

서명시스템을 이용한 키 분배 프로토콜

이 창 순*

초 록

서명시스템을 이용한 키 분배 프로토콜을 제안한다. 기존의 서명시스템을 이용한 키 분배 프로토콜에 대하여 분석하고 고찰한다. 그리고 공격 당할 수 있는 여러 방법에 대하여도 논한다. 여기에 보다 안전한 키 분배 프로토콜을 제시한다. 제안된 키 분배 프로토콜은 인증기능도 겸하고 있어 불법 사용자를 사전에 막을 수 있다.

I. 서 론

정보통신 기술의 발달과 더불어 컴퓨터의 이용보급이 확대되면서 컴퓨터 통신망을 통한 정보의 교환이 일반화되어가고 있다. 앞으로의 정보사회에서는 보다 신속하고 대량으로 각종 정보를 다자간에 주고 받게된다. 컴퓨터 통신망을 이용함으로써 파일전송, 전자사서함, 문서 송수신, 전자회의 그리고 사업 및 기타 업무에 관한 정보교환 등 다양한 서비스가 제공될 수 있다. 이와같은 편리한 장점에도 불구하고 전송되는 정보데이터에 대한 적절한 보호조치가 없으면, 귀중한 정보가 불법적인 행위에 의한 도청, 삭제 및 수정등의 위협에 놓이게된다. 따라서 이런 불법적인 행위로 인하여 사적 침해, 국가 기밀 누설 뿐아니라 막대한 경제적 손실 등이 초래될 수도 있어 정보보호가 중요하고도 필수적인 문제로 되고 있다.

정보보호를 위하여는 설비면에서의 물리적인 대책이나 제도적인 면에서의 법적인 대책에 비해 기술면에서의 정보보호 대책인 암호화시스템(cryptosystem)의 사용이 보다 효과적이고 경제적인 방법이다 (Hellman, 1978, Newman and Omura, and Pickholtz, 1987) 암호화시스템에는 크게 두 가지가 있다. 하나는 암호화 및 복호화에 동일한 키이를 사용하는 대칭 키이 암호화시스템(symmetric key cryptosystem)이고, 다른 하나는 암호화 키이와 복호화 키이가 서로 다른 비대칭 키이 암호화시스템(asymmetric key cryptosystem)이다[3]. 대칭 키이 암호화시스템은 속도가 빠르고, 구현방법 등에 있어서 비대칭 키이 암호화시스템에 비해 우수하나 가장 문제가 되는 것이 키이관리(key management)이다. 대표적인 대칭 키이 암호화시스템의 예로는 Lucifer(Sorkin, 1984), DES(Data Encryption Standard)(FIFP, 1977) FEAL

* 경산대학교 자연과학대학 정보처리학과 교수

(Fast Data Encryption Algorithm)(Shimizu and Miyaguchi, 1987), IDEA(Lai and J. Massey, 1994) 및 RC5(Rivest, 1995)등이 있다. 반면에 비대칭 키이 암호화시스템을 두 통신자간의 키이를 분배하는데 사용하면 처리 속도는 상대적으로 느리지만 키이관리 및 인증(authentication) 등의 기능들이 효과적으로 수행될 수 있다. 그래서 일반적으로 정보보호를 위한 암호화시스템의 구현에서는 키이의 분배에는 비대칭 키이 암호화시스템을 이용하고, 이 분배된 암호화 키이를 사용하는 실제 정보의 송수신에는 대칭 키이 암호화시스템을 이용한다. 키이 분배 시스템에 적용될 수 있는 대표적인 것으로는 Diffie-Hellman (Diffie and Hellman, 1976)방식과 RSA (Rivest, Shamir, and Adleman , 1978)방식 등이 있다.

