情 報 保 護 學 會 誌
第 13 券 第 4 號, 2003. 8

XTR 공개키 암호 알고리즘의 효율성 분석

김 근 옥*, 곽 진*, 김 승 주**, 원 동 호***

요 약

최근 무선 인터넷의 발달과 함께 제약사항이 많은 무선 단말기에 적용 가능한 암호학적 알고리즘들의 필요성이 대두되었다. 그 중 2000년 Lenstra에 의해 제안된 XTR은 $GF(p^6)$상의 서브그룹(subgroup)에서 안전한 암호 연산을 수행하며, 기존의 알고리즘에 비해 파라미터 선택과, 원소들의 트레이스(trace)를 이용한 효율적인 암호 연산으로 인해 무선 단말기 환경에 적용되기에 알맞다. 본 고에서는 XTR 공개키 알고리즘의 개념과 수학적인 배경에 대해 연구하고, 현재까지 제안된 XTR 공개키 암호 알고리즘의 효율성을 분석하여 앞으로 무선 환경에서의 응용분야를 논한다.

Ⅰ. 서 론

Lenstra가 2000년 제안한 XTR(Efficient Compact Subgroup Trace Representation)은 ElGamal과 같은 이산대수 문제에 기반한 암호시스템이다[1]. XTR은 서브 그룹을 정의하고, 특정 성질을 만족하는 원소를 이용해서 효율적인 암호연산을 수행한다. XTR 서브 그룹은 $GF(p^6)^*$ 상에서 위수가 p^2-p+1의 약수인 원소들의 집합이다. 이러한 원소를 사용하여 암호연산을 수행할 경우 이 원소들의 켤러원소(conjugates)를 이용해서 그룹 $GF(p^2)^*$상으로 트레이스(trace)가 가능하기 때문에 메모리와 암호연산 수행시간을 모두 줄여줄 수 있다.

XTR의 키 길이는 RSA 1024비트와 같은 안전성을 제공하기 위해서 170비트를 사용하며, 이는 ECC (Elliptic Curve Cryptography)의 160비트와 거의 유사하다. 현재 무선 환경에서 빠르고 안전한 연산을 위해 ECC를 사용하고 있으며, XTR 또한 ECC와 같은 효율성과 안전성을 갖는 다면 무선 환경에 효과적으로 활용할 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 XTR의 배경이론에 대해서 설명하며, 3장에서는 기존에 제안된 XTR 공개키 시스템의 효율성에 대하여 비교·분석하고 4장에서는, 기존의 암호시스템과의 효율성을 분석한다. 마지막으로 결론에서 그 응용분야를 제시한다.

Ⅱ. 배경 이론

1. 트레이스

XTR 서브 그룹의 원소를 $GF(p^2)$상의 원소로 표현하는 트레이스(trace)는 $GF(p^6)$상의 원소를 $GF(p^2)$상의 원소의 대응을 말한다.

[정의 1. (켤레원소)]

$GF(p)$상의 원소 β의 켤레원소는 $GF(p)$상의 원소 β의 최소 다항식(minimal polynomial)을 0으로 만드는 값이다.

이를 XTR 그룹에 적용시키면, $GF(p^6)$상의 최소 다항식을 0으로 만드는 원소 h의 켤레원소는 h, h^{p^2}과 h^{p^4}이다. XTR은 표수로 $p\equiv 2 \bmod 3$을 사용하기 때문에 h^{p^3}과 h^{p^5}은 모두 h와 같아지므로 이 값들 또한 $GF(p^6)$상의 최소 다항식을 0으로 만족시킨다.

* 성균관대학교 전기전자 및 컴퓨터 공학부 정보통신 보호연구실({kokim, jkwak}@dosan.skku.ac.kr)
** 한국정보보호진흥원(sjkim@kisa.or.kr)
*** 성균관대학교 정보통신공학부 정교수(dhwon@dosan.skku.ac.kr)

〔정의 2. (트레이스)〕

$GF(p^6)$상의 원소의 켤레원소들을 모두 더해주면 $GF(p^2)$상의 원소로의 트레이스이다.

$GF(p^6)$상의 최소 다항식을 0으로 만족시키는 원소 h의 트레이스는 다음과 같이 표현한다.

$$Tr(h) = h + h^{p^2} + h^{p^4}$$

이러한 트레이스는 선형연산(linear operation)의 성질을 만족시킨다.

실질적으로 암호 프로토콜에 사용되기 위해서 $GF(p^6)$상의 원소를 h대신 원시원소인 g를 선택하면, g의 켤레원소는 g, g^{p^2}와 g^{p^4}이다. 표수 p는 $p=2 \bmod 3$이기 때문에, g의 켤레원소는 g, g^{p-1}과 g^{-p}이다. 이 원소를 $GF(p^2)$상의 원소로 트레이스하면 $Tr(g) = g + g^{p-1} + g^{-p}$이 된다.

