ECDLP 병렬 계산에서 특정점 비율에 대한 분석

김병관, 이창우, 강주성[1)]
국민대학교 수학과
e-mail : jskang@kookmin.ac.kr

An Analysis for the Distinguished Points in the Parallel Computing of ECDLP

Byung-Gwan Kim, Chang-Woo Lee, Ju-Sung Kang
Dept. of Mathematics, Kookmin University

요 약

특정점 개념을 적용한 ECDLP 병렬 계산 방법은 현재까지 가장 효율적인 것으로 알려져 있다. 이론상으로는 특정점에 대한 비율이 증가할수록 충돌쌍을 발견할 때까지의 계산량 및 수행시간은 점진적으로 감소한다. 하지만 우리의 실험적 결과는 특정점 비율이 일정 수준 이상 증가할 경우 오히려 계산량 및 수행시간도 증가함을 보여준다. 클러스터 환경 실험에서 얻은 이러한 결과를 바탕으로 본 논문에서는 통신부하를 고려한 실질적 수행시간을 이론적으로 분석함으로써 실험적 결과가 합리적임을 밝힌다. 더욱이 계산 환경에 맞는 특정점 비율을 축소 모델링을 통하여 결정할 수 있는 메커니즘을 제안한다.

1. 서론

현재 ECDLP 계산 알고리즘 중에서 가장 효율적인 방법은 Pollard-rho[1] 알고리즘을 병렬화한 것으로 알려져 있다. 이 공격법은 Oorschot-Wiener[2]가 제안한 특정점(Distinguished Point, DP)을 저장하여 충돌쌍을 찾는 방식으로 Pollard-rho 알고리즘을 병렬화한 것이다. 여기에서 DP는 특정한 성질을 만족시키는 점(예를 들어 50-bit 데이터에서 상위 15-bit가 0인 점)을 말한다. Oorschot-Wiener[2]의 연구결과에 의하면 전체 점의 개수가 n이고 DP의 비율이 θ일 때, DP를 이용한 병렬 계산 방법의 수행시간은 m개의 동일한 성능의 프로세서를 사용할 경우 약 $\sqrt{\pi n/2}/m+1/\theta$이다. 이 수행시간은 프로세서의 개수 m에 대하여 선형적으로 감소한다는 데에 큰 의미가 있다. DP를 이용하지 않는 직접적인 병렬 계산법은 $\sqrt{m}$에 비례하여 수행시간이 감소하기 때문에 DP를 도입한 병렬화는 대단히 효율적인 것이라 할 수 있다.

하나의 DP를 발견하는 데 수행되는 계산량의 기댓값은 $1/\theta$이므로 DP의 비율인 θ가 증가할수록 충돌쌍이 발견될 때까지의 계산량은 상대적으로 감소한다. 하지만 분산환경이나 클러스터와 같은 실질적인 실험 환경에서는 이와 같은 이론적 결과와 달리 θ의 특정 지점까지는 평균 계산량과 수행시간이 감소하지만 그 이후에는 실험 환경의 여러 요소들로 인하여 계산량과 수행시간이 급격히 증가하는 현상을 관찰할 수 있다.

본 논문에서는 통신부하를 고려한 실질적 수행시간을 이론적으로 분석함으로써 우리가 얻은 실험적 결과가 합리적이라는 사실을 밝힌다. 또한, 계산을 실질적으로 수행하는 환경에 적합한 DP 비율을 축소 모델링을 통하여 결정할 수 있는 메커니즘을 제안한다.

1) 교신저자

2. ECDLP를 위한 병렬 프로그램

2.1 ECDLP 알고리즘

대부분의 공개키 암호시스템의 안전성은 수학적인 난제를 기반으로 하고 있다. 대표적인 공개키 암호시스템의 하나인 타원곡선 암호시스템은 Koblitz[3]와 Miller[4]에 의해 제안된 시스템으로 안전성은 타원곡선 이산대수문제(Elliptic Curve Discrete Logarithm Problem, ECDLP)에 기반을 두고 있다. 다음은 ECDLP의 정의를 나타낸다.

