□ 硏 究 論 文 □

韓國軍事運營分析學會誌
第 26 券, 第 1 號, 2000. 6. 30.

Kerberos와 X.509를 이용한 스마트카드 기반 인증시스템에 관한 연구

(A Study on Smartcard-based Certification System using Kerberos and X.509)

박 정용, 남 길현*

Abstract

In this paper, we are introduced a certification system for open network environment. The Kerberos which was developed by MIT uses a secret key cryptosystem for authentication. It is secure and efficient for closed network users to authenticate each others. However, the kerberos has a disadvantage of managing a lot of secret keys for users in the open network environment. This paper suggests a method that uses X.509 to provide public keys with certification to Kerberos users for authentication in the X.500 directory standard. And we also suggest the smartcard as data storage device to enhance the security and availability.

* 국방대학교 국방관리대학원

1. 서 론

컴퓨터 시스템의 네트워크화 추세가 가속되면서 전 세계가 전자 상거래 환경들로 변화하고 있다. 그러나 정보의 공유와 개방을 목표로 개발된 인터넷이 갖고 있는 기본적인 취약성 때문에 거래 내용, 신용카드 정보, 계좌 번호, 혹은 관련 비밀 번호 등의 정보들이 쉽게 노출될 수 있다[2].

이러한 전자상거래의 위협요소들은 시스템과 네트워크에서의 위협요인들과 밀접한 관계가 있다. 전자상거래시스템은 기존의 다른 응용 시스템과는 보안 요구사항의 핵심이 현저히 다르다. 기존 응용 시스템은 데이터와 시스템 자원에 대한 사용자의 접근 통제 및 시스템 이용에 대한 자료의 관리를 근간으로 하는데 반해서 전자상거래시스템에서는 앞에서 기술한 것 이외에도 사용자 실체에 대한 증명과 데이터 내용에 대한 사후 검증이 필요하게 된다[5]. 그러므로 전자상거래 네트워크 환경에서 메시지를 교환하는 상인과 소비자의 적법성을 확인하는 인증 절차는 반드시 필요한 부분이다. 이에 따라 기존의 인증시스템보다 사용자의 편의성 및 보안성이 강화된 개선된 인증시스템의 개발이 시급한 일이다.

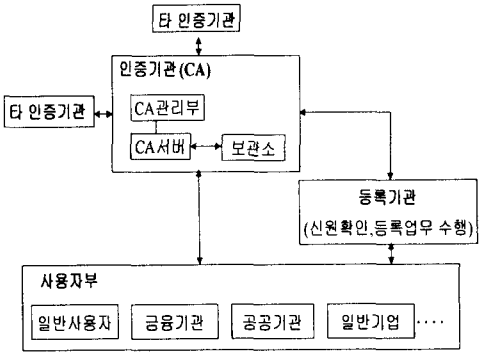
한편, 인증프로토콜의 대부분이 공개키 암호 방식을 사용하는데 반해, Kerberos는 관용 암호 방식을 사용하고 있으며, 두 가지 버전의 Kerberos가 사용되는데 가장 많이 사용되는 버전은 Kerberos v4이고, 이를 수정한 v5가 인터넷 표준(RFC 1510)으로 1994년에 발표되어 사용되고 있다[3].

본 논문에서는 Kerberos가 관용 암호 방식을 사용하기 때문에 외부 영역에 대한 상호 인증을 할 때 비밀키를 사전에 분배되어 있어야 하는데 이것이 많은 영역이 있을 경우에는 키 관리하는 문제점이 발생하게 된다. 이를 해소하기 위해서는 키 관리가 용이한 공개키 암호 방식을 사용해야 되는데 디렉토리 인증 서비스 표준 프로토콜인 X.509를 사용하여 Kerberos와 접목시킨다면 많은 외부 영역에 대한 키 관리 문제점을 해소할 수가 있다. 또한 여기에 보안성이 우수한 스마트 카드를 이용하여 X.509인증서를 저장하여 사용자인증정보를 안전하게 보호할 수 있는 매체로 활용하여 안전한 인증시스템을 구축할 수 있도록 하였다.

2. 인증시스템 기반 기술

2.1 인증시스템의 기본구조

인증시스템의 기본구조는 (그림 1)과 같이 사용자부, 인증기관, 등록기관, 타인증기관으로 구성된다[8][10][11].