미국등 각 국가에서는 표준 대칭 키이 암호화 시스템을 규정하여 사용하고 있다. 반면 비대칭 키이 암호화시스템은 아직 표준화 되어 있지 않고 있으며, 비대칭 키이 암호화시스템 중에서 비교적 안전하면서도 실제 적용 가능한 것으로 알려지고 있는 D-H 방식과 RSA 방식을 이용한 키이 분배 및 인증 등에 관한 많은 연구가 이루어지고 있다. 또한 최근에는 키이 분배와 신분 인증을 동시에 수행하여 채널 설정단계에서부터 안전성과 신뢰성을 확보하기 위한 연구도 활발하다. RSA 방식은 합성수 법(modulus)에서의 떡승을 이용하고, D-H 방식은 유한체 GF(q), q 는 소수 혹은 소

수의 떡, 에서의 떡승을 이용한다. 전자에서는 통신자마다 서로 다른 법(modulus)을 가지는 반면에 후자는 모든 가입자들이 동일한 유한체를 사용할 수도 있으므로 실제적인 구현에 효과적이며 안전도면에서도 전자와 별 차이가 없다

그리고 디지털 서명 시스템은 사용자의 상호 인증을 제공하면서 대칭키 암호를 위한 세션키를 분배하는 시스템으로 응용이 가능하다.(McCurley, 1988, ISO/IEC, 1996)

RSA의 경우에는 메세지를 상대방의 공개키로 암호하여 전송할 경우 이를 복호할 수 있는 사람은 비밀키를 가지고 있는 합법적인 사용자 뿐이므로 일방향으로 세션키 분배를 하기에 용이하다. 쌍방이 모두 키 생성에 관여하기를 원한다면 서로의 공개정보를 전송하고 수신한 상대방의 공개정보에 자신의 랜덤수를 떡승하여 D-H형 세션 키의 형성이 가능하다. 또는 공개정보를 상대방의 공개키로 암호하여 전송할 수도 있으며 센타에서 공통의 모듈러스 p 와 원시원 g 를 생성하여 공개하고 이를 이용할 수도 있다.(Kaliski, 1993)

이산대수 문제에 근거한 서명 시스템을 이용하는 방식에는 직접적인 인증은 제공하지 않지만 합법한 사용자만이 옳은 서명을 생성할 수 있다는 가정하에 서명을 검증하여 확인된 정보로부터 세션키를 형성하는 방법과 서명 시스템을 통하여 인증을 행하고 인증과정에 사용된 값들로부터 세션키를 생성하는 방법이 있다. 이들 방법은 대부분 이산대수 문제에 근거한 ElGamal의 방식을

변형한 방식에 적용이 가능하다. 본 연구에서는 기존의 여러 서명시스템을 이용한 키 분배 프로토콜을 살펴보고 그에 대한 문제점을 고찰한다. 그리고 보다 안전한 키 분배 프로토콜을 제안한다.

II. 기존의 키 분배 프로토콜

Arazi(Arazi, 1993)는 DSS를 위한 키 분배 프로토콜을 제시하였다. 하지만 dl 연구에서는 세션키가 한 번이라도 노출될 경우에 안전하지 못함을 보였고(Nyberg, 1994) Lee(Lee, 1996)에서 Arazi의 방식을 개선하여 서명 시스템으로 안전하게 인증된 키 분배를 할 수 있는 방법을 제시하였다. 이를 살펴보면 다음과 같으며 사용되는 파라미터는 서명에 사용된 것과 동일하다. 사용자 A는

(1)랜덤수 k_A 를 선택하고
 $s_{A1} = y_B^{k_A} \mod p$ 와 $s_{A2} = g^{k_A} \mod p$ 를 계산한다.

(2) $s_{A1}' = x_A \cdot s_{A2} + k_A \cdot s_{A3} \mod q$ 를 만족하는 s_{A3} 를 계산한다. 여기서 s_{A1}' 는 $s_{A1} \mod q$ 이다. (3) s_{A1} 과 s_{A3} 를 사용자 B에게 전송한다. 사용자 B는 (1)수신한 s_{A1} 과 s_{A3} 로부터 $s_{A2}' = (s_{A1})^{x_B^{-1}} \mod p$ 를 계산한다.

