논문에서는 Montgomery ladder 방법을 확장한 효율적인 스칼라 곱셈 알고리즘을 제안한다. 제안하는 방법은 효율성을 높이기 위하여 스칼라를 ternary 또는 quaternary로 표현하고 아핀좌표계에서 Montgomery ladder 방법과 같이 x 좌표만을 이용하여 연산 가능하도록 하는 새로운 연산식을 적용한다. 그리고 단순전력분석에 안전하도록 Side-channel atomicity를 적용하였다. 또한 Montgomery trick을 사용하여 연산속도를 높였다. 제안하는 방법은 기존에 효율적으로 알려진 window method, comb method에 비해서 연산속도가 26% 이상 향상된다. 또한 이 방법들보다 저장공간을 적게 사용하는 장점도 가지고 있다.

ABSTRACT

This paper proposes efficient scalar multiplication algorithms based on Montgomery ladder method. The proposed algorithm represents the scalar as ternary or quaternary and applies new composite formulas utilizing only x coordinate on affine coordinate system in order to improve performance. Furthermore, side-channel atomicity mechanism is applied on the proposed composite formulas to prevent simple power analysis. The proposed methods saves at least 26% of running time with the reduced number of storage compared with existing algorithms such as window-based methods and comb-based methods.

Keywords: Elliptic curve cryptosystem, Montgomery ladder, Simple power analysis

I. 서 론

타원곡선암호시스템은 1985년에 Koblitz와 Miller에 의해서 처음으로 제안되었다[1,2]. 타원곡선암호시스템은 기존의 RSA와 ElGamal 공개키 암호시스템과 비교하여 훨씬 짧은 길이의 키로도 비슷한 보안성을 제공한다는 장점을 가지고 있다. 키의 길이가 짧다는 장점 때문에 저전력을 사용하는 스마트카드나 PDA등에 많이 사용된다. 타원곡선암호시스템의 안전성과 효율성에 가장 큰 영향을 주는 연산은 타원곡선 스칼라 곱셈이다. 타원곡선 스칼라 곱셈의 안전성은 타원곡선 이산대수문제 (Elliptic curve discrete logarithm problem, ECDLP)에 기반을 두고 있으며 이에 대한 수학적 안전성이 증명되면서 효율성을 높이기 위한 많은 연구들이 진행되었다. 타원곡선 스칼라 곱셈의 효율

접수일(2009년 2월 19일), 게재확정일(2009년 6월 18일)
* 본 연구에 참여한 연구자(의 일부)는 '3 단계 BK21사업'의 지원비를 받았음.
† 주저자, csm82@cist.korea.ac.kr
‡ 교신저자, hsh@cist.korea.ac.kr

성을 높이기 위한 연구는 알고리즘 단계와 타원곡선 연산 단계에서 주로 진행되었다. 알고리즘 단계에서는 부호화 표현과 사전테이블 등을 이용한 효율적인 덧셈체인을 제안하여 효율성을 높일 수 있었고[3,4], 타원곡선 연산 단계에서는 사영 좌표계를 이용하거나 복합연산(composite operation)을 이용하여 효율성을 높일 수 있었다[5-7]. 또한 아핀좌표계에서는 동시에 계산되는 여러 개의 유한체 역원을 한 번에 계산 가능한 Montgomery trick 방법을 사용하여 효율성을 높일 수 있었다[8].

1999년 Kocher에 의해서 부채널 공격에 대한 개념이 소개된 이후에 수학적으로 안전한 알고리즘도 알고리즘이 실행될 때 발생하는 시간, 전력 소모량, 전자파와 같은 물리적 정보를 이용한 공격에 안전하지 않다는 사실이 증명되었다[9]. 이 중에서 전력분석은 타원곡선 스칼라 곱셈 알고리즘에 대한 강력한 공격방법으로 스마트카드와 같은 모바일 장치에서 암호알고리즘이 동작할 때 소모되는 전력정보를 이용한 공격 방법이다. 전력분석은 단순전력분석과 차분전력분석으로 나눌 수 있다. 단순전력분석은 한 번의 알고리즘 수행에 의한 전력소모량을 가지고 분석하는 방법으로 비밀키에 의존하는 공개키 암호 연산에 적용된다. 차분전력분석은 고정된 비밀키에 대한 암호 연산의 서로 다른 데이터에 대한 다수의 수행과정에서 비밀키의 특정 비트 값에 의존하는 중간 계산 값과 해당 전력소모량 사이의 상관관계를 통계적으로 분석하는 공격 방법이다. 타원곡선 스칼라 곱셈에 사용되는 연산들은 서로 다른 연산량과 연산순서를 가지고, 키 비트에 의존해서 실행되는 연산에 차이가 생기기 때문에 단순전력분석에 의한 분석이 많이 진행되었다[10]. 스칼라 곱셈 알고리즘에 대한 단순전력분석 연구가 진행되면서 이를 방어하기 위한 많은 대응방법들도 제안되었다[11-14]. 기존의 단순전력분석에 대응하는 방법들은 크게 두 가지로 분류할 수 있다. 그 중 하나는 키 비트에 관계없이 항상 고정된 패턴으로 스칼라 곱셈을 수행하는 방법으로 Montgomery ladder 방법과 Double-and-add always방법 등이 있고[15,16], 두 번째 방법은 타원곡선 연산을 구분할 수 없도록 하는 방법으로 indistinguish operation과 Side-channel atomicity방법 등이 있다[17,18]. 그러나 이러한 방법들에는 문제점이 있다. 첫 번째 방법은 추가적인 연산이 많이 사용된다는 점이다. 즉, 연산량이 많아지므로 연산속도가 느려지는 단점이 있다. 두 번째 방법은 키 비트의 해밍웨이트에 의존해서 발생하는 연산의 횟수에 차이가 생긴다. 따라서 해밍웨이트의 정보가 노출될 수 있다.

Izu-Takagi들은 키 값을 이진법으로 표현하고 두 개의 레지스터를 이용하는 기본적인 Montgomery ladder 방법을 제안하고 효율성을 높이기 위하여 extended binary method를 이용한 변형된 Montgomery ladder 알고리즘을 제안하였다. 변형된 Montgomery ladder방법은 한번의 Elliptic curve point doubling과 세 번의 Elliptic curve point addition를 이용하며 모두 병렬적으로 연산이 가능하다[16]. 그러나 이 방법은 네 개의 레지스터를 사용한다는 단점을 가지고 있다.

본 논문에서는 Montgomery ladder 방법에서의 키 값을 ternary 또는 quaternary로 표현했을 때, 두 개의 레지스터만을 사용하는 새로운 Montgomery ladder방법을 제안한다. 그리고 알고리즘의 효율적인 연산을 위해서 복합연산에 대해서 x 좌표만을 이용하는 새로운 연산식들을 제안하며, 단순전력분석에 취약한 문제점을 해결하기 위해 Side-channel atomicity를 적용한다. 또한 두 개의 레지스터에서 동시에 연산되는 유한체 역원연산들에 Montgomery trick을 적용하여 연산의 효율성을 높인다. 제안하는 단순전력분석에 안전한 알고리즘 중 quaternary로 확장한 방법은 기존에 제안된 단순전력분석에 안전한 방법들에 비해서 약 26%이상의 연산속도의 향상을 볼 수 있다. 또한 연산속도를 빠르게 하기 위해 사전테이블을 사용하는 comb method와 window method의 경우는 사전테이블을 저장하기 때문에 저장공간이 많이 필요하다는 단점이 있다. 즉, 제안하는 방법은 적은 저장공간을 사용하면서 comb method와 window method들 보다 26%이상 연산속도가 향상된다.

본 논문의 구성은 다음과 같다. 2절은 타원곡선 암호시스템과 부채널공격에 대해 설명하고, 3절에서는 제안하는 스칼라 곱셈 알고리즘과 알고리즘에 필요한 새로운 연산식들을 소개한다. 그리고 4절에서는 제안하는 방법이 단순전력분석에 안전하게 할 수 있는 Side-channel atomicity의 적용과 Montgomery trick을 적용한 방법에 대한 설명을 하고, 5절에서 기존에 제안된 방법들과 효율성을 비교한다.

II. 타원곡선 암호시스템과 부채널 공격

2.1 타원곡선 암호시스템

본 절에서는 타원곡선에 대한 기본적인 사항에 대

해서 알아본다. 본 논문에서는 $GF(2^m)$ 상에서의 non-supersingular 타원곡선에 대해서만 고려한다. 유한체 $GF(2^m)$상에서 정의되는 weierstrass 식을 만족하는 해의 집합은 항등원 역할을 하는 무한원점 (Point at infinity)과 함께 아벨군을 형성한다[15].

$$E/GF(2^m): y^2+xy=x^3+ax^2+b, a,b\in GF(2^m) \quad (1)$$

아벨군의 규칙에 의거하여 타원곡선 상에 존재하는 두 점 P_1과 P_2의 합의 결과 P_3은 여전히 타원곡선 위에 존재한다. 타원곡선상의 임의의 두 점을 각각 $P_1=(x_1,y_1), P_2=(x_2,y_2)$라 할 때 $(P_1\neq -P_2)$, P_1과 P_2의 합의 결과인 $P_3=(x_3,y_3)$의 아핀 좌표는 아래와 같이 계산될 수 있다[19]

$$x_3=\lambda^2+\lambda+x_1+x_2+a, y_3=(x_1+x_3)\lambda+x_3+y_1,$$
$$\lambda=\frac{(y_1+y_2)}{(x_1+x_2)} \text{ if } P_1\neq P_2, \text{ and } \lambda=\frac{y_1}{x_1}+x_1 \text{ if } P_1=P_2 \quad (2)$$

서로 다른 두 점을 더하는 연산을 Elliptic curve point addition (*ECADD*)라 하고 같은 점을 더하는 연산을 Elliptic curve point doubling (*ECDBL*)이라고 한다. *ECADD*와 *ECDBL*은 모두 두 번의 유한체 곱셈, 한 번의 유한체 제곱과 한 번의 유한체 역원 연산이 필요하다. 본 논문에서는 유한체 곱셈, 제곱, 역원연산을 각각 M, S, I로 표현한다. 즉, *ECADD*와 *ECDBL*의 연산량은 1I+2M+1S로 나타낸다. 두 연산을 이용해서 타원곡선 스칼라 곱셈을 정의할 수 있다. d를 양의 정수라 하고, P를 타원곡선 위의 한 점이라고 할 때, P를 d번 더하는 연산을 타원곡선 스칼라 곱셈이라 한다. 타원곡선 스칼라 곱셈은 타원곡선 암호 시스템의 안전성과 효율성에 가장 큰 영향을 주는 연산이다. 스칼라 곱셈을 구현하는 가장 일반적이고 기본적인 방법으로 Left-to-right binary method가 있다. 이 방법은 상수 d를 $d=d_{n-1}2^{n-1}+\cdots+d_0(d_{n-1}=1)$와 같이 이진표현으로 나타내고 dP를 계산한다. 이 방법은 [알고리즘 1]로 나타낼 수 있다.

