情 報 保 護 學 會 誌
第 11 券 第 4 號, 2001. 8

단일 인증 시스템의 인증 기법과 인증 모델 분석

손 태 식*, 이 상 하**, 유 승 화***, 김 동 규***

요 약

인터넷 사용의 증가와 함께 점점 확대되는 인트라넷 및 익스트라넷 환경에서 사용자 인증 과정은 필수적인 요소이다. 이런 환경에서의 인증 과정에 있어 현재 사용자 편의와 보안을 위한 단일 인증 시스템이 널리 사용되고 있다. 본 논문에서는 단일 인증 시스템에서 사용되는 인증 기법을 사용자의 신원을 증명할 수 있는 매체(증거물:witness)에 따라 사용자가 아는 정보(something known by user), 사용자가 소유한 정보(something possessed) 그리고 사용자가 지니고 있는 정보(something embodied)로 분류하고 또한 인증 기법의 보안 강도에 따라 약한 인증(weak authentication) 기법과 강한 인증(strong authentication) 기법으로 나누어 분석한다. 인증 모델의 분석은 우선 단일 인증 시스템과 유사한 몇 개의 인증 모델을 먼저 분석하고 그 다음에 인증 시스템의 적용 환경, 인증 시스템의 인증 위치 그리고 인증 모델이 포함되는 시스템 계층에 따라 인증 모델을 분류한 후 각 인증 모델의 특징에 대하여 분석한다.

Ⅰ. 서 론

최근 몇 년 사이에 컴퓨터 환경은 급속도로 변화하고 있다. 사용자들은 분산 네트워크 환경에서 서로 상이한 플랫폼을 기반으로 하는 여러 시스템의 서비스를 이용하며 각각의 서비스를 제공받기 위하여 시스템마다 기본적으로 ID(Identification)와 PW(Password)를 통한 식별(Identification)과정, 인증(Authentication) 과정 그리고 권한 부여(Authorization) 과정을 거친다. 사용자의 관점에서 여러 서비스마다 인증 과정에 필요한 각각의 ID와 PW를 기억하는 것은 서비스 사용의 불편함을 초래하는 가장 큰 원인이다. 또한 시스템 관리자 측면에서도 제공하는 서비스 사용자 개개인에 대하여 개별적인 ID와 PW를 관리하는 것은 시스템 관리의 비효율과 데이터 관리의 어려움을 발생시킨다.

결국 사용자는 여러 ID와 PW의 기억 및 관리의 불편함을 극복하기 위해 기억하기 쉽게 짧게 만들거나, 추측이 쉽게 간단히 만들거나, 보기 쉬운 곳에 적어 두거나, 간단한 로그인 링크를 만들어 둘 수도 있고 아니면 사용하는 모든 시스템에 대해 동일한 ID와 PW를 사용하게 되어 보안 측면에 있어 매우 큰 위협 요소로 작용하게 된다. 또한 관리자는 서비스 질 향상보다 사용자 정보 관리를 위한 관리 시간과 설비 투자비용 증가를 위해 더 많은 노력을 기울이게 된다.[1]

따라서 사용자 측면에서 이종의 여러 시스템들로부터 제공받는 서비스에 대한 손쉬운 사용과 관리자 측면에서 사용자 관리 효율성 증대 그리고 사용자와 관리자 양 측면에 있어서 고려해야 되는 보안 문제를 해결하기 위한 방법으로 사용자 인증 과정에서 다양한 보안 강도를 가지는 인증 기법과 서비스 환경, 인증 특징, 적용 계층 등에 따른 여러 인증 모델들이 대두되고 있다.

본 논문에서는 단일 시스템의 개요 및 특징을 알아보고 단일 시스템에서 사용되는 다양한 인증 기법 및 인증 모델에 대하여 분석한다.

Ⅱ. 단일 인증 시스템

단일 인증 시스템은 각 시스템마다 고유의 ID와 PW를 기억하는데 불편함을 느낀 사용자들이 인증 과정이 필요한 모든 시스템에 동일한 ID와 PW를 사용하려는 의도에서 시작되었다. 즉 단일 인증 시

* 아주대학교 정보통신공학과 정보통신 및 시큐리티 연구실 (tsshon@madang.ajou.ac.kr)
** 동서울대학교 정보통신공학과 교수 (shyi@haksan.dsc.ac.kr)
*** 아주대학교 정보통신공학과 교수 (swyoo@madang.ajou.ac.kr, dkkim@madang.ajou.ac.kr)

스템은 모든 시스템에서 ID/PW와 같이 기존에 유지하고 있던 각 사용자 정보를 하나로 통합하여 사용자 인증 과정에서 반복되던 비효율적인 시스템 각각에서의 인증 과정을 별도의 인증 서버에서 한 번의 단일 과정으로 가능하게 하는 것이다. 그러므로 어떤 한 시스템에서 인증 과정을 거친 사용자의 신원 증명은 다른 서비스를 제공하는 시스템에서 부가적인 인증 과정 없이 자동적으로 가능하게 된다. 또한 기존의 권한 관리는 각 시스템 인증 과정에서 시스템 고유 권한 정보 데이터베이스의 정보에 기인하여 개별적인 권한을 부여하였다. 그러나 단일 인증 시스템에서는 이러한 부가적인 작업에서 벗어나 접근 권한 관리 문제도 사용자에 대한 권한 부여 정보와 서비스 자원에 대한 접근 제어 정보를 중앙에서 관리하여 시스템 관리자들은 단일 인증 과정에 포함된 권한 부여 과정을 통해 접근 권한을 관리한다.

단일 인증 시스템을 사용함으로써 얻을 수 있는 특징은 기존의 반복적인 사용자 로그인 과정이 필요하던 인증 과정이 사용자 측면에서 볼 때 한번의 인증 과정을 통해 모든 서비스를 제공받을 수 있고, 다양한 인증 기법의 사용으로 사용자 개인 정보 유출 등의 보안 위협에서 안전성이 강화되었다. 또한 관리자 관점에서는 사용자 정보의 통합적인 관리로 인하여 사용자들에 대한 사용자 계정 생성, 추가, 삭제 등의 사용자 정보 관리가 용이하며, 보안 측면에서도 중앙에서 사용자 접근 권한을 제어함으로써 높은 보안성을 가져 올 수 있다. 그 외에도 사용자 계정 관리에 걸리는 시간의 감소, 인증 과정 단일화로 인한 시스템 부하 감소 등을 들 수 있다.

이러한 단일 인증 시스템의 특징 외에도 단일 인증 과정에 있어 고려해야 할 문제로는 상이하고 이질적인 플랫폼 위에서 운용되는 시스템의 서비스들에 대하여 적합한 인증 모델을 선택하는 문제, 보안성을 높이기 위한 알맞은 인증 기법을 선택하는 문제, 사용자에게 각 시스템 또는 서비스마다 알맞은 접근 권한을 부여하는 문제 등이 있다.[1][2][3][4]

Ⅲ. 단일 인증 시스템의 인증 기법의 분석

일반적으로 서비스를 제공하는 시스템에서 사용자를 인증하는 과정은 사용자 식별(User Identification), 사용자 인증(User Authentication), 사용자 권한 부여(User Authorization)의 세 과정을 거친다. 사용자 식별 단계에서는 사용자 고유의 식별자를 통해 시스템의 서비스를 제공받는 사용자들 사이에서 유일성을 확인하는 단계이며, 사용자 인증 단계는 사용자의 신원 확인 후 서비스를 제공받을 있는 정당한 사용자임을 증명하는 단계이며 마지막으로 사용자 권한 부여 단계는 인증 된 사용자가 접근 할 수 있는 응용 서버의 자원에 대한 접근 권한을 결정하는 단계이다.

단일 인증 시스템에서 사용하는 인증 기법은 사용자 인증 과정 중에서 사용자의 신원을 증명 할 수 있는 매체(증거물 : witness)에 따라 분류 될 수 있으며, 또한 사용자 인증 과정의 보안 강도에 차이를 두고 분류 될 수 있다.

사용자 신원을 증명 할 수 있는 매체를 분류하는 방법은 사용자가 알고 있는 정보(something known by the user)에 의한 인증, 사용자가 소유하고 있는 정보(something possessed)에 의한 인증, 사용자가 지니고 있는 정보(something embodied)에 의한 인증으로 구분 할 수 있다.

