페어링 및 ECC 상수배 연산의 계산 비용에 관하여

정회원 구 남 훈*, 준회원 조 국 화*, 정회원 김 창 훈*, 권 순 학*°

On the Computational Cost of Pairing and ECC Scalar Multiplication

Namhun Koo* *Regular Member*, Gook Hwa Jo* *Associate Member*,
Chang Hoon Kim*, Soonhak Kwon* *Regular Members*

요 약

겹선형 페어링(bilinear pairing)을 기반으로 하는 암호 프로토콜들은 이산 대수 문제를 기반으로 하는 전통적인 타원 곡선 암호시스템을 대신하여 여러 방면에의 응용성을 제공한다. 겹선형 페어링의 빠른 계산을 위하여 최근 활발한 연구가 진행 중이지만, 여전히 ECC 상수배 연산에 비해서 페어링 연산에 사용되는 계산 비용은 상당히 크다고 여겨진다. 그러나 이진 유한체상의 페어링 계산 연구는 최근 많은 발전이 이루어졌다. 본 논문에서는 이진 유한체상에서의 BLS 서명스킴과 ECDSA 서명 스킴의 복잡도를 비교한다. 공정한 비교를 위하여 1024-bit RSA와 같은 레벨의 보안성을 가지는 160-bit ECDSA와 250-bit BLS를 선택하였다. 분석결과 BLS 스킴은 ECDSA에 비해 하드웨어 복잡도 및 계산 지연시간의 측면에서 많은 차이가 나지 않음을 설명해준다.

Key Words : Tate pairing, elliptic curves, BLS signature scheme, ECDSA

ABSTRACT

Cryptographic protocols based on bilinear pairings provide excellent alternatives to conventional elliptic curve cryptosystems based on discrete logarithm problems. Through active research has been done toward fast computation of the bilinear pairings, it is still believed that the computational cost of one pairing computation is heavier than the cost of one ECC scalar multiplication. However, there have been many progresses in pairing computations over binary fields. In this paper, we compare the cost of BLS signature scheme with ECDSA with equvalent level of security parameters. Analysis shows that the cost of the pairing computation is quite comparable to the cost of ECC scalar multiplication for the case of binary fields.

I. 서 론

타원곡선 상에서 Tate 혹은 Weil 페어링과 같은 암호 스킴들은 겹선형 페어링(bilinear pairing)을 기반으로 한다. 이 같은 암호 프로토콜의 예로 Boneh와 Franklin[1]의 ID기반 암호화 스킴, Boneh 등[2]의 BLS 서명 스킴, Joux[3]의 삼자간 Diffie-Hellman 키 동의 프로토콜, Smart[4]의 ID기반 서명 스킴이 있다.

최근 효율적인 Tate 페어링의 계산에 대한 많은 연구가[5-11,18] 진행되어 왔다. 그러나 페어링의 계산 비용은 타원곡선 상수배 연산의 비용에 비해 매우 무겁고 이 같은 계산적인 부담은 페어링 기반 암호시스템의 사용에 있어서 가장 큰 제약조건이 된다. 게다가 표수가 $p > 2$인 유한체(finite field)에 대한 하드웨어 응

※ 이 논문은 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음. (KRF-314-2008-1-C00040)

* 성균관대학교 수학과(komaton@skku.edu, achimheasal@nate.com, kobhh@naver.com, shkwon@skku.edu) (° : 교신저자)

논문번호 : KICS2009-09-391, 접수일자 : 2009년 9월 7일, 최종논문접수일자 : 2010년 12월 17일

용은 상대적으로 매우 많은 게이트 개수를 요구한다.

본 논문은 [10,11,18]의 결과를 이용하여 이진체상에서 Tate 페어링 계산의 비용과, 같은 보안 레벨의 파라미터를 가지는 타원곡선 상수배 연산 비용[27]을 비교한다. 특히 이진 유한체상에서 BLS 서명 스킴[2]과 ECDSA[14]를 하드웨어 및 시간 복잡도 측면에서 비교분석한다. Tate 페어링 계산과 ECC 상수배 연산에 대한 좋은 알고리즘이 많이 제안되었지만, 이들 연산의 비용에 대해서 아직까지 자세한 비교를 시도한 연구가 없었기에 이들 연구를 위한 좋은 출발점이 될 것으로 기대한다.