$GF(p^6)$상의 원소 g가 주어졌을 때 g^n을 구하는 것 보다, $GF(p^2)$상의 원소 $Tr(g)$가 주어졌을 때 $Tr(g^n)$을 계산하는 것이 계산상 더 빠르고 적은 메모리를 요구하기 때문에 XTR 서브 그룹의 트레이스를 이용한 연산은 효과적이라 할 수 있다.

2. 파라미터 생성

XTR 공개키 알고리즘은 다른 공개키 알고리즘에 비해 파라미터를 선택하는데 원시원소를 구하거나 하는 과정이 필요 없기 때문에 상대적으로 적은 시간이 걸린다는 장점이 있다. 실질적으로 XTR 공개키 알고리즘이 타원곡선 암호 알고리즘과 RSA 공개키 알고리즘에 비해서 빠르게 파라미터를 선택함을 보여주고 있다.[1]

파라미터 $Tr(g)$는 다음의 과정을 거쳐서 구해진다. 우선 $GF(p^6)$상의 원소 중 위수가 p^2-p+1의 약수이면서 소수인 XTR 서브 그룹 내의 원소 g를 찾는다. 다음으로 원소 g의 켤레원소를 구한 후에 이를 모두 더해줌으로써, $GF(p^2)$상의 원소를 구하게 된다.

하지만, XTR 공개키 알고리즘에서는 파라미터 $Tr(g)$를 구하기 위해서 위의 과정을 거치지 않고, 원시원소 g 없이도 $Tr(g)$를 구할 수 있기 때문에 파라미터를 선택하는데 매우 효과적이다.

XTR 공개키 암호 알고리즘에서 파라미터를 선택하는 방법은 크게 두 가지 과정으로 이루어진다. 먼저 〔알고리즘 3〕의 방법을 이용해서 기약 다항식 여부를 판별하는 과정과 〔알고리즘 4〕의 과정을 이용해서 실제 파라미터인 $Tr(g)$를 구하는 과정이다.

〔알고리즘 3. (다항식의 기약 여부 판별)〕

다항식 $F(c,X) \in GF(P^2)[X]$의 기약여부를 판별하기 위해서 다음의 과정을 따른다.

1. $f_0 = \dfrac{-27 + 9c^{p+1} - 2c^3}{27}$과
   $f_1 = c^p - \dfrac{c^2}{3} \in GF(p^2)$을 계산한다.
2. 만약 $\triangle = f_0^2 + 4(\dfrac{f_1}{3})^3 \in GF(p)$가 $GF(p)$에 서 이차비잉여이면, 다항식 $F(c,X)$는 가약(reducible)이다.
3. $\triangle$가 이차잉여이면,
   $r_1 = -f_0 + \sqrt{\triangle} \in GF(p^2)$을 계산한다.
4. $y = r_1^{(p+1)/3} \in GF(p^2)$을 계산하고, 만약 $y \neq y^p$이면 다항식 $F(c,X)$는 기약이다.

〔알고리즘 4. (파라미터 $Tr(g)$구하기)〕

1. 랜덤 수 $c \in GF(p^2)/GF(p)$를 선택하여, 위의 〔알고리즘 3〕을 이용해서 다항식 $F(c,X)$의 기약 여부를 판별한다.
2. 다항식 $F(c,X)$가 가약일 경우 위의 1단계로 다시 돌아간다.
3. 다항식 $F(c,X)$가 기약일 경우
   $d = c_{(p^2-p+1)/q}$을 계산한다.
4. 만약 $d=3$이면 위의 1단계로 돌아간다.
5. 만약 $d \neq 3$이면 $Tr(g) = d$가 된다.

Lenstra가 제시한 알고리즘은 기약다항식 판별을 거쳐 생성한 파라미터가 $c_{p2-p+1/q} \neq 3$을 만족하는지 여부를 판별하는 것이다. 만약 만족한다면 $c_{p2-p+1/q}$는 XTR의 파라미터가 되는 것이다. 즉, $Tr(g) = c_{p2-p+1/q}$이 된다.

그렇기 때문에 XTR 공개키 알고리즘에서 계산의 효율성은 기약다항식을 테스트하는 과정의 연산량과 실제로 XTR 파라미터를 구하는 과정의 연산량에 의해 좌우된다. 이 과정에서 알 수 있듯이 원시원소 g 없이도 $Tr(g)$를 구할 수 있으며, 특별히 g를 구하고자 할 경우에는 삼차다항식 $F(Tr(g),X)$의 근을 구하면 된다.

3. 암호연산

암호 프로토콜에서 사용되는 암호연산은 $S_n(Tr(g)) = (Tr(g^{n-1}), Tr(g^n), Tr(g^{n+1}))$과

$Tr(g^a * g^{bk})$이 있다.

