상호 실체인증 기능을 갖는 안전한 키 교환 알고리즘

강 창 구† · 최 용 락††

요 약

본 논문에서는 Diffie-Hellman(DH) 키 교환 원리를 응용하여 상호 실체 인증 기능을 갖는 2-패스와 3-패스의 2가지 키 교환 알고리즘을 제안하였다. 제안된 방식의 키 교환 안전성은 본래의 DH 키 교환 메커니즘과 동일한 이산대수 문제와 관계되며, 상호 실체인증의 안전성은 사용된 디지털 서명구조 및 포함된 인수들에 의존한다. 제안된 방식은 Kerberos, X.509 인증 교환방식 및 ISO 3단계 인증 프로토콜 보다 단순하고 효과적이며, 다양한 형태의 신분위장, 재전송 및 가로채기 공격 등을 방어할 수 있다.

An Algorithm for Secure Key Exchange based on the Mutual Entity Authentication

Chang-Goo Kang† · Yong-Rak Choi††

ABSTRACT

In this paper, we propose two authentication exchange schemes which combine public key-based mutual authentication with a Diffie-Hellman key derivation exchange. The security of key exchange of the proposed schemes depends on the discrete logarithm problem. The security of the entity authentication depends on that of the signature mechanism to be used in the proposed scheme. In comparison with the Kerberos, X.509 exchanges, and ISO 3-way authentication protocol, the proposed schemes are not only simple and efficient, but also are resistant to the full range of replay and interception attacks.

1. 서 론

정보 통신망을 이용하는 많은 응용 분야들이 정보의 기밀성 노출 및 실체의 신분위장 등 다양한 형태의 위협들로부터 보호하기 위하여 필요한 보안 요구사항들을 갖고 있다. 사실상 다양한 위협들에 대하여 보호하기 위해서 사용되는 대부분의 보안 서비스와 메커니즘들은 통신 상대에 대하여 신뢰할 수 있도록 알려진 신분 즉,인증에 의존한다[1]. 본 논문은 *Diffie-Hellman* 키 교환원리를[2] 기반으로 통신 실체간에 인증 기능을 갖도록 보장하는 2-패스와 3-패스 인증 프로토콜을 다룬다.

인증은 다양한 다른 보안 서비스들이 인증된 실체를 기반으로 수행되기 때문에 보안 서비스 중에서 가장 기본적이며, 다른 보안 서비스들과도 결합되어 이용될 수 있다[4]. 예를 들면, 접근 제어 정책에서 주체의 접근을 통제하기 위하여 필요한 접근 제어 정보를 요구할 때 우선 인증되어야 할 필요가 있을 수 있다. 특히, 기밀성과 무결성 서비스들을 제공하면서 키들을 분배하기 위한 구조에서 인증 시스템을 이용할 수 있다. 실체 인증은 관용 암호화 기법이나 공개키 암호화 기법에 기초하여 성립될 수 있다[3,4].

† 정 회 원 : 한국전자통신연구원 부호기술연구부
†† 종신회원 : 대전대학교 컴퓨터통신전자공학부 교수
논문접수 : 1998년 3월 14일, 심사완료 : 1998년 6월 1일

관용 암호화에 기초한 인증 구조는 두 실체가 어떻게 키를 분배하였는 이미 비밀 키를 공유하고 있거나 또는 별도로 키 분배 센터의 사용을 필요로 한다.

*Kerberos*는 기본적으로 관용 암호화 기술을 적용하고 있는데, 비교적 저렴한 가격의 기술을 사용하여 높은 수준의 보호를 제공한다. 그러나, 물리적으로 안전한 온라인 서버가 필요하고, 인증 서비스를 얻기 위하여 많은 세션 키 및 안전한 클럭 구조가 요구되는 약점을 가진다[3-5].

공개키 기술을 이용하는 메커니즘은 공개키 인증서와 인증서 취소 리스트를 분배하기 위해서 오프라인 인증 서버가 필요하다[1,2]. *X.509* 인증 교환은 공개키 기반 인증 기법의 좋은 예이다. 사용자 A와 사용자 B사이의 상호 인증을 제공하기 위해서, 사용자 A는 사용자 B의 공개키 인증서가 필요하고, 사용자 B는 사용자 A의 공개키 인증서가 필요하다.

*X.509*는 상당히 일반적 구조이고, 다양한 응용을 위해 제안되었다[8,9]. 인증 시스템의 넓은 응용 범위가 *X.509*하에서 정의될 수 있지만, *X.509*는 단순히 인증된 키 교환 서비스를 제공하는데는 효과적이지 못하다.

본 논문에서는 기존의 *Kerberos*, *X.509* 및 ISO 3-패스 인증 프로토콜의 방법과 다른 형태의 공개키 기반 상호 인증과 키 교환 기능을 갖는 2-패스와 3-패스 방식의 두가지 키 교환 알고리즘을 제안한다.

