特輯

키 분배와 인증 기법

高承哲, 李相珍
韓國電子通信研究所

요약문

본고에서는 키 분배 및 인증과 관련된 기본적 사항들을 간단히 고찰 소개한다. 키 분배 방식을 구성하는 주요 요소, 기본적인 키 분배 방식과 인증 기법 및 그 기법의 안전성, 인증과 키 분배를 동시에 수행하는 프로토콜과 UNIX 망에서 실질적으로 서비스되는 Kerberos 프로토콜을 소개한다.

본고에서는 키 분배 및 인증과 관련된 기본적 사항들을 간단히 고찰 소개하고자 한다.2장에서는 키 분배 방식의 주요 구성요소들을 소개하며, 3장에서는 기본적인 키 분배 방식을 소개하며, 4장에서는 여러 종류의 인증 기법과 그 기법들의 안전성을 소개하며, 5장에서는 인증과 키 분배를 동시에 수행하는 프로토콜과 UNIX 망에서 서비스되는 Kerberos 프로토콜을 소개한다

Ⅰ. 도 입

암호 기술을 기반으로 하는 정보 보호 서비스는 정보 보호의 핵심인 암호 알고리즘에서 사용되는 키가 송.수신 양측에 안전하게 분배될 수 있다는 가정하에 기본적인 기능을 제공할 수 있다. 그러나 실제로 복잡한 망을 대상으로 키를 안전하게 분배하는 일은 매우 어려운 일이며, 비록 암호 알고리즘이 매우 안전하다고 하여도 키 분배 과정 또는 키를 송신한 상대방의 신분 확인 과정에서 안전성이 결여된다면 사용자들이 목적하는 안전한 통신 서비스를 제공할 수 없게 된다.

키 분배의 문제점을 해결하기 위하여 공개 키 암호 알고리즘의 개념이 소개되었으며, 실질적으로 충분한 안전성을 보장할 수 있도록 구현에 관한 연구가 지난 20년간 진행되어 왔다. 그러나 구현상의 난이점, 효율성, 가격상의 문제점으로 인하여 여전히 비밀 키 암호 알고리즘을 기반으로 하는 키 분배 방식을 구현한 S/W들이 실질적인 데이타 통신망에서 널리 사용되고 있다.

Ⅱ. 키 분배 과정의 구성 요소

키 분배 과정은 키 분배와 관련된 정보를 암호화하는 암호 알고리즘, 위조를 방지하는 위조 식별코드, 반복하여 타인을 가장하는 Replay 공격을 방지하는 Replay 식별코드, 송.수신자가 서로를 인증하는 인증코드로 구성된다.

키 분배와 관련된 정보를 암호화하는 알고리즘의 안전성이 보장되어야 하며, 또한 비밀 키, 공개 키 등 알고리즘의 종류에 상관없이 키 분배와 관련된 정보를 암호화하는 키, 즉 마스타 키가 사전에 송.수신 양측에 분배되어야 하며, 이러한 기능을 일반적으로 키 관리 센타가 수행한다.

위조 방지 코드는 정당한 수신자가 통신로 상으로 키 분배와 관련된 정보에 위조가 발생된 지를 검증하는 코드이다. 일반적으로 일방향 함수에 의해 발생된 해쉬값을 사용하며, 메세지 인증 코드(Message Authentication Code, MAC)이라고 불린다. 일방향 함수로는 주로 미국 표준 암호 알고리즘 DES가 사용된다.

Replay 식별코드는 키 분배 과정에서 제3자가 개입하여 정당한 송.수신자를 가장하여 키를 가로채는 공격방식을 방지하는 코드이며, 송.수신 인증 코드는 송.수신자가 서로 상대방이 정당한 통신 상대인가를 식별하는 코드이다. 이러한 코드는 송.수신자 인증 프로토콜에 의해 결정되며, 주로 time stamp, 랜덤수, 사용자의 Identity등이 사용된다.

