韓國 컴퓨터情報學會 論文誌
第9卷 第2號, 2004. 6.

2004-9-2-3-4

유한체의 부분군을 이용한 효율적인 사용자 인증 프로토콜 설계

정 경 숙*

Design of an Efficient User Authentication Protocol Using subgroup of Galois Field

Kyoungsook Jung *

요 약

본 논문에서는 최근 증가되고 있는 사이버스페이스 상에서의 다양한 전자상거래를 효율적이고 안전하게 거래하기 위하여 유한체 부분군 기반의 효율적인 사용자 인증 프로토콜을 설계하였다. Lenstra와 Verheul에 의해 제안된 XTR은 짧은 키 길이와 빠른 연산 속도의 장점을 가지고 있기 때문에 복잡한 연산에 유용하게 사용될 수 있다. 이를 기반으로 하여 문제 도메인을 기존의 유한체인 $GF(p^6)$을 사용하지 않고, $GF(p^6)$의 부분군인 $GF(p^2)$을 이용하여 문제를 해결하는 XTR-ElGamal 기반의 사용자 인증 프로토콜을 제안하였다. XTR-ElGamal 기반의 사용자 인증 프로토콜은 유한체 상의 부분군들을 이용함으로써 키 교환 시 요구되는 키의 비트 수를 줄였다. 또한 계산 시 필요한 오버헤드를 줄여 빠른 계산과 실행을 제공하였다. XTR 기반에서 사용자 인증을 하기 위한 과정에서 필요한 인증 프로토콜을 설계하였다. 따라서 통신량과 계산량이 적게 요구되는 환경에서도 유용하게 이용될 수 있도록 하였다.

Abstract

If the protocol has fast operations and short key length, it can be efficient user authentication protocol Lenstra and Verheul proposed XTR. XTR have short key length and fast computing speed. Therefore, this can be used usefully in complex arithmetic. In this paper, to design efficient user authentication protocol we used a subgroup of Galois Field to problem domain. Proposed protocol does not use $GF(p^6)$ that is existent finite field, and uses $GF(p^2)$ that is subgroup and solves problem. XTR-ElGamal based user authentication protocol reduced bit number that is required when exchange key by doing with upside. Also, proposed protocol provided easy calculation and execution by reducing required overhead when calculate. In this paper, we designed authentication protocol that is required to do user authentication.

▸ Keyword : XTR, 사용자 인증, 엘가말, 유한체(XTR, user authentication, ElGamal, Finite Field)

* 가톨릭대학교 컴퓨터정보공학부 강의전담초빙교수

Ⅰ. 서 론

사이버스페이스에서의 다양한 활동들이 증가하면서 가장 중요한 문제로 다루어지는 것이 전자상거래에서의 신분 확인 문제이다. 정당한 사용자의 인증, 즉 신분 확인 기술은 개인 정보 보호 문제와 맞물려 중요한 개인의 정보는 보호하면서 정당한 사용자임을 입증할 수 있는 기술들이 다양하게 제시되고 있다. 또한 최근에는 사용자들이 많아지고, 데이터 통신이 많아지고 있기 때문에 적은 양의 정보를 교환하면서 안전하게 사용자 인증을 할 수 있는 기술들이 요구되고 있다. 또한 무선 인터넷이 발달함에 따라 스마트카드와 같이 전력과 대역폭이 제한된 응용 부분에서 요구되는, 연산의 속도는 빠르고 키 길이를 줄일 수 있는 효율적인 사용자 인증 시스템이 요구되고 있다.

타원 곡선 암호 시스템(ECC)를 제외하고, 대부분의 공개키 시스템은 사이즈가 큰 키를 갖는다. 이러한 성질은 스마트카드와 무선 통신과 같이 전력과 대역폭이 제한된 응용 부분에서 비실용적이다. 많은 암호학자들은 제한된 하드웨어 환경에 적용 가능한 가장 효율적인 공개키 시스템을 ECC라고 생각하였다[1][2][3][4].

본 논문에서는 공유되는 큰 키에 대한 문제점을 해결하고, 좀 더 효율적이면서 빠르게 사용자 인증을 할 수 있는 프로토콜을 설계하기 위한 새로운 방안을 제시하였다. 기본 아이디어는 문제 도메인을 기존의 유한체인 $GF(p^6)$을 명확하게 구성하여 사용하지 않고, $GF(p^6)$의 부분군인 $GF(p^2)$을 이용하여 문제를 해결하는 데에 바탕을 두고 있다[5][6][7].

