Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.5, Oct. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.5.769

재전송 공격에 안전하고 개선된 Single Sign-On 인증 시스템에 관한 연구*

김 현 진,† 이 임 영‡
순천향대학교 컴퓨터소프트웨어공학과

A Study on Secure and Improved Single Sign-On Authentication System against Replay Attack*

Hyun-Jin Kim,† Im-Yeong Lee‡
Department of Computer Software Engineering, Soonchunhyang University

요 약

보통 사용자들은 여러 서비스 사이트를 이용함에 있어 여러 개의 아이디와 패스워드를 기억하여 사용한다. 이러한 불편함을 해결하고 관리측면에서 효과적인 방법으로 제안된 인증 시스템이 SSO이다. 특히 SSO 인증 모델 중 브로커 기반의 경우 중앙 집중식 시스템 관리를 사용하는 Kerberos 인증이 대표적이다. 그러나 기존의 Kerberos 인증은 패스워드 공격 및 재전송 공격에 많은 보안 취약성을 가지고 있어 이에 대한 연구가 진행되고 있다. SSO 인증 시스템에 있어 주된 보안 취약점은 재전송 공격이다. 사용자의 인증정보가 공격자에 의해 탈취되었을 경우, 단순히 재전송 공격을 통한 세션 획득이 가능하다는 문제점이 존재한다. 이에 본 논문에서는 보다 개선된 브로커 기반 SSO 인증 모델을 제안하고, 인증정보 재전송 공격에 안전한 SSO 경량화 메커니즘을 제안한다.

ABSTRACT

In general, internet users need to remember several IDs and passwords when they use diverse web sites. From an effective management perspective, SSO system was suggested to reduce user inconvenience. Kerberos authentication, which uses centralized system management, is a typical example of a broker-based SSO authentication model. However, further research is required, because the existing Kerberos authentication system has security vulnerability problems of password and replay attacks. In SSO authentication systems, a major security vulnerability is the replay attack. When user credentials are seized by attackers, an authorized session can be obtained through a replay attack. In this paper, an improved SSO authentication model based on the broker-based model and a secure lightweight SSO mechanism against credential replay attack is proposed.

Keywords: Single Sign-On, Authentication, Credential Privacy

I. 서 론

최근 IT 기술력의 고도화로 초고속 인터넷 망이 발달하게 되었고 과거 인터넷 환경과는 다른 모습을 보이고 있다. 일상생활에서 다양한 인터넷 서비스를 접할 수 있게 되었고, 이러한 인터넷 서비스는 우리 생활의 필수요소라 할 수 있다. 많은 사용자가 이용할 수 있는 서비스가 다양해짐에 따라 효용가치도 증가하였다. 또한 휴대성이 편리한 다양한 스마트 기기들의 발전과 보급으로 인해 더 많은 정보와 서비스들이 웹

접수일(2014년 5월 26일), 수정일(1차: 2014년 8월 3일, 2차: 2014년 9월 2일), 게재확정일(2014년 9월 19일)
* 본 연구는 순천향대학교 학술연구비 지원으로 수행하였음.
† 주저자, hjkim128@sch.ac.kr
‡ 교신저자, imylee@sch.ac.kr(Corresponding author)

기반 형태로 변화되고 있다.

하지만 웹 기반 형태의 서비스들은 각 사용자에게 인증정보를 요구하게 되고, 정당한 사용자로 인증되었을 경우에만 요청한 서비스들을 제공 받을 수 있다. 이에 따라 사용자측면에서 다양한 웹 서비스 별로 개별적인 아이디와 패스워드를 설정하고 기억해야 하는 불편함이 따르게 되었다. 또한 서비스 업체에서 여러 연계된 서비스들을 제공하게 됨에 따라 중복 사용자들의 인증정보를 따로 관리하게 되는 문제점이 발생되었다. 이러한 문제점을 극복하기 위해 한 번의 인증으로 다양한 서비스들을 이용할 수 있는 SSO(Single Sing-On)이라는 개념이 등장하였다[1].

SSO 인증의 경우 한 번의 인증을 통해 다양한 서비스들을 이용할 수 있기 때문에 인증 방법에 있어 안전성이 고려되어야 한다. 특히 SSO 인증에 사용되는 사용자 인증정보가 공격자에게 탈취되었을 경우 단순히 재전송 공격을 통해 연계된 모든 서비스에 정당한 사용자로써 인증된 세션을 취득할 수 있는 취약점을 가지고 있다. 이에 인증 시스템 적용 환경에 따라 다양한 SSO 인증 모델이 개발되어, 서비스 별 제공자의 환경에 맞는 인증모델을 선택하여 사용하게 된다.

SSO 인증 시스템에서 인증모델은 중요한 요소로써 작용되며, 전체적인 사용자 인증 과정의 안전성과 효율성에 큰 영향을 미친다[2]. 그 중 브로커 기반 모델의 경우 중앙집중식 시스템 관리를 사용하여 이를 통한 인증 연산처리의 효율성을 증가시킨다. 이러한 장점을 가지는 브로커 기반 모델은 Kerberos를 활용하여 인증을 실시한다. 전통적인 SSO 프로토콜의 경우 패스워드 공격 및 재전송 공격에 비교적 심각한 위험성을 가지고 있다. 이에 Jian[3]은 전통적인 SSO 프로토콜에 두 개의 새로운 데이터 흐름을 추가하여 보다 개선된 SSO 프로토콜을 제안하였다. 하지만 불필요한 데이터 생성 및 분배로 인해 비효율적인 단점을 가지고 있다.

또한 SSO 인증 시스템은 사용자 인증 방식에 따라 분류될 수 있는데 크게 인증 대행 방식과 인증정보 전달 방식으로 나뉘게 된다. 현재 웹 기반 형태의 서비스가 다양해짐에 따라 토큰 혹은 쿠키와 같은 인증정보를 생성하여 서비스 제공자에게 전달함으로써 인증을 받는 형태가 주를 이루고 있다. 이러한 인증정보 전달 방식에 있어서도 재전송 공격을 통한 세션 취득의 문제점이 야기되고 있다. 이에 Lee[4]는 사용자의 민감한 인증정보를 대신하여 랜덤 값 생성을 통한 재전송 공격에 강인한 방법을 제안하였다. 하지만 불필요한 데이터 생성 및 연산, 인증정보 부인 등의 문제점 존재로 인해 인증 프로토콜의 효율성이 감소한다.

