情報保護學會論文誌
第19卷 第3號, 2009. 6

인터넷 환경에서의 사용자 중심 ID정보 관리 모델에 관한 연구

이 해 규,† 신 현 식‡
서울대학교 컴퓨터공학부

A Study On User-centric Identity Information Management Model In Internet Environment

Hae-gyu Rhy,† Hyeonshik Shin‡
Department of Computer Science and Engineering, Seoul National University

요 약

인터넷의 발전으로 사용자들이 이용하는 인터넷 서비스들의 수가 많아지면서 서비스 제공자들이 수집하는 사용자의 실체(Identity)에 관한 개인정보, 즉 ID정보의 적절한 관리는 사용자의 편리성 증대와 프라이버시 보호 측면에서 매우 중요하게 되었다. 그러나 기존의 ID정보 관리 모델들은 서비스 제공자 중심의 관리이거나 제공하는 ID정보 관리 기능이 부족하여 사용자의 기대를 만족시키지 못했다. 본 논문에서는 ID정보 관리에서의 사용자 중심 개념을 정립하여, 사용자들의 인터넷 서비스 이용이 더욱 편리해지고, 궁극적으로 프라이버시가 강화될 수 있는 새로운 사용자 중심 ID정보 관리 모델을 제안한다.

ABSTRACT

With the increase of number of internet services due to development of internet, it becomes a critical issue to appropriately manage the personal information about user's identity, that is ID information that service providers collect in terms of user's convenience improvement and privacy protection. But the existing ID information management models are service provider-centric or they insufficiently provide ID information management functions, so that they have not been satisfying user's requirement. In this paper, by establishing user-centric concept in ID information management, we propose the new user-centric ID information management model in which user can make use of the internet service more conveniently and eventually enhance user's privacy.

Keywords: ID Management, Authentication, Trust, Security

I. 서 론

인터넷 기술 (Internet Technology)의 지속적인 발전으로 오프라인의 거의 모든 서비스들이 인터넷에서도 가능하게 되었고 이에 따라 사용자가 이용하는 인터넷 서비스들의 수도 점점 많아지게 되었다. 일반적으로 인터넷 서비스 제공자 (Service Provider, SP)들은 서비스 제공을 명분으로 사용자의 실체(Identity)와 관련된 개인정보(이하 ID정보)를 수집하여 개별적으로 관리하는데 이러한 개별적인 ID정보 관리 방식은 한 사용자가 이용하는 서비스들의 수가 많아지면서 사용자 편리성과 프라이버시 측면에서 문제점을 보이기 시작했다[1]. 사용자는 SP에 회원가입을 할 때 마다 거의 비슷한 내용의 ID정보를 중복 제공해야 하고, ID정보의 일부가 변경 되었을 때 이를

접수일(2008년 12월 18일), 수정일(2009년 4월 7일), 게재확정일(2009년 5월 18일)
† 주저자, rhg@kt.com
‡ 교신저자, shinhs@snu.ac.kr

반영하려면 ID정보 제공을 했던 SP들을 모두 기억해서 일일이 변경을 해야 하며, SP 서비스를 이용할 때마다 매번 인증을 받아야 한다. 더구나 SP들의 주 목적은 서비스 제공이기 때문에 ID정보 관리에 소홀한 경향이 많아 ID정보 유출로 인한 프라이버시 침해가 많이 발생할 수 있다. 이러한 문제를 해결하기 위해 ID정보 관리를 전담하는 Identity Service Provider (IDP)를 두거나 사용자 중심 개념을 적용한 ID정보 관리 모델들이 제시되었으나 사용자의 편리성 증대와 프라이버시 문제를 근본적으로 해결하지 못했다. 본 논문에서는 ID정보 관리에서 사용자 중심 개념을 새롭게 정립하여, 사용자들의 인터넷 서비스 이용이 더욱 편리해지고, 궁극적으로 사용자의 프라이버시가 강화될 수 있는 새로운 사용자 중심 ID정보 관리 모델 (User-centric ID Information Management Model, U2IM)을 제안한다. 논문의 구성은 다음과 같다. 2장에서는 ID정보 관리 모델의 이해를 위해 필요한 개념 정의와 기존의 ID정보 관리 모델들을 소개하고, 3장에서는 본 논문에서 제안하는 U2IM을 설명하고, 4장에서는 U2IM을 몇 가지 관점에서 분석하고, 5장에서 결론을 맺는다.