Ⅲ. 키 분배 기법(KeyDistributionTechniques)

암호시스템을 구현하기 위하여 필요한 기술 중 가장 난해한 기술이 바로 송수신 양측의 암.복호화에 사용되는 세션 키(session key)를 공유할 수 있도록 안전하고 효율적으로 키를 분배하는 기술이다. 일반적으로 송신자가 세션 키를 랜덤하게 생성, 암호화하여 수신 측에 전송한 후, 수신자가 이를 복호화하여 동일한 키를 송.수신 양측에서 공유하게 된다. 세션 키 암호화에 사용되는 키를 마스타 키(Master key)라고 정의하며,이 마스타 키는 일반적으로 통신망 관리자가 각 사용자들에게 안전한 전송 수단을 이용하여 사전에 분배한다. 세션 키 보호용 알고리즘의 종류에 따라, 키 분배 기법을 대칭형 암호에 의한 키 분배기법, 공개 키 암호에 의한 키 분배 기법으로 분류할 수 있다. 이제 여러 종류의 키 분배기법을 간략히 고찰한다.

1. 키 분배 센타에 의한 키 분배 방식

1) 송신자 A는 키 분배 센타(Key Distribution Center, KDC)에 B와 통신할 수 있도록 세션 키를 부여해 달라고 요청한다.

2) KDC는 먼저 랜덤한 세션 키를 생성한 후, A와 B의 마스타 키로 각각 암호화하여 A에게 전송한다. 이때 A의 Identity에 관한 정보도 역시 B의 키로 암호화하여 전송한다.

3)A는 KDC로부터 수신한 세션 키에 관한 정보를 복호화하며, 동시에 B에게 B의 마스타 키로 암호화된 세션 키와 A의 Identity에 관한 정보를 송신한다.

4) B는 A로부터 수신된 정보를 복호화하여, 동일한 세션 키를 A와 공유한다.

5) 공유된 세션 키로 A와 B는 데이타를 암.복호화하여 통신한다.

KDC는 전 가입자들의 모든 통신 내용을 도청할 수 있다. 그러므로 가입자들은 KDC를 전적으로 신뢰하여야 하며, 만약 제3자가 KDC의 데이타 베이스에 침입할 수 있다면 전 시스템의 안전은 보장될 수 없다.

2. 공개 키 암호에 의한 키 분배 방식

1) 송신자 A가 자신의 공개 키를 B에게 전송한다.

2) B는 랜덤 키 K를 생성한 후, 이를 A의 공개 키로 암호화하여 A에게 전송한다.

3) A는 자신의 비밀 키로 복호화하여, 송.수신 양측에서 세션 키 K를 공유한다.

실질적으로 이런 종류의 기법은 키 분배를 보다 용이하게 하기위하여, 가입자들의 공개 키에 관한 데이타 베이스를 구축하여 이용한다. 이러한 키 분배 기법은 매우 취약하다. 제3자인 M은 3)의 방법으로 매우 간단히 A와 B사이의 통신 내용을 도청할 수 있다.

3. 공개 키 암호에 의한 키 분배 방식 공격 기법

1) 송신자 A가 B에게 자신의 공개 키를 전송할 때, C는 이를 가로챈 후, B에게 자신의 공개 키를 전송한다.

2) B가 A에게 자신의 공개 키를 전송할 때, C는 이를 가로챈 후, A에게 자신의 공개 키를 전송한다.

3) A가 B의 공개 키라고 생각하고 있는 실질적으로는 C의 공개 키로 데이타를 암호화하여 B에게 전송할 때, C는 이를 가로채어 자신의 비밀 키로 복호화한 후, 다시 이 데이타를 B의 공개 키로 암호화하여 B에게 전송한다.

4) B가 A에게 보내는 데이타도 동일한 방법으로 가로채어 A와 B 사이의 모든 통신을 도청한다

이러한 공격을 방지할 수 있는 가장 효과적인 방식은 A와 B가 서로 동일한 공개 키를 사용하고 있는지를 확인할 수 있는 방법을 제공하는 것이다.

4. Interlock 프로토콜[7]

R. Rivest와 A. Shamir가 제안한 프로토콜을 소개한다. 이 프로토콜은 3)의 공격방식에 효과적으로 대처할 수 있다.

1) 송신자 A가 B에게 자신의 공개 키를 전송한다.

2) B 역시 A에게 자신의 공개 키를 전송한다.

3) A는 B의 공개 키로 통신하고자 하는 데이타를 암호화한 후, 암호문의 절반을 B에게 전송한다.

4) B 역시 통신하고자 하는 데이타를 A의 공개 키로 암호화한 후, 암호문의 절반을 A에게 전송한다.

5) A는 나머지 절반의 암호문을 B에게 전송한다.

6) B는 수신된 암호문을 모은 후 이를 복호화하며, 이후 자신의 나머지 절반 암호문을 A에게 송신한다.

