J. KSIAM **Vol.3**, No.1, 17-29, 1999

일회성 티켓을 필요로 하는 사용자에게 동기화된 시계를 요구하지 않는 방식의 Kerberos *

김해영, 한상근

요 약

The reliable authentication of a communicating party and a network component is an essential factor to achieve the security in a computer network. The Kerberos Authentication Services has been the most successful solution which is widely used today but its requirement for synchronized clocks has been a serious limitation to use it.

In this paper we presented an extended Kerberos method which avoids the synchronization requirement for a single-time ticket user. We modified the Kerberos protocol minimally by replacing the synchronization requirement with the challenge-response method.

1 서론

통신을 하고 있는 둘 또는 그 이상의 대상들은 자신의 정보에 대한 안전과 기밀성을 유지하기 위해 통신 대상의 신원을 확인할 방법이 필요하다. 이와 같은 필요성에 의해 통신 대상간에 실제 신원 확인을 제공하는 과정을 인증(Authentication)이라 하고, 인증의 기능을 수행하는 프로토콜을 인증 프로토콜(Authontication protocol)이라고 한다. Kerberos는 네트워크 인증 프로토콜로, 사용자와 서버가 비밀키 암호를 이용하여 인증 기능을 제공하도록 MIT에서 Athena 프로젝트의 부분으로 개발되었다.

Kerberos 인증 시스템에서 동기화된 시계는 사용자가 보내는 메시지에 포함된 Timestamp 정보를 읽어내어 사용자가 보낸 메시지가 시기 적절한가를 확인한다. 즉, 동기화된 시계를 통하여 시기 적절하지 않는 정보(재전송 공격)를 거부한다.

서로 연관관계가 없는 두 개 이상의 Kerberos 영역에 등록된 사용자가 있다고 가정해보자. 이 사용자는 서비스를 받고자 하는 Kerberos 영역이 바뀔 때 마다 자신의 시계를 고쳐야 하는 번거로움이 생긴다. 이 경우 만약 사용자가 특정 영역에서는 소수의 서버에만 접속 하고자 한다면, 영역이 바뀜에 따라 사용자에게 동기화된 시계를 요구하는 것이 더욱 어려워진다. 본 논문에서는 이와 같이 영역내의 상대적으로 작은 수

*본 연구는 1997년도 정보통신부 출연과제 지원에 의하여 수행된 것임

의 서버에 접속을 원하는 사용자에게 편의를 제공하기 위해 동기화된 시계를 요구하지 않는 방식의 Kerberos인 Kerbers-T 방식을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 Kerberos V5 사용자로부터 동기화된 시계를 제거하되 공유해야 할 정보를 증가시키지 않을 목적으로 도입될 시도·응답 방식의 UofV 프로토콜에 대해 살펴본다. 3장에서는 Kerberos V5의 특징과 인증 메카니즘에 대해 알아본다. 그리고 4장에서는 앞장의 내용을 바탕으로 작은 수의 서버에 접속하고자 하는 사용자에게 서버와의 동기화된 시계를 제거한 Kerberos-T 방식을 제시한다. 그리고 마지막으로 결론부를 5장에서 언급한다.

2 University of Virginia 인증 방식

University of Virginia(UofV) 인증은 W. A. Wulf, A. Yasinac, K. S. Oliver, R. Peri에 의해 개발되었다 [1]. UofV 인증 방식은 다변수 일방향 함수(multiple one-way function)를 이용하여 신분 위장(masquerade), 패스워드 추측, 재전송 공격을 방지한다. UofV 인증 방식의 특징은 통신을 하고자 하는 두 대상(갑,을)이 특정 정보를 공유하지 않고 실체 인증을 제공하는데 있다. 이 방식은 실체 인증에서 좋은 신뢰도를 제공한다. 간단한 예제를 제시하면 다음과 같다. 실체 인증을 받고자 하는 상대방(proover) 갑은 본인만이 대답할 수 있는 질문에 답을 해야만 한다. 여기에서 상대 을은 갑의 답을 예상할 수 없으나, 갑이 제시한 답의 맞고 틀림은 분별할 수 있다. 이 방법은 패스워드나 비밀키의 사용이 필요하지 않는다. 결과적으로 패스워드나 비밀키의 관리 문제를 해결하였다. UofV 인증방식은 다음과 같은 성질에 기초한다.

- 일방향 함수 f, g는 주어진 함수값에 대해 그 역상을 구하기 어렵다.
- 두 함수 f, g는 교환법칙이 성립한다. $g(f(x), f(y)) = f(g(x, y))$

프로토콜의 시작자인 갑은 다음과 같은 과정을 거쳐 반응자 을에게 본인이 을과 대화하고 있음을 증명한다.

1 두 상대방 갑과 을은 module 단위로 쓰일 소수 n을 약속한다.

2 갑은 을과의 통신을 시작하고자 할 때 을에게 본인(갑)임을 확인하는 데 쓰일 함수(f)를 선택한다.

3 갑은 임의의 정수 x를 택한다.