제안된 사용자 인증 프로토콜은 XTR-ElGamal을 기반으로 하여 사용자 인증을 위한 연산의 속도 향상과 키 길이를 축소시켰다. 구체적인 방안으로, XTR 기반에서 사용자 인증을 하기 위한 과정에서 필요한 $y_i = g^{b \cdot p^{2(i-1)}} \cdot v \bmod q, 1 \le i \le 3$와 인증 프로토콜을 설계하였다. 또한 XTR-ElGamal 기반의 사용자 인증 프로토콜의 안전성은 Diffie-Hellman 기반의 이산 대수에 기반을 두고 있다[8][9][10].

본 논문의 2장에서는 XTR 공개키 시스템에 대하여 알아본다. 3장에서는 XTR-ElGamal 기반의 사용자 인증 프로토콜을 제시하고, 이론적으로 증명하였다. 4장에서는 XTR-ElGamal 기반의 사용자 인증 프로토콜의 안전성과 효율성을 비교, 분석하였으며, 5장에서 결론을 맺는다.

Ⅱ. XTR 공개키 시스템 분석

Crypto 2000에서 소개된 XTR[1]은 Arjen K, Lenstra1과 Eric R.Verheul에 의해서 개발되었다. XTR은 안전이 보장되지 않은 채널들 상의 Diffie-Hellman 키 교환 프로토콜에 기반을 둔다. 1997년 Arjen Lenstra가 유한체의 부분군을 이용한 방법을 제안했다. 이것은 키 교환 시 요구되는 비트 수는 줄었으나 계산상의 어려움이 있었는데, XTR은 위의 방법을 개선해 오버헤드를 줄이고 쉬운 계산과 실행을 제공함으로써 스마트카드와 같은 작은 컴퓨팅 장치에 적당하다.

XTR은 부분군의 원소를 표현하고 계산하는데 표준적인 방법을 사용하지 않으며, $GF(p^6)$을 사용하는 것보다 통신상/계산상의 이점을 갖는다. 또한 XTR은 $GF(p^2)$을 이용하여 $GF(p^6)$의 안전성을 획득하였다. XTR의 공개키는 ECC보다 두 배만큼 사이즈가 크지만, 파라미터를 초기화 하는 데에는 RSA와 ECC의 파라미터 초기화 시간보다 작은 시간이 소요된다. 따라서 XTR은 스마트카드나 무선 통신과 같은 응용 부분에서 RSA나 ECC에 대한 훌륭한 대안이 될 수 있다[11][12][13].

1. Traces

XTR은 유한체의 부분군의 원소를 표현하고 그것의 지수승을 계산하는 데에 Trace를 이용하는 방법이다. XTR에서 공개적으로 이용 가능한 정보는 그 원소가 아니라, 유한체의 그룹에서 어떤 원소의 자취를 표현하는 것이다. XTR의 안전성은 $Tr(g^a)$와 $Tr(g^b)$로부터 $Tr(g^{ab})$을 계산해

1) XTR : Efficient and compact Subgroup Trace Representation(ECSTR)

내기 어렵다는 것에 의존하고, $GF(p^6)$ 의 안전성은 보장된다.

$h \in GF(p^6)$의 $GF(p^2)$위에서의 공액(conjugates)은 h, h^{p^2}, h^{p^4}이다. $h \in GF(p^6)$에 대한 $GF(p^2)$위에서의 Trace $Tr(h)$는 h의 $GF(p^2)$에서의 공액의 합이다. 즉, $Tr(h) = h + h^{p^2} + h^{p^4} \in GF(p^2)$이다.

$c \in GF(p^2)$에 대하여, 다항식 $F(c, X) = X^3 - cX^2 + c^pX - 1 \in GF(p^2)$은 $GF(p^2)$에서 h_0, h_1, h_2을 근으로 갖으며, 정수 n에 대하여 $c_n = h_0^n + h_1^n + h_2^n$ 라고 정의하자. 만약 $F(c, X)$가 $GF(p^2)$위에서 기약 다항식이면, $c_n = Tr(h_0^n)$이다.[1]

2. XTR 시스템의 키 분배 알고리즘

XTR 시스템에서 키 분배를 위해서 [정리1]과 [알고리즘 1]을 사용한다.