(2) $g^{s_{A1}'} = (y_A)^{s_{A2}} \cdot (s_{A2}')^{s_{A3}} \mod p$ 가 만족되면 A를 합법한 사용자로 인증한다. (3) A가 인증되면 B도 랜덤수 k_B 를 선택하고

동일한 방법으로 자신을 증명한다. 인증이 완료되면

$$K = (s_{A2}')^{k_B} = (s_{B2}')^{k_A} = g^{k_A k_B} \mod p \text{로 세션키를 형성한다.}$$

이는 통신 요구자가 세션키 생성 정보를 서명식으로 변형하여 전송하면 상대방은 이를 검증하고 자신의 세션키 생성 정보를 변형하여 전송한다. 통신 요구자는 원하는 상대방인지를 수신한 값을 검증함으로 확인하고 교환한 정보로 세션키를 생성한다. 하지만 이 방식은 비밀키외에 검증과정을 위해 자신의 비밀키의 역수를 저장하고 있어야 하는 단점이 있다.

한편 Schnorr의 인증 방식과 디지털 서명을 이용하여 상호인증을 제공하는 효율적인 키 분배 프로토콜이 제시되었다(임 및 이, 1992) 사용자 A는 (1)랜덤수 k_A 로 $T_A = g^{k_A} \mod p$ 를 계산하고 통신 상대방의 ID로 $X_A = T_A \oplus ID_B \mod q$ 를 구하고 X_A 를 B에게 보낸다. (2)사용자 B는 A와 동일하게 k_B 와 T_B 로 (1)의 과정을 수행하고 $E = h(X_A) \oplus h(X_B)$ 를 구하여 $s_B = k_B - x_B \cdot E \mod q$ 를 계산한다. X_B 와 s_B 를 사용자 A에게 전송한다. (3)A는 수신정보 X_B 와 s_B 로부터

$$E = h(X_A) \oplus h(X_B) \text{를 구하고}$$

$$T_B' = g^{s_B} \cdot y_B^E \mod p \text{와}$$

$$X_B' = (T_B' \oplus ID_A) \mod q \text{를 계산하여 수}$$

신한 X_B 와 같은지를 확인한다. (4) 상대방을 확신하면 A는 $s_A = k_A - x_A \cdot E \bmod q$ 를 계산하여 B에게 보낸다. (5) B는

$$T_A' = g^{s_A} \cdot y_A^E \bmod p$$

$X_A' = (T_A' \oplus ID_B) \bmod q$ 를 계산하여 (1)에서 수신한 X_A 와 같은지를 검증한다. 인증이 완료되면

$$T = T_A^{k_B} = T_B^{k_A} = g^{k_A k_B} \bmod p$$

$K = T \bmod q$ 를 구하여 세션키로 사용한다.

위의 프로토콜은 전송정보를 이용하여 공통의 챌린지(challenge)를 형성하여 제 3자의 끼어듦을 막았으며 세션키를 q 로 감소시켜 사용하므로 예전의 세션키가 노출되더라도 다른 세션키의 복구가 불가능하다.

그러나 이는 어떤 시점에 한 사용자의 비밀키가 노출될 경우 전송되었던 정보만 가지고 있다면 예전에 생성되었던 세션키가 모두 노출된다. 앞서 언급한 Lee의 방식은 세션키를 형성한 두 사용자의 비밀키가 모두 노출되면 과거의 세션키를 복구할 수 있다.