ECDLP
타원곡선 군 E, 위수가 n인 점 $P\in E$ 그리고 점 $Q\in E$가 주어졌을 때, $dP=Q$를 만족시키는 d를 찾는 문제

ECDLP 계산에 가장 많이 사용되는 알고리즘은 Pollard-rho 알고리즘이다. 이 알고리즘은 ρ자 형태의 수열을 나타내는 확률보행(random walk)[5]을 이용하는 알고리즘으로 ECDLP가 충돌쌍 발견 문제와 동치라는 사실을 이용한다.

$$X_i=a_iP+b_iQ,\ \ X_j=a_jP+b_jQ$$
$$X_i=X_j,\ \ a_i\neq a_j,\ \ b_i\neq b_j$$

를 만족하는 두 충돌쌍 (X_i,a_i,b_i)와 (X_j,a_j,b_j)가 발견되면 다음과 같은 계산으로 ECDLP를 해결할 수 있는 것이다.

$$a_iP+b_iQ=a_jP+b_jQ \Leftrightarrow d=(a_i-a_j)(b_j-b_i)^{-1} \bmod n.$$

2.2 병렬화 알고리즘

Oorschot-Wiener[2]는 Pollard-rho[1]알고리즘을 병렬화 할 수 있는 두 가지 방식을 제안하였는데 하나는 직접적인 병렬화이고, 또 다른 하나는 DP를 이용한 병렬화 방법이다. 이 두 가지 알고리즘들은 클러스터와 같은 슈퍼컴

퓨터나 인터넷 기반의 분산 컴퓨팅 환경에 모두 적용될 수 있다. m개의 프로세서로 병렬 계산할 경우 직접적인 병렬화의 계산량은 $\sqrt{\pi n/2m}$이나, DP를 이용한 병렬화는 $\sqrt{\pi n/2}/m$이 된다. 이는 만약 100개의 프로세서로 병렬 계산을 수행한다면 직접적인 병렬화가 단일 프로세서 계산량의 약 10배 정도의 효과를 얻을 수 있는 반면, DP를 이용한 방법은 약 100 정도의 효과를 볼 수 있다는 의미이다. 그래서 현재 ECDLP 공격에 주로 이용되는 방법은 DP를 이용한 병렬화 기법이다.

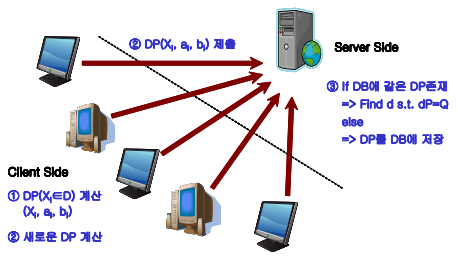
DP를 이용한 ECDLP 계산법은 충돌쌍을 찾는 병렬화 방법으로 Pollard-rho 알고리즘과 결합되어 사용되기 때문에 Parallel Pollard-rho라는 말을 주로 사용한다. 우리는 ECDLP 계산을 위한 DP를 다음과 같이 정의한다.

특정점(Distinguished Point)
n 비트 타원곡선 군 E에서 점 $X(\in E)$의 x 좌표 값이 상위 $n-x$ 비트 모두 0인 성질을 만족시킬 때, 점 X를 x-bit DP라고 부른다.

n 비트 타원곡선 군 E에서 x-bit DP의 집합을 D_x라고 하면 DP 비율은 다음과 같이 나타낼 수 있다.

$$\theta_{n-x} = \frac{|D_x|}{|E|} \approx 2^{x-n}.$$

위 식에서 x값이 설정되면 DP의 비율 θ_{n-x}가 결정되며, 결정된 DP의 성질을 이용하여 다음 그림 1과 같은 절차에 의해서 Parallel Pollard-rho 알고리즘을 구현할 수 있다.



(그림 1) Parallel Pollard-rho 수행 절차

Parallel Pollard-rho 알고리즘을 구현하기 위하여 그림 1과 같이 개념적으로 클라이언트, 서버, 데이터베이스 3가지 구성요소가 필요하다. 각 클라이언트는 Pollard-rho의 확률 보행을 이용하여 랜덤한 점을 시작으로 정해진 성질을 만족시키는 DP를 탐색한다. 만약 DP가 발견되면 그 점을 서버에 전송하고 다시 새로운 시작점에서 DP 탐색을 계속한다. 서버는 클라이언트로부터 DP를 수신하여 데이터베이스에 똑같은 점이 있는지 쿼리를 통해 확인한다. 만약 똑같은 점이 존재하면 ECDLP를 계산하여 결과를 클라이언트에 통보한다. 그렇지 않으면 수신된 DP를 데이터베이스에 저장한다.

