비밀키를 이용한 새로운 인증 프로토콜에 관한 연구

° 조 진, 남 길현
국 방 대 학 원

A Study on New Authentication Protocol using Secret Key

Jin Cho, Kil-Hyun Nam
National Defense University

요 약

컴퓨터 통신망을 이용하여 메세지를 교환하려는 주체의 신분을 확인하고, 메세지 전송시 예상되는 침입자의 도청을 예방하기위해 메세지의 암호화에 필요한 세션키를 분배하는 것이 인증 프로토콜의 목적이다.

본 논문에서는 Li Gong의 다항식 보간법을 이용한 인증 프로토콜중에서 비밀키의 사전 공유와 인증서버의 고신뢰도의 문제점을 해결할 수 있는 새로운 인증 프로토콜을 제안하였으며, 제안된 프로토콜의 정확성은 GNY로직을 이용하여 분석하였다.

1. 서 론

컴퓨터 통신망에서의 사용자 인증은 통신하고자 하는 주체간의 신분을 확인하는 것을 목적으로 하는 인증 프로토콜(Authentication Protocol)로서, 제3자(Intruder)의 도청이나 신분위장 등을 방지하기위해 암호화 방법으로 설계되어지며, 아울러 이러한 신분확인 기능에 부가하여 현재 세션에서 메세지를 암호화하는데 이용할 세션키를 분배하는 것도 인증프로토콜의 목적에 포함하고 있다.[모송95]

암호화 방법을 기반으로 설계된 인증 프로토콜은 암호화 방법에 따라 비밀키(Symmetric-key or Private-key) 방식과 공개키(Asymmetric-key or Public-key)방식의 인증 프로토콜로 나눌 수 있으며, 비밀키 방식의 인증 프로토콜은 인증서버가 세션키를 분배하기 전에 인증서버와 통신하고자 하는 주체만이 알고 있는 비밀키를 공유하고 있어야 하므로 인증 이전에 인증자체를 위한 비밀키의 분배가 이루어져야 한다. 따라서 분배된 비밀키가 침입자의 여러가지 공격적인 방법에 의해 노출되어지면 비밀키의 변경이 불가피하며, 만약 노출된 사실을 은폐한다면 중요 메세지에 사용되는 세션키 또한 노출되는 문제점을 안고 있다. 그리고 공개키 방식의 인증 프로토콜은 통신하고자 하는 상대의 공개키를 인증서버의 비밀키로 암호화하여 전송하므로 전송되는 메세지는 보안성을 보장받지 못한다. 그러므로 본 논문에서는 인증자체를 위한 키 분배, 노출된 키의 지속적인 사용, 그리고 키의 생성 및 분배를 전담하고 있는 인증서버(또는Third-Party)의 고 신뢰성에 따른 문제점을 해결할 수 있는 인증 프로토콜로서, Li Gnog의 다항식 보간법을 이용한 비밀키 방식의 인증 프로토콜에 Diffie-Hellman의 공개키 분배 방식을 도입한 새로운 세션키 분배 및 인증 프로토콜을 설계하고, 이 프로토콜의 안전성 및 정확도를 분석한다.

2. 보간법을 이용한 NED-A 프로토콜

Li Gong은 Needham-Schroeder의 비밀키 방식의 인증 프로토콜을 바탕으로 다항식 보간법과 안전한 키로 구성된 일방향 해쉬함수를 사용하여 암호의 효율성을 증대시킨 NED-A 프로토콜을 제안하였다.[Gong94]

2.1 기본적인 알고리즘

NED-A 프로토콜은 Needham-Schroeder의 비밀키 방식 인증 프로토콜을 배경으로 하였으므로 인증서버 S는 사용자 A와는 비밀키 K_{as} ,B와는 K_{bs} 를 공유하고 있다고 가정한다. 그리고 S는 A, B에게 세션키를 분배하기위해 세션키 K를 선택한 다음 a_1 , a_2 에 대한 다음 방정식을 만든다.

$$K + a_1 \times 1 + a_2 \times (1)^2 = K_{as}$$
$$K + a_1 \times 2 + a_2 \times (2)^2 = K_{bs} \quad (1)$$

(위 식은 다항식 보간법의 일반식 $P(x) = a_0 + a_1 x^1 + \cdots + a_n x^n$ 의 형태로서 2차 항까지만 사용됨)

여기서 모든 계산은 mod(q)로 계산되며, q는 큰 소수이다.[Knut81] 그리고 $f(x) = K + a_1 \times x + a_2 \times (x)^2$ 식을 이용하여 인증서버 S는 다음식(2)와 같이 $f(3)$ 과 $f(4)$ 를 계산하여 A와 B에게 보내게 되면, A는 다음 세 식을 이용하여 세션키 K를 계산할 수 있다. 즉

$$K + a_1 \times 1 + a_2 \times (1)^2 = K_{as}$$
$$K + a_1 \times 3 + a_2 \times (3)^2 = f(3) \quad (2)$$
$$K + a_1 \times 4 + a_2 \times (4)^2 = f(4)$$

위의 방정식을 풀기위해 보간법을 사용한다. 여기서 제3의 공격자는 K의 값을 알지 못한다. 왜냐하면 한 쌍의 a_1 과 a_2 가 있어서 위의 방정식을 안전하게 할 것이다. 그러나 식(1)과 (2)의 집합으로 쉽게 다음과 같은 관계를 도출할 수 있다. 즉 $K_{as} - 3 \times K_{bs} = f(4) - 3 \times f(3)$ 가 된다.그래서 키 분배는 단지 한번의 세션으로 제한하여야 하는데 이에 대한 하나의 해결책으로 안전한 키를 가진 일방향 해쉬함수를 사용하는 것이다.

여기서 일방향 해쉬함수는 효율성과 키에 대한 보안성을 위하여 다음을 만족하도록 해야 한다.[Gong94]

① k 와 x 가 주어지더라도 $h(k,x)$ 를 계산하기 쉬워야 한다.

② k 와 $h(k,x)$ 가 주어지면 x 를 계산하기 어려워야 한다.