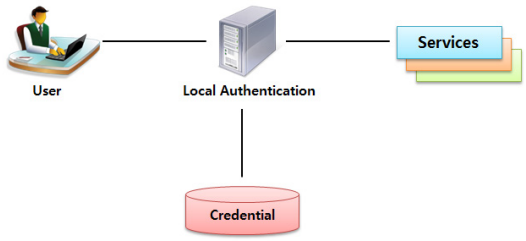

Fig. 1. General Structure of SSO Authentication

이에 본 논문에서는 Jian의 개선된 SSO 프로토콜 시스템을 기반으로 한 보다 개선된 브로커 기반 SSO 인증 모델을 제안하며, Lee의 제안 방식을 기반으로 간접적인 인증정보를 활용하여 재전송 공격에 안전하고 기존 통신 횟수의 변화 없이 경량화 된 연산의 인증 메커니즘을 제안한다.

본 논문의 구성은 2장에서는 SSO 인증 시스템 분류 등 관련 연구에 대해 분석한다. 3장에서는 안전한 SSO 인증 시스템 설계를 위한 보안요구사항에 대해 분석하고, 4장에서는 제안 방식에 대하여 기술한다. 5장에서는 기존 방식과 제안 방식을 비교 분석하며, 마지막으로 6장에서 결론으로 마친다.

II. 관련연구

2.1 일반적인 SSO 인증 시스템 구조

SSO는 단 한 번의 인증으로 다양한 서비스들을 이용할 수 있는 인증 기술로써 일반적인 구조는 Fig.1.과 같다[4].

로컬 인증은 SSO 인증 시스템 자체의 메커니즘에 의해 사용자를 인증하며, SSO 인증 시스템을 통해 사용자가 제공받는 서비스 혹은 자원들에 대한 자격증명 정보(로그인 정보, 권한정보 등)를 부여하거나 해제한다. 즉, 로컬 인증은 사용자의 신원을 자격증명 정보로 매핑 하는 기능을 수행한다. 자격증명은 사용자 인증에 사용되는 정보들로써 예로 아이디와 패스워드, 권한 정보 등을 들 수 있다. 이러한 정보들은 보안의 중요성을 가지고 있으므로 안전한 데이터 저장소에 저장 관리된다. 서비스는 사용자가 요청하는 응용 프로그램이다. SSO 인증 시스템을 통해 정상적인 사용자로 인가된 후 접근 권한을 부여받는다[5].

Table 1. Classification of SSO Systems According to User Authentication Method

	Strength	Weakness	Example
Broker-based	- Efficiency through a centralized system management is increased.	- It requires modification of existing applications.	Kerberos SESAME
Agent-based	- Modification of the existing Application is reduced.	- Efficient user account management is required.	SSH
Agent-Broker -based	- Modification of the existing Application is reduced. - Efficiency through a centralized system management is increased.	- The managed component is increased.	Axent's ERM
Gateway-based	- Efficiency through a centralized system management is increased.	- The synchronization is required between many gateways.	Cylink's Private Wire

2.2 시스템 구조에 따른 SSO 분류

SSO 인증 시스템을 시스템 구조에 따라 분류하면 브로커기반 모델, 에이전트기반 모델, 에이전트-브로커기반 모델, 게이트웨이기반 모델로 분류할 수 있다[3]. 브로커기반 모델은 중앙 집중적인 인증 형태를 가지며 이를 위해 사용자 계정을 관리하는 서버가 별개로 존재하게 된다. 한 번의 인증을 완료하게 되면 이후 접속 요청 시 사용자 인증에 사용할 토큰을 생성하여 부여하게 된다. 에이전트기반 모델은 별도의 중앙 인증 서버가 존재하지 않는 대신에 SSO 시스템과 연계된 서로 다른 응용시스템에 사용자를 자동으로 인증시켜주는 에이전트가 존재한다. 에이전트-브로커기반 모델은 브로커 기반의 모델을 적용할 때 시스템을 SSO 모델로 변경하는 비용이 크거나 불가능한 경우를 위한 모델로써 중앙 집중식 관리와 유연성을 결합한 모델이다. 게이트웨이기반 모델은 게이트웨이나 프록시 형태의 강력한 암호화 인증 서버를 두어서 도메인으로 들어오는 네트워크 흐름을 여과하면서 SSO기능을 지원하는 모델이다. 시스템 구조에 따른 SSO 분류의 장점과 단점은 다음 Table 1.과 같다[6].

2.3 사용자 인증 방식에 따른 SSO 분류

SSO 인증 시스템은 사용자 인증 방식에 따라 인증 대행 방식과 인증정보 전달 방식으로 나눌 수 있다[4]. Fig.2.는 인증 대행 방식의 구조로써 사용자 인증에 사용되는 인증정보를 하나의 에이전트가 중앙 관리하고 사용자를 대신하여 해당 서비스 서버에 로그인 해주는 방식이다. 이는 SSO와 관련된 정보는 사용자 클라이언트를 거치지 않고 에이전트와 서비스 서버들 사이에서만 전달하여 처리한다. 그렇기 때문에 사용자와 에이전트 사이의 채널이 안전하게 보호된다고 가정하면 비교적 안전한 방식이다. 보통 대상 서비스 서버의 인증 방식을 변경하기 어려울 때 많이 사용된다. Fig.3.은 인증정보 전달 방식의 구조로써 사용자가 별도의 인증 서버를 통해 정당한 사용자로 판별되면 인증 서버로부터 서비스 서버로 전달할 SSO 토큰을 발급받아 서비스를 요청한다. 서비스 서버로 접근할 때에는 사용자가 자동으로 발급받은 SSO 토큰을 전달하여 서비스 서버가 사용자를 확인하는 방식이다. 실제로 웹 기반 서비스를 사용하는 환경에서는 보통 쿠키(Cookie)를 사용하여 서비스 서버에 전달하는

