通信情報保護學會論文誌
第 7 卷 第 2 號, 1997. 6

패스워드 기반 시스템을 위한 효율적이고 안전한 인증 프로토콜의 설계 및 검증

권 태 경*, 강 명 호*, 송 주 석*,

An Efficient and Reliable Authentication Protocol for Password-based Systems

Taekyoung Kwon, Myeongho Kang, Jooseok Song

요 약

본 논문은 패스워드 기반 시스템을 위해서 안전하고 효율적인 인증 및 키 분배 프로토콜을 제안한다. 기존의 프로토콜들이 패스워드 기반 시스템에서 사용될 경우 패스워드 추측 공격(guessing attack)에 노출되었으며, 이를 저지하기 위해서 다시 설계된 프로토콜들은 부가적인 오버헤드를 수반하였다. 제안하는 프로토콜은 일시 패드(one-time pad)와 강한 성질의 일방향 해쉬 함수를 이용하여, 안전성과 효율성을 도모한다. 따라서 다른 프로토콜과 비교할 때 제안하는 프로토콜은 추측 공격을 포함한 다양한 프로토콜 공격에 대해서 안전하며, 또한 통신량 및 계산량을 최소화하는 면에서 효율적이다.

Abstract

We propose a new authentication and key distribution protocol which is efficient and reliable for password-based systems. Various guessing attacks have been detected in applying conventional protocols to the password-based systems and additional overheads have been made in refined protocols to defeat those attacks. Using a one-time pad and a strong hash function, our proposed protocol promotes reliability and efficiency. Compared with other protocols, our protocol is secure against various protocol attacks including guessing attacks. In addition, this protocol is efficient in reducing communication and computation costs.

Key Words : Security, Authentication, Key Distribution, Password Guessing Attack

---

* 연세대학교 컴퓨터과학과 정보통신연구실

1. 서 론

통신망에서 사용자 정보를 안전하게 전송하도록 하기 위한 연구는 암호학 분야에서 매우 중요하게 다루어지고 있으며, 이것을 달성하기 위해서는 암호화 알고리즘과 함께 암호화 프로토콜이 필요하다.[2] 암호화 프로토콜은 통신의 세션 설정을 위한 초기 단계에 해당되는데, 그 목적은 통신 참여자들에 대한 인증과 사용자 메시지의 암호화를 위한 세션키 분배를 안전하게 이루도록 하는 것이다. 본 논문에서는 두 가지 기능을 모두 갖는 암호화 프로토콜을 인증 프로토콜이라고 부르도록 한다. 인증 프로토콜의 역사를 살펴보면, Needham과 Schroeder는 인증된 통신을 이루기 위해서 암호화를 이용하도록 하는 프로토콜을 처음으로 제안하였으며, 그 개념은 Denning and Sacco, Otway and Rees, Fumy and Munzert 등에서 계승되었다.[2,12,9,8] Diffie와 Okamoto는 공개적인 메시지의 교환만을 통해서 안전한 세션키 분배를 이루는 절차를 각각 제안한 바 있다.[1,11] 현재까지 다양한 종류의 인증 프로토콜들이 제안되었으며, 많은 컴퓨터 통신 시스템에서 활용되고 있다.[15]

대부분의 인증 프로토콜은 통신에 참여하는 사용자의 식별을 위한 고유의 비밀값을 공유한 후 이것을 통해서 인증을 이루게 되며, 따라서 이와 같은 비밀값이 공개될 경우 안전한 통신을 이루는 것이 불가능하게 된다. 실제로 인증 프로토콜이 활용되는 컴퓨터 통신 시스템을 살펴보면, 사용자의 식별을 위해서 사용자가 임의로 선택한 비밀, 즉 패스워드가 사용되는데 이와 같은 시스템을 패스워드 기반 시스템이라고 한다. 그러나 사용자들은 기억하기 쉬운 패스워드를 선택하는 것이 일반적이며 특히 키보드나 키패드 등을 이용하여 선택하므로, 이것은 임의의 난수를 이용하여 만들어진 값보다 추측되기 쉽다. 따라서 이와 같이 약한 비밀(weak secret)을 이용할 경우 인증 프로토콜은 패스워드 추측 공격(password guessing attack)에 노출된다.[5,7]

따라서 이러한 추측 공격에 대응하는 프로토콜들도 다양하게 제안된 바 있다.[10,7,5,6] 그러나 이들은 프로토콜 수행에 필요한 계산량과 통신량의 측면에서 볼 때 추측 공격을 고려하지 않았던 기존의 프로토콜보다 훨씬 낭비적이다.[3] 특히 이와 같은 오버헤드는 프로토콜 자체를 비실용적으로 만드는 한편 이를 축소시키려고 할 경우에는 오히려 다른 취약성을 수반할 가능성이 커진다.[4] 따라서 이와 같은 여러 가지 사항들을 고려하여 새로운 프로토콜을 설계할 필요가 있다.

본 논문에서는 다양한 패스워드 기반 시스템을 위해서 효율적이고 안전한 인증 프로토콜을 새롭게 제안한다. 제안하는 프로토콜은 추측 공격을 포함한 다양한 공격에 대해서 안전하며, 기존 프로토콜들이 갖는 오버헤드를 줄이는 면에서도 효율적이다. 다음 장에서는 본 논문에서 사용하는 표기법에 대한 설명과 함께 인증 프로토콜이 갖는 문제점과 요구사항을 정의하며, 이어지는 3장에서는 새로운 인증 프로토콜을 제안한다. 또한 4장에서는 형식 로직을 이용하여 프로토콜을 검증하는 한편 안전성에 대해서 검증하며, 5장에서는 제안하는 프로토콜을 기존의 다른 프로토콜들과 비교하여 종합적으로 평가한다. 프로토콜의 검증을 위한 형식 로직으로는 BAN 로직[13]을 확장한 GNY 로직[14]을 이용하였다.

2. 인증 프로토콜 설계를 위한 고려사항

본 장에서는 프로토콜의 설계를 위한 고려

사항으로서 먼저 프로토콜 기술을 위해서 사용될 표기법을 설명한 후, 인증 프로토콜이 갖는 문제점과 요구사항을 명확하게 정의한다.

2.1 표기법

A와 B는 통신 참가자의 시스템 주체(system principal)를 의미하며, S는 안전성 센터(trusted center) 또는 서버의 시스템 주체를 나타낸다. 클라이언트-서버 모델로 구성할 경우 주체 A는 클라이언트에 해당되며 S'는 서버에 해당된다. na1은 주체 A가 생성한 첫 랜덤값을 나타내며, 끝에 붙은 숫자는 생성된 순서를 의미한다. 즉, na1은 A가 생성한 첫 번째 난수를, 그리고 na2는 A가 생성한 두 번째 난수를 나타낸다. K_A는 A가 선택한 비밀, 즉 사용자가 선택한 패스워드를, K_S는 S의 공개키를, 그리고 K는 새롭게 생성된 안전한 세션키를 나타낸다. $\{M\}_K$는 K를 암호키로 하여 메시지 M을 암호화한 결과를 나타내며, h(M)은 입력 메시지 M의 해쉬값을 의미한다. f(M)은 미리 정의된 간단한 함수값을 나타내며, |X|는 값 X의 길이를 의미한다. F_A는 A가 전송한 메시지의 부분값을 의미하는데, 부분값이란 어떤 메시지로부터 도출된 일련의 비트스트림을 말한다. ⊕와 ,는 각각 XOR 연산자와 concatenator에 해당된다. 마지막으로 A → B : X는 A가 메시지 X를 B에게 보내는 것을 의미한다.