[정리 1]

i . $c = c_1$

ii . $c_{-n} = c_{np} = c_n^p$, for $n \in Z$

iii . $c_n \in GF(p^2)$, for $n \in Z$

iv . $c_{u+v} = c_u \cdot c_v - c_v^p \cdot c_{u-v} + c_{u-2v}$, for $u, v \in Z$

v . 모든 h_i에 대하여, $o(h_i) \mid p^2 - p + 1$이고, $h_i > 3$이거나, $h_i \in GF(p^2)$이다.

XTR에서 주어진 $Tr(g)$로부터 $Tr(g^n)$을 계산하는 알고리즘은 이산 대수 문제에 기반을 둔 공개키 시스템에서 g^n을 계산하는 것과 유사하다.[7]

[정의 1]

$S_n(c) = (c_{n-1}, c_n, c_{n+1}) \in GF(p^2)^3$이라고 하자. n, c가 주어졌을때, $S_n(c)$를 계산하는 알고리즘은 다음과 같다.

[알고리즘 1]

① If $n < 0$ then apply to algorithm1 by $-n, c$.

② If $n = 0$ then $S_0(c) = (c^p, 3, c)$.

③ If $n = 1$ then $S_1(c) = (3, c, c^2 - 2c^p)$.

④ If $n = 2$ then we use $S_1(c)$ to calculate c_3.

⑤ Set $\overline{S_i}(c) = S_{2i+1}(c)$ and $\overline{m} = n$, calculate $s_n(c)$.

If $\overline{m}$=2n then $\overline{m} = \overline{m} - 1$

and $\overline{m} = 2m + 1, k = 1$.

And calculate $\overline{S_k}(c) = S_3(c)$ by using $s_2(c)$.

For each $j = r-1, r-2, \cdots, 0$, $m_j \in \{0, 1\}$

and $m_r = 1$, $m = \sum_{j=0}^{r} m_j 2^j$ calculate as follow.

If $m_j = 0$ then $\overline{S_{2k}}(c) = (c_{4k}, c_{4k+1}, c_{4k+2})$.

If $m_j = 1$ then $\overline{S}_{2k+1}(c) = (c_{4k+2}, c_{4k+3}, c_{4k+4})$

If n=2k then $S_{\overline{m}+1}(c) = (c_{\overline{m}}, c_{\overline{m}+1}, c_{\overline{m}+2})$.

[정리 2]

위수가 q인 원소 $g \in GF(p^6)$에 대해, $Tr(g^i) = Tr(g^j)$는 "g^i와 g^j가 $GF(p^2)$에서 공액이다"라는 것과 동치이다[3].

[증명]

(⇒)위수가 q인 원소 $g \in GF(p^6)$에 대해 $F(X) = X^3 - Tr(g^i)X^2 + Tr(g^i)^pX - 1 \in GF(p^2)$는 $GF(p^2)$에서 기약 다항식이며 그것의 근들은 $GF(p^2)$에서의 공액이다. 즉, g^i, g^{ip^2}, g^{ip^4}이 $F(X)$의 근이 된다. $Tr(g^i) = Tr(g^j)$이기 때문에,

$F(X) = X^3 - Tr(g^i)X^2 + Tr(g^i)^pX - 1$

$= X^3 - Tr(g^j)X^2 + Tr(g^j)^pX - 1 \in GF(p^2)$

가 된다. 따라서 g^j는 $F(X)$의 근이 된다. 그러므로 g^i와 g^j가 $GF(p^2)$에서 공액이다.

(⇐)g^i와 g^j가 $GF(p^2)$에서 공액이므로, $g^i = g^j, g^i = g^{jp^2}$또는 $g^i = g^{jp^4}$이 된다.

$h \in GF(p^6)$에 대해, $h^{p^6} = 1$이므로 $Tr(g^i) = Tr(g^j)$을 만족하게 된다.

[정리 3]

p와 q는 $q \mid (p^2-p+1)$을 만족하는 소수라고 하자. 만약 $g \in GF(p^6)$의 위수가 q이면 부분군 g는 $GF(p^6)$의 부분체 $GF(p), GF(G), GF(p^3)$에 속하지 않는다[7].

암호 프로토콜에서 XTR의 응용은 안전성은 감소시키지 않고, 통신량과 계산량을 둘 다 감소시킬 수 있다. XTR은 유한체의 군을 모두 사용하지 않고, 부분군을 이용한 이산 대수 문제에 의존하는 암호 시스템에 적용될 수 있다.