알고리즘 1. Left-to-right binary method

1. INPUT: 상수
   $d=d_{n-1}2^{n-1}+\cdots+d_0(d_{n-1}=1)$, 점 P
2. OUTPUT: dP

3. $Q[0]=P$
4. for $i\leftarrow n-2$ downto 0
5. $\quad Q[0]=ECDBL(Q[0])$
6. $\quad$ if $d[i]=1$ then
7. $\quad\quad Q[0]=ECADD(Q[0],P)$
8. Return ($Q[0]$)

2.2 부채널 공격

부채널 공격은 Kocher에 의해 처음 제안된 공격 방법이다[9]. 이 중 전력분석공격은 공격자가 암호장비를 분해하지 않고 사용되는 전력만을 이용한 공격법으로 간단하고 강력한 공격방법이다. 전력분석은 크게 단순전력분석 (Simple Power Analysis, SPA)과 차분전력분석 (Diffential Power Analysis, DPA)으로 나눌 수 있다[20]. 이중 단순전력분석은 한 개의 입력에 대하여 암호 연산의 차이에 의하여 발생하는 전력소모량의 차이를 이용하여 키 정보를 알아내는 방법이다. 타원곡선 스칼라 곱셈 알고리즘에 사용되는 연산들은 서로 다른 전력소모량을 가지고, 키의 정보에 따라서 실행되는 연산이 다르다는 특징이 있기 때문에 단순전력분석 공격이 가능하다. 예를 들어 [알고리즘 1]의 경우는 키 값인 d의 비트에 의존해서 비트값이 0일 때는 *ECADD*연산이, 비트값이 1일 때는 *ECADD*와 *ECDBL*연산이 각각 사용된다. 비록 *ECADD*와 *ECDBL*은 같은 유한체 역원, 곱셈, 제곱 연산량을 가지지만 유한체 덧셈 연산에서 차이가 생기고, 덧셈 연산에 의한 연산의 순서에도 차이가 생기는 것을 알 수 있다. 따라서 d의 비트에 따른 전력파형의 차이를 이용해서 쉽게 키의 비트값을 알아낼 수 있다.

타원곡선 스칼라 곱셈 알고리즘에서 단순전력분석에 대한 연구가 활발히 진행되면서 공격에 안전한 많은 알고리즘들이 제안되었다[16,18,21-23]. 이 중에서도 1999년 Coron에 의해서 더미연산을 추가하는 Dummy addition method가 처음 제안되었고, 2002년에 Izu-Takagi들에 의해서 Montgomery ladder를 이용한 단순전력분석에 안전한 방법이 제안되었다[16,20]. Dummy addtion method와 Montgomery ladder는 모두 d의 비트에 상관없이 매 루프 마다 *ECADD*와 *ECDBL*연산이 동일한 패턴으로 연산되기 때문에 단순전력분석에 안전하다. 그러나 이 방법들은 단순전력분석에 대한 대응방법이 적용되지 않은 Left-to-right binary method에 비해서 연산속도가 매우 느려지는 단점을 가지고 있다. 하지만, Montgomery ladder 방법은 Lopez, Dahab 등이 제안한 $GF(2^m)$상에서의 x좌표만을 이용한

*ECADD*와 *ECDBL*방법의 적용으로 연산속도를 향상시킬 수 있다[24]. x좌표만을 이용한 연산은 Montgomery에 의해서 처음으로 제안된 방법으로 Lopez, Dahab등은 $GF(2^m)$ 상에서의 연산을 다음과 같이 정리했다. 타원곡선상의 임의의 두 점을 각각 $P_1=(x_1,y_1), P_2=(x_2,y_2)$이고 $P_2-P_1=(x,y)$라 할 때, $(P_1\neq -P_2)$, P_1과 P_2의 합의 결과인 $P_3=(x_3,y_3)$의 아핀 좌표는 아래와 같이 계산될 수 있다.

$$x_3=\begin{cases} x+\left(\frac{x_1}{x_1+x_2}\right)+\left(\frac{x_1}{x_1+x_2}\right)^2 & \text{if } P_1\neq P_2, \\ x_1^2+\frac{b}{x_1^2} & \text{if } P_1=P_2. \end{cases} \quad (3)$$

이러한 x좌표만을 이용해 서로 다른 두 점을 더하는 연산을 x-Elliptic curve point addition ($ECADD^x$)이라 하고 같은 점을 더하는 연산을 x-Elliptic curve point doubling($ECDBL^x$)라고 한다. 그리고 $ECADD^x$와 $ECDBL^x$의 연산량은 두 개 모두 1I+1M+1S이다. Montgomery ladder는 $Q[0]$와 $Q[1]$의 차이가 P을 유지하는 특징을 가지고 있기 때문에 이러한 효율적인 연산방법이 가능하다. 이러한 연산식을 적용한 Montgomery ladder는 [알고리즘 2]와 같이 나타낼 수 있다.

x 좌표만을 사용하는 연산의 적용으로 Montgomery ladder는 Dummy addition method 보다 효율적이긴 하나 x 좌표만을 사용하는 Montgomery ladder 방법은 단순전력분석에 대한 대응방법이 적용되지 않은 Left-to-right binary method에 비해서 연산량이 80개의 유한체 역원연산이 늘어나고 160개의 유한체 곱셈이 줄어드는 것을 알 수 있다. 유한체 역원연산과 유한체 곱셈연산의 속도비율이 일반적으로 약 I/M=8이므로 Montgomery ladder는 단순전력분석에 대한 대응방법이 적용되지 않은 Left-to-right binary method 보다 연산이 느리다는 것을 알 수 있다.

이 후 2003년에 Chevalier-Mames, M. Ciet, and M. Joy등은 추가적인 연산을 최소화 하고, 단순전력분석에 안전하도록 하는 Side-channel atomicity방법을 제안했다[18]. 이 방법은 스칼라 곱셈을 계산할 때 나타나는 연산들을 키 비트에 관계없이 항상 같은 연산블록들로 분리한다. 예를 들어 *ECADD*와 *ECDBL*의 연산에 유한체 덧셈 연산만을 추가하여 같은 형태의 연산을 하지만 *ECADD* 또는 *ECDBL*의 값을 출력하는 하나의 atomic block I을 만드는 것이다. atomic block을 적용한 Left-to-right binary method는 유한체 덧셈 연산만을 추가해서 단순전력분석에 안전하도록 한다. 유한체 덧셈 연산은 유한체 역원연산과 곱셈연산에 비해서 연산속도가 매우 빠르므로 추가적인 연산 없이 Left-to-right binary method 방법을 단순전력분석에 안전하게 할 수 있다.

알고리즘 2. Montgomery ladder

1. INPUT:
   상수 $d=d_{n-1}2^{n-1}+\cdots+d_0(d_{n-1}=1)$, 점 P
2. OUTPUT: dP
3. $Q[0]=P,\ Q[1]=2P$
4. for $i\leftarrow n-2$ downto 0
5. $\quad Q[2]=ECDBL^x(Q[0])$
6. $\quad Q[1]=ECADD^x(Q[0],Q[1])$
7. $\quad Q[0]=Q[2-d[i]]$!
8. $\quad Q[1]=Q[1+d[i]]$
9. Return ($Q[0]$)

III. 제안하는 확장된 Montgomery ladder 알고리즘

본 절에서는 키 값 d를 ternary 또는 quaternary로 표현했을 때의 확장된 Montgomery ladder 알고리즘에 대해 설명한다. 그리고 확장된 Montgomery ladder 알고리즘의 효율적인 연산을 위해 x 좌표만을 이용한 복합연산에 대해 정의한다.

3.1 확장된 Ternary Montgomery ladder 알고리즘

양의 정수 d를 $d=d_{n-1}3^{n-1}+\cdots+d_0(d_{n-1}=1,2)$와 같이 ternary로 표현하자. Montgomery ladder의 특성을 유지하도록 i번째 비트까지의 합을 $Q[0]_i=\left(\sum_{k=1}^{i}d_{n-k}3^{i-k}\right)P$와 같이 정의하고, $Q[1]_i=Q[0]_i+P$이라고 정의하면, d_{n-i-1}의 값에 의존하여 $Q[0]_{i+1}$, $Q[1]_{i+1}$와 $Q[0]_i$, $Q[1]_i$의 관계를 다음과 같이 나타낼 수 있고, $Q[1]_{i+1}=Q[0]_{i+1}+P$의 관계를 유지함을 알 수 있다.

$d_{n-i-1}=0$ 이면,

$Q[0]_{i+1}=3Q[0]_i,\ Q[1]_{i+1}=2Q[0]_i+Q[1]_i,$

알고리즘 3. 확장된 Ternary Montgomery ladder

1. INPUT: 상수
   $d=d_{n-1}3^{n-1}+\cdots+d_0(d_{n-1}=1,2)$, 점 P
2. OUTPUT: dP
3. $Q[0]=d_{n-1}P$d, $Q[1]=(d_{n-1}+1)P$
4. for $i\leftarrow n-2$ downto 0
5. if $d_i=0$ then
6. $Q[2]=ECTPL(Q[0])$,
7. $Q[1]=ECDA(Q[0],Q[1])$,
8. if $d_i=1$ then
9. $Q[2]=ECDA(Q[0],Q[1])$,
10. $Q[1]=ECDA(Q[1],Q[0])$,
11. if $d_i=2$ then
12. $Q[2]=ECDA(Q[1],Q[0])$,
13. $Q[1]=ECTPL(Q[1])$,
14. $Q[0]=Q[2]$.
15. Return ($Q[0]$)

$d_{n-i-1}=1$ 이면,
$Q[0]_{i+1}=2Q[0]_i+Q[1]_i$ $Q[1]_{i+1}=2Q[1]_i+Q[0]_i$,
$d_{n-i-1}=2$ 이면,
$Q[0]_{i+1}=2Q[1]_i+Q[0]_i$, $Q[1]_{i+1}=3Q[1]_i$.

이와 같이 d을 ternary로 표현했을 때, $Q[0]_{i+1}$과 $Q[1]_{i+1}$은 d_{n-i-1}의 값에 따라서 이전에 계산된$Q[0]_i$와 $Q[1]_i$를 이용하여 $3P$ 또는 $2P+Q$의 형태로 계산된다. $3P$와 $2P+Q$의 연산은 Eisenträger et al.등이 제안한 복합연산이 적용가능하다[7]. 서로 다른 두 점을 $2P+Q$하는 연산을 Elliptic curve point double-and-add (*ECDA*)라 하고, 같은 점을 $3P$하는 연산을 Elliptic curve point tripling (*ECTPL*)이라고 한다. 그러면, d를 ternary로 표현했을 때의 확장된 Montgomery ladder는 [알고리즘 3]으로 나타낼 수 있다.

확장된 Ternary Montgomery ladder는 매 루프마다 $Q[0]$와 $Q[1]$의 차이가 P를 유지하는 성질을 가지고 있다. 이 특성을 이용하여 x 좌표만을 이용하는 Montgomery ladder를 구성할 수 있다. 본 논문에서는 x 좌표만을 이용한 $ECDA^x$와 $ECTPL^x$의 연산식을 3.3절에서 제안한다.

3.2 확장된 Quaternary Montgomery ladder 알고리즘

양의 정수 d를 $d=d_{n-1}4^{n-1}+\cdots+d_0(d_{n-1}=1,2,3)$와 같이 quaternary로 표현하자. 3.1절에서와 마찬가지로 i번째 비트까지의 합을 $Q[0]_i=\left(\sum_{k=1}^{i}d_{n-k}4^{i-k}\right)P$와 같이 정의하고, $Q[1]_i=Q[0]_i+P$이라고 정의하면, d_{n-i-1}의 값에 의존하여 $Q[0]_{i+1}$, $Q[1]_{i+1}$와 $Q[0]_i$, $Q[1]_i$의 관계를 다음과 같이 나타낼 수 있고, $Q[1]_{i+1}=Q[0]_{i+1}+P$의 관계를 유지함을 알 수 있다.