2.2 문제점과 요구사항

인증 프로토콜에서의 문제점은 다양한 공격과 오버헤드의 잠재성이라고 할 수 있다. 따라서 여기서는 문제점과 프로토콜 설계시 고려해야할 요구사항을 정의한다.

2.2.1 인증 프로토콜에 대한 공격

인증 프로토콜에 대한 공격을 공격자의 행동 형태 측면에서 보면 수동적인 공격 형태와 능동적인 공격 형태로 나눌 수 있다. 수동적 공격은 프로토콜 수행에 관련되지 않는 제 삼자의 프로토콜 도청을 통한 공격에 해당하며, 능동적 공격은 프로토콜 수행에 직접 참여하는 악의적 참여자에 의한 공격에 해당한다. 이때 합법적인 사용자가 공격자가 될 경우를 내부 공격자(inside attacker)라고 하며, 그렇지 않은 경우를 외부 공격자(outside attacker)라고 부른다. 내부 공격자는 프로토콜에 합법적으로 참여할 수 있으므로 세션키를 알 수 있는 가능성이 크다는 면에서 매우 주의해서 대처해야 한다.[3,4]

본 논문에서는 패스워드 기반 시스템에서 안전하게 사용할 수 있는 인증 프로토콜의 설계를 목표로 하므로 추측 공격에 대한 안전성을 중점적으로 다루도록 한다. 추측 공격을 암호분석(cryptanalysis) 측면에서 보면 주로 알려진 평문 공격(known-plaintext attack)이나 선택된 평문 공격(chosen-plaintext attack)의 형태를 통해서 이루어진다고 할 수 있으며, 특히 검증 가능문 공격(verifiable-text attack)의 형태는 전형적인 오프라인 공격 방법에 해당된다.[3] 이 공격은 사용자가 선택한 비밀값이 비교적 좁은 키보드 공간, 숫자 키패드 공간, 그리고 무엇보다 기억하기 쉬운 축소된 단어 공간에서 선택된다는 사실을 이용하여 이루어진다. 즉, 추측 공격이란 공격자가 좁은 공간에서 패스워드에 대한 추측 및 검증을 반복하는 것을 의미하며, 축소된 전사 공격(brute-force attack)이라고 할 수 있는데 그 이유는 공격 대상이 되는 패스워드의 비트 길이를 $|K_A|$라고 할 때 가능한 패스워드의 개수, 즉 패스워드의 선택공간은 이론상 $2^{|K_A|}$가 되지만,

실제로 패스워드가 선택되는 공간은 그보다 작은 공간인 $2^{|G(K_A)|}$가 되기 때문이다(이 경우 $|G(K_A)| < |K_A|$). 이와 같은 공격의 목적은 불법적으로 패스워드를 알아내고 이를 이용하여 인증 및 세션키 분배의 안전성을 파괴하는 것으로서, 공격자의 행동 유형 측면에서 다음과 같은 두 가지 유형으로 분류할 수 있다.

(1) 오프라인 추측 공격(off-line guessing attack) : 공격 대상자의 패스워드를 추측한 공격자는 프로토콜의 정상적인 메시지를 도청하고 저장한 후, 저장된 메시지를 이용하여 추측에 대한 검증을 오프라인으로 반복한다.[3] 이와 같은 형태의 추측 공격은 발견할 수 없으며, 따라서 방어하기 위한 방법은 공격자가 추측한 패스워드를 검증하는데 필요한 계산량을 늘리는 것이다.[5]

(2) 온라인 추측 공격(on-line guessing attack) : 공격 대상자의 패스워드를 추측한 공격자는 추측에 대한 검증을 위해서 온라인으로 반복하여 프로토콜에 참여한다. 온라인 참여를 위해서는 도청한 메시지를 재전송하거나 공격 대상자를 가장하여 위조 메시지를 만든 후 프로토콜에 참여하는 방법으로 이루어질 수 있다. 프로토콜의 서버나 상대방이 검증에 직접 참여하게 되므로, 따라서 방어하기 위해서는 서버측이나 상대방이 추측한 패스워드의 실패 여부를 신속히 발견할 수 있어야 한다.[4]

2.2.2 추측 공격을 막기 위한 오버헤드

추측 공격을 방어하도록 설계된 프로토콜들[10,7,5,6]을 살펴보면 다음과 같이 계산량과 통신량의 측면에서 볼 때 추측 공격을 고려하지 않았던 기존의 프로토콜보다 훨씬 낭비적이다.[3] 이와 같은 오버헤드는 프로토콜 자체를 비실용적으로 만드는 한편 이를 축소시키려고 할 경우에는 오히려 다른 취약성을 수반할 가능성이 커진다는 문제가 있다.[4]

(1) 계산량의 오버헤드 : 암호화 연산 또는 그와 같은 정도의 연산을 필요로 하는 값의 갯수나 크기, 그리고 암호화 연산에 필요한 시간량이 증가하면, 인증 프로토콜의 계산량은 증가한다.

(2) 통신량의 오버헤드 : 메시지에 포함된 값들의 갯수 및 그 크기와 프로토콜 단계의 수가 증가하면, 인증 프로토콜의 통신량은 증가한다.

2.2.3 인증 프로토콜을 위한 요구사항

인증 프로토콜의 설계시 앞에서 언급한 문제점들을 해결하기 위해서 고려해야할 요구사항을 정의하도록 한다.

- 비밀성(confidentiality) : 메시지에 포함된 패스워드 및 세션키를 비롯한 민감한 데이터들은 비밀성이 유지되어야 한다.
- 인증(authentication) : 메시지의 전송자 및 전달자는 명확하게 인증되어야 한다.
- 배달 증명(proof of delivery) : 전송자 또는 전달자는 합법적인 수신자가 바르게 메시지를 받았는지 여부를 확인할 수 있어야 한다.[13]
- 키 공유(key sharing) : 참가자는 서로 같은 세션키를 얻어야 하며 상대방의 수신 여부를 확인할 수 있어야 한다.