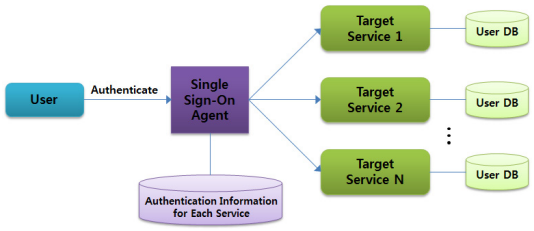

Fig. 2. Certification Agency Structure

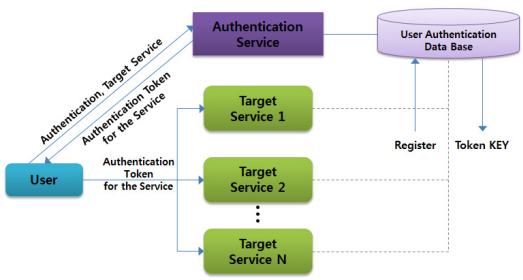

Fig. 3. Authentication Information-Delivery Structure

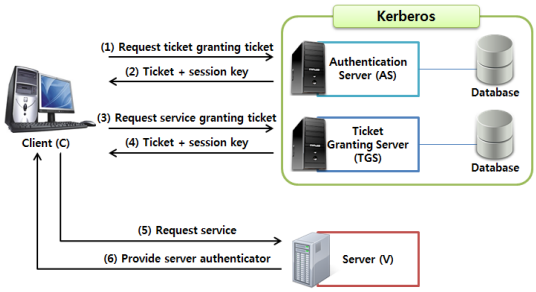

Fig. 4. Process of the Kerberos Protocol

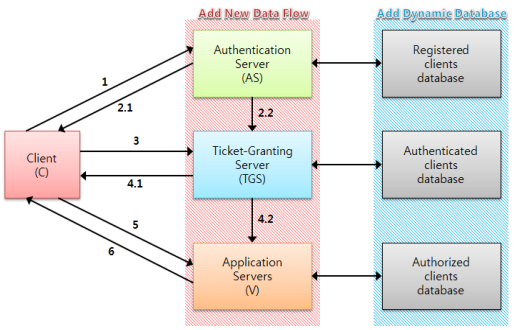

Fig. 5. Structure of Jian's Scheme

것만으로 사용자를 인증하게 된다. 이러한 방식은 사용자 인증정보가 클라이언트를 통해 전송되기 때문에 스니핑 공격에 취약하며, 재전송 공격 등이 가능한 취약점이 존재한다[5].

2.4 Kerberos

Kerberos는 MIT(Massachusetts Institute of Technology)에서 개발한 Athena 프로젝트의 한 부분으로써 비밀키 인증 프로토콜이다. 개방된 네트워크 환경에서 클라이언트와 서버간의 인증을 제공하기 위해 중앙에 신뢰된 제3자인 인증 서버를 두고 있다. 이처럼 중앙 집중식 인증 서버를 사용하여 클라이언트와 서버 간에 강력한 인증 기능을 제공한다.

Needham-Schroeder의 인증 모델을 근거로 하여 설계되었으며, 현재는 Kerberos V4와 V5가 사용되고 있다. Kerberos V5의 경우 문서 표준화 RFC 1510으로 발표되었다. Kerberos 프로토콜의 전체적인 흐름은 Fig.4.과 같으며 하위 프로토콜로써 세 가지로 구분된다[7].

- Authentication Service Exchange :
  KDC(Key Distribution Center)가 클라이언트에게 세션키와 티켓허가티켓을 제공하는 프로토콜, (1)~(2)과정.
- Ticket Granting Service Exchange :
  KDC가 클라이언트에게 서비스 세션키와 서비스 티켓을 제공하는 프로토콜, (3)~(4)과정.
- Client/Server Exchange :
  클라이언트가 서비스에 접근하기 위해 세션티켓을 제시하는 프로토콜, (5)~(6)과정.

2.5 Jian[3] 제안방식

Jian은 Kerberos를 기반으로 하는 전통적인 SSO 프로토콜을 개선한 방식을 제안하였다. 기존 Kerberos 프로토콜의 형태를 따르되, 인증서버와 티켓허가서버, 티켓허가서버와 어플리케이션 서버 간에 새로운 두 가지 데이터 흐름을 추가하였다. 이는 클라이언트에 티켓이 전송되는 과정을 줄임으로써 공격자가 수집한 데이터양을 줄임과 동시에 자동으로 이 단계로의 공격을 감소시키는 효과를 가진다. 또한 처리해야 될 정보감소로 인해 처리 속도를 향상시킨다.

각 서버에 구축되는 데이터베이스의 경우 전송 받은 데이터를 통해 매 과정마다 동적으로 등록하게 되며 등록된 사용자 인증 확인, 인증된 클라이언트 확인, 권한을 부여받은 클라이언트 확인에 활용된다. Jian 프로토콜 구조는 Fig.5.와 같으며 각 통신 순서는 번호로 나타내었다.

하지만 해당 방식의 경우 불필요한 데이터 생성 및 분배 과정을 통해 인증 프로세스가 비효율적이며, 클라이언트 인증자를 복호화 하기 위한 키 값 검색과정이 잘못 설계되어 원활한 SSO 인증이 이루어지지 않는 문제점을 가지고 있다.

2.6 Lee[4] 제안방식

실제 웹서비스 환경에서 많은 서비스들이 SSO 인증 시 서비스 토큰 등의 인증 정보를 인터넷 쿠키에 저장하여 사용자가 서비스서버로 접근 할 때 자동 전달되는 방식을 사용하고 있다. 그러나 쿠키는 도난과 재전송 공격 등의 보안상 문제점을 가지고 있다. 특히 SSO 인증 시스템에서는 쿠키를 통해 사용자 인증을 처리하기 때문에 쿠키 도난 또는 재전송 공격에 이용되었을 경우 SSO 인증 시스템에 연계된 모

든 서비스에 접근 가능한 권한을 가지게 된다. 이러한 문제점을 고려하여 Lee는 쿠키를 사용하지 않는 대신에 인증서버에서 서비스 토큰과 추가적인 인증 정보를 발행하여 서비스 서버에 접근하는 재전송 공격에 강인한 SSO 인증 방식을 제안하였다[4]. 하지만 제안된 인증 프로토콜에 있어 불필요한 데이터 생성 및 연산, 인증정보 부인 등의 문제점이 존재하며, 스마트기기에서의 사용을 고려하였을 경우 연산량 측면에서 부담감이 존재하는 문제점을 가지고 있다.