$S_n(Tr(g)) = (Tr(g^{n-1}), Tr(g^n), Tr(g^{n+1}))$ 연산은 주로 통신하는 주체간의 세션키를 생성하기 위해 사용되는 연산이며 [알고리즘 5]를 이용해서 계산한다

[알고리즘 5. (n과 c가 주어졌을 때, $S_n(c)$ 구하기)]

1. 만약 $n<0$이면, $c_{-n} = c_{np} = c^p{}_n$을 이용해서 구할 수 있다.
2. 만약 $n=0$이면, $S_0(c) = (c^p, 3, c)$이다.
3. 만약 $n=1$이면, $S_1(c) = (3, c, c^2 - 2c^p)$이다.
4. 만약 $n=2$이면,
   $c_{n+2} = c \cdot c_{n+1} - c^p \cdot c_n + c_{n-1}$과 c_3를 계산하기 위해서 $S_1(c)$를 이용해서, $S_2(c)$를 구할 수 있다.
5. $n > 2$인 경우, $S_n(c)$를 계산하기 위해서 $S_i(c)' = S_{2i+1}(c)$로 정의하고 $m' = n$으로 놓는다.
   만약 m'가 짝수인 경우, $m' = m' - 1$로놓는다.
   $m' = 2m+1$과 $k=1$이라 놓고 $c_{n+2} = c \cdot c_{n+1} - c^p \cdot c_n + c_{n-1}$과 $S_2(c)$을 이용해서 $S_k(c)' = S_3(c)$을 계산한다.
   $m = \sum_{j=0}^{r} m_j 2^j$이라 놓는다. (단, $m_j \in \{0,1\}$이고, $m_r = 1$) $j = r-1, r-2, \ldots, 0$에 대해서 다음의 과정을 수행한다.
   - 만약 $m_j = 0$이면
     $S_{2k}(c)' = (c_{4k}, c_{4k+1}, c_{4k+2})$를 계산하기 위해서 $S_k(c)' = (c_{2k}, c_{2k+1}, c_{2k+2})$을 이용한다.
   - 만약 $m_j = 1$이면
     $S_{2k+1}(c)' = (c_{4k+2}, c_{4k+3}, c_{4k+4})$를 계산하기 위해서 $S_k(c)' = (c_{2k}, c_{2k+1}, c_{2k+2})$을 이용한다.
   - k 대신 $2k + m_j$로 대체한다.
6. $k = m$과 $S_{m'}(c) = S_m(c)'$을 반복한다. 만약 n이 짝수이면, $S_{m'+1}(c) = (c_{m'}, c_{m'+1}, c_{m'+2})$를 구하기 위해서 $S_{m'} = (c_{m'-1}, c_{m'}, c_{m'+1})$을 이용하고, m'를 $m'+1$로 대체한다. 그 결과 $S_n(c) = S_{m'}(c)$이다.

$Tr(g^a \cdot g^{bk})$ 연산은 XTR- NR 서명 방식에서는 메시지를 복원하기 위해 사용하며, [알고리즘 6]을 이용해서 계산한다.

[알고리즘 6 ($Tr(g^a \cdot g^{bk})$ 구하기)]

$Tr(g)$, $S_k(Tr(g))$와 $a, b \in Z$(단, $0 < a, b < q$)가 주어졌을 때, $Tr(g^a \cdot g^{bk})$구하기

1. $e = a/b \bmod q$를 구한다.
2. [알고리즘 4]에 $n=e$와 $c = Tr(g)$를 적용시켜서 $S_e(Tr(g))$를 계산한다.
3. $Tr(g)$와 $S_e(Tr(g))$를 이용해서 $C(A(Tr(g)^e)$를 구한다.(자세한 정의[4])
4. $Tr(g^{e+k}) = S_k(Tr(g)) \cdot C(A(Tr(g)^e)$를 계산한다.
5. [알고리즘 4]를 이용해서 $n=b$와 $c = Tr(g^{e+k})$일 경우 $S_b(Tr(g^{e+k}))$를 계산한다.
6. $Tr(g^{(e+k)b}) = Tr(g^a \cdot g^{bk})$

Ⅲ. XTR 암호 시스템의 효율성 분석

1. XTR 공개키 시스템(이하 LV[1])

XTR 공개키 알고리즘은 Lenstra와 Verheul에 의해 처음 소개되었다. 제안된 XTR 공개키 알고리즘은 같은 안전성을 제공하는 다른 공개키 알고리즘에 비해 작은 키 를 사용하며, 빠른 암호연산을 수행한다. 특히 파라미터 선택에 있어서 다른 공개키 알고리즘에 비해 매우 효과적이다. XTR 공개키 알고리즘에서는 RSA 1020비트와 같은 안전성을 제공하기 위해 170비트를 사용하며, 이는 현재 표준화되어 무선 환경에서 사용되고 있는 타원곡선 암호 알고리즘과 같은 크기이다.