- 수정 발견(modify detection) : 불법적인 수정을 알 수 있도록 무결성이 보장되어야 한다.
- 재전송 발견(replay detection) : 논스(nonce) 값이나 타임스탬프를 이용하여 참가자들이 신속하게 재전송 여부를 발견할 수 있어야 한다.[12]
- 인식성(recognizability) : 메시지의 내용을 이해하고 데이터를 구분할 수 있어야 한다.[14]
- 메시지 잉여(message redundancy) : 추측 검증에 필요한 계산량을 증가시키기 위한 잉여 데이터가 메시지에 적절하게 포함되어야 한다. 여기서 잉여란 암호학적 측면에서 제거되어야 할 잉여와는 다른 의미를 가지며, 난수가 이용된다.[3]
- 암호화 값의 최소화 : 랜덤값과 같이 암호화 연산을 필요로 하는 값의 갯수나 크기는 최소화되어야 한다.[7]
- 암호화 연산의 최소화 : 암호화 연산의 양 또는 그 횟수는 최소화되어야 하며, 복잡하고 계산량 많은 알고리즘은 피해야 한다.
- 메시지 크기의 최소화 : 메시지는 그 내용물의 갯수 및 크기를 최소화해야 한다.
- 프로토콜 단계의 최소화 : 프로토콜의 단계수, 즉 통신 횟수는 최소화되어야 한다.[6]

3. 효율적이고 안전한 인증 프로토콜 설계

본 장에서는 제안하는 인증 프로토콜을 설명하도록 한다. 먼저 3.1에서는 프로토콜의 기본적인 개념과 함께 그 핵심 내용을 제안하고, 3.2와 3.3에서는 이를 이용하여 프로토콜을 설계한 후 3.4에서는 프로토콜을 확장하도록 한다. 제안하는 프로토콜은 K1P(k-won protocol)라고 명명한다.

3.1 제안하는 프로토콜의 기본 개념

기존에 제안된 일반 프로토콜들 혹은 추측 공격을 막도록 제안된 새로운 프로토콜들을 살펴보면, 프로토콜의 참여자가 공유하고 있는 비밀을 통해서 참여자를 식별한 후 그 안전성에 근간하여 세션키를 분배하도록 한다. 1장에서 언급한 바와 같이 공유 비밀은 일반적으로 패스워드에 해당한다. 이 때 세션키의 안전한 분배를 위해서 패스워드 혹은 패스워드에 의해 생성된 키가 암호화키로 사용되어왔다.[2,5,6,7,8,9,10,12] 그러나 이로 인하여 야기되는 단점은 오프라인 추측 공격을 막기 위해서 다수의 랜덤값이 메시지에 추가되어야 하는 것이며,[3,5,6,10] 본 논문에서는 이와 같은 단점을 극복하기 위해서 일시 패드와 강한 성질의 일방향 해쉬 함수를 사용하여 세션키를 안전하고 효율적으로 분배하도록 한다. 즉, K1P의 설계를 위한 기본 개념은 키 분배를 위한 암호화 방법으로 일시 패드를 사용하며, 키의 무결성검사를 위해서 해쉬 함수를 사용하도록 하는 것이다. 특히 일시 패드는 해당 세션의 논스값을 이용하여 생성하도록 하며 세션키와 그 크기가 같아야 한다. 평문과 같은 크기의 난수가 일시 패드 역할을 할 경우 결과값은 평문과 통계적 연관성(statistical relationship)을 전혀 갖지 않으며, 따라서 안전하다. 정보의 암호화를 위한 일시 패드 개념의 안전성은 이미 널리 알려져있다.[15]

강한 성질의 일방향 해쉬 함수를 간략히 해쉬 함수라고 부르도록 한다. 추측 공격을 막기 위해서 공개키 암호화를 통하여 인증을 시도하는 것은 필수적이며,[6] 따라서 다음과 같은 기본적인 프로토콜 메시지를 구성할 수 있다.

1. $A \rightarrow S$: $\{na1, na2, K_A \oplus na1\}K_s$
2. $S \rightarrow A$: $na1 \oplus na2 \oplus K, h(K_A \oplus na1, K, na2)$ (3.1)

먼저 시스템 주체 A는 na1과 na2를 S에게 안전하게 전달하기 위해서 S의 공개키로 메시지 1을 암호화하여 보낸다. 따라서 S의 공개키의 안전성에 대한 가정이 필요하며 특히 공개키에 대응하는 비밀키가 노출되지 않도록 해야 한다. S는 이 메시지를 복호화하여 메시지의 내용인 na1, na2, $K_A \oplus$na1을 얻을 수 있으며, 이 값을 메시지 2를 구성하는데 사용한다. S는 먼저 복호화한 값에서 얻은 na1과 $K_A \oplus$na1을 XOR하여 K_A를 얻을 수 있는데, 이것을 지역 기억장치에 저장된 A의 패스워드 값과 비교하여 A를 인증할 수 있다. K_A값이 유효할 경우 S는 메시지 2를 보내서 인증이 무사히 되었음을 A에게 알려준다.

S가 생성한 세션키 K는 na1$\oplus$na2를 일시 패드로 이용하여 암호화되며 메시지 2를 통해서 A에게 전달된다. 따라서 na1$\oplus$na2값과 K값의 크기는 반드시 같아야 (|na1$\oplus$na2|=|K|) 한다. na1$\oplus$na2를 일시 패드로 사용하는 이유는 차후에 세션키 K가 유출되었을 경우, 즉 Denning-Sacco 공격이나 세션키 K를 아는 내부 공격자의 추측 공격에 대해서도 안전하게 하기 위해서이다. na1$\oplus$K를 메시지 2에 포함할 경우에는 세션키 K를 이용하여 오프라인 추측 공격의 가능성이 커진다. 이에 대한 자세한 설명은 4장의 프로토콜 검증에서 다루도록 한다.

해쉬 함수는 na1, na2 그리고 K_A에 대한 메시지 인증 역할을 수행하는데, K_A가 바르게 포함되었다는 것은 결국 A에 대한 인증이 바르게 이루어졌음을 의미한다. 해쉬 함수의 입력값에서 $K_A \oplus$na1은 na2와 함께 입력값 전체가 임의성(randomness)을 갖도록 하는 한편 인증이 바르게 이루어졌다는 결과를 보여주는데 이용된다. 메시지 2에서는 해쉬 함수 입력값으로 키 K를 포함하여, A가 자신의 인증 결과를 검증할 뿐만 아니라 세션키에 대한 인증도 가능하도록 한다. 즉, A는 메시지 2에서 얻은 K값과 난수 na1과 na2, 그리고 패스워드 K_A를 이용하여 해쉬값을 생성한 후 메시지 2의 해쉬값과 비교하여 검증을 이룬다.

메시지 2에서 요구되는 연산은 단지 XOR와 해쉬 함수 계산뿐이며, 오직 na1, na2 그리고 K_A를 아는 시스템 주체만이 (3.1) 메시지를 구성하거나 해쉬값 계산으로 검증할 수 있다. 비록 공격자가 A의 비밀인 K_A를 추측하고 메시지들을 도청할 수는 있지만, 이 정보만으로는 그 추측값이 맞는지의 여부를 검증할 수 없는데 그 이유는 난수 na1과 na2를 찾는 것이 계산적으로 실행 불가능하기 때문이다. 이와 같이 (3.1)을 통하여 인증 및 키 분배를 안전하게 이룰 수 있으며, 세션키 K를 획득하거나 공유한 내부 공격자조차도 K_A에 대한 추측 공격을 성공할 수 없다. 특히 기존 프로토콜들은 메시지 2와 같은 키 분배문에서 불완전한 비밀 K_A를 내용에 포함하거나 암호키로 이용하는 반면,[2,5,6,7,8,10] 본 프로토콜에서는 상대적으로 빠른 XOR 연산과 해쉬 함수 계산만 수행하고, K_A를 해쉬 입력값으로만 사용한다. 이것은 인증 프로토콜의 문제점인 공격 잠재성[3]과 오버헤드 문제를 해결하는 것이다.