III. 보안요구사항

SSO 인증 시스템의 보안상 핵심 요인은 바로 인증이 완료된 세션에 대한 보호이다. 정상적으로 인증된 세션 탈취가 행해지게 되면 악의적인 공격자에 의한 보안 취약성이 발생할 수 있다. 따라서 안전한 SSO 설계를 위한 보안 요구사항은 다음과 같다[8].

· 기밀성 : 사용자 및 서버 인증을 위해서 인증 과정에 사용자 인증정보, 인증토큰, 쿠키 등 다양한 인증정보가 전송 및 저장된다. 이와 같은 사용자 인증정보는 공격자가 내용을 확인할 수 없도록 해야 한다. 이를 위해 통신 프로토콜을 통해 전달되는 메시지는 해시연산 혹은 암호 알고리즘을 통한 암호화로 데이터에 대한 안전성을 제공해야 한다.

· 무결성 : SSO을 통한 로그인 과정을 위해 사용자에게 전송되는 인터페이스는 위조 또는 변조될 수 있다. 인터페이스를 통해 입력된 사용자의 인증 정보가 노출되어 악용될 수 있기 때문에 인터페이스의 위조 또는 변조가 불가능 하도록 하거나, 서버가 위조 또는 변조된 사실을 알 수 있도록 설계해야 한다.

· 재전송 공격 보안 : 사용자와 서버간의 인증 과정을 위한 통신과정의 트래픽을 통해 정상적인 사용자 패킷을 쉽게 도청 가능하다. 이를 통해 공격자는 재전송 공격이 가능하므로 재전송 공격에 안전한 SSO를 설계해야 한다.

· 인증정보 프라이버시 : 만약 통신 상 전달되는 메시지에 대해 Dictionary Attack 등으로 공격자에게 사용자 인증정보가 노출되었다 하더라도 정상 사용자의 민감한 정보가 담겨있지 않도록 간접적인 인증 정보만을 사용하도록 설계해야 한다.

IV. 제안방식

4.1 개선된 브로커 기반 SSO 모델(제안방식 1)

본 제안방식에서는 기존 Jian[3] 방식의 Kerberos 프로토콜 구조를 기반으로 하며, 문제점을 보완하여 보다 개선된 브로커 기반의 SSO 인증 모델을 제안하였다. 인증 프로토콜은 사전등록 단계, 인증 서비스 교환 단계, 티켓 허가 서비스 교환 단계, 클라이언트-서비스서버 교환단계로 구성되며, 전체 프로토콜 구조는 Fig.6.과 같다.

4.1.1 시스템 계수

본 제안방식에서는 다음과 같은 시스템 계수를 사용하여 프로토콜을 설계한다.

· $*$: 개체 (C : 클라이언트, AS : 인증서버, TGS : 티켓허가서버, V : 서비스 서버)
· ID_* : 해당 개체의 ID
· AD_* : 해당 개체의 IP 주소
· $K_{*,*'}$: $*$와 $*$‘ 사이에 공유된 비밀키
· AU_C : 클라이언트를 나타내는 인증자
· $Times$: 티켓의 유효시간
· $Ticket_{TGS}$: TGS에게 전송될 티켓허가티켓
· $Ticket_V$: V에게 전송될 서비스티켓
· TS : 현재 시간을 나타내는 타임스탬프
· $Subkey$: C가 정한 V와의 세션키

4.1.2 사전등록 단계

사용자는 브로커 기반 SSO 인증 모델을 사용하기 위하여 자신의 아이디와 패스워드를 인증서버 AS에 등록하는 과정을 거친다. 이는 일차적으로 사용자를 인증하는데 사용된다. 아이디와 패스워드를 통한 일차 인증 여부에 따라 이후 단계 수행을 결정한다.

사용자 등록이 완료되면 인증서버 AS는 비밀키 K_{AS}를 생성하여 클라이언트에게 안전하게 사전 분배한다. 또한 서버 간에 티켓 전송 시 암호화를 위해 사용되는 비밀키 K_{TGS}, K_V를 생성하여 해당 서버인 TGS 및 V에 안전하게 사전 분배한다.

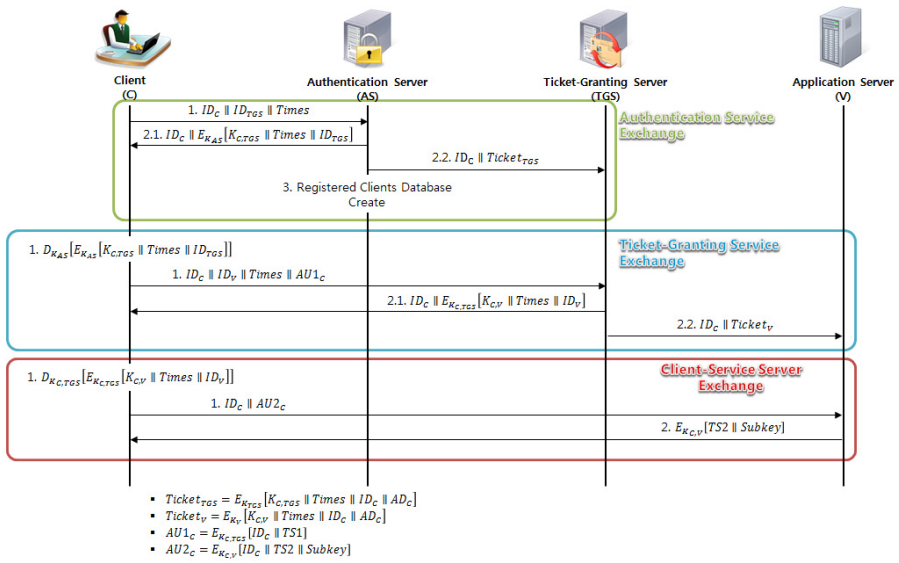

Fig. 6. Proposed Scheme 1 : Structure of Protocol

4.1.3 인증 서비스 교환 단계

사용자가 자신의 아이디와 패스워드를 통해 일차 인증을 요청했을 때 인증서버 AS에 정상적인 사용자로 등록이 되어 있다면 다음의 과정을 실시한다.