2. 인증과 키 교환 요구사항

2.1 상호 인증

분산 컴퓨팅 시스템에서의 인증은 메시지 인증과 실체 인증의 두 가지 형태의 인증이 있다. 메시지 인증은 받은 메시지의 내용이 메시지를 보냈을 때와 같은지를 증명한다. 실체 인증은 일반적으로 통신 상대방의 신분이 주장하는 것과 같은지를 증명한다[10]. 실체 인증은 단독 인증이거나 또는 상호 인증일 수 있다. 단독 인증은 통신 활동의 오직 한쪽 상대방만이 다른 쪽 상대방을 인증한다. 상호 인증은 양쪽의 상대방들이 서로를 인증한다.

통신 세션의 각 통신 상대방이 서로를 인증하는 것이 필요한 상호 인증은 키 교환에 있어서 중요한 요구 사항이다. 원리적으로 상호 인증은 두 단독 인증 프로토콜 교환의 결합으로 얻을 수 있다. 그러나, 이러한 결합은 가로채기-재전송 공격에 대한 기회가 증가하므로 주의 깊게 설계되어야 한다.

또한, 프로토콜 메시지의 수가 결합된 단독 인증에서 필요한 메시지의 수보다 훨씬 적게 상호 인증 교환을 설계할 수도 있다. 그러므로, 안전성과 성능상의 이유 때문에, 키 교환 기능을 갖는 고유의 상호 인증 알고리즘을 특별하게 설계할 필요가 있으며, 본 연구에서는 다자간 인증 또는 단독 인증보다는 양자간의 상호 인증에 기반을 둔 키 교환을 다룬다.

키 교환 없는 상호 인증을 제공하는 프로토콜은 인증이 완료되기를 기다린 다음에 통신 라인의 한쪽을 점유하는 침입자들에게 공격받기가 쉽다. 이러한 공격은 인증과는 독립적으로 키 교환이 이루어질 경우에 시도될 수 있는 취약점이다.

따라서, 키 교환은 통신 상대방이 교환된 키를 실제로 침입자가 아닌 인증된 통신 상대방과 공유한다는 것을 확신하기 위해서 인증과 연결되어야 한다[3,11]. 이러한 요구 사항을 만족시키기 위해서 인증 프로토콜의 설계와 분석에 있어서 키 교환을 고려해야 할 필요가 있다.

2.2 키 교환

인증된 키 교환의 문제에서 중요한 요구 사항은 기밀성과 적시성이다[7]. 신분 위장과 세션 키의 손상을 방지하기 위해서 기본적으로 신분과 세션 키 정보들은 암호화된 형태로 통신하여 기밀성을 보장해야 한다. 이러한 키 교환의 기밀성 유지를 목적으로 관용 암호 방식의 비밀 키와 공개키 암호 방식의 공개 키가 다양한 형태로 이용될 수 있다. 그러나, 각 통신 상대방이 암호화된 형태의 통신 없이 직접적으로 계산하여 세션 키를 얻는다면 더욱 바람직 할 것이다.

적시성은 메시지의 재전송 위협을 방어하는데 중요하다. 재전송은 침입자가 세션 키를 위태롭게 하거나 다른 통신 상대방으로 위장할 수 있게 한다. 최소한, 성공적인 재전송은 정당한 것처럼 보이는 메시지를 통신 상대방에게 보임으로써 동작을 방해할 수 있다. 재전송 공격을 방어하기 위한 방법 중 하나는 인증 교환에서 사용되는 각 메시지에 일련 번호를 부여하는 방법이다. 이 방법의 어려운 점은 각 통신 상대방이 그들이 사용한 각 신청자별 마지막 일련 번호를 유지해야 한다는 것이다. 대신에 다음의 일반적인 두 방법 중 하나가 사용될 수 있다.

(1) 타임 스탬프: 사용자 A는 메시지가 현재 시간과

충분히 근접하다고 판단되면 타임 스탬프를 가지고 있을 때만 올바른 메시지로 받아들인다. 이 방법은 다양한 참여자들 간에 동기화 된 클럭을 요구한다.

(2) *Challenge/Response* : 사용자 B로부터의 올바른 메시지를 기대하는 사용자 A는 우선 사용자 B에게 임시비표를 보내고, 사용자 B로부터 받은 응답 메시지에서 동일한 임시비표의 확인을 요구한다.

타임 스탬프가 관리와 문서화 목적으로는 편리한 반면, 문제점은 각 통신 상대방이 가변적이고 예측할 수 없는 네트워크 지연의 속성하에서 정확한 동기화를 유지하는 것이다. 한편, *Challenge-Response* 방법은 프로토콜 설계에서 핸드쉐이크의 오버 헤드를 요구하고, 가로채기-재전송 공격에 취약한 문제점을 갖고 있다.