그리고 IC카드에 사용에도 사용될 수 있으며 국내 표준(안) 전자서명 시스템을 이용한 키 분배 프로토콜도 제시되었는데 그 수행과정을 보면

(1) 사용자 A는 랜덤수 R_A 를 생성

$$K_A = g^{R_A} \bmod p$$

ID_A , $CERT_A$, Y_A , K_A , j_A 를 단말기에

게 전송한다. (2) 사용자 B는 $CERT^2 = h(j_A, ID_A, Y_A) \bmod n$ 공개 키 인증한 후, 랜덤수 R_B 를 생성

$$K_B = g^{R_B} \bmod p, E_{AB} = h(K_A, K_B)$$

$$S_B = X_B^{-1} \cdot (R_B - E_{AB}) \bmod q$$

계산한다. (3) 사용자 B는 ID_B , $CERT_B$, Y_B , K_B , S_B 를 IC카드에게 전송한다. (4) IC카드는 $CERT^2 = h(j_B, ID_B, Y_B) \bmod n$ 공개 키 인증한 후

$$K_B' = Y_B^{S_B} \cdot g^{E_{AB}} \bmod p = K_B$$

검증하여 단말기의 신분을 인증한다. (4) 서로의 신분이 인증되면 사용자 A는

$$E_{AB} = h(K_A, K_B),$$

$S_A = X_A^{-1} \cdot (R_A - E_{AB}) \bmod q$ 를 사용자 B에게 전송한다. (5) 단말기는

$K_A' = Y_A^{S_A} \cdot g^{E_{AB}} \bmod p = K_A$ 검증하여 사용자 A의 신분을 인증한다. 사용자 B의 경우도 마찬가지다. 그러나 이 방식도 다른 서명시스템을 이용한 키 분배 프로토콜과 같이 랜덤 수가 한 번이라도 노출되면 비밀키가 노출된다.

비밀키가 노출되더라도 과거의 생성되어 사용했던 세션키의 노출을 막을 수 있는 것을 Perfect Forward Secrecy(PFS)라고 한다.(Günther, 1989) 이것은 원래 개인정보에 기초한 방식에서 센타의 비밀키가 노출됨으로 인해 개인의 비밀키가 노출되더라도 과거의 비밀키가 노출되지 않도록 한 것이다. 이는 공개키 확인서에 기반한 방식에서도 마찬가지로 적용이 가능하다.

직접인증으로 세션키를 생성시키는 프로토콜은 대부분 PFS를 만족하지 못한다. 이것은 직접인증을 제공하기 위해 서명식을 이용하게 되는데 챌린지를 공개 전송하거나 공개된 정보로부터 생성시켰을 경우에는 비밀키가 노출되면 전송 정보로부터 사용된 랜덤수가 노출되기 때문이다. 그러므로 챌린지를 프로토콜에 관계된 합법적인 사용자만이 생성할 수 있는 비밀 값으로 사용하면 비밀키가 노출되었을 경우에도 사용되었던 랜덤수를 알 수 없어 과거의 세션키를 복구

할 수 없다.

III. 키 분배 프로토콜 제안

국내표준(안) 서명시스템을 사용한 키 분배 프로토콜은 다음과 같으며 준비단계, 사용자등록단계 그리고 키 분배 프로토콜 순으로 설명한다.

[프로토콜]