$d_{n-i-1}=0$ 이면,
$Q[0]_{i+1}=4Q[0]_i$, $Q[1]_{i+1}=3Q[0]_i+Q[1]_i$,
$d_{n-i-1}=1$ 이면,
$Q[0]_{i+1}=3Q[0]_i+Q[1]_i$ $Q[1]_{i+1}=2Q[0]_i+2Q[1]_i$,
$d_{n-i-1}=2$ 이면,
$Q[0]_{i+1}=2Q[0]_i+2Q[1]_i$, $Q[1]_{i+1}=3Q[1]_i+Q[0]_i$,
$d_{n-i-1}=3$ 이면,
$Q[0]_{i+1}=3Q[1]_i+Q[0]_i$, $Q[1]_{i+1}=4Q[1]_i$.

이와 같이 d을 quaternary로 표현했을 때, $Q[0]_{i+1}$과 $Q[1]_{i+1}$은 d_{n-i-1}의 값에 따라서 이전에 계산된 $Q[0]_i$와 $Q[1]_i$를 이용하여 $2P+2Q$, $3P+Q$ 또는 $4P$의 복합연산의 형태로 계산된다. 서로 다른 두 점의 $2P_1+2P_2$형태의 연산을 Elliptic curve point double-add-double (*ECDAD*), $3P_1+P_2$형태의 연산을 Elliptic curve point triple-and-add (*ECTA*)라 하고 같은 점을 $4P_1$하는 연산을 Elliptic curve point quadrupling (*ECQPL*)이라고 한다. 그러면, d을 quaternary로 표현했을 때의 확장된 Montgomery ladder는 [알고리즘 4]와 같이 나타

알고리즘 4. 확장된 Quaternary Montgomery ladder

1. INPUT: 상수
   $d=d_{n-1}4^{n-1}+\cdots+d_0(d_{n-1}=1,2,3)$, 점 P
2. OUTPUT: dP
3. $Q[0]=d_{n-1}P$, $Q[1]=(d_{n-1}+1)P$
4. for $i\leftarrow n-2$ downto 0
5. if $d_i=0$ then
6. $Q[2]=ECQPL(Q[0])$,
7. $Q[1]=ECTA(Q[0],Q[1])$,
8. if $d_i=1$ then
9. $Q[2]=ECTA(Q[0],Q[1])$,
10. $Q[1]=ECDAD(Q[0],Q[1])$,
11. if $d_i=2$ then
12. $Q[2]=ECDAD(Q[0],Q[1])$,
13. $Q[1]=ECTA(Q[1],Q[0])$,
14. if $d_i=3$ then
15. $Q[2]=ECTA(Q[1],Q[0])$,
16. $Q[1]=ECQPL(Q[1])$,
17. $Q[0]=Q[2]$.
18. Return ($Q[0]$)

낼 수 있다.

확장된 Quaternary Montgomery ladder 또한 매 루프마다 $Q[0]$와 $Q[1]$의 차이가 P를 유지하는 성질을 가지고 있다. 이 특성을 이용하여 x 좌표만을 이용하는 확장된 Quaternary Montgomery ladder를 구성할 수 있다. x 좌표만을 이용한 $ECTA^x$, $ECDAD^x$과 $ECQPL^x$의 연산식 또한 3.3절에서 제안한다.

3.3 x 좌표만을 이용하는 복합연산

본 절에서는 [알고리즘 3]과 [알고리즘 4]에 적용되는 x 좌표만을 이용한 복합연산에 대한 연산식 들을 제안한다. 제안하는 연산식은 x 좌표만을 사용하는 $ECADD^x$와 $ECDBL^x$에 기반을 두고 식을 변형한 형태이다. 타원곡선상의 임의의 두 점을 각각 $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$라 하고, 두 점의 차를 $P=P_2-P_1=(x,y)$라 하자($P_1\neq -P_2$). 그리고 식의 전개를 위해 중간 계산과정의 점 A의 x 좌표를 x_A의 형태로 나타내자.

먼저 [알고리즘 3]에 필요한 $ECTPL^x$에 대하여 $P_3=3P_1=(x_3,y_3)$라 하자. 그러면 $P_3=3P_1=2P_1+P_1$의 형태로 연산가능하다. $ECDBL^x$의 식 (3)을 이용하여 $2P_1$의 x좌표 x_{2P_1}은 다음과 같이 나타낼 수 있다.

$$x_{2P_1}=x_1^2+\frac{b}{x_1^2}=\frac{x_1^4+b}{x_1^2} \tag{4}$$

그리고 $2P_1$과 P_1의 차이는 $2P_1-P_1=P_1$이므로 $ECADD^x$의 식 (3)을 이용해서 P_3의 x좌표를 $2P_1$과 P_1의 x좌표로 다음과 같이 나타낼 수 있다.

$$x_3=x_1+\left(\frac{x_1}{x_1+x_{2P_1}}\right)+\left(\frac{x_1}{x_1+x_{2P_1}}\right)^2 \tag{5}$$

위 식에 x_{2P_1}의 값인 식 (4)을 대입하면 다음과 같은 x_1만을 이용한 표현이 가능하다.

$$x_3=x_1+\left(\frac{x_1^3}{x_1^4+x_1^3+b}\right)+\left(\frac{x_1^3}{x_1^4+x_1^3+b}\right)^2 \tag{6}$$

이 때의 연산량은 1I+2M+3S가 사용되는데, 이는 $ECDBL^x$과 $ECADD^x$을 각각 계산했을 때의 연산량인 2I+2M+2S보다 한 번의 유한체 역원이 감소하고 한 번의 유한체 제곱이 증가하는 것을 알 수 있다. 앞으로 제안하는 x 좌표만을 사용하는 복합연산의 연산식들은 위와 같은 방법을 이용하여 전개한다.

이번에는 [알고리즘 3]에 필요한 $ECDA^x$에 대하여 $P_4=2P_1+P_2=(x_4,y_4)$라 하면, $P_4=2P_1+P_2=(P_1+P_2)+P_1$의 의 형태로 연산가능하다. $ECADD^x$의 식 (3)을 이용하여 P_1+P_2의 x좌표 $x_{P_1+P_2}$를 다음과 같이 나타낼 수 있다.

$$\begin{aligned}x_{P_1+P_2}&=x+\left(\frac{x_1}{x_1+x_2}\right)+\left(\frac{x_1}{x_1+x_2}\right)^2\\&=\frac{x(x_1+x_2)^2+x_1x_2}{(x_1+x_2)^2}\end{aligned} \tag{7}$$

그리고 P_1+P_2와 P_1의 차이는 P_2이므로 $ECADD^x$의 식 (3)을 이용하여 P_4의 x좌표를 P_1+P_2, P_1과 P_2의 x좌표 x_1, x_2, $x_{P_1+P_2}$로 다음과 같이 나타낼 수 있다.

$$x_4=x_2+\left(\frac{x_1}{x_1+x_{P_1+P_2}}\right)+\left(\frac{x_1}{x_1+x_{P_1+P_2}}\right)^2 \tag{8}$$

위 식에 $x_{P_1+P_2}$의 값인 식 (7)을 대입하면 다음과 같이 표현된다.

$$\begin{aligned}x_4=x_2&+\left(\frac{x_1(x_1+x_2)^2}{(x+x_1)(x_1+x_2)^2+x_1x_2}\right)\\&+\left(\frac{x_1(x_1+x_2)^2}{(x+x_1)(x_1+x_2)^2+x_1x_2}\right)^2\end{aligned} \tag{9}$$

이때의 연산량은 1I+4M+2S로 $ECADD^x$를 두 번 연산할 때의 연산량인 2I+2M+2S에 비해서 한 번의 유한체 역원이 감소하고 두 번의 유한체 곱셈이 증가한다.

마찬가지로 [알고리즘 4]에 적용되는 적용 가능한 $ECQPL^x$, $ECDAD^x$과 $ECTA^x$의 연산식에 대해 설명한다. 먼저 $ECQPL^x$에 대하여 $P_5=4P_1=(x_5,y_5)$라 하자. 그러면 $P_5=4P_1=2P_1+2P_1$의 형태로 연산가능하다. $ECDBL^x$의 식 (3)을 사용해서 P_5의 x좌표를 $2P_1$의 x좌표 x_{2P_1}로 다음과 같이 나타낼 수 있다.

$$x_5=x_{2P_1}^2+\frac{b}{x_{2P_1}^2} \tag{10}$$

위 식에 x_{2P_1}의 값인 식 (4)을 대입하면 다음과 같은 x_1만을 이용한 표현이 가능하다.

$$x_5 = x_1^4 + \frac{b^2(x_1^4+b)^2 + bx_1^8}{x_1^4(x_1^4+b)^2} \tag{11}$$

이때의 연산량은 1I+4M+5S가 사용되는데, 이는 $ECDBL^x$을 두 번 계산했을 때의 연산량인 2I+2M+2S에 비해서 한 번의 유한체 역원이 감소하고 두 번의 유한체 곱셈과 세 번의 유한체 제곱이 증가한다.

다음으로 $ECDAD^x$의 연산식을 보자. $P_6 = 2P_1 + 2P_2 = (x_6, y_6)$라 하고, 두 값의 차를 $2P_2 - 2P_1 = (x_0, y_0)$라 하자. 그러면 $ECDBL^x$의 식 (3)을 이용해서 $2P_2$의 x좌표는 다음과 같이 나타낼 수 있다.

$$x_{2P_2} = x_2^2 + \frac{b}{x_2^2} = \frac{x_2^4+b}{x_2^2} \tag{12}$$

$ECADD^x$의 식 (3)을 사용해서 P_6의 x좌표를 $2P_1$과 $2P_2$의 x좌표 x_{2P_1}, x_{2P_2}로 다음과 같이 나타낼 수 있다.

$$x_6 = x_0 + \left(\frac{x_{2P_2}}{x_{2P_1}+x_{2P_2}}\right) + \left(\frac{x_{2P_2}}{x_{2P_1}+x_{2P_2}}\right)^2 \tag{13}$$

위 식에 x_{2P_1}과 x_{2P_2}의 값인 식 (4)와 식 (12)를 대입하면 다음과 같이 표현된다.

$$x_6 = x_0 + \left(\frac{x_1^2(x_2^4+b)}{(x_1^2x_2^2+b)(x_1^2+x_2^2)}\right) + \left(\frac{x_1^2(x_2^4+b)}{(x_1^2x_2^2+b)(x_1^2+x_2^2)}\right)^2 \tag{14}$$

이 때의 연산량은 1I+4M+4S가 사용되는데, 이는 $ECDBL^x$을 두 번과 $ECADD^x$을 한 번 계산했을 때의 연산량인 3I+3M+3S에 비해서 두 번의 유한체 역원이 감소하고 한 번의 유한체 곱셈과 한 번의 유한체 제곱이 증가한다.