*Diffie*와 *Hellman*은 공개키 암호화에 기초한 키 교환 구조를 제안했는데, 일반적으로 *Diffie-Hellman (DH)* 키 교환이라고 한다[2]. 이 알고리즘 자체의 목적은 키 교환에 있으며, 실제로 유용하기 위해서는 실체 인증 과정과 결합될 필요가 있다. 인증기능을 갖지 않는 키 교환에서 침입자 C가 전송된 메시지를 가로챌 수 있다면, 침입자 C는 사용자 A 또는 사용자 B로 위장할 수 있고, 네트워크에서 사용자 A와 사용자 B 사이에 위치하여 그들 둘 사이에 전송되는 메시지를 조사하고, 수정 전송해서 사용자 A 및 사용자 B와 공유하는 각각의 키를 만들어 이용할 수 있다. 이러한 문제점들은 키 교환과 인증의 두 가지 기능을 결합시킨 인증된 키 교환 프로토콜에 대한 필요성을 요구한다.

따라서, 본 논문에서는 *DH* 키 교환 방식의 원리를 응용하여 상호 실체인증 기능을 갖는 2-패스와 3-패스 2 가지의 안전한 키 교환 알고리즘을 제안한다.

3. 상호 실체인증 기능을 갖는 키 교환

3.1 2-패스 인증 방식

사용자 A가 사용자 B와 통신을 연결하고, 연결된 통신상에서 메시지를 암호화하기 위해 비밀 키를 획득 하고자 할 때, 2-패스 인증과 키 교환 구조는 <그림 1>과 같다. 소수 P와 그것의 원시 근 α는 미리 알려져 있거나, 사용자 A가 P와 α를 선택적으로 고르고 첫 메시지에 그것들을 포함했다고 가정한다.

사용자 A는 사용자 B의 서명 $S_B(Y_a \| Y_b \| T$ 또는 $SN)$을 검증함으로써 사용자 A의 실체를 인증할 수 있다. 사용자 A는 이 단계와 관련된 특정 메시지를 서명했기 때문에 사용자 A는 자신이 보낸 메시지의 생성자임을 사용자 B에게 증명한다. 타임 스탬프 T는 사용자 B에게 사용자 A의 서명이 유일하게 생성된 것임을 확신시킨다. 사용자 B는 다음을 검사함으로서 적시성을 증명할 수 있다.

$$|clock - T| < \delta t_1 + \delta t_2$$

δt_1은 사용자 A와 사용자 B의 클럭간의 측정된 표준편차이고, δt_2는 예상되는 네트워크 지연 시간이다. 타임 스탬프 T는 서명에 포함되기 때문에 침입자가 과거의 메시지를 안다고 할지라도 사용자 B에 의해서 메시지의 적시성을 만족시킬 수 없으므로 성공할 수 없다. 그러나, 침입자가 정당한 시간 구간 안에서 타임 스탬프가 있는 메시지를 재전송할 수 있다면, 선택적으로, 순서번호 SN을 추가하거나 타임 스탬프 T 대신에 사용할 수 있다. 이 구조에서, 임시비표의 사용은 그 값이 각 통신 상대방들에 의해서 모방될 수 있고 가로채기-재전송 공격의 방어가 어렵기 때문에 적절하지 못하다.

본 논문에서 사용된 기호들은 다음과 같다.

A, B	: 통신 상대방인 사용자들
P	: 소수
α	: 소수 P의 원시 근($\alpha(P)$
X_a	: A에게만 알려진 비밀 값
X_b	: B에게만 알려진 비밀 값
Y_a	: 공개 값($Y_a = \alpha^{X_a} \bmod P$)
Y_b	: 공개 값($Y_b = \alpha^{X_b} \bmod P$)
$S_A(M)$	: 메시지 M에 해쉬함수를 적용한 메시지 다이제스트에 대한 사용자 A의 서명 값
$S_B(M)$	: 메시지 M에 해쉬함수를 적용한 메시지 다이제스트에 대한 사용자 B의 서명 값
K	: 사용자 A와 B 사이의 비밀 키
T	: 타임 스탬프
SN	: 순서 번호
(M)	: 전송 메시지 M

사용자 A는 사용자 B의 서명 $S_B(Y_a \| Y_b \| K)$를 검증 함으로써 사용자 B의 실체를 인증할 수 있다. 이 서명은 비밀 값 K를 알지 못하면 정당한 서명을 생성할 수 없기 때문에 재전송과 위장 공격에 대하여 안전

하다. 결국, 두 사용자는 통신 상대방의 서명을 검증함으로써 상호 실체 인증을 달성하고, 사용자 A는 $K = (Y_b)^{X_a} mod\ P$ 를, 사용자 B는 $K = (Y_a)^{X_b} mod\ P$를 각각 계산하여, 두 사용자 모두가 성공적으로 동일한 비밀 키를 획득할 수 있다.