③ k 를 모르면 어떤 x 에 대해서도 $h(k,x)$ 를 계산하기 어려워야 한다.

④ k 를 알아도 다음식의 x,y 를 찾기가 힘들어야 한다. ($h(k,x) = h(k,y),\ x \neq y;$)

⑤ x 와 $h(k,x)$ 의 쌍을 알고 있어도 k 를 계산하기 어려워야 한다.

이러한 일방향 해쉬함수를 식(1)에 적용하면 다음과 같다.

$$K + a_1 \times 1 + a_2 \times (1)^2 = h(K_{as}, r)$$
$$K + a_1 \times 2 + a_2 \times (2)^2 = h(K_{bs}, r) \quad (3)$$

(r은 S가 선택한 난수(Random Number))

그러나 위의 식(3)에 대한 내부공격으로 A는 $h(K_{as}, r)$ 을 계산할 수 있기 때문에 K_{bs} 의 값을 추측할 수 있다. 그러므로 r 을 재 사용하면 더욱 쉽게 세션키를 획득할 수 있다. 또한 외부 공격자는 $h(K_{as}, r) - 3 \times h(K_{bs}, r) = f(4) - 3 \times f(3)$ 의 관계를 이용하여 $h(K_{as}, r)$ 과 $h(K_{bs}, r)$ 을 계산할 수 있으므로 난수 r 을 같은 사용자에 대해 다중 인증세션에 사용해서는 안된다.[Gong94] 따라서 내·외부의 침입자가 쉽게 공격하지 못하도록 식(1)에서 다항식에 사용된 $x(1,2)$ 대신에 해쉬함수를 사용하는 것이다.

$$K + a_1 \times h(K_{as}, r) + a_2 \times (h(K_{as}, r))^2 = h(K_{as}, r+1)$$
$$K + a_1 \times h(K_{bs}, r) + a_2 \times (h(K_{bs}, r))^2 = h(K_{bs}, r+1) \quad (4)$$

2.2 Nonce를 사용한 NED-A 프로토콜

위에서 설계된 식(4)을 이용하여 Nonce(Time-stamp, Random-number)와 개인식별수를 포함하는 NED-A의 프로토콜은 다음과 같은 과정으로 이루어진다.

1. A → B : I_a, I_b, N_a
2. B → S : I_a, I_b, N_a, N_b
3. S → B : $r, f(1), f(2), h(K_{as}, N_a, K, r, I_a, I_b), h(K_{bs}, N_b, K, r, I_b, I_a)$
4. B → A : $r, f(1), f(2), h(K_{as}, N_a, K, r, I_a, I_b), h(K, N_a, I_a, I_b), N_b$
5. A → B : $h(K, N_b, I_b, I_a)$

메세지1에서 A는 자신과 B의 식별수인 I_a, I_b 와 함께 Nonce를 B에게 보내면, B는 자신의 Nonce를 추가하

여 S에게 보내게 된다. 그러면 S는 세션키 K와 Nonce r 을 선택하고 $a1$ 과 $a2$ 를 이용한 다음 방정식(5)을 만든 후 $f(1)$ 과 $f(2)$ 를 계산하여, 메세지3의 내용을 사용자 B에게 보낸다.

$$K + a1 \times h(Kas, Ia, r) + a2 \times (h(Kas, Ia, r))^2 = h(Kas, Ia, r+1)$$
$$K + a1 \times h(Kbs, Ib, r) + a2 \times (h(Kbs, Ib, r))^2 = h(Kbs, Ib, r+1) \quad (5)$$
$$K + a1 \times 1 + a2 \times (1)^2 = f(1)$$
$$K + a1 \times 2 + a2 \times (2)^2 = f(2)$$

여기서 $h(Kas, Ia, r)$ 와 $h(Kbs, Ib, r)$ 의 해쉬함수에 사용되어진 Ia, Ib 는 만약에 $Kas = Kbs$ 와 같은 경우가 발생하게 될 경우에도 방정식에 사용된 변수가 같지 않도록 하기 위해서이다.[Gong94] 메세지 3을 수신한 사용자 B는 메세지 4의 내용을 A에게 보내게 되며, 이때 A와 B는 보간법을 이용하여 세션키 K를 알게 된다. 또한 A와 B가 Two-way Handshake를 원할 경우에는 메세지 4에서 B는 $h(K, Na, Ia, Ib), Nb$ 를 A에게 보내고 A는 다시 $h(K, Nb, Ib, Ia)$ 를 B에게 보냄으로써 가능하다.

그러나 이 프로토콜은 메세지의 송수신을 원하는 사용자의 의사결정과는 상관없이 모두 인증서버가 생성한 비밀키와 세션키를 그대로 믿는 수동적인 프로토콜로서 다음과 같은 문제점이 있다.

① 인증 프로토콜 실행전에 사용자와 인증서버간에 비밀키를 공유하고 있어야하며, 이러한 비밀키의 반복사용은 제3자에게 쉽게 노출될 수 있고, 노출시 키의 변경을 위해 키 분배 프로토콜을 재 수행해야 하는 어려움을 안고 있다.

② 인증서버는 각 사용자의 비밀키를 관리해야 하므로 많은 메모리가 필요하며, 외부 침입자의 다양한 공격에 대비한 디렉토리의 관리가 어려울 뿐 아니라, 인증서버의 내부 침입자에 대한 특별대책이 필요하다.

③ 인증서버가 사용자들의 세션키를 생성, 분배하기 때문에 언제던지 정보의 도청이 가능하므로, A와 B간에 전송되는 메세지를 인증서버가 도청할 수 없는 Two-way Handshake를 위한 세션키 획득방안이 없다.

그러므로 인증서버를 이용하는 사용자들은 침해자들의 다양한 공격으로부터 세션키에 대한 노출의 위험이 없어야 하며, 인증서버를 통해 인증을 하되 인증서버에게도 노출되지 않는 세션키를 이용한 Two-Way Handshake를 달성할 수 있어야 한다. 또한 인증서버는 인증 이전의 비밀키 분배와 각 사용자들의 비밀키 관리를 위한 접근제어, 그리고 메모리 관리의 어려움을 해결할 수 있어야 한다.