마지막으로 $ECTA^x$에서 $P_7 = 3P_1 + P_2 = (x_7, y_7)$라 하자. 그러면 $P_7 = 3P_1 + P_2 = (P_1 + P_2) + 2P_1$의 형태로 연산가능하다. $P_1 + P_2$와 $2P_1$의 차는 $P_1 - P_2$이므로, $ECADD^x$를 사용해서 $P_1 + P_2$, $2P_1$와 $P_1 - P_2$의 x좌표로 다음과 같이 나타낼 수 있다.

$$x_7 = x + \left(\frac{x_{2P_1}}{x_{P_1+P_2}+x_{2P_1}}\right) + \left(\frac{x_{2P_1}}{x_{P_1+P_2}+x_{2P_1}}\right)^2 \tag{15}$$

위 식에 x_{2P_1}과 $x_{P_1+P_2}$의 값인 식 (4)와 식 (7)를 대입하면 다음과 같이 표현된다.

$$x_7 = x + \lambda + \lambda^2 \quad where$$
$$\lambda = \frac{(x_1^4+b)(x_1+x_2)^2}{x_1^2\{x(x_1+x_2)^2 + x_1x_2\} + (x_1^4+b)(x_1+x_2)^2} \tag{16}$$

이때의 연산량은 1I+5M+4S가 사용되는데, 이는 $ECDBL^x$을 한 번과 $ECADD^x$을 두 번 계산했을 때의 연산량인 3I+3M+3S에 비해서 두 번의 유한체 역원이 감소하고 두 번의 유한체 곱셈과 한 번의 유한체 제곱이 증가한다.

제안하는 x 좌표만을 이용하는 복합연산은 한 번의 유한체 역원연산을 사용한다는 장점을 가지고 있다. 위 연산들을 〔알고리즘 3〕에 적용하면 루프마다 사용되는 연산량이 2I+(20/3)M+(14/3)S이다. 기존의 Montgomery ladder 방법에서는 매 루프마다 $ECADD^x$와 $ECDBL^x$가 각각 한 번씩 연산되므로 루프마다 사용되는 연산량은 2I+2M+2S이다. 〔알고리즘 3〕은 d의 표현을 ternary의 형식으로 표현했기 때문에 루프의 수는 0.63배만큼이 줄어든다. n을 이진 표현을 했을 때의 비트 수 라고 하면, 〔알고리즘 3〕은 Montgomery ladder에 비해서 0.74n번의 유한체 역원 연산이 줄어들고 2.2n번의 유한체 곱셈과 0.9n번의 유한체 제곱 연산이 늘어난다. 즉, I>3M일 경우에는 〔알고리즘 3〕이 연산속도가 빠르다는 것을 의미한다. 마찬가지로 〔알고리즘 4〕에 대한 적용에서는 루프마다 사용되는 연산량이 2I+9M+5.4S이다. 〔알고리즘 4〕는 d의 표현을 quaternary의 형식으로 표현하므로 루프의 수는 0.5배만큼이 줄어든다. 따라서 〔알고리즘 4〕는 Montgomery ladder에 비해서 n번의 유한체 역원 연산이 줄어들고, 2.5n번의 유한체 곱셈과 2.3n번의 유한체 제곱 연산이 늘어난다. 즉, I>2.5M일 경우에는 〔알고리즘 4〕가 연산속도가 빠르다는 것을 의미한다. 자세한 연산의 비교는 5절에서 설명한다.

IV. 단순전력분석에 안전한 알고리즘

3절에서 제안된 확장된 Montgomery ladder 방법들은 x 좌표만을 이용하여 연산이 가능하지만

Montgomery ladder와는 다르게 단순전력분석에 의한 공격이 가능하다는 문제점이 있다. 본 절에서는 제안하는 알고리즘이 단순전력분석에 안전하도록 Side-channel atomicity의 적용과 연산속도를 더욱 빠르게 하는 방법을 제안한다.

4.1 단순전력분석에 안전한 설계

Montgomery ladder 방법은 매 루프마다 키 값에 상관없이 같은 연산이 일어나는 장점을 가진다. 반면에 제안하는 방법의 경우는 키 값에 의존해서 다른 연산이 발생하므로 단순전력분석에 의한 공격이 가능하다는 단점을 가지고 있다. 예를 들어 확장된 Ternary Montgomery ladder 알고리즘의 경우 $d_i=0$ 또는 1일 경우에는 $ECTPL^x$과 $ECDA^x$연산이, $d_i=1$일 때는 두 번의 $ECDA^x$연산이 일어나는 차이를 보인다. $ECTPL^x$과 $ECDA^x$연산은 연산량과 연산 패턴이 다르기 때문에 단순전력분석에 쉽게 구별된다. 마찬가지로 확장된 Quaternary Montgomery ladder 알고리즘의 경우는 $d_i=0$ 또는 3과 $d_i=1$ 또는 2일 때의 연산이 차이가 나기 때문에 단순전력분석에 쉽게 키 값이 노출된다. 따라서 단순전력분석에 안전하도록 Side-channel atomicity를 적용한다.

먼저 확장된 Ternary Montgomery ladder 알고리즘의 경우에 대한 Side-channel atomicity를 적용하는 방법에 대하여 설명한다. 이 알고리즘은 키의 값에 의존해서 $ECTPL^x$, $ECDA^x$ 또는 $ECDA^x$, $ECDA^x$가 연산되는 문제점이 있다. 한 번의 루프에서 연산되는 $ECTPL^x$, $ECDA^x$ 또는 $ECDA^x$, $ECDA^x$를 두 개의 atomic block인 γ_1과 γ_2로 나눈다. 여기서 γ_1을 공통으로 연산되는 $ECDA^x$, γ_2를 서로 다른 연산인 $ECTPL^x$와 $ECDA^x$로 한다. γ_1은 같은 $ECDA^x$연산이므로 연산순서나 더미 연산이 불필요하다. 반면에 γ_2는 $ECTPL^x$ 또는 $ECDA^x$를 연산해 주어야 하므로 atomic block을 만들어 줄 필요가 있다. atomic block을 만들어 주기 위하여 먼저 $ECTPL^x$의 식 (6)을 $ECDA^x$와 비슷한 연산을 가지는 식 (17)의 형태로 나타내어 준다. 그 이유는 더미연산을 줄이기 위한 작업이다.

$$x_3 = x_1 + \frac{x_1(x_1^2)}{x_1(x_1(x_1^2+x_1))+b} + \left(\frac{x_1(x_1^2)}{x_1(x_1(x_1^2+x_1))+b}\right)^2 \quad (17)$$

식 (9)와 식 (17)에 의해서 $ECTPL^x$ 또는 $ECDA^x$를 출력해주는 atomic block 〔표 2〕를 만들 수 있다.

atomic block 〔표 2〕을 $atomicECDA^x$ (atomic

〔표 1〕 $GF(2^m)$상에서 Side-channel atomic block이 적용된 double-and-add 알고리즘

Input : $P_1=T_1, P_2=T_2, P_1-P_2=T_3$ Output : $2P_1+P_2$ 또는 $3P_1$	
$ECDA^x$: $P_1 \leftarrow 2P_1+P_2$	$ECTPL^x$: $P_1 \leftarrow 3P_1$
$T_4 \leftarrow T_1+T_2$ $(=x_1+x_2)$ $T_4 \leftarrow T_4^2$ $(=(x_1+x_2)^2)$ $T_5 \leftarrow T_1 \cdot T_4$ $(=B)$ $T_3 \leftarrow T_3+T_1$ $(=x+x_1)$ $T_3 \leftarrow T_3 \cdot T_4$ $(=(x+x_1)(x_1+x_2)^2)$ $T_4 \leftarrow T_1 \cdot T_2$ $(=x_1x_2)$ $T_2 \leftarrow b$ $(dummy)$ $T_3 \leftarrow T_3+T_4$ $(=A)$ $T_5 \leftarrow T_5/T_3$ $(=B/A)$ $T_1 \leftarrow T_1+T_5$ $(=x_1+B/A)$ $T_5 \leftarrow T_5^2$ $(=(B/A)^2)$ $T_1 \leftarrow T_1+T_5$ $(=x_1+B/A+(B/A)^2)$ *where* $A=(x+x_1)(x_1+x_2)^2+x_1x_2$ $B=x_1(x_1+x_2)^2$	$T_4 \leftarrow T_1+T_2$ $(dummy)$ $T_3 \leftarrow T_1^2$ $(=x_1^2)$ $T_5 \leftarrow T_1 \cdot T_3$ $(=B)$ $T_3 \leftarrow T_3+T_1$ $(=x_1^2+x_1)$ $T_3 \leftarrow T_3 \cdot T_1$ $(=x_1(x_1^2+x_1))$ $T_4 \leftarrow T_1 \cdot T_3$ $(=x_1(x_1(x_1^2+x_1)))$ $T_2 \leftarrow b$ $(=b)$ $T_3 \leftarrow T_3+T_2$ $(=A)$ $T_5 \leftarrow T_5/T_3$ $(=B/A)$ $T_1 \leftarrow T_1+T_5$ $(=x_1+B/A)$ $T_5 \leftarrow T_5^2$ $(=(B/A)^2)$ $T_1 \leftarrow T_1+T_5$ $(=x_1+B/A+(B/A)^2)$ *where* $A=x_1(x_1(x_1^2+x_1))+b$, $B=x_1(x_1^2)$

〔표 2〕 Side-channel atomic double-and -add($AtomicECDA^x$)

Input : $Q[a]=T_1$, $Q[b]=T_2$, $P=T_3$ Output : $2Q[a]+Q[b]$ 또는 $3Q[a]$
$k=a \oplus b$ $T_4 \leftarrow T_1+T_2$ $T_{3+k} \leftarrow T_{1+3k}^2$ $T_5 \leftarrow T_1 \cdot T_{3+k}$ $T_3 \leftarrow T_3+T_1$ $T_3 \leftarrow T_3 \cdot T_{1+3k}$ $T_4 \leftarrow T_1 \cdot T_{3-k}$ $T_2 \leftarrow b$ $T_3 \leftarrow T_3+T_{2+2k}$ $T_5 \leftarrow T_5/T_3$ $T_1 \leftarrow T_1+T_5$ $T_5 \leftarrow T_5^2$ $T_1 \leftarrow T_1+T_5$ return T_1

알고리즘 5. 단순전력분석에 안전한 확장된 Ternary Montgomery ladder

1. INPUT: 상수
   $d = d_{n-1}3^{n-1} + \cdots + d_0 (d_{n-1} = 1 \text{ 또는 } 2)$, 점 P
2. OUTPUT: dP

3. $Q[0] = d_{n-1}P$, $Q[1] = (d_{n-1}+1)P$
4. for $i \leftarrow n-2$ downto 0
5. $\quad Q[2] = ECDA^x(Q[d_i^H], Q[\overline{d_i^H}])$,
6. $\quad Q[d_i^H \vee d_i^L] = atomicECDA^x(Q[d_i^H \vee d_i^L], Q[d_i^H])$,
7. $\quad Q[\overline{d_i^H \vee d_i^L}] = Q[2]$.
8. Return ($Q[0]$)

block을 적용한 x-Elliptic curve point double and add)라고 정의 한다. 이 때 $atomicECDA^x$의 연산량은 1I+4M+2S이다. 〔알고리즘 3〕에서 루프마다 $ECTPL^x$ 또는 $ECDA^x$을 연산할 때의 평균 연산량이 1I+8/3M+8/3S이므로 4/3M정도의 추가 연산이 필요하다.