3.2 2자(two party) 인증 프로토콜

앞서 제안한 개념을 바탕으로 패스워드 기반 시스템을 위한 2자 인증 프로토콜을 다음과 같이 설계한다.

1. $A \rightarrow S$: $A, \{na1, na2, K_A \oplus na1\}K_s$
2. $S \rightarrow A$: $na1 \oplus na2 \oplus K, h(K_A \oplus na1, K, na2)$ (3.2)
3. $A \rightarrow S$: $h(K_A \oplus na2, K, na1)$

이 프로토콜은 패스워드를 기반으로 하는 클라이언트-서버 환경에서 적용하기에 적합하며 다음과 같이 수행된다. 클라이언트 A가 난수 na1과 na2를 생성하여 새로운 메시지 1을

구성하고, 자신의 이름과 함께 이것을 S에게 보낸다. 서버 S는 이 메시지를 복호화하여 그 내용을 얻은 후 na1과 K_A⊕na1을 이용하여 메시지 1이 A로부터 왔음을 인증한다. 이와 같은 인증에 관한 자세한 내용은 앞의 3.1에서 설명하였다. 인증이 유효하면 S는 새로운 세션키 K를 생성하고 메시지 2를 구성하여 응답을 보낸다. A는 na1과 na2를 이용하여 세션키 K를 얻고, 해쉬값을 만들어 메시지 2의 해쉬값과 비교 검증한다. 이를 통하여 A는 이 메시지가 S로부터 전송되었으며 새로운 메시지에 해당하는지의 여부와 함께 또한 전송중의 변조 여부를 검사할 수 있다. 마지막으로 A는 메시지 3을 구성하여 응답한다. 메시지 3은 A가 메시지 2를 바르게 수신했으며 S가 생성한 새로운 세션키 K를 획득했음을 S에게 알리기 위한 것이다. 메시지 2와 메시지 3에서 사용된 해쉬값은 na1과 na2의 위치가 다르며, 따라서 다른 결과 값을 갖는다. 결국 메시지 2의 해쉬값은 메시지 3과 함께 세션키에 대한 도전-응답(Challenge-Response)의 역할을 담당하게 된다. 이 프로토콜의 특징은 세단계만으로 인증, 키 분배 및 키 수락 여부 확인을 수행할 수 있다는 것이다. 특히 인증은 클라이언트와 서버가 공유하고 있는 약한 비밀, 즉 패스워드에 의존하고 있으며, 공개키에 대한 안전성이 보장될 경우 서버에 대한 인증도 이루어진다고 볼 수 있다.

3.3 3자 상호(three party mutual) 인증 프로토콜

3.1절에서 제안한 개념을 바탕으로 패스워드 기반 시스템을 위한 3자 상호 인증 프로토콜을 다음과 같이 설계한다.

1. $A \rightarrow B$: $\{A, B, na1, na2, K_A \oplus na1\}K_S$.
2. $B \rightarrow S$: $\{A, B, na1, na2, K_A \oplus na1\}K_S$, $\{B, A, nb1, nb2, K_B \oplus nb1\}K_S$ (3.3)
3. $S \rightarrow B$: $na1 \oplus na2 \oplus K$, $h(K_A \oplus na1, K, na2, A \oplus B)$, $nb1 \oplus nb2 \oplus K$, $h(K_B \oplus nb1, K, nb2, A \oplus B)$
4. $B \rightarrow A$: $na1 \oplus na2 \oplus K$, $h(K_A \oplus na1, K, na2, A \oplus B)$, $\{f(F_A), nb1 \oplus nb2\}K$
5. $A \rightarrow B$: $\{f(nb1 \oplus nb2)\}K$

3자 상호 인증 및 키 분배는 안전성 센터에 해당하는 제 삼자를 통해서 이루어진다. 이 프로토콜은 패스워드를 기반으로 하고 안전성 센터의 설치가 가능한 통신망 환경에서 적용하기에 적합하며 다음과 같이 수행된다. 먼저 (3.3)에서 A는 난수 na1과 na2를 생성하여 새로운 메시지 1을 구성하고, 이것을 B에게 보낸다. 메시지에 A와 B를 포함한 이유는 S로 하여금 세션의 참가자를 첫 메시지에서 확증할 수 있도록 하기 위해서이다. B 역시 자신의 메시지를 생성하고, A의 것과 함께 메시지 2를 S에게 보낸다. S는 이 메시지를 복호화하여 그 내용을 얻은 후 설명한 바와 같이 A와 B를 각각 인증한다. 이와 같은 인증에 관한 자세한 사항은 앞의 3.1에서 설명하였다. A와 B의 인증 결과가 모두 유효하면 S는 하나의 세션키 K를 선택한 후 각각에 대한 인증 응답 메시지 3을 구성하여 B에게 응답한다. 여기서 A⊕B가 해쉬 함수의 입력값에 추가되었는데, 그 이유는 A와 B 각각으로 하여금 S가 서로를 모두 인증했다는 사실을 확증하도록 하기 위한 것이다. B는 자신에 대한 응답 메시지를 분석한 후 결과가 유효하면, S로부터 받은 A에 대한 응답을 메시지 4로 A에게 전달한다. 이 때 메시지 1의 부분값을 정해진 함수로 처리한 결과 $f(F_A)$와 이미 구성한 nb1⊕nb2를 세션키 K로 암호화하여 함께 보낸다. 부분값이란 메시지에서 도출된 일련의 비트스트림을 의미하는데, 근원 메시지가 암호화되어 있으면 임의성을 갖는다. nb1⊕nb2는 랜덤값들의 XOR 결과이므로 역시 임의성을 갖는다. 따라서 이 두 값을 키 확인을 위한 도전-응답값으로 이용

한다. 이것은 도전-응답 시도를 위해서 새로운 랜덤값들을 생성해야 하는 기존 프로토콜들과 비교해볼 때 효율적이다.[2,5,6,8,9,10,12] 메시지 4를 받은 A는 자신에 대한 인증 응답 메시지를 분석한 후 얻은 세션키 K로 B의 도전값을 복호화한다. $f(F_A)$가 유효할 경우 f(nb1⊕nb2)를 세션키 K로 암호화한 메시지 5를 B에게 전달한다.

결과적으로 S는 A와 B를 무사히 인증한 후 새로운 세션키 K를 분배해 줄 수 있으며, A와 B는 S에 대한 신뢰를 통하여 서로에 대한 인증과 세션키 공유 여부를 확증할 수 있다.

4. 제안 프로토콜 검증

인증 프로토콜의 검증 방법 중에서 가장 일반적인 형식 로직을 이용할 경우 프로토콜의 기본적인 안전성과 그 기능에 대해서는 검증할 수 있지만, 추측 공격에 대한 안전성과 오버헤드에 대한 명확한 검증은 할 수 없다. 따라서 본 논문에서는 추측 공격에 대해서는 비형식적(informal)인 방법으로 자세하게 검증하는 한편, 기본적인 기능과 안전성 면에서는 BAN 로직을 확장한 GNY 로직을 통해서 검증한다.[13,14] 또한 오버헤드에 대한 평가는 안전성과 기능에 대한 검증을 다루는 본 장과 분리하여 5장에서 다른 프로토콜들과의 비교하여 다루도록 한다.