Lenstra에 의해 처음으로 XTR 공개키 시스템이 소개된 이후 파라미터 선택의 장점을 개발해서 적은 메모리와 빠른 암호연산을 요구하는 무선 단말기에 적용되기에 적합한 방법들이 제시되고 있다.

2. 파라미터 선택의 효율성(이하 LV[2])

Asiacrypt 2000에서 LV[2]는 XTR 공개키 알고리즘의 파라미터 선택의 효율성을 좀 더 높이고자 하였다. XTR 공개키 알고리즘을 여러 응용분야에 적용시키기 위해서 표수를 $p \neq 8 \bmod 9$로 선택해서 다항식의 기약다항식 여부를 판별하는 단계에서 Sipione del Ferro의 3차식의 근을 찾는 알고리즘을 사용했다. 이 알고리즘을 이용해서 주어진 다항식의 기약 여부를 판별하고 한다. 이렇게 함으로써 서명에 사용되는 XTR 공개키의 크기를 줄여 주고, 파라미

터를 효과적으로 선택하는 방법을 제시함으로써 초기의 $LV^{[1]}$ 보다 효율성을 증가시켰다.

이러한 알고리즘을 이용하면 다항식의 기약다항식 여부를 판별하는데 곱셈연산을 $2.4\log_2(p)$번 수행하기 때문에 기존의 방식보다 약 70% 정도의 연산상의 효율성이 좋아진다. 기존의 방식과 비교할 때 기약다항식 테스트시간과 기약다항식을 구하는 곱셈연산을 비교하면 다음 〔표 1〕와 같다.

〔표 1〕 기약다항식 테스트와 기약다항식 선택의 연산량(곱셈연산량) 비교

	기약다항식 테스트	기약다항식 구하기
$LV^{[1]}$	$8\log_2(p)$	$24\log_2(p)$
$LV^{[2]}$	$2.4\log_2(p)$	$7.2\log_2(p)$
연산 감소량	70%	70%

기약다항식 여부를 판별하는데 있어 $LV^{[1]}$의 경우에서는 결과값인 트레이스에 관계없이 $8\log_2(p)$의 곱셈연산이 필요하지만, $LV^{[2]}$에서는 트레이스의 반에 대해서는 아무런 연산 없이 다항식의 기약 여부를 판별할 수 있으며, 나머지 반에 대해서는 $4.8\log_2(p)$의 연산이 필요하다. 그렇기 때문에 Lenstra[2]의 경우 다항식의 기약 여부를 판별하기 위해서 평균적으로 $2.4\log_2(p)$의 연산이 필요하게 된다. 이 결과는 $LV^{[1]}$에서보다 약 3배 정도의 연산량을 줄여준다.

다음으로 기약다항식 테스트를 거쳐 기약다항식을 구한 후에 실질적인 파라미터를 구하는 과정이다. 이 과정은 표수 p에 제약을 주어 효과적으로 계산할 수 있다.

XTR 공개키 알고리즘은 표수 $p\equiv 2 \bmod 3$을 사용한다. 이에 알맞은 $GF(p)$상의 기약다항식은 $(X^3-1)/X-1=X^2+X+1\in GF(p)[X]$이된다.

$GF(p^2)$상의 옵티멀 노멀 베이시스 $\alpha^2+\alpha+1=0$을 이용하면 확대체 상의 원소에 대한 p승 연산은 아무런 곱셈연산을 수행하지 않고 자리만 바꿔주어 계산이 가능하다.

$LV^{[2]}$에서는 이러한 XTR 공개키 알고리즘의 의 표수에 약간의 제약을 가해서 $p\equiv 2 \bmod 9$이거나 $p=5 \bmod 9$인 표수를 선택한다. 즉, $p\neq 8 \bmod 9$인 표수를 선택한다. 이러한 표수 p는 $GF(p)$상에서 기약다항식 $(Z^9-1)/(Z^3-1)=Z^6+Z^3+1\in GF(p)[Z]$을 갖는다. 이 다항식을 t^{th} 싸이클로토믹(cyclotomic) 다항식이라 부르며 $\phi_t(Z)$로 나타낸다. 만약 $GF(t)^*$가 싸이클릭(cyclic)이고 $p \bmod t$에 의해서 생성된다면, $\phi_t(Z)$는 기약다항식이다. 여기서 곱셈 그룹 $GF(t)^*$가 싸이클릭을 만족하기 위해서는 $t=2$, 4이거나, t가 홀수인 소수의 지수승이거나, 홀수인 소수의 두 배 혹은 네 배의 지수승이면 된다. 이러한 특정한 표수인 $p\neq 8 \bmod 9$를 이용해서 XTR 그룹의 생성원에 대한 파라미터를 구하는 알고리즘을 $LV^{[2]}$에서 이용하고 있다.