4 갑은 x와 $f(x) = x^a \bmod n$을 을에게 보낸다.

5 을은 x와 x^a를 이용하여 이후 갑으로부터 오는 인증 요청을 검증할 것이다.

6 을은 함수 g와 임의의 수 y를 선택하고, y와 $g(x,y) = (xy)^b \bmod n$을 갑에게 보낸다.

7 갑은 함수값 $f(y) = y^a \bmod n$과 $f(g(x,y)) = g(x,y)^a \bmod n$을 생성하고 을에게 보낸다.

8 을은 $g(f(x), f(y)) = ((x^a \bmod n) * (y^a \bmod n))^b \bmod n$을 계산하여 갑으로부터 받은 $f(g(x,y))$와 비교한다.

9 위의 두 값이 같다면 을은 통신하는 상대방이 프로토콜의 시작자 갑임을 알 수 있다.

이 프로토콜 보안에서 가장 중요한 요소는 함수 f, g 다. UofV 프로토콜의 핵심은 갑 이외에는 $g(f(x), f(y))$를 예측하지 못하는데 있다. 함수 f와 g의 형태(form)는 공개되지만, 실제 함수 f와 g의 생성 요소인 a와 b는 비밀성이 유지되어야 한다. 갑의 실체 인증시 갑은 인증자 $(f(y), f(g(x,y))$를 생성함으로써 본인임을 증명하므로, 실제 함수 f의 생성 요소 a의 공개는 임의의 사람이 갑으로 가장할 수 있게 한다.

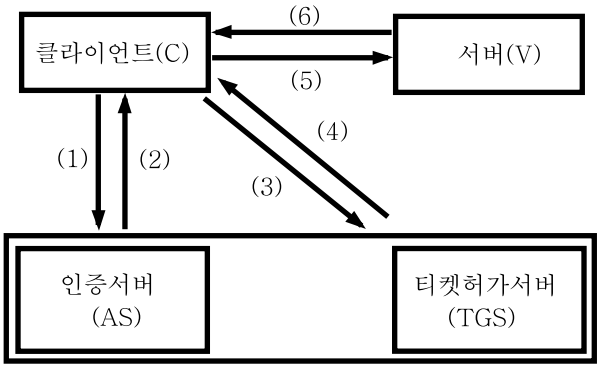
함수 f와 g의 형태(form)의 공개를 통하여, f를 전송하거나, 통신상에서 f를 전송하여 제 3의 신뢰기관에서 f의 형태를 점검 받을 필요가 없도록 한다.

UofV 프로토콜은 신뢰기관을 필요로 하지 않으므로, 분산처리 환경에 적합하다. UofV 프로토콜은 다양한 인증자 (예:$(x, f(x))$)의 선택을 통하여 재전송 공격을 방어하는데, 인증자는 임의의 수 x와 지정된 형식의 함수 f를 선택함으로써 얻어져 그 선택의 폭이 매우 넓음을 알 수 있다.

3 Kerberos V5 인증 메카니즘

Kerberos 는 MIT에서 Athena 프로젝트의 부분으로 개발된 비밀키 네트워크 인증 프로토콜로 Needham-Shroeder의 인증 프로토콜에 그 기초를 두고 있다 [5] [6]. Kerberos는 분산형 환경에 적합한 인증 메카니즘으로 현재 Kerberos 버전 4와 5가 사용되고 있으며, Kerberos 버전 5(V5)는 인터넷 Draft 표준화, RFC 1510으로 발표되었다.

Kerberos 의 전체적인 인증 메카니즘을 그림으로 보면 다음과 같다.



메시지 1에서 사용자가 인증 서버에 티켓 허가 티켓을 요구하면 인증 서버는 Kerberos 데이터 베이스로부터 사용자의 해쉬된 비밀번호를 이용하여 암호화한 티켓 허가 티켓을 생성하고 메시지 2에서 사용자에게 전송한다. 사용자는 해쉬된 사용자의 패스워드를 이용하여 메시지 2를 복호화하고 자신의 컴퓨터에 보관한다. 서비스를 받고자 할 때 사용자는 티켓 허가 티켓을 포함한 메시지 3을 보낸다. 메시지 3을 받은 TGS는 메시지의 정당성을 확인한 뒤 서비스 티켓을 포함한 메시지 4를 보낸다. 메시지 5에서 사용자는 서버에게 서비스 티켓과 세션키의 소유를 증명함으로써 실체 인증을 받는다. 사용자가 서버의 인증을 원하는 경우에 한해서 메시지 6을 통해 서버 인증을 받는다.

앞으로 나올 기호와 용어에 대해 간단히 살펴보면 다음과 같다.