2.3 XTR-DH 키 분배 알고리즘

사용자 인증을 하거나 암호화를 위해서는 서로 공유키를 가지고 있어야 한다. XTR 기반에서는 DH 키 분배 알고리즘[2]에 의해 서로 비밀키를 공유할 수 있다. XTR-DH 키 분배 알고리즘은 다음과 같다. XTR 공개키 데이터로 $p, q, Tr(g)$를 공유한다. 클라이언트와 서버는 비밀키 K를 공유하기 위해 다음과 같이 동작한다.

① 서버는 랜덤 정수 $a \in [2, q-3]$을 선택하고, $n=a, c=Tr(g)$를 [알고리즘 1]에 적용하여 $S_a(Tr(g))$를 계산한다.
그리고 $Tr(g^a) \in GF(p^2)$을 클라이언트에게 보낸다.

② 클라이언트는 서버에게 $Tr(g^a)$를 받고, 랜덤 정수 $b \in [2, q-3]$를 선택한다.
그리고 $n=b, c=Tr(g)$를 [알고리즘 1]에 적용하여 $S_b(Tr(g))$를 계산한다.

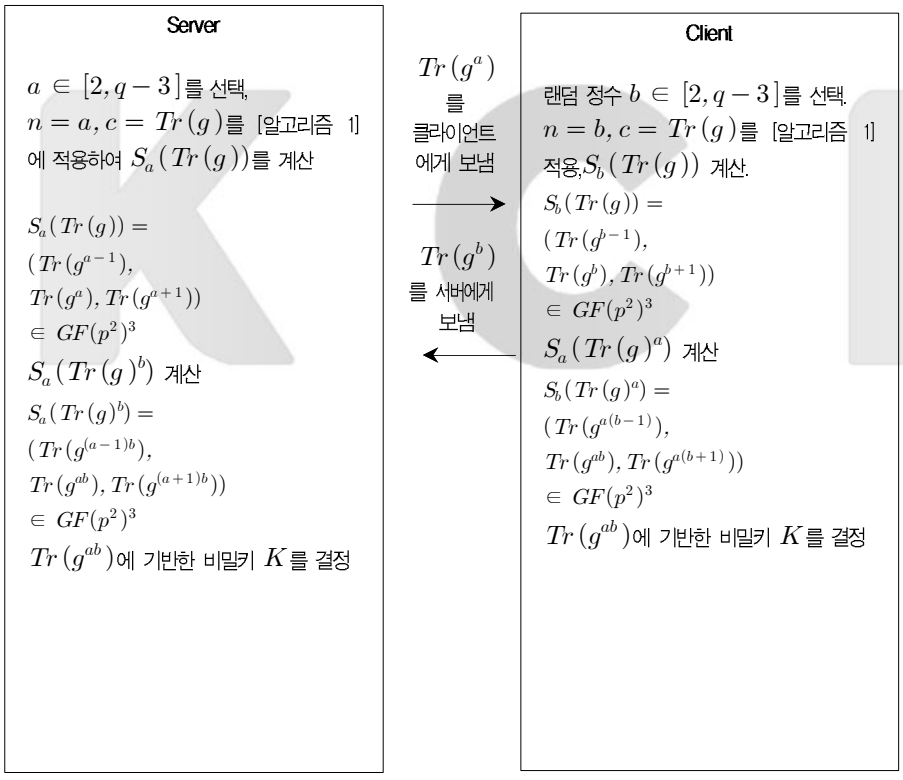

그림 1. XTR-DH 키 분배 알고리즘
Fig. 1 XTR-DH Key distribution Algorithm

2) Diffie-Hellman Key Distribution Algorithm

그리고 $Tr(g^b) \in GF(p^2)$을 서버에게 보낸다.

$S_b(Tr(g)^a) = (Tr(g^{a(b-1)}), Tr(g^{ab}), Tr(g^{a(b+1)})) \in GF(p^2)^3$

③ 서버는 $Tr(g^b)$를 클라이언트로부터 받고, $n = a, c = Tr(g^b)$를 [알고리즘 1]에 적용하여 $S_a(Tr(g)^b)$를 계산한다.

$S_a(Tr(g)^b) = (Tr(g^{(a-1)b}), Tr(g^{ab}), Tr(g^{(a+1)b})) \in GF(p^2)^3$

④ 서버는 $Tr(g^{ab}) \in GF(p^2)$에 기반한 비밀키 K를 결정한다.