Lenstra[21]의 알고리즘을 이용해서 파라미터를 구하는 연산은 〔$8\log_2((p^2-p+1)/q)+C$(상수)〕 만큼의 곱셈연산을 수행하기 때문에 $LV^{[1]}$의 알고리즘에 비해 약 두 배 정도 연산속도가 향상되었다. 하지만, 표수 $p\neq 8 \bmod 9$의 제약이 있다는 단점이 있다.

3. 파라미터 선택의 개선된 효율성(이하 $LV^{[3]}$)

2001 PKC에 제안된 방식[3]은 XTR 공개키 알고리즘의 XTR 그룹의 생성원의 트레이스인 파라미터를 구하는 새로운 방법이다. 이 방법은 $LV^{[2]}$보다 효과적으로 파라미터를 구할 수 있으며, 효과적인 서브 그룹 맴버쉽(*subgroup-membership*) 테스트 방법을 제시했다.

$LV^{[3]}$에서의 기약다항식 판별을 위해서 $LV^{[2]}$에서 사용했던 Scipione del Ferro의 방법을 이용한다. 하지만, $LV^{[2]}$와 다르게 직접적으로 $F(c,X)\in GF(p^2)[X]$을 대입해서 다항식의 기약 테스트를 수행하지 않고, 계산의 효율성을 위해서 새로운 다항식 $p(x)$를 정의한다. 삼차다항식 $P(c,X)\in GF(p)[X]$의 기약 여부를 판별함으로써, 원래 다항식 $F(c,X)$의 기약 여부를 판별할 수 있다. 이 방법은 $GF(p^2)$상의 다항식 $F(c,X)$에서 $GF(p)$상의 다항식 $P(c,X)$을 구하는데 드는 연산량은 매우 적고(상수시간) 다항식 $P(c,X)$의 기약 여부를 판별하는 연산량 또한 $LV^{[2]}$보다 상대적으로 적기 때문이다. 다항식 $P(c,X)$의 기약 여부를 판별하는 연산은 평균적으로 $0.9\log_2(p)$의 곱셈연산이 수행되기 때문에 이 방식을 이용하면 다항식 $F(c,X)$의 기약다항식 여부를 판별하는데 〔$0.9\log_2(p)$ +작은 상수〕의 곱셈연산이 필요하게 된다.

이 방법 또한 $LV^{[2]}$의 방법과 마찬가지로 트레이스의 반에 대해서는 아무런 연산 없이 다항식의 기약 여부를 판별할 수 있으며, 나머지 반에 대해서는 $1.8\log_2(p)$의 연산이 필요하다. 그렇기 때문에 다항식의 기약 여부를 판별하기 위해서 평균적으로 $0.9\log_2(p)$의 연산이 필요하게 된다. 이는 기존의 방법에 비해

약 60%의 계산상의 효율을 얻을 수 있다.

또한 이러한 방법을 이용해서 다항식의 기약 여부를 판별한 후에 실질적인 기약다항식을 구하기 위해서 $2.7\log_2(p)$번의 곱셈연산이 필요하다. 따라서 XTR 공개키 알고리즘의 파라미터를 구하기 위해 $\frac{q}{q-1}(2.7\log_2(p)+8\log_2((p^2-p+1)/q))$번의 곱셈연산이 필요하다.

결과적으로 실질적인 파라미터를 구하는데 드는 연산량은 LV[2]와 많은 차이가 있지는 않지만, LV[2]의 표수에 대한 제약사항을 제거했기 때문에 $p\equiv 2 \bmod 3$뿐만 아니라, $p\equiv 2,5 \bmod 9$에서도 적용 가능하다. 이는 기존의 방식들보다 좀 더 일반적으로 XTR 공개키 알고리즘을 적용시킬 수 있다는 장점이 있다.

LV[3]의 알고리즘을 이용해서 다항식의 기약다항식 여부를 판별하고 실질적으로 테스트를 통해서 기약다항식을 구하는 연산량(곱셈연산회수)을 비교하면 다음 [표 2]와 같다.

[표 2] 기약다항식 테스트와 기약다항식 선택의 연산량(곱셈연산량) 비교

	기약다항식 테스트	기약다항식 구하기
LV[2]	$2.4\log_2(p)$	$7.2\log_2(p)$
LV[3]	$0.9\log_2(p)$	$2.7\log_2(p)$
연산 감소량	62%	62%

결과적으로 LV[2]보다 LV[3]의 알고리즘은 약 60% 정도의 연산 감소 효과를 얻을 수 있다.

4. 연산방법 개선을 이용한 연산 효율성 증가(이하 LM[4])

기존의 XTR 방식에서는 파라미터 선택의 효율성을 증가시킴으로써 XTR 공개키 알고리즘의 효율성을 증가시키고자 하였다.