사용자 A(송신자)
A : Generate a secret value X_a and timestamp T ; (X_a< P-1)
A : Calculate $Y_a = \alpha^{X_a} mod\ P$;
A : Generate signature S_A{Y_a, T or SN};
A-->B: Send the message {Y_a, T, S_A{Y_a, T or SN}};
A<--B: Receive the message {Y_b, S_B{Y_a, Y_b, K}};
A : Calculate the key $K = (Y_b)^{X_a} mod\ P$;
A : Verify B's signature S_B{Y_a, Y_b, K};

사용자 B(수신자)
B : Verify A's signature S_A{Y_a, T or SN};
B : Check the validity of timestamp T;
B : Generate a secret value X_b; (X_b< P-1)
B : Calculate $Y_b = \alpha^{X_b} mod\ P$;
B : Calculate the key $K = (Y_a)^{X_b} mod\ P$;
B : Generate signature S_B{Y_a, Y_b, K};
B-->A: Send the message {Y_b, S_B{Y_a, Y_b, K}};

(그림 1) 2-패스 인증과 키 교환
(Fig. 1) 2-pass authentication and key exchange

3.2 3-패스 방식

3-패스 인증과 키 교환은 (그림 2)와 같다. 사용자 A는 사용자 B의 서명 $S_B\{Y_a||Y_b||K\}$ 을 검증 함으로써 사용자 B의 실체를 인증할 수 있다. 사용자 B는 사용자 A의 서명$S_A\{Y_b||Y_a||K\}$ 을 검증 함으로써 사용자 A의 실체를 인증할 수 있다. 그러므로, 사용자 A와 사용자 B는 상호 실체 인증을 달성할 수 있다. 또한, 2-패스 인증 및 키 교환에서와 마찬가지로 두 사용자는 상대방이 전송한 공개 값 Y_a 및 Y_b 값을 각각 이용하고 각자가 생성한 비밀 값 X_b 및 X_a를 적용하여 필요한 비밀 키를 직접 계산할 수 있다.
이 구조는 3-패스 메시지 교환을 통하여 가로채기 및 재전송 공격을 방어할 수 있다. 또한, *W. Diffie*[11]에서 제안된 방식과는 다르게 서명 자체를 암호화할 필요가 없는 장점이 있다.

4. 알고리즘 분석

(D1) 송수신자가 생성한 공개값 Ya 와 Yb 의 전송

사용자 A(송신자)
A : Generate a secret value X_a;
A : Calculate $Y_a = \alpha^{X_a} mod\ P$;
A-->B: Send the message {Y_a};
A<--B: Receive the message {Y_b, S_B{Y_a, Y_b, K}};
A : Calculate the key $K = (Y_b)^{X_a} mod\ P$;
A : Verify B's signature S_B{Y_a, Y_b, K};
A : Generate signature S_A{Y_b, Y_a, K};
A-->B: Send the message {S_A{Y_b, Y_a, K}};

사용자 B(수신자)
B : Generate a secret value X_b;(X_b<P-1)
B : Calculate $Y_b = \alpha^{X_b} mod\ P$;
B : Calculate the key $K = (Y_a)^{X_b} mod\ P$;
B : Generate signature S_B{Y_a, Y_b, K};
B-->A: Send the message {Y_b, S_B{Y_a, Y_b, K}};
B<--A: Receive the message {S_A{Y_b, Y_a, K}};
B : Verify A's signature S_A{Y_b, Y_a, K};

(그림 2) 3-패스 인증과 키 교환
(Fig. 2) 3-pass authentication and key exchange

순수한 *DH* 키 교환 방식에서 사용자 A 는 $K=(Y_b)^{X_a} mod\ P$를 계산하기 위하여 통신 상대방 B로부터 송신된 공개값 Y_b 가 반드시 필요하다. 마찬가지로, 사용자 B 는 $K=(Y_a)^{X_b} mod\ P$를 계산하기 위하여 통신 상대방 A로부터 송신된 Y_a 가 반드시 필요하다. 즉, 이산 대수 문제를 이용한 키 교환 문제를 해결하기 위하여 두 통신 실체는 상호간에 서로 상대방에서 비밀리 X_a 및 X_b를 선택하고, 각자가 계산한 Y_a와 Y_b 값을 통신 양자간에 갖고 있어야 기본적으로 키 계산이 가능하다. 따라서, 어떠한 형태로든 공개값 Y_a와 Y_b 값이 전송 메시지에 포함되어야 한다.

(D2) 송신자가 생성한 공개값 서명

2-패스 방식에서 송신자는 자신이 생성한 Y_a만을 서명할 경우에 문제점을 갖는다. Y_a 는 공개된 값이지만 송신자 A의 비밀 서명키를 이용하여 서명되었기 때문에 제 3자가 송신자 A의 서명키를 획득할 수 없다면 송신자 A로 위장할 수 없다. 그러나, 송신자 A가 계산한 Y_a 값만이 포함되어 있기 때문에 제 3자가 가로채서 임의의 시간에 재전송하면 수신자 B는 그것이 합법적인 것인지 아닌지 구분할 수 없다. 이것은 Y_a의 이산 대수 문제를 풀기 위한 시간 확보 일 수 있으며, 그렇지 않더라도 통신체계에서 다수의 실체가 송신자 A로 위장하여 통신을 개시한다면 혼란을 초래하게 된다. 따

다시, 송신자 A는 Y_a만을 서명하는 것 보다 재전송 서명에 대하여 직접한 제한을 갖는 시간 종속 변수를 추가하여 포함 할 필요가 있다.