3. 새로운 인증 프로토콜 제안

본 장에서는 NED-A 인증 프로토콜의 문제점들 - 특히 인증 및 세션키 분배 이전에 인증서버와 통신대상자간에는 비밀키 공유와 공유된 비밀키가 제3자의 공격에 노출된 후에도 통신대상자는 이를 인식하지 못하고 비밀키를 계속사용하는 것 - 을 해결하고, 통신대상자 A와 B만의 Two-way Handshake를 위한 세션키 획득을 가능하게 하는 새로운 프로토콜을 소개 한다.

3.1 Diffie-Hellman의 키 분배 프로토콜

본 논문에서는 Li Gong의 NED-A 프로토콜에 Diffie-Hellman의 키 분배 방식을 도입한 새로운 프로토콜을 제안하고자 하므로 먼저 Diffie-Hellman의 공개키 방식을 간단히 살펴본다. Diffie-Hellman의 공개키 방식은 통신 대상자간의 세션키를 안전하게 공유하기 위한 것으로, 큰 소수 P와 P의 원시근 g는 공개정보이다. 따라서 각 사용자 A,B는 각각의 비밀키 Ka, Kb 를 선택한 후 A는 $Ya \equiv g^{Ka} \bmod P$, B는 $Yb \equiv g^{Kb} \bmod P$ 를 계산하여 서로에게 전송한다. 그러면 두 사용자의 비밀키는 $Kab = g^{Ka \cdot Kb} \bmod P = Ya^{Kb} \bmod P = Yb^{Ka} \bmod P$ 의 식으로 얻을 수 있다. 여기서 제3자의 입장에서 Kab 를 계산한다는 것은 이산대수문제의 어려움에 근거하여 대단히 어려운 일이다.[한국91]

3.2 키 분배 및 인증 프로토콜 제안

제안하는 프로토콜은 Li Gong의 NED-A 프로토콜 방식으로 사용하되 비밀키 방식에서 요구하는 통신대상자와 인증서버간의 비밀키 사전공유 조건을 인증 프로토콜 실행중에 자연스럽게 새로운 비밀키를 분배할 수

있도록 하여 키 노출에 대한 위험부담을 최소화하고, 인증서버의 메모리관리를 용이하게하며, 인증서버에서 생성한 세션키를 사용하지 않고 별도의 세션키를 통신 대상자간에 생성할 수 있도록 하여 인증서버에게도 노출되지 않는 Two-way Handshake를 달성하는 새로운 프로토콜이다. 따라서 S는 모든 사용자가 사용할 수 있는 소수 P와 원시근 g, 그리고 각 사용자만을 공개 디렉토리에서 관리하면 되고, 사용자 A,B는 별도의 키 관리가 필요없이 인증 알고리즘만을 가지고 있으면 된다.

본 프로토콜에 사용된 용어는 다음과 같다.

- S : 인증서버(Third-party)
- A,B : 사용자 A,B
- P : 큰 소수 (약 512 bit 정도의 수)
- g : P 의 원시근(160 bit)
- K : 세션키
- r : 난수(random number)
- I_a, I_b : 사용자 A,B의 개인 식별자
- K_a, K_b : 사용자 A,B의 비밀키
- K_{as}, K_{bs} : 사용자 A,B와 S간의 비밀키
- K_{ab} : 사용자 A, B만의 인증값
- Y_a, Y_b : 사용자 A, B의 Diffie-Hellman식의 결과값
- $f(x)$: $f(x) = K + a_1 \times x + a_2 \times x^2$
- $h(x)$: 일방향 해쉬함수(2장에서 제시한 조건을 만족하는 함수)

◆ 제안 프로토콜의 설계

제안하는 키분배 및 인증 프로토콜은 5단계의 메세지로 이루어져 있으며, 사용자가 선택하는 난수는 Diffie-Hellman의 키 분배를 위한 변수로 사용하도록 하였고, 메세지의 내용은 다항식 형태를 구성하는 요소로서 침해자의 공격으로부터 보호받을 수 있도록 해쉬함수를 수행한 결과를 사용하였다.

1. $A \rightarrow B : I_a, Y_a, I_b$
2. $B \rightarrow S : I_a, I_b, Y_a, Y_b$
3. $S \rightarrow B : r, f(1), Y_s, h(K_{as}, K, Y_a, I_a, I_b), h(K_{bs}, K, Y_b, I_b, I_a)$
4. $B \rightarrow A : r, f(1), f(2), Y_s, h(K_{as}, K, Y_a, I_a, I_b), h(K, K_{ab}, I_a), Y_b$
5. $A \rightarrow B : h(K, K_{ab}, I_b)$

▶ 메세지 1

A는 B에게 통신을 원하는 사용자가 A 자신이라는 것을 알려 주기위해 A의 ID(Identification Number) I_a 와 $Y_a \equiv g^{K_a} \bmod P$ 를 보낸다.

▶ 메세지 2

B는 B와 통신을 원하는 A의 ID인 I_a 와 Y_a 를 포함하여, B 자신의 ID인 I_b 와 $Y_b \equiv g^{K_b} \bmod P$ 를 계산하여 인증서버인 S 에게 보낸다.

