ECC를 이용한 키분배 프로토콜

A Key Distribution Protocol based on ECC

이 준* 김 인 택*
Lee, Jun Kim, In-Taek

ABSTRACT

In this paper we suggest a key distribution protocol based on ECC. This could be apply to multi connection to a sensitive system on a computer network. SSL based on RSA is generally used as a key distribution protocol. By reducing two times encryption/decryption procedures to one time and using ECC algorithm this protocol is faster than SSL. Analyzing the key distribution time on a normal PC experiment, we show that this could be practically used in real world without a hardware implementation.

주요기술용어(주제어) : Key Distribution(키분배), ECC(타원곡선암호)

1. 머리말

국방망에서 사용되는 컴퓨터 시스템은 보안을 요구한다. 특히 워게임 시뮬레이션은 군사작전을 가상으로 개발된 시스템[1]이므로 요구수준이 높고 다수의 인가된 사용자만 동시에 접속해야 한다. 이러한 워게임 시뮬레이션 시스템은 군사지도 및 군 관련 자료가 포함된 시나리오 파일 등의 군사기밀자료를 이용하여 시뮬레이션을 한다[2]. 네트워크 제어 기술 발달로 워게임 참여자는 특별히 지정된 보안장소로 이동하여 진행하기 보다는 네트워크를 통하여 물리적인 이동 없이 접속하여 게임하기를 원한다. 네트워크 제어는 대부분 TCP/IP를 이용하여 구현하고 있다. TCP/IP 기반으로 하고 있는 서비스는 IP 네트워크가 갖고 있는 특징을 그대로 따르기 때문에 IP 네트워크에서 알려진 보안 취약점 역시 그대로 갖고 있다. IP 네트워크는 전송되는 데이터의 도청이 기본적으로 아주 용이하다[3].

도청이 용이한 네트워크에 연결된 컴퓨터에 대해서 정당한 사용권한이 없는 제 3 자의 악의적인 접근이 가능하게 된다면 워게임 시뮬레이션 시스템처럼 내부에 저장된 군 관련 자료와 같은 중요한 자료는 쉽게 유출된다. 무자격자의 악의적인 접근 방지에 대한 연구로 네트워크에 연결된 시스템에 접근할 자격 여부를 확인하고, 자격자만 시스템에 참가할 자격을 부여해야 하며 네트워크 사용에 따른 도청과 누설 방지 대책이 필요하다. 워게임과 같이 다수가 동시에 사용하는 중요 시스템의 접속과 대화는 112비트 이상의 블록 암호 알고리즘을 사용하면 악의적인 공격으로부터 안전하다[4]. 본 논문은 워게임 시뮬레이션 시스템을 예로 보안을 요구하는 시스템에서 다수의 참여자가 원거리에서 네트워크로 참여할 때 자격자에게만 블록 암호 알고리즘의 비밀키를 신속하게 분배하는 방안을 제안한다.

† 2007년 4월 30일 접수~2007년 6월 8일 게재승인
* 공군사관학교(Korea Air Force Academy)
주저자 이메일 : jlee@afa.ac.kr

2. 기존방법과 문제점

워게임 시뮬레이션 시스템 보안체계에 대한 연구는 다음과 같이 요약된다[2].

① 워게임 시뮬레이션 시스템의 접속 자격을 확인하기 위하여 별도의 인증서버를 설치한다.

② 워게임 시뮬레이션 시스템 관리자는 워게임 참가자에게 id와 password를 발급하고 SSL을 통하여 인증서버에 id와 password를 등록한다.

③ 워게임 시뮬레이션 시스템과 워게임 참가자(이하 참가자) 사이는 128비트 이상의 블록 암호 시스템이 설치된다.

④ 워게임 참여시 참가자는 SSL로 인증서버에 접속하여 id와 password를 안전하게 제출함으로 워게임 참가 자격을 인증 받는다.

⑤ 인증서버는 id와 password가 확인된 참가자에게 비밀키(워게임 시스템과 참가자 사이 블록암호체계)를 SSL을 이용하여 참가자에게 전달한다.