하지만 Asiacrypt 2001에 발표된 XTR 공개키 암호 시스템[5]은 지수승 연산 방법을 개선함으로써 XTR 공개키 알고리즘의 효율성을 높이고자 하였다. 기존 방식에서 사용되던 지수연산에 비해 제안한[5] 지수연산 방식은 더블 지수연산(double exponentiation)과 싱글 지수연산(single exponentiation) 모두에서 50%의 연산 효율성을 가진다. 더블 지수연산은 XTR 서명을 사용하는 응용분야에서 매우 효과적으로 이용될 수있으며, 싱글 지수연산은 기존의 방식[1]에서의 연산에 부담을 주는 행렬 연산을 없애주므로, 연산의 효율성 뿐만 아니라 구현상의 용의성 또한 제공한다. 하지만, 이 방식의 경우 한번의 사전계산(precomputation)이 필요하다는 단점이 있다. 본 논문에서는 이러한 단점을 보완하기 위해서 또 다른 싱글 지수연산방식을 제안하였는데, 이 방식은 기존의 방식에 비해 15%의 연산상의 효율성을 보이지만, 사전계산 과정이 없다는 장점이 있다.

기존의 논문에서는 파라미터 선택을 효율적으로 수행함으로써 알고리즘의 효율성을 높이고자 하였다면, LM[4]에서는 지수연산 자체의 효율성을 높임으로 알고리즘의 효율성을 높이고자 하였다. 다음 [표 3]에서는 LV[1]의 지수연산과 LM[4]의 지수연산을 통한 XTR의 수행시간을 비교한다.

[표 3] LV[1]과 LM[4]의 연산속도 비교

	LV[1]	LM[4]		LM[4]/LV[1]	
		no-precom.	precom.	no-precom.	precom.
암호화	21ms	16ms	10ms	0.7	0.47
복호화	10ms	8ms		0.8	
서명생성	10ms	8ms	5ms	0.8	0.5
서명검증	21ms	10ms		0.47	

[표 3]은 NT의 600MHz PentiumⅢ에서 수행된 결과이다.

[표 3]에서 알 수 있듯이 LM[4]의 지수연산을 사용할 경우 메시지의 암·복호화와 서명의 생성검증에 드는 연산 속도가 평균 50%정도 향상된다.

5. $GF(p^{6m})$상의 확장 XTR 공개키 알고리즘 (이하 LK[5])

2001년 SAC에서는 확장 XTR 공개키 암호 알고리즘에 대해서 제안하였다.[6] 확장 XTR 공개키 시스템이란 XTR 공개키 시스템은 확대체 $GF(p^6)$상에서 정의되었는데 이를 확대체 $GF(p^{6m})$상으로 확장해서 정의한 것이다. 확대체 $GF(p^{6m})$에서 적당한 파라미터 p와 m을 선택함으로써 컴퓨터의 시스템적인 관점에서 효율적인 연산을 수행할 수 있게 한다.

확대체에서 표수 p는 컴퓨터 시스템상의 마이크로프로세서의 처리 능력과 밀접한 관련이 있다. 현재 우리가 사용하고 있는 컴퓨터는 마이크로프로세서가 한번에 처리할 수 있는 데이터의 기본 단위로 워드(word)를 사용하는데, 표수 p와 워드의 크기에 따라 컴퓨터의 연산 속도와 연산량이 결정된다. 예를들면

표수 p가 마이크로프로세서가 처리할 수 있는 워드보다 클 경우 계산 과정에서 워드 사이의 캐리(carries)를 다루어야 하며, 피 연산자를 법 p에 관해서 리덕션(reduction)해야 하는 등의 문제가 발생한다.

이러한 문제점을 해결해서 시스템 측면에서 통신량을 줄이고 계산상의 효율성을 높이고자 마이크로프로세서의 워드보다 작은 크기의 p값을 사용한다. 또한 이 경우 발생할 수 있는 보안상의 문제를 해결하기 위해 파라미터 m값을 선택할 때, 몇가지 제약사항을 두었다. 확장 XTR 공개키 알고리즘에서 사용하는 m은 $2m+1$이 페르마의 소수이거나 m과 $2m+1$이 모두 소수인 m을 사용한다.

하지만 확장 XTR 공개키 알고리즘에서 위의 조건을 만족하는 m값을 사용하면, m을 크게 잡아주어 상대적으로 소수 p, q가 작아서 생길 수 있는 안전성 문제를 해결할 수 있다. 위의 조건을 만족하는 파라미터를 선택하는데 많은 시간이 소요되긴 하지만, 소수 p, q를 선택하는 과정은 암호통신이 이루어지기전에 단 한번만 수행되기 때문에 대부분의 시스템에서는 심각한 문제가 되지 않는다.

p와 $2m+1$이 소수이고, p가 법 $2m+1$상에서 원시원소라 할 때, $GF(p^{2m})$상의 원소 x, y, z에 대해서 확장 XTR 공개키 알고리즘에서의 연산은 x^{b^m}은 단 한번의 곱셈연산 없이도 가능하며, x^2은 $GF(p^{2m})$상에서 기존의 방법보다 80%의 연산 효율을 얻을 수 있다.