(여기서 사용된 K_a, K_b 는 사용자 A,B가 임의적으로 선택한 큰 수로서 비밀키이고, Y_a, Y_b 는 Nonce의 성질을 만족한다)

▶ 메세지 3

S는 메세지2를 통해 상호 통신자가 A,B라는 것을 알게 되며, 임의의 큰 수 K_s 를 선택한 후 다음을 계산한다.

$$K_{as} \equiv Y_a^{K_s} \bmod P$$
$$K_{bs} \equiv Y_b^{K_s} \bmod P$$
$$Y_s \equiv g^{K_s} \bmod P$$

그리고 난수 r 을 선택하여 다음 식을 생성하고,

$$K + a_1 \times h(K_{as}, I_a, r) + a_2 \times (h(K_{as}, I_a, r))^2 = h(K_{as}, I_a, Y_s)$$
$$K + a_1 \times h(K_{bs}, I_b, r) + a_2 \times (h(K_{bs}, I_b, r))^2 = h(K_{bs}, I_b, Y_s) \quad (6)$$

$f(1)$ 은 $f(1) = K + a_1 \times 1 + a_2 \times 1^2$ 으로 얻는다. 그리고 다항식의 변수 a_2 는 $a_2 = h(K_{bs}, I_b, Y_b)$ 의 식으로 계산된 값을 사용함으로 NED-A 프로토콜에서 보다 $f(2)$ 함수 하나를 줄일 수 있으며, 프로토콜에 사용된 해쉬함수는 2장에서와 같은 해쉬함수의 조건을 만족함으로 침해자의 다양한 공격으로부터 비밀키를 보호할 수 있다. 또한 해쉬함수 $h(K_{as}, K, Y_a, I_a, I_b)$ 와 $h(K_{bs}, K, Y_b, I_b, I_a)$ 를 계산하여 A와 B의 인증함수값으로 사용하기위해 B에게 보낸다.

▶ 메세지 4

S로부터 메세지 3을 받은 B는 $K_{bs} \equiv Y_s^{K_b} \bmod P$, $K_{ab} \equiv Y_a^{K_b} \bmod P$ 의 식으로 K_{bs}, K_{ab} 를 얻는다. 또한 다음 식(7)로 나타나는 다항식에서 $f(1), Y_S$ 를 적용하여 K, a_1, a_2 를 얻을 수 있으며, 해쉬 함수 $h(K_{bs}, K, r, I_b, I_a)$ 는 S와의 인증을 위해 보관한다.

$$K + a_1 \times h(K_{bs}, I_b, r) + a_2 \times (h(K_{bs}, I_b, r))^2 = h(K_{bs}, I_b, Y_s)$$
$$K + a_1 \times 1 + a_2 \times (1)^2 = f(1) \quad (7)$$
$$a_2 = h(K_{bs}, I_b, Y_b)$$

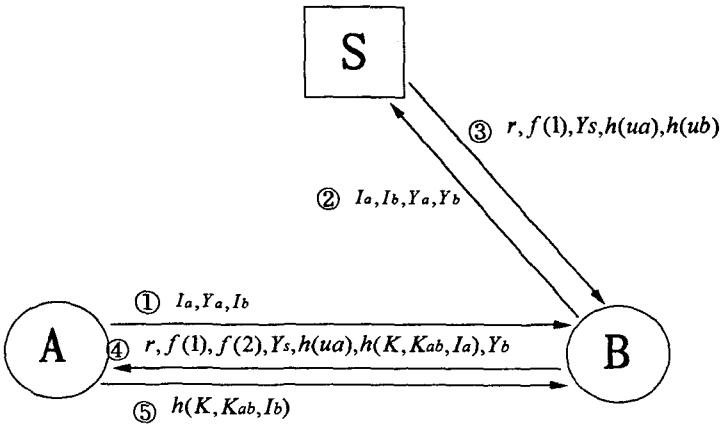
그리고 보간법으로 얻어진 K, a_1 의 값을 이용하여 $f(2)$ 와 해쉬함수 $h(K, K_{ab}, I_a)$ 를 계산하여 A에게 보내며, 특히 해쉬함수 $h(K, K_{ab}, I_a)$ 는 B가 가지는 A의 인증 함수값으로 사용한다.

▶ 메세지 5

A는 B에게서 온 메세지중 $f(1), f(2)$ 와 Diffie-Hellman의 키 분배식으로 얻을 수 있는 K_{as}, K_{ab} 를 이용하여 K, a_1, a_2 를 구할 수 있으며, 해쉬함수 $h(K, K_{ab}, I_b)$ 는 B의 인증함수값으로, $h(K_{as}, K, Y_a, I_a, I_b)$ 는 S의 인증 함수값으로 사용한다. 그러므로 A는 S,B에게서 온 해쉬함수를 B는 S,A에게서 온 해쉬함수를 각각 보관한다. 그리고 A와 B가 메세지 전송시에는

$$K_{ab} \equiv Y_b^{K_a} \bmod P$$
$$K_{ab} \equiv Y_a^{K_b} \bmod P$$

를 서로 계산한 후 $h(K_{ab}, Y_s)$ 를 세션키로 사용하여, 인증서버에게도 노출되지 않는 비밀 세션키를 얻는다. 지금까지 설명한 내용을 도식화하면 <그림 3-1>과 같다.



※ $ua = K_{as}, K, Y_a, I_a, I_b$, $ub = K_{bs}, K, Y_b, I_b, I_a$

<그림 3-1> 제안된 인증 프로토콜

앞의 그림으로 알 수 있듯이 제안된 프로토콜은 각 사용자와 인증서버 S간의 비밀키를 인증 프로토콜 실행시마다 자연스럽게 새로운 키를 생성할 수 있어 인증 이전의 키 분배의 문제점과 키 노출에 따른 위험성을 해결하였고, Diffie-Hellman의 키 분배 방식을 사용하여 인증서버의 효율적인 디렉토리 관리를 도모하였으며, A와 B만의 비밀 세션키를 생성, 사용하도록 함으로써 인증서버의 불법적인 도청을 예방하여 메세지의 보안성을 향상시켰고, 보간 다항식의 변수를 사전 지정하여 줌으로써 알고리즘의 수행 속도를 향상시켰다.

3.3 NED-A 인증 프로토콜과의 비교

새롭게 제시한 프로토콜을 앞에서 제시한 NED-A 프로토콜과 비교하여 보면 다음과 같다.