⑥ 참가자는 비밀키를 사용하여 워게임 시뮬레이션 시스템과 안전한 대화통로를 확보한 후 id와 password로 자격을 다시 확인 받고 워게임 시뮬레이션 시스템에 접속하여 게임어를 전달하고 그 결과를 받음으로 제 3 자의 참여 및 도청을 불가능하게 한다.

위와 같이 구성된 보안체계는 워게임 시뮬레이터의 성능저하 초래 없이 실험하였음을 보고하고 있다. 하지만 다음과 같은 문제점을 예상할 수 있다

① 워게임 시뮬레이션 시스템 관리자 외에 인증서버 관리자 역시 참가자의 id와 password를 관리하므로 인증서버 관리자에 의한 man in middle의 문제가 발생할 수 있다. 전산관련 보안사고의 대부분은 man in middle 문제이므로 인증서버 관리자는 워게임 참가자의 id와 password를 알 수 없게 하는 보안대책이 필요하다.

② SSL은 전자서명과 암호화 기법에 SHA-1, RSA와 DES를 사용한다. RSA는 단위비트 당 암호의 강도가 ECC보다 낮으며 1024비트의 RSA와 160비트의 ECC는 안전도가 비슷하다[5]. 또한 동일한 계산능력의 컴퓨터 실험에서 같은 크기의 키에 대하여 ECC는 RSA보다 암호화 및 복호화 속도가 훨씬 빠르며 키 크기가 클수록 ECC가 훨씬 빠른 것이 보고되었다[6]. 따라서 블록암호 비밀키 분배는 160비트 ECC를 활용함이 1024비트의 RSA와 동일한 안전도를 보장받을 수 있고 속도면의 향상도 함께 예측할 수 있으므로 ECC 사용이 바람직하다.

③ 참가자는 인증서버에 자신의 id와 password를 안전하게 제출하기 위하여 SSL로 서버와 클라이언트 사이의 대화통로를 개설한다. 이 과정에서 인증서버는 참가자를 인증하고, 참가자에게 블록암호의 비밀키를 안전하게 전달한다. 하지만 SSL을 사용하지 않고 절차를 더 단순화하여 공개키로만 참가자를 인증하고, 비밀키를 안전하고 신속하게 분배할 수 있다.

이에 대한 개선방안을 다음과 같이 제안 한다

3. 프로토콜 제안

가. 기호 및 용어

본문에서는 워게임 시뮬레이션 시스템 접속서버를 WS, 인증서버를 AS, 인증서버의 메시지는 M_X로 표기한다. 또한 클라이언트를 C, 메시지 M에 대한 해쉬 함수는 h(M)으로 표시하고 해쉬 값은 H로 나타낸다.

나. 유한체 F_p위에서의 타원곡선 군과 공개키

유한체 F_p위에서의 타원곡선 군의 구성 요소는 다음과 같이 정의된다.

$$\{(x, y) \mid y^2 \bmod p = (x^3 + ax + b) \bmod p\} \cup O$$

O(infinite) $a, b \in F_p$ $\quad 4a^3 + 27b^2 \neq 0 \bmod p$

ECC의 공개키는 P, a, b, 타원곡선의 한 점 Q, 개인키 k로 곱한 kQ로 구성된다. 이때 k는 공개키에 대한 개인키로 비공개한다.

1) ECC 암호화

① 송신자는 공개키를 이용하여 타원방정식의 한

점 M을 생산한다.
② 공개키 kQ에 m을 곱하여 M을 더하여 S1= m(kQ)+M을 얻는다.
③ 공개키 Q에 m을 곱하여 S2 = mQ를 얻는다.
④ 송신자는 { S1, S2 }를 공개자(수신자)에게 송신한다.

2) ECC 복호화
① 공개자는 { S1, S2 }를 수신한다.
② 수신자의 개인키 k를 수신한 S2에 곱하여 k(S2)를 얻는다.
③ 수신한 S1에 ②에서 얻은 결과 k(S2)를 이용하여 메시지 M =S1-k(S2)을 얻는다.

다. 시스템 구성

시스템의 구성은 성능저하가 초래되지 않음 보고[2] 결과에 따라서 워게임 시뮬레이션 시스템과 인증서버를 별도로 운영한다. 워게임 시뮬레이션 시스템과 참가자 사이는 128비트 이상의 블록암호 알고리즘으로 구성된다. 인증서버는 접속자의 자격을 확인하고 워게임 시뮬레이션 시스템 접속 블록암호의 비밀키를 전달한다.