Ⅱ.Tate 페어링의 계산을 위한 Miller의 알고리즘

E를 유한체 $GF(q)$상에서 정의된 타원곡선이라고 하자. 여기서 q는 소수의 거듭제곱이다. E에서 divisor D는 곡선위의 점 P들의 유한합, $D=\sum n_P(P), n_P\in Z$으로 정의한다. 만약 $\sum n_P=0$이면, divisor D를 차수가 0인 divisor라고 한다. 곡선 E상의 유리 함수 f에 대하여 임의의 점 P에서 의 곱셈위수를 라 할 때, $(f)=\sum n_P(P)$를 principal divisor라고 한다.

두 개의 divisor D, D'에 대하여 $D-D'$이 principal divisor인 경우에 D와 D'은 동치라고 하며 다음과 같은 동형사상은 잘 알려져 있다[16].

$$\mathrm{Div}_0/\mathrm{Div}_{\mathrm{prin}}\rightarrow \mathrm{E} \text{ with } D=\sum n_P(P)\rightarrow\sum n_P P, \quad (1)$$

여기서 우측식의 합 $\sum n_P P$은 타원곡선 E상에서 포인트 덧셈이고 Div_0(혹은 $\mathrm{Div}_{\mathrm{prin}}$)은 차수가 0인 divisor (혹은 principal divisor)에 의해 생성되는 free 아벨군이다. 이제 l이 소수이고 P가 $E[l]=\{Q|lQ=O\}$의 원소라고 하자. 그러면 $l(P)-l(O)$는 principal divisor이며 $(f_P)=l(P)-l(O)$을 만족하는 유리함수 f_P가 존재한다. 임의의 유리함수 g와 divisor $D=\sum n_P(P)$에 대하여, $g(D)=\prod g(P)^{n_P}$로 정의한다. 집합 $E[l]$에서 Tate 페어링 τ_l은 다음과 같이 정의된다.

정의 1. 점 P는 $E[l](GF(q))$의 원소이고 Q는 $E[l](GF(q^k))$의 원소라고 하자. 여기서 k는 $q^k\equiv 1(\mathrm{mod}\, l)$을 만족하는 최소의 자연수이다. Tate 페어링은 다음과 같이 정의되는 겹선형 함수이다;

$$\tau_l : E[l](GF(q))\times E[l](GF(q^k))\rightarrow\{\zeta_l\} \text{ with}$$
$$\tau_l(P,Q)=f_P(D_Q)^{\frac{q^k-1}{l}}$$

여기에서 f_P는 $(f_P)=l(P)-l(O)$을 만족하는 유리함수이고 D_Q는 $(Q)-(O)$와 동치관계인 차수가 0인 divisor이며 D_Q와 (f_P)는 disjoint support를 갖는다. (즉, $D_Q=\sum n_T(T)$라 할 때 points T는 $T\neq P,O$이다.) 또한 $\{\zeta_l\}$는 $GF(q^k)^\times$에 속하는 l번째 원시근의 순환군이다.

$(f_P)=l(P)-l(O)$을 만족하는 유리함수 f_P를 계산하는 효율적인 알고리즘은 Miller[16,19]에 의해 제안되었다. 차수가 0인 divisor D와 D'에 대해서 식(1)의 동형사상은 유리함수 f와 f'에 대해서 $D=(P)-(O)+(f)$와 $D'=(P')-(O)+(f')$을 만족하는 점 P와 P'이 존재함을 의미한다. 이를 이용하여 Miller 공식이라는 다음 식을 얻게 된다.

$$D+D'=(P+P')-(O)+(ff'\frac{l_{P,P'}}{l_{P+P'}}), \quad (2)$$

여기에서 $l_{P,P'}$는 P와 P'을 지나는 직선 방정식이고 $l_{P+P'}$는 P와 $-P$를 지나는 수직 방정식이다.

φ이 프로베니우스(Frobenius) 맵이고 p가 $GF(q)$의 표수라고 하자. 만약 $Tr(\varphi)=0(\mathrm{mod}\ p)$라면, $GF(q)$에서 타원곡선 E를 초특이(supersingular)라고 한다. 만약 $GF(q)$에서 타원곡선 E가 초특이라면 embedding degree k는 6이하의 값이 된다[16]. 또한 $GF(q)$의 표수가 3인 경우 embedding degree k는 6이고, $GF(q)$의 표수가 2인 경우 embedding degree k는 4가 된다.