⑤ 클라이언트는 $n = b, c = Tr(g^a)$를 [알고리즘 1]에 적용하여 $S_b(Tr(g)^a)$를 계산한다.

$S_b(Tr(g)) = (Tr(g^{b-1}), Tr(g^b), Tr(g^{b+1})) \in GF(p^2)^3$

⑥ 클라이언트는 $Tr(g^{ab}) \in GF(p^2)$에 기반한 비밀키 K를 결정한다.

여기서 서버에 의해 계산된 이웃인 $Tr(g^{(a-1)b})$, $Tr(g^{(a+1)b})$와 클라이언트에 의해 계산된 이웃 $Tr(g^{(a-1)b})$, $Tr(g^{(a+1)b})$은 일반적으로 다르지만, $Tr(g^{ab}) \in GF(p^2)$은 같은 값을 가지므로 이에 기반하여 비밀키를 결정할 수 있다. XTR-DH 기반 키 분배 방식의 계산량과 통신 오버헤드 비용은 기존의 DH 프로토콜보다 훨씬 적다. 이것은 유한체의 곱셈군의 부분군을 기반으로 하는 프로토콜이기 때문이다.

III. 제안하는 XTR-ElGamal 기반 사용자 인증 프로토콜

이 장에서는 XTR-ElGamal 기반의 사용자 인증 프로토콜을 설계하였으며, 이것은 ElGamal 암호 시스템을 기반으로 하고 있다[13]. 또한 XTR-ElGamal 기반의 사용자 인증 프로토콜의 특징에 대해 살펴 보았다. 전체적인 프로토콜은 다음과 같다. 사전에 미리 약속된 정보와 공개되어 있는 정보들은 다음과 같다.

Advance Preparations

- 사전에 클라이언트와 서버 사이에 약속된 대칭키 알고리즘 E
- 클라이언트는 사전에 사용하는 응답의 크기를 서버와 약속해 둔다.

System Setup

- XTR-ElGamal 기반의 사용자 인증 프로토콜에서 시스템 파라미터는 $q \mid (p^2 - p + 1)$을 만족하는 소수 p, q, Tr이다.
- $p \equiv 2 \pmod 3$는 170비트 정도의 소수이며 q는 160비트 정도의 소수이다.
- 위수가 q인 원소 $g \in GF(p^6)$에 대해 적당한 $Tr(g)$를 찾는다.

서버의 변수 선택

- 서버의 개인키 : $s \in [0, q-1]$
- 서버의 공개키 : $v = g^{-s}, Tr(g^a)$
- XTR-ElGamal 기반의 사용자 인증 프로토콜은 다음과 같다.

① 클라이언트는 난수 $b(0 \le b \le q-1)$를 선택하고, $x = Tr(g^b)$를 계산한 후, 서버에게 보낸다. 클라이언트는 $Tr(g^{ab})$를 계산하고, XTR-DH 기반 키 분배 알고리즘에 의해 $Tr(g^{ab})$에 기반하여 대칭키 알고리즘의 비밀키 K를 결정한다. 여기서 대칭키 알고리즘의 키 K를 결정하는 방법은 사전에 클라이언트와 서버 사이에 약속된 방법을 이용한다. 예를 들어 K가 l비트라 하면, 340비트 길이를 갖는 $Tr(g^{ab})$로부터 최상위 l비트를 K로 사용할 수 있다.

② 서버는 $Tr(g^{ab})$을 계산하고, 마찬가지로 XTR-DH 기반 키 분배 알고리즘에 의해 $Tr(g^{ab})$에 기반하여 대칭키 알고리즘의 비밀키 K를 결정한다.

③ 클라이언트는 $y_i = g^{b \cdot p^{2(i-1)}} \cdot g^{-s} \bmod p,$ $1 \le i \le 3$ 을 계산한다. $1 \le i \le 3$에 대하여, 사전에 서버와 약속된 크기 y_i를 선택한다. 이 값을 공유된 비밀키 K를 사용하여 사전에 약속된 대칭키 알고리즘 E로 암호화하여 $E_K(y_i)$를 서버에 전달한다.

④ 서버는 $E_K(y_i)$를 공유된 비밀키 K를 사용하여 복호화한 후 y_i를 구한다. $1 \le i \le 3$에 대하여, $x = Tr(y_i \cdot g^s)$인지를 확인한다. 만약 이 값이 같지 않다면 클라이언트의 인증은 실패한다. 서버는 y_i가 사전에 미리 클라이언트와 약속했던 크기인지를 확인한다.