2.3 ECDLP 연구 동향

타원곡선 암호시스템 개발의 선두주자인 캐나다 Certicom사[6]는 타원곡선 암호시스템의 실질적인 안전성에 관한 연구의 활성화를 위해서 1997년 11월부터 현재까지 지속적으로 ECC챌린지를 실시하고 있다. ECC챌린지는 유한체 F_p 타원곡선을 다루는 ECCp, 유한체 F_{2^m} 타원곡선을 다루는 ECC2, 마지막으로 Koblitz 타원곡선을 다루는 ECC2K 등 3가지 형태의 챌린지를 진행하고 있다. 다음은 Certicom사에서 제공한 ECC 챌린지의 기록현황을 나타낸다.

<표 1> ECC 챌린지 공식 기록

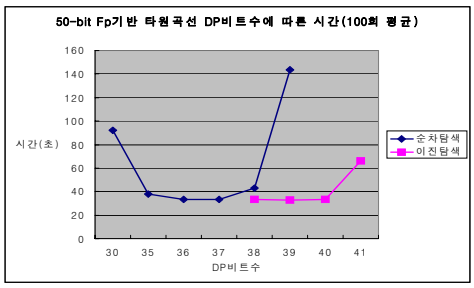
Challenge	End Date	Elliptic Curve Operations	Iterations per second*	Machine Days*
ECC2-79	Dec. 16, 1997	1.7×10^{12}	170000	116
ECC2-89	Feb. 9, 1998	1.8×10^{13}	187000	1114
ECC2K-95	May 21, 1998	2.2×10^{13}	149000	1709
ECC2-97	Sep. 22, 1999	1.2×10^{14}	227000	6118
ECC2K-108	Apr 4, 2000	2.3×10^{15}	160364	166000
ECCp-79	Dec. 6, 1997	1.4×10^{12}	314000	52
ECCp-89	Jan. 12, 1998	2.4×10^{13}	388000	716
ECCp-97	Mar. 18, 1998	2.0×10^{14}	361000	6412

*on 500 MHz Digital Alpha workstation running Linux

<표 1>의 공식기록 이외에 ECCp-109는 약 1만대 PC의 분산공격으로 2002년 12월에 해결되었고, ECC2-109는 약 2600만대 PC로 2004년 4월에 종료되었다. 현재는 131-bit에 대한 챌린지가 진행 중에 있다. 지금까지 해결된 모든 챌린지는 Pollard-rho 기반의 공격법이 적용되었다. Certicom사의 ECC챌린지 문서에 따르면 Parallelized Brent-type Cycle-finding 알고리즘을 이용한 ECCp-79를 제외하고 대부분 Parallel Pollard-rho 알고리즘을 사용한 것을 볼 수 있다. 그러므로 컴퓨터 연산 능력과 인터넷의 지속적인 발전을 감안할 경우, 타원곡선 암호의 안전성에 대한 연구는 Parallel Pollard-rho 알고리즘을 적용한 분산 환경이나 클러스터 환경 하에서의 계산복잡도 관점의 분석이 가장 실용적이고 중요할 것으로 생각된다.

3. DP 비율에 따른 계산효율성

우리는 DP 비율에 따른 이론적인 값과 실험적인 값을 비교하기 위하여 위수가 50-bit인 F_p기반 타원곡선을 사용하였다. 10 개의 표본 (P, Q) 쌍을 랜덤하게 생성한 후 각 표본 당 10회씩 반복하여 총 100회의 실험을 실시하였다. 데이터베이스에서 충돌쌍을 찾기 위한 DP 쿼리 방식은 순차탐색과 이진탐색[7] 두 가지 방식을 각각 실험하였다. 실험에 사용된 클러스터 제원은 각 노드별로 Linux Intel Xeon 3.2GHz/2M(Dual Procssor) 2GB 이고, 순차탐색에는 30-bit와 35-bit에서 39-bit까지의 DP를 실험하였으며, 이진탐색에는 38-bit에서 41-bit까지의 DP를 실험하였다. 실험에 참여한 클라이언트는 25 개의 프로세서이며, 한 대의 서버를 별도로 운영하였다.



(그림 2) DP비율에 따른 계산 효율성

그림 2는 실험 결과를 그래프로 표현한 것이다. 이 실험 결과를 살펴보면 서버에서 순차탐색을 이용할 경우 DP 비트수가 37-bit, 이진탐색일 경우 DP 비트수가 40-bit일 때 이론값에 가장 가깝다는 것을 알 수 있다. 이 값들은 서버가 클라이언트로부터 DP를 수신하는 간격과 서버의 탐색 시간 사이에 지연(delay)이 거의 발생하지 않는 지점으로 볼 수 있다. 이러한 지점은 DP를 수신하는 통신시간과 충돌쌍을 탐색하는 계산시간이 조화를 잘 이루고 있는 상태로 생각된다. 우리는 이러한 지점을 harmonic-point로 부르기로 한다.