NED-A 키 분배 및 인증 프로토콜	제안된 키 분배 및 인증 프로토콜
NED 1. A → B : Ia, Ib, Na	NDH 1. A → B : Ia, Ya, Ib
NED 2. B → S : Ia, Ib, Na, Nb	NDH 2. B → S : Ia, Ib, Ya, Yb
NED 3. S → B : $r, f(1), f(2),$ $h(Kas, Na, K, r, Ia, Ib),$ $h(Kbs, Nb, K, r, Ib, Ia)$	NDH 3. S → B : $r, f(1), Ys,$ $h(Kas, K, Ys, Ia, Ib),$ $h(Kbs, K, Ys, Ib, Ia)$
NED 4. B → A : $r, f(1), f(2),$ $h(Kas, Na, K, r, Ia, Ib),$ $h(K, Na, Ia, Ib), Nb$	NDH 4. B → A : $r, f(1), f(2), Ys,$ $h(Kas, K, Ys, Ia, Ib),$ $h(K, Kab, Ia), Yb$
NED 5. A → B : $h(K, Nb, Ib, Ia)$	NDH 5. A → B : $h(K, Kab, Ib)$

(A) NED-A 키 분배 및 인증 프로토콜 (B) 제안된 키 분배 및 인증 프로토콜

NED-A 인증 프로토콜은 사용자와 인증서버간에 비밀키를 인증이전에 공유하고 있어야 한다. 그러므로 인증 프로토콜 외에 별도의 키 분배 프로토콜이 필요하며, 비밀키가 침입자에게 노출된 후에도 사용자는 비밀키가 노출된 사실을 모르고 계속해서 사용하는 위험이 있다. 그리고 NED3에서 인증서버 S가 두 사용자 A,B 간의 세션키 (K) 를 만들어 준다. 따라서 차후에 A,B간에 전송되는 모든 암호화된 정보는 S에게 도청되어질 수 있다.

반면에 새롭게 설계된 인증 프로토콜은 첫째 NDH(Ned-a, Diffie, Hellman)1에서 난수 Na 대신에 Ya 를 사용함으로써 난수의 효과를 가지는 동시에 인증서버 S 와의 비밀키를 생성할 수 있는 바탕이 되며, NDH2에서도 Nb 대신에 Yb 를 사용하였다. 둘째 NDH3에서는 비밀키 Kbs 를 Ys 와 Kb 를 이용하여 생성하도록 하여 매통신시 마다 자연스럽게 비밀키가 변경되어진다. 셋째 NDH3에서 a_2 를 $a_2 = h(Kbs, Ib, Yb)$ 의 식으로 계산할 수 있도록 하여 NED-A 프로토콜보다 보간다항식의 수를 줄여 수행 속도를 증진시켰다. 넷째 NDH4와 NDH5에서는 Diffie-Hellman의 키 분배 방식을 이용하여 사용자 A,B 만이 얻을 수 있는 Kab 를 상호인증인수로 사용하여 신뢰도를 증대시켰으며, 특히 인증서버에게도 비밀로 할 수 있는 Two-Way Handshake용 세션키의 인수로도 사용할 수 있어 정보의 보안성 및 무결성을 향상시켰다.

따라서 제안된 프로토콜에서는 다른 비밀키 방식의 인증 프로토콜과는 달리 인증서버와 사용자가 비밀키를 동시에 생성하는 능동적인 프로토콜이며, 인증서버가 물리적, 논리적으로 안전하다는 가정을 하지않더라도 두 사용자만의 세션키를 생성, 사용하도록하여 메세지의 보안성을 향상시킨 새로운 프로토콜이다.

4. GNY로직을 이용한 제안 인증 프로토콜의 분석

설계된 프로토콜이 바르게 동작하는지를 검증,평가하는 것을 프로토콜 분석이라 하는데 최근 이러한 프로토콜 분석에 대한 연구가 활발히 진행되고 있으나 본 논문에서는 형식로직을 이용하여 프로토콜의 정확성을 분석하는 BAN 로직을 소유와 인식성 개념을 추가하여 제시한 Gong, Needham, Yahalom의 GNY 로직으로 제안된 프로토콜을 분석한다.

4.1 GNY Logic

GNY 로직은 BAN로직이 가지는 문제점 - BAN 로직의 모든 규칙은 프로토콜 참여자가 믿고 있는 것을 처

리할 수 있도록 정의되었기 때문에 참여자가 무엇을 알고 있는가를 표현하는 방법이 결여되어 있음 - 을 보완하기 위해 각각의 참여자에 대해 믿음집합(Belief Set)과 소유집합(Possession Set)을 유지하도록 하였으며, 인식성이라는 새로운 개념을 추가하였다. 즉 인식성이란 메세지 수신자가 자신이 받을 메세지의 내용을 예측하고 있음을 의미한다. [GNY90]

GNY 로직을 이용한 분석과정은 다음과 같다.[모송95]

① 원래의 인증 프로토콜을 GNY 로직에서 제시한 구조체를 이용하여 이상화된 프로토콜(Idealized Protocol)로 바꾼다.

② 초기 가정 및 조건을 GNY 로직의 Formular로 작성한다.

③ 가정과 이상화된 프로토콜의 각 단계에 로직의 규칙을 적용하여 참가자의 믿음을 도출한다.

④ 인증의 목적에 해당하는 Formular를 도출할 수 있을때까지 위 ③의 과정을 반복하며, 더 이상 논리적 규칙을 적용할 수 없을때에는 프로토콜의 정확성이 결여된 것으로 간주하고 종료한다.

◆ 기본 표기 (Basic Notation)

GNY로직에서 사용되는 기본적인 표기방식은 다음과 같다.

▶ 형식(Formulae)

① X : 메세지 또는 문장 ② F(X) : X를 변수로하는 함수

③ H(X) : X를 입력변수로 하는 해쉬함수 ④ (X,Y) : X와 Y를 포함하는 형식

⑤ <S> : 노출되지 않은 S (※ 예를 들면 (X,<S>)는 S를 이용해서 만들어진 X라 한다.)

⑥ * : *기호가 있는 메세지를 수신하는 사용자는 그 메세지를 받았거나, 생성한적이 없다.

⑦ $(X \succ C)$: C는 X의 선행조건이다.

▶ 문장(Statements)

① P ◁ X : 누군가 X를 P에게 보냈다. ② P ∋ X : P는 X를 소유한다.

③ P |≡ $\varphi(X)$: P는 X를 인식한다고 믿는다. ④ P |~ X : P가 X를 보냈다.

⑤ P |⇒ X : P가 X에 대한 판단권을 가진다. 즉 P가 X를 통제한다.