4.1 추측 공격에 대한 2자 인증 프로토콜 검증

패스워드 추측 공격에 대한 안전성을 검증하기 위해서 프로토콜 (3.2)에 대해서 먼저 오프라인 추측 공격을 시도해 보도록 한다. 또한 이어서 온라인 추측 공격을 시도해 보도록 한다. 오프라인 추측 공격을 위해서는 먼저 프로토콜에서 사용된 메시지들을 도청하여 지역 기억장치에 저장해야 하는데 메시지 1부터 메시지 3까지 총 3개의 메시지가 수집된다. 세션키 K가 유출되었을 경우도 고려하도록 한다.

4.1.1 오프라인 추측 공격

(1) 세션키 K가 유출되지 않았을 경우의 오프라인 추측 공격

공격 대상자 A의 K_A를 추측한 값들의 집합을 구성한다. 이 집합의 원소를 K_A'라고 한다. 추측한 값 K_A'의 진의 여부를 반복적으로 검사하기 위해서는 프로토콜 (3.1)에서 메시지 1을 재구성하거나 메시지 2 또는 메시지 3의 해쉬값을 구할 수 있어야 하는데, 따라서 na1과 na2를 구해야 한다. 그러나 난수인 이들은 K_A보다 큰 공간에서 선택되었으며, 추측을 위해서는 $2^{|na1|} \times 2^{|na2|}$의 계산량이 필요하다. 이것은 계산적으로 수행 불가능하므로 곧 추측 공격이 불가능한 것을 의미한다.

(2) 세션키 K가 유출되었을 경우의 오프라인 추측 공격

공격 대상자 A의 K_A를 추측한 값들의 집합을 구성한다. 이 집합의 원소를 K_A'라고 한다. 획득한 키 K_A'를 이용하여 프로토콜 (3.1)에서 메시지 2의 내용 na1⊕na2⊕K에서 na1⊕na2를 얻는다. K'가 K와 같을 경우, 즉 획득한 키가 실제 키와 같을 경우 다음의 연산이 성립한다.

$$(na1 \oplus na2 \oplus K) \oplus K' = na1 \oplus na2$$

그러나 이 값만으로는 na1과 na2를 얻을 수 없으므로 오프라인 추측 공격에 전혀 도움이 되지 않는다. 따라서 추측한 값 K_A'의 진의 여부를 검사하기 위해서 필요한 계산량은 세션키가 유출되지 않았을 경우와 여전히 같게 되며 세션키 정보의 획득은 추측 공격에 전혀

도움이 되지 않는다. 결과적으로 K1P에 대한 오프라인 추측 공격은 $2^{|na1|}$이상의 계산량을 요구하므로 불가능함을 알 수 있다.

4.1.2 온라인 추측 공격

(1) 메시지 재전송을 통한 온라인 추측 공격

프로토콜 (3.1)에서 먼저 도청한 메시지 1을 S에게 전송한다. S로부터 새롭게 받은 응답 메시지 2'와 도청한 메시지 2의 값을 이용해서 분석할 경우 두 가지 경우의 연산이 가능하다. 도청한 메시지에 포함된 세션키를 K라고 하고 새롭게 얻은 응답 메시지에 포함된 세션키를 K'라고 할 때, 세션키 K가 유출되지 않았을 경우에는 다음과 같은 연산을 하게 되며,

$$(na1 \oplus na2 \oplus K) \oplus (na1 \oplus na2 \oplus K') = K \oplus K'$$

세션키 K가 유출되었을 경우에는 다음과 같은 연산을 하게 된다.

$$(na1 \oplus na2) \oplus (na1 \oplus na2 \oplus K') = K'$$

그러나 이 결과로는 어느 경우에도 새로운 메시지 3'를 구성할 수 없으므로, 온라인으로는 추측한 K_A를 검증할 수 없게 된다. 따라서 na1과 na2에 관한 추측을 시도할 수 밖에 없는데 이것은 $2^{|na1|}$이상의 계산량이 필요하다는 것을 의미하며, 결국 재전송 공격이 불가능함을 수 있다.

(2) 메시지 재구성을 통한 온라인 추측 공격

프로토콜 (3.1)에서 도청한 메시지 2를 분석한 결과와 추측한 패스워드 K_A'를 조합하여 메시지 1'를 재구성한 후 S에게 전송해야 한다. 그러나 메시지 2를 분석한 결과로는 메시지 1'을 쉽게 재구성할 수 없으며, 결국 이를 위해서는 $2^{|na1|} \times 2^{|na2|}$의 계산량이 필요하게 된다.

임의로 생성한 값 na1'및 na2'와 추측한 패스워드 K_A'를 이용하여 메시지 1'을 구성하여 전송할 경우에는 패스워드의 오류 여부를 서버가 발견할 수 있으므로 온라인 추측 공격은 무시할만 하다. 결국 메시지 재구성을 통한 온라인 추측 공격도 불가능함을 알 수 있다.

4.2 추측 공격에 대한 3자 인증 프로토콜 검증

먼저 3자 상호 인증을 위한 K1P(3.6)에 대한 오프라인 추측 공격을 시도하도록 한다. 2자 인증 프로토콜과 마찬가지로 세션키 K가 유출되었을 경우와 그렇지 않을 경우로 나누어 볼 수 있다. 특히 세션키 K를 이미 알고 있는 내부 공격자의 공격이 가능하므로 세션키 K가 유출되었을 경우를 현실적으로 고려할 수 있다. 즉, 내부 공격자 B는 이미 공격 대상자 A와 정상적인 프로토콜 수행을 마친 후 세션키 K를 공유하게 되며, 단지 해당 메시지들을 저장하는 것만으로 세션키 K가 유출되었을 경우의 오프라인 추측 공격을 시도할 수 있다.

4.2.1 오프라인 추측 공격

(1) 세션키 K가 유출되지 않았을 경우의 오프라인 추측 공격

2자 인증 프로토콜의 경우와 마찬가지로 추측한 값 K_A'의 진의 여부를 검사하기 위해서는 메시지 1을 재구성하거나 메시지 4의 해쉬값을 구할 수 있어야 하는데, 여기에는 $2^{|na1|} \times 2^{|na2|}$의 계산량이 필요하다. 이것은 추측 공격이 불가능한 것을 의미한다.

(2) 세션키 K가 유출되었을 경우의 오프라인 추측 공격

2자 인증 프로토콜의 경우와 마찬가지로 공유 세션키 K를 이용하여 메시지 4의 내용 na1⊕na2⊕K에서 na1⊕na2를 얻는다. 그러나

이 값만으로는 na1과 na2를 얻을 수 없으므로 오프라인 추측 공격에 전혀 도움이 되지 않는다. 따라서 추측한 값 K_A'의 진의 여부를 검사하기 위해서 필요한 계산량은 세션키가 유출되지 않았을 경우와 여전히 같게 되며, 이것은 세션키 정보의 획득이 추측 공격에 전혀 도움이 되지 않음을 의미한다.