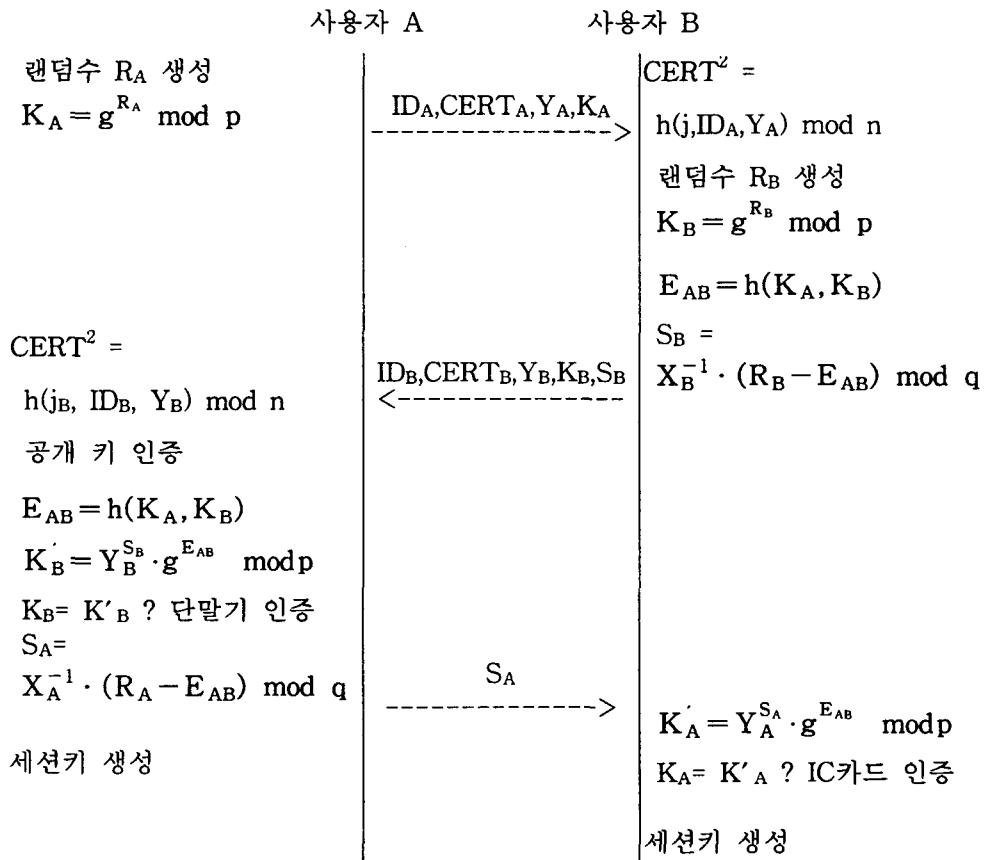


Fig. 1. Proposed Key Distribution Protocol using Digital Signature.

[준비 단계]

- ① 시스템 파라메터 : p 및 $(q | p-1)$ 의 소수 q , 원시원 g : $g = a^{(p-1)/q} \text{ mod } p$, 여기서 $g > 1$ 이고 a 는 $0 < a < p-1$, $n = C \times D$ (C, D 는 큰 소수) - 공개 키 인증용 해쉬함수 h
- ② 사용자 파라메터 : - 비밀 키 X_A , 공개 키 $Y_A = g^{X_A} \text{ mod } p$
- 개인 정보 ID_A

[사용자 등록단계]

- ① 사용자 비밀키 X_A 와 공개키 Y_A ($X_A \in \{1, 2, \dots, q-1\}$, $Y_A = g^{X_A} \text{ mod } p$)를 계산
- ② 공개키 증명서 CERT KAC가 발급 ($\text{CERT}^2 = h(j, ID_A, Y_A) \text{ mod } n$)
- ③ 카드 저장 정보 :

비밀 - X_A 공개 - ID_A , p , q , g , n , Y_A , MC , CV 및 $CERT$

- ▶ 공개 키 인증 : ID_A , $CERT$, Y_A , j
 $\rightarrow \text{CERT}^2 = h(j, ID_A, Y_A) \text{ mod } n$
- ▶ 객체 인증

$$S_A = X_A^{-1} \cdot (R_A - E_{AB}) \text{ mod } q \quad \rightarrow$$

$$K_A = Y_A^{S_A} \cdot g^{E_{AB}} \text{ mod } p$$

제안한 프로토콜에서 전송정보로부터 세션키나 공통의 비밀 챌린지를 알 수 없으며 세션키가 노출되었을 경우 세션키로 부터

비밀키나 랜덤수를 알아 내는 것은 해쉬함수의 역함수를 구하는 문제와 이산대수 문제를 푸는 어려움과 대등하므로 계산상 불가능하다.

랜덤수가 노출되었을 경우 그림 1에서는 비밀키가 노출되면 해당 세션의 키가 노출된다. 그러나 과거에 사용되었던 랜덤수를 알 수 없으므로 여전히 PFS가 만족된다.