7) A는 수신된 암호문을 모은 후 이를 자신의 비밀 키로 복호화한다.

C는 3)의 1-2 단계와 같이 A와 B에게 자신의 공개 키를 사용하여 데이타를 암호화하도록 유도할 수는 있지만, 3-4단계를 수행할 수 없으므로, 이 프로토콜은 3)의 공격방식에 대하여 안전하다

5. 전자 서명을 동반한 키 분배 기법

KDC는 A와 B의 공개 키에 전자 서명을 부가하여 3)과 같은 능동적 공격에 대처할 수 있다. 즉 KDC는 자신의 비밀 키로 가입자 개개인의 공개 키와 Identity 정보를 암호화한 후 이를 공개하며, 모든 가입자는 KDC의 공개 키를 사용하여 각 가입자의 공개 키 및 Identity를 확인할 수 있다. A와 B는 각자 상대방으로부터 공개 키와 관련된 정보를 수신한 후, KDC의 공개 키로 복호화하여 서로 상대방의 공개 키와 Identity를 확인한 후, 통신을 시작한다. 제 3자인 C는 A와 B의 비밀 키를 모르기 때문에 통신로 상의 암호문을 복호화 할 수 없으며, 또한 KDC의 비밀 키를 모르기 때문에 A, B의 공개 키와 관련된 서명을 할 수 없으므로 타인을 가장할 수도 없다

6. 키와 메세지를 동시에 전송하는 분배 기법

1) 송신자 A는 먼저 랜덤 세션 키를 생성한 후, 이 키를 사용하여 메세지를 암호화한다.

2) A는 공개 키 데이타 베이스에서 B의 공개 키 및 Identity를 확인한 후,

3) 세션 키를 B의 공개 키로 암호화하고

4) 키를 암호화 결과와 메세지를 암호화한 결과를 동시에 B에게 전송한다.

7. 동보 통신에서의 키 분배 기법

1) 송신자 A는 랜덤 세션 키를 생성, 이 키를 사용하여 메세지를 암호화한다.

2) A는 수신자들의 공개 키를 DB에서 확인한다.

3) A는 수신자들의 공개 키를 사용하여 세션 키를 암호화한다.

4) A는 메세지 암호화한 결과와 세션 키를 암호화한 결과를 동시에 동보한다.

5) 각 수신자는 자신의 비밀 키를 사용하여 세션 키를 복호화하며,

6) 복호화된 세션 키를 사용하여 암호문을 복호화한다.

Ⅳ. 인증 기법(Authentication Techniques)

1. 일방향 함수에 의한 사용자 인증 방식

사용자가 컴퓨터(또는 현금 자동 지급기) 사용을 요청하였을 때, 컴퓨터는 요청을 한 사용자가 정당한 사용자인지 불법적인 사용자인지를 판단해야 하는 문제점이 발생한다. 전통적으로는 패스워드를 이용하여 이런 문제점을 해결한다.

A가 자신의 패스워드를 입력하면, 컴퓨터는 입력된 패스워드가 정당한지 여부를 판별하게 된다. 컴퓨터와 사용자는 모두 비밀 정보인 패스워드를 공유해야 한다. 그러므로 컴퓨터가 공유한 모든 가입자들의 패스워드 파일이 침입자에 의해 노출된다면, 그 순간 컴퓨터 시스템의 안전성은 붕괴되고 말 것이다. 실질적으로 컴퓨터가 사용자의 패스워드를 정확하게 확인할 필요는 없으며, 단지입력된 패스워드의 정당성 여부만 확인하면 되며, 정당성 여부는 일방향 함수를 사용하여 쉽게 확인할 수 있다.

사용자 인증 절차

1) 사용자가 컴퓨터에 패스워드를 입력한다.

2) 일방향 함수에 의해 해쉬값을 계산한다.

3) 컴퓨터 내부에 저장된 해쉬값과 비교하여 사용자를 인증한다.

침입자가 해쉬값을 기록한 파일을 확보하여도 그는 사용자들의 패스워드를 알 수 없다일방향 함수는 패스워드로부터 해쉬값을 계산할 수는 있지만 해쉬값으로 부터 패스워드를 계산할 수는 없기 때문이다.