사용자 인증 과정에서 보안 강도에 의한 인증 기법 분류는 약한 인증(weak authentication) 기법과 강한 인증(strong authentication) 기법 두 가지로 구분된다. 또한 위의 분류 기준과 상관없이 인증 기법들간에 필요로 하는 환경과 보안 강도에 따라 혼합되어 사용 될 수 있다 아래의 표 1은 앞서 구분한 증명 매체와 보안강도를 기준으로한 분류 방법에 따라 다양한 인증 기법들을 지니고 있는 특성을 정리한 것이다.[4][5][10]

〔표 1〕 인증 기법 분류

증명매체 / 보안강도	사용자가 아는 정보	사용자가 소유한 정보	사용자가 지닌 정보
약한 인증	Password trusted channel	없음	없음
강한 인증	One-Time Password	Public-key, H/W token, SmartCard	Biometric (Retini, Palm print etc..)

1. 패스워드(PW, Password) 기법

PW기법은 인증 기법의 분류 중에서 증명 매체에 의한 분류 중 사용자가 아는 정보이며 인증 보안 강도에 의한 분류 중 약한 인증 기법에 속한다. PW기법은 대부분의 시스템에서 가장 기본적인 사용자 인

증 기법으로서 사용자가 이미 알고 있는 PW를 서비스 제공 시스템에 전송하여 시스템이 가지고 있는 사용자의 PW와 비교하여 정당한 사용자임을 확인하는 인증 기법이다. 기존에 일반적으로 사용되던 PW기법은 PW를 암호화하지 않고 그냥 평문 상태로 네트워크 통신 경로에 전송하여 보안에 많은 취약점을 가지고 있으며 또한 반복적으로 같은 PW를 사용하는데 따른 보안 문제가 있다. 최근에는 SSL (Secure Socket Layer)등과 같은 암호화 기법과 일회용 패스워드 기법(OTP, One Time Password) 등을 함께 사용하여 모든 시스템에 기본적으로 사용되는 PW 인증 기법의 보안성을 강화하고 있다.

2. 신뢰 채널(Trusted Channel) 기법

신뢰 채널 기법은 인증 기법의 분류 중에서 PW 기법과 같은 분류에 속한다. 신뢰 채널은 보통 유닉스 계통 시스템의 rlogin, rsh 등의 명령에서 사용되던 인증 기법으로서 사용자는 서비스 접속 이전에 서비스를 제공받을 시스템과의 신뢰 관계를 구축한 후, 실제 서비스 접속 시에 미리 가지고 있는 신뢰 정보를 통해 부가적인 사용자 인증 없이 서비스를 제공받을 수 있다. 하지만, 서비스 접속 시 전달되는 사용자 신뢰 정보의 안전한 전송이 보장되지 않으므로 신뢰 정보가 유출 될 수 있다는 문제가 있다. 현재 신뢰 채널을 사용하는 rlogin, rsh과 같은 명령어는 시스템 취약점의 한 부분으로 많이 알려져 그 사용 빈도가 적으며 따라서 신뢰 채널 기법도 사용자와 서비스 시스템간에 신뢰 관계가 확실히 맺어져 있는 환경 외에는 사용하지 않는다.

3. 일회용 패스워드(One-Time Password)기법

OTP(One Time Password)기법은 인증 기법의 분류 중에서 증명 매체에 의한 분류 중 사용자가 아는 정보이며 인증 보안 강도에 의한 분류 중에서 강한 인증 기법에 속한다. OTP는 기존의 PW기법이 같은 PW를 반복하여 사용함으로서 발생할 수 있는 보안상의 문제를 해결한다.

OTP 기법에서 많이 사용하는 Challenge-Response방식은 사용자의 토큰 카드에 고유 PW와 시스템에서 가지고 있는 사용자의 PW를 일치시켜 이 값을 키로 랜덤하게 숫자를 만들어 시스템에서 Challenge하고 사용자가 Response하여 인증하는 방식이다. 여기에서 사용되는 암호 기법이 DES(Data Encryption Standard)이며, 이를 DES Challenge-Response방식이라고도 부른다.

인증 절차는 사용자가 접속하여 사용자 ID를 입력하면 시스템의 키 데이터 베이스에서 사용자의 PW를 읽어 랜덤 숫자를 생성하고 사용자의 PW로 암호화하여 보관한다. 이 때 생성된 랜덤 숫자를 사용자에게 Challenge로 보내게 된다. 사용자는 Challenge된 숫자를 자신의 PW로 암호화하여 시스템에 Response 값으로 보낸다. 시스템에서는 Response된 값과 최초 만들어 저장한 값을 비교하여 같으면 정당한 사용자로 인증 하게 된다. 이러한 Challenge-Response 방식의 OTP는 S/W와 H/W 카드로 운영 될 수 있으며 카드와 PW를 임의로 변경하는 것이 가능하고 구현이 비교적 쉽다는 장점이 있다. 하지만 사용자의 인증 절차가 복잡하기 때문에 시간이 좀 많이 걸리는 단점이 있다.

4. 공개키 사용 기법(인증서 사용)

공개키 사용 기법은 인증 기법의 분류에서 증명 매체에 의한 분류 중 사용자가 소유한 정보이며, 인증 보안 강도에 의한 분류 중 강한 인증 기법에 속한다. 공개키 사용 기법은 신뢰 기관(Certificate Authority)에서 사용자가 인증 과정 전에 사용자 자신의 인증서와 개인키를 발급 받은 후, 인증 과정에서 자신의 인증서와 자신의 개인키로 전자 서명된 값을 같이 보냄으로써 정당한 사용자임을 인증 받는 방법이다. 공개키 사용 방식은 공개키 암호화를 이용한 challenge-response 방식과 전자 서명을 기반으로 하는 challenge-response 방식이 있다.

공개키 암호화를 이용한 challenge-response 방식의 한 예로 공개키 복호화와 증거물(witness)을 이용한 방법을 들 수 있는데, 이 방법은 해쉬 함수를 사용함으로써 chosen-text 공격을 막을 수 있다.

〔표 2〕 전자 서명을 기반으로 한 Challenge-response 방식 인증 과정 기호 설명

기호	설명
P_A	공개키 알고리즘
h	단방향 해쉬 함수
$cert_A$, $cert_B$	A, B의 인증서
S_A, S_B	A, B의 서명
t_x	X의 타임 스탬프
r_x	X의 랜덤 값
D_a	A의 데이터 블록 $D_a = (t_a, r_a, B)$
D_b	B의 데이터 블록 $D_b = (t_b, r_b, A, r_a)$

1) 인증 과정

A <- B : h(r), B, P_A(r, B)
A -> B : r

(1) B는 증거물(witness) 'r'을 해쉬 함수를 이용해 생성한 h(r)과 h(r)을 생성하는데 사용한 랜덤 값 'r' 그리고 자신의 신원 정보와 랜덤 값 'r'을 자신의 개인키로 암호화하여 A에게 인증을 요청한다.
(2) A는 B로부터 받은 랜덤 값 'r'을 사용하여 해쉬 값을 생성하고 B가 생성하여 보내온 해쉬 값(h(r))과 같은지 검사해서 같지 않으면 끝내고, 같으면 'r'을 전송해 B와의 인증 과정이 성립되었음을 알린다.

전자 서명을 기반으로 한 Challenge-response 방식은 아래와 같이 X.509 strong two-way protocol, X.509 strong three-way protocol 그리고 ISO/IEC 9798-3 mechanism 등이 있다.

1) X.509 strong two-way protocol 인증 과정

A -> B : $cert_A$, D_A, $S_A(D_A)$
A <- B : $cert_B$, D_B, $S_B(D_B)$

(1) A는 B에게 자신의 인증서와 타임 스탬프, 랜덤 값, B의 ID가 담긴 데이터와 사인 값을 보낸다.
(2) B는 A로부터 받은 인증서와 서명 값을 확인하여 A를 인증하고 A에게 자신의 인증서와 A가 자신에게 보낸 랜덤 값이 포함된 데이터를 사인하여 보낸다.
(3) A는 마찬가지로 인증서와 서명 값을 확인하여 B를 인증 한다.

2) X.509 strong three-way protocol 인증 과정

A -> B : $cert_A$, D_A, $S_A(D_A)$
A <- B : $cert_B$, D_B, $S_B(D_B)$
A -> B : (r_b, B), $S_A(r_b, B)$

(1) X.509 strong three-way protocol은 two-way protocol에 비하여 A가 B의 인증서와 사인 값을 받은 후 자신이 먼저 보냈던 r_a를 확인한 후에 다시 r_b를 B에게 보내어 확인 받는 과정이 추가되었다.