⑥ P |≡#(X) : P는 문장 X가 새로운 것(fresh)이라 믿는다.

⑦ P |≡ $P \xleftrightarrow{S} Q$: P는 S가 P와 Q사이에 안전한 키라고 믿는다.

◆ 논리적 원칙 (Logical Postulates)

GNY 로직은 Being-Told Rules 6가지, Possession Rules 8가지, Freshness Rules 11가지, Recognizability Rules 6가지, Message Interpretation Rules 7가지, Jurisdiction Rules 3가지, 그리고 Not-originated-here Rules 3가지 등으로 이루어져 있다. 여기서는 GNY 로직의 Rule중에서 제안된 프로토콜 분석하는데 사용되는 규칙만을 기록한다.

(1) Being-Told Rules

T1 $\dfrac{P \triangleleft *X}{P \triangleleft X}$ "Not-originated-here" 기호(*)가 있는 메세지 X는 *가 없는 메세지의 특별한 경우이다.

(2) Possession Rules

P1 $\dfrac{P \triangleleft X}{P \ni X}$ P는 그가 받는 메세지 X는 모두 소유한다.

P8 $\dfrac{P \ni -K, P \ni X}{P \ni \{X\}_{-K}}$ P가 메세지(X)와, 비밀키(-K)를 가지고 있으면 그 비밀키로 복호화된 메세지를 소유한다고 할 수 있다.

(3) Freshness Rules

F1 $\dfrac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y),\ P \mid\equiv \#(F(X))}$ P가 X를 새로운 것(fresh)이라 믿으면 X가 포함된 형식 (formula) (X,Y)와 X를 이용한 함수 F(X)도 새로운 것이라 믿는다.

F9 $\dfrac{P \mid\equiv \varphi(X),\ P \mid\equiv \#(-K),\ P\ \ni\ -K}{P \mid\equiv \#(\{X\}_{-K})}$

P가 X를 알고 있다고 믿고, 비밀키를 소유하고, 그 비밀키를 새로운 것이라 믿으면 그 비밀키로 X를 복호화된 것도 새로운 것이라 믿는다.

F10 $\dfrac{P \mid\equiv \#(X),\ P \ni X}{P \mid\equiv \#(H(X))}$

P가 X를 새로운 것이라 믿고, X를 소유하면, 일방향 해쉬함수 H(X)도 새로운 것이라 믿는다.

F11 $\dfrac{P \mid\equiv \#(H(X)),\ P \ni H(X)}{P \mid\equiv \#(X)}$

P가 일방향 해쉬함수H(X)를 가지고 있고 그것을 새로운 것이라 믿으면 X도 새로운 것이라 믿는다.

(4) Recognizability Rules

R1 $\dfrac{P \mid\equiv \varphi(X)}{P \mid\equiv \varphi(X,Y),\ P \mid\equiv \varphi(F(X))}$

P가 X를 인식한다고 믿으면 X가 포함된 어떤 형식도, X를 이용한 함수도 인식한다고 믿는다.

R6 $\dfrac{P \ni H(X)}{P \mid\equiv \varphi(X)}$

P가 해쉬함수H(X)를 소유하면 X를 인식하고 있다고 믿는다.

(5) Message Interpretation Rules

I3 $\dfrac{P \lhd^{*} H(X,<S>),\ P \ni (X,S),\ P \mid\equiv P \overset{S}{\leftrightarrow} Q,\ P \mid\equiv \#(X,S)}{P \mid\equiv Q \mid\sim (X,<S>),\ P \mid\equiv Q \mid\sim H(X,<S>)}$

P가 다음 조건을 모두 만족하면 P는 Q가 S와 연관된 X의 형식을 보냈다고 믿고, Q가 S와 연관된 X의 해쉬함수를 보냈다는 것을 믿는다.

① P가 S와 연관된 X의 Not-originated-here가 표시된 일방향 해쉬함수를 받고
② P가 S와 X를 소유하고 있으며
③ P는 S가 P와 Q사이에 안전한 키라고 믿고
④ P가 X와 S를 새로운 것이라 믿는다.

(6) Jurisdiction Rules

J1 $\dfrac{P \mid\equiv Q \mid\Rightarrow C,\ P \mid\equiv Q \mid\equiv C}{P \mid\equiv C}$

P는 Q가 C를 통제하고, Q가 C를 믿는 것을 믿으면 P도 C를 믿는다.

J2 $\dfrac{P \mid\equiv Q \mid\Rightarrow Q \mid\equiv *,\ P \mid\equiv Q \mid\sim (X \succ C),\ P \mid\equiv \#(X)}{P \mid\equiv Q \mid\equiv C}$

P는 Q가 Q가 믿는 모든 것을 통제한다고 믿고, Q가 X의 선행조건으로 C를 보냈다고 믿고, X가 새로운 것이라 믿으면 P는 Q가 C를 믿는 것을 믿는다.

J3 $\dfrac{P \mid\equiv Q \mid\Rightarrow Q \mid\equiv *,\ P \mid\equiv Q \mid\equiv Q \mid\equiv C}{P \mid\equiv Q \mid\equiv C}$

P는 Q가 믿고 있는 모든것을 Q가 통제한다고 믿고, Q가 C를 믿고 있음을 Q가 믿는 것을 믿는다면, P는 Q가 C를 믿는 것을 믿는다.

◆ 이상화된 프로토콜(Idealized Protocol)

인증 프로토콜을 이상화된 프로토콜로 변환하는 과정은 다음과 같다.

만약 다음과 같은 프로토콜이 있다고 하면 $A \rightarrow B : \{A, K_{ab}\}K_{bs}$

이 프로토콜은 키 K_{bs} 를 알고 있는 B에게 키 K_{ab} 가 A와 통신하는데 사용되는 키라는 것을 말해 주는 것이다. 이 프로토콜을 이상화된 프로토콜 형태로 변환하면 $A \rightarrow B : \{A \overset{K_{ab}}{\longleftrightarrow} B\}K_{bs}$ 가 되고, 이 메세지가 B에게 전달되면 $B \triangleleft : \{A \overset{K_{ab}}{\longleftrightarrow} B\}K_{bs}$ 의 형식을 도출할 수 있다. 그러므로 이상화된 프로토콜을 앞에서 제시한 프로토콜 분석과정을 통해 다음과 같은 믿음을 도출할 수 있으면 인증의 목적을 달성하게 된다.