한편, 3-패스 방식의 세번째 메시지에서 계산하거나 또는 유도할 수 있는 어떤 비지수를 알아내고 송신자 A의 서명을 그 비지수에 근거하여 산출할 수 있다면 제 3자는 A로 위장하는데 그 비지수를 사용할 것이다. 이러한 경우에 Y_a 이외에 또 하나의 다른 Y_b 값을 포함 시키는 것은 이산대수 계산 문제를 보다 어렵게 만드는 복합적 효과가 있다.

(D3) 수신자가 생성한 공개값 서명

2-패스와 3-패스 방식의 두번째 메시지는 송신자가 생성했던 공개값 Y_a를 각각 포함하고 3-패스 방식의 세번째 메시지는 수신자가 생성한 공개값 Y_b를 포함하고 있다. 이것은 앞 단계의 메시지에서 수신 하였던 공개 값으로서 본래의 생성자에게 서명내에 포함하여 반환하기 때문에 *challenge/response*와 유사한 효과가 있다. 그러나, 수신자가 생성한 한개의 공개값 Y_b만을 서명에 포함한다면 키의 계산은 단순한 하나의 이산대수 문제로 축소되므로 그 만큼 재전송 및 신분위장 공격이 용이해진다. 따라서, 3-패스의 세번째 메시지에서 수신자가 생성한 Y_b를 서명에 포함 시키고 추가적으로 송신자가 생성한 공개값 Y_a를 포함 시킴으로써 공격 시도를 더욱 어렵게 만드는 효과가 있을 것이다.

(D4) 타임 스탬프 T의 사용

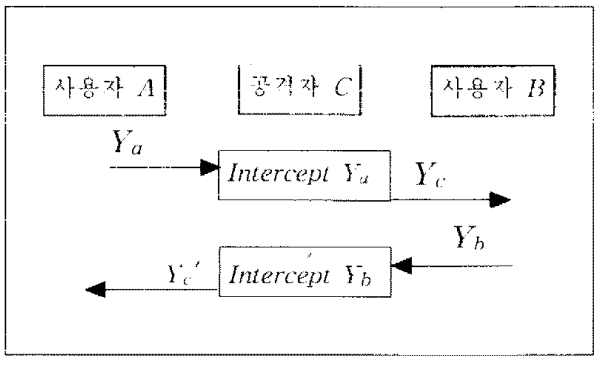
앞의 절에서 언급된 바와 같이 타임 스탬프는 재전송 공격에 효과적으로 대응할 수 있는 수단이다. 2-패스 방식의 첫번째 메시지에서 타임 스탬프 T의 포함은 서명의 유효성을 검증하기 위해서 필요하지만 분산 네트워크에서 동기화된 클럭을 요구하며, 비록 동기화가 보장된 환경에서도 시간 윈도우 내에서의 공격 위협 요소는 여전히 남게 된다. 다른 대안으로써 순서 번호 SN이 사용될 수 있으며, 이것은 다자간의 통신 프로토콜 개시에서 수많은 신청자별 순서번호를 유지하기가 어려운 문제점을 갖고 있다. 또한, *challenge / response* 방식의 임시비표를 사용한다면 가로채기-재전송 공격과 함께 두번째 메시지에서도 동일한 임시비표를 포함해야 하는 핸드쉐이크의 오버헤드를 갖는다. 따라서, 이상과 같은 문제점에 대한 다른 선택으로 3-패스 방식을 사용할 수 있다.

(D5) 비밀키 K의 포함

2-패스 와 3-패스 두가지 방식 모두가 비밀키 K를 계산할 수 있게 되면, 다음 메시지에서 그 비밀키를 포함시켜서 서명을 수행하였다. 서명의 안전성은 서명자의 비밀 서명키에 대한 노출로부터 안전성과 서명 정보의 대상이 되는 데이터들의 구성에 의존적이다. 서명 정보에는 송신자가 전송한 공개값 Y_a를 이용하여 산출한 비밀키 K가 포함되어 있다. 제 3자가 올바른 서명 정보를 구성하기 위해서는 K값을 계산할 수 있어야 한다. 2-패스 방식의 두번째 메시지에서 K를 계산하기 위하여 제 3자는 사용자 B에서 비밀리 선택된 X_b를 찾아낼 수 있어야 하는데, 공개값 Y_b로부터 X_b를 산출하는 것은 대단히 어렵기 때문에 큰 소수에 대하여 실행 불가능한 것으로 간주된다. 마찬가지로 3-패스의 세번째 메시지에서 비밀키 K를 포함하는 것은 동일한 효과를 갖으며, 추가적으로 송신자 A는 K를 올바르게 계산하여 서명에 포함시킬 수 있음을 수신자 B에게 증명하여 보임으로써 강력한 상호인증의 효과를 제공한다.