2. 사전식 공격방식

침입자가 해쉬값을 기록한 파일을 입수한 경우, 비록 해쉬값으로부터 패스워드를 계산할 수는 없지만, 일반적으로 컴퓨터 사용자 특히 현금 지급기 사용자들은 자신과 연관되고 기억하기 용이한 내용으로 패스워드를 만든다는 사실을 이용하여 사용자들의 패스워드를 구할 수 있다.

침입자는 먼저 일반적으로 널리 이용되는 패스워드

나 또는 사용자들과 특별히 연관된 패스워드를 예측하여 약 100,000개 정도의 예상 패스워드를 만든 후, 일방향 함수를 적용하여 각각의 해쉬값을 계산하고 이를 입수한 해쉬값 파일과 비교한다. 만약 동일한 해쉬값을 발견한다면 이는 곧 특정 사용자의 패스워드를 발견한 것이 된다. 이런 공격방식을 사전식 공격방식이고 정의한다. 실질적으로 패스워드는 8 바이트의 문자로 구성되므로, 결국 800K 정도의 메모리와 패스워드를 예상하고 해쉬값을 계산하는 약간의 시간만으로 사전식 공격방식을 수행할 수 있으며, 매우 성공적인 공격방식으로 판명되었다.

3. 공개 키 암호에 의한 사용자 식별

패스워드를 사용하여 사용자 식별을 하는 방식은 사용자의 비밀정보가 통신로 상에 또는 컴퓨터 내부에 노출되므로 안전성이 취약한 방식으로 평가된다. 사용자 A가 자신의 패스워드를 입력할 때, 이 패스워드가 전달되는 경로를 제3자가 접근한다면, A의 프라이버시는 그 이후부터 보장될 수 없다. 또한 컴퓨터 침입자가 해쉬값을 계산하는 프로세스의 메모리에 침입할 수 있다면, 그는 곧 모든 가입자의 비밀정보를 알 수 있게 된다.

그러나 이런 문제점은 공개 키 암호를 도입하면 쉽게 해결된다. 호스트 컴퓨터 내부에 모든 사용자들의 공개 키를 기록한다. 이제 공개 키 암호에 의한 사용자 식별 프로토콜을 소개한다.

1) 컴퓨터를 이용하고자 하는 사용자 A에게 컴퓨터는 랜덤 비트 수열을 전송한다.

2) A는 자신의 비밀 키로 랜덤 비트 수열을 암호화하여, 자신의 Identity와 함께 호스트에 전송한다.

3) 호스트는 A의 공개 키를 사용하여, 수신한 비트 수열을 복호화한다.

4) 호스트는 자신이 A에게 전송한 랜덤 비트 수열과 복호화된 결과를 비교하여 사용자 A를 식별한다.

A 자신의 비밀 정보는 통신로 상에는 물론 호스트 컴퓨터 내부에도 절대 노출되지 않는 점이 이 방식의 안전성을 보장한다.

4. SKID [2]

SKID2, SKID3는 RACE's RIPE 프로젝트에서 개발된 비밀 키 암호에 의한 사용자 식별 방식이다. SKID2를 사용하기 위해서는 먼저 A와 B는 상호간의 비밀 키 K를 공유 해야 한다. B가 A에게 자신의 Identity를 SKID2에 의해 증명하는 과정을 고찰한다.

1) A는 64 비트 랜덤 수 RA를 선택, B에게 전송한다.

2) B 역시 64 비트 랜덤 수 RB를 선택, A에게 $R_B, H_K(R_A, R_B, B$의 ID)를 전송한다. 여기서 HK는 RIPE 보고서에서 제시된 RIPE-MAC 함수이다.

3) A는 $H_K(R_A, R_B, B$의 ID)를 계산하여 수신된 값과 비교, 상대방이 B임을 확인한다.

SKID3은 B가 A에게 자신을 증명하는 것뿐만 아니라, A 역시 B에게 자신을 증명하는 상호 인증 프로토콜이다. SKID3 1 - 3 단계는 SKID2와 동일하며, 이에 다음의 4 - 5단계를 추가로 수행한다(4) A는 B에게 $H_K(R_B$ B의 ID)를 송신한다.

5) B는 $H_K(R_B, B$의 ID)를 계산하여, 이를 A로부터 수신한 값과 비교 A를 인증한다.

Ⅴ. 인증과 키 분배 동시 수행 기법

A와 B가 비밀 통신을 수행하기 위해서는 동일한 키를 공유해야 할 뿐만 아니라 서로 상대방을 인증 제3자의 침입을 방지하여야 된다. 이제 A와 B가 세션 키를 공유함과 동시에 서로 상대방을 인증하는 기법들을 고찰한다.