Step 1. 클라이언트 C는 티켓허가티켓 발급 및 티켓허가서버 TGS와의 비밀키 분배를 위하여 인증서버 AS에게 티켓유효시간과 함께 다음과 같은 요청 메시지를 전송한다.

$$C \rightarrow AS: ID_C \| ID_{TGS} \| Times \quad (1)$$

Step 2.1. 인증서버 AS는 요청 메시지에 대한 응답으로 클라이언트 C와 티켓허가서버 TGS간에 사용될 비밀키를 생성하여 티켓유효시간과 함께 K_{AS}로 암호화 하여 전송한다.

$$AS \rightarrow C: ID_C \| E_{K_{AS}}[K_{C,TGS} \| Times \| ID_{TGS}] \quad (2)$$

Step 2.2. 인증서버 AS는 비밀키 분배와 함께 티켓허가티켓 $Ticket_{TGS}$를 생성하여 티켓허가서버 TGS에게 전송한다.

$$AS: Ticket_{TGS} = E_{K_{TGS}}[K_{C,TGS} \| Times \| ID_C \| AD_C] \quad (3)$$

$$AS \rightarrow TGS: ID_C \| Ticket_{TGS} \quad (4)$$

Step 3. 티켓허가티켓 $Ticket_{TGS}$를 전송받은 티켓허가서버 TGS는 해당 클라이언트 ID_C에 대해 인증된 클라이언트 데이터베이스를 구축한다.

4.1.4 티켓 허가 서비스 교환 단계

티켓 허가 이후 클라이언트는 자신의 신원을 밝힌 후 이용하고자 하는 서비스에 대한 서비스티켓을 요청한다.

Step 1. 클라이언트 C는 키 분배 메시지를 복호화하여 비밀키를 획득하고 자신의 신원을 밝히기 위한 인증자 AU_C를 생성한다. 또한 인증자 $AU1_C$ 및 서비스서버 V와의 비밀키 분배를 위한 요청메시지를 티켓 유효시간과 함께 티켓허가서버 TGS에게 전송한다.

$$C: D_{K_{AS}}[E_{K_{AS}}[K_{C,TGS} \| Times \| ID_{TGS}]] \quad (5)$$

$$C: AU1_C = E_{K_{C,TGS}}[ID_C \| TS1] \quad (6)$$

$$C \rightarrow TGS: ID_C \| ID_V \| Times \| AU1_C \quad (7)$$

Step 2.1. 티켓허가서버 TGS는 요청 메시지에 대한 응답으로 클라이언트 C와 서비스서버 V간에 사

용될 비밀키를 생성하여 티켓유효시간과 함께 $K_{C,TGS}$로 암호화 하여 전송한다.

$$TGS \rightarrow C: ID_C \| E_{K_{C,TGS}}[K_{C,V} \| Times \| ID_V] \quad (8)$$

Step 2.2. 티켓허가서버 TGS는 비밀키 분배와 함께 서비스티켓 $Ticket_V$를 생성하여 서비스서버 V에게 전송한다.

$$TGS: Ticket_V = E_{K_V}[K_{C,V} \| Times \| ID_C \| AD_C] \quad (9)$$

$$TGS \rightarrow V: ID_C \| Ticket_V \quad (10)$$

4.1.5 클라이언트-서비스서버 교환 단계

서비스티켓 발급 이후 클라이언트는 서비스를 이용하기 위해 서비스서버에 자신의 신원을 밝히는 과정을 실시한다.

Step 1. 클라이언트 C는 키 분배 메시지를 복호화하여 비밀키를 획득하고 자신의 신원을 밝히기 위한 인증자 $AU2_C$를 생성하여 서비스서버 V에게 전송함으로써 서비스를 요청한다. 이때 서비스서버 V와 사용될 $Subkey$를 생성하여 함께 전송한다.

$$C: D_{K_{C,TGS}}[E_{K_{C,TGS}}[K_{C,V} \| Times \| ID_V]] \quad (11)$$

$$C: AU2_C = E_{K_{C,V}}[ID_C \| TS2 \| Subkey] \quad (12)$$

$$C \rightarrow V: ID_C \| AU2_C \quad (13)$$

Step 2. 서비스서버 V는 클라이언트 C에게 서비스 요청에 대한 응답메시지를 전송한다.

$$V \rightarrow C: E_{K_{C,V}}[TS2 \| Subkey] \quad (14)$$

4.1.6 문제점 보완사항

· 잘못된 프로토콜 설계 : 기존 Jian 방식의 경우 사용자에게 전송된 티켓 정보를 복호화하기 위한 키 값 검색에 있어서 각 서버들은 그 대상이 되는 클라이언트의 ID_C 값을 알 수 없게 설계되어 있다. 그로 인한 정상정인 인증 프로세스가 불가능하므로 이에 대한 개선 방안을 제안하였다.

· 불필요한 데이터 생성 및 분배 : 클라이언트에서 전송되는 요청 메시지에 포함되는 $Nonce$값이 불필요하게 생성되어 각 서버에게 전송되고 수신하는 과정을 거치는 문제점이 있다. 이에 통신 데이터에 대한 효율성을 증가시키도록 제안하였다.