(그림 2) 인증시스템 기본구조

1) 사용자부 : 인증기관에 사용자 등록을 사전에 하였고 인증서를 신청하여 발급받아 사용한다.

2) 인증기관(CA: Certificate Authority) : 사용자에 대한 인증서 발급 및 인증업무를 수행한다.

3) 등록기관 : 사용자의 신원확인 및 등록 업무를 수행한다. 인증서는 발행하지는 않으며 필요에 따라서 구현되지 않을 수도 있다.

4) 타인증기관 : 인증시스템의 수평 또는 수직적인 관계를 맺고 있는 인접 CA부분에 해당한다. 타 CA를 인증하는 역할을 하며 또 다른 CA와 연결될 수 있다.

2.2 Kerberos 인증 프로토콜

2.2.1 Kerberos v5 인증프로토콜

Kerberos v5는 6단계의 메시지 교환절차를 거쳐서 인증서비스를 한다[4].

o 인증 서비스 교환 단계

1) C → AS : Options, ID_c, $Realm_c$, ID_{tgs} Times, Nonce1

2) AS → C : ID_c, $Realm_c$, $Ticket_{tgs}$, EK_c [$K_{c,tgs}$, Times, Nonce1, ID_{tgs}, $Realm_{tgs}$]

$Ticket_{tgs}$ = EK_{tgs}[Flags, $K_{c,tgs}$, ID_c, $Realm_c$, AD_c, Times]

o 티켓-승인 서비스 교환 단계

3) C → TGS : Option, ID_s, $Ticket_{tgs}$, Times Authenticator$_c$, Nonce2

4) TGS → C : ID_c, $Realm_c$, $Ticket_s$, $EK_{c,tgs}$ [$K_{c,s}$, Times, Nonce2, ID_s, $Realm_s$]

$Ticket_{tgs}$ = EK_{tgs}[Flags, $K_{c,tgs}$, ID_c, $Realm_c$, AD_c, Times]

$Ticket_s$ = EK_s[Flags, $K_{c,s}$, ID_c, $Realm_c$, AD_c, Times]

Authenticator$_c$ = $EK_{c,tgs}$[ID_c, $Realm_c$, TS1]

o 클라이언트/서버 인증 교환 단계

5) C → S : Options, $Ticket_s$, Authenticator$_c$

6) S → C : $EK_{c,s}$[TS2, Subkey, Seq#]

$Ticket_s$ = EK_s[Flags, $K_{c,s}$, ID_c, $Realm_c$, AD_c, Times]

Authenticator$_c$ = $EK_{c,s}$[ID_c, $Realm_c$, TS2, Subkey, Seq#]

2.2.2 Kerberos v5 인증프로토콜 문제점

N개의 외부 영역과의 상호 인증 서비스가 곤란하다. 즉, 하나의 Kerberos 영역이 다른 Kerberos 영역과 상호 인증하기 위해서는 반드시 [N(N-1)]/2개의 비밀키가 교환되어야 한다. 이는 개방 네트워크 환경에서는 불가능하다 하겠다[1].

2.3 X.509 디렉토리 인증서

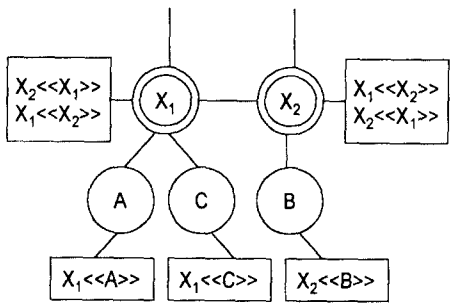
2.3.1 X.509 인증서

X.509 디렉토리 인증서는 1988년 CCITT (현재의 ITU-T)에 의해 초기 버전이 공표되었고 1993년 개정되었다. 이를 1995년 ISO와 IEC는 X.509(1993)를 디렉토리 인증 표준 ISO/IEC 9594-8[ISO95]로 제정하여 국제적 표준으로 인정하고 있다[3][6]. X.509 구조의 핵심은 각 사용자와 연관된 공개키 인증서이다. 이 인증서는 어떤 신뢰할 만한 인증기관(CA: Certification Authority)에 의해 발행되어 CA나 사용자에 의해 디렉토리에 위치하는 것으로 가정된다. 디렉토리 서버 그 자체는 공개키의 생성이나 인증 기능에 대한 책임이 없으며, 단지 사용자가 인증서를 쉽게 얻을 수 있는 접속 장소를 제공할 뿐이다[7][9].