라. 인증 발급 및 키분배

신청절차에서 워게임 관리자는 신청자의 신분을 확인한 후 인증서버에 참가자의 인증서 발급을 신청한다. 게임 참가시 인증서버는 신청자의 인증서를 확인한 후 워게임 서버 접속 비밀키를 참가자에게 분배하여 참여한다. 참가자는 인증서버로부터 1회의 인증 발급을 받으며, 비밀키는 워게임 참가마다 새롭게 받는다.

1) 사용 신청
① 신청자 C는 자신의 ECC 공개키{ P, a, b, Q, kQ }를 WS에 제출한다. 이때 C는 개인키 k를 공개하지 않는다.
② WS는 C의 참가자격을 확인한 후 적격자에 대해서 AS에 C의 공개키 { P, a, b, Q, kQ }를 제출하고 인증키를 요구한다.
③ AS는 인증키를 발급하기 위하여 인증시스템을 준비한다. 예를 들면 1024비트 크기의 RSA의 공개키 d 와 개인키 e를 준비한다.
④ AS는 공개키를 문자열 Str로 만든다.
Str = P + a + b + Q + kQ
⑤ AS는 해쉬함수 h를 이용하여 공개키에 대한 해쉬값 H를 구한다. H = h(Str)
⑥ 개인키 e를 이용하여 인증키 A를 구한다.
$A = H^e$
⑦ AS는 C에 대한 인증키 A를 WS에 송신한다.
⑧ WS는 C에게 A와 WS접속을 위한 id 및 pass-word를 발행한다.

2) 프로토콜
① WS는 AS에게 메시지 블록암호 비밀키 M_x를 워게임 마다 안전한 경로를 통하여 송신한다.
② 인증서가 있는 C는 AS에 자신의 공개키와 인증키 A를 제출한다.
③ AS는 C의 공개키로 문자열 Str을 만든다. 해쉬함수 h를 이용하여 H = h(Str)을 구한 후 자신의 공개키 d를 이용하여 인증키를 확인한다. 만일 $H = A^d$이면 참가자격을 인정하고 C의 공개키를 이용하여 메시지 M(M_x, M_y)을 생산한다.
④ AS는 M을 C의 공개키로 M을 암호화하여 C에게 { m(kQ)+M, mQ } 송신한다.
⑤ C는 개인키 k를 이용하여 AS로부터 받은 암호문을 복호화하여 M을 구한다.
⑥ M의 x 성분 M_x를 이용하여 WS와 C 사이의 블록 암호의 비밀키로 사용한다.
⑦ WS와 C 사이에 블록 암호체계가 개설된 후 C는 id와 password를 사용하여 워게임 참여 자격자임을 확인한다.
⑧ 참여자격 확인 후 WS와 C 사이의 의사소통은 블록 암호를 사용하여 도청을 방지한다.

3) 안정성 검토
가) man in middle

인증서버 관리자가 WS와 C 사이의 블록 암호의 비밀키 M_x를 알고 있다 하더라도 WS에 등록된 id와 password에 관한 정보가 없으므로 워게임에 접속할 수 없다. WS에 의해 확인된 참여자만 워게임에 접속

할 수 있다.

나) 도청

C가 인증서버 AS에 접속할 때 공개키와 인증키를 공개적으로 접속하여 제 3 자에게 노출될 수 있지만 C의 공개키를 이용하여 비밀키 M_x를 분배하므로 제 3 자가 AS가 C에게 주는 암호문을 도청했다 하더라도 C의 개인키 k가 없으므로 M_x를 구할 수 없으며 WS와 제 3 자의 접속은 불가능하다. 따라서 AS가 C의 공개키를 이용하여 비밀키 M_x를 전달함은 도청 방지 및 공개키 소유를 확인하는 절차가 된다.