Ⅲ. 이진유한체상에서의 기존 연구들

$GF(2^m)$상에서의 초특이 타원곡선 : [16]에 기술된 바와 같이 m이 홀수인 $GF(2^m)$상에서 정의된 embedding degree k가 4인 초특이(supersingular) 타원곡선은 다음과 같이 단 두 개가 존재하며

$$E_b: Y^2+Y=X^3+X+b,\ b=0,1, \quad (3)$$

타원곡선 군 $E_b(GF(2^m))$의 위수가 다음 식을 만

족함을 알 수 있다.

$$|E_b(GF(2^m))| = \begin{cases} 2^m+1+(-1)^k \cdot 2^{\frac{m+1}{2}} \\ \quad \text{if } m \equiv 1,7 (\bmod 8) \\ 2^m+1-(-1)^k \cdot 2^{\frac{m+1}{2}} \\ \quad \text{if } m \equiv 3,5 (\bmod 8) \end{cases} \quad (4)$$

$GF(2^m)$상에서의 **Tate** 페어링의 닫힌 공식: 효율적인 Tate 페어링 계산을 위해서 다음과 같은 distortion map(nontrivial automorphism)이 [10,11]에서와 같이 사용된다.

$$\phi : E_p \rightarrow E_p, \quad \phi(x,y) = (x+s^2, y+sx+t) \quad (5)$$

여기에서 $s^2+s+1=0$이고 $t^2+t+s=0$이다. 즉, $GF(2)(s) = GF(2^2)$, $GF(2)(t) = GF(2^t)$, $s=t^5$, $t^4+t+1=0$ 이고, t는 $GF(2^4)^\times$의 생성원이다. 이때 $p^{2m}(P)-p^{2m}(O)=(f_P)$를 만족하는 함수 f_P는 다음과 같이 나타낼 수 있다.

$$f_P = \prod_{i=1}^{2m} g_{2^{i-1}P}^{2^{2m-i}} = g_P^{2^{2m-1}} g_{2P}^{2^{2m-2}} \cdots g_{2^{2m-2}P}^{2} g_{2^{2m-1}P} \quad (6)$$

$P=(\alpha,\beta)$와 $Q=(x,y)$을 $E_b(GF(2^m))$의 점이라 할 때, [10,11]에서 소개된 것처럼

$$(g_{2^{i-1}P}(\phi(Q)))^{2^{2m-i}} = \alpha^{(i-1)}x^{(-i)} + \beta^{(i-1)} + y^{(-i)} + s(\alpha^{(i-1)} + x^{(-i)}) + t + b$$

을 얻을 수 있다. 여기에서 $\alpha^{(i)}$는 α^{2^i}를 의미하며 이는 $\beta^{(i)}, x^{(i)}, y^{(i)}$에서도 같은 의미이다.

정리 2. [10,11] Tate 페어링

$\tau_l(P,Q) = f_P(\phi(Q))^{2^{2m}-1}$은 다음을 만족한다.

$$f_P(\phi(Q)) = \prod_{i=1}^{m} (\alpha^{(i)}x^{(-i+1)} + \beta^{(i)} + y^{(-i+1)} + s^2(\alpha^{(i)} + x^{(-i+1)}) + t^2 + b)$$

위의 알고리즘은 위의 정리를 간단하게 구현한 것이다.

Tate 페어링 계산의 병렬 알고리즘: 위 알고리즘은 스텝 2에서 1개의 $GF(2^m)$-곱셈, 스텝 3에서 6개의 $GF(2^m)$-곱셈이 필요하다[11]. 정리 2의 일반항 $\alpha^{(i)}x^{(-i+1)} + \beta^{(i)} + y^{(-i+1)} + s^2(\alpha^{(i)} + x^{(-i+1)}) + t^2 + b$을 적당히 변형함으로서 7개의 곱셈을 병렬로 처리할 수 있다는 것이 [18]에서 알려져 있다. 다음은 페어링 계산을 위한 병렬 알고리즘이다.

표 1의 알고리즘을 효율적으로 구현하기 위해서는 7개의 $GF(2^m)$의 곱셈 회로, 2개의 $GF(2^m)$의 제곱회로, 그리고 2개의 $GF(2^m)$의 제곱근 회로가 필요하다. 표 2의 알고리즘에서는 제곱근 회로가 필요 없으며 스텝 2에서의 6개의 $GF(2^m)$-곱셈, 스텝 4에서의 1개의 $GF(2^m)$-곱셈이 모두 병렬로 처리가능하다. 페어링 계산에서는 정규기저보다는 다항식기저가 선호되고 있다. 이는 복잡도가 낮은 가우시안 정규기저보다 해밍 중(Hamming weight)이 낮은 3항식기저가 더 자주 나타나기 때문이다.