2.3.2 인증서 획득 절차

X.509인증서를 정의하기 위해서는 다음의 표기법이 사용된다[4].

X<<A>> ; '인증기관 X에 의해서 발행된 사용자 A의 인증서'란 의미이다. (그림 2)를 보면 연결된 원은 인증기관들 사이의 계층적인 구조를 나타내며, 연관된 네모들은 각 인증기관 엔트리에 대하여 디렉토리에 유지되고 있는 인증서들을 나타낸다. 만약 A가 인증기관 X_1으로부터 인증서를 획득하고 B가 인증기관 X_2로부터 인증서를 획득했다고 가정하자.



(그림 3) X.509 인증기관 계층 구조

만일 A가 X_2의 공개키를 알지 못한다면, X_2에 의해 발행된 B의 인증서는 A에게 아무 소용이 없다. 즉, A는 B의 인증서를 볼 수는 있지만 B의 전자서명을 확인할 수는 없다. 그러나, 두 인증기관이 안전하게 자신들의 공개키를 상호 교환한다면 다음의 절차가 A로 하여금 B의 공개키를 획득할 수 있도록 한다.

1) A가 X_1에 의해 서명된 X_2의 인증서를 디렉토리로부터 획득한다. A는 X_1의 공개키를 알기 때문에 인증서로부터 X_2의 공개키를 얻을 수 있고 인증서에 있는 X_1의 서명을 사용하여 X_2임을 확인 할 수 있다.

2) 그 다음 A는 X_2에 의해 서명된 B의 인증서를 획득할 수 있다. A는 X_2를 인증할 수 있는 복제된 공개키를 얻을 수 있기 때문에 서명을 확인할 수 있고 B의 공개키를 안전하게 획득할 수 있다.

A가 B의 공개키를 획득하는 과정을 보면 다음과 같은 인증서의 체인을 사용한다.

Forward certificate : X_1<<X_2>>X_2<<B>>

동일한 방법으로 B는 역방향 체인을 사용하여 A의 공개키를 획득한다.

Reverse certificate : X_2<<X_1>>X_1<<A>>

이러한 체인은 두 개의 인증서에 제한되는 것이 아니고 다음과 같이 N개 요소로 구성된 체인을 생성할 수도 있다.

X_1<<X_2>>X_2<<X_3>>...<<X_N>>B

이 경우에 체인(Xi, Xi+1)의 각 CA쌍은 서로가 상호 인증서를 갖고 있어야 한다.

2.4 스마트카드

스마트카드는 컴퓨터 등의 기기와 인터페이스를 통하여 통신이 가능하고 자체프로세서를 내장하고 있어 가치저장 및 연산이 가능하며 메모리의 용량 및 안정성이 우수한 기록매체로 단순히 EEPROM을 내장하여 메모리 능력만 향상시킨 메모리카드와 구분된다[2].

본 논문에서는 다음의 보안 기능 때문에 스마트카드를 제안하는 인증시스템에 점목하였다. 1) 카드자체의 위조가 불가하도록 특수한 재질과 모양으로 설계되도록 하였고 카드의 외부에는 디지털 사진기로 찍은 사용자의 사진을 부착하며 입체사진술(holography)을 이용하여 복제가 어려운 문양이나 표식을 넣을 수 있다. 2) 내부 IC는 위조방지장치