다) 비밀키의 재사용

새로운 워게임 운영마다 WS는 M_x를 변경하여 AS에게 SSL과 같은 안전한 경로로 통보한다. 따라서 이전 워게임에서 사용했던 비밀키를 알고 있다 하더라도 새 게임에서는 비밀키 M_x가 변경되었으므로 WS와 C사이의 블록 암호체계를 이용한 대화통로가 개설될 수 없으므로 새로운 워게임에는 참여할 수 없다.

4) 효율성 검토

인증서버 AS와 참가자 C 사이에 SSL을 사용할 경우 C는 AS의 공개키를 이용하여 인증자료를 전달하고 AS는 C의 공개키를 이용하여 메시지 M을 전달한다. 그 후 AS와 C 사이의 블록 암호의 비밀키로 M을 사용하여 안전한 대화 통로를 개설한다. 이때 공개키 암호화 및 복호화 절차는 2회, 블록 암호절차는 1회를 완료한 후 WS접속 비밀키 M 을 얻는다.

새롭게 제안한 프로토콜은 비밀키 송신에 AS가 C의 공개키로 1회의 암호화 및 C는 개인키 k를 이용한 1회의 복호화만 요구된다.

또한 일반적으로 SSL은 공개키의 암호화 및 복호화에 안전한 1024비트 RSA 알고리즘을 이용하여 속도가 떨어지나 ECC 공개키 체계는 160비트의 규모에서 비슷한 안전도를 얻을 수 있으므로 속도가 빠르다.

비밀키 분배에서 효율성은 표 1과 같이 정리된다.

[표 1] 키분배 효율성 비교

내 용	SSL	제안된 방식
공개키 방식	1024비트 RSA	160비트 ECC
암호화 요구	2회	1회
복호화 요구	2회	1회
블록 암호 요구	1회	없음

4. 실험

본 논문은 워게임에 접속하는 다수의 사용자에게 비밀키를 안전하게 전달하는 방안을 제시하고 있으며 기본적으로 ECC공개키를 이용하여 비밀키를 배분하는 방식을 제안한다. 공개키를 이용한 사용자의 인증과 블록 암호의 비밀키 생산 및 암호화 및 복호화 과정이 계산의 대부분을 차지한다.

본 실험은 160비트의 ECC공개키를 생산하고 인증서버가 인증에 소요되는 시간과 암호화 및 복호화에 소요되는 시간을 측정한다. 이하의 실험에서 표기된 수는 16진법을 사용하여 표기하였다.

가. 실험장비

실험에 사용된 장비는 노트북(Intel Celeron M 1.40 GHz, 496MB, window XP)을 사용하였으며 Visual C++로 프로그램을 작성하였다.

나. ECC 공개키 생산

160비트 규모의 ECC[7]의 공개키는 다음과 같이 임의적으로 생산하였다. 개인키 k는 비공개이며 괄호 내부는 기호에 대한 설명이다.

P : 12cf 1216 00e4 39cd 2ef2 70f3 6bb7 4604 0059 7cab (ECC field Z_q)
a : 0e15 5ec2 0e65 36c1 3d9e 2ea6 1fc3 56f0 6ec2 50f3 (타원 방정식 계수 a)
b : 01cc 526e 433f 5303 2ca4 5a2c 6f5e 1128 0a2c 28aa (타원 방정식 계수 b)
Q_x : 05e1 61ea 326b 0b27 61d6 211c 35cd 1967 3d27 4a66 (Q의 x 좌표)

Q_y : 020f 7ec1 6d2d 2259 5594 84f2 6137 c18a 6a55 f456 (Q의 y 좌표)
kQ_x : 04b2 9f5e 3f09 55bf f20a 8180 dd2a 5695 f075 2d7d (kQ의 x 좌표)
kQ_y : 04ce dd29 fca7 9c12 a584 1305 fd77 037f 860d 0e26 (kQ의 y 좌표)
k : 0d8c 27d5 228e 41c0 3c28 4366 0b17 0f6a 11bd 54cc (ECC 개인키)

다. 문자열

문자열은 Str = P + a + b + Q_x + Q_y + kQ_x + KQ_y 순서로 문자열을 형성하였으며 해쉬함수는 MD5[8]를 사용하였다. 공개키에 대한 해쉬함수 값은 다음과 같다.