3) ISO/IEC 9798-3 mechanism 인증 과정

(1) 일방적인 인증 (time stamp 사용)
A -> B : $cert_A$, t_A, B, $S_A(t_A, B)$

(2) 일방적인 인증 (random number 사용)
A <- B : rB
A -> B : $cert_A$, r_A, B, $S_A(r_A, r_B, B)$

(3) 상호 인증 (random number 사용)
A <- B : r_B
A -> B : $cert_A$, r_A, B, $S_A(r_A, r_B, B)$
A <- B : $cert_B$, A, $S_B(r_B, r_A, A)$

5. 하드웨어 토큰(H/W token) 기법

하드웨어 토큰 기법은 인증 기법의 분류 중에서 공개키 사용 기법과 같은 분류에 속한다. 하드웨어 토큰 기법은 30-90초마다 계속 난수를 만드는 알고리즘이 탑재된 서버와 토큰 카드가 필요하며, 같은 시간과 간격으로 난수를 만들기 위해서는 만드는 시간이 일치해야 하기 때문에 시간 동기화 방식을 보통 사용한다. 시간 동기화 방식의 인증 절차는 로그인을 하기 위해 접속을 시도하면 시스템은 6자리의 랜덤 숫자와 PIN(Personal Identification Number) 코드를 요구한다. 사용자는 토큰 카드의 6자리 숫자와 자신의 PIN 코드를 시스템에 입력한다. 그러면 시스템에서는 PIN 코드를 키로 사용자의 정보가 저장된 DB를 검색하여 6자리의 숫자를 비교한다. 비교 결과가 서로 같으면 사용자의 권한이 인증 되어 로그 인이 가능케 된다. 그리고 같은 시간대에 같은 난수를 사용하는 한 명 이상의 사용자는 시스템에 접속을 못하게 한다. 만들어지는 난수가 일치하기 위해서는 서버와 토큰 카드 사이에 같은 시간이 보장되어야 한다. 만약 시간이 0.1초씩 틀리게 되면 얼마의 시간이 지난 후에 서버와의 동기는 보장될 수 없다는 단점을 가지고 있다.

6. 스마트카드(Smart Card) 기법

스마트 카드 기법은 인증 기법의 분류 중에서 하

드웨어 토큰 기법과 마찬가지로 공개키 사용 기법과 같은 분류에 속한다. 단일 인증 시스템에서 사용되는 스마트 카드 기법에서 스마트 카드에 저장되는 정보는 사용자의 개인키와 인증서이다. 사용자의 개인키는 사용자만이 아는 패스워드로 스마트 카드에 암호화되어 저장되므로 사용자 외의 다른 사용자는 스마트 카드에 저장되어 있는 개인키를 얻을 수 없다.[8]

스마트 카드에 저장되는 인증서는 보통 특정 API나 카드 인터페이스 프로토콜과 함께 사용되어 진다. 스마트 카드에 저장되어 있는 인증서는 보통 X.509 표준을 따르며 인증서를 액세스하기 위한 표준 인터페이스로는 RSA Security의 PKCS#11 (Cryptographic Token Interface Standard number 11)[6]과 MS(MicroSoft)의 PC/SC (Personal Computer/Smart Card)등이 있다. 스마트 카드에 사용되는 인증 방식으로는 DES를 사용하며 NIST(National Institute of Standards Technology)에서 제안된 ASACS(Advanced Secure Access Control Systems) 등을 들 수 있는데 인증 방식은 다음과 같다.[21]

[표 3] 스마트 카드 인증 과정에 사용되는 기호 설명

기호	설명
Card	스마트카드
Reader	스마트카드 리더
K	스마트카드와 스마트카드 리더 간의 비밀키
PIN	스마트카드 사용자를 식별하기 위한 개인 식별 번호
r1, r2	스마트카드와 리던간에 생성하는 랜덤 값

1) 스마트카드의 사용자 인증 과정

User -> Reader : PIN
Card <- Reader : Ek[PIN]
Card : if (Dk[PIN] == PIN) then
Authentication

(1) 사용자는 스마트 카드 리더에 스마트카드를 삽입한다. 스마트 카드의 사용자를 확인하기 위해서 스마트 카드 리더에 사용자만이 아는 PIN을 입력한다.
(2) 스마트 카드 리더는 입력된 PIN을 자신과 스마트 카드가 공유하는 비밀키로 암호화하여 스마트 카드에 보낸다.
(3) 스마트 카드는 공통의 비밀키로 암호화된 PIN을 복호화 한 후 자신이 가지고 있는 PIN과 비교하여 사용자를 인증 한다.

2) 스마트 카드와 스마트카드 리더간의 인증 과정

Card -> Reader : r1
Card <- Reader : Ek[r1] | r2
Card : if (Dk[r1] == r1) ok!
Card -> Reader : E_k[r2]
Reader : if (D_k[r2] == r2) then
Authentication

(1) 스마트 카드는 임의의 난수를 생성하여 이를 스마트 카드 리더에 보낸다.
(2) 스마트 카드 리더는 스마트 카드로부터 받은 난수를 비밀키로 암호화한 값과 새로 생성한 난수 값을 함께 스마트 카드 리더에 보낸다.
(3) 스마트 카드는 스마트 카드 리더로부터 받은 암호화 값을 받아 복호화 한 후 자신이 가지고 있는 난수와 비교하여 같으면 새로 받은 난수를 암호화하여 스마트 카드 리더에 보낸다.
(4) 스마트 카드 리더는 스마트 카드에서 다시 보낸 암호화 값을 복호화 한 후 자신의 난수와 비교하여 인증 과정을 마친다.

7. 생체측정(Biometrics) 인증 기법

생체 인증 기법은 인증 기법의 분류 중에서 증명 매체에 의한 분류 중 사용자가 소유한 정보이며 인증 보안 강도에 의한 분류 중에서 강한 인증 기법에 속한다. 생체측정(Biometrics)은 생리학적 또는 행동상의 특징을 기반으로 신원을 자동으로 인식하는 것을 말한다.

생체측정 인증 기법에서 인증 되는 사람은 인식 시점에 실제로 존재해야 하며, 패스워드를 기억하거나 토큰을 가지고 다녀야 하는 필요성이 없기 때문에 현재 가장 보안성이 높은 인증 기법의 하나로 연구되고 있다. 현재 지문, 홍채, 망막 스캔, 음성, 얼굴 모양, 손 모양 등 다양한 형태의 생체측정 시스템이 실시간 인식에 이용되고 있는데, 가장 대중적인 것은 얼굴과 지문을 이용하는 생체측정 시스템이다.[20]

8. 인증 기법의 혼합 사용

인증 기법의 혼합 사용은 현재 대부분의 시스템에서 기본적으로 사용되는 패스워드 기반 인증 방식과 인증 서비스의 특성에 따라 보안 강도를 더하기 위하여 스마트카드, 생체인증 등 여러 인증 기법을 함께 사용하는 것이다. 그 예로는 패스워드와 스마트카드에 저장되어 있는 인증서를 같이 사용하는 방법 등이 있다. 이 경우 기본적으로 패스워드 기반의 인증이 가능하며 스마트 카드에 저장된 인증서를 통한 강한 인증 기법 또한 사용 할 수 있다. 이때 인증서는 스마트 카드에 저장됨으로써 물리적 보안이 가능하며 스마트 카드에 저장된 인증서 및 개인키 정보를 얻어 오기 위해선 Challenge-Response 방식을 사용하여 자신만의 고유한 비밀키를 사용하므로 스마트 카드를 가지고 있을지라도 사용이 불가능하게 된다. 이렇게 가능한 인증 기법의 혼합 방식은 토큰과 패스워드 방식의 조합, 토큰과 생체측정 인증 기법의 조합 그리고 패스워드(PIN), 공개키(인증서), 토큰의 조합 등 여러 가지 방법이 있을 수 있다.

Ⅳ. 단일 인증 시스템의 인증 모델 분석

단일 인증 시스템에 적용되는 인증 모델은 표 4.와 같이 기본적으로 적용 환경, 인증 위치 그리고 적용 시스템 계층을 기준으로 나눌 수 있으며 또한 기존의 단일 인증 모델과 유사한 기능을 가지는 인증 모델로서 패스워드 동기화 인증 모델과 로그인 자동화 인증 모델이 있다.