II. 개념 정의와 기존 ID정보 관리 모델

본장에서는 ID정보와 ID정보 관리 모델의 개념을 정의 하고, 기존의 대표적인 ID정보 관리 모델인 .NET Passport, Liberty Alliance, OpenID 및 CardSpace 대해 소개하겠다.

2.1 ID정보 정의

ID정보는 사용자의 실체(Identity)와 관련된 모든 정보를 의미하는데 본 논문에서는 ID정보를 사용자 실체와 연관되는 정도에 따라 신원 ID정보, 인증 ID정보, 일반 ID정보, 및 부가 ID정보로 분류하였다.

1) 신원 ID정보

사용자의 신원을 유일하게 식별하기 위해 공공기관이 발급한 정보이다. 대한민국의 주민등록번호나 i-PIN(internet-Personal Identification Number), 미국의 SSN(Social Security Number) 등이 여기에 해당한다.

2) 인증 ID정보

SP가 사용자의 인증을 위해서 요구하는 정보를 의미한다. 현재 대부분의 SP들이 사용하는 id와 password가 인증 ID정보에 해당한다. id는 identifier의 약자로서 사용자의 실체를 의미하는 대문자 약자인 ID와는 구분된다.

3) 일반 ID정보

SP들이 서비스 제공을 위해 사용자로부터 수집하는 ID정보들 중 공통되는 정보들로서 사용자의 실체와 관련된 일반적인 성격을 띠고 있는 정보들이다. 사용자의 이름, 성별, 나이, 주소, 전화번호 및 e-mail 주소 등이 여기에 포함되며 일반 ID정보의 단독 혹은 결합으로 사용자의 식별이 가능하다.

4) 부가 ID정보

부가 ID정보는 사용자의 실체와 관련된 부가적인 정보로서 SP에 따라 수집하는 부가 ID정보의 종류가 달라진다. 특정 SP 내에서의 사용자 별명 (Nick Name), 영화정보 제공 SP에서의 좋아하는 영화 종류나 축구정보 제공 SP에서의 좋아하는 축구선수 등의 정보가 부가 ID정보에 속한다. 부가 ID정보의 단독 혹은 결합으로는 사용자의 식별이 가능하지 않다.

2.2 ID정보 관리 모델

여러 SP들이 개별/독립적으로 ID정보를 관리하는 구조에서 나타나는 문제점을 해결하기 위한 여러 시도들은 공통적인 구조를 갖는 ID정보 관리 모델로 정형화 되었다. 일반적인 ID정보 관리 모델은 사용자, IDP 및 SP로 이루어지는 구조로서 ID정보 관리를 전담하는 IDP가 사용자의 ID정보를 저장/관리하면서 ID정보 관리 기능을 제공한다. ID정보가 사용자 쪽에 저장되어 관리되는 모델도 있었지만 거의 활성화 되지 않아 일반적인 ID정보 관리 모델에서 제외하였다.

2.2.1 구조

1) 사용자

ID정보 소유자로서의 사람과 사용하는 웹브라우저를 의미한다.

2) IDP

사용자의 ID정보를 관리해주는 서비스 제공자로서 사용자의 ID정보를 저장/관리하면서 사용자의 요구에 따라 ID정보 관리 기능을 제공한다.

3) SP

일반적인 인터넷 서비스 제공자이다.

2.2.2 정적신뢰와 동적신뢰

ID정보 관리 모델에서 IDP와 SP 간에는 ID정보와 ID정보 관리에 필요한 각종 제어 정보들의 전송이 이루어져야 하는데, 이를 위해서 IDP와 SP 상호 간에 정보 전송에 대한 신뢰관계가 형성되어야 한다. 신뢰관계의 형태는 사업협약의 유무에 따라서 나누어질 수 있는데, IDP와 SP 간의 사업협약을 바탕으로 형성되어 협약이 파기되기 전까지 계속 유지되는 신뢰를 정적신뢰 (Static Trust)라고 하고, 사업협약 없이 IDP와 SP 간에 동적으로 형성되고 해제되는 신뢰를 동적신뢰 (Dynamic Trust)라고 한다[6].

2.2.3 ID정보 관리 기능

IDP가 제공할 수 있는 ID정보 관리 기능은 크게 4가지가 있을 수 있는데 ID정보 관리 모델에 따라서 일부 혹은 전부가 지원될 수 있다.