표 1. [10,11]에서 소개된 $f_p(\phi(Q))$의 계산을 위한 알고리즘

Input : $P=(\alpha,\beta), Q=(x,y)$
Output : $C=f_P(\phi(Q))$
$C \leftarrow 1$
for($i=1$ to m; i++)
1. $\alpha \leftarrow \alpha^2$, $\beta \leftarrow \beta^2$
2. $z \leftarrow \alpha+x+1$, $w \leftarrow \alpha+x+\alpha x+\beta+y+b$
3. $C \leftarrow C \cdot (w+zs+t)$
4. $x \leftarrow \sqrt{x}$, $y \leftarrow \sqrt{y}$
end for

Ⅳ. 서명 스킴에서의 Tate 페어링

암호학에서 겹선형 페어링의 중요한 장점중 하나는 타원곡선의 순환군상에서의 이산로그문제의 개념만으로 실현하기 힘든 ID기반의 암호화, 서명 등의 아이디어를 구현해 줄 수 있다는 것이다. 이러한 서명 스킴들에서 가장 비용이 많이 드는 연산은 페어링 계산 비용이다. 서명 과정에서 대부분의 서명 스킴들은 타원곡선 상수배 연산만을 요구하며 검증과정에서 페어링의 계산이 필요하다. ECDSA와 비교가 용이한 페어링 서명알고리즘은 BLS 서명 스킴[2]인데 두 스킴 모두 서명자의 ID를 공개키로 이용하지는 않는다. 다음은 BLS 서명스킴의 기본적인 아이디어이다.

만약 (M,M')이 정당한 서명이라면, $M'=dM$로부터 다음 등식을 얻게 된다.

$\tau(M,P') = \tau(M,dP) = \tau(dM,P) = \tau(M',P)$,

그리고 이 서명이 검증되었음을 알게 된다. 이 스킴

의 장점은 $\tau_l(P,Q)=f_P(\phi(Q))^{2^{2m}-1}$의 정확한 값을 알지 못하더라도 $\tau_l(P,Q)=\tau_l(P',Q')$가 성립하는지의 여부만 확인하면 된다는 것이다. 즉 마지막 지수승의 계산 없이 (역원회로 필요 없이) 다음 등식이 성립하는 지만 체크하면 된다.

$$f_P(\phi(Q))^{2^{2m}} f_P(\phi(Q'))=f_P(\phi(Q))f_P(\phi(Q'))^{2^{2m}} \quad (7)$$

그리고 이 연산에 필요한 비용은 단 두 번의 $GF(2^{4m})$ 곱셈이며 이는 $GF(2^m)$ 곱셈기로 구현가능하다. 그러므로 BLS 서명스킴의 검증을 위한 역원회로(inverter)는 필요하지 않은 반면에 ECC 상수배 연산은 (반복되는 곱셈의) Fermat-like 방법을 사용하지 않는 한 좌표 변환 과정이 필요하며 이 과정에서 역원을 구해주는 역원회로가 반드시 필요하다.

V. 페어링 계산과 타원곡선 상수배 연산 사이의 복잡도 비교

BLS 서명 스킴과 ECDSA는 둘 다 서명 스테이지에서 한 번의 ECC 상수배 연산이 필요하므로 서명스테이지의 복잡도는 같다고 볼 수 있다. 다만 BLS 서명 스킴에서는 키 크기 $m\approx 250$인 초특이 타원곡선이 사용되는 반면에 ECDSA에서는 1024-bit RSA에 필적하는 보안성을 갖는 키 크기 $m\approx 160$인 비초특이 타원곡선이 사용된다는 점이다. 비록 BLS 서명 스킴은 ECDSA와 비교했을 때 bandwidth가 열등하지만, 페어링의 계산 과정은 단순한 초특이 타원곡선의 반복적인 포인트 더블링이다. 각 프로토콜의 검증 스테이지를 비교한다면, 가장 비싼 연산은 BLS 서명 스킴에서는 2개의 페어링 계산이고, ECDSA에서는 2개의 ECC 상수배 연산이다.