- Realm(영역) : Kerberos의 서비스를 받을 수 있는 자격을 갖는 클라이언트 또는 서버들의 집합 또는 그들이 형성한 네트워크 도메인이라고 말할 수 있다. 즉, 하나의 Kerberos 서버가 영향력을 미치는 범위이다.
- Option(선택사항) : 사용자가 발급 받고자 하는 티켓의 성격을 나타낸다.
- Realm_C: C가 소속된 영역
- ID_C : C의 ID
- AD_C : C의 IP Address
- K_C : C의 비밀키
- $\text{E}_K[M]$: 비밀키 K를 이용한 M의 암호화
- Times : 티켓의 유효기간
- TS : 현재 시간을 나타내는 Timestamp
- Nonce : 통신 대상이 생성하는 임의의 수, 이전의 정보로부터 예측할 수 없는 특징을 가짐

- $\text{Ticket}_{\text{TGS}}$: TGS에게 전송될 티켓 허가 티켓
- Ticket_{V} : 서버 V에게 전송될 서비스 티켓

(1) 클라이언트-인증 서버간의 교환

<메시지 1>

$\text{C} \longrightarrow \text{AS} : \text{Options}||\text{ID}_{\text{C}}||\text{Realm}_{\text{C}}||\text{ID}_{\text{TGS}}||\text{Times1}||\text{Nonce1}||\text{AD}_{\text{C}}$

<메시지 2>

$$\text{AS} \longrightarrow \text{C} : \text{Realm}_{\text{C}}||\text{ID}_{\text{C}}||\text{Ticket}_{\text{TGS}}||$$
$$\text{E}_{\text{K}_{\text{C}}}[\text{K}_{\text{C,TGS}}||\text{Nonce1}||\text{Flag}||\text{Times1}||\text{Realm}_{\text{TGS}}||\text{ID}_{\text{TGS}}||\text{AD}_{\text{C}}]$$
$$\text{Ticket}_{\text{TGS}} = \text{E}_{\text{K}_{\text{TGS}}}[\text{Flag}||\text{K}_{\text{C,TGS}}||\text{Realm}_{\text{C}}||\text{ID}_{\text{C}}||\text{Times1}||\text{AD}_{\text{C}}]$$

메시지 1은 사용자가 AS에게 티켓 허가 티켓을 요구하는 단계이다. 메시지 1에서 사용자는 선택한 Option과 자신의 ID와 자신이 속해있는 영역의 ID, 서비스를 받고자 하는 티켓 허가 서버의 ID, 자신이 요구하는 티켓의 유효기간, 반복 되지 않는 임의의 수를 평문으로 보낸다. C의 IP Address 전송은 선택적이다. 티켓의 유효기간 요구 필드는 경우에 따라 공란으로 보내질 수도 있는데, 그 경우는 4장에서 설명한다.

반복되지 않는 임의의 수(Nonce1)는 Kerberos V4에는 없던 형식으로 메시지 2에서 AS로부터 반환되어지는데 메시지 1과 메시지 2를 연결시키는 기능을 하고 AS의 응답이 현재 시간이고 제삼자에 의한 재전송 공격에 의한 메시지가 아님을 나타내는 역할을 한다. 메시지1은 평문으로 전송되기 때문에 도청자는 메시지 1을 모두 읽을 수 있다.

메시지 2는 AS의 메시지 1에 대한 응답으로 티켓 허가 티켓과 사용자의 비밀키로 암호화된 정보를 포함한다. 사용자의 티켓 허가 티켓은 사용자의 요구(Option)에 의해 생성된 Flag, 사용자와 티켓 허가 서버가 공유할 세션키, 사용자의 신원, 티켓의 유효기간을 AS와 TGS가 공유한 비밀키로 암호화한 정보이다. 티켓 허가 티켓은 AS와 TGS만 아는 비밀키로 암호화 되었기 때문에 사용자와 도청자는 그 내용을 볼 수 없다. 사용자의 신원 정보에서 AD_{C}는 사용자가 티켓의 사용을 위해 접속해야 하는 호스트의 주소를 나타낸다. 이것은 선택사항이고, 도청된 티켓의 재사용을 어렵게 하기 위해 고안된 것이다. 사용자의 호트가 기록되지 않은 티켓은 모든 주소에서 사용될 수 있다. 사용자는 메시지 2에서 TGS와 공유할 세션키를 얻고 자신이 생성해서 보냈던 Nonce1, 티켓의 유효기간, 서비스를 신청했던 TGS의 ID 등을 전송받아 메시지 1에서 전송된 메시지의 변경 여부를 확인한다. 메시지 2를 통해 AS는 사용자와 TGS에게 안전하게 세션키를 분배함을 알 수 있다.