1) id연계
한 사용자의 id는 IDP와 SP들에 각기 다르게 설정되는 것이 일반적이다. 이렇게 다르게 설정된 id들을 IDP id를 중심으로 개별 SP의 id들과 연계시켜 한 사용자로 인식할 수 있게 하는 것이 id연계 기능으로 ID정보 관리의 기본적인 기능이다.

2) 일반 ID정보 제공
사용자가 SP에 회원가입을 할 때 사용자의 동의를 받아 IDP에 저장되어 있는 일반 ID정보를 SP에 제공하는 기능이다.

3) 일반 ID정보 변경
사용자의 일반 ID정보 중 일부가 변경 되었을 때 IDP는 사용자의 요청에 의해 변경된 정보내역을 SP에 전송하여 변경된 내역을 반영할 수 있게 하는 기능이다.

4) SSO (Single Sign-On)
사용자가 IDP에 한번의 인증으로, 추가의 인증 없이 다른 SP들을 이용할 수 있게 해주는 기능이다[14].

2.3 기존 ID정보 관리 모델

2.3.1 .NET Passport[7]

인터넷 ID정보 관리 모델로서는 최초라고 할 수 있는 MS의 .NET Passport는 SP들 대상의 서비스 형태로 시작된 것이었다. .NET Passport 서비스는 MS가 IDP가 되어 SP들을 대상으로 ID정보 관리 서비스를 제공하고, 이러한 SP들을 이용하는 사용자들은 자동적으로 MS IDP로부터 ID정보 관리 서비스를 제공 받는 구조이다. IDP가 제공하는 ID정보 관리 기능은 일반 ID정보 제공과 SSO이다. .NET Passport는 MS만이 IDP가 되겠다는 독점적인 발상에서 나온 모델로 다른 주요 인터넷 사업자들의 반발을 불러 일으켰다. 또한 사용자의 실질적인 이용과 무관한 MS와 SP간의 사업협약으로 서비스가 이루어지기 때문에 사용자의 편리성 관점에서도 한계가 있었다. 이러한 문제로 인해서 .NET Passport는 크게 활성화 되지 못했으며 MS에서도 더 이상 서비스 확장을 시도하지 않고 있다.

2.3.2 Liberty Alliance[5]

.NET Passport에 반발하여 SUN 등을 중심으로 하는 인터넷 주요 사업자들이 새로운 인터넷 ID정보 관리 모델로 제시한 것이 Liberty Alliance이다. 서비스 형태인 .NET Passport와는 달리 Liberty Alliance는 표준 규격 형태로 발표되었다. 따라서 어떤 SP도 Liberty Alliance의 표준 규격에 맞게 시스템을 갖추게 되면 IDP가 될 수 있는 개방적인 구조이다. IDP와 SP간에 형성되는 신뢰그룹(Circle Of Trust, COT)의 형태는 사업협약을 기반으로 하는 정적신뢰이며 IDP가 제공하는 ID정보 관리 기능은 id연계와 SSO이다. 또한 COT IDP들 간에 정적신뢰가 형성되면 COT간에도 연동을 할 수 있는 확장 가능한 구조이다. 그러나 ID정보 관리의 기본 단위인 COT가 사용자의 요구가 아닌 IDP와 SP간의 사업협약에 따라 이루어지는 것이고 제공되는 ID정보 관리 기능도 주로 SSO에 한정되기 때문에 사용자의 실질적인 요구수준을 만족하기에는 한계가 있다.

2.3.3 OpenID[9]

OpenID는 사용자 중심 개념을 ID정보 관리에 최초로 도입한 모델로서 인터넷에서 사용하는 id를 사용자가 직접 지정하고 이 id로 모든 SP들을 이용하자는 개념이다. OpenID 사용자 id는 URL (Uniform Resource Locator)형태로서 사용자의 IDP가 무엇인지 알 수 있는 정보가 포함되어 있다. 사용자가 SP

에게 URL id를 제출하면 SP는 id로부터 사용자의 IDP를 알아내어, IDP에게 사용자의 인증확인 요청을 한다. 사용자가 인증이 되면 IDP는 SP에게 사용자의 인증 확인을 해주고 SP는 이를 신뢰하여 추가의 인증 없이 사용자에게 서비스를 제공한다. 즉 ID정보 관리 기능 중 SSO 기능이 제공된다. OpenID는 Liberty Alliance와 마찬가지로 표준 규격 형태이나, IDP와 SP의 신뢰형태는 Liberty Alliance의 정적신뢰와는 달리 IDP와 SP간의 사업협약을 전제로 하지 않는 동적신뢰이다. 그러나 OpenID에서의 동적신뢰는 사업협약은 없지만 표준규격을 바탕으로 한 IDP와 SP들의 선택에 의해 이루어져서 결국 사용자의 요구와는 무관하게 이루어지기 때문에 사용자 입장에서의 효과는 정적신뢰와 다를 바가 없다. 또한 IDP가 제공하는 관리 기능이 SSO에만 한정 돼 있어 ID정보의 전반적인 관리 모델로서도 한계가 있다.