5.1 ECC 상수배 알고리즘

$P\in E(GF(2^m))$에 대하여 kP를 계산하는 ECC 상수배 연산은 ECDSA와 같이 이산 로그 문제를 기반으로 하는 스킴들에서 사용되는 가장 중요한 연산인데 이를 위한 많은 알고리즘이 있다. 아핀 (affine), 사영 (projective) 그리고 mixed 좌표와 같은 좌표 시스템이 이 같은 목적을 위해 사용되어졌다. 또한 상수의 표현에 대한 다양한 방법[22]으로서 이진(binary), signed binary, Frobenius, 혹은 효율적인 자기준동형사상(endomorphism)을 사용하는 방법 등이 알려져 있다. 각각의 ECC 구현들은 특별한 목적을 위해 특수한 형태를 취하므로 Tate 페어링의 결과를 ECC 연산의 특정한 형태의 구현에 비교하기 위한 시도 보다는 연산 각 단계의 알고리즘 복잡도에 초점을 맞추고자 한다.

적당한 하드웨어를 선택했을 경우 덧셈 및 제곱의 연산은 곱셈연산과 비교하여 무시해도 되는 비용이기 때문에, 지금부터는 덧셈과 제곱에 대한 비용은 고려하지 않는다. Tate 페어링 계산에서 각 단계는 7번의 $GF(2^m)$ 곱셈을 요구한다. 그리고 표 2의 알고리즘에 의하면 이 모든 곱셈연산들은 7개의 곱셈기를 이용하여 병렬로 실행될 수 있다.

타원곡선 상수배 연산의 문제점은 하드웨어적으로 병렬 처리가 쉽지가 않다는 것인데 이는 연산에서 중간 변수의 데이터 종속성 때문이다. 포인트 곱셈의 각 단계는 일반적으로 병렬처리가 힘든 포인트 더블링과 포인트 덧셈을 포함한다. 대부분의 ECC 하드웨어 구현에서는 보통 적합한 디지트(digit) 크기를 갖는 곱셈 유닛을 사용하지만, 본 논문에서는 비교의 공정성을 위해 bit-serial 구조의 곱셈기를 사용한다고 가정한다. 페어링 연산과 달리 ECC 상수배 연산은 역원 연산이 필요하다. 역원 연산을 위해 확장 유클리드 알고리즘을 많이 사용하며, 사용하는 bit-serial 역원회로[20,21]의 공간복잡도는 bit-serial 곱셈기에 비해 대략 5배 정도 높게 나타난다. 따라서 역원 회로가 ECC 상수배 연산의 계산 유닛에 장착되어 있다면, ECC 상수배 연산의 공간 비용은 이미 6개의 곱셈 유닛 (bit serial multiplier)이 필요하므로 페어링 계산에서 필요한 7개의 곱셈 유닛과 비교해 그다지 차이가 없다.

사영좌표계 혹은 Mixed 좌표계가 사용된 대다수의

표 2. [18]에서 소개된 $f_n(\phi(Q))$의 계산을 위한 알고리즘

Input : $P=(\alpha,\beta), Q=(x,y)$
Output : $C=f_P(\phi(Q))$
$C\leftarrow 1,\ \alpha\leftarrow\alpha^4,\ \beta\leftarrow\beta^4$
$u\leftarrow x^2+y^2+b+\frac{m-1}{2},\ v\leftarrow x^2+1,\ \theta\leftarrow\alpha v$
// 초기화
for $(i=1$ to $m\ ;\ i++)$
1. $C\leftarrow C^2$
2. $C\leftarrow C\cdot A$, where
   $A=\beta+\theta+u+(\alpha+v)s+t$
3. $\alpha\leftarrow\alpha^4,\ \beta\leftarrow\beta^4,\ u\leftarrow u+v,\ v\leftarrow v+1$
4. $\theta\leftarrow\alpha v$
end for

표 3. BLS 서명스킴의 기본적인 구조[2]

1. 키 생성: E를 낮은 embedding degree를 갖는 $GF(2^m)$상의 타원곡선이라고 하고 점 P를 $E(GF(2^m))$에서 큰 소수의 위수를 갖는 부분군의 생성원이라고 하자. 서명자는 임의의 정수 d를 선택하여 $P' = dP$를 계산한다. 그러면 P와 P'는 공개키이고 d는 비밀키가 된다.
2. 서명: $E(GF(2^m))$의 원소인 메시지 M을 서명하기 위해, 서명자는 $M' = dM$을 계산하고 순서쌍 (M, M')는 서명이 된다.
3. 검증: 주어진 쌍 (M, M')에 대해서 페어링 $\tau(M, P')$와 $\tau(M', P)$를 계산하고 이 두 개의 값이 같은지를 알아본다. 여기에서 τ은 Tate 페어링이다.