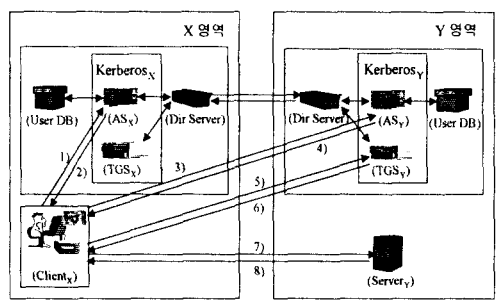
를 이용, 복제나 위조 노력 시에는 스스로 소멸하는 기능(Kill Bit Logic)이 가능하며 외부에서 주파수나 광학으로 판독을 방지하는 장치 등으로 IC칩 내부를 분석할 수 없도록 하고 있다. 3) 메모리 접근통제를 하여 미리 결정된 조건이 합치할 경우에만 스마트 카드가 데이터 메모리에 접근을 허용할 수 있다. 4) 각 파일은 지정된 키에 의해서만 접근이 가능하며 그 파일내용을 갱신할 권한이 있는 인증기관에서만 해당키를 운용하여 관리할 수 있다. 5) 카드의 보안성을 유지하기 위하여 암호키나 PIN은 내부에서만 읽을 수 있도록 비밀영역에 저장되어 있고 제시될 때마다 내장된 칩 운영체제(Chip Operating System)에 의하여 감시되도록 하고 있다. 6) 카드의 분실로 인하여 불법 사용자에 의한 부정사용을 방지하고자 3번의 회수이상 잘못된 비밀번호를 입력할 때에는 스마트카드가 정당한 사용자가 아님을 인식하고 카드를 Lock상태로 돌입하게 하여 이후에는 사용이 불가능하도록 할 수 있다[12].

3. 개선된 인증시스템 제안

3.1 제안 알고리즘

제안하는 인증시스템의 특징은 클라이언트와 서버의 환경에서 인증 서버(AS) 및 티켓승인서버(TGS)를 두어 다단계 인증 서비스를 제공하는 메커니즘을 제공하고 있다. 즉, Kerberos의 다단계 인증 서비스 장점을 활용하였다. 그리고 Kerberos v5에서 메시지 교환 시 $Realm_c$가 있었는데 제안하는 알고리즘에는 필요가 없다. 이는 X.509를 Kerberos와 접목하면서 디렉토리서버가 새로 필요하게 되었는데 이것이 $Realm_c$의 역할을 대신하여 영역들의 구분을 해주며 외부 영역과의 상호 연결을 해주기 때문이다. 또 다른 특징은 X.509디렉토리 인증표준을 이용하여 인증서비스 교환단계에서 패스워드의 평문 전송없이 상대방의 공개키로 암호화해서 전송하여 인증 서버로부터 인증을 받을 수 있도록 하였다. 이는 패스워드 전송에 대한 도청 및 가로채기 위협을 방어하게 해준다. 그리고 스마트카드의 접목으로 제안한 시스템의 보안성을 강화시켰다. 또한 카드를 위조하기 위해서는 인증기관의 인증서를 위조해야하기 때문에 안전하다고 할 수 있다.

(그림 3)은 $Client_X$와 Y영역의 서비스서버사이에서 인증 정보가 교환되는 순서를 제안한 것이다.



(그림 3) 제안 인증시스템의 전체 흐름도

1) 클라이언트에 로그인한 다음 이용할 서비스서버(예: 데이터베이스 자료, E-mail 서버, 웹 서버 등)를 요구하면, 클라이언트는 메시지를 AS_X로 보낸다. 메시지에는 요구하는 티켓의 플래그 값에 대한 옵션과 사용자 ID, TGS_Y ID가 포함되며 클라이언트와 인증서버간의 동조를 표시하기 위한 현재시간과 앞으로 돌아오는 메시지가 불법적으로 재 전송되지 않았다는 것을 나타내기 위한 임시비표를 같이 보낸다. 메시지를 받은 AS_X는 $Ticket_{TGS_Y}$을 요청한 $Client_X$가 현재 유효한 사용자인지 User DB에서 접근 권한을 검색하여 적법성을 확인한다. 클

라이언트가 인증받은 사용자라면 AS_X는 $Client_X$가 요구한 TGS_Y의 영역을 디렉토리 서버를 이용하여 검색한다. 디렉토리 서버는 영역 검색이 끝나면 $Client_X$가 요구한 TGS_Y를 담고 있는 영역 Y의 디렉토리 서버와 전방 인증 체인을 생성한다. 반대로 영역 Y의 디렉토리 서버는 공개키 PK_Y를 포함한 후방 인증 체인을 영역 X의 디렉토리 서버로 전송한다.

○ $Client_X$ → AS_X : 메시지①([Option], ID_c, ID_{TGS_Y}, Times, Nonce1)

2) Y영역 디렉토리 서버로부터 공개키 PK_Y를 획득한 AS_X는 다음 메시지를 보낸다. 메시지에는 사용자의 ID와 패스워드(K_C)로 암호화한 Y영역의 공개키(PK_Y)와 메시지 ①에서 보낸 시간 및 임시비표, Y영역의 TGS_Y ID가 $Client_X$로 보내진다. 이때 $Client_X$는 PK_Y를 획득하는 것이다.