2.3.4 CardSpace[8]

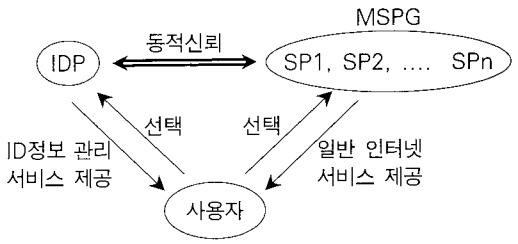
.NET Passport의 추진으로 ID정보 독점화에 대한 비판을 받았던 MS는 인터넷 환경이 사용자 중심 개념이 적용되는 웹2.0 시대로 변화함에 따라, 기존의 ID정보 관리 모델을 수용하면서 사용자의 ID정보 통제 역할을 강화하는 메타 시스템 개념인 CardSpace라는 새로운 ID정보 관리 모델을 발표하여 OS 비스타에 탑재하기 시작했다. 즉 CardSapce는 사용자, IDP 및 SP 구조로 돼 있는 기존의 여러 ID정보 관리 모델들을 수용, 통합하는 메타 모델로서, 현실 세계의 신원확인 방식을 모델링해서 사용자의 직관력을 높인 UI를 통해 SP에게 제공하는 ID정보를 사용자가 직접적으로 통제할 수 있는 구조를 갖고 있다. 따라서 CardSpace에서의 사용자 중심 개념은 사용자의 이용 편리성 보단 프라이버시 보호에 더 초점을 맞춘 것이라고 볼 수 있다. 그러나 클라이언트 ID정보의 사용자 통제가 과연 실효성 높은 프라이버시 보호 장치인지는 의문이다. 또한 메타 모델이기 때문에 사용자 편리성 관점에서 볼 때 IDP와 SP 간의 정적신뢰를 기본으로 하는 기존 ID정보 관리 모델의 한계점을 그대로 갖고 있다.

III. 사용자 중심 ID정보 관리 모델 (U2IM)

본 논문에서 제안하는 U2IM은 사용자가 실질적으로 중심이 되어 ID정보를 관리할 수 있는 모델로서 본 장에서는 U2IM의 구조, 사용자 중심 개념, ID정보 관리 기능 및 프로토콜에 대해서 설명하겠다.

3.1 구조

U2IM 구조는 그림1에서 볼 수 있듯이 ID정보의 소유자인 사용자와 ID정보 관리 기능을 수행하는 IDP 및 서비스를 제공하는 SP들로 이루어진다. 사용자는 여러 IDP들 중 자신의 ID정보를 관리할 수 있는 신뢰하는 IDP를 선택하여 회원가입을 한다. 회원가입 과정에서 IDP는 사용자의 신원을 확인하고 사용자의 모든 ID정보, 즉 신원/인증/일반 ID정보와 필요하면 부가 ID정보를 수집한다. IDP에 회원가입이 되면 사용자는 IDP를 통해서 자신이 이용하려는 SP들을 대상으로 ID정보를 관리할 수 있다.



〔그림 1〕 U2IM 구조

3.2 사용자 중심 개념

U2IM에서의 사용자 중심 개념은 사용자가 SP들에 제공한 ID정보들에 대한 관리, 즉 제공, 변경, 폐기 및 제어 등을 사용자의 요구에 따라 완전하게 할 수 있는 것을 의미한다. 이를 위해서 U2IM에서는 관리 SP 그룹 (Management SP Group, MSPG)이라는 것을 정의한다. MSPG는 사용자가 이용하는 SP들 중에서 IDP를 통해 ID정보를 관리하기를 원하는 SP들의 그룹을 의미한다. 사용자가 이용하는 SP들 중에서 IDP를 통한 ID정보 관리가 필요치 않는 경우는 MSPG에 속하지 않는다. MSPG 형성은 사용자의 요청에 의해서 IDP가 해당 SP들과 동적신뢰 관계를 맺음으로써 이루어진다. MSPG가 형성되면 사용자는 MSPG를 대상으로 ID정보를 완전하게 관리할 수 있다. 〔그림 1〕은 IDP와 SP들 간의 동적신뢰를 통해서 MSPG가 형성된다는 것을 보여준다.