1. Wide-Mouth Frog 프로토콜 [3]

Wide-Mouth Frog 프로토콜은 비밀 키 알고리즘을 이용하며, 절대적 신뢰가 보장되는 키 관리 센타를 가정한 가장 단순한 프로토콜이다. 모든 가입자는 키 관리 센타와 사전에 세션 키 암호화에 사용되는 마스타 키를 공유하고 있다고 가정하며, 세션 키 분배 및 상호 인증을 동시에 수행한다.

1) 송신자 A는 timestamp T_A, B의 ID, 랜덤 세션 키 K를 연쇄시킨 후, 이를 센타와 공유한 마스타 키로 암호화하여 자신의 ID와 함께 센타에 송신한다. (전송 정보: A, $E_A(T_A, B, K)$)

2) 센타는 A와 공유한 마스타 키를 사용하여, A로부터 수신된 정보를 복호화한 후, time stamp T_B, A의 ID, A가 생성한 랜덤 세션 키 K 를 연쇄 시킨 후, 이를 B와 공유한 마스타 키로 암호화하여 B에게 송신한다.(전송 정보: $E_B(T_B, A, K)$)

3) B는 센타와 공유한 마스타 키로 수신한 정보를

복호화하여, 교신 상대방이 A이며 세션 키가 K 임을 확인한다.

2. Yahalom 프로토콜[3]

Yahalom에 의해 제안된 프로토콜로서, Wide-Mouth Frog 프로토콜과 같이 비밀 키 알고리즘을 이용하며, 절대적 신뢰가 보장되는 키 관리 센타를 가정하며,모든 가입자는 키 관리 센타와 사전에 세션 키 암호화에 사용되는 마스타 키를 공유하고 있다고 가정하며, 세션 키 분배 및 상호 인증을 동시에 수행한다.

1) 송신자 A는 자신의 Identity와 랜덤 수 RA를 연쇄시켜, 이를 수신자 B에게 전송한다.(전송 정보: A, R_A)

2) 수신자 B는 A의 ID와 랜덤 수 RA 및 자신이 생성한 랜덤 수 RB를 연쇄 시킨 후, 이를 센타와 공유한 마스타 키로 암호화하여 자신의 ID와 함께 센타에 전송한다. (전송 정보: B, $E_B(A, R_A, R_B)$)

3) 센타는 A와 B에게 전송할 2개의 메세지를 생성한다. 첫번째 메세지는 먼저 B의 ID, A와 B가 사용할 랜덤 세션 키 K, A가 생성한 랜덤 수 RA, B가 생성한 랜덤 수 RB를 연쇄 시킨 후, 이를 A와 공유한 마스타 키로 암호화한 메세지이며, 두번째는 A의 ID 와 랜덤 세션 키 K 를 연쇄 시킨 후, 이를 B와 공유한 마스타 키로 암호화하여 메세지이다. 센타는 이 메세지를 모두 A에게 전송한다. (전송 정보: $E_A(B, K, R_A, R_B)$, $E_B(A, K)$)

4) A는 수신한 정보 중 첫번째 메세지를 자신과 센타가 공유한 마스타 키로 복호화하여 랜덤 수가 자신이 생성한 수인가를 확인하고, 세션 키 K를 추출한다. A는 B가 생성한 랜덤 수 RB를 세션 키 K로 암호화한 후, B의 마스타 키로 암호화된 정보와 같이 B에게 전송한다.(전송 정보: $E_B(A, K)$, $E_K(R)$)

5) B는 수신된 메세지를 자신과 센타와 공유한 마스타 키로 복호화하여 세션 키 K를 추출한 후, 다시 이 세션 키를 사용하여 A가 암호화한 정보를 복호화하여 랜덤수가 자신이 생성한 수인지를 확인한다.