적용 환경에 따라서는 현재 급속히 퍼져가고 있는 인터넷 웹 기반 단일 인증 모델과 기존의 클라이언트/서버 환경에서 동작하는 단일 인증 모델로 나눌 수 있다. 또한 웹 기반 단일 인증 시스템은 시스템의 적용시 사용자 정보 관리의 관점에서 HTTP-redirection 등의 기술을 이용한 중앙 집중 모델과 일반 모델로 다시 분류 할 수 있다. 웹 기반의 중앙 집중 인증 모델은 사용자에 대한 인증 및 관련 과정이 모두 중앙의 인증 서버에서 이루어지며 웹 기반의 일반 모델에서는 각 웹서버에 각각의 사용자 정보를 바탕으로 인증 및 관련 기능을 수행하는 모듈이 부가된다.

사용자를 인증하는 위치에 따라서는 사용자들과 응용 서버들 사이에서 인증을 중개하는 서버를 가지고 있는 브로커 기반 단일 인증 모델, 단일 인증 기능을 대신해서 수행해주는 에이전트 기반 단일 인증 모델, 네트워크의 구성요소 중 게이트웨이에 장착되어 특정 네트워크로 접근하려는 사용자들을 관리해주는 게이트웨이 기반 단일 인증 모델 그리고 위의 시스템들을 한데 혼합하여 사용하는 혼합 단일 인증 모델이 있다. 특히 에이전트 기반 단일 인증 모델은 에이전트의 위치에 따라 사용자 측 에이전트 기반 단일 인증 모델과 서버 측 에이전트 기반 단일 인증 모델로 나뉜다.

단일 인증 모델이 적용되는 시스템 계층에 따른 분류로는 시스템에 사용되는 운영체제 포함되어 단일 인증 기능을 제공하는 운영체제 포함 단일 인증 모델과, 특정 애플리케이션들을 통합하여 애플리케이션 사이에서 단일 인증 기능을 수행해주는 애플리케이션 통합 단일 인증 모델 그리고 애플리케이션 계층 위에서 새로운 단일 인증 에이전트 계층을 생성해 시스템이나 환경에 독립적인 기능을 수행하는 단일 인증 에이전트 계층 모델이 있다.

본 논문에서는 위와 같이 단일 인증 모델을 분류하여 각 인증 모델이 가지고 있는 특징과 기능 그리고 인증 모델의 알맞은 적용 환경, 대상 등 여러 가지 측면을 분석한다.[2][3][4][5][17][18][19][20][22]

〔표 4〕 단일 인증 모델의 분류

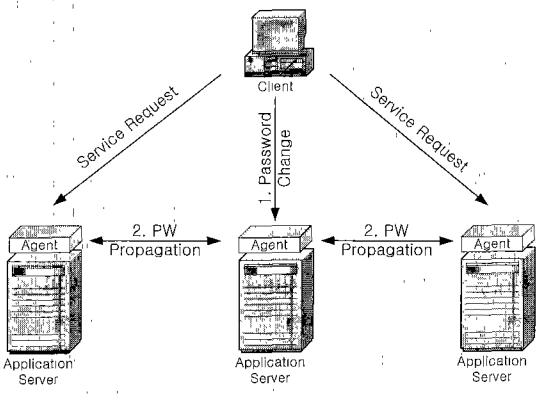
분류기준	Authentication Model		etc,.
유사 SSO	Password Synchronization		
	Login Automation		
적용 환경	Web-based SSO	Centralization Model	HTTP redirection
		Normal Model	Added Auth-Module
	Client/Server-based SSO		
인증 위치	Broker-based SSO		
	Agent-based SSO	Client-side SSO	
		Server-side SSO	
	Gateway-based SSO		
	Mixing SSO		Agent와 Broker 기능 포함
시스템 계층	Agent-layer SSO		web및 C/S 환경 모두 적용 가능
	Application-Integration SSO		
	OS-Embedded SSO		Broker기반 SSO와 유사

1. 유사 단일 인증 모델

1) 패스워드 동기화 모델

패스워드 동기화 모델은 단일 인증 모델에 비하여 낮은 비용과 부가적인 관리 없이 사용자들이 하나의 패스워드를 사용하여 여러 응용 서비스를 제공받게 해준다. 하지만 단일 인증 모델과는 다르게 사용자가 각 응용 서비스를 제공받기 위해서 매 번 동일한 패스워드를 입력해야 되는 불편함이 있다.

패스워드 동기화 모델은 그림 1과 같이 각 응용 서버에 패스워드 동기화 에이전트가 장착되어 사용들이 입력한 패스워드에 대해서 인증 과정을 수행하며 또한 사용자가 특정 응용 서버에서 패스워드를 변경하는 경우 그 응용 서버의 패스워드 동기화 에이전트가 다른 응용 서버의 패스워드 에이전트들에게 변경된 패스워드를 전송하여 사용자에게 투명하게 각 서버의 패스워드를 일치시켜 준다. 하지만 사용자가 특정 응용 서버에 사용하는 패스워드를 악의의 사용자에게 노출되는 경우 사용자가 서비스를 제공받는 도메인 내의 모든 응용 서버 보안에 있어 큰 위협을 발생시킬 수 있는 문제가 있다.

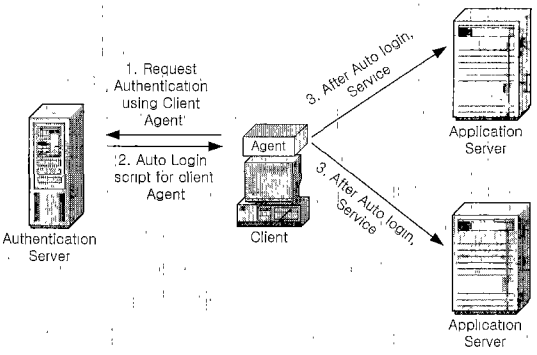


〔그림 1〕 Password Synchronization Model

2) 로그인 자동화(Login Automation) 모델

로그인 자동화 모델은 간단한 단일 인증 시스템의 일종이다. 로그인 자동화 모델에서는 인증 서버가 사용자 로그인 정보와 스크립트 등으로 이루어진 응용 서버 연결 정보를 가지고 있다.

사용자 측면에서 사용자는 자신이 서비스를 원하는 응용 서비스를 선택하면 그림 2와 같이 사용자의 에이전트 프로그램이 인증 서버에 인증을 요청하며, 인증 과정이 성공하면 사용자에게 사용자가 응용 서버에 자동으로 로그인 할 수 있는 로그인 스크립트가 실행되어 사용자는 응용 서버로부터 서비스를 제공받을 수 있게 된다. 서비스를 제공하는 응용 서버 측면에서는 사용자 인증, 사용자 관리 등의 문제에



〔그림 2〕 Login Automation Model

신경을 쓰지 않고도 기존 시스템의 변화 없이 사용자에게 응용 서비스를 제공할 수 있는 장점이 있다. 또한 로그인 자동화 인증 모델은 사용자의 인증 정보 및 관련 정보를 보안 DB를 통하여 관리하여 패스워드 동기화 인증 모델이 갖는 패스워드 누출의 문제와 같은 보안 위협에 대하여 좀 더 안전성을 가진다. 하지만 이 로그인 자동화 인증 모델은 인증 서버에서 사용자 인증 정보와 자동 접속 정보 등 많은 양의 사용자 정보 및 응용 서버 서비스 연결 정보를 유지해야 하므로 적용 환경이나 대상에 있어 소규모의 적은 사용자들에게 특화된 서비스를 제공하는 환경을 대상으로 한다.