○ AS_X → $Client_X$: 메시지②(ID_c, $EK_c(PK_Y$, Times, Nonce1, ID_{TGS_Y}))

3) $Client_X$는 AS_Y에게 메시지를 보내는데 여기에는 요구하는 티켓-승인티켓의 플래그에 대한 옵션과 사용자 ID, TGS_Y ID가 포함되며 클라이언트와 AS_Y간의 동조를 표시하기 위한 현재시간과 앞으로 전송되는 메시지는 불법적으로 재 전송되지 않았다는 것을 나타내기 위한 임시비표를 상대방의 공개키 PK_Y로 암호화하여 전송한다

○ $Client_X$ → AS_Y : 메시지③(EPK_Y([Option], ID_c, ID_{TGS_Y}, Times, Nonce2))

4) AS_Y는 $Client_X$가 보낸 메시지 ③을 자신의 비밀키로 복호화하여 알아낸 다음 $Client_X$에게 사용자의 ID, AS_Y와 TGS_Y간에 공유하는 비밀키(K_{TGS_Y})로 암호화된 $Ticket_{TGS_Y}$, $Client_X$와 TGS_Y사이에서 사용할 세션키(K_{C,TGS_Y}), TGS_Y ID, 티켓-승인티켓이 발행된 시간, 메시지 ③에 있는 임시 비표를 상대방의 공개키 PK_X로 암호화하여 전송한다. 이때 $Ticket_{TGS_Y}$은 AS_Y와 TGS_Y간의 비밀키(K_{TGS_Y})로 암호화되었기 때문에 $Client_X$는 $Ticket_{TGS_Y}$의 내용을 알 수 없다.

○ AS_Y → $Client_X$: 메시지④(ID_c, $Ticket_{TGS_Y}$, EPK_X(K_{TGS_Y}, ID_{TGS_Y}, Times, Nonce2))

$Ticket_{TGS_Y}$ = EK_{TGS_Y}([Flags], K_{C,TGS_Y}, ID_c, AD_c, Times)

5) $Client_X$는 AS_Y가 보낸 메시지 ④를 자신의 비밀키로 복호화한 다음 TGS_Y에게 메시지를 보낸다. 메시지에는 다음과 같은 내용이 있다. 요구한 서비스-승인티켓의 플래그에 대한 Options과 암호화된 $Ticket_{TGS_Y}$, 그리고 암호화된 인증자 및 서비스-승인티켓 요구한 현재시간과 임시비표, 서비스서버 ID가 포함된다. $Ticket_{TGS_Y}$은 메시지 ④에서 보낸 것과 동일한 것이며 인증자는 사용자가 본인을 인증하기 위해 보내는 인증정보로서 사용자의 ID와 인증자가 생성된 시간을 나타내는 타임스탬프를 포함하고 있으며 이들은 클라이언트와 TGS_Y간의 세션키(K_{C,TGS_Y})로 암호화된다.

○ $Client_X$ → TGS_Y : 메시지⑤([Option], $Ticket_{TGS_Y}$, $Authenticator_c$, Times, Nonce3, ID_{S_Y})

$Ticket_{TGS_Y}$ = EK_{TGS_Y}([Flags], K_{C,TGS_Y}, ID_c, AD_c, Times)

Authenticator$_c$ = EK_{C,TGS_Y}(ID$_c$, TS1)

6) 메시지 ⑤를 받은 TGS$_Y$는 AS$_Y$와 공유하는 비밀키(K_{TGS_Y})를 가지고 TGT$_Y$을 복호화하고, TGT$_Y$ 안에 포함된 세션키(K_{C,TGS_Y})를 가지고 인증자 안에 포함된 사용자의 ID를 TGT$_Y$안에 있는 정보와 비교해 본다. 모든 것이 일치한다면 TGS$_Y$는 TGT$_Y$을 보낸 사람이 실제 소유자라는 것을 알 수 있다. 그러면 TGS$_Y$는 다음과 같은 메시지를 보낸다. 메시지에는 사용자의 ID, $Ticket_{S_Y}$과 세션키(K_{C,TGS_Y})로 암호화 된 클라이언트와 S$_Y$간의 세션키(K_{C,S_Y}), $Ticket_{S_Y}$ 발행된 시간, 임시비표, S$_Y$ ID 등을 전송한다. 이때 $Ticket_{S_Y}$에는 메시지 ⑤의 옵션에 대한 플래그 값과 세션키(K_{C,S_Y}), 사용자 ID 및 사용자 네트워크 주소와 $Ticket_{S_Y}$이 발행된 시간이 TGS$_Y$와 S$_Y$간의 비밀키(K_{S_Y})로 암호화된다. $Ticket_{S_Y}$은 TGS$_Y$와 S$_Y$가 공유하는 비밀키(K_{S_Y})로 암호화하기 때문에 클라이언트는 $Ticket_{S_Y}$의 내용을 읽을 수 없다.