XTR-ElGamal 기반의 사용자 인증 프로토콜은 다음과 같은 특징을 가진다.

[정리 4]

$y_i = g^{b \cdot p^{2(i-1)}} \cdot g^{-s} \bmod q$일 때, $1 \le i \le 3$에 대하여 세 개의 y_i는 XTR 기반 ElGamal 사용자 인증 프로토콜의 확인 과정을 통과한다.

[증명]

만약 $y_i = g^b \cdot g^{-s} \bmod q$이 서버에게 보내진다면, $i = 1$인 경우에 다음을 만족한다. $Tr(y_i \cdot g^s) = Tr(g^b \cdot g^{-s} \cdot g^s) = Tr(g^b) = x$ 이므로 $x = Tr(y_i \cdot g^s)$가 된다. 만약 $y_2 = g^{b \cdot p^2} \cdot g^{-s} \bmod q$이 서버에게 보내어진다면, $i = 2$인 경우에 다음을 만족한다.

$$Tr(y_i \cdot g^s) = Tr(g^{b \cdot p^2} \cdot g^{-s} \cdot g^s) = Tr(g^{b \cdot p^2})$$

이고 $x = Tr(g^{b \cdot p^2})$이 된다. 또한 이것은 [정리 2]에 의해 $Tr(g^{b \cdot p^2}) = Tr(g^b) = x$가 된다. 만약 $y_3 = g^{b \cdot p^4} \cdot g^{-s} \bmod q$이 서버에게 보내어진다면, $i = 3$인 경우에 다음을 만족한다. $Tr(y_i \cdot g^s) = Tr(g^{b \cdot p^4} \cdot v \cdot g^s) = Tr(g^{b \cdot p^4})$이고 $x = Tr(g^{b \cdot p^4})$이 된다. 또한 이것은 [정리 2]에 의해 $Tr(g^{b \cdot p^4}) = Tr(g^b) = x$가 된다.

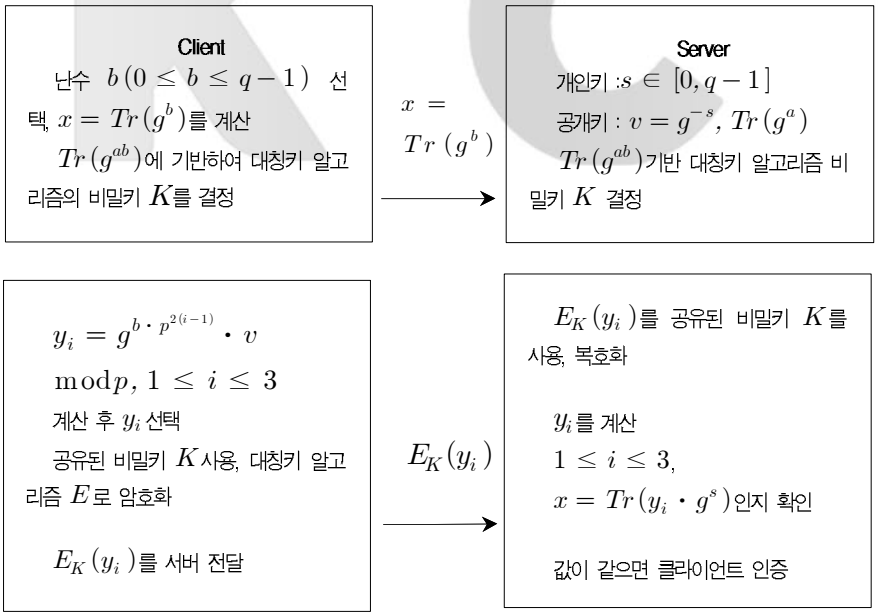

그림 2. XTR-ElGamal 기반의 사용자 인증 프로토콜
Fig. 2 XTR-ElGamal based User Authentication protocol

Ⅳ. XTR-ElGamal 기반의 사용자 인증 프로토콜의 효율성 분석

본 논문에서는 ECC와 XTR에서의 키 사이즈를 비교하였다. ECC와의 비교에서 170비트의 소수체 상의 랜덤 곡선으로 170비트 위수의 부분군의 곡선을 가정하고, $GF(p)$ 상에서 170비트의 ECC와 170비트의 XTR 응용 프로그램에서 요구되는 곱셈 수를 비교하였다. ECC의 결과는 [8]에 기반을 두고 이론적으로 계산하였으며, XTR의 경우는 [1][2]에 근거한 지료에 의해 실험하였다. ECC와 XTR에서 공개키가 공유된다면, ECC와 XTR의 공개키들은 공유되는 점으로 구성된다. ECC의 경우에, y좌표는 x좌표에 의해 유도되며, 싱글 비트이다.