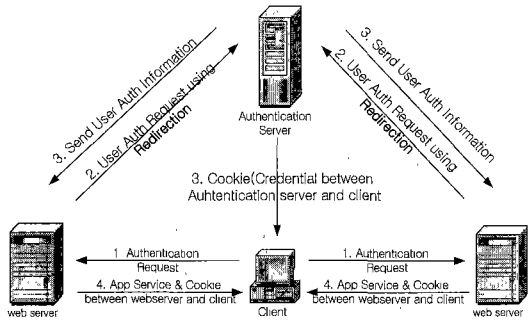
2. 적용 환경에 따른 분류

1) 웹 기반 단일 인증 모델

웹 기반의 단일 인증 모델에서 사용자는 부가적인 사용자 프로그램의 실행 없이 웹 브라우저를 사용하여 여러 웹서버에 한번의 로그 인 과정으로 여러 가지 웹 서비스를 받을 수 있는 단일 인증 모델이다. 웹 기반의 단일 인증 모델은 현재 인터넷 기반의 웹 서비스가 널리 보급되면서 그 사용이 계속 늘어나고 있다. 따라서 단일 인증 모델의 분류에 있어서도 웹 환경의 수요가 증가로 인한 환경적 측면 외에 다른 여러 분류의 인증 모델에서 고려되고 있다. 여기서

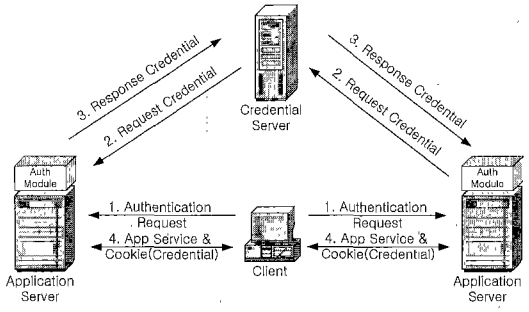
는 특히 구현 측면에서 웹 기반의 단일 인증 모델을 기존의 웹 서버 수정 없이 한 곳의 단일 인증 서버에서 모든 요청을 처리하는 중앙 집중형 웹 기반 단일 인증 모델과 기존의 웹서버에 인증 모듈과 같은 웹 서버 수정이 필요한 두 가지 모델로 구분한다.

다음의 그림 3은 중앙 집중형 웹 기반 단일 인증 모델로서 사용자가 응용 서버에 인증 및 서비스를 요청하면 응용 서버는 사용자의 인증 및 서비스 요청을 HTTP-redirection 등의 기술을 통해 중앙의 인증 서버로 보낸다. 결국 응용 서버는 사용자 인증 과정에서 있어 독립적이며 중앙의 인증 서버에서 인증을 받은 사용자에게 대해서만 응용 서비스를 제공해 주면 된다. 따라서 부가적인 사용자 정보 관리 등의 오버헤드를 줄일 수 있다. 다만 중앙의 인증 서버에서 사용자 인증 후 권한을 부여하는 단계와 부여한 권한에 대하여 권한 속성을 결정하는 등의 접근 제어 기능은 제공 서비스와 제공 환경 및 대상 등에 따라 인증 과정과 같이 중앙 집중적으로 관리하거나 아니면 권한 속성 관리에 있어 각 응용 서버에서 자신의 보안 정책에 맞게 분산하여 관리 할 수 있다. 중앙 집중형 웹 기반 단일 인증 모델은 중앙의 인증 서버에서 사용자 인증 과정이 성공하면 사용자 인증 정보(신임장)를 사용자에게 쿠키 형태로 넘겨주어 추후의 사용자 단일 인증에 이용한다. 또한 응용 서버에게도 응용 서버가 사용자 서비스에 이용 할 수 있는 사용자 정보를 넘겨준다. 이때 인증 서버와 사용자 사이에 공유하는 쿠키 외에 응용 서버 사용자 사이에 공유하는 로컬 쿠키가 생성되어 사용자 서비스에 이용된다.



〔그림 3〕 Centralized Web-based SSO Model

그림 4.는 각 사용자에게 서비스를 제공하는 응용 서버에 인증 모듈이 부가되는 분산형 웹 기반 단일 인증 모델로서 사용자에 대한 정보는 각 응용 서버들이 유지하며 인증이 성공한 사용자에게 대해서 별도로 운영되는 신임장 서버에게 같은 도메인내의 응용 서버로부터 인증 후에 부가적인 인증 없이 응용 서비스를 제공받을 수 있는 신임장을 발급 받아 사용한다. 이러한 인증 모델은 각 응용 서버에서 사용자들에 대한 정보를 관리하며 접근 제어와 같은 기능을 서로간에 자신의 보안 정책을 가지고 독립적으로 수행 가능한 특징을 지닌다.

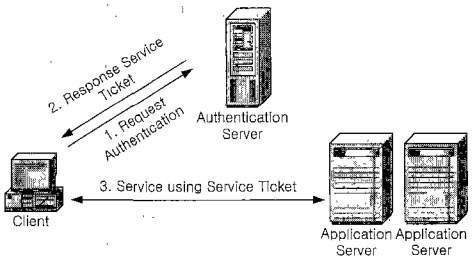


〔그림 4〕 Decentralized Web-based SSO Model

2) 클라이언트/서버 기반 단일 인증 모델

클라이언트/서버 기반 단일 인증 모델은 기존의 클라이언트/서버 환경에서 클라이언트가 여러 응용 서버에 서비스를 요청하는 경우 한 번의 인증 과정으로 여러 응용 서버의 응용 서비스를 제공받기 위한 일반적인 인증 모델이다. 이 인증 모델에서 클라이언트는 응용 서버에 접속하여 응용 서비스를 제공받기 위한 부가적인 프로그램을 실행시킨다.

아래의 그림 5와 같이 클라이언트는 자신의 인증 프로그램을 통해 인증 서버에 접속한 후 인증 서버에서 인증 과정을 거친 후에 자신의 인증 정보와 제공받을 수 있는 응용 서비스에 대한 정보 등을 서비스 티켓과 같은 형태로 발급 받고 응용 서버에 자신이 원하는 서비스를 제공받기 위해 인증 서버로부터 받은 서비스 티켓을 보냄으로써 자신이 정당한 사용자이고 서비스를 제공받을 수 있음을 증명한다. 이러한 클라이언트/서버 기반의 단일 인증 모델은 대부분의 인증 모델에 있어 기본이 된다.



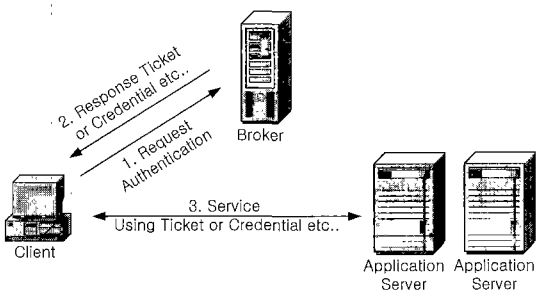
〔그림 5〕 Client/Server-based SSO Model

2. 인증 위치에 따른 분류

1) 브로커 기반 단일 인증 모델

브로커 기반 단일 인증 모델은 그림 6과 같이 클라이언트와 응용 서버간에 위치하며 클라이언트와 응용 서버들을 감시하는 서버(브로커)가 사용자 계정을 관리하고 사용자의 인증 요청을 처리한다. 이 인증 모델은 적용 환경에 따른 인증 모델의 분류 중에서 클라이언트/서버 기반의 인증 모델과 유사하다. 클라이언트/서버 기반의 인증 모델처럼 클라이언트와 서버를 중개해주는 브로커(인증 서버)가 존재하여 사용자 인증의 기능을 중앙에서 처리하며, 클라이언트와 응용 서버는 인증 과정을 위한 부가적인 프로그램 혹은 모듈을 설치해야 한다. 현재 상용화되어 있는 많은 제품들이 이러한 브로커 기반의 단일 인증 모델 모델을 따른다. 이러한 예로는 SESAME(Secure European System for Applications in a Multi-vendor Environment)[14]나 커버로스(Kerberos)[7]나 같은 인증 모델을 들 수 있다.

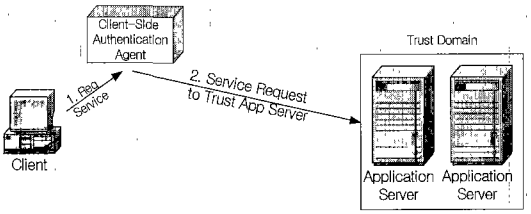
브로커 기반의 단일 인증 모델은 사용자의 인증 요청을 중앙의 중개 서버에서 이미 가지고 있는 사용자 정보를 기반으로 사용자가 원하는 서비스를 제공받을 수 있는 서비스 티켓이나 토큰 등을 발급한다. 즉 응용 서버는 중앙의 중개 서버가 발급한 서비스 티켓이나 토큰을 사용하여 사용자를 신뢰하게 된다. 이러한 브로커 기반의 단일 인증 모델은 사용자 인증 및 관리 기능이 한 곳의 서버에서 일어나기 때문에 서버 자체의 물리적 보호 및 정보 유출에 대한 보안 관리가 중요하다.