(2) 클라이언트-티켓 허가 서버간의 교환

<메시지 3>

$C \longrightarrow TGS$:

$Options||ID_V||Times2||Nonce2||Ticket_{TGS}||Authenticator_c$

$Ticket_{TGS} = E_{K_{TGS}}[Flag||K_{C,TGS}||Realm_C||ID_C||Times1||AD_C]$

$Authenticator_c = E_{K_{C,TGS}}[ID_C||Realm_c||TS1]$

<메시지 4>

$TGS \longrightarrow C : Realm_C||ID_C||Ticket_V||$

$E_{K_{C,TGS}}[K_{C,V}||Nonce2||Flag||Times2||Realm_V||ID_V||AD_C]$

$Ticket_V = E_{K_V}[Flag||K_{C,V}||Realm_C||ID_C||Times2||AD_C]$

메시지 3은 사용자가 서비스를 받고자 할 때 TGS에게 보내어진다. 사용자는 메시지 3에서 자신이 선택한 Option과 서비스를 받고자 하는 서버의 ID, 자신이 요구하는 티켓의 유효기간, 반복되지 않는 임의의 수를 평문의 형태로 보낸다. 또한, 메시지 2에서 받은 티켓 허가 티켓과 C가 생성한 인증자를 보내다. 여기에서 티켓은 유효기간 내의 재사용이 가능한 반면 인증자(Authenticator)는 일회성으로 AS가 분배한 세션키로 암호화되어 있다. 인증자에 기록된 사용자의 ID와 영역은 티켓에 기록된 것과 일치해야 한다. 서버는 인증자를 적당한 시간 동안 보관하여 동일한 사용자가 같은 Timestamp를 갖는 인증자를 보내올 경우 메세지를 거부한다. 메시지 1에서와 마찬가지로 사용자는 경우에 따라 티켓의 유효기간 요구 필드를 비울 수 있다. 이 예외적인 경우도 4장에서 설명하도록 한다. 반복되지 않는 임의의 수(Nonce2)는 메시지 4에서 AS로부터 반환 되어지는데 이것은 메시지 3과 메시지 4를 연결 시키는 기능을 하고 TGS의 응답이 현재 시간이고 제삼자의 재전송 공격에 의한 메시지가 아님을 나타내는 역할을 한다.

메시지 3을 받은 TGS는 AS와 공유한 비밀키로 티켓을 복호화하고 세션키를 얻어낸다. TGS를 제외한 세션키 소유자는 C뿐이므로 세션키를 이용해 인증자를 복호화하여 사용자의 ID와 Timestamp를 확인함으로써 티켓의 소유자가 티켓을 보낸 사람이라는 것을 알 수 있다. TGS로부터의 응답 메시지 4는 메시지 2와 동일한 형식으로 동일한 역할을 한다. 즉, C와 V가 공유할 세션키를 C와 V에게 안전하게 분배하고, Nonce2를 반환하여 TGS의 응답이 현재이고 재전송 공격에 의한 메시지가 아님을 알린다.

(3) 클라이언트-서버간의 교환

<메시지 5>

$C \longrightarrow V : Options||Ticket_V||Authenticator_c$

$$\text{Ticket}_{\text{V}} = \text{E}_{\text{K}_{\text{V}}}[\text{Flag}||\text{K}_{\text{C,V}}||\text{Realm}_{\text{C}}||\text{ID}_{\text{C}}||\text{Times2}||\text{AD}_{\text{C}}]$$

$$\text{Authenticator}_{\text{c}} = \text{E}_{\text{K}_{\text{C,V}}}[\text{ID}_{\text{C}}||\text{Realm}_{\text{c}}||\text{TS2}||\text{Subkey}||\text{Seq\#}]$$

<메시지 6>

$$\text{V} \longrightarrow \text{C} : \text{E}_{\text{K}_{\text{C,V}}}[\text{TS2}||\text{Subkey}||\text{Seq\#}]$$

메시지 5에서 사용자는 서버에게 서비스 티켓과 인증자를 통해 세션키의 소유를 증명함으로써 실체 인증을 받는다. Authenticator는 메시지 3에서와 같은 역할을 하고, 선택적으로 Subkey와 Seq#를 추가할 수 있다. 추가된 Subkey 필드의 Subkey는 현재 세션에 사용될 키를 V에게 분배하는데 쓰이고, 만약 이 필드가 생략된다면 현재 세션은 TGS 가 분배한 세션키를 사용하게 된다. 선택적으로 추가되는 Seq#는 서버가 현재 세션에 교환될 메시지에 일련 번호를 매기는 것을 요청한다. 메시지의 일련 번호는 재전송 공격을 방어한다. 메시지 5를 받은 서버는 선택 사항인 서버 인증 요구가 있으면 메시지 6으로 응답한다. 이 때, 메시지 5의 선택 사항 Subkey, Seq#의 요청이 있으면 Subkey와Seq#를 추가적으로 보낸다. 메시지 6의 응답시 TS2를 반환하는 것은 메시지 2, 메시지 4에서 Nonce를 반환하는 것과 같은 역할을 한다.

사용자가 KDC(AS와 TGS)에게 보내는 티켓의 Option 필드는 KDC 에게 사용자가 원하는 특정한 Flag를 요구하는 필드이다. Keberos V4에는 없던 티켓의 Flag 필드는 티켓의 특수한 기능을 나타내는 필드로 사용자의 Option 요구와 KDC의 판단에 의해 티켓에 기록되는 정보이다. 티켓의 forwardable flag는 사용자에게 다른 IP Adress에서의 접속을 허락받는 티켓을 받을 수 있도록 허가하고, renewable flag는 장기간 서비스를 받고자 하는 사용자에게 편의를 제공하기 위한 방법으로 티켓 재발급 기간내에 확장된 유효기간을 갖는 티켓을 받을 수 있도록 허가하고, postdatable flag는 현재 시간이 아닌 미래 시간부터 유효기간을 갖는 티켓을 받을 수 있도록 허락하는 정보이다.