또한 본 논문에서는 ElGamal 사용자 인증 프로토콜에 XTR 암호 시스템을 적용함으로써 프로토콜 진행 시 키 길이의 감소와 연산량의 효율성을 높일 수 있었다. XTR은 170비트의 키 길이로 1024 비트의 키 길이를 갖는 이산 대수 문제에 기반한 암호 시스템과 동일한 안전성을 가지며, 빠른 연산 속도를 가진다[3][4][10].

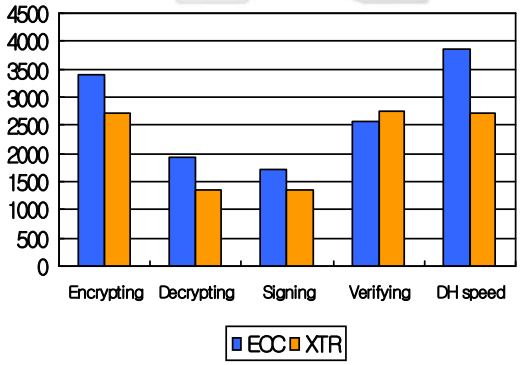

그림 3. GF(p)에서의 ECC와 XTR의 곱셈수의 비교
Fig. 3 Compare of multiplication number in GF(p)

본 논문에서의 효율성은 ElGamal 사용자 인증 프로토콜에 XTR 암호 시스템을 적용함으로써 프로토콜 진행시 키 길이의 감소와 연산량의 효율성을 높일 수 있었다.

표 1. XTR-ElGamal 사용자 인증 프로토콜의 효율성 비교
Table 1. Compare efficiency of user authentication protocol based of XTR-ElGamal

구분		DLP 기반사용자 인증	XTR 기반 ElGamal 사용자 인증
p		1024비트	170비트
q		512비트	160비트
키길이		1024비트키길이	160비트
연산량	키분배	pow mod 연산 1번(1024비트)	곱셈 연산 $8\log_2 k$ 번(160비트)
	사용자 키분배, 키 설정	pow mod 연산 1번(1024비트)	곱셈 연산 $8\log_2 k+1$ 번(160비트)
	서버의 키 분배, 인증	pow mod 연산 1번(1024비트)	곱셈 연산 $32\log_2 p+9$ 번(160비트)
전송량		1024비트의 K	1024비트의 $S_k(Tr(g))$

XTR은 170비트의 키 길이로 1024 비트의 키 길이를 갖는 이산 대수 문제에 기반한 암호 시스템과 동일한 안전성을 가지며, 빠른 연산 속도를 가진다.

<표 1>은 XTR 기반과 DLP 기반에 대한 사용자 인증 프로토콜의 효율성을 비교하였다. DLP 기반의 프로토콜에서는 키 공유를 위해 pow mod 연산을 1번하게 된다. 그리고 사용자 인증을 위해 클라이언트가 pow mod 연산을 1번하고, 서버가 사용자 인증을 위해 pow mod 연산을 1번을 한다.

[5][12]의 결과와 비교하여 볼 때에 연산량이 줄어든 것을 알 수 있으며, 또한 [9]에 기반하여 <표 1>에서와 같이 DLP 기반의 사용자 인증 시스템에서의 효율성을 알 수 있다.

XTR 기반의 프로토콜에서는 키 분배를 위해 필요한 $S_k(Tr(g))$ 연산에는 $8\log_2 k$ 번의 곱셈 연산이 필요하다. 제안하는 프로토콜에서는 비밀키 공유와 공개키를 위해 $8\log_2 k+1$ 의 곱셈 연산이 필요하다.

또한 서버에서의 사용자 인증을 위해 키 공유에 필요한 $8\log_2 p$ 번의 곱셈 연산과 인증시 요구되는 $24\log_2 p+9$ 의 곱셈 연산이 필요하다.