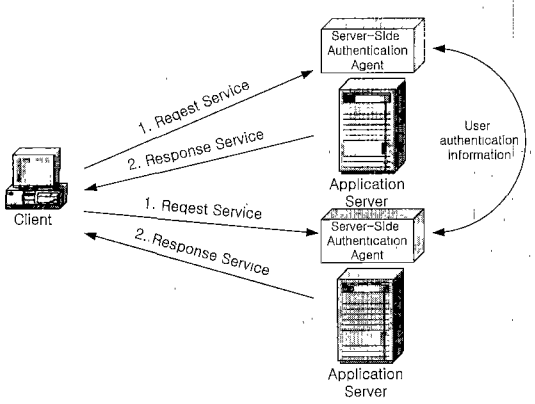
〔그림 6〕 Broker-based SSO Model

2) 에이전트 기반 단일 인증 모델

에이전트 기반 단일 인증 모델에서는 서로 다른 응용 서비스에 대해서 사용자들의 신원을 자동으로 확인하여 인증 과정을 대행하는 에이전트 프로그램이 존재한다. 에이전트는 클라이언트와 서버간의 인증 과정 대행을 위해 패스워드 리스트 또는 암호화 키 등을 사용할 수 있으며 이런 것들을 통하여 인증 과정에 필요한 클라이언트나 서버의 부가적인 노력을 줄일 수 있다. 인증 에이전트는 그림 7과 같이 서버 측에서 각 응용 서버마다 신뢰하는 사용자 정보를 공유하며 단일 인증 기능을 제공하는 서버 측 에이전트 기반 단일 인증 모델과 그림 8과 같이 클라이언트 측에서 클라이언트가 가지고 있는 신뢰 정보를 기반으로 클라이언트의 서비스 접속 요구를 대행하는 클라이언트 측 에이전트 기반 단일 인증 모델로 분류 할 수 있다. 에이전트 기반 단일 인증 모델의 예로 SSH(Secure SHell)[15]가 있다.



〔그림 7〕 Client-Side Agent-based SSO Model



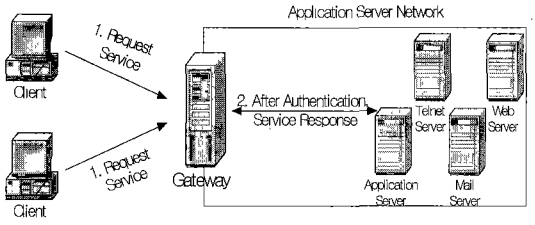
〔그림 8〕 Server-Side Agent-based SSO Model

3) 게이트웨이 기반 단일 인증 모델

네트워크에서 클라이언트와 응용 서버사이의 인증 및 서비스를 위해 감시 및 중개 역할을 하는 인증 서버 형태의 인증 모델을 제공하는 브로커 기반 단일 인증 모델에 비해 게이트웨이 기반 단일 인증 모델은 클라이언트가 접근을 원하는 응용 서버 측의 네트워크 진입을 위해 통과해야 할 문에 인증 서버를 두는 것이다. 이

문은 방화벽으로 사용될 수 도 있고 또한 암호화 및 클라이언트 인증을 위한 서버로 사용될 수 있다.

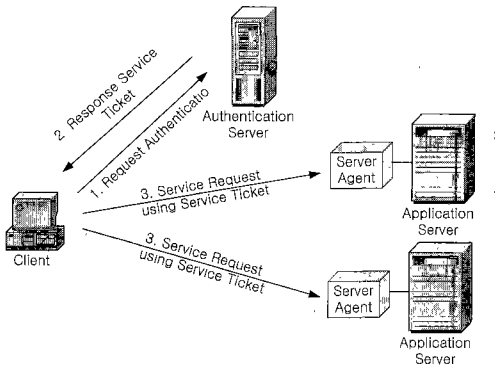
게이트웨이 기반 단일 인증 모델에서 모든 응용 서비스는 게이트웨이 뒤에 존재한다. 게이트웨이는 서비스 고유의 IP 주소를 통해서 각각의 서비스를 구별하고 사용자는 게이트웨이가 알고있는 자신의 고유 정보나 IP 주소 등을 통하여 자신이 받고자 하는 서비스에 대한 인증을 받게 된다. 한 번 게이트웨이에서 인증 과정이 수행되면 그 후에 게이트웨이를 통한 응용 서비스 요청은 부가적인 인증 없이 수행된다. 게이트웨이는 자신이 속한 네트워크의 응용 서버로 향하는 모든 데이터의 흐름을 조절하며 감시할 수 있기 때문에 응용 서버로 접근하는 클라이언트의 정보의 직접적인 관리를 통하여 해당 응용 서버의 부가적인 수정 없이 접근 제어와 같은 사용자 관리 기능을 제공할 수 있는 장점이 있다. 하지만 해당 네트워크내의 서비스를 이용하는 사용자의 수가 증가할 경우 게이트웨이를 통과하는 모든 정보를 관리하는데 오버헤드가 발생한다.



〔그림 9〕 Gateway-based SSO Model

4) 혼합 단일 인증 모델

혼합 단일 인증 모델은 에이전트 기반의 단일 인증 모델의 특징인 인증 과정의 유연성 및 간편성과 브로커 기반 단일 인증 모델의 특징인 중앙 집중식 사용자 관리 및 인증 기능을 모두 포함한다.



〔그림 10〕 Mixing SSO Model

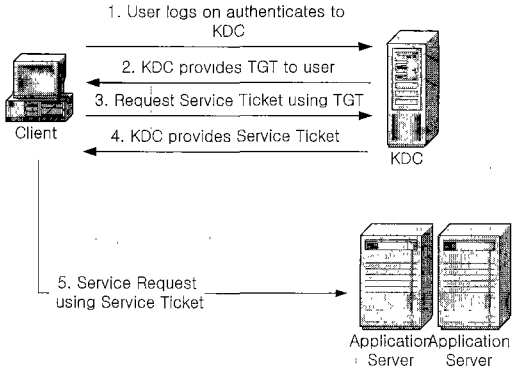
따라서 그림 10과 같은 혼합 단일 인증 모델은 브로커 기반 인증 모델에서 클라이언트와 응용 서버 사이에 존재하는 사용자 인증 관리 서버를 두며, 사용자 인증 관리 서버에서 발급한 서비스 티켓을 처리하는 응용 서버 측 에이전트가 응용 서버에 장착된다. 응용 서버 측 에이전트의 사용을 통해 응용 서버에 대한 부가적인 수정이 필요 없고 사용자에게 받은 서비스 티켓을 통해 사용자를 확인하고 알맞은 응용 서비스를 제공한다.

3. 시스템 계층에 따른 분류

1) 운영체제 포함 단일 인증 모델

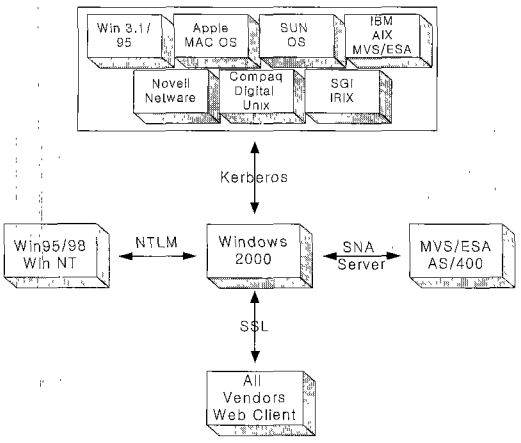
운영체제 포함 단일 인증 모델은 단일 인증 기능이 운영체제 내부에 포함되어 있어 동일한 운영체제를 사용하는 시스템들 간에 단일 인증 기능을 제공할 수 있는 모델이다. 이러한 인증 모델의 예로는 윈도우2000 운영체제가 있으며 여기서는 운영체제 포함 단일 인증 모델로서 윈도우2000 운영체제로 구성된 시스템들 간의 단일 인증 기능을 분석한다.