A believes $A \overset{K}{\longleftrightarrow}$ B, B believes $A \overset{K}{\longleftrightarrow}$ B

A believes B believes $A \overset{K}{\longleftrightarrow}$ B, B believes A believes $A \overset{K}{\longleftrightarrow}$ B

즉 A와 B가 안전한 세션키를 공유하고 있음을 믿고, 그러한 믿음을 가지고 있는 상대방을 믿는 깊까지가 분석의 목적이다.

4.2 제안된 프로토콜 분석

다음은 제안된 인증 프로토콜을 GNY로직으로 분석한 내용이다.

◆ 이상화된 프로토콜로 변환

인증 프로토콜을 분석하기 위해서는 먼저 이상화된 프로토콜 형태로 변환 하여야 하는데 그 변환 과정은 앞에서 설명한 바와 같으며, 제안된 프로토콜을 각 메세지별 이상화된 프로토콜로 변환하면 다음과 같다.

1. $B \triangleleft : {*I_a}, {*Y_a}, I_b$
2. $S \triangleleft : {*I_a}, I_b, {*Y_a}, {*Y_b}$
3. $B \triangleleft : {*r}, {*f(1)}, {*Y_s}, {*h(*K_{as}, *K, Y_a, I_a, I_b)}, \succ S| \equiv A \overset{K}{\longleftrightarrow} B$

   $*h(*K_{bs}, K, Y_b, I_b, I_a) \succ S| \equiv A \overset{K}{\longleftrightarrow} B$
4. $A \triangleleft : {*r}, {*f(1)}, f(2), {*Y_s}, {*h(*K_{as}, *K, Y_a, I_a, I_b)}, \succ S| \equiv A \overset{K}{\longleftrightarrow} B$

   $*h(K, *K_{ab}, I_a), Y_b \succ S| \equiv A \overset{K}{\longleftrightarrow} B, B| \equiv A \overset{K_{ab}}{\longleftrightarrow} B$
5. $B \triangleleft : *h(K, K_{ab}, I_b) \succ A| \equiv A \overset{K}{\longleftrightarrow} B, A| \equiv A \overset{K_{ab}}{\longleftrightarrow} B$

(여기서 $P \succ Q$ 는 Q가 P의 전제조건임을, *기호는 *기호가 있는 메세지는 그러한 메세지를 받는 수신자가 소유하지 않은 것을 나타낸다.)

◆ 초기 가정사항

$A \ni K_a, Y_a \quad A| \equiv \#(Y_a),\ A| \equiv \#(K_a),\ A| \equiv \varphi(B)$

$B \ni K_b, Y_b, f(2) \quad B| \equiv \#(Y_b),\ B| \equiv \#(K_b),\ B| \equiv \#(f(2)),\ B| \equiv \varphi(A)$

$S \ni K_s, Y_s, K, r, f(1)$

$S| \equiv \#(Y_s),\ S| \equiv \#(K_s),\ S| \equiv \#(K),\ S| \equiv \#(r),\ S| \equiv \#(f(1))$

$S| \equiv A \overset{K}{\longleftrightarrow} B$

◆ 제안 프로토콜 분석

▶ 메세지 1

$B \ni (I_a, Y_a, I_b)$: B는 수신된 메세지를 소유한다. (T1,P1 적용)

▶ 메세지 2

$S \ni (I_a, I_b, Y_a, Y_b)$: S는 수신된 메세지를 소유한다. (T1,P1 적용)

▶ 메세지 3

. $B \ni \{r, f(1), Y_s, h(K_{as}, K, Y_a, I_a, I_b), h(K_{bs}, K, Y_b, I_b, I_a)\}$ (T1, P1적용)

여기서 B는 $K_{bs} = Y_s^{K_b} \bmod P$ 로 K_{bs} 를 구한 다음 $f(1)$ 을 사용한 보간다항식으로 K를 구할 수 있다. 그러므로 P8을 적용하여 $B \ni (r, f(1), Y_s, K_{bs}, K)$ 를 얻는다.

. $B \mid\equiv \varphi\{r, f(1), Y_s, K_{bs}, K, Y_b, I_b, I_a\}$ (R6, R1 적용)

B는 메세지의 내용을 인식한다고 믿는다.

. $B \mid\equiv \#\{r, f(1), Y_s, h(K_{bs}, K, Y_b, I_b, I_a)\}$ (F9, F10, F11, F1 적용)

B는 메세지의 내용이 새로운 것(fresh)이라 믿는다. 그러므로 이 메세지는 재 전송된 것이 아니다.

. $B \mid\equiv S \mid\sim \{r, f(1), Y_s, h(K_{as}, K, Y_a, I_a, I_b), h(K_{bs}, K, Y_b, I_b, I_a)\}$ (I3 적용)

B는 메세지의 내용이 S로부터 온 것임을 믿는다.

. $B \mid\equiv S \mid\equiv S \mid\equiv A \overset{K}{\longleftrightarrow} B$ (J2 적용)

B는 S가 A와 B사이의 안전한 키가 K임을 믿는 것을 믿고 있음을 믿는다.

. $B \mid\equiv S \mid\equiv A \overset{K}{\longleftrightarrow} B$ (J3 적용)

B는 S가 A와 B사이의 안전한 키가 K임을 믿는 것을 믿는다.