H = MD5(Str) = 2a91 9971 5384 9569 4e91 e903 45a4 5d10

위의 공개키로 형성된 문자열 Str에 대해서 해쉬함수 MD5를 10000번 반복하는데 소요되는 시간은 3초이며 함수 1회 소요시간은 0.0003초 이다.

라. 인증키 생산

170비트의 DSA형식을 사용해서 EC전자서명을 만드는 것이 RSA보다 훨씬 빠르다. 그러나 전자서명을 사용하는데 RSA는 EC DSA보다 빠르다[9]. 그러므로 대부분의 인증서는 RSA방식을 사용한다. 실험에서 인증서버 AS는 1024비트 RSA[10]방식으로 인증키를 생산하기 위하여 512비트의 유사소수(pseudo prime number) 두 개 p1과 p2를 선택하였다. Φ(N) = (p1-1)(p2-1) 이다.

p1 = 69d7 0ed4 4ce3 7d62 518c 7309 07d3 2427 2bc0 7925 369b 56bd 5076 69db 0b2f 1b16 1a68 2f4e 183d 05fb 24cb 2b2e 5ebe 6251 6582 3824 59cc 5d9a 3792 5a71 7cb3 5dbf
p2 = 7f4d 6e37 3c9f 7a7e 134a 289c 04b5 2aa8 1ec6 1275 017e 6793 2777 2e53 2f55 6ffe 15ac 1595 0b2c 31e0 3110 7c8c 1623 0fac 60db 41dd 3759 0ad3 3c49 0d26 7001 778a
Φ(N) = 7eab f596 5a4c 1c6b 7340 377c 9a1a e703 cedd 65b6 a696 c230 d9dc 709e ad4b 4abc 2aee 3147 e7f7 d3a9 fd47 974c fe53 1054 594b d132 d0a9 f8ac 2bcd fc13 661b c42a 2855 60e1 b444 987e 2de5 a0b6 7b58 c4e3 4181 5d68 e1de 2944 4d72 c5fe 00e7 626f 35d4 a968 1b27 7a70 b0d0 1575 db5d 8b77 4e05 d9b4 0e67 202d 0ff7 c0d8 b534 2bc6

512비트 규모의 유사 소수(pseudo prime number)를 선택하여 공개키 d로 선택하고 그 역수를 개인키 e로 하였다.

d = 19b 1249 486 7e70 c82 65ca 2aab 184c 4f4a 4ca1 47d9 2de8 2047 1879 4b22 5db 3042 6948 7db1 975 34de 978 7e5e 2201 60ca 5545 1b7f 7b47 2c3b 3166 7f00 4f55
e = 387b 732 52f4 bb73 50b3 cecb c35b 9064 2227 cd00 946d 90a0 8497 2707 e59e e244 5f41 512a eb83 b3e5 4af3 6e33 3138 89ea 160d 0d16 aa45 6f0f 6e2f 1d3e 293c 2a82 df54 0b41 a3ed e0eb e870 1ffc 63e9 6cdb 1c1a 5f1a ef01 7414 2aaa abcd 0f79 8f38 6aa3 9ddf 36a5 74af b1ff 037e 16a3 0b93 e5de 7625 18a3 ffc1 ffb2 93b7 8e41 196c

512비트의 고정된 수에서 공개키를 선택하고 그 역수를 찾으므로 개인키는 공개키보다 수의 규모가 크다. 수의 규모가 클 경우 인증키를 이용하여 인증하는데 시간이 많이 소요된다. 인증키를 생산하는 횟수는 1회이고 사용하는 횟수는 매번 접속할 때 이므로 매 사용마다 시간단축을 위해서 적은 수를 공개키로 사용했다.

H^e mod N(where N = p1 * p2)으로 계산한 결과 인증키 A는 다음과 같다.