(D6) 인증기능의 포함

인증 기능을 갖지 않는 단순한 *DH* 키 교환 알고리즘은 (그림 3)과 같은 위협요소를 갖고 있다. 사용자 A가 Y_a를 사용자 B에게 전송할 때 중간에 공격자 C가 Y_a를 가로채고, 자신이 계산한 Y_c를 사용자 B에게 전송할 수 있다. 사용자 B는 Y_c를 근거로 키를 산출하고, Y_b를 사용자 A에게 전송하면 마찬가지로 공격자 C는 Y_b를 가로채고 Y_c'를 사용자 A에게 전송할 수 있다. 즉, 공격자 C는 사용자 A와 사용자 B사이에 위치하여 양쪽 사용자로 위장하여 개입할 수 있는 "*intruder-in-the-middle-attack*"이 가능하다. 이것은 비록 공개



(그림 3) DH 키 교환 알고리즘의 문제점
(Fig. 3) The problem of DH key exchange algorithm

된 값이지만 Y_a 및 Y_b를 검증할 수 있는 아무런 보호 조치가 없기 때문이다. 따라서, 각 사용자가 생성한 공개값에 대하여 합법적 발행자임을 증명할 수 있는 서명이 필요하다. 제안된 방식은 다양한 신분위장 및 재전송 공격을 방어하기 위하여 서명 정보에 Y_a와 Y_b를 포함시키고, 추가하여 타임스탬프 및 비밀로 계산된 키 정보 등을 보완하여 사용하였다.

(D7) 3-패스 방식의 사용 실제

3-패스 방식을 사용한 경우 하나의 실제적인 예는 (그림 4)와 같다. 사용자 A는 첫번째 메시지에서 *DH* 키 교환 알고리즘에서 필요한 파라메터(*DH* 파라메터)들을 사용자 B에게 전송한다. 두번째 메시지에서 사용자 B는 수신된 파라메터들을 사용하여 필요한 정보를 생성하고 사용자 A에게 자신의 인증서($Cert_A$)를 전송한다. 사용자 B의 인증서를 수신한 사용자 A는 신뢰된 인증기관의 서명(KR_{CertB} ())을 검사함으로써 사용자 B의 공개키(Bp)와 *DH* 파라메터들을 검증할 수 있다. 세번째 메시지에서 사용자 A 는 사용자 B에게 $Cert_A$를 전송한다. 마찬가지로, 사용자 A의 인증서를 수신한 사용자 B는 인증기관의 서명(KR_{CertA}())을 검사하여 사용자 A의 공개키(Ap)와 *DH* 파라메터를 검증한다.

사용자 A가 인증서를 세 번째 메시지에서 전송하는 것은 사용자 B가 실제로 필요한 시간에 인증서를 전송함으로써 불필요한 보관 시간을 줄일 수 있고, 두 통신 실체가 비밀키를 상호 공유하는 시점에서 인증서를 검증하도록 하기 위함이다. 비록 인증서가 공개된 값을 갖는다고 해도 양쪽 통신 실체의 신분을 미리 밝히지 않는 것이 바람직 할 것이다. 그러나, 특별히 상대방의 공개키를 첫 번째 메시지에서 필요로 하는 경우에는 인증서를 미리 전달하도록 할 수 있다.

첫 번째 메시지를 양쪽의 통신 실체가 동시에 개시하는 경우가 있을 수 있다. 이 경우에 어느 쪽의 *DH* 파라메터가 사용될 것인지 일정한 규칙이 있어야 한다. 하나의 단순한 예는 보다 큰 솟수 P를 갖는 실체의 *DH* 파라메터가 사용 되도록 선택할 수 있다. 두 번째 메시지 이후에는 상호간에 선택된 파라메터를 사용하여 계속 수행하게 한다.

한편, 인증서가 없다면 첫번째 메시지에서 전송되는 *DH* 파라메터를 공격자 C가 임의로 수정하고 재전송할 수 있다. 수정된 파라메터들에 의한 지수 계산이 다음의 서명을 구성한 메시지 검증 과정에서 사용자 A가 발견할 수 있지만, 사용자 B는 수정된 파라메터를 이용하여 불필요하게 지수를 계산하고 메시지를 전송해야 할 것이다. 따라서, *DH* 파라메터가 공개된 값이지만 안전하게 전달하는 어떠한 수단이 있거나, 또는 전송 트래픽을 증가시키지 않고 가능한 한 초기에 불법적 변조를 검증하는 방식이 바람직 할 것이다.