Ⅴ. 결론

본 논문에서는 효율적인 사용자 인증 프로토콜을 설계하기 위해서 문제 도메인을 기존의 유한체인 $GF(p^6)$을 명확하게 구성하여 사용하지 않고, $GF(p^6)$의 부분군인 $GF(p^2)$을 이용하여 문제를 해결하고자 XTR-ElGamal 기반의 사용자 인증 프로토콜을 제안하였다.

XTR-ElGamal 기반의 사용자 인증 프로토콜은 유한체상의 부분군들을 이용하여 키 교환 시 요구되는 비트 수를 줄이고, 계산 시 필요한 오버헤드를 줄여 쉬운 계산과 실행을 제공하였다.

XTR-ElGamal 기반에서 사용자 인증을 하기 위한 과정에서 필요한 $y_i = g^{b \cdot p^{2(i-1)}} \cdot v \bmod q, 1 \le i \le 3$와 인증 프로토콜을 설계하였다. 또한 XTR-DH 기반 키 분배 알고리즘을 적용하여 $Tr(g^{ab})$에 기반하여 대칭키 알고리즘의 비밀키 K를 공유하기 때문에 함축적 키 인증성을 제공하며, 사용자의 비밀키를 알지 못하는 공격자는 사용자로 위장하여 서버에 로그인을 할 수 없도록 설계하였다. 사용자의 비밀키가 노출되는 경우에 대하여서는, 서버와 약속된 크기 y_i의 값을 알 수 없기 때문에 사용자 인증확률을 낮춤으로써 안전성을 높였다. 또한 각 세션마다 서로 다른 랜덤수를 선택하여 암호화하여 전송하기 때문에 알려진 키에 대한 안전성도 보장하였다. 그리고 <표 1>에서와 같이 연산량과 전송량 면에서도 효율적임을 알 수 있다.

참고문헌

[1] David Jablon, "Public Key Methods for Shared Secret Authentication.", RSA'98, Jan. 14, 1998

[2] ANSI X9.63, Elliptic Curve Key Agreement and Key Transport Protocol, working draft, July 1998

[3] A. Menezes, Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston 1993

[4] Miller, V.,Uses of elliptic curves in cryptography, Advances in Cryptology, Lecture Notes in Computer Science, Volume 218, Springer-Verlag, pages 417-426, 1986.

[5] 이재욱, 전동호, 최영근, 김순자, XTR 암호 시스템 기반의 대리 서명, 정보보호학회 논문지, VOL.13, NO.3, 2003

[6] A. K. Lenstra, E. R. Verheul, Key improvements to XTR, Proceedings of Asiacrypt 2000, LNCS 1976, Springer-Verlag, 220-233

[7] A. K. Lenstra, E. R. Verheul, Fast irreducibility and subgroup membership testing in XTR, Proceedings of PKC 2001

[8] A. Menezes, S.Vanstone, ECSTR(XTR): Elliptic Curve Singular Trace Representation, rump session of Crypto2000.

[9] 김경진, 김성덕, 심경아, 원동호, 이산 대수 기반의 Diffie-Hellman형 표준 키 분배 프로토콜의 안전성 분석에 관한 연구, 정보처리학회 논문지C, VOL.9-C, NO.6, 2003

[10] ANSI X9.42, Agreement of symmetric Key on using Diffie-Hellman Cryptography, 2001

[11] ANSI X9.63, Public Key Cryptography for the financial services industry : key agreement and key transport using elliptic curve cryptography, 2001

[12] 한동국,박혜영, 박영호, 김창한, 임종인, XTR 버전의 개인 식별 프로토콜을 이용해 블랙메일링을 막는 실질적인 방법, 정보보호학회 논문지, VOL.12, NO.1,1998

[13] T. ElGamal, "A Public key Cryptosystemand Signature Scheme Based on Discrete Logarithm", IEEE Trans. Information Theory, Vol.31, No.44, pp. 469 ~472, 1985

저 자 소 개

정 경 숙

1995년 2월 경희대학교
수학과 졸업
1997년 8월 경희대학교
컴퓨터공학과 석사
2004년 2월 경희대학교
컴퓨터공학과 박사
2000년 3월 ~ 2004년 2월
용인송담대학 컴퓨터소프트웨어
학과 겸임 교수
2004년 3월 ~ 현 재
가톨릭대학교 컴퓨터정보학부
강의전담초빙교수
<관심분야> 정보보호, 인공지능,
전자상거래, 기계학습