A = 1909 eaa2 9b4a 0366 0726 31eb a6c8 ed55 32c3 1ac8 ab46 8d9a 7fd2 addb bd1c 899b b4c2 8344 68ff d46b 60e8 ac49 2b20 4982 3ea1 7922 3bec 01b5 f0ed c9d8 b262 7172 6845 622d 37ea b2fa cf3c 6d1e cb8a 3aee 2712 052c bf5e 271b b8b2 04f2 102f 58c2 0a55 365a 55b1 1a9e 4924 6bd9 a3eb e78d 43dd b8e0 f194 a17e 31c8 c8d0 50da 19dc

인증키를 접수한 후 인증기관의 공개키를 이용하여 $H' = A^d \bmod N$ 계산한 결과와 MD5를 계산한 해쉬 값 H를 비교함으로 인증한다. 인증시간에 사용되는 시간을 점검하기 위해서 10000번을 반복함으로 소요시간을 점검하였다. 1024비트 인증키를 10000번 계산에 소요되는 시간은 1시간 33분 57초로 1회 계산 시간은 0.5637초가 소요되었다.

마. ECC 암호화 및 복호화

위의 160비트 ECC 공개키를 사용하여 임의의 난수를 사용하여 메시지 M을 만든 후 암호화/복호화를 10000번 실험하였다. 소요된 시간은 2시간 38분 19초로 암호화/복호화의 1회 평균 시간은 0.9499sec 이다. 메시지 M은 160비트로 구성된 (M_x, M_y) 순서쌍으로 M_x의 적당한 위치에서 128비트를 선택하여 WS와 C 사이의 블록 암호체계의 비밀키로 사용할 수 있다. AS가 M을 생산할 때 알고리즘에 의해서 선택될 128비트는 모든 참가자에 대해서 공통으로 유지한다.

5. 결 론

워게임 시뮬레이션 시스템과 같이 높은 수준의 보안이 요구되면서 다수가 함께 접속하는 시스템의 비밀키 배분을 안전도가 보장된 160비트의 ECC를 이용하여 제안하였고 신속하게 배분할 수 있음을 보였다.

제안된 프로토콜은 공개키에서 생산된 인증키를 확인하여 비밀키를 메시지로 전송함으로 인증서버에서 공개키 암호화 및 복호화 과정이 SSL에 비해서 반으로 축소되었으며 블록 암호화 과정은 필요하지 않아 기존의 제안된 방식보다 절차적인 측면에서 향상을 보였다.

블록암호의 암호화 및 복호화는 많은 시간이 소요되지 않는다[11]. 본 실험에서는 일반적 수준의 노우트북 컴퓨터를 사용하여 인증서버가 인증에서부터 비밀키 분배완료까지 소요되는 시간을 분석적으로 측정하였다. 실험결과에서 인증서버가 공개키를 인증하고 비밀키를 배분하는데 일반적 계산 환경에서 1.5초미만이 소요됨을 보임으로 특별한 하드웨어 구성없이 소프트웨어로 실현가능함을 보였다.

참 고 문 헌

[1] DoD, USD(A&T), “DoD Modeling and Simulation Master Plan”, October, 1995.

[2] 송종석, 김진수, 신문선, 류근호, “워게임 시뮬레이션 시스템을 위한 보안시스템 설계 및 구현”, 정보처리 학회지 논문지C 제12-C권, 3호, 2005. 6.

[3] 신효필, VoIP에 연동하는 VXML 기반의 음성 플랫폼 개발(정보통신산업기술개발사업 연구보고서), 정보통신부, 2003.

[4] 홍성룡, 조정호, “SDR System 적용을 위한 한국형 암호 알고리즘(SEED) 구현 및 성능분석”, 한국정보과학회 2002.

[5] 이인수, 타원곡선 공개키 암호시스템 현황(연구자료 보고서 1호), 한국정보보호센터, 1998.

[6] 정은희, 이병관, “ECSSL 기반 DIT 에이전트”, 정보 처리학회지 논문지B 제9-B권, 5호, 2002. 10.

[7] 이 준, ECC 공개키 암호의 실용적 구현에 관한 연구(연구보고서 KAFA 05-23), 공군사관학교, 2005.

[8] http://www.ietf.org/rfc/rfc1321.txt

[9] H. X. Mel, Doris Baker, Cryptography Decrypted, Addison Wesley, 2001.

[10] 이 준, RSA 공개키 암호의 실용적 구현에 관한 연구(연구보고서 KAFA 02-1-3-9), 공군사관학교, 2002.

[11] http://www.nsri.re.kr/ARIA/