○ TGS$_Y$ → Client$_X$: 메시지⑥(ID$_c$, $Ticket_{S_Y}$, EK_{C,TGS_Y}(K_{C,S_Y}, Times, Nonce3, ID_{S_Y}))

$Ticket_{S_Y}$ = EK_{S_Y}([Flags], K_{C,S_Y}, ID$_c$, AD$_c$, Times)

7) 메시지 ⑥을 받은 클라이언트는 세션키(K_{C,TGS_Y})를 사용하여 복호화한 다음 S$_Y$에게 옵션과 암호화된 $Ticket_{S_Y}$, 그리고 인증자를 포함한 메시지를 보낸다. 이때 메시지의 옵션은 돌아오는 티켓의 플래그 값이 아니라 클라이언트가 상호 인증이 필요하다고 요구하는 옵션으로 만약 상호 인증이 필요하다면 인증자에는 Subkey , Sequence number 필드가 포함된 Authenticator$_c$가 추가된다.

○ Client$_X$ → Server$_Y$: 메시지⑦([Option], $Ticket_{S_Y}$, Authenticator$_c$)

$Ticket_{S_Y}$ = EK_{S_Y}([Flags], K_{C,S_Y}, ID$_c$, AD$_c$, Times)

Authenticator$_c$ = EK_{C,S_Y}(ID$_c$, TS2, [Subkey, Seq#])

8) 메시지 ⑦를 받은 S$_Y$는 TGS$_Y$와 공유하는 비밀키(K_{S_Y})를 이용하여 $Ticket_{S_Y}$을 복호화하고, $Ticket_{S_Y}$안에 포함된 세션키(K_{C,S_Y})를 이용하여 인증자를 복호화한다. 그런 다음 인증자 안에 포함된 사용자 ID를 $Ticket_{S_Y}$안에 포함된 정보와 비교한다. 일치한다면 S$_Y$는 $Ticket_{S_Y}$을 보낸 사람이 인증된 자라는 것을 알 수 있다. 결국 클라이언트와 S$_Y$는 세션키(K_{C,S_Y})를 공유하며 이는 이후 세션동안에 주고받는 메시지를 암호화하는데 사용되거나 새로운 랜덤 세션키를 교환하는데 사용될 수 있다. 선택적인 Seq# 필드는 클라이언트가 사용하는 시작 순서 번호를 지정하여 메시지의 재전송을 알려준다.

○ Server$_Y$ → Client$_X$: 메시지⑧(EK_{C,S_Y}(TS2, [Subkey, Seq#])

최종적으로 클라이언트가 동작하는 것을 알아보면, 클라이언트는 S$_Y$가 보낸 메시지 ⑧를 세션키(K_{C,S_Y})로 복호화한다. 메시지 ⑧이 세션키(K_{C,S_Y})로 암호화되어 있기 때문에 클라이언트는

오로지 S_Y에 의해서 만들어졌을 것이라고 확신한다. 이제 클라이언트와 S_Y간의 외부 영역 상호 인증이 끝났으므로 둘 사이에 서비스 요구와 서비스 응답 형태의 메시지 교환이 이루어 질 수 있다. 만일 이 메시지의 암호화가 필요하다면 세션키(K_{C,S_Y})를 직접 사용하거나 새로운 암호화용 랜덤 세션키를 생성하여 사용하는데, 새로운 랜덤 세션키는 클라이언트와 서비스서버간의 세션키(K_{C,S_Y})로 암호화되어 교환된다.