〔표 2〕에 의해서 만들어진 $atomicECDA^x$를 적용한 확장된 Ternary Montgomery ladder 방법은 〔알고리즘 5〕와 같이 나타낼 수 있다. (d_i값인 0,1,2를 비트정보로 각각 00, 01, 10으로 나타내고, 상위 비트를 d_i^H, 하위 비트를 d_i^L으로 나타낸다.)

마찬가지로 확장된 Quaternary Montgomery ladder 방법의 경우에도 키의 값에 의존해서 $ECQPL^x$, $ECTA^x$ 또는 $ECDAD^x$, $ECTA^x$가 연산되므로 연산의 차이에 의한 단순전력분석에 의한 공격이 가능하다. 한 번의 루프에서 연산되는 $ECQPL^x$, $ECTA^x$ 또는 $ECDAD^x$, $ECTA^x$를 두 개의 atomic block인 γ_1과 γ_2로 나눈다. 여기서 γ_1을 공통으로 연산되는 $ECTA^x$, γ_2를 서로 다른 연산인 $ECQPL^x$와 $ECDAD^x$로 한다. γ_1은 연산이 같으므로 추가 연산이 필요하지 않다. 반면에 γ_2는 서로 다른 연산을 출력하기 위한 atomic block의 구성이 필요하다. 자세한 atomic block은 Appendix A에서 다룬다. 이와 같이 만들어진 atomic block을 $atomicECDAD^x$ (atomic block을 적용한 x-Elliptic curve point double-add-double)라고 정의 한다. 이 때 $atomicECDAD^x$의 연산량은 1I+4M+5S이다. $atomicECDAD^x$를 적용한 확장된 Quaternary Montgomery ladder 방법은 〔알고리즘 6〕과 같이 나타낼 수 있다. (d_i값인 0,1,2,3을 비트정보로 각각 00, 01, 10, 11으로 나타내고, 상위 비트를 d_i^H, 하위 비트를 d_i^L으로 나타낸다.)

〔알고리즘 4〕에서 루프마다 $ECQPL^x$ 또는 $ECDAD^x$을 연산할 때의 평균 연산량이 1I+4M+4.5S이므로 0.5S정도의 추가 연산이 필요하다. 유한체 제곱연산은$GF(2^m)$상에서 유한체 곱셈과 유한체 역원에 비해 연산속도가 빠르기 때문에 무시할 수 있다. 즉, Side-channel atomicity을 적용한 확장된 Quaternary Montgomery ladder 방법은 추가적인 연산 없이 단순전력분석에 안전하도록 할 수 있다.

4.2 Montgomery trick의 적용

제안하는 〔알고리즘 5〕와 〔알고리즘 6〕은 Izu-Takagi가 제안한 것처럼 병렬로 동작 가능하다〔16〕. 즉, 병렬 동작 가능한 환경에서는 Izu-Takagi가 제안한 방법과 같이 두 개의 저장공간을 이용해서 연산 속도를 높일 수 있다. 그러나 병렬동작이 불가능한 환경에서는 매 루프마다 유한체 역원이 두 번 연산되어야 한다. 본 절에서는 두 번의 유한체 역원연산의 효율성을 높이기 위한 방법으로 Montgomery trick을 이용한다〔8〕.

Montgomery trick은 n개의 원소에 대한 유한체 역원을 한 번의 유한체 역원 연산과 $3(n-1)$번의 유한체 곱셈연산으로 구하는 방법이다. 예를 들어 a와 b를 $GF(2^m)$의 두 원소라 하자. 그러면 두 원소의 유한체 역원인 a^{-1}와 b^{-1}는 $a^{-1} = (a \cdot b)^{-1} \cdot b$와 $b^{-1} = (a \cdot b)^{-1} \cdot a$와 같이 계산할 수 있다. 이 때 연산량은 한 번의 유한체 역원과 세 번의 유한체 곱셈이 필요하다. 즉, 한 번의 유한체 역원을 세 번의 유한체 곱셈으로 나타낸다. 〔알고리즘 3〕과 〔알고리즘 4〕에 Montgomery trick을 사용할 경우에도 단순전력분석에 의한 키 정보 노출이 생긴다. 이 경우에도 4.1절과 같이 Side-channel atomicity방법을 적용하여 단순전력분석

알고리즘 6. 단순전력분석에 안전한 확장된 Quaternary Montgomery ladder

1. INPUT: 상수
   $d = d_{n-1}4^{n-1} + \cdots + d_0 (d_{n-1} = 1,2,3)$, 점 P
2. OUTPUT: dP

3. $Q[0] = d_{n-1}P$, $Q[1] = (d_{n-1}+1)P$
4. for $i \leftarrow n-2$ downto 0
5. $\quad Q[2] = ECTA^x(Q[d_i^H], Q[\overline{d_i^H}])$,
6. $\quad Q[d_i^L] = atomicECDAD^x(Q[d_i^H], Q[d_i^L])$,
7. $\quad Q[\overline{d_i^L}] = Q[2]$.
8. Return ($Q[0]$)

알고리즘 7. 단순전력분석에 안전한 확장된 Ternary Montgomery ladder(Montgomery trick 적용)

1. INPUT: 상수
$d=d_{n-1}3^{n-1}+\cdots+d_0(d_{n-1}=1,2)$, 점 P
2. OUTPUT: dP
3. $Q[0]=d_{n-1}P$, $Q[1]=(d_{n-1}+1)P$
4. for $i\leftarrow n-2$ downto 0
5. $(Q[0],Q[1])=atomicECDDA^x(Q[d_i^H],Q[\overline{d_i^H}])$
6. Return $(Q[0])$

알고리즘 8. 단순전력분석에 안전한 확장된 Quaternary Montgomery ladder(Montgomery trick 적용)

1. INPUT: 상수
$d=d_{n-1}4^{n-1}+\cdots+d_0(d_{n-1}=1,2,3)$, 점 P
2. OUTPUT: dP
3. $Q[0]=d_{n-1}P$, $Q[1]=(d_{n-1}+1)P$
4. for $i\leftarrow n-2$ downto 0
5. $(Q[0],Q[1])=atomicECTADAD^x(Q[d_i^H],Q[\overline{d_i^H}])$
6. Return $(Q[0])$

에 안전하게 할 수 있다. Montgomery trick을 적용한 경우에 대한 atomic block의 구성은 Appendix B, C에서 설명한다. [알고리즘 3]에서의 $ECDA^x$, $ECTPL^x$ 또는 $ECDA^x$, $ECDA^x$를 연산하는 atomic block을 $atomicECDDA^x$ (atomic block을 적용한 x-Elliptic curve point Double double-and-add)라 정의하고, [알고리즘 4]에서의 $ECTA^x$, $ECQPL^x$ 또는 $ECTA^x$, $ECDAD^x$를 연산하는 atomic block을 $atomicECTADAD^x$ (atomic block을 적용한 x-Elliptic curve point triple-and-add and double-add-double)라 정의하자. $atomicECDDA^x$와 $atomicECTADAD^x$는 두 개의 입력값 $Q[0]$와 $Q[1]$에 대해서 비트의 정보에 따라서 다음루프의 입력값인 새로운 $Q[0]$와 $Q[1]$값을 출력하는 연산 알고리즘이다. 이 때 $atomicECDDA^x$와 $atomicECTADAD^x$의 연산량은 각각 1I+10M+5S와 1I+11M+9S이다. $atomic ECDDA^x$와 $atomicECTADAD^x$을 적용한 확장된 Montgomery ladder 방법은 각각 [알고리즘 7]과 [알고리즘 8]으로 나타낼 수 있다.

V. 비교 및 분석

본 절에서는 4절에서 제안한 atomic block을 적용한확장된 Momtgomery ladder 방법과 기존에 제안된 방법들과 비교하고 효율성에 대해서 설명한다. 비교는 환경을 고려해서 병렬 동작이 불가능한 환경에서 Montgomery trick과 atomic block을 적용한 확장된 Montgomery ladder 방법과 기존의 방법들을 비교한다. $GF(2^m)$상에서의 유한체 제곱의 연산은 유한체 역원과 유한체 곱셈에 비해서 연산속도가 빠르기 때문에 무시할 수 있다. 따라서 비교는 유한체 역원과 유한체 곱셈에 대해서만 한다.

일반적으로 I/M=8이라는 가정했을 때의 정확한 연산량을 비교해 보자. 비교는 크게 단순전력분석에 안전한 알고리즘과 단순전력분석에 안전하지 않은 알고리즘에 대해서 기존의 알고리즘과 비교한다. 먼저 단순전력분석에 안전하지 않은 기존의 알고리즘과의 비교는 [표 3]과 같이 나타낼 수 있다.

확장된 Ternary Montgomery ladder 방법은 binary와 w-NAF방법에 대해서는 각각 약 23%, 3-13%, 연산속도의 향상을 보이는 것을 알 수 있다. 그리고 확장된 Quaternary Montgomery ladder 방법은 binary, w-NAF, termary/binary, DB-chain, Multi base2 방법에 비해서 각각 약 37%, 21-29%, 16%, 11%, 4% 연산속도가 향상되는 것을 알 수 있다. 반면에 확장된 Quaternary Montgomery ladder 방법은 multi-base 1에 비해서는 2% 연산이 느려지는 것을 알 수 있다. 그러나 Muti-base를 이용하는 스칼라 곱셈 알고리즘의 경우는 최적화된 비트를 찾는데 어려움이 있고 단순전력분석에 의한 공격이 가능하다는 단점이 있다. 반면에 확장된Quaternary Montgomery ladder 방법은 비트의 표현이 쉽고, 단순전력분석에 안전하다는 특징을 가지고 있다. Multi-base를 이용하는 스칼라 곱

[표 3] binary, NAF, ternary/binary, DB-chain과 Multi-base 방법의 평균 연산 횟수

알고리즘	I/M=8		
	I	M	≈M
binary	240	480	2400
NAF[25]	213	426	2130
3-NAF[25]	200	400	2000
4-NAF[25]	192	384	1920
ternary/binary[5]	129	787	1819
DB-chain[15]	114	789	1701
Multi-base1[26]	97	693	1469
Multi-base2[26]	113	677	1581
알고리즘 7	101	1006	1814
알고리즘 8	80	878	1518

셈 알고리즘을 단순전력분석에 안전하도록 side-channel atomicity를 적용해 주면 연산량이 크게 늘어난다. 즉, 제안하는 알고리즘에 비해서 연산속도가 느려지는 것을 알 수 있다. 제안하는 방법은 단순전력분석에 안전하면서 기존의 단순전력분석에 안전하지 않은 알고리즘들보다 빠르거나 비슷한 연산속도를 가지고 있다.

기존에 제안된 단순전력분석에 안전한 알고리즘은 더미연산을 추가하는 Coron's dummy addition method[21], 덧셈체인을 이용한 Montgomery ladder[16], Side-channel atomicity를 적용한 side-channel atomic double and add[18]들과 사전테이블을 이용하는 Comb method[27,28], window method[22,23]등의 Table look-up 방법이 있다. 일반적으로 키는 160 비트 정수라 할 때, 기존의 단순전력분석에 안전한 알고리즘들과 [알고리즘 7], [알고리즘 8]과의 break even point는 각각 [표 4]과 [표 5]로 나타낼 수 있다.