그러므로 제안한 프로토콜들은 어떤 시점에 세션 키 생성을 위한 비밀키가 노출이 되더라도 노출되기 이전에 사용되었던 세션키는 복구가 불가능하며 과거의 세션에 통신되었던 내용의 노출을 막을 수 있다. 즉, 비밀키의 노출로 인한 피해를 최소로 줄일 수 있다. 노출 후에는 인증기능과 세션키 생성 기능이 상실되며 비밀키를 바꾸어 사용한다.

IV. 결 론

본 연구에서는 국내표준(안) 서명시스템을 이용한 키 분배 프로토콜을 제안하였다. 제안한 방식은 과거 랜덤수가 알려져도 과거에 생성된 세션 키는 노출되지 않는다. 또한 계산량은 늘었어도 비밀 키가 알려져도 과거에 생성된 세션 키는 안전하다. 따라서 보다 안전하게 세션 키를 생성할 수 있다. 따라서 단지 비밀 키 등은 주기적으로 변경해 줄 필요는 있겠지만 완전한 SM(secure module)가 보장되지 않는 인터넷 등 일반적으로 소프트웨어 만으로 구현

되어 사용되는 환경에서도 이용가능하다. 그리고 실제 통신을 하기 전에 사전에 상대자를 인증할 수 있어 불법 사용자를 차단할 수 있다.

참 고 문 현

M.E.Hellman, "An Overview of Public Key Cryptography," IEEE Comm. Society Mag., Vol.16, No.6, pp.24-32, Nov. 1978.

D.B.Newman Jr., J.K.Omura, and R.L.Pickholtz, "Public-Key management for Network Security," IEEE Network Mag., Vol.1, No.2, pp.11-16, Apr. 1987.

A.Sorkin, "Lucifer, a Cryptographic Algorithm," Cryptologia, Vol.8, No.1, pp.22-35, Jan. 1984.

National Bureau of Standards, *Data Encryption Standard*, U.S. FIPS PUB 46, pp.1-18, 1977.

A.Shimizu and S.Miyaguchi, "Fast Data Encipherment Algorithm FEAL," Eurocrypt'87, pp.267-271, 1987.

X. Lai and J. Massey, "A Proposal for a New Block Encryption

Standard," *Advances in Cryptology - Eurocrypt '90*, pp. 389-404, 1994.

R. L. Rivest, "The RC5 Encryption Algorithm," *CryptoBytes*, pp. 9-11, 1995.

W.Diffie and M.E.Hellman, "New Directions in Cryptography," IEEE Trans. on Inform. Theory, Vol.IT-22, pp.644-654, Nov. 1976.

R.L.Rivest, A.Shamir, and L.Adleman, "On Digital Signatures and Public Key Crpytosystems," Comm. ACM., Vol.21, pp.120-126, Feb. 1978.

K.S.McCurley, "A Key Distribution System Equivalent to Factoring," J. of Cryptology, Vol.1, No.2, pp.95-106, 1988.

ISO/IEC, Information Technology - Security Techniques - Key Management, *ISO/IEC CD 11770*, March 1996.

A. Arazi, "Integrating a Key Cryptosystem into the Digital Signature Standard," *Electronic Letters*, vol.29, pp.966-967, 1993.

N.Nyberg and R.A.Rueppel, "Weaknesses in Some Recent Key Agreement

Protocol," *Electronic Letters*, vol.30,
pp.26-27, 1994.

W.B.Lee and C.C. Chang, "Integrating
Authentication in Public Key Distribution
System," *Information Processing
Letters*, vol.57, pp.49-52, 1996.

임채훈, 이필중, "상호 신분 인증 및 디지털
서명기법에 관한 연구," 통신정보보호학회
논문지, 제2권 1호, pp.16-35, 1992.

C.G.Günther, "An Identity Based
Key-Exchange Protocol," *In Advances in
Cryptology - EUROCRYPT '89*, pp. 29
-37, Springer -Verlag, 1989.