4 동기화된 시계를 요구하지 않는 Kerberos

Kerberos 인증 시스템에서 동기화된 시계는 Timestamp 정보를 읽어 냄으로써 정보가 시기 적절한가를 확인한다. 즉, 동기화된 시계를 통하여 시기 적절하지 않는 정보를 거부한다 [4].

Kerberos의 사용자는 Timestamp를 암호화된 비밀 정보에 기록해 서버가 시기 적절하지 않는 비밀 정보를 거부할 수 있도록 한다. Kerberos V4에서는 사용자가 제시하는 비밀 정보 내의 Timestamp가 서버의 현재 시간과 5분 이내의 시간이면 합당한 정보로 받아들이고, 5분 이상이 지난 시간이면 재전송 공격으로 간주하고 정보를 거부한다. 최소 5분의 동기화를 요구하는 Kerberos V5는 5분이내에 재사용된 사용자의

정보도 거부할 수 있도록 각 서버가 받은 모든 암호화된 Timestamp를 저장하여 서버에게 받아들여진 정보에 포함된 시간과 비교함으로써 재사용 여부를 검증하는 기능을 제공한다.

사용자의 시계가 서버의 시계보다 빠르면 서버는 재전송 공격에 의한 정보와 합당한 정보를 구별할 수 없게 되어 합당한 정보를 거부하는 경우가 생긴다. 마찬가지로 사용자의 시계가 서버의 시계보다 느리면 사용자의 정보는 공격자가 기밀 정보를 사용할 수 있는 보다 긴 시간을 제공함으로써 재전송 공격을 쉽게 한다.

위와 같이 동기화된 시계를 이용하는 방식의 인증 시스템은 여러 단점을 가지고 있을 뿐만 아니라, network상에서 폭 넓은 사용자를 갖기 위해서는 경우에 따른 사용자의 동기화된 시계의 제거가 바람직하다. 본 장에서는 앞서 2장에서 제시한 UofV 프로토콜을 Kerberos 프로토콜에 이용해서 동기화된 시계를 요구하지 않는 Kerberos (Kerberos-T)방식을 제안한다.

본 논문에서는 사용자가 영역내의 소수의 서버에만 접속을 하고자 한다고 가정한다. 이러한 사용자는 티켓 허가 서버로부터 서비스 티켓을 받는 횟수(메시지 교환 3, 4)가 적다고 볼 수 있다. 이러한 경우 사용자가 서비스 티켓을 받고자 할 때(메시지 교환 3,4)마다 메시지 교환 1, 2를 불러도 프로토콜의 총 메시지 교환 횟수가 그다지 증가하지는 않는다. 즉, 소수의 서버에 접속하고자 하는 사용자의 경우 Kerberos V5에서의 티켓 허가 티켓을 일회성으로 사용하더라도 프로토콜의 메시지 총 교환 횟수가 그다지 증가하지 않음으로, 본 논문에서 사용자는 메시지 교환1부터 4까지를 연속적으로 하는 일회성 티켓 사용자라고 가정 한다.

AS와 TGS는 서로 신뢰하는 기관이고, 물리적으로 가까이 있을 뿐만 아니라 Kerberos 인증 메카니즘의 구조상 이미 공유키를 가지고 있기 때문에 Kerberos의 AS와 TGS를 KDC라는 하나의 주체로 볼 수 있다. 실제로 AS와 TGS는 Kerberos 데이타 베이스를 공유하고 있다.

UofV 프로토콜의 이용 방법은 다음과 같다. UofV 프로토콜의 두 주체를 사용자 C와 KDC로 보고 UofV 프로토콜을 Kerveros V5에 다음과 같은 메시지를 메시지 1, 2, 3의 Nonce 필드, Timestamp 필드 그리고 원문 필드에 삽입한다. 메시지 $2'$의 송신은 C와 AS와의 대화(UofV 프로토콜로 볼 때)를 C와 TGS의 대화로 연장시키기 위해 필요한 절차이다.