기본적인 Kerberos의 인증서비스 교환단계에서는 전송데이터가 평문상태로 전송되지만 본 알고리즘에서는 개방형네트워크환경에서의 보안성을 보장하기 위해서 이 전송데이터를 상대방의 공개키로써 암호화하여 보안성이 보장되는 인증시스템을 설계하였다.

3.2 KXS인증카드 내부정보설계

본 논문에서 제안하는 인증시스템에서 사용하는 스마트카드를 KXS인증카드라고 하며 제안하는 KXS인증카드의 내부 정보를 묘사하기 위한 표현방법은 다음과 같다.

- o A, C, AS : 인증기관, 사용자, 인증서버
- o ID_c, ID_{AS}, ID_{tn}, CID_c : 사용자ID, 인증서버ID, 단말의 ID, KXS카드 ID
- o PK_A, PK_c : 인증기관, 사용자의 공개키
- o K_{AC} : 인증기관-사용자간 공통키
- o 키 K에 의한 메시지M(암호문X)에 대한 암호화 EK(M), 복호화 DK(X)
- o Cert(A,B) : A에서 B의 공개키를 인증하는 인증서

인증기관에서 사용자에게 KXS카드를 발행시에는 EEPROM에 각종 암호키, 인증서 및 데이터를 입력하여 발행하는데 그 내용은 [표 1]과 같으며 이 모든 정보는 인증기관을 통해서만 입력되고 갱신될 수 있도록 한다.

전자서명을 위한 RSA알고리즘 및 해쉬함수는 ROM과 보조프로세서(Co-Processor)에서 작동한다. 사용자의 개인키, PIN, 사용되는 암호프로그램은 카드의 인증 및 전자서명에 사용되는 KXS카드의 핵심부분으로서 카드내부에서만 동작가능하고 외부로 유출될 수 없도록 보호된다. 인증서Cert(A,C)는 인증기관에서 키 길이 2048비트의 RSA를 사용하여 서명한다. 또한, 인증서에 사용자 IDc와 카드의 고유번호 CIDc를 동시에 포함하여 카드의 고유번호와 틀린 CIDc가 입력되는 것을 방지하여 카드위조가 불가능하게 한다.

[표 1] 사용자 KXS카드 정보

영역	표현	설명(사용용도)	방법
비밀영역	K_C PIN	사용자의 개인키(2048 비트) 사용자 식별 번호(Personal ID Number) : 비밀번호에 해쉬함수 적용	갱신및접근불가
접근통제	Trans No PK_A Cert(A,C)	서명시마다 1씩 올라가는 트랜잭션번호 인증기관 공개키 인증기관에서 사용자를 인증한 인증서 $(ID_c, CID_c, PK_c, Edate)_{SKc}$: 사용자, 카드, 사용자의 공개키, 인증서 유효일에 대한 인증서	
공개영역	개인정보	개인이름, 주민등록번호, 병역사항, 주소, 본적, 호주, 세대주, 사진 등	읽기가능

3.3 제안한 KXS인증시스템 평가 및 분석

본 논문에서 제안한 KXS인증시스템을 보안성, 효율성, 사용자 편의성측면에서 평가 및 분석해보면 다음과 같다.

3.3.1 보안성

본 논문에서 제안된 프로토콜은 Kerberos와 X.509 프로토콜 및 스마트카드에서 보장해 주는 보안성에 기반을 두고 있다.

Kerberos에서는 인증 서버 및 티켓승인서버를 두어 다단계 인증 서비스를 제공하는 메커니즘을 제공하고 있는데 이는 인증서비스 교환단계에서 패스워드의 평문 전송없이 상대방의 공개키로 암호화하여 인증 서버로부터 인증을 받기 때문에 패스워드 전송에 대한 도청 및 가로채기 위협을 방어할 수 있다. 그리고 X.509에 의해 얻게되는 공개키는 소인수분해의 난점을 이용한 RSA 방식을 사용하였는데 키의 길이가 2048비트로 이는 약 10^{616}자리 수를 사용하는 것인데 이를 분리해 내는 일은 현재의 소인수 분해 알고리즘으로는 불가능하다고 할 수 있다. 마지막으로 스마트카드의 보안성은 인증기관에서 발행하는 인증서에 영향을 받는데 이는 공개키의 길이가 X.509처럼 2,048비트로서 인증기관의 인증서를 위조하는 것은 불가능하기 때문에 안전하다고 볼 수 있다.