4. DP 비율에 관한 이론적 분석

클러스터 시스템을 이용한 실질적인 실험 결과로 얻은 그림 2의 결과를 이론적으로 분석하기 위하여 다음과 같은 기호를 정의한다.

- θ : 타원곡선 상에서 DP의 비율
- m : 프로세서의 개수
- t : 하나의 iteration을 수행하는 시간(상수)
- $N_{\theta,m}(\tau)$: τ초 동안 서버에 저장되는 DP 개수를 나타내는 확률변수
- I : 서버에 DP가 도착하는 간격(초)을 나타내는 확률변수
- $S(d)$: 데이터 양이 d일 때, 서버에서 충돌쌍을 탐색하는 계산시간

DP를 찾는 과정은 파라미터가 θ인 기하확률변수로 모델링되기 때문에 $N_{\theta,m}(\tau)$에 대한 기댓값은 다음과 같다.

$$E[N_{\theta,m}(\tau)] = \frac{\theta}{t} \cdot m \cdot \tau$$

또한, 충돌쌍이 발견될 때까지의 수행시간 T에 대한 기댓값은

$$E[T] = \left(\sqrt{\frac{\pi n}{2}} \cdot \frac{1}{m} + \frac{1}{\theta} \right) t$$

임이 알려져 있다. 이를 보면 θ가 증가할수록 조금씩 수행시간이 감소할 것이라 예상되지만, 실제 실험에서는 θ가 일정 수준 이상 증가한 후에는 오히려 수행시간이 늘어나는 것을 확인하였다. 이는 클라이언트에서 생성되는 DP 개수와 충돌이 발생할 때까지 필요한 전체 DP 개수가 θ에 비례하여 증가하기 때문에 상대적으로 서버의 데이터베이스에 부하가 생기는 현상으로 해석된다. 이를 합리적으로 분석하기 위하여 통신부하를 정의하자.

통신부하

클라이언트에서 생성한 DP가 서버에 전송되는 간격이 충돌쌍 탐색 시간보다 작을 경우 통신부하가 발생했다고 한다. 즉, $I < S(d)$ 이면 통신부하가 발생한다.

순차탐색(sequential search)의 계산량은 데이터 양에 비례하므로 탐색 시간은 $S(\alpha \cdot d) = \alpha \cdot S(d)$를 만족하고, 이진탐색(binary search)에 대한 $S(d)$는 $\log_2 d$에 비례하므로 $S(d^\alpha) = \alpha \cdot S(d)$라는 관계식을 만족한다.

한편, 통신부하에 대한 대비책으로 서버의 데이터베이스를 적절히 분할하는 방법을 고려할 수 있다. 탐색시간을 줄이기 위하여 데이터베이스를 k개로 분할할 경우의 충돌쌍 탐색시간을 $S_k(d)$라 놓으면 $S_1(d) \approx S_k(kd)$를 만족하기 때문에 데이터베이스를 적절히 분할함으로써 탐색시간을 줄일 수 있는 것이다.

통신부하의 정의에서 기댓값을 고려하면, 고정된 τ초 동안 통신부하가 발생했다는 의미는 $S(N_{\theta,m}(\tau)) > E[I]$이 성립한다는 것이고, $E[I] \simeq \tau / N_{\theta,m}(\tau)$ 임을 알 수 있다. 이러한 관계식을 바탕으로 우리는 축소모델링을 통하여 원하는 harmoic-point를 합리적으로 추정할 수 있다. 먼저, 축소모델인 n'-bit 타원곡선에서, m'개의 프로세서로 x'-bit DP를 이용하여 최적의 파라미터인 DP 비율 θ'이 결정되었다고 하자. $E[S(N_{\theta',m'}(T))] < E[I]$이 성립하는 최대값이 x'-bit가 되는 θ'이 최적의 파라미터이다. 다음으로 고정된 θ'에 대하여 통신부하가 발생하지 않는 최대의 DP 개수 $D_{\theta'}$을 구한다. 즉, 고정된 τ에 대하여 $D_{\theta'}$은 $S(D_{\theta'}) \le \tau / N_{\theta',m'}(\tau)$ 을 만족하는 최대값으로 놓는다.