. $B \mid\equiv A \overset{K}{\longleftrightarrow} B$: B는 A와 B사이의 안전한 키가 K임을 믿는다. (J1 적용)

▶ 메세지 4

. $A \ni \{r, f(1), f(2), Y_s, Y_b, h(K_{as}, K, Y_a, I_a, I_b), h(K, K_{ab}, I_a), Y_b\}$ (T1,P1 적용)

. $A \mid\equiv \varphi\{r, f(1), f(2), Y_s, Y_b, K_{as}, K, Y_a, I_a, I_b, K_{ab}\}$ (R6,R1 적용)

. $A \mid\equiv \#\{r, f(1), f(2), Y_s, Y_b, h(K_{as}, K, Y_a, I_a, I_b)\}$ (F9,F10,F11,F1 적용)

. $A \mid\equiv S \mid\sim \{r, f(1), Y_s, h(K_{as}, K, Y_a, I_a, I_b)\}$ (I3 적용)

. $A \mid\equiv B \mid\sim \{h(K, K_{ab}, I_a), f(2), Y_b\}$ (I3 적용)

. $A \mid\equiv S \mid\equiv S \mid\equiv A \overset{K}{\longleftrightarrow} B$ (J2 적용)

. $A \mid\equiv S \mid\equiv A \overset{K}{\longleftrightarrow} B$ (J3 적용)

. $A \mid\equiv A \overset{K}{\longleftrightarrow} B$ (J1 적용)

. $A \mid\equiv B \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ (J2 적용)

. $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ (J3 적용)

. $A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ (J1 적용)

▶ 메세지 5

. $B \mid\equiv A \mid\sim h(K, K_{ab}, I_b)$ (I3 적용)

. $B \mid\equiv A \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ (J2 적용)

. $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ (J3 적용)

. $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ (J1 적용)

메세지 3과 메세지 4에서 K가 A와 B사이에 안전한 세션키임을 서로 믿을 수 있고, 재전송 되지 않은 것임을 믿을 수 있다. 그러나 그러한 믿음은 인증서버 S를 완전히 믿고 있는 것을 전제로 한다. 따라서 인증서버에 대한 믿음을 배제한다면 메세지 4와 메세지 5에서 A와 B만이 믿을 수 있는 세션키를 얻을 수 있도록

설계 되었다.

지금까지 5단계의 메세지를 분석한 과정를 살펴보면 최초 가정사항에서 인증서버와 각 사용자는 자신이 생성한 내용을 소유하며, 소유된 내용이 새로운 것으로 믿고 있다. 그러므로 사용자가 메세지를 생성할 때는 자신이 소유하고 있는 내용을 포함시켜야 하며, 메세지를 보낼때에는 자신의 믿음이 포함되어 있어야 한다. [GNY90] 따라서 이러한 소유와 믿음을 바탕으로 수신된 메세지에 대해 앞에서 제시한 각 규칙들을 적용하여 메세지의 믿음을 확장하게 되며, 최종적으로 그 프로토콜의 목적을 달성하는지를 분석하게된다.

5. 결 론

본 논문에서는 Needham-Schroeder의 비밀키 방식의 인증 프로토콜을 바탕으로 한 Li Gong의 보간 다항식 인증 프로토콜을 분석하여 비밀키의 노출에 따른 위험을 최소화하고 안전한 세션키 획득을 가능하게 하는 새로운 인증 프로토콜을 제안하였으며, 프로그램은 Windows 95에 포함된 마이크로소프트 네트워크의 환경에서 구현하였다. 제안한 인증 프로토콜의 암호화 방법은 NED-A 프로토콜의 보간 다항식을 이용하되 인증을 위한 비밀키 분배와 분배된 비밀키의 노출에 따른 위험을 최소화할 수 있도록 Diffie-Hellman의 키 분배 개념을 도입하였다. 그러므로 인증 프로토콜이 수행되면서 자연스럽게 인증서버와 사용자가 비밀키와 세션키를 생성할 수 있으며, 이러한 키들은 인증 프로토콜 실행시마다 변경되므로 키가 노출될 위험이 적다. 그리고 인증서버에 대한 사용자의 입장이 비밀키를 동시에 생성하는 능동적인 형태로 이루어짐에 따라 인증서버의 안전성을 고집하지 않아도 된다. 따라서 Two-way Handshake를 위해 두 사용자만의 세션키를 사용하여 인증서버에 대한 신뢰도를 최소화 시켰다.

한편 제안된 프로토콜의 정확성을 분석하기 위해 본 논문에서는 형식로직 분석방법인 GNY 로직을 이용하였으며, 분석결과 인증 프로토콜의 분석 목적을 달성하고 있음을 알 수 있다. 그러나 본 논문에서는 두 사용자에 대한 인증 프로토콜만 제안하고 있으므로, 그룹에서 사용될 수 있는 인증 프로토콜의 설계에 대한 연구와 인증 프로토콜 분석시 비밀성, 복잡도등 다양한 방면에서도 분석이 가능하도록 많은 연구가 이루어져야 할 것이다.

참 고 문 헌

[GNY90] L.Gong, R. Needham, and R.Yahalom, "Reasoning about Belief in Cryptographic Protocols," In Proceedings of the IEEE Symposium on Research in Security and Privacy,Oakland, California, May 1990.

[Gong93] L. Gong, "Increasing Availability and Security of an Authentication Service," IEEE Journal on Selected Areas in Communications, June 1993.

[Gong94] L. Gong, "New protocols for third-party-based authentication and secure broadcast," Poceedings of 2nd ACM Conference on computer and Communications Security, pp.176-183 November 1994, Fairfax,Virginia.

[Knut69] D.E.Knuth, The Art of Computer Programming, Vol.2: Seminumerical Algorithms, Addison-Wesley, Reading, Massachusetts, 1981. Second edition.

[BAN90] M. Burrows, M. Abadi, and R.M. Needham, "A Logic for Authentication," ACM Transactions on Computer Systems, February 1990.

[남길94] 남길현, "정수론", 이산수학, 교학사, 1994

[남길95] 남길현, " 메세지 처리시스템에서 디지탈 다중서명 서비스 구현에 관한 연구," 한국무선국관리사업단, 1995. 8

[모송95] 모영범, 송주석, "반복 인증을 고려한 인증 프로토콜 제안 및 분석," 통신정보보호학회논문지, 제5권 제2호, pp46-54, 1995. 6

[우상92] 우상흠, "컴퓨터 통신에서의 사용자 인증에 관한 연구," 동국대학교 교육대학원 석사 학위논문,1992

[한국91] "현대암호학", 한국전자통신연구소, 1991