$A \text{---}> B : \{\alpha, p, Y_a\}$
$A <\text{---} B : \{Y_b, Cert_B, S_B\{Y_a, Y_b, K\}\}$
$A \text{---}> B : \{Cert_A, S_A\{Y_b, Y_a, K\}\}$
$Cert_A = (A, Ap, \alpha, p, KR_{CertA}\{A, Ap, \alpha, p\})$
$Cert_B = (B, Bp, \alpha, p, KR_{CertB}\{B, Bp, \alpha, p\})$

(그림 4) 3-패스 방식의 사용 예
(Fig. 4) The example of 3-pass method

(D8) 다른 기법과의 차이

Kerberos 프로토콜은 관용 암호방식에 기반하고 있으며, 여러 가지 응용 분야에서 바람직하지 않는 몇 가지 특징들을 갖고 있다.[5][12] 즉, 동기화된 클럭을 요구하는 타임 스탬프와 온라인 인증 서버를 요구하고 있으며, 클라이언트 및 인증 서버들 사이의 다양한 세션키와 많은 잉여 토큰들을 갖고 있다. 이러한 논제들은 *Bellovin*과 *Merritt*[5]에서 언급되어 있다.

X.509 인증구조는 공개키 암호방식에 기반하여 잘 알려진 국제적 표준 인증 프로토콜이다. 1-단계 및 2-단계 *X.509* 프로토콜은 타임 스탬프를 요구하는 반면에 3-단계 프로토콜은 실제로 타임 스탬프의 가치가 없기 때문에 불필요한 잉여 항목이다. 또한, 1과 2-단계에서 선택적으로 암호화된 데이터 필드의 사용을 제안한 것은 키 교환을 위한 것으로써 두 통신 실체간에 완전한 기밀성을 보장하지 못 할 수 있다. 즉, 적의 있는 통신 상대자가 암호화된 데이터를 자신의 것으로서 다른 통신 상대방들에게 통용시킬 수 있다[13][14]. 그리고, 사용된 서명 시스템이 서명과 암호화 데이터 두 가지 모두를 포함해서 처리해야 하기 때문에 서명 방식에 어떤 제한점을 갖게 된다.

*ISO*에서 정의된 인증 구조에는 타임 스탬프, 순서번호, 그리고 난수를 이용하는 *Challenge/Response* 방식의 다양한 인증 기법들이 제시되어 있다[15]. 공개키 암호방식을 사용하는 *ISO* 인증 구조는 실체가 자신의 비밀 서명키를 사용하여 신분을 인증해 보이는 메커니

증으로서, 특정 메시지를 서명하기 위해서 실체 자신의 비밀 서명키를 사용하고, 그 서명은 공개 검증키를 사용하여 다른 실체에 의하여 검증되는 방식이다. 3-패스 프로토콜에서 메시지의 의도된 수신자 신분과 추가적인 임의의 텍스트 필드 외에 양쪽 통신 실체에 의하여 사용되는 잉여의 두 난수들이 포함되어야 한다. 또한, 키 교환 기능을 갖는 인증 프로토콜 수행을 위하여 서명 부분의 내외부에 선택적으로 텍스트 필드를 이용할 때 *X.509*에서와 유사한 두 통신 실체간의 완전한 기밀성을 보장하지 못한다.

제안된 2-패스와 3-패스 방식은 *X.509* 및 *ISO*의 인증 구조와는 다르게 전송 메시지의 암호화된 데이터, 또는 텍스트 필드를 이용하지 않고 자신의 위치에서 직접 계산하기 때문에 키 자체를 전송 메시지에 포함시킬 필요가 없다. 그리고, 서명에 포함된 정보에서 *K*는 오로지 합법적 통신 상대만이 생성할 수 있기 때문에 두 통신 실체간에 완전한 기밀성을 보상 한다.

인증 기능을 갖는 키 교환 프로토콜에서 두 통신 실체가 공유된 비밀키의 지식을 상호 증명하는 순간 까지는 완전하지 않다. *Needham/Schroeder* 및 *Denning*의 프로토콜[16][17]에서는 연속되는 메시지에서 획득한 비밀키를 사용하여 간접적으로 비밀키의 공유를 증명하기 때문에 상호 실체 인증이 완성되기 전에 비밀키를 이미 공유하게 된다. 2-패스 방식은 상호 실체인증의 완성과 동시에 직접적으로 비밀키의 공유를 증명한다. 제안된 2-패스와 3-패스 방식은 각각의 장점을 갖고 있으며, *Kerberos*, *X.509*, *ISO* 3-패스 인증 구조와 비교하여 물리적으로 안전한 온라인 서버를 필요로 하지 않고, 메시지에 비밀키 자체를 전송할 필요가 없으며, *(D1)* 부터 *(D6)*까지 논의된 각종 재전송 및 신분위장 공격에 대하여 방어할 수 있다.