실제 암호연산인 $S_n(c)=(c_{n-1}, c_n, c_{n+1})$은 $GF(p)$상에서 최대 $11.2m^2\log n$번의 곱셈연산이 수행되며, XTR-N-R 서명 방식의 $Tr(g^a * g^{bk})$연산은 $(11.2\log(a/b(\bmod q))+11.2\log b+36)m^2$번의 곱셈연산을 수행한다. 연산량을 비교해본 결과 LV[1]에서 제시한 초기의 XTR과 연산량에서는 차이가 없지만 표수 p의 크기를 마이크로프로세서의 워드 크기에 맞춰 줌으로써 다정도(multiprecision) 문제를 해결했다.

다음에서는 XTR 공개키 암호방식과 기존에 소개된 다른 공개키 암호 방식과의 효율성을 비교함으로써 XTR 공개키 암호 시스템이 무선 통신과 스마트카드와 같은 메모리와 연산량에 제약사항이 많은 응용분야에 적합한 공개키 암호 시스템임을 증명하고자 한다.

Ⅳ. 기존 암호시스템과의 효율성 비교 · 분석

현재 무선 통신의 단말기에서 사용하고 있는 암호 알고리즘은 기존의 RSA 암호방식에 연산상의 효율을 높이기 위해서 중국인의 잉여정리를 적용한 고속 RSA 암호방식과 타원곡선의 연산의 효율성을 이용한 ECC 암호방식이 주로 쓰이고 있다.

고속 RSA 암호방식은 서명 검증 연산이 타 암호 시스템에 비해서 월등히 빠르다는 장점을 가지고 있다. 또한 ECC 암호방식은 RSA 1024 비트와 같은 안전성을 제공하기 위해서 170비트만이 필요하기 때문에 상대적으로 작은 키를 사용해서 연산을 수행해서 연산량이 RSA 암호방식보다 매우 적다.

일반적인 시스템에서 RSA 1020비트와 같은 안전성을 제공하기 위해서 XTR 공개키 암호 시스템은 170비트를 사용한다. 이는 ECC와 같은 크기이다.

1. RSA vs. XTR

이 프로그램은 XTR의 파라미터 생성과 암·복호화와 서명의 생성과 검증 과정의 모든 연산에 관해서 수행한다. 하지만, 프로그램을 간단히 하기 위해서 대칭키 암호 방식과 해쉬 알고리즘은 실제로 사용되는 경우보다 매우 간단한 것을 사용하였다.

RSA와의 효율성 분석은 NT의 펜티엄 Ⅲ 800MHz에서 수행한 결과이다.

키 생성시간은 50번의 키 생성과정을 통해서 평균시간을 의미하며, 연산시간은 각 키 마다 열 번의 랜덤한 실험을 통한 평균 시간을 의미한다.

〔표 4〕 RSA와 XTR의 연산속도 비교 · 분석

	RSA	XTR	XTR/RSA
크기	1020bit	170bit	0.16배
파라미터와 키 생성	1102ms	62ms	0.06배
암호화	7ms	25ms	3.6배
복호화	51ms	15ms	0.29배

위의 〔표 4〕에서 키의 크기는 RSA는 법 연산의 크기를 의미하며, XTR은 서브 그룹의 크기를 의미한다. 〔표 4〕에서 RSA와 XTR의 연산속도를 비교한 결과, 크기에서는 RSA에 비해서 XTR이 0.16배로 상대적으로 매우 작다. 파라미터와 키 선택과 복호화(서명생성)에 있어서도 RSA에 비해 매우 빠르게 연산을 수행한다. 하지만 RSA의 가장 큰 장점인 메시지의 암호화(서명검증)시간은 여전히 RSA가 훨씬 빠르게 수행한다.

2. ECC Vs. XTR

다음으로 XTR과 ECC의 연산 속도를 비교한다.

〔표 5〕 ECC와 XTR의 연산속도 비교분석

	160비트	256비트
ECC	9.6 ms	25.2ms
XTR	11.2ms	16.7ms
XTR/ECC	1.2배	0.66배

두 알고리즘 모두 160비트 일 때, XTR의 지수 연산과 ECC의 커브 포인트에 관한 연산의 시간을 비교한다. ECC와의 효율성 분석은 NT의 펜티엄 Ⅲ 800MHz에서 수행한 결과이다.

〔표 5〕의 ECC와 XTR의 연산속도 비교·분석 결과 현재 무선 환경에서 사용되고 있는 가장 효율적인 ECC에 비해 160비트 일때는 1.2배정도로 느리지만, 좀 더 큰 비트인 256비트에서는 ECC의 0.66배로 매우 효과적이라고 할 수 있다.

위의 결과는 LV[1]을 바탕으로 구현되었기 때문에 LV[3]과 LM[4]의 알고리즘을 적용한다면, 좀 더 좋은 결과를 얻을 수 있을 것으로 기대된다.