3. Needham-Schroeder 프로토콜[4]

Needham과 Schroeder가 제안한 프로토콜로서, 비밀 키 알고리즘을 이용하며, 절대적 신뢰가 보장되는 키 관리 센타를 가정하며,모든 가입자는 키 관리 센타와 사전에 세션 키 암호화에 사용되는 마스타 키를 공유하고 있다고 가정하며, 세션 키 분배 및 상호 인증을 동시에 수행한다

1) 송신자 A는 센타에 자신의 ID, 수신자 B의 ID, 랜덤 수 R_A를 전송한다. (전송 정보: A, B, R_A)

2) 센타는 랜덤 세션 키 K를 발생하여, 이를 A 의 ID와 연쇄 시킨 후, 수신자 B와 공유한 마스타 키로 암호화하고, 이를 다시 A가 생성한 랜덤 수 R_A, B 의 ID, 세션 키 K,와 연쇄시켜, A와 공유한 마스타 키로 암호화하여 A에게 전송한다.(전송 정보: $E_A(R_A, B, K, E_B(K, A))$)

3) A는 센타로부터 수신한 메세지를 센타와 공유한 마스타 키로 복호화하여 K를 추출하며, 랜덤 수가 자신이 생성한 수인지를 확인한다. 그후, B의 마스타 키로 암호화된 정보를 B에게 전송한다.(전송 정보: $E_B(K, A)$)

4) 수신자 B는 A로부터 수신한 정보를 자신과 센타가 공유한 마스타 키로 복화화하여 세션 키 K를 추출한 후, 랜덤 수 R_B를 생성하여 이를 세션 키 K로 암호화하여 A에게 전송한다.(전송 정보: $E_K(R_B)$)

5) A는 B로부터 수신된 메세지를 K를 사용하여 복호화 R_B를 추출한 후, R_B-1을 K를 사용하여 암호화 B에게 전송한다.(전송 정보: $E_K(R_B-1)$)

6) B는 수신된 메세지를 복호화하여, 그 수가 R_B-1임을 확인한다.

이 프로토콜은 이미 A와 B간에 사용된 키 K를 제3자 C가 알고 있는 경우, 다음과 같은 절차에 의해 제3자 C가 A인채를 할 수 있기 때문에 안전하지 못하다.

(1) 제3자 C는 A와 B간의 이전 통신에서 가로챈 정보 $E_B(K, A)$를 B에게 전송한다.

(2) B는 이 정보를 센타와 공유한 마스타 키로 복호화한 후, 랜덤 수 R_B를 생성하여 세션 키 K로 암호화하여 A에게 전송한다.

(3) C는 이 메세지를 가로채어 K로 복호화하여 RB를 추출한 후, B에게 R_B-1을 암호화하여 전송한다.

(4) B는 제3자 C를 A로 확인하고 비밀 통신을 시작한다.

4. Otway-Rees 프로토콜[6]

Otway-Rees 프로토콜은 Needham-Schroeder 프로토콜의 취약점을 개선한 프로토콜로서,비밀 키 알고리즘을 이용하며, 절대적 신뢰가 보장되는 키 관

리 센타 가정하며,모든 가입자는 키 관리 센타와 사전에 세션 키 암호화에 사용되는 마스타 키를 공유하고 있다고 가정하며, 세션 키 분배 및 상호 인증을 동시에 수행한다.

1) 송신자 A는 Index I, 자신의 ID, 수신자 B의 ID, 랜덤 수 RB를 센타와 공유한 마스타 키로 암호화하여, 다시 Index I, 자신의 ID, B의 ID 와 연쇄시켜 B에게 전송한다.
(전송 정보: I, A, B, $E_A(R_A,I,A,B)$)

2) 수신자 B는 먼저 랜덤 수 RB를 생성하여 이를 Index I, A의 ID, 자신의 ID와 연쇄시켜 센타와 공유한 마스타 키로 암호화한 후, 이를 A로부터 수신한 Index I, A의 ID,B의 ID, A가 센타와 공유한 마스타 키로 암호화한 정보를 함께 센타에 전송한다. (전송 정보: I, A, B, $E_A(R_A,I,A,B)$, $E_B(R_B,I,A,B)$)

3) 센타는 랜덤 세션 키 K를 생성하여, 이를 A가 생성한 랜덤 수 RA와 연쇄시켜 A와 공유한 마스타 키로 암호화하고, 다시 B가 생성한 랜덤 수 RB와 연쇄시켜 암호화한 후, 이를 모두 Index I와 함께 B에게 전송한다. (전송 정보: I, $E_A(R_A,K)$, $E_B(R_B,K)$)

4) B는 A에게 I, $E_A(R_A,K)$를 전송한다.