표준 알고리즘은 $GF(2^m)$상의 ECC 상수배 연산 kP에 사용되는 평균적인 비용이 적게는 $6m$개의 $GF(2^m)$ 곱셈이[12,22], 많게는 $14m$개의 $GF(2^m)$ 곱셈이 사용된다[22].

5.2 ECDSA와 BLS 서명스킴 비교

160-비트 ECC는 1024-비트 RSA와 필적하는[23] 보안성을 보인다. 또한 이진 초특이 타원곡선을 사용한 페어링 기반 암호에 대해서, 1024-비트 RSA와 필적하는 보안성을 갖는 m의 크기는 약 250-비트 정도이다. 따라서 이진체상에서 160-비트 ECC 연산과 250-비트 페어링 연산을 비교한다. 공정한 비교를 위해서 $a \approx 160$인 경우의 $GF(2^a)$상에서 ECC 상수배 연산속도와 $b \approx 250$인 경우의 $GF(2^b)$상에서 Tate 페어링 계산 속도를 비교하며, BLS 서명 스킴은 [18]에서 소개된 연산기법을, ECC 상수배 연산은 [27]에서 소개된 연산기법을 각각 사용한다. [27]은 다수개의 연산기를 사용하는 최근의 대표적 연구결과로서, 변형된 Lopez-Dahab 알고리즘을 사용하며, 3개의 곱셈기와 1개의 역원 연산기를 사용한다. 두 스킴의 기본적인 특징은 아래 표 4와 같다.

표 4에서 하드웨어 및 시간 복잡도는 개략적인 비교결과이다. 정확한 하드웨어 복잡도의 비교를 위해서는 트랜지스터 레벨에서 비교가 되어야 하지만 회로는 구현 기술에 따라 많은 차이가 있기 때문에 정확한 비교는 비교적 어려운 문제이다. 따라서 다음과 같이 산출한다. $GF(2^m)$상에서 선형 곱셈기 (linear or bit-serial)의 공간 복잡도는 비트 크기 m에 비례한다고 가정한다. 따라서 $GF(2^{250})$상의 Tate 페어링 계산에서 7개의 곱셈 유닛의 공간복잡도의 비율과 $GF(2^{160})$상의 ECC 상수배 연산에서 3개의 곱셈 유닛의 공간복잡도의 비율은 $7 \cdot 250/3 \cdot 160 = 3.65$이다. 여기에서 ECC 상수배 연산에는 역원 연산이 사용되며, 확장된 유클리드 알고리즘을 사용할 경우 공간복잡도는 다항식 기저의 bit-serial 곱셈기의 공간복잡도보다 약 5배 정도의 복잡도를 갖는다[20,21]. 따라서 역원 연산기의 복잡도를 고려하면 페어링 연산기는 ECC 상수배 연산기에 비해 약 $7 \cdot 250/8 \cdot 160 = 1.37$배의 하드웨어 복잡도를 가진다.

시간 복잡도의 정확한 산출을 위해서는 계산 지연시간과 최대 동작 주파수 모두를 비교하여야 하나 연산기 외에 제어유닛, 메모리 등 다양한 유닛들이 포함되기 때문에 표 4에서는 계산지연 시간만을 고려하였다. 참고로 페어링 연산기는 알고리즘이 매우 간단하기 때문에 제어유닛의 최장 경로 지연(critical path delay)는 ECC의 상수배 연산기 보다 매우 적을 것으로 예측된다. 표 4에서 페어링 알고리즘은 250개의 루프가 필요하며 7개의 bit-serial 연산기를 병렬로 사용하기 때문에 $250^2 = 62{,}500$사이클이 소요된다. ECC

표 4. $GF(2^m)$상에서 ECDSA와 BLS 서명 스킴의 비교

검증 과정의 비교	BLS 서명 스킴: 한 번의 페어링 계산	ECDSA: 한 번의 ECC 상수배 연산	
핵심 연산	$(\cdots(((A_1)^2A_2)^2A_3)^2\cdots)^2A_m$	$2(\cdots(2(2P+k_1P)+k_2P)+\cdots)+k_mP$	
필요한 곱셈회로의 갯수	7	3	
필요한 역원회로의 갯수	없음	1	
Clock Cycles	$250^2 = 62{,}500$	Main Loop	$160\times2\times160 = 51{,}200$
		좌표변환	$(3\times320)+(5\times160) = 1{,}760$
		Total	52,960
제어 유닛의 구조	단순함	복잡함	
1024-비트 RSA와 비교되는 키 크기	$m \approx 250$	$m \approx 160$	