<메시지 1>

C $\longrightarrow$ AS(=KDS) : $x||f(x)$

<메시지 2>

AS(=KDS) $\longrightarrow$ C : $y||g(x,y)$

<메시지 $2'$ >

AS $\longrightarrow$ TGS(=KDS) : $f(x)||E_{K_{TGS}}(g)$

<메시지 3>

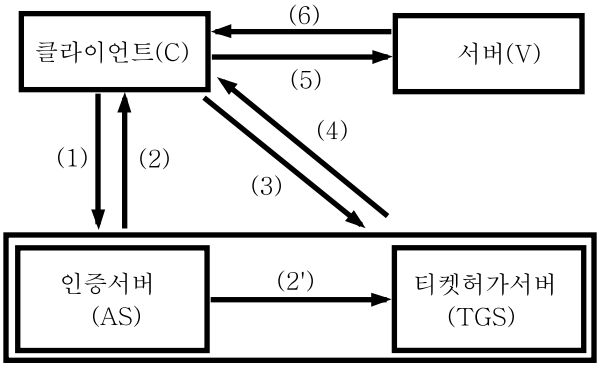
C $\longrightarrow$ TGS(=KDS) : $f(y)||f(g(x,y))$

위와 같은 방법으로 C와 KDC간의 UofV 프로토콜을 Kerberos V5에 삽입함으로써 TGS에게 C가 메시지 1을 AS에게 보낸 사용자라는 것을 확인시켜 준다.

즉, 사용자가 KDC(AS와 TGS)와의 대화시 Kerberos의 Nonce 필드와 Timestamp 필드를 이용하여 UofV 시도·응답 방식 프로토콜을 삽입함으로써 TGS가 사용자에게 현재 시간을 요구하지 않고도 사용자 인증과 메시지 재전송 공격을 방어할 수 있게 한다.

사용자와 서버 간의 교환은 난수를 이용하는 시도·응답 방식을 택하여 사용자에게 Timestamp를 요구하지 않는 메시지 교환을 허락한다.

Kerberos-T 방식은 다음과 같다. 사용자, AS, TGS는 n(큰 소수)를 이미 알고 있고, AS, TGS, Server들의 시계는 안전한 timming 서비스에 의해 동기화되어 있다고 가정한다. 앞으로 나올 x, y,f,g는 위에서 제시된 임의의 수와 함수이다.



(1) 클라이언트-인증 서버-티켓 허가 서버간의 교환

<메시지 1>

C $\longrightarrow$ AS : $\mathrm{Options}||\mathrm{ID_C}||\mathrm{Realm_C}||\mathrm{ID_{TGS}}||x||f(x)$

<메시지 2>

AS $\longrightarrow$ C :$\mathrm{Realm_C}||\mathrm{ID_C}||\mathrm{Ticket_{TGS}}||y||g(x,y)||$

$\mathrm{E_{K_C}}[\mathrm{K_{C,TGS}}||x||\mathrm{Times1}||\mathrm{Flag}||\mathrm{Realm_{TGS}}||\mathrm{ID_{TGS}}||\mathrm{AD_C}]$

$\mathrm{Ticket_{TGS}} = \mathrm{E_{K_{TGS}}}[\mathrm{Flag}||\mathrm{K_{C,TGS}}||\mathrm{Realm_C}||\mathrm{ID_C}||\mathrm{Times1}||\mathrm{AD_C}]$

<메시지 $2'$ >

AS $\longrightarrow$ TGS : $C||f(x)||E_{K_{TGS}}(g)$

메시지 1의 경우 이전의 프로토콜에서 사용자가 티켓의 유효기간을 요구하는 Times1을 비워두고 Nonce 필드에 사용자가 생성한 임의의 수 x에 $f(x)$를 덧붙인 $x||f(x)$를 놓는다. $x, f(x)$는 2장에서 제시한 UofV 프로토콜에서 갑이 제시하는

수 $x, f(x)$와 동일한 역할을 한다. 즉, 함수 f에 대한 지식을 통해 실체 갑(C)에 대한 인증을 한다. UofV 프로토콜에서는 을(AS)과의 연속적인 대화를 통해 갑(C)이 을(AS)에게 본인임을 확인시킨 반면, Kerberos-T에서는 을(AS)이 자신과 공통키를 갖는 제삼자(TGS)와 을(AS)의 정보를 공유함으로써 갑이 제삼자(TGS)에게 본인임을 확인받는다.

메시지 2의 경우를 보자. Kerberos V5에서 사용자가 메시지 1에서 Times1(티켓의 유효기간)을 제시하지 않을 경우 AS는 티켓 유효기간의 시작 시간을 티켓 발행 시간으로 부여한다. 즉, 메시지 2에서 AS가 생성한 Times1은 다음과 같다.

$$\mathrm{Times1} = (\mathrm{T_{start}}, \mathrm{T_{end}}, \mathrm{T_{renew}})$$

위에서 $\mathrm{T_{start}}$, $\mathrm{T_{end}}$, $\mathrm{T_{renew}}$는 각각 티켓 발행 시간, 티켓의 유효기간 만료 시간, 티켓 재발급 만료 시간을 나타낸다.

메시지 1의 경우에서와 마찬가지로 AS는 Nonce 필드에 C에게서 받은 임의의 수 x를 놓는다. 여기에서 x는 C의 시도(Challenge) x에 대한 응답으로 AS의 응답이 현재 시간임을 나타낸다. Kerberos V5에서 Nonce1과 같은 역할을 한다. $y||g(x,y)$는 후에 C가 TGS에게 본인임을 확인 시키기 위해 쓰일 정보이다. 비밀성이 요구되지 않음으로 원문 필드에서 보내진다.