결과적으로 상호 인증의 경우에도 K1P에 대한 오프라인 추측 공격은 $2^{|na1|}$이상의 계산량을 요구하므로 불가능함을 알 수 있다. 이어서 온라인 추측 공격을 시도해 보도록 한다. 온라인 추측 공격 방법은 역시 직접 인증의 경우와 같이 크게 두 가지로 분류된다.

4.2.2 온라인 추측 공격

(1) 메시지 재전송을 통한 온라인 추측 공격

먼저 B는 도청한 메시지 1과 자신의 새로운 메시지로 구성된 메시지 2'를 S에게 전송한다.

S로부터 받은 응답과 도청한 메시지 1의 값을 이용해서 분석할 경우 역시 2자 인증의 경우와 같은 연산들만 가능한데, 이와 같은 결과로는 추측에 관한 정보를 얻을 수 없으며 결국 오프라인 추측 공격을 시도할 수 밖에 없게 된다. 이것은 $2^{|na1|}\times 2^{|na2|}$의 계산량이 필요하다는 것을 의미하며, 결국 재전송 공격은 불가능함을 알 수 있다.

(2) 메시지 재구성을 통한 온라인 추측 공격

도청한 메시지 4를 분석한 결과와 추측한 패스워드 K_A'를 조합하여 메시지 1'를 재구성한 후 S에게 전송해야 하는데 이를 위해서는 $2^{|na1|}\times 2^{|na2|}$의 계산량이 필요하게 된다.

또한 임의로 생성한 값 na1'및 na2'와 추측한 패스워드 K_A'를 이용하여 메시지 1'을 구성하여 전송할 경우에는 패스워드의 오류 여부를 서버가 발견할 수 있으므로 결국 메시지 재구성을 통한 온라인 추측 공격도 불가능함을 알 수 있다.

4.3 형식 로직을 이용한 K1P의 검증

GNY 로직은 암호화 프로토콜의 수행을 이해하기 위한 체계적인 방법으로서 신뢰성 여부를 검증할 수 있도록 1990년에 발표되었다. 검증 결과 K1P의 3자 상호 인증 프로토콜이 갖는 안전성은 2자 인증 프로토콜의 안전성에 근간하므로 본 논문에서는 K1P 2자 인증 프로토콜의 검증 결과만을 보이도록 한다.

GNY 로직을 통한 검증을 위해서는 먼저 프로토콜을 이상화된 프로토콜(idealized protocol)로 표현하고, 프로토콜이 내포하는 기본적인 가정(assumptions)을 정의해야 하는데, 여기서 이상화된 프로토콜이란 GNY 규칙을 적용할 수 있는 형태의 표현을 의미한다. 가정과 이상화를 마친 후에는 프로토콜의 진행 순서에 따라 메시지마다 GNY 규칙과 가정을 적용하면 된다. [14]에서는 GNY 로직을 통한 검증 방법에 대해서 자세하게 설명하고 있다. 본 논문에서는 GNY 로직에 대한 자세한 설명은 생략하도록 하며, 근간 사항은 [14]를 참고하도록 한다.

4.3.1 이상화된 프로토콜과 가정

먼저 GNY 로직에 따라 K1P (3.2)를 이상화된 프로토콜 형태로 표현하면 다음과 같다.

$$
\begin{aligned}
&1.\ S\triangleleft *A,\ *\{*na1,\ *na2,\ *(*K_A\oplus *na1)\}+K_S\rightarrow A|\equiv A\overset{K_A}{\longleftrightarrow}S\\
&2.\ A\triangleleft *(na1\oplus na2\oplus *K)\rightarrow S|\equiv A\overset{K}{\longleftrightarrow}S,\qquad (4.1)\\
&\qquad *H(*K_A\oplus na1,\ *K,\ na2)\rightarrow S|\equiv A\overset{K}{\longleftrightarrow}S\\
&3.\ S\triangleleft *H(*K_A\oplus na2,\ K,\ na1)\rightarrow A|\equiv A\overset{K}{\longleftrightarrow}S
\end{aligned}
$$

K1P가 내포하는 프로토콜 가정은 다음 표와 같이 GNY 표기법으로 표현된다.

표 1 K1P의 가정

$A \ni K_A$, $A \ni na1$, $A \ni na2$,
$S|\equiv \varnothing(K_A)$, $A|\equiv \varnothing(na1)$, $A|\equiv \varnothing(na2)$,
$A|\equiv \#(K_A)$, $A|\equiv \#(na1)$, $A|\equiv \#(na2)$
$A|\equiv A \overset{K_A}{\longleftrightarrow} S$, $A|\equiv S \Rightarrow S|\equiv *$, $A|\equiv S \Rightarrow A \overset{K}{\longleftrightarrow} S$

$S \ni K_A$, $S \ni -K_S$, $\underline{S \ni K}$, $\underline{S|\equiv A \overset{K}{\longleftrightarrow} S}$
$S|\equiv \varnothing(K_A)$, $S|\equiv \varnothing(K)$, $S|\equiv \#(K_A)$, $S|\equiv \#(K)$,
$S|\equiv \otimes(S)$, $S|\equiv \overset{+K_S}{\longrightarrow} S$, $S|\equiv A \overset{K_A}{\longleftrightarrow} S$, $S|\equiv A \Rightarrow A|\equiv *$
$A \ni (na1 \oplus na2)$, $A \ni (K_A \oplus na1)$
$A|\equiv \varnothing(na1 \oplus na2)$, $A|\equiv \varnothing(K_A \oplus na1)$
$A|\equiv \#(na1 \oplus na2)$, $A|\equiv \#(K_A \oplus na1)$

4.3.2 이상화된 프로토콜에 대한 GNY 규칙 적용

이상화된 프로토콜에 단계적으로 GNY 규칙과 표 1에서 정의된 가정을 적용하여 K1P (3.2)의 검증을 수행하도록 한다. 검증 단계는 생략하여 표기하며 적용되는 규칙은 순서대로 기술한다.

(1) 메시지 1

- 규칙 T1, P1, P3, P8, 가정 $S \ni -K_S$, P3에 의해서 다음 결과를 얻는다.

$$S \ni na1, \quad (4.2)$$
$$S \ni na2, \quad (4.3)$$
$$S \ni K_A \oplus na1$$

- 규칙 R1, 가정 $S|\equiv \varnothing(K_A)$, R1, 그리고 규칙 T1, I2, 가정 $S \ni -K_S$, $S \ni K_A$, $S|\equiv \otimes(S)$, $S|\equiv \overset{+K_S}{\longrightarrow} S$, $S|\equiv A \overset{K_A}{\longleftrightarrow} S$에 의해서 다음 결과를 얻는다.

$$S|\equiv A|\sim(na1, na2, \langle K_A \rangle \oplus na1), \; S|\equiv A|\sim\{na1, na2, \langle K_A \rangle \oplus na1\}_{+K_S}$$

(2) 메시지 2

- 규칙 T1, P1, P3, P5와 가정 $A \ni (na1 \oplus na2)$에 의해서 다음 결과를 얻는다.