윈도우2000 운영체제는 기본적으로 브로커 기반의 단일 인증 모델의 전형인 커버로스 인증 프로토콜을 지원하며 액티브 디렉토리, 인증서 서비스 그리고 SSL과 같은 기능을 제공한다. 윈도우2000 운영체제는 위와 같은 인증 및 보안 기능을 통하여 윈도우2000 운영체제들로 이루어진 도메인간에 단일 인증 모델의 구축을 가능하게 한다. 그림 11은 윈도우2000 운영체제가 제공하는 커버러스 인증 모델이다. 이러한 인증 모델은 앞서 분석한 클라이언트/서버 기반 인증 모델 또는 브로커 기반의 인증 모델과 유사하다.



〔그림 11〕 OS embedded SSO Model

윈도우2000 운영체제는 여러 벤더들로 구성된 이종의 분산 네트워크 환경에서 커버로스와 같은 단일 인증 모델을 제공하고 여러 보안기능과 유연성 있는 인터페이스 및 연동 프로토콜, 게이트웨이 등을 통하여 현재의 분산 네트워크에서 부가적인 노력 없이 단일 인증 기능을 제공한다.



〔그림 12〕 Windows 2000 within Heterogeneous Networks

그림 12는 실제로 윈도우2000 운영체제에서 지원하는 커버로스 인증 프로토콜, NTLM(New Technology Lan Manager), SNA(System Network Architecture) 서버 그리고 SSL 등을 통해 이종의 네트워크들과의 상호 단일 인증 기능을 제공하는 것이 가능함을 보여준다. 또한 윈도우2000 운영체제 자체에서 인증서, SSL 그리고 IPSec 등의 보안 기능을 제공하므로 단일 인증 서비스 과정에서의 높은 보안성을 지닌다. 비록 윈도우2000 운영체제뿐만 아니라 UNIX 계열의 운영체제에서도 보안 기능과 단일 인증 기능을 포함하여 윈도우2000 운영체제와 같은 단일 인증 서비스를 제공하는 것이 가능하다.

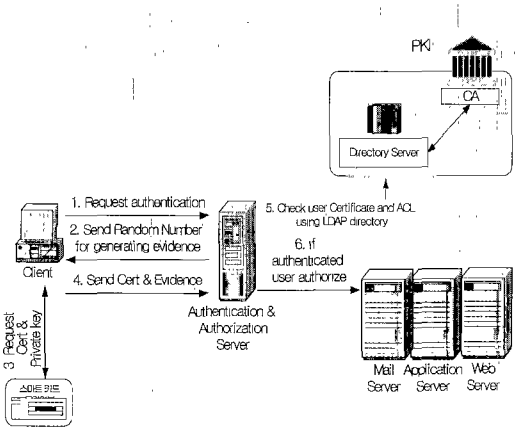
운영체제 포함 단일 인증 모델에서는 기본적으로 사용자의 인증 정보가 중앙의 인증 서버에서 관리된다. 이때 사용자 인증을 위한 정보를 관리하는 백업 서버 등을 추가로 배치함으로써 Single point of failure의 문제를 해결할 수 있다. 또한 사용자의 사용 편리성 관점에서 운영체제 포함 단일 인증 모델의 사용자 인터페이스는 운영체제 고유의 사용자 인터페이스 디자인에 종속되어 타 시스템의 운영체제에 익숙한 사용자에게는 사용의 어려움을 가져 올 수 있다. 따라서 단일 인증 기능에 관련된 사용자 인터페이스는 모든 사용자를 대상으로 하여 사용자의 편의성을 높여야 한다.

2) 응용 프로그램 통합 단일 인증 모델

인터넷 사용의 폭발적인 증가와 함께 기업이나 조직 내에서도 인트라넷이나 확장된 인트라넷인 익스트라넷 형태로 특정 집단내의 정보 공유를 위한 네트워크를 구축하여 사용하는 것이 현재의 추세이다.

인트라넷이나 익스트라넷 네트워크 환경에서 동일 벤더에 의해서 제공되는 응용 서비스를 이용한다면 이러한 응용 서비스를 하나로 묶어 단일 인증 모델을 적용하는 것이 가능하다. 예를 들면 웹 기반의 인트라넷 환경에서 웹서비스, 메일 서비스, 뉴스 그룹 서비스 등을 제공하는 응용 프로그램들에 대해서 단일 인증 모델을 적용할 수 있다.

이러한 응용 프로그램 통합 단일 인증 모델은 웹 환경에서 사용자들에게 필수적인 웹 브라우저와 함께 SSL 프로토콜, 인증서 서비스, LDAP 디렉토리 서버 등과 같은 보안 기능을 제공함으로써 보다 안전하게 사용자들이 사용 할 수 있게 한다.



〔그림 13〕 Application Integration SSO Model

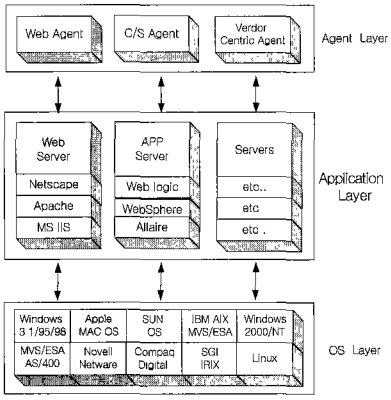
본 논문에서는 그림 13과 같은 웹 환경에서의 응용 프로그램 통합 단일 인증 모델을 분석한다. 사용자는 스마트카드에 미리 발급 받은 사용자의 인증서를 보관하며 challenge-response 스킴을 사용하여 인증서 및 사용자의 개인키에 접근 할 수 있다. 사용자가 응용 서버의 서비스를 원하는 경우에는 사용자는 자신의 인증서와 개인키로 서명된 증명서(evidence)를 신임장 서버에 보내어 정당한 사용자

임을 인증 받고 응용 서버에 접근 할 수 있는 신임장을 발급 받는다. 이때 사용자의 인증서 및 관리 그룹 정보, 접근 권한 리스트 정보, 인증서 취소 목록 등은 LDAP을 사용하여 디렉토리 서버의 사용자 정보를 통하여 검증된다. 발급 받은 신임장은 암호화되어 쿠키 형태로 응용 서버에게 보내지고 사용자는 쿠키 폼의 신임장을 가지고 웹 기반의 여러 응용 서비스를 제공받을 수 있다.

응용 프로그램 통합 인증 모델은 사용자 정보를 관리하는 인증 서버의 보안을 위해 사용자의 정보를 추가로 LDAP 디렉토리 서버에 보관한다. 또한 사용자 인증서 및 키의 보안을 위해 스마트카드를 사용하며, 웹 환경에서 사용되는 응용 프로그램 통합 인증 모델의 경우는 사용자 인터페이스로 웹 브라우저를 사용함으로써 사용자들의 편의성을 극대화 할 수 있다.

3) 단일 인증 에이전트 계층 모델

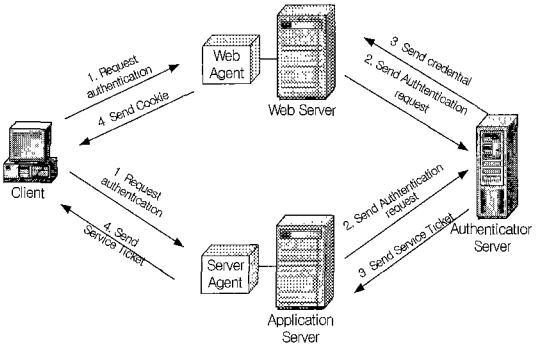
단일 인증 에이전트 계층 모델은 현재 이종의 분산 네트워크 환경에서 기존에 구성되어 있는 응용 서버 그리고 새롭게 추가되는 응용 서버들에 대하여 독립적으로 단일 인증 모델을 구축할 수 있게 만들어주는 에이전트(인터페이스 또는 플러그 인(Plug-in)) 개념의 구성 요소가 추가되는 계층을 가진 단일 인증 모델을 말한다.



〔그림 14〕 단일 인증 에이전트 계층 모델 구성도

단일 인증 에이전트 계층의 에이전트들은 현재 네트워크에 구성되어 있는 응용 서버 환경에 맞게 웹 응용 서버 에이전트, 일반 응용 서버에이전트(NT 에이전트, UNIX 에이전트 etc.)등으로 구성되며 또한 응용 서비스를 제공하려는 벤더들에게 제공하여 수정 가능한 에이전트 API도 제공된다. 그림 14는 단일 인증 에이전트 계층 모델의 전체 구조이다.

본 논문에서는 단일 인증 에이전트 계층 모델에서 제공하는 웹 서버 인증 에이전트 모델과 응용 서버 인증 에이전트 모델을 분석한다.