5. 결 론

본 논문에서는 상호 실체 인증을 기반으로 한 키 교환을 위하여 2-패스 및 3-패스의 두 가지 키 교환 알고리즘을 제시하였다. 키 교환의 안전성은 지수에 의한 모듈러 소수 계산은 비교적 쉬운 반면에 이산 대수를 계산하기가 대단히 어렵다는데 있다. 상호 실체 인증의 안전성은 제안된 구조에서 사용되는 서명 메커니즘에 의존적이다.

제안된 알고리즘은 *Kerberos*, *X.509* 3-단계 교환, 및 *ISO* 3-패스 프로토콜보다 비밀 키를 암호화된 형태로 전송 할 필요가 없이 직접 계산할 수 있다는 장점을 갖는다. 이러한 구조는 간단하고 효과적이며, 다양한 형태의 재전송 및 신분 위장 공격을 방어할 수 있다.

참 고 문 헌

[1] ITU-T Recommendation X.509 : *The Directory - Authentication Framework*, 1993.

[2] W. Diffie and M. Hellman, "*New Directions in Cryptography*," IEEE Transactions on In-formation Theory, Vol. IT-25(6), pp.644-654, Nov., 1976.

[3] W. ford, "*Computer Communications Securi-ty*", Prentice Hall, 1995.

[4] ISO/IEC DIS 10181-2, "*Information Technology-Open Systems Interconnection Security Frameworks for Open Systems : Authentic-ation Framework*", 1993.

[5] S. M. Bellovin, M. Merritt, "*Limitations of the Kerberos Authentication System*", ACM Computer Communication Review, 20(5), pp.119 -132, 1990.

[6] D. Gollmann, T. Beth and F. Mamm, "*Aut-hentication Services in Distributed Systems*", IEEE Computers & Security, Vol.12, No.8, pp. 753-764, 1993.

[7] William Stalling, "*Network and Internetwork Security*", Prentice Hall, 1995.

[8] S. Mendes and C. Huitema, "*A new Approach to the X.509 Framework : Allowing a Global Authentication Infrastructure Without a Gl-obal Trust Model*", Symp. on Network and Distributed System Security, pp.172 189, 1995.

[9] P. J. Bumbulis, D. O. Cowan, C. M. Durance and T. M. Stepien, "*An Introduction to the OSI Directory Services*", Computer Networks and ISDN Systems 26 pp.239-249, 1993.

[10] Y. C. Woo Thomas and S. Lam Simon, "*Aut-hentication for Distributed Systems*", IEEE

Computer, pp.39-52, Jan, 1992.

[11] W. Diffie, "*Authentication and Authenticated Key Exchange*", Designs, Codes and Cryptography, 2, pp.107-125, 1992.

[12] J. Kohl, B.C.Neuman, "*The Kerberos Network Authentication Service*", MIT Project Athena Version 5, 1991.

[13] C. l'Anson, C. Michell, "*Security Defects in CCITT Recommendation X.509-The Directory Authentication Framework*", Computer Communication Review 20(2), pp.30-34, 1990.

[14] M. Burrows, M. Abadi and R.Needham, "*A Logic of Authentication*", ACM Transactions on Computer Systems 8(1), pp.18-36, 1990.

[15] ISO/IEC9798-3 : *Information Technology-Security Techniques-Entity Authentication Mechanisms Part 3 : Entity Authentication Using a Public-Key Algorithm*, Nov., 1991.

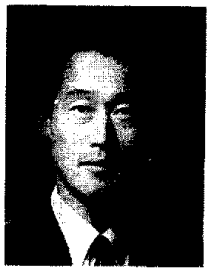
[16] R. Needham, M. Schroeder, "*Using Encryption for Authentication in Large Network of Computers*", Communications of the ACM, Dec. 1978.

[17] D. Denning, "*Timestamps in Key Distribution Protocols*", Communications of the ACM, Aug. 1981.

[18] C.G. Kang and Y.R. Choi, "Mutual Entity Authentication and Key Exchange Schemes", ISITA'96, Vol.2, pp.452-455, Sept. 1996.

강 창 구

1979년 한국항공대학교 전자공학과(학사)
1986년 충남대학교 전자공학과(공학석사)
1993년 충남대학교 전자공학과(공학박사)
1979년~1982년 대한민국 공군장교
1987년~현재 한국전자통신연구원 책임연구원 부호3팀장
1997년~현재 한국통신정보보호학회 충청지부 부지부장
관심분야 : 부호 및 통신이론, 정보보호 이론, 디지털 서명, 전산 및 통신 정보보호 기술



최 용 락

1976년 중앙대학교 전자계산학과(학사)
1982년 중앙대학교 전자계산학과(석사)
1989년 중앙대학교 전자계산학과(박사)
1982년~1986년 한국전자통신연구원 선임연구원
1986년~현재 대전대학교 컴퓨터통신전자공학부 교수
1997년~현재 한국통신정보보호학회 충청지부 지부장
관심분야 : 운영체제, 컴퓨터통신보안, 접근제어,보안 API