4.2 인증정보 재전송 공격에 안전한 SSO 경량화 메커니즘(제안방식 2)

SSO 인증 시스템의 경우 정상적인 사용자의 인증정보 탈취 후 재전송 공격을 통한 정당한 세션을 취득하였을 경우 보안 위협이 발생하게 된다. 이에 사용자의 민감한 정보를 사용하는 것이 아닌 간접적인 정보를 생성하여 사용자를 인증하는 방법으로 민감 정보 탈취 위협을 해결할 수 있다. 뿐만 아니라 최근에는 다양한 스마트 기기를 통한 웹 서비스를 이용하기 때문에 SSO 인증 시스템을 사용할 경우 연산량 측면의 문제를 고려하지 않을 수 없다.

따라서 본 제안방식에서는 인증정보 재사용 공격에 안전한 SSO 경량화 메커니즘을 제안하였다. 전체적인 구조는 Lee[4]의 방식을 기반으로 구성되었으며, 단계별 프로토콜은 Fig.7.와 Fig.8.과 같다.

4.2.1 시스템 계수

본 제안방식에서는 다음과 같은 시스템계수를 사용하여 프로토콜을 설계한다.

- $*$: 개체 (U : 사용자, AS : 인증서버, S : 서비스 서버)
- N_U : 임의 생성된 Random Nonce 값
- OTP : 임의의 일회용 값
- $SS_{*,*}$: 인증서버와 개체 간 안전하게 공유된 SSL(Secure Sockets Layer) 프로토콜 세션키
- ST_U : 사용자 U의 정보 식별에 사용되는 서비스토큰
- TK_U : 사전에 사용자 U와 서비스 서버 간에 안전하게 협상된 서비스토큰 암호화 비밀키
- K_U : 사용자 U의 간접적인 인증 값
- SK_U : 상호인증 완료 후 사용자 U와 서비스 서버 간 통신에 사용될 비밀키
- $H()$: 일방향 해시함수

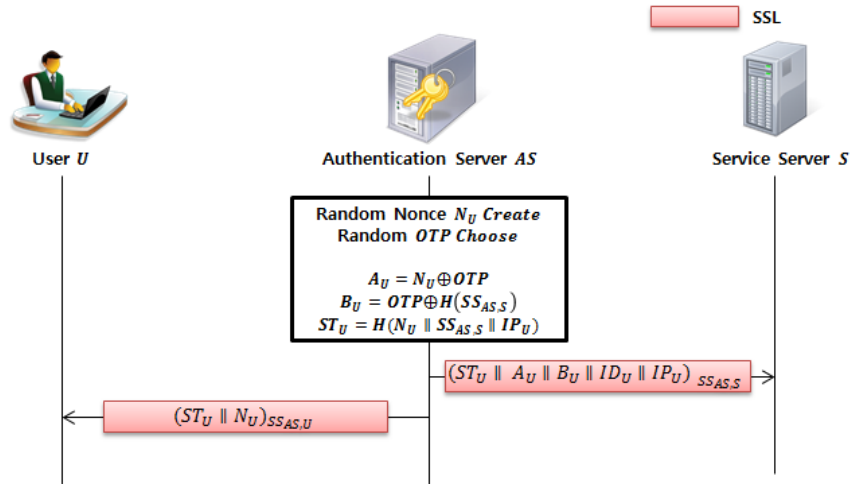

Fig. 7. Proposed Scheme 2 : Issuing Phase of Credential

4.2.2 초기설정 단계

SSO 인증을 위해서는 인증서버와 사용자, 서비스서버 사이의 인증 과정을 실시한다. 초기설정 단계는 SSL 프로토콜을 사용하며 해당 과정을 통해 개체 간에 사용될 암호화 알고리즘, SSL 세션키의 초기협상을 실시한다.

인증서버와 사용자 사이의 인증에서는 등록 시 고유한 사용자의 아이디, 패스워드 인증 방식을 사용하며, SSL을 통해 안전하게 전송된 아이디, 패스워드를 인증서버의 데이터베이스의 정보와 비교함으로써 사용자 인증을 완료한다. 인증서버와 서비스서버 사이의 인증에서는 서로 간에 절대적으로 신뢰할 수 있는 인증기관으로부터 발급된 인증서를 SSL을 통해 전송함으로써 인증을 완료한다.

4.2.3 인증정보 발급 단계

사용자가 자신의 아이디와 패스워드를 통해 인증서버에 정상적인 사용자로 인증되면 인증정보를 생성하고 발급한다. 인증서버와 각 개체간의 통신은 SSL 프로토콜 사용을 전제로 한다.

Step 1. 인증서버 AS는 인증정보 생성에 사용될 Random Nonce N_U 및 OTP를 생성하고, 서비스서버의 SSL 세션키 $SS_{AS,S}$, IP를 활용하여 인증정보를 생성한다.

$$AS: A_U = N_U \oplus OTP \quad (15)$$

$$AS: B_U = OTP \oplus H(SS_{AS,S}) \quad (16)$$

$$AS: ST_U = H(N_U \| SS_{AS,S} \| IP_U) \quad (17)$$

Step 2. 인증서버 AS는 사용자 U와 서비스서버 S에게 인증정보를 SSL을 통해 분배한다. 서비스서버는 발급받은 인증정보를 서비스토큰별로 임시 저장소에 저장한다.

$$AS \rightarrow S: (ST_U \| A_U \| B_U \| ID_U \| IP_U)_{SS_{AS,S}} \quad (18)$$

$$AS \rightarrow U: (ST_U \| N_U)_{SS_{AS,U}} \quad (19)$$

4.2.4 서비스서버와 사용자의 상호인증 단계

사용자는 발급받은 인증정보를 활용하여 해당 서비스서버에게 인증정보와 함께 인증요청을 하며, 서비스서버가 이를 확인함으로써 인증이 이루어지게 된다.