$$\underline{A \ni K} \quad (4.4)$$

- 규칙 R1, 가정 $A|\equiv \varnothing(na1 \oplus na2)$, 규칙 P2, P4, 그리고 (4.4)의 결과 및 가정 $A \ni (K_A \oplus na1)$, $A \ni na2$에 의해서 다음 결과를 얻는다.

$$A \ni (K_A \oplus na1, K, na2) \quad (4.5)$$
$$A \ni H(K_A \oplus na1, K, na2)$$

- 규칙 R1, 가정 $A|\equiv \varnothing(K_A \oplus na1)$ $A|\equiv \varnothing(na2)\}$, 규칙 R5, 그리고 (4.5)를 적용한 후 이어서 규칙 F1, 가정 $A|\equiv \#(K_A \oplus na1)$, $A|\equiv \#(na2)$에 의해서 다음 결과를 얻는다.

$$A|\equiv \#(K_A \oplus na1, K, na2) \quad (4.6)$$

- 규칙 F10과 (4.5)의 결과에 의해서 다음 결과를 얻는다.

$$A|\equiv \#(H(K_A \oplus na1, K, na2)) \quad (4.7)$$

- 규칙 I3, 가정 $A|\equiv A \overset{K_S}{\longleftrightarrow} S$, (4.5)와 (4.6)의 결과, 규칙 J2, 가정 $A|\equiv S \Rightarrow S|\equiv *$, (4.7)의 결과를 적용한 후, 규칙 J3과 가정 $A|\equiv S \Rightarrow S|\equiv *$을 적용하면 다음 결과를 얻는다. 이것은 S가 세션키 K를 신뢰하고 있다는 사실을 A가 확증하였음을 의미한다.

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

- 규칙 J1, 가정에 의해서 다음 결과를 얻는다. 이것은 A가 세션키 K를 신뢰하게 되었음을 의미한다.

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

(3) 메시지 3

• 규칙 T1, P1, P2, 가정 $S \ni K_A$, (4.3)의 결과, 규칙 P2, P4, 가정 $S \ni K$, (4.2)의 결과에 의해서 다음 결과를 얻는다.

$$S \ni (K_A \oplus na2,\ K,\ na1) \quad (4.8)$$
$$S \ni H(K_A \oplus na2,\ K,\ na1)$$

• 규칙 R1, 가정 $S| \equiv \varnothing(K)$, 규칙 R5, (4.8), F1, 가정 $S| \equiv \#(K)$에 의해서 다음 결과를 얻는다.

$$S| \equiv \#(K_A \oplus na2,\ K,\ na1) \quad (4.9)$$

• 규칙 F10, (4.8)의 결과에 의해서 다음 결과를 얻는다.

$$S| \equiv \#(H(K_A \oplus na2,\ K,\ na1) \quad (4.10)$$

• 규칙 I3, 가정 $S| \equiv A \overset{K_S}{\longleftrightarrow} S$, (4.8) , (4.9)의 결과, 규칙 J2, 가정 $S| \equiv A \Rightarrow A| \equiv *$, (4.10)의 결과, 규칙 J3, 가정 $S| \equiv A \Rightarrow A| \equiv *$에 의해서 다음 결과를 얻는다. 이것은 A가 세션키 K를 신뢰하고 있다는 사실을 S가 확증하였음을 의미한다.

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

(3) 검증 결과

GNY 로직을 통한 결과는 앞에서 밑줄로 표시하였다. 이 결과를 종합하여 보면 다음과 같다.

$$S \ni K,\ S| \equiv A \overset{K}{\longleftrightarrow} S,\ S| \equiv A| \equiv A \overset{K}{\longleftrightarrow} S$$
$$A \ni K,\ A| \equiv A \overset{K}{\longleftrightarrow} S,\ A| \equiv S| \equiv A \overset{K}{\longleftrightarrow} S$$

먼저 S가 세션키 K를 생성하므로 가정에 의해서 S가 K를 소유하고 신뢰하는 것은 분명하다. 프로토콜을 통하여 A 또한 K를 소유하고 신뢰하게 됨을 검증 결과로 알 수 있으며, 특히 S와 A 서로가 상대방이 키를 신뢰하고 있음을 확증하게 됨을 알 수 있다. 이 결과는 인증과 키 분배가 안전하게 이루어졌음을 보여준다.

5. 제안 프로토콜 평가

본 논문에서 제안한 프로토콜의 주 요소 중에서 na1⊕na2⊕K값은 새로운 세션키 K에 대한 외부 공격자의 분석을 차단하며, na1⊕na2 값과 해쉬값 $h(K_A \oplus na1,\ na2)$는 세션키 K를 알고 있는 내부 공격자를 차단한다. 패스워드를 추측하거나 새로운 세션키 K를 얻은 이후에도 추측 공격을 위한 계산량은 여전히 $2^{|na1|}$ 이상이 되며 S가 패스워드 실패를 발견할 수 있으므로 모든 추측 공격에 대해서 안전하다. 또한 GNY 로직으로 검증한 결과는 재전송 공격이나 Dennig-Sacco 공격 등에 대해서 안전함을 알 수 있다. 종합적으로 공격에 관한 면을 고려할 때 추측 공격을 고려하지 않았거나 부분적으로 취약성을 보이는 기존 프로토콜에 비해서 우수하다.[1,2,8,9,11] 이에 비하여 Needham-Shroeder 프로토콜[2]을 근간으로 하는 Kerberos 시스템에 대한 추측 공격 사례는 [5]에서 언급하고 있으며, 그 밖의 기존 패스워드 기반 프로토콜에 대한 추측 공격 사례는 [3]과 [4]에서 잘 설명하고 있다.

따라서 추측 공격을 고려하지 않았거나 고려했어도 부분적으로 취약성을 보이는 프로토콜보다는 추측 공격에 대해서 안전한 것으로 평가되는 프로토콜들과 오버헤드 측면에서 비교해볼 필요가 있다.[2,5,6,7,10] 비교한 결과는 표2와 같다.

먼저 (a)에서 직접 인증 프로토콜 비교를 보면 K1P가 Strengthened EKE나 GLNS nonce direct 프로토콜보다 프로토콜 단계, 난수 생성 횟수, 암호화 관련 연산 횟수 면에서 더욱 우수한 것을 알 수 있다. Strengthened EKE는 기존의 EKE에서 Denning-Sacco 공격에 대한 취약

성을 제거한 프로토콜이며,[7] GLNS nonce direct 프로토콜은 검증 가능문 공격을 고려하여 설계된 프로토콜로서[5] 모두 패스워드 기반 시스템을 대상으로 한다.

한편 (b)에서 상호 인증 프로토콜 비교를 보면 K1P가 GLNS nonce 프로토콜이나 Gong Optimal 프로토콜보다 역시 프로토콜 단계, 난수 생성 횟수, 암호화 관련 연산 횟수 면에서 더욱 우수한 것을 알 수 있다. GLNS nonce 프로토콜은 검증 가능문 공격을 고려하여 설계된 프로토콜이며[5] Gong Optimal 프로토콜[6]은 GLNS nonce 프로토콜을 최적화한 프로토콜로서 역시 모두 패스워드 기반 시스템을 대상으로 한다.