상수배 연산의 경우 메인 루프는 160이며, 변형된 Lopez-Dahab알고리즘에서는 3개의 bit-serial 연산기를 사용하기 때문에 $160\times2\times160=51{,}200$사이클이다. 또한 좌표 변환에서는 3번의 역원연산과 5번의 곱셈연산을 필요로 한다. 여기서 역원연산은 사이클이 필요하다. 따라서 전체적으로 52,960의 사이클이 소요된다. 즉, 페어링계산은 병렬연산이 가능하기 때문에 $m\approx250$라는 큰 필드사이즈를 이용함에도 속도 측면에서는 ECC 상수배 연산에 비해 $62{,}500/52{,}960=1.18$배 정도의 차이를 보임을 알 수 있다.

뿐만 아니라 완전한 ECDSA를 위해서는 $GF(q)$에서의 역원 연산을 필요로 한다. 여기서 q는 생성원 $P\in E(GF(2^m))$의 위수이다. 이는 dual field 연산기의 필요성을 의미하며, dual field 연산기는 전가산기(full adder)를 기본적으로 요구하므로 ECDSA구현의 복잡도를 증가시킨다. 따라서 실제적인 하드웨어 복잡도 측면을 고려하면 ECDSA와 BLS 스킴은 거의 비슷한 수준의 복잡도를 보일 것으로 예상된다.

VI. 결 론

이진체상에서 초특이 타원곡선의 Tate 페어링 계산과 비초특이 타원곡선의 상수배 연산을 비교하였다. 페어링 연산 알고리즘은 실행하기 쉽고 구현회로가 ECC 상수배 연산과 비교해서 단순한 구조를 갖는다. 250비트 BLS 스킴을 160비트 ECDSA와 비교했을 경우 하드웨어 복잡도 측면에서는 약 1.38배 증가하였고 계산속도 측면에서는 약 1.18배 정도 느리다. 이러한 비교는 실제 ECDSA의 구현에서 나타나는 복잡한 제어유닛이나 dual field 연산기를 고려하지 않았음을 감안할 때 BLS서명스킴의 구현복잡도가 ECDSA의 그것과 많은 차이가 나지 않음을 설명해준다. 따라서 BLS 서명 스킴이 서명을 필요로 하는 다양한 응용에 사용될 수 있음을 의미한다.

참 고 문 헌

[1] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," *Crypto 2001, Lecture Notes in Computer Science*, Vol.2139, pp.213-229, 2001.

[2] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Asiacrypt 2001, Lecture Notes in Computer Science*, Vol.2248, pp.514-532, 2002.

[3] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *ANTS 2000, Lecture Notes in Computer Science*, Vol.1838, pp.385-394, 2000.

[4] N.P. Smart, "An identity based authentication key agreement protocol based on pairing," *Electronics Letters*, Vol.38, pp.630-632, 2002.

[5] R. Granger, D. Page, and M. Stam, "Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three," preprint, available at http://eprint.iacr.org/2004/157.pdf, 2004.

[6] R. Granger, D. Page, and M. Stam, "On small characteristic algebraic tori in pairing based cryptography," *LMS J. Comput. Math.*, Vol.9, pp.64-85, 2006.

[7] 장남수, 김태현, 김창한, 한동국, 김호원, "페어링 기반 암호시스템의 효율적인 유한체 연산기," 정보보호학회 논문지, Vol.18, pp.33-34, 2008.

[8] I. Duursma and H. Lee, "Tate pairing implementation for hyperelliptic curves," *Asiacrypt 2003, Lecture Notes in Computer Science*, Vol.2894, pp.111-123, 2003.

[9] P. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing based cryptosystems," *Crypto 2002, Lecture Notes in Computer Science*, Vol.2442, pp.354-368, 2002.

[10] P. Barreto, S. Galbraith, C. O hEigeartaigh, and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Design, Codes and Cryptography*, Vol.42, No.3, pp.239-271, 2007.

[11] S. Kwon, "Efficient Tate pairing computation for supersingular elliptic curves over binary fields," *ACISP 2005, Lecture Notes in Computer Science*, Vol.3574, pp.134-145, 2005.