[표 4] 단순전력분석에 안전한 알고리즘들과 [알고리즘 7]과의 break even point (d는 160비트 정수, (4) : window 크기)

알고리즘	I/M	알고리즘	I/M
coron's dummy addition method	1.8	HPB's comb(4)	5.4
Izu-Takagi	3.3	signed Odd-only comb(4)	5.6
side-channel atomic double and add	4	OT's window(4)	5.8
-	-	Moller's window(4)	6

[표 5] 단순전력분석에 안전한 알고리즘들과 [알고리즘 8]과의 break even point (d는 160비트 정수, (4) : window 크기)

알고리즘	I/M	알고리즘	I/M
coron's dummy addition method	1	HPB's comb(4)	3.5
Izu-Takagi	2.4	signed Odd-only comb(4)	3.7
side-channel atomic double and add	2.5	OT's window(4)	3.8
-	-	Moller's window(4)	4

[알고리즘 7]의 경우는 [표 3]에서 나타난 결과처럼 I>6M일 경우에는 기존의 제안된 방법들 보다 빠른 연산속도를 가지는 것을 알 수 있다. [알고리즘 8]의 경우는 [표 4]에서 보는 것과 같이 I>4M일 경우에 기존의 방법들 보다 연산속도가 향상되는 것을 알 수 있다.

일반적으로 I/M=8이라는 가정했을 때의 정확한 연산량 비교는 [표 6]와 같이 나타낼 수 있다.

제안하는 [알고리즘 7]과 [알고리즘 8]은 기존에 제안되었던 단순전력분석에 안전한 알고리즘들 보다 연산이 빨라지는 것을 알 수 있다. [알고리즘 7]은 Window method와 Comb method에 비해서는 약 9%정도 연산속도가 빨라지고, Coron's method, Izu-Takagi method와 Side-channel atomic double and add에 비해서 각각 41%, 35%, 23%의 연산속도 향상을 보였다. [알고리즘8]은 Window method와 Comb method에 비해서는 약 26%의 연산속도 향상을 보였고, Coron's method, Izu-Takagi method와 Side-channel atomic double andadd에 비해서 각각 52%, 46%, 36%의 연산속도 향상을 보였다. 게다가 Comb method, window method방법의 경우는 사전테이블을 저장해야하는 단점이 있다. 비교하는 방법의 window 크기는 가장 효율적이라고 할 수 있는 4로 했을 때의 연산량이다. 즉 사전테이블에 사용되는 저장공간이 8개 이상이 된다는 것을 알 수 있다.

반면에 제안하는 방법의 경우는 입력 P와 $2P$의 값만을 사전저장해서 사용하므로 Comb method, window method방법에 비해서 저장공간을 적게 사

[표 6] 기존의 단순전력분석에 안전한 알고리즘들의 평균 연산 횟수

알고리즘	I/M=8		
	I	M	≈M
coron's dummy addition method	318	636	3180
Izu-Takagi	318	318	2862
side-channel atomic double and add	240	480	2400
HPB's comb(4)	211	422	2110
signed Odd-only comb(4)	207	414	2070
OT's window(4)	205	410	2050
Moller's window(4)	203	406	2030
알고리즘 7	101	1006	1814
알고리즘 8	80	878	1518

용한다. 즉, 저장공간을 적게 사용하면서 기존의 방법보다 빠른 연산속도를 가진다.

VI. 결 론

본 논문에서는 기존의 Montgomery ladder 방법을 확장한 새로운 스칼라 곱셈 알고리즘을 제안하였다. 제안하는 방법은 Montgomery ladder 방법과 마찬가지로 단순전력분석에 안전하고, x 좌표만을 이용하는 연산이 가능하도록 하였다. 제안하는 방법들은 기존의 단순전력분석에 안전한 알고리즘들에 비해서 연산속도가 향상되는 것을 알 수 있다. 특히 확장된 Quaternary Montgomery ladder 방법은 기존의 단순전력분석에 안전한 알고리즘들보다 26% 이상의 연산속도 향상을 볼 수 있다. 그리고 적은 저장공간을 사용하면서 사전테이블을 이용하는 방법들보다 연산의 효율성이 향상되는 것을 알 수 있다. 또한 제안하는 방법은 병렬동작이 가능하므로 보다 효율적인 연산도 가능하다.

참 고 문 헌

[1] N. Koblitz, "Elliptic Curve Cryptosystems," Mathematics of Computation, vol. 48, no. 177, pp. 203-309, Jan. 1987.

[2] V. Miller, "Uses of Elliptic Curves in Cryptography," Advances in cryptography -CRYPTO 85', LNCS 218, pp. 417-426, 1986.

[3] I. Blake, G. Seroussi, and N. Smart, Elliptic Curves in Cryptography, Cambridge University Press, July 1999.

[4] F. Morain and J. Olivos, "Speeding up the computation of an elliptic curve using addition-subtraction chains," Informatique théorique et Applications, pp. 531 - 544, Sep. 1990.

[5] M. Ciet, K. Lauter, M. Joye, and P.L. Montgomery, "Trading inversions for multiplications in elliptic curve cryptography," Designs, Codes and Cryptography, vol. 39, no. 2, pp. 189-206, May 2006.

[6] H. Cohen, A. Miyaji, and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Asiacrypt98, LNCS 1514, pp. 51 - 65, 1998.

[7] K. Eisenträger, K. Lauter, and P.L. Montgomery, "Fast elliptic curve arithmetic and improved Weil pairing evaluation. In M. Joye, editor," Topics in Cryptology-CT-RSA 2003, LNCS 2612, pp. 343-354, 2003.

[8] H. Cohen, A course in computational algebraic number theory : Graduate Texts in Mathematics, Springer-Verlag, Sep. 1993.

[9] P. Kocher, "Timing Attacks on Implementations of Diffie- Hellman, RSA, DSS, and Others Systems," CRYPTO'96, LNCS 1109, pp. 104-113, 1996.

[10] 한동국, 김성경, 김태현, 김호원, 임종인, "단순전력분석에 안전한 Signed Left-to-Right 리코딩 방법," 정보보호학회논문지, 17(1), pp. 127-132, 2007년 2월.

[11] 김성경, 한동국, 김호원, 정교일, 임종인, "SPA에 안전한 Unsigned Left-to-Right 리코딩 방법," 정보보호학회논문지, 17(1), pp. 21-32, 2007년 2월.

[12] 김태현, 장상운, 김웅희, 박영호, "부채널 공격에 안전한 타원곡선 스칼라 곱셈 알고리즘," 정보보호학회논문지, 14(6), pp. 125-134, 2004년 12월.

[13] 임채훈, "부가채널 공격에 안전한 효율적인 타원곡선 상수배 알고리즘," 정보보호학회논문지, 12(4), pp. 99-114, 2002년 8월.

[14] 한동국, 장남수, 장상운, 임종인, "랜덤한 덧셈-뺄셈 체인에 대한 부채널 공격," 정보보호학회논문지, 14(5), pp. 121-133, 2007년 10월.

[15] V. Dimitrov, L. Imbert, and P.K. Mishra, "Efficient and Secure Elliptic Curve Point Multiplication using Double Base Chain. In: Roy, B. (ed.)," ASIACRYPT 2005, LNCS 3788, pp. 59-79, 2005.

[16] T. Izu and T. Takagi, "A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks," PKC 2002, LNCS 2274, pp. 280-296, 2002.

[17] E. Brier and M. Joye, "Weierstrass

Elliptic Curves and Side-Channel Attacks," Public Key Cryptography (PKC2002), LNCS 2274, pp. 335-345, 2002.

[18] B.C. Mames, M. Ciet, and M. Joye, "Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity," IEEE Transactions on Computers, vol. 53, no. 6, pp. 760-768, June 2004.

[19] D. Hankerson, A.J. Menezes, and S.A. Vanstone, Guide to elliptic curve cryptography, Springer-Verlag, Jan. 2004.

[20] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis. In: Wiener," M.J.(ed.) CRYPTO 1999, LNCS 1666, pp. 388-397, 1999.

[21] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," CHES'99, LNCS 1717, pp. 292-302, 1999.

[22] B. M¨oller, "Securing Elliptic Curve Point Multiplication against Side-Channel Attacks," ICS 2001, LNCS 2200, pp. 324 - 334, 2001.

[23] K. Okeya and T. Takagi, "The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplication Secure against Side Channel Attacks," CT-RSA 2003, LNCS 2612, pp. 328-342, 2003.

[24] J. Lopez and R. Dahab, "Fast Multiplication on Elliptic Curves over $GF(2^m)$ without Precomputation," Cryptographic Hadware and Embedded Systems-CHES'99, LNCS 1717, pp. 316-327, 1999.

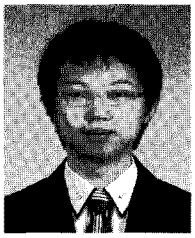
[25] J.A. Solinas, "Efficient Arithmetic on Koblitz Curves," Designs, Codes and Cryptography, vol. 19, no. 2-3, pp. 195-249, Mar. 2000.

[26] P.K. Mishra and V. Dimitrov, "Efficient Quintuple Formulas for Elliptic Curves and Efficient Scalar Multiplication Using Multibase Number Representation," ISC 2007, LNCS 4779, pp. 390-406, 2007.

[27] M. Feng, B.B. Zhu, M. Xu, and S. Li, "Efficient Comb Elliptic Curve Multiplication Methods Resistant to Power Analysis," http://eprint.iacr.org/2005/222.ps.gz, 2005.

[28] M. Hedabou, P. Pinel, and L. B'eb'eteau, "A Comb Method to Render ECC Resistant against Side Chiannel Attacks," http://eprint.iacr.org/2004/342.pdf, 2004.

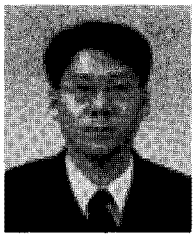
〈著者紹介〉



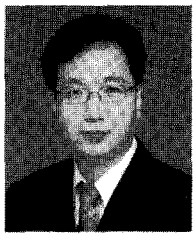
조 성 민 (Sung Min Cho) 학생회원
2008년 2월 : 광운대학교 수학과 학사
2008년 3월~현재 : 고려대학교 정보경영공학전문대학원 석사 과정
〈관심분야〉 암호 구현, 공개키 암호 알고리즘

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〉 암호 구현, 센서 네트워크

김 태 현 (Tae Hyun KIM) 정회원
2002년 2월 : 서울 시립대학교 수학과 이학사
2004년 8월 : 고려대학교 정보보호 대학원 공학석사
2009년 2월 : 고려대학교 정보경영공학전문대학원 공학박사
〈관심분야〉 부채널 공격, 공개키 암호 알고리즘, 암호칩 설계 기술



박 영 호 (Young Ho Park) 정회원
1990년: 고려대학교 수학과 이학사
1993년: 고려대학교 수학과 이학석사
1997년: 고려대학교 수학과 이학박사
2006년 2월~현재 세종 사이버 대학교 부교수
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜, 부채널 공격



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~현재 : 고려대학교 정보경영전문대학원 부교수
〈관심분야〉 대칭키 암호 알고리즘, 공개키 암호 알고리즘, 포렌식

부 록

본 절에서는 알고리즘에 적용되는 Montgomery trick을 적용한 연산들에 Side-channel atomicity를 적용한 atomic block에 대해서 설명한다.