종합적으로 효율성 측면에서 볼 때, K1P는 추측 공격에 대해서 안전한 프로토콜로서 기존 프로토콜에 비해 효율적이다.[5,6,7,10]

6. 결론

기존에 제안된 대부분의 프로토콜들은 패스워드 추측 공격에 노출되는 약점을 보여주었다.[3,10] 또한 추측 공격을 막기 위해서 제안된 여러 프로토콜들은 많은 계산량과 통신량을 요구하는 오버헤드를 수반하였으며, 이를 제거할 경우에 새로운 약점을 나타내었다.[4]

본 논문에서는 이와 같은 문제점을 해결하기 위해서 일시패드와 해쉬함수의 특성을 이

표 2 추측 공격에 대해 안전한 인증 프로토콜 비교

프로토콜	프로토콜단계	난수 생성 횟수		암호화 관련 연산 횟수			
					Pub.	Conv.	Hash.
K1P	3	A	2	A-S	1	0	2
		S	0				
Strengthened EKE[7]	5	A	2	A-S	1	3	2
		S	2				
GLNS nonce direct[5]	5	A	1	A-S	1	5	0
		S	4				

(a) 2자 인증 프로토콜

프로토콜	프로토콜단계	난수 생성 횟수		암호화 관련 연산 횟수			
					Pub.	Conv.	Hash.
K1P	5	A	2	A-S	1	0	1
		B	2	B-S	1	0	1
		S	0	A-B	0	2	0
GLNS nonce[5]	7	A	4	A-S	1	2	0
		B	4	B-S	1	2	0
		S	1	A-B	0	2	0
Gong Optimal[6]	5	A	5	A-S	1	2	0
		B	5	B-S	1	2	1
		S	0	A-B	0	2	0

(b) 3자 상호 인증 프로토콜

용하였으며, 따라서 안전성과 효율성을 도모하도록 연구하였다. 제안한 프로토콜인 K1P는 일반적으로 널리 이용되는 패스워드 기반 시스템에서 효율적이고 안전한 인증 기능과 함께 세션키의 안전한 분배 기능을 제공토록 활용될 수 있다. 예를 들면 다양한 클라이언트-서버 모델에서 패스워드를 이용한 안전한 인증 및 키 분배가 가능하며, UNIX와 같이 사용자 인증을 요하는 운영체제 환경이나 여러 통신망 서비스에서 활용할 수 있다. 즉, 본 논문에서 제안하는 프로토콜 K1P는 일시 패드와 해쉬 함수의 안전성을 근간으로 하며, 패스워드 기반 시스템에 활용되기에 적합한데, 그 이유는 오프라인 및 온라인 추측 공격을 포함한 다양한 공격에 대해서 안전하며, 추측 공격에 대해 안전한 인증 프로토콜로서 기존의 패스워드 기반 프로토콜에 비해서 통신량 및 계산량의 오버헤드를 줄였다는 면에서 설명될 수 있다.

참 고 문 헌

[1] W.Diffie, M.Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, vol. 22, no. 6, Nov. 1976, pp. 644-654

[2] R.Needham, M.Schroeder, Using Encryption For Authentication in Large Networks of Computers, Communications of the ACM, vol. 21, no. 12, Dec. 1978, pp.993-999

[3] L.Gong, Verifiable-text Attacks in Cryptographic Protocols, Proceedings of IEEE INFOCOM'90, June 1990, pp. 686-693

[4] Y.Ding, P.Horster, Undetectable On-line Password Guessing Attacks, ACM Operating Systems Review, vol. 29, no. 4, Oct. 1995, pp. 77-86

[5] L.Gong, M.Lomas, R.Needham, J.Saltzer, Protecting Poorly Chosen Secrets from Guessing Attacks, IEEE Journal on Selected Areas in Communications, vol. 11, no. 5, June 1993, pp. 648-656

[6] L.Gong, Optimal Authentication Protocols Resistant to Password Guessing Attacks, Proceedings of the 8th IEEE Computer Security Foundations Workshop, June 1995, pp. 24-29

[7] S.Bellovin, M.Merritt, Encrypted Key Exchange : Password-Based Protocols Secure Aganist Dictionary Attacks, Proceedings of IEEE Symposium on Security and Privacy, 1992, pp. 72-84

[8] W.Fumy, M.Munzert, A Modular Approach to Key Distribution, Proceedings of Crypto'90, 1991, pp. 274-283

[9] D.Otway, O.Rees, Efficient and Timely Mutual Authentication, ACM Operating Systems Review, vol. 21, no. 1, Jan. 1987, pp. 8-10

[10] MLomas, L.Gong, J.Saltzer, R.Needham, Reducing Risks from Poorly Chosen Keys, Proceedings of the 12th ACM Symposium on Operating System Principles, ACM Operating Systems Review, vol. 23, no. 5, Dec. 1989, pp. 14-18

[11] E.Okamoto, K.Tanaka, Key Distribution Based on Identification Information, IEEE Journal on Selected Areas in Communication, vol. 7, no. 4, May 1989, pp. 481-485

[12] D.Denning, G.Sacco, Timestamps in Key Distribution Protocols, Communications of ACM, vol. 24, no. 8, Aug. 1981, pp. 533-536

[13] M.Burrows, M.Abadi, R.Needham, A Logic of Authentication, ACM Transactions on Computer Systems, vol. 8, no. 1, Feb. 1990, pp. 18-36

[14] L.Gong, R.Needham, R.Yahalom, Reasoning about Belief in Cryptographic Protocols, Proceedings of the IEEE Symposium on Research in Security and Privacy, 1990, pp.234-248

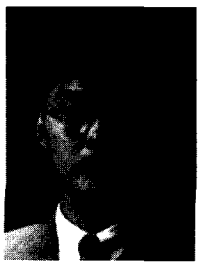
[15] B.Schneier, Applied Cryptography 2nd edition, John Wiley & Sons, 1996

□ 著者紹介

권 태 경

1992년 2월 연세대학교 컴퓨터과학과 이학학사
1995년 2월 연세대학교 컴퓨터과학과 이학석사
1995년 3월 ~ 현재 연세대학교 컴퓨터과학과 박사과정 재학중

※주관심 분야 : 컴퓨터 통신망 보안, 암호학, PCS, 지능망 시스템, EDI 시스템



강 명 호(文 松 天)

1994년 2월 연세대학교 컴퓨터과학과 이학학사
1996년 2월 연세대학교 컴퓨터과학과 이학석사
1996년 3월 ~ 현재 연세대학교 컴퓨터과학과 박사과정 재학중

※주관심 분야 : 컴퓨터 보안, 지능망 시스템, 형식 언어, 암호 이론

송 주 석

1976년 2월 서울대학교 전기공학과 학사
1979년 2월 한국과학원 전기 및 전자공학과 졸업 석사
1988년 8월 Univ. of California at Berkeley 전산과학과 박사
1979년 2월 ~ 1982년 2월 한국전자통신연구소 전임연구원
1988년 9월 ~ 1989년 2월 Naval Postgraduate School Information System Department 조교수
1989년 3월 ~ 현재 연세대학교 컴퓨터과학과 교수

※ 주관심 분야 : 프로토콜 공학, ATM 통신망, 통신망 보안 등