[12] J. Lopez and R. Dahab, "Fast multiplication on elliptic curves over GF(2^m) without precomputation," *CHES 1999, Lecture Notes in Computer Science*, Vol.1717, pp.316-327, 1999.

[13] N. Gura, S.C. Shantz, H. Eberle, S. Gupta, V. Gupta, D. Finchelstein, E. Goupy, and D. Stebila, "An end-to-end systems approach to elliptic curve cryptography," *CHES 2002,*

Leture Notes in Computer Science, Vol.2523, pp.349-365, 2003.

[14] NIST, "Digital Signature Standard," *FIPS Publication*, 186-2, February, 2000.

[15] F. Hess, "A Note on the Tate pairing of curves over finite fields," *Arch. Math. Vol.82*, pp.28-32, 2004.

[16] A.J. Menezes, Elliptic Curve Public Key Cryptosystems, *Kluwer Academic Publisher*, 1993.

[17] H. Wu, "On complexity of polynomial basis squaring in GF(2^m)," *SAC 2000, Lecture Notes in Computer Science*, Vol.2012, pp.118-129, 2001.

[18] C. Shu, S. Kwon, and K. Gaj, "FPGA accelerated Tate pairing based cryptosystems over binary fields," *FPT 2006, IEEE Internatonal Conference on Field Programmable Technology*, pp.173-180, 2006.

[19] V. Miller, "Short programs for functions on curves," unpublished manuscript, 1986.

[20] H. Brunner, A. Curiger, and M. Hofstetter, "On computing multiplicative inverses in GF(2^m)," *IEEE Trans. Computers*, Vol.42, pp.1010-1015, 1993.

[21] C.H. Kim and C.P. Hong, "High-speed division architecture for GF(2^m)," *Electronics letters*, Vol.38, pp.835-836, 2002.

[22] D. Hankerson, A.J. Menezes, and S.A. Vanstone, Guide to Elliptic Curve Cryptography, Springer-Verlag, 2004.

[23] A.K. Lenstra and E.R. Verheul, "Selecting cryptographic key sizes," *J. Cryptology*, Vol.14, pp.255-293, 2001

[24] K. Fong, D. Hankerson, J. Lopez, and A. Menezes, "Field inversion and point halving revisited," Technical Report CORR 2003-18, Univ. of Waterloo, 2003.

[25] C. Shu, S. Kwon, and K. Gaj, "Reconfigurable Computing Approach for Tate Pairing Cryptosystems over Binary Fields," *IEEE Trans. Computers*, Vol.58, No.8, pp.1221-1237, 2009.

[26] D. Hankerson, J.L. Hernandez, and A.J. Menezes, "Software implementation of elliptic curve cryptography over binary fields," *CHES 2000, Leture Notes in Computer Science*, Vol.1965, pp.1-24, 2000.

[27] C.H. Kim, S. Kwon, and C.P. Hong "FPGA implementation of high performance elliptic curve cryptographic processor over GF(2163)", *Journal of Systems Architecture*, Vol.54, pp.893-900, 2008.

구 남 훈 (Namhun Koo) 정회원

2007년 8월 성균관대학교 수학과 학사

2009년 2월 성균관대학교 수학과 석사

2009년 3월~현재 성균관대학교 수학과 박사과정

<관심분야> 공개키 암호 시스템, 타원곡선 암호시스템, Pairing 기반 암호시스템, NTRU 암호시스템, USN 보안

조 국 화 (Gook Hwa Jo) 준회원

2007년 2월 전북대학교 수학과 학사

2009년 3월~현재 성균관대학교 수학과 석사과정

<관심분야> 정수론, 암호학, 타원곡선, 공개키 암호시스템

김 창 훈 (Chang Hoon Kim) 정회원

2001년 2월 대구대학교 컴퓨터정보공학부 학사
2003년 2월 대구대학교 컴퓨터정보공학과 석사
2006년 8월 대구대학교 컴퓨터정보공학과 박사
2007년 8월~현재 대구대학교 컴퓨터IT공학부 조교수

<관심분야> 암호 시스템, Embedded System, RFID / USN 보안

권 순 학 (Soonhak Kwon) 정회원

1990년 2월 KAIST 수학과 학사
1992년 2월 서울대학교 수학과 석사
1997년 5월 Johns Hopkins University 박사
1998년 3월~현재 성균관대학교 수학과 교수

<관심분야> 정수론, 암호론, Cryptographic Hardware