AS로부터 암호화된 정보를 받은 C는 자신의 비밀키로 복호화하고, 티켓 발행 시간($\mathrm{T_{start}}$)과 본인이 사용하는 터미널의 시간($\mathrm{T_C}$)을 확인하여, 티켓의 유효기간을 자신의 터미널 시간에 맞춰 해석한다. 사용자 C의 터미널에서 읽어내는 티켓의 유효기간은 다음과 같다 [2].

$$(\mathrm{T_C}, \mathrm{T_C} + (\mathrm{T_{end}} - \mathrm{T_{start}}), \mathrm{T_C} + (\mathrm{T_{renew}} - \mathrm{T_{start}}))$$

메시지 $2'$을 보자. 이것은 이전의 프로토콜에 없었던 메시지 전송으로 C와 AS와의 대화를 C와 TGS의 대화로 연장 시키기 위해 필요한 절차이다. TGS에게 서비스 티켓을 요구하고 있는 대상이 AS와 통신을 한 대상이 맞는지를 TGS가 확인할 수 있도록 AS가 보내는 정보다. TGS는 메시지 $2'$를 C에게서 서비스 티켓 요청이 올 때까지 훼손되지 않게 보관한다.

(2) 클라이언트-티켓 허가 서버간의 교환

<메시지 3>

$\mathrm{C} \longrightarrow \mathrm{TGS} : \mathrm{Options}||\mathrm{ID_V}||f(y)||\mathrm{Ticket_{TGS}}||\mathrm{Authenticator_c}$

$\mathrm{Ticket_{TGS}} = \mathrm{E_{K_{TGS}}}[\mathrm{Flag}||\mathrm{K_{C,TGS}}||\mathrm{Realm_C}||\mathrm{ID_C}||\mathrm{Times1}||\mathrm{AD_C}]$

$\mathrm{Authenticator_c} = \mathrm{E_{K_{C,TGS}}}[\mathrm{ID_C}||\mathrm{Realm_C}||\mathrm{f(g(x,y))}]$

<메시지 4>

$\mathrm{TGS} \longrightarrow \mathrm{C} : \mathrm{Realm_C}||\mathrm{ID_C}||\mathrm{Ticket_V}$

$||\mathrm{E_{K_{C,TGS}}}[\mathrm{K_{C,V}}||f(y)||\mathrm{Times2}||\mathrm{Flag}||\mathrm{Realm_V}||\mathrm{ID_V}||\mathrm{AD_C}]$

$\mathrm{Ticket_V} = \mathrm{E_{K_V}}[\mathrm{Flag}||\mathrm{K_{C,V}}||\mathrm{Realm_C}||\mathrm{ID_C}||\mathrm{Times2}||\mathrm{AD_C}]$

메시지 3의 경우는 메시지 1의 경우와 마찬가지로 Times2를 비워둔다. Nonce 필드에 사용자가 생성한 $f(y)$를 놓는다. y와 $g(x,y)$는 메시지 2에서 AS로부터 받은 값이고, 함수 f는 사용자 C의 비밀 정보이다. 일회성 티켓 사용자의 경우 인증자의 Timestamp 필드에 $f(g(x,y))$을 포함시켜 제삼자에 의한 인증자의 재사용을 방지한다. 그리고 Kerberos V5에서 Timestamp를 포함하는 인증자의 경우와 마찬가지로 서버는 인증자를 일정 기간 동안 보관하여 그 기간동안 동일한 사용자가 같은 Nonce를 갖는 인증자를 전송할 경우 메시지를 거부하는 기능을 가지도록 한다. 메시지 3을 받은 TGS는 자신의 비밀키로 $\mathrm{Ticket_{TGS}}$를 복호화하여 세션키 $\mathrm{K_{C,TGS}}$를 얻고 Kerberos V5와 같은 방식으로 티켓의 유효성을 검증한다. 그리고 메시지 2′의 정보 $f(x)$,g와 메시지 3의 정보 $f(y)$를 이용하여 $g(f(y),f(x))$를 계산한다. 이때 메시지 3의 인증자에서 얻은 $f(g(x,y))$과 TGS가 계산한 $g(f(y),f(x))$ 이 다르면 TGS는 C의 서비스 티켓을 발급 요청을 거부한다. TGS는 서비스 티켓과 C가 메시지 3의 Nonce 필드에서 보내온 $f(y)$를 반환하여 TGS의 응답이 현재 시간임을 나타낸다. $f(y)$는 Kerberos V5에서 Nonce2와 같은 역할을 한다.