A *AtomicECDAD*^{x}

〔알고리즘 6〕에 적용되는 *ECDAD*^{x} 또는 *ECQPL*^{x}을 계산하는 *AtomicECDAD*^{x}를 설명한다. 타원곡선상의 임의의 두 점을 각각 $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$라 하고, $2(P_2-P_1)=(x_0,y_0)$라 하자($P_1\neq -P_2$). 그러면, 식 (11)과 식 (14)를 이용하여 다음의 〔표 7〕과 〔표 8〕을 만들 수 있다.

B *AtomicECDDA*^{x}

〔알고리즘 7〕에 적용되는 Montgomery trick을 적용한 *ECDA*^{x}, *ECTPL*^{x} 또는 *ECDA*^{x}, *ECDA*^{x}을 계산하는 *AtomicECDDA*^{x}를 설명한다. 타원곡선상의 임의의 두 점을 각각 $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$라 하고, 두 점의 차를 $P=(P_2-P_1)=(x,y)$라 하자($P_1\neq -P_2$). 그러면, 식 (6)과 식 (9)를 이용하여 다음의 〔표 9〕과 〔표 10〕을 만들 수 있다.

C *AtomicECTADAD*^{x}

〔알고리즘 8〕에 적용되는 Montgomery trick을 적용한 *ECTA*^{x}, *ECQPL*^{x} 또는 *ECTA*^{x}, *ECDAD*^{x}을 계산하는 *AtomicECTADAD*^{x}를 설명한다.

타원곡선상의 임의의 두 점을 각각 $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$라 하고, 두 점의 차를 $P=(P_2-P_1)=(x,y)$라 하자 ($P_1\neq -P_2$). 그리고 $2P=2(P_2-P_1)=(x_0,y_0)$라 하자. 그러면, 식 (11), 식 (14)와 식 (16)을 이용하여 다음의 〔표 11〕과 〔표 12〕를 만들 수 있다.

〔표 7〕 Side-channel atomic double-add-double for elliptic curves over $GF(2^m)$

Input : $P_1=T_1$, $P_2=T_2$, $2(P_1-P_2)=T_3$ Output : $2P_1+2P_2$ 또는 $4P_1$	
ECMDDA^x : $P_1\leftarrow 2P_1+2P_2$	*ECMQPL*^{x} : $P_1\leftarrow 4P_1$
$T_1\leftarrow T_1^2$ $(=x_1^2)$	$T_1\leftarrow T_1^2$ $(=x_1^2)$
$T_2\leftarrow T_2^2$ $(=x_2^2)$	$T_1\leftarrow T_1^2$ $(=x_1^4)$
$T_6\leftarrow b$ $(=b)$	$T_4\leftarrow b$ $(=b)$
$T_4\leftarrow T_1+T_2$ $(=(x_1^2+x_2^2)$	$T_2\leftarrow T_1+T_4$ $(=x_1^4+b)$
$T_5\leftarrow T_2^2$ $(=x_2^4)$	$T_5\leftarrow T_2^2$ $(=(x_1^4+b)^2)$
$T_2\leftarrow T_1\cdot T_2$ $(=x_1^2x_2^2)$	$T_2\leftarrow T_4\cdot T_5$ $(=b(x_1^4+b)^2)$
$T_6\leftarrow T_6^2$ $(dummy)$	$T_6\leftarrow T_1^2$ $(=x_1^8)$
$T_6\leftarrow b$ $(=b)$	$T_4\leftarrow b$ $(=b)$
$T_2\leftarrow T_2+T_6$ $(=x_1^2x_2^2+b)$	$T_2\leftarrow T_2+T_6$ $(=b(x_1^4+b)^2+x_1^8)$
$T_4\leftarrow T_4\cdot T_2$ $(=A)$	$T_4\leftarrow T_4\cdot T_2$ $(=B)$
$T_5\leftarrow T_5+T_6$ $(=x_2^4+b)$	$T_2\leftarrow T_5+T_6$ $(dummy)$
$T_5\leftarrow T_1\cdot T_5$ $(=B)$	$T_5\leftarrow T_1\cdot T_5$ $(=A)$
$T_5\leftarrow T_5/T_4$ $(=B/A)$	$T_5\leftarrow T_4/T_5$ $(=B/A)$
$T_2\leftarrow T_3+T_5$ $(=x_0+B/A)$	$T_2\leftarrow T_3+T_5$ $(dummy)$
$T_6\leftarrow T_5^2$ $(=(B/A)^2)$	$T_6\leftarrow T_5^2$ $(dummy)$
$T_1\leftarrow T_2+T_6$ $(=x_0+B/A+(B/A)^2)$	$T_1\leftarrow T_1+T_5$ $(=x_1^4+B/A)$
$A=(x_1^2x_2^2+b)(x_1^2+x_2^2)$,	$A=x_1^4(x_1^4+b)^2$,
$B=x_1^2(x_2^4+b)$	$B=b(b(x_1^4+b)^2+x_1^8)$

〔표 8〕 Side-channel atomic double-add -double (*AtomicECDAD*^{x})

Input : $Q[a]=T_1$, $Q[b]=T_2$, $P=T_3$ Output : $2Q[a]+Q[b]$ 또는 $3Q[a]$
$k=a\oplus b$
$T_1\leftarrow T_1^2$
$T_{1+k}\leftarrow T_{1+k}^2$
$T_{4+2k}\leftarrow b$
$T_{2+2k}\leftarrow T_1+T_{4-2k}$
$T_5\leftarrow T_2^2$
$T_2\leftarrow T_{4-3k}\cdot T_{5-3k}$
$T_6\leftarrow T_{1+5k}^2$
$T_{4+2k}\leftarrow b$
$T_2\leftarrow T_2+T_6$
$T_4\leftarrow T_4\cdot T_2$
$T_{2+3k}\leftarrow T_5+T_6$
$T_5\leftarrow T_1\cdot T_5$
$T_5\leftarrow T_{4+k}/T_{5-k}$
$T_2\leftarrow T_3+T_5$
$T_6\leftarrow T_5^2$
$T_1\leftarrow T_{1+k}+T_{5+k}$
return T_1

〔표 9〕 Side-channel atomic Double double-and-add for elliptic curves over $GF(2^m)$

Input : $P_1 = T_1$, $P_2 = T_2$, $P_2 - P_1 = T_3$
Output : $(3P_1, 2P_1 + P_2)$ 또는 $(2P_1 + P_2, 2P_2 + P_1)$

$(3P_1, 2P_1 + P_2)$	$(2P_1 + P_2, 2P_2 + P_1)$
$T_4 \leftarrow T_1^2 \quad (= x_1^2)$	$T_4 \leftarrow T_1^2 \quad (dummy)$
$T_5 \leftarrow T_4 \cdot T_1 \quad (= x_1^3)$	$T_5 \leftarrow T_2 \cdot T_1 \quad (= x_1 x_2)$
$T_4 \leftarrow T_4^2 \quad (= x_1^4)$	$T_4 \leftarrow T_4^2 \quad (dummy)$
$T_6 \leftarrow T_1 + T_2 \quad (= x_1 + x_2)$	$T_6 \leftarrow T_1 + T_2 \quad (= x_1 + x_2)$
$T_6 \leftarrow T_6^2 \quad (= (x_1 + x_2)^2)$	$T_6 \leftarrow T_6^2 \quad (= (x_1 + x_2)^2)$
$T_7 \leftarrow T_1 + T_3 \quad (dummy)$	$T_7 \leftarrow T_1 + T_3 \quad (= x + x_1)$
$T_3 \leftarrow T_1 + T_3 \quad (= x + x_1)$	$T_3 \leftarrow T_2 + T_3 \quad (= x + x_2)$
$T_3 \leftarrow T_3 \cdot T_6 \quad (= (x + x_1)(x_1 + x_2)^2)$	$T_3 \leftarrow T_3 \cdot T_6 \quad (= (x + x_2)(x_1 + x_2)^2)$
$T_7 \leftarrow T_1 \cdot T_2 \quad (= x_1 x_2)$	$T_7 \leftarrow T_7 \cdot T_6 \quad (= (x + x_1)(x_1 + x_2)^2)$
$T_6 \leftarrow T_1 \cdot T_6 \quad (= x_1 (x_1 + x_2)^2)$	$T_4 \leftarrow T_1 \cdot T_6 \quad (= x_1 (x_1 + x_2)^2)$
$T_3 \leftarrow T_3 + T_7 \quad (= A)$	$T_7 \leftarrow T_7 + T_5 \quad (= A)$
$T_7 \leftarrow T_2 \cdot T_6 \quad (dummy)$	$T_6 \leftarrow T_2 \cdot T_6 \quad (= x_2 (x_1 + x_2)^2)$
$T_4 \leftarrow T_4 + T_5 \quad (= x_1^4 + x_1^3)$	$T_3 \leftarrow T_3 + T_5 \quad (= B)$
$T_7 \leftarrow b \quad (= b)$	$T_5 \leftarrow b \quad (dummy)$
$T_4 \leftarrow T_4 + T_7 \quad (= B)$	$T_5 \leftarrow T_4 + T_6 \quad (dummy)$
$T_7 \leftarrow T_3 \cdot T_4 \quad (= AB)$	$T_5 \leftarrow T_3 \cdot T_7 \quad (= AB)$
$T_7 \leftarrow T_7^{-1} \quad (= (AB)^{-1})$	$T_5 \leftarrow T_5^{-1} \quad (= (AB)^{-1})$
$T_4 \leftarrow T_7 \cdot T_4 \quad (= A^{-1})$	$T_3 \leftarrow T_5 \cdot T_3 \quad (= A^{-1})$
$T_3 \leftarrow T_7 \cdot T_3 \quad (= B^{-1})$	$T_7 \leftarrow T_5 \cdot T_7 \quad (= B^{-1})$
$T_6 \leftarrow T_6 \cdot T_3 \quad (= x_1 (x_1 + x_2)^2 A^{-1})$	$T_4 \leftarrow T_3 \cdot T_4 \quad (= x_1 (x_1 + x_2)^2 A^{-1})$
$T_5 \leftarrow T_5 \cdot T_3 \quad (= x_1^3 B^{-1})$	$T_5 \leftarrow T_7 \cdot T_6 \quad (= x_2 (x_1 + x_2)^2 B^{-1})$
$T_1 \leftarrow T_1 + T_5$	$T_1 \leftarrow T_1 + T_5$
$T_5 \leftarrow T_5^2$	$T_5 \leftarrow T_5^2$
$T_1 \leftarrow T_1 + T_5 \quad (= 3P)$	$T_1 \leftarrow T_1 + T_5 \quad (= 2Q + P)$
$T_2 \leftarrow T_2 + T_6$	$T_2 \leftarrow T_2 + T_4$
$T_6 \leftarrow T_6^2$	$T_6 \leftarrow T_4^2$
$T_2 \leftarrow T_2 + T_6 \quad (= 2P + Q)$	$T_2 \leftarrow T_2 + T_6 \quad (= 2P + Q)$
$A = (x + x_1)(x_1 + x_2)^2 + x_1 x_2,$	$A = (x + x_1)(x_1 + x_2)^2 + x_1 x_2,$
$B = x_1^4 + x_1^3 + b$	$B = (x + x_2)(x_1 + x_2)^2 + x_1 x_2$

〔표 10〕 Side-channel atomic Double double-and-add ($AtomicECDDA^x$)

Input : $Q[a] = T_1$, $Q[b] = T_2$, $P = T_3$, d_i
Output : $3Q[a], 2Q[a] + Q[b]$ 또는 $2Q[a] + Q[b], 2Q[b] + Q[a]$