Step 1. 사용자 U는 서비스서버 S에게 인증요청을 위해 전송할 인증 값 K_U를 생성하여 암호화된 서비스토큰과 함께 전송한다.

$$U: K_U = H(ID_U \| TK_U \| ST_U \| N_U) \quad (20)$$

$$U \rightarrow S: (ST_U)_{TK_U}, K_U \quad (21)$$

Step 2. 서비스서버 S는 암호화된 서비스토큰을

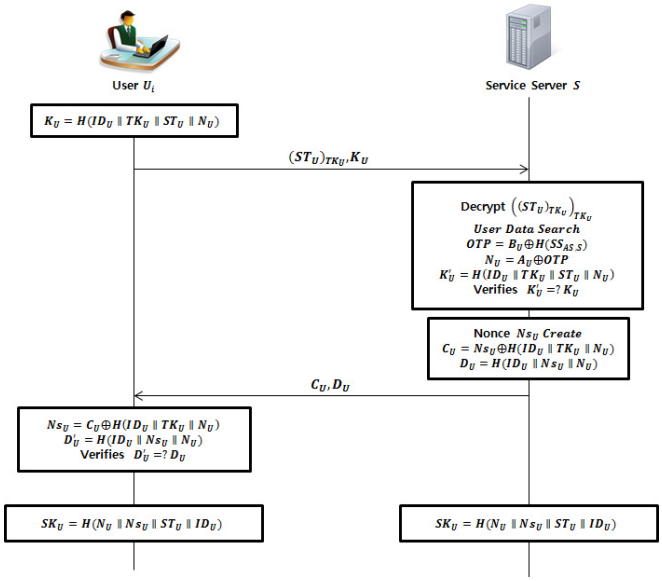

Fig.8. Proposed Scheme 2 : Authentication Phase

복호화 하고 ST_U를 통해 임시저장소에 저장된 인증정보를 검색하여 인증 값 생성을 위한 인증요소를 복원한다. 이를 통해 사용자 인증 값 K_U'을 생성하고 수신한 K_U와 비교하여 인증을 실시한다.

$$S: ((ST_U)_{TK_U})_{TK_U} \quad (22)$$
$$S: OTP = B_U \oplus H(SS_{AS,S}) \quad (23)$$
$$S: N_U = A_U \oplus OTP \quad (24)$$
$$S: K_U' = H(ID_U \| TK_U \| ST_U \| N_U) \quad (25)$$
$$S: K_U' = ? K_U \quad (26)$$

Step 3. 사용자 인증이 완료되면 서비스서버의 상호인증을 실시한다. 상호인증에 사용될 인증정보를 생성하기 위하여 Random Nonce Ns_U를 생성하고 이를 통해 인증정보를 생성하여 사용자 U에게 전송한다.

$$S: C_U = Ns_U \oplus H(ID_U \| TK_U \| N_U) \quad (27)$$
$$S: D_U = H(ID_U \| Ns_U \| N_U) \quad (28)$$
$$S \rightarrow U: C_U, D_U \quad (29)$$

Step 4. 사용자는 자신이 소유한 인증요소를 활용하여 Ns_U를 복원한다. 이를 통해 서비스서버 인증 값 D_U'을 생성하고 수신한 D_U와 비교함으로써 상호인증을 실시한다.

$$U: Ns_U = C_U \oplus H(ID_U \| TK_U \| N_U) \quad (30)$$
$$U: D_U' = H(ID_U \| Ns_U \| N_U) \quad (31)$$
$$U: D_U' = ? D_U \quad (32)$$

Step 5. 사용자와 서비스서버간의 상호인증이 완료되면 이후 통신에 사용될 비밀키를 생성한다.

$$U: SK_U = H(N_U \| Ns_U \| ST_U \| ID_U) \quad (33)$$
$$S: SK_U = H(N_U \| Ns_U \| ST_U \| ID_U) \quad (34)$$

4.2.5 문제점 보완사항

· 기밀성 : 실제 사용자의 민감한 정보를 사용하지 않고 Random Nonce N_U, Ns_U, OTP와 같은 간접정보를 사용하여 인증 값을 생성하고 인증을 실시하기 때문에 민감 정보 노출 위협이 존재하지 않는다.

· 무결성 : 서비스가 요청된 IP값을 서비스서버에 전송하여 관리됨에 따라 폼에 대한 위조 및 변조를 통한 부적절한 공격자의 접근에 대해 서버에서 파악할 수 있다.

· 인증정보 프라이버시 : 인증정보는 각 세션마다 새롭게 생성되는 랜덤 값으로 구성되며 안전한 SSL 프로토콜을 통해 분배된다. 또한 인증 요청을 위해 전

Table 2. Analysis of Proposed Schemes

		Jian[3]	Proposed Scheme 1	Lee[4]	Proposed Secheme 2
Confidentiality		○	○	○	○
		Symmetric-key Cryptography	Symmetric-key Cryptography	Symmetric-key Cryptography	Symmetric-key Cryptography
Integrity		○	○	○	○
Authentication		X	○	X	○
		Users can not distinguish	Offer	Credential Repudiation	Offer
Credential Privacy		○	○	○	○
Traffic	Issuing	-	-	2rounds	2rounds
	Authentication	Kerberos	Kerberos	Delivery Structure	Delivery Structure
		8rounds	8rounds	2rounds	2rounds
Computation	Issuing	-	-	3S+4R+1E+1X +1H	2S+2R+2X+2H
	인증	10S	9S	1S+1R+1E+3X +7H	1S+1R+4X+9H

○ : offer, secure △ : usually-offer × : non-offer, insecure
S : symmetric key cryptography H : hash function E : exponentiation X : XOR R : random number

송하는 인증 값 K_U도 세션별로 새롭게 생성되기 때문에 공격자의 재전송 공격으로부터 안전하다.

V. 제안방식 분석

제안방식 1의 경우 Jian 방식의 통신 프로토콜을 기반으로 제안된 방식이다. 기존 Jian 방식의 경우 클라이언트에 대한 식별이 불가능하여 암호화된 데이터에 해당하는 클라이언트의 비밀키 획득이 불가능하도록 설계되어 있다. 이에 사용자 인증이 불가능하여 올바른 서비스 제공이 이루어지지 않는다. 뿐만 아니라 불필요한 Nonce, 키 값 생성 및 전달이 이루어져 서비스 전체에 대한 계산 낭비 및 부담감을 준다. 이에 제안방식 1은 사용자 식별이 원활하도록 구성하여 사용자 인증 서비스가 가능하도록 제안하였으며, 불필요한 데이터 생성 및 전달을 제거함으로써 효율성을 증가시켰다.