(3) 클라이언트-서버간의 교환

<메시지 5>

$\mathrm{C} \longrightarrow \mathrm{V} : \mathrm{Options}||\mathrm{Ticket_V}||\mathrm{Nonce1}||\mathrm{E_{K_{C,V}}}[\mathrm{Subkey}||\mathrm{Seq\#}]$

$\mathrm{Ticket_V} = \mathrm{E_{K_V}}[\mathrm{Flag}||\mathrm{K_{C,V}}||\mathrm{Realm_C}||\mathrm{ID_C}||\mathrm{Times2}||\mathrm{AD_C}]$

<메시지 6>

$\mathrm{V} \longrightarrow \mathrm{C} : \mathrm{Nonce2},\ \mathrm{E_{K_{C,V}}}[\mathrm{Nonce1}||\mathrm{Subkey}||\mathrm{Seq\#}]$

<메시지 7>

$\mathrm{C} \longrightarrow \mathrm{V} : \mathrm{Authenticator_c}$

$\mathrm{Authenticator_c} = \mathrm{E_{K_{C,V}}}[\mathrm{ID_C}||\mathrm{Realm_c}||\mathrm{Nonce2}]$

메시지 5에서 티켓은 서버에게 세션키를 분배하는 역할을 하고 원문 필드의 Nonce1은 서버에게 상호 인증을 유도한다. 메시지 5에서 Subkey와 Seq#는 Kerberos V5와 마찬가지로 선택 사항이다. 사용자 인증은 메세지 7의 인증자 전송에서 이루어진다. 메시지 6에서 서버는 세션키로 암호화된 Nonce1으로 응답함으로써 서버 인증을 받고, 만약 메시지 5에서 선택 사항 Subkey, Seq#의 요청이 있으면 Subkey와Seq#를 추가적으로 보내 분배된 Subkey와Seq#의 사용을 사용

자로부터 확인 받는다. 그리고 Nonce2로 사용자의 인증에 쓰일 임의의 수를 전송한다. 마지막으로 메시지 7에서 사용자는 서버가 전송한 Nonce2를 인증자에 포함시켜 사용자 인증을 받는다.

일방향 인증(사용자 인증)만이 요구될 경우를 보자. 이 경우에는 Clock Adrift 방식의 Kerberos V5와 동일하다 [2]. 일방향 인증에서는 한 번의 메시지 전송만이 요구되는데, 사용자는 <메세지 5>에서 Kerberos V5 메시지 형식을 그대로 따르고, 인증자 Anthenticator_C의 Timestamp 필드(TS)에 메세지 4에서 TGS로부터 받은 티켓 유효 기간의 첫 번째 시간인 티켓의 발행 시간(T_{start})을 포함시킨다. 메시지 5를 수신한 서버는 자신의 시간과 인증자의 Timestamp를 비교하여 인증자의 유효성을 판단한다. 이것은 티켓 분배 센터(KDC)와 모든 서버간의 시계가 동기화되었다는 가정이 있기 때문에 가능하다.

5 결론

본 연구에서는 현재 분산 환경에서의 인증 시스템으로 널리 쓰이고 있는 Kerberos 인증 메카니즘을 조사하고 일회성 티켓을 원하는 사용자를 위해 사용자에게 서버와의 동기화된 시계를 요구하지 않고 안전한 인증을 제공하는 Kerveros-T 방식을 제안하였다.

본 연구에서 제안된 Kerveros-T 방식은 일회성 티켓을 원하는 사용자와 서버의 동기화된 시계를 제거하면서, Kerveros V5의 인증 메카니즘의 구조와 인증 프로토콜을 이용하여 ,시도·응답 방식을 채용했는데, 이때 나타날 수 있는 고정적 비밀 정보를 만들지 않았다는 장점을 지닌다.

Kerberos V5의 인증 프로토콜의 구조에 변화를 거의 영향을 주지 않았기때문에 Kerberos-T 방식은 서버와의 동기화된 시계를 갖기 어려운 사용자가 요구할 수 있는 Option으로 구현하기 적합하다.

참조 서적

[1] Charles Cavaiani, Jim Alves-Foss, **A Mutual Authenticating Protocol with key Distribution in a Cline/Server Enviroment** http://www.acm.org/crossroads/xrds2-4/authen.html

[2] Don Davis, Daniel Geer, and Theodore Ts'o **Kerberos With Clocks Adrift: History, Protocols, and Implementation** USENIX Computing Systems 9:1 (Jan. '96). http://world.std.com/ dtd

[3] L. Gong, **Variations on the Themes of Message Freshness and Replay** In Proceedings of the IEEE Computer Security Foundations Workshop VI, Franconia, New Hampshire, June,1993, pp.131-136 http://www.csl.sri.com/ gong/papers/pubs93.html

[4] Steven Bellovin, Michael Merritt, **Limitation of the Kerberos Authentication System** In the Proceeding fo the USENIX Winter'91 Conference, Dallas Tx, 1991, PP.253-276

[5] **The Kerberos Network Authentication Service V5** RFC1510, September 1993

[6] William Stallings, **NETWORK AND INTERNETWORK SECURITY** IEEE PRESS, 1995

한국과학기술원 수학과, 대전 유성구 구성동 373-1, 305-701
e-mail : start@math.kaist.ac.kr
한국과학기술원 수학과, 대전 유성구 구성동 373-1, 307-701
e-mail : sghanhn@math.kaist.ac.kr