5. Kerberos

Kerberos는 Needham-Schroeder 프로토콜을 기반으로 설계되어, UNIX TCP/IP에서 사용되는 인증 프로토콜이다. MIT의 Athena 프로젝트에 의해 개발되었으며, DES 알고리즘을 이용하며, 절대적 신뢰가 보장되는 키 관리 센타를 가정하며,모든 가입자는 키 관리 센타와 사전에 세션 키 암호화에 사용되는 마스타 키를 공유하고 있다고 가정하며, 세션 키 분배 및 상호 인증을 동시에 수행한다. Kerberos Version 1 - 3은 프로젝트 팀 내부에서 사용되는 시제품이었고, Version 4가 original이며[7], 이를 개선한 Version 5가 있다.[8] 실제 망에서의 응용에 관해서는[9] 에 소개되어 있다. Version 5를 간략히 소개한다.

1) 송신자 A는 센타에 자신과 수신자 B의 ID를 전송한다.(전송 정보: A, B)

2) 센타는 time stamp T, 유효시간 L, 랜덤 세션 키 K를 생성, 이를 A의 ID와 연쇄시켜 B와 공유한 마스타 키로 암호화하고, 다시 B의 ID와 연쇄시켜 A 와 공유한 마스타 키로 암호화한 후, A에게 전송한다.($E_A(T,L,K,B)$, $E_B(T,K,L,A)$)

3) A는 수신된 메세지 중 자신의 마스타 키로 암호화된 정보를 복호화하여, 세션 키 K를 추출하고 이를 사용하여 자신의 ID와 time stamp T를 연쇄시켜 암호화한 후, 수신한 메세지 중 B의 마스타 키로 암호화된 정보와 함께 B에게 전송한다.(전송 정보: $E_K(A,T)$, $E_B(T,L,K,A)$)(4) B는 자신의 마스타 키로 센타가 암호화한 정보를 복호화 세션 키, time stamp, 유효 시간, A의 ID 를 확인한 후, T+1을 세션 키 K로 암호화하여 A에게 전송한다.(전송 정보: $E_K(T+1)$)

Kerberos 프로토콜은 망내에 모든 clock 동기가 가능할 때 성립되는 프로토콜이다.

Ⅵ. 결 론

본고에서는 키 분배 및 인증과 관련된 기본적 사항들을 간단히 고찰 소개하였다. 키 분배 방식을 구성하는 주요 요소들을 소개하였으며, 기본적인 키 분배 방식과 여러 종류의 인증 기법 및 그 기법들의 안전성을 소개하였으며, 인증과 키 분배를 동시에 수행하는 프로토콜과 UNIX 망에서 실질적으로 서비스되는 Kerberos 프로토콜을 소개하였다.

參 考 文 獻

[1] R. Rivest and A. Shamir, "How to Expose an Eavesdropper", Communications of ACM, V. 27(7), 1984, pp. 393-395.

[2] RACE, RIPE Integrity Primitives: Final Report of RACE Integtity Primitive Evaluation(R1040), June 1992.

[3] M. Burrows, M. Abadi and R. Needham, A Logic of Authentication, Digital Equipment Corp. Systems Research Center, Feb. 1990.

[4] R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers",

Commu. ACM, Vol. 21(12), 1978, pp.993-999.
[5] D. E. Denning and G. M. Sacco, "Timestamps in Key Distribution Protocols",Commu. ACM, Vol. 24(8), 1981, pp.533-536.
[6] D. Otway and O. Rees, "Efficiently and Timely Mutual Authentication", Operating Systems Review, Vol. 21(1), 1987, pp. 8-10.
[7] S. P. Miller, B. C. Neuman, J. I. Schiller, and J. H. Saltzer, "Section E.2.1: Kerberos Authentication and Authorization System", MIT Project Athena, 1987
[8] J. T. Kohl, "The Evolution of the Kerberos Authentication Service", Europen Conference Proceedings, 1991, pp.295-313.
[9] T. Jin, Care and Feeding of Your Three-Headed Dog, Doc. IAG-90-011, Hewlett-Packard, 1990.

筆者紹介

高 承 哲
1957年 3月 14日生
1981年 2月 연세대 수학과 졸업(학사)
1983年 2月 연세대 대학원 수학과(이학석사)
1992年 8月 포항공대 대학원 수학과(이학박사)

1984年 3月 ~ 현재 한국전자통신연구소 연구원, 선임 책임 연구원

주관심 분야 : 정보 보호 기술

李 相 珍
현재 한국전자통신연구소 근무중