이제 우리의 실제 공격 대상인 n-bit 타원곡선에서 m개의 프로세서를 이용하여 병렬 계산할 경우, 최적의 DP 비율 θ를 다음과 같은 관계식을 통하여 합리적으로 추정할 수 있다. 순차탐색의 경우 다음을 만족한다.

$$\begin{aligned} S(D_{\theta'}) = E[I] &\Leftrightarrow S(D_{\theta'}) = \frac{\tau}{\frac{\theta' m'}{t}\tau} \\ &\Leftrightarrow \frac{\theta' m'}{t} S(D_{\theta'}) = 1 \\ &\Leftrightarrow \frac{\theta m}{t} \frac{m'}{m} 2^{n-x-(x'-n')} S(D_{\theta'}) = 1 \end{aligned}$$

이로부터 $S(\frac{m'}{m} 2^{n-x-(x'-n')} D_{\theta'}) = t/(\theta m)$를 얻을 수 있으며, 이진탐색의 경우에는 유사한 논리에 의하여 $S(D_{\theta'}^{\frac{m'}{m}2^{n-x-(x'-n')}}) = t/(\theta m)$을 알 수 있다. 데이터베이스를 k개로 분할할 경우에는 순차탐색과 이진탐색 각각

$$S_k(k \frac{m'}{m} 2^{n-x-(x'-n')} D_{\theta'}) = t/(\theta m),$$

$$S_k(k D_{\theta'}^{\frac{m'}{m}2^{n-x-(x'-n')}}) = t/(\theta m)$$

의 관계식을 얻을 수 있다.

한편, 충돌쌍이 발견될 때까지 필요한 전체 DP 개수 $N_{\theta,m}(T)$의 기댓값을 Λ_θ이라 놓으면, 전체적으로 통신부하가 발생하지 않았다는 것은 $S(\Lambda_\theta) \le t/(\theta m)$이 성립한다는 의미이므로 순차탐색과 이진탐색 각각의 경우,

$$S_k(\Lambda_\theta) \le S_k(k \frac{m'}{m} 2^{n-x-(x'-n')} D_{\theta'}),$$

$$S_k(\Lambda_\theta) \le S_k(k D_{\theta'}^{\frac{m'}{m}2^{n-x-(x'-n')}})$$

이 성립하는 Λ_θ이 최적의 DP 개수에 대한 기댓값일 것이다. $S(d)$는 증가함수이므로 순차탐색과 이진탐색 각각

$$\Lambda_\theta \le k \frac{m'}{m} 2^{n-x-(x'-n')} D_{\theta'}, \quad \Lambda_\theta \le k D_{\theta'}^{\frac{m'}{m}2^{n-x-(x'-n')}}$$

이 성립한다. 이 두 개의 부등식을 우리가 결정하고자 하는 x에 대하여 정리하면, 순차탐색의 경우

$$x \le \frac{1}{2} \log_2 \left(\frac{m' k D_{\theta'}}{m \sqrt{\pi}} \right) + \frac{3n}{4} + \frac{x' - n'}{2} + \frac{1}{2}$$

를 얻을 수 있고, 이진탐색의 경우

$$x \le 2^{-x}\left(\frac{m'}{m}2^{n+x'-n'}\log_2 D_{\theta'}\right)+\log_2\frac{k}{\sqrt{\pi}}+\frac{n}{2}+1$$

이라는 관계식을 얻을 수 있다. 여기에서 Λ_θ 의 값은 충돌쌍 발견 시까지 필요한 DP의 개수이므로 클라이언트에서 DP를 찾는 과정이 랜덤하다고 가정하면, DP 발견 시까지 테스트되는 점의 개수 $\sqrt{\pi|E|/2}$ 중에서 θ 비율만큼으로 생각할 수 있다. 즉, $\Lambda_\theta = \theta \cdot \sqrt{\pi|E|/2} \approx \theta\sqrt{\pi 2^n/2}$ 이 성립한다.

우리는 위에서 축소모델링에 의하여 원하는 공격 대상인 타원곡선에 대한 ECDLP 공격 시 적절한 DP 비율을 합리적으로 결정하는 메커니즘을 살펴보았다. 이를 구체적으로 설명하기 위하여 다음 두 개의 예를 제시한다.