$k = d_i^L$
$T_4 \leftarrow T_1^2$
$T_5 \leftarrow T_1 \cdot T_{4-2k}$
$T_4 \leftarrow T_4^2$
$T_6 \leftarrow T_1 + T_2$
$T_6 \leftarrow T_6^2$
$T_7 \leftarrow T_1 + T_3$
$T_3 \leftarrow T_3 + T_{1+k}$
$T_3 \leftarrow T_3 \cdot T_6$
$T_7 \leftarrow T_{1+6k} \cdot T_{2+4k}$
$T_{6-2k} \leftarrow T_1 \cdot T_6$
$T_{3+4k} \leftarrow T_7 + T_{3+2k}$
$T_{7-k} \leftarrow T_2 \cdot T_6$
$T_{4-k} \leftarrow T_5 + T_{4-k}$
$T_{7-2k} \leftarrow b$
$T_{4+k} \leftarrow T_4 + T_{7-k}$
$T_{7-2k} \leftarrow T_3 \cdot T_{4+3k}$
$T_{7-2k} \leftarrow T_{7-2k}^{-1}$
$T_{4-k} \leftarrow T_{7-2k} \cdot T_{4-k}$
$T_{3+4k} \leftarrow T_7 \cdot T_{3+2k}$
$T_5 \leftarrow T_{5+2k} \cdot T_{3+3k}$
$T_{6-2k} \leftarrow T_{6-2k} \cdot T_3$
$T_1 \leftarrow T_1 + T_5$
$T_5 \leftarrow T_5^2$
$T_5 \leftarrow T_1 + T_5$
$T_2 \leftarrow T_2 + T_{6-2k}$
$T_6 \leftarrow T_{6-2k}^2$
$T_6 \leftarrow T_2 + T_6$
$T_1 \leftarrow T_{5+(d_i^H \vee d_i^L)}$
$T_2 \leftarrow T_{6-(d_i^H \vee d_i^L)}$
$return(T_1, T_2)$

〔표 11〕 Side-channel atomic triple-and-add and double-add-double for elliptic curves over $GF(2^m)$

Input : $P_1 = T_1,\ P_2 = T_2,\ P_2 - P_1 = T_3,\ 2(P_2 - P_1) = T_4$
Output : $(4P_1,\ 3P_1 + P_2)$ 또는 $(2P_1 + 2_2,\ 3P_1 + P_2)$

$(4P_1,\ 3P_1 + P_2)$	$(2P_1 + 2P_2,\ 3P_1 + P_2)$
$T_5 \leftarrow T_1^2 \quad (= x_1^2)$	$T_5 \leftarrow T_1^2 \quad (= x_1^2)$
$T_6 \leftarrow T_5^2 \quad (= x_1^4)$	$T_6 \leftarrow T_2^2 \quad (= x_2^2)$
$T_7 \leftarrow T_5^2 \quad (= dummy)$	$T_6 \leftarrow T_6^2 \quad (= x_2^4)$
$T_7 \leftarrow b \quad (= b)$	$T_7 \leftarrow b \quad (= b)$
$T_4 \leftarrow T_6 + T_7 \ (= x_1^4 + b)$	$T_6 \leftarrow T_6 + T_7 \ (= x_2^4 + b)$
$T_8 \leftarrow T_1 + T_2 \ (= x_1 + x_2)$	$T_8 \leftarrow T_1 + T_2 \ (= x_1 + x_2)$
$T_2 \leftarrow T_1 \cdot T_2 \ (= x_1 x_2)$	$T_2 \leftarrow T_1 \cdot T_2 \ (= x_1 x_2)$
$T_8 \leftarrow T_8^2 \quad (= (x_1 + x_2)^2)$	$T_8 \leftarrow T_8^2 \quad (= (x_1 + x_2)^2)$
$T_9 \leftarrow T_3 \cdot T_8 \ (= x(x_1 + x_2)^2)$	$T_9 \leftarrow T_3 \cdot T_8 \ (= x(x_1 + x_2)^2)$
$T_9 \leftarrow T_9 + T_2 \ (= x(x_1 + x_2)^2 + x_1 x_2)$	$T_9 \leftarrow T_9 + T_2 \ (= x(x_1 + x_2)^2 + x_1 x_2)$
$T_9 \leftarrow T_9 \cdot T_5 \ (= x_1^2\{x(x_1 + x_2)^2 + x_1 x$	$T_9 \leftarrow T_9 \cdot T_5 \ (= x_1^2\{x(x_1 + x_2)^2 + x_1 x$
$T_2 \leftarrow T_4^2 \quad (= (x_1^4 + b)^2)$	$T_2 \leftarrow T_2^2 \quad (= (x_1 x_2)^2)$
$T_5 \leftarrow T_2 + T_7 \ (= dummy)$	$T_2 \leftarrow T_2 + T_7 \ (= (x_1 x_2)^2 + b)$
$T_5 \leftarrow T_2 \cdot T_6 \ (= A)$	$T_2 \leftarrow T_2 \cdot T_8 \ (= A)$
$T_4 \leftarrow T_4 \cdot T_8 \ (= (x_1^4 + b)(x_1 + x_2)^2)$	$T_5 \leftarrow T_5 \cdot T_6 \ (= x_1^2(x_2^4 + b))$
$T_8 \leftarrow T_7^2 \quad (= b^2)$	$T_1 \leftarrow T_1^2 \quad (= x_1^2)$
$T_7 \leftarrow T_8 + T_7 \ (= b^2 + b)$	$T_1 \leftarrow T_1^2 \quad (= x_1^4)$
$T_2 \leftarrow T_6^2 \quad (= x_1^8)$	$T_6 \leftarrow T_1 + T_7 \ (= x_1^4 + b)$
$T_8 \leftarrow T_8^2 \quad (= b^4)$	$T_7 \leftarrow T_8^2 \quad (= dummy)$
$T_2 \leftarrow T_2 \cdot T_7 \ (= x_1^8(b^2 + b))$	$T_6 \leftarrow T_6 \cdot T_8 \ (= (x_1^4 + b)(x_1 + x_2)^2)$
$T_8 \leftarrow T_2 + T_8 \ (= x_1^8(b^2 + b) + b^4)$	$T_8 \leftarrow T_2 + T_8 \ (= dummy)$
$T_9 \leftarrow T_9 + T_4 \ (= B)$	$T_9 \leftarrow T_9 + T_6 \ (= B)$
$T_2 \leftarrow T_5 \cdot T_9 \ (= AB)$	$T_6 \leftarrow T_2 \cdot T_9 \ (= AB)$
$T_2 \leftarrow T_2^{-1} \quad (= (AB)^{-1})$	$T_6 \leftarrow T_6^{-1} \quad (= (AB)^{-1})$
$T_5 \leftarrow T_2 \cdot T_5 \ (= B^{-1})$	$T_2 \leftarrow T_2 \cdot T_6 \ (= B^{-1})$
$T_9 \leftarrow T_2 \cdot T_9 \ (= A^{-1})$	$T_9 \leftarrow T_6 \cdot T_9 \ (= A^{-1})$
$T_1 \leftarrow T_4 \cdot T_5 \ (= B^{-1}(x_1^4 + b)(x_1 + x_2)$	$T_1 \leftarrow T_6 \cdot T_2 \ (= B^{-1}(x_1^4 + b)(x_1 + x_2)$
$T_3 \leftarrow T_3 + T_1$	$T_3 \leftarrow T_3 + T_1$
$T_1 \leftarrow T_1^2$	$T_1 \leftarrow T_1^2$
$T_3 \leftarrow T_1 + T_3 \ (= 3P + Q)$	$T_3 \leftarrow T_1 + T_3 \ (= 3P + Q)$
$T_5 \leftarrow T_9 \cdot T_8 \ (= A^{-1}x_1^8(b^2 + b) + b^4)$	$T_5 \leftarrow T_9 \cdot T_5 \ (= A^{-1}x_1^2(x_1^4 + b))$
$T_4 \leftarrow T_6 + T_4 \ (= 4P)$	$T_4 \leftarrow T_4 + T_5$
$T_5 \leftarrow T_5^2 \ (= dummy)$	$T_5 \leftarrow T_5^2$
$T_3 \leftarrow T_4 + T_5 \ (= dummy)$	$T_4 \leftarrow T_4 + T_5 \ (= 2P + 2Q)$
$A = x_1^4(x_1^4 + b)^2,$	$A = ((x_1 x_2)^2 + b)(x_1 + x_2)^2,$
$B = x_1^2\{x(x_1 + x_2)^2 + x_1 x_2\}$ $+ (x_1^4 + b)(x_1 + x_2)^2$	$B = x_1^2\{x(x_1 + x_2)^2 + x_1 x_2\}$ $+ (x_1^4 + b)(x_1 + x_2)^2$

〔표 12〕 Side-channel atomic Double double-and-add ($AtomicECDDA^x$)

Input : $Q[a] = T_1,\ Q[b] = T_2,$
$P = T_3, 2P = T_4, d_i$
Output : $4Q[a], 3Q[a] + Q[b]$ 또는
$2Q[a] + 2Q[b], 3Q[a] + Q[b]$

$k \leftarrow d_i^H \oplus d_i^L$
$T_5 \leftarrow T_1^2$
$T_6 \leftarrow T_5^2$
$T_{7-k} \leftarrow T_{5+k}^2$
$T_7 \leftarrow b$
$T_{4+2k} \leftarrow T_6 + T_7$
$T_8 \leftarrow T_1 + T_2$
$T_2 \leftarrow T_1 \cdot T_2$
$T_8 \leftarrow T_8^2$
$T_9 \leftarrow T_3 \cdot T_8$
$T_9 \leftarrow T_9 + T_2$
$T_9 \leftarrow T_9 \cdot T_5$
$T_2 \leftarrow T_{4-2k}^2$
$T_{5-3k} \leftarrow T_2 + T_7$
$T_{5-3k} \leftarrow T_2 \cdot T_{6+2k}$
$T_{4+k} \leftarrow T_{4+k} \cdot T_{8-2k}$
$T_{8-7k} \leftarrow T_{7-6k}^2$
$T_{2-k} \leftarrow T_{6-5k}^2$
$T_{7-k} \leftarrow T_{8-7k} + T_7$
$T_{8-k} \leftarrow T_8^2$
$T_{2+4k} \leftarrow T_{2+4k} \cdot T_{7+k}$
$T_8 \leftarrow T_2 + T_8$
$T_9 \leftarrow T_9 + T_{4+2k}$
$T_{2+4k} \leftarrow T_{5-3k} \cdot T_9$
$T_{2+4k} \leftarrow T_{2+4k}^{-1}$
$T_{5-3k} \leftarrow T_2 \cdot T_{5+k}$
$T_9 \leftarrow T_{2+4k} \cdot T_9$
$T_1 \leftarrow T_{4+2k} \cdot T_{5-3k}$
$T_3 \leftarrow T_3 + T_1$
$T_1 \leftarrow T_1^2$
$T_3 \leftarrow T_1 + T_3$
$T_5 \leftarrow T_9 \cdot T_{8-3k}$
$T_4 \leftarrow T_{6-2k} + T_{4+k}$
$T_5 \leftarrow T_5^2$
$T_{3+k} \leftarrow T_4 + T_5$
$T_1 \leftarrow T_{4-d_i^L}$
$T_2 \leftarrow T_{3+d_i^L}$
$return(T_1, T_2)$