Ⅴ. 결 론

최근들어 무선 환경의 제약사항을 고려해서 적용가능한 알고리즘들이 소개되고 있다. 그 중 XTR 알고리즘은 현재 무선 환경에서 사용되고 있는 타원곡선 암호 알고리즘 만큼의 작은 키(170비트)를 사용해서 암호연산을 하는 알고리즘이다. 특히 이 방식은 타원곡선 암호 방식만큼 암호연산 속도가 빠를 뿐만 아니라, 암호연산시 사용하는 메모리도 효율적이기 때문에 차세대 무선 통신에 알맞은 알고리즘이라고 하겠다.

XTR은 2000년 처음 제안된 이후 현재까지 표준화는 진행 중에 있으며, P 1363 표준안에 포함되지 않은 다른 알고리즘과 프로토콜들과 함께 앞으로 P 1363b에 포함시킬 주제로 논의 되고 있다.

본 고에서는 현재까지 발표된 XTR 관련 논문들의 효율성을 분석하고, 기존의 다른 암호 알고리즘과의 효율성을 비교·분석함으로 무선 환경의 요구사항을 만족함을 분석하였다.

좀 더 효율적인 알고리즘에 대한 연구가 진행된다면, XTR 공개키 암호 알고리즘은 계산의 효율성과 메모리의 제약이 있는 현재 타원곡선 암호 시스템과 RSA 알고리즘이 사용되고 있는 많은 분야에서 사용될 것이다. 특히 스마트카드와 같은 보안 모듈내에서 사용시 그 이용 가치가 높을 것으로 생각된다.

참 고 문 헌

〔1〕 Arjen K. Lenstra, Eric R. Verheul, "The XTR public key systems", *crypto 2000*, '2000.

〔2〕 Arjen K. Lenstra, Eric R. Verheul, "Key improvements to XTR", *Asiacrypt 2000*, '2000.

〔3〕 Arjen K. Lenstra, Eric R. Verheul, "Fast irrecucibility and subgroup membership testing in XTR", *PKC 2001*, '2001.

〔4〕 Arjen K. Lenstra, Eric R. Verheul, "An overview of the XTR public key system", *PKC & CNT*, '2001.

〔5〕 Martijn Stam, Arjen K.Lenstra, "Speeding up XTR", *Asiacrypt 2001*, '2001.

〔6〕 Seongan Lim, Seungjoo Kim, Ikkwon Yie, Jaemoon Kim, and Hongsub Lee, "XTR Extended to $GF(p^{6m})$", *SAC 2001*, '2001.

〔7〕 Wieb Bosma, James Hutton, and Eric, R. Verheul, "Looking beyond XTR", *Asiacrypt 2002*, '2002.

〔8〕 Peter J. Smith and Michal J. J. Lennon, "A New Public Key System", *Proceedings of the Ninth IFIP International Symposium on Computer Security '93, pp.97~111*, Elsevier Science Publications, 1994.

〈著 者 紹 介〉

김 근 옥 (Keun-Ok Kim)
학생회원

2002년 2월 : 서울여자대학교 수학과 졸업(이학사)
2002년 3월~현재 : 성균관대학교 정보통신공학부 석사 과정

곽 진 (Jin Kwak) 학생회원

2000년 8월 : 성균관대학교 생물기전공학과 졸업(공학사)
2001년 3월~2003년 2월 : 성균관대학교 대학원 전기전자 및 컴퓨터 공학부 졸업 (공학석사)
2003년 2월~현재 : 성균관대학교 대학원 전기전자 및 컴퓨터 공학부 박사 과정

김 승 주 (Seung-Joo Kim)
종신회원

1994년 2월 : 성균관대학교 정보공학과 졸업(공학사)
1996년 2월 : 성균관대학교 대학원 정보공학과 공학석사 (암호학 전공)
1999년 2월~성균관대학교 대학원 정보공학과 공학박사 (암호학 전공)
1998년 12월~현재 : 정보보호진흥원(KISA) 암호기술팀장
2000년 6월~현재 : 한국정보통신기술협회(TTA) 정보통신기술위원회 암호기술연구반 의장
2002년 4월~현재 : 한국정보통신기술협회 국제 표준화 전문가

원 동 호 (Dongho Won)
종신회원

성균관대학교 전자공학과 졸업 (학사, 석사, 박사)
1978년~1980년 : 한국전자통신연구원 전임연구원
1985년~1986년 : 일본 동경공업대 객원연구원
1988년~1999년 : 성균관대학교 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학 원장, 정보통신기술연구소장
1996년~1998년 : 국무총리실 정보화추진위원회 자문위원
2002년~2003년 : 한국정보보호학회장
현재 : 성균관대학교 정보통신공학부 교수, 정통부 지정 정보보호 인증기술 연구센터장, 성균관대학교 연구처장