제안방식 2의 경우 Lee 방식을 기반으로 제안된 방식이다. 기존 Lee 방식의 경우 사용자 인증 시 전달되는 인증정보를 구성함에 있어 랜덤으로 생성된 값을 사용한다. 이는 부인 가능성 및 공격자의 인증 시도가 가능한 문제점이 존재하게 된다. 뿐만 아니라 필요 이상으로 랜덤 값을 생성하여 전달하고 지수승 연산을 실시함에 따라 연산 부담감을 준다. 이에 제안방식 2는 사용자 인증 시 간접적인 인증 값을 생성하고 전달하여 통신상 인증정보가 노출되지 않도록 한다. 또한 기존방식 5R의 불필요한 랜덤값 생성을 3R로 최소화 하였으며, 해시연산이 11H로 다소 증가되었지만 2E의 지수승 연산을 제거함으로써 경량화 효과를 가져다준다.

VI. 결 론

인터넷 환경의 변화로 인해 각종 서비스들은 웹 기반 형태로 변화하고 있으며 각 서비스를 이용하기 위해서는 사용자별 인증 정보를 요구하게 된다. 이에 따라 사용자측면에서는 다양한 웹 서비스 별로 개별적인 아이디와 패스워드를 설정하고 기억해야하는 문제점이 존재하며, 서비스 업체 측면에서는 중복 사용자들의 인증정보를 따로 관리해야하는 문제점이 발생되었다. 이에 한 번의 인증을 통해 다양한 서비스를 이용할 수 있는 SSO 인증 시스템을 도입하게 되었다.

SSO 인증 시스템에 있어 인증 모델의 선택과 서비스 제공자의 환경에 따른 인증 방식의 선택은 매우 중요한 요소로 작용하게 된다. 이에 다양한 인증모델

과 인증 서비스 형태가 개발되어 제공되고 있다.

기존 방식 중 Jian 제안방식의 경우 잘못된 프로토콜 설계로 인해 정상적인 인증 서비스가 이루어지지 않는 형태로 제안되어 있어 그에 대한 보완과 더불어 불필요한 데이터 생성 및 분배 과정을 개선하였다. 이에 본 논문에서는 사용자에게 전송된 티켓 정보를 복호화하기 위한 키 값 검색이 원활하게 이루어지도록 설계하였으며, *Nonce*를 포함한 불필요한 데이터를 제거함으로써 인증 프로세스의 효율성을 증가시켰다.

Lee 제안방식의 경우 불필요한 데이터 생성 및 연산, 인증정보 부인, 연산 과부하의 문제점을 가지고 있어 통신 횟수는 동일하되 연산 효율성을 증가시킨 방식을 제안하였다. 이에 본 논문에서는 다소 XOR 및 해시연산의 횟수 증가가 발생하였지만 기존 방식에 있어 지수승 연산 사용을 제거함으로써 연산 효율성을 증가시켰다. 증가된 XOR 연산의 경우 단순한 비트연산이며, 해시연산의 경우 지수승 연산에 비해 월등히 빠른 속도를 보이고 해시 알고리즘에 안전성을 기반하고 있어 연산 경량화의 효과를 얻을 수 있다.

하지만 본 논문에서 제안한 두 방식의 경우 전체적인 인증 과정이 모두 비밀키 방식을 택하고 있다는 점에서 공개키 방식의 시스템보다 안전성이 미흡할 수 있다고 판단된다. 이에 향후 공개키 방식을 활용한 SSO 인증 프로토콜에 대한 확장된 연구가 필요할 것으로 판단된다.

References

〔1〕 A. Volchkov, “Revisiting Single Sign-On:A Pragmatic Approach in a New Context,” IT Professionals, vol. 3, no. 1, pp. 39-45, Feb. 2001.

〔2〕 Dae-Hee Seo and Im-Yeong Lee, “A Study on Single Sign-On Authentication Model using Multi Agent,” The Journal of Korea Information and Communications Society, 29(7C), pp. 997-1006, June. 2004

〔3〕 Yang Jian, “An Improved Scheme of Single Sign-on Protocol,” Fifth International Conference on Information Assurance and Security, vol. 1, pp. 495-498, Aug. 2009.

〔4〕 Seung-Ah Lee, ″A Robust SSO(Single Sign-On) Authentication Method against Replay Attack,″ M.S. Thesis, Chonbuk National University, Feb. 2012

〔5〕 Hyun-Jin Kim and Im-Yeong Lee, ″A Study on Secure Lightweight Single Sign-On Mechanism against Credential Replay Attack,″ The 40th Conference of the KIPS, pp. 811-814, Nov. 2013

〔6〕 Soo-Jin Park, Il-Sun You and Yong-Rak Choi, “Requirements Analysis for the Design of SSO Service System,” Conference of the KSII, pp. 378-382, May. 2001

〔7〕 C. Neuman, S. Hartman, K. Raeburn, and T. Yu, ″The Kerberos Network Authentication Service(V5),″ RFC 4120, July. 2005.

〔8〕 Young-Jae Maeng and Dae-Hun Nyang, ″An Analysis of Replay Attack Vulnerability on Single Sign-On Solutions″, Journal of The Korea Institute of Information Security & Cryptology, 18(1), pp. 103-114, Feb. 2008

〈저 자 소 개〉

김 현 진 (Hyun-Jin Kim) 학생회원
2013년 2월: 순천향대학교 컴퓨터소프트웨어공학과 졸업
2013년 3월~현재: 순천향대학교 컴퓨터학과 석사과정
〈관심분야〉 암호프로토콜, 인증, 전자서명, 컴퓨터보안

이 임 영 (Im-Yeong Lee) 종신회원
1981년 2월: 홍익대학교 전자공학과 졸업
1986년 2월: 오사카대학 통신공학전공 석사
1989년 2월: 오사카대학 통신공학전공 박사
1989년~1994년: 한국전자통신연구원 선임연구원
1994년~현재: 순천향대학교 컴퓨터소프트웨어공학과 교수
〈관심분야〉 암호이론, 정보이론, 컴퓨터보안