예1) 축소모델링 할 수 있는 파라미터가 n'=50, x'=37, m'=25 (위수가 50-bit인 타원곡선, 37-bit DP, 25개 프로세서)일 때, 실험적으로 순차탐색을 이용하여 $D_{\theta'}$=12,000이라는 결과를 얻었다면, n=60, m=50, $k=1$(위수가 60-bit인 타원곡선, 50개 프로세서, 1개 데이터베이스)일 때, 효율적인 계산량을 갖는 DP는 44-bit 라는 사실을 다음과 같이 얻을 수 있다.

$$\frac{1}{2}\log_2\left(\frac{m'kD_{\theta'}}{m\sqrt{\pi}}\right)+\frac{3n}{4}+\frac{x'-n'}{2}+\frac{1}{2}=44.86$$

$$x \le 44.86 \quad \therefore x_{optimal}=44 \ .$$

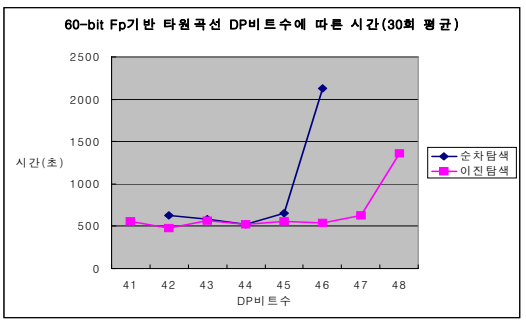
예2) 축소모델링 가능한 파라미터가 n'=50, x'=39, m'=25 일 때, 실험적으로 이진탐색을 이용하여 $D_{\theta'}$=70,000이라는 결과를 얻었다면, n=60, m=50, $k=1$(위수가 60-bit인 타원곡선, 50개 프로세서, 1개 데이터베이스)일 때, 효율적인 계산량을 갖는 DP는 47-bit 라는 사실을 다음과 같이 얻을 수 있다.

$$x \le 2^{48-x} \bullet (16.1)+30.18$$

$$x=49 \Rightarrow 49 > 38.23$$
$$x=48 \Rightarrow 48 > 46.28$$
$$x=47 \Rightarrow 47 < 62.38$$

$$\therefore x_{optimal}=47 \ .$$

예1과 2의 이론적인 분석 결과를 가지고 실제로 클러스터 환경에서 시뮬레이션을 실시하였다. 그 결과는 그림 3과 같다. 시뮬레이션 결과는 위에서 얻은 결과가 최적의 DP 비율에 가깝다는 사실을 뒷받침해주고 있다.



(그림 3) DP비율에 따른 시간(초)

5. 결론

본 논문에서는 DP 개념을 적용한 ECDLP 병렬 계산 방법에 대하여 심도있는 분석을 실시하였다. 이론상으로는 DP에 대한 비율이 증가할수록 충돌쌍을 발견할 때까지의 계산량 및 수행시간은 점진적으로 감소한다. 하지만 우리는 실험적 결과로 DP 비율이 일정 수준 이상 증가할 경우 오히려 계산량 및 수행시간도 증가하는 현상을 발견할 수 있었다. 클러스터 환경 실험에서 얻은 이러한 결과를 바탕으로 통신부하를 고려한 실질적 수행시간을 이론적으로 분석함으로써 실험적 결과가 합리적임을 밝혔다. 더욱이 계산 환경에 맞는 DP 비율을 축소 모델링을 통하여 결정할 수 있는 메커니즘을 제안하였다.

우리가 제안한 DP 비율 결정 메커니즘은 통신부하를 고려하지 않는 이론적인 결과에 비하여 실질적인 구현 환경을 고려한 것이므로 대단히 실용적인 연구 결과라 할 수 있다. 향후 좀 더 다양한 시뮬레이션과 세밀한 분석을 통하여 개선된 연구 결과를 얻고자 한다.

참고문헌

[1] J. M. Pollard, "Monte Carlo methods for index computation (mod p)", Mathematics of Computation Vol. 32(1978), No. 143, pp. 918-924.

[2] P. C. van Oorschot, M. J. Wiener "Parallel collision search with cryptanalytic applications" Journal of Cryptology, 12 (1999), 1-28.

[3] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation,* volume 48, pages 203-209, 1987.

[4] V. Miller, "Uses of elliptic curves in cryptography", *Advances in Cryptology - CRYPTO'85,* Lecture Notes in Computer Science, volume 218, Springer-Verlag, pages 417-426, 1986.

[5] E. Teske, "Better Random Walks for Pollard's rho method", Research Report CORR 98-52, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada. November 1998.

[6] www.certicom.com

[7] F. M. Carrano, W. Savitch, "Chapter 16 Searching", Data Structures and Abstractions with Java, pp. 355-373