〔그림 15〕 Single Sign Agent Layer Model

그림 15의 단일 인증 에이전트 계층 모델은 각각의 응용 서버에 장착되는 응용 에이전트들과 사용자 인증 기능을 수행하는 인증 서버로 구성된다. 이 모델은 앞서 분석된 웹 기반 에이전트 인증 모델이나 브로커 기반 인증 모델 그리고 혼합 모델 등과 유사하다. 하지만 이 단일 인증 에이전트 계층 모델은 서비스를 제공하는 응용서버의 부가적인 수정없이 응용 서버에 적합한 에이전트의 장착으로 사용자의 서비스 요청을 처리하며 중앙의 인증 서버는 다양한 플랫폼으로 구성된 모든 응용 서버에 대해서 중앙 집중적 인증 권한 부여기능을 수행한다.

단일 인증 에이전트 계층 모델은 Single points of failure와 같은 위험에 대처하기 위하여 사용자 정보를 관리하는 인증 서버와 같은 정보를 공유하는 백업 서버를 가지고 있고, 사용자 정보의 중앙 집중적 관리를 통해 사용자에 대한 인증 및 권한 부여 등 사용자 관리에 있어서 높은 효율성을 가진다. 하지만 상이한 플랫폼에서 운영되는 응용 서버에 대해 새로운 에이전트를 추가로 개발해야 되는 것과 같은 문제점도 있다.

V. 결 론

본 논문에서는 사용자가 응용 서비스를 받을 때 반복되는 로그 인 과정에서 오는 불편함을 해결하기

위해 사용되는 단일 인증 시스템의 인증 기법과 인증 모델에 대하여 분석하였다.

단일 인증 시스템의 인증 기법은 기존의 패스워드 기반 간단한 인증 기법에서부터 현재는 PKI(Public Key Infrastructure)와 연동하여 인증서를 사용하거나 생체인증을 사용하는 등, 단일 인증 시스템이 적용된 환경에서 필요로 하는 보안 요구 사항에 따라 여러 가지 인증 기법들이 사용되고 있다.

단일 인증 시스템의 인증 모델은 인터넷 웹 환경의 증가로 웹 기반의 단일 인증 모델이 기존의 클라이언트 서버 인증 모델을 대신할 만큼 그 비중이 커지고 있으며, 또한 이종의 분산 네트워크 환경에서 적용 시 플랫폼 독립적인 특징으로 인해 에이전트 기반의 단일 인증 모델 사용이 많이 이루어지고 있다. 결국 단일 인증 모델은 본래 목적인 사용자 및 관리자의 편리성 추구는 물론이고 기존에 구축된 네트워크와 새롭게 추가될 시스템간의 호환성을 고려하여 유연성을 가질 수 있어야 한다.

앞으로는 웹 기반의 단일 인증 모델에서 신임장 폼으로 많이 사용하는 쿠키의 보안에 관한 연구와 분산 네트워크에서 서로 다른 도메인간 상호 연동이 가능한 단일 인증 모델의 연구가 필요하다.

참 고 문 헌

[1] Harold F. Tipton, *Information Security Management*, 4th Edition, Auerbach publications USA, pp.27-44

[2] J.Hursti, "Single Sign-On", *in Proceeding of Helsinke Univ. of Technology*, Seminar on Network Security, 1997

[3] T.Tervoi, "Single Sign-On Solutions in a Mixed Computing Environment", *in Proceeding of Helsinke Univ. of Technology*, Seminar on Network Security, 1998

[4] Camilloi, "Unified Single Sign-On", *in Proceeding of Helsinke Univ. of Technology*, Seminar on Network Security, 1998

[5] Anonymous, *Single Sign-On Deployment* Guide, Netscape, Inc.,1997,〈http://developer.netscape.com/docs/manuals/security/SSO/index.htm〉

[6] Anonymous, *PKCS #11: Cryptographic Token Interface Standard* - Version 2.01, RSA,1997, 〈http://www.rsa.com/rsalabs/pubs/PKCS/html/pkcs-11.html〉

[7] Anonymous, Kerberos: *The Network Authentication Protocol*, Massachusetts Institute of Technology, 1998, 〈http://web.mit.edu/kerberos/www〉

[8] Anonymous, *Smart Cards White Paper*, Microsoft,1998,〈http://www/tech/smartcards/scardwp.asp〉.microsoft.com/security〉

[9] Anonymous, *X/Open Single Sign-On Service (XSSO) - Pluggable Authentication Modules*, The Open Group,1997

[10] Schneier, B., *Applied Cryptography*, 2nd Edition, John Wiley&Sons, Inc., NY, USA, 1996, 758p.

[11] Trickey, *F. Single Sign-On: Fantasy or Reality?*, CSI Advisory Council, 1997, 〈http://www.gocsi.com/sso_ft.htm〉

[12] J. Linn. 1997. *Generic Security Service Application Program Interface, Version 2.* ftp://ftp.funet.fi/pub/standards/RFC/rfc2078.txt

[13] Transarc Corporation. 1996. *Overview of DCE.* http://www.ux1.eiu.edu/~csjay/dce/

[14] Mark Vandenwauver. 1995. *SESAME V3.* http://www.esat.kuleuven.ac.be/cosic/sesame3_2.html

[15] DataFellows Ltd. 1997. *SSH Users Administrators Guide*, pp. 97-98. Gummerus Printing.

[16] Atkinson, R., *Security Architecture for the Internet Protocol*, RFC 1825, August 1995 〈ftp://ftp.funet.fi/pub/standards/RFC/rfc1825.txt〉

[17] eTrust, *eTrust SSO white paper*, Computer Associates, http://www.ca.com

[18] Securant, *Securant SSO*, OpenInfra, http://www.openinfra.com/report/sso

[19] Anonymous, *Single Sign On white paper*, Systems Approach Corporation, 1999, http://1canada.com/sso.htm

[20] ETRI, *주간기술동향*, 955호, 2000, 7.19

[21] 김태훈, "스마트카드에 적합한 효율적인 인증 모

델", *한국 통신 학회*, 하계종합학술발표회 논문집, pp.605-608, 1996
[22] 김동규, "브로커 및 에이전트 기반의 통합 단일 인증 시스템", *한국정보과학회*, 춘계종합학술대회 논문집, pp.829-831, 2001

〈著 者 紹 介〉

손 태 식 (Tae-Shik Shon)
2000년 : 아주대학교 정보 및 컴퓨터 공학부 졸업(학사)
2000년~현재 : 아주대학교 정보통신공학과 석사과정
관심분야 : 네트워크 보안, 인터넷 프로토콜 보안

이 상 하 (Sang-Ha Lee)
1987년 : 울산대학교 전자계산학과 졸업(학사)
1991년 : 아주대학교 컴퓨터공학과 졸업(석사)
2000년 : 아주대학교 컴퓨터공학과 박사 과정 수료
2000년 3월-현재 : 동서울대학교 정보통신공학과 교수
관심분야 : 정보통신 Security, 네트워크 관리, 분산처리시스템 보안

유 승 화 (Seung-Wha Yoo)
1972년 : 서울대학교 공과대학 응용수학과(학사)
1980년 : University of Kansas Computer science (석사)
1983년 : University of Kansas Computer science (박사)
1999년~현재 : 아주대학교 정보 및 컴퓨터 공학부 교수, AT&T Bell Labs 연구원, Amdahl Corporation 수석연구원, 삼성전자 정보통신 전무, 충남대학교 공과대학 겸임교수, 한국정보과학회 부회장, 한국 네트워크 연구조합 이사장 역임
관심분야 : 고급국지 통신망, 망관리, VoIP

김 동 규 (Dong-Kyoo Kim)
1973년 : 서울대학교 공과대학 응용수학과 졸업(학사)
1979년 : 서울대학교 자연과학대학원 전자계산학 졸업(석사)
1984년 : Kansas State University 전자계산학(박사)
1979년~현재 : 아주대학교 정보 및 컴퓨터공학부 교수, IEEE 802.4,802.6,802.10 Working Group Member, Asiacrypt '96 조직위원회 위원장, 건설교통부 항공교통관제소 신항공 교통관제 시스템 평가위원회 위원, 한국과학기술연구소 연구원, 한국통신학회 상임이사, 한국통신정보보호학회 부회장 역임
관심분야 : 컴퓨터 통신, 정보보호, 프로토콜 엔지니어링