3.3.2 효율성

본 논문에서 제안한 KXS인증시스템은 인증프로토콜 Kerberos의 단점을 해소하였다. 즉, 이는 많은 수의 Kereberos인증시스템 외부 영역 사용자 및 서비스서버의 인증을 하는데 있어서 관용 암호 방식으로 하기 때문에 비밀키 관리의 문제점이 있었는데 이는 N개의 외부 영역에 대하여 사전에 [N(N-1)]/2개의 비밀키가 교환되어야 하는 것으로, 만약 외부 영역의 갯수가 10,000개라면 비밀키가 49,995,000개가 필요한 것을 말한다. 이런 문제점을 해소하기 위해서 외부 영역에 대한 디렉토리 관리를 할 수 있는 디렉토리 서버를 두고 디렉토리 표준프로토콜인 X.509의 인증서 형식을 이용하여 공개키 암호 방식을 타 영역과의 인증시 사용하여 10,000개의 비밀키만 관리할 수 있도록 하여 많은 수의 Kerberos 사용자가 인증시스템을 이용할 수 있도록 하였다.

3.3.3 사용의 편의성

제안한 인증시스템은 실세계 및 인터넷에서 인증 정보를 교환 할 시 편리하게 사용될 수 있다. 즉 실세계에서 주민등록증과 같이 본인의 카드라는 인증을 위하여 본인의 사진과 주민등록번호, 주소 등이 명시되어 있다면 다른 이가 분실된 것을 재사용할 수 없도록 할 것이다. 또한 신용카드와 같이 물리적 매체를 제공하여 휴대하기가 간편하고 다른 용도로의 전환이 가능하다.

4. 결론 및 향후 연구 방향

전자상거래 보안의 기본요건인 보안성과 신뢰성을 충족시키기 위해서는 인증시스템의 개발이 필수적이어야 한다. 본 논문에서는 이러한 요구조건을 충족시킬 수 있는 인증시스템을 제안하였다. 제안한 인증시스템은 공인 인증기관과 사용자간의 스마트카드로 이어지는 강력한 인증 사슬로서 보다 강화된 인증 기능을 제공하며 사용자들이 편리하게 사

용할 수 있도록 카드를 사용하였다. 제안하는 인증 시스템의 안전도는 2048비트의 RSA공개키 암호를 사용하여 강력하게 인증을 지원하며 스마트카드에 있는 위조방지장치를 활용할 수 있으므로 사용자 인증정보를 안전하게 보호할 수 있다.

한편, 전자상거래가 아직은 우리 나라가 초창기에 지나지 않아 인증 시장의 규모가 크지 않지만 앞으로는 전자상거래가 보다 더 확산될 것으로 예상하고 있다. 이에 우리는 좀 더 안전한 전자상거래 환경 구현을 위해서는 암호화를 고속으로 수행할 수 있는 스마트 카드의 연구를 통해 보안성이 강화된 인증시스템이 더 연구해야 할 과제이다.

참 고 문 헌

[1] Bruce Schneider, Applied Cryptography, John Wiley & Sons, Inc, 1994, pp.417-424.

[2] Charles P.Pfleeger, Security in Computing, Prentice-Hall International, Inc, 1997, pp.254-264.

[3] R.Housley and W.Ford, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", Internet RFC 2459, January 1999, pp.5-41.

[4] S.M.Bellovin and M.Merritt, Limitations of the kerberos authentication system, Computer Communication Inc, 1990, pp.119-132.

[5] 김지홍, "공개키 기반구조(PKI)", 한국통신정보보호센터, NETSEC-KR'99, 1998.6, pp.9-32.

[6] 류춘열, 이경현, 박지환, 암호이야기, 동영출판사, 1993, pp.119-167.

[7] 염흥렬, "인증서/CRL규격과 생성 기법", 한국통신정보보호학회, NETSEC-KR'99, 1999.5, pp.19-48.

[8] 최용락, 소우영, 이재광, 이임영, 통신망정보보호, 그린출판사, 1996, pp.343-372.

[9] 홍기용, "인증관리센터 구축 및 운영계획", 한국정보보호센터, SIS'99정보보호심포지움, 1999.4, pp.33-112.

[10] http://www.cse.dnd.ca/cse/english/gov.html

[11] http://www.verisign.com/cpsch2.htm

[12] http://grus.hkstar.com/~alanchan#smartcard