3.3 ID정보 관리 기능

U2IM에서는 IDP가 제공할 수 있는 모든 ID정보 관리 기능을 제공한다.

1) id연계
한 사용자에 대한 IDP와 MSPG의 각기 다른 id들은 IDP와 MSPG의 각 SP간에 id연계를 통해 한 사용자로 인식된다.

2) 일반 ID정보 제공
IDP에 저장되어 있는 일반 ID정보는 사용자가 MSPG의 SP에 회원가입 시에 사용자의 동의 범위 내에서 해당 SP에 제공된다.

3) 일반 ID정보 변경
사용자의 일반 ID정보의 일부가 변경 되었을 때 사용자는 IDP에 저장되어 있는 일반 ID정보를 변경하고 IDP는 변경된 내역을 MSPG의 모든 SP에 전송하여 이를 반영하게 한다.

4) SSO
SSO를 위해서 사용자는 MSPG 중에서 SSO로 인증되길 원하는 SP 그룹을 정의하는데 이를 SSPG (SSO SP Group)라고 한다. MSPG 내에서 SSPG를 따로 정하는 이유는 U2IM의 SSO 방식은 사용자의 IDP 인증으로 IDP 인증쿠키와 SSPG의 모든 SP들의 인증쿠키를 일괄 발행하기 때문에 효율성 차원에서 SSO 인증이 꼭 필요한 SP들의 인증쿠키만 발행하는 것이 바람직하기 때문이다.

3.4 프로토콜

U2IM 프로토콜은 U2IM의 각 구성요소들이 ID정보 관리 기능을 위해 수행하는 약속된 절차를 의미한다. U2IM 프로토콜을 주요 단계 별로 다음과 같이 나누어 설명하겠다.

- IDP 선택과 회원가입
- 동적신뢰를 통한 MSPG 형성
- id연계와 일반 ID정보 제공
- 일반 ID정보 변경
- SSO
- 동적신뢰 해제와 사용자 동의정보 a의 변경

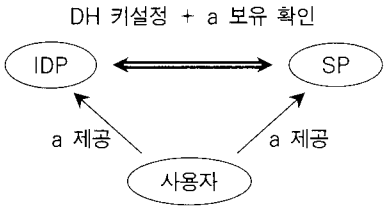
3.4.1 IDP 선택과 회원가입

사용자가 자신이 신뢰하는 IDP를 선택하여 회원가입을 할 때 IDP는 사용자의 ID정보들을 수집하는데 U2IM에서의 IDP는 사용자 프라이버시 보호를 위한 특별한 인증 ID정보로 사용자 동의정보 a(greement)를 추가로 수집한다. a는 사람이 기억할 수 있는 password 정도의 크기로 HTTPS와 같은 안전한 채널을 통해 IDP에게 전달된다. a는 나중에 IDP가 특정 SP와 동적신뢰 관계를 맺을 때 IDP의 경우와 마찬가지로 사용자가 안전한 채널을 통해 SP에 제공한다. 이렇게 사용자의 동의정보 a를 IDP와 SP가 공유하게 되면 IDP와 SP는 a의 보유 여부 확인을 통해 동적신뢰 관계를 맺게 되고, 이 후 ID정보 관리 기능을 위해 IDP와 SP 간에 정보 전송이 필요할 때 a 보유여부의 상호확인을 통해 서로를 인증하게 된다. 결국 a는 사용자에게는 IDP와 SP에게 자신의 동의를 알려주는 정보로서의 역할을 하며 IDP와 SP에게는 a 보유 여부의 상호 확인을 통해서 서로를 인증하고 사용자의 동의 여부를 확인하게 해주는 역할을 한다.

3.4.2 동적신뢰를 통한 MSPG 형성

동적신뢰는 사용자의 요청으로 IDP와 SP간에 이루어지는데 동적신뢰가 형성되면 해당 SP는 MSPG에 포함된다. 사업협약을 전제로 하는 정적신뢰는 신뢰 당사자들이 사전에 신뢰를 위한 비밀정보를 미리 교환하거나 혹은 기존에 구축 돼 있는 PKI나 Kerberos 같은 암호학적인 인프라를 이용할 수 있지만 U2IM의 동적신뢰는 사용자의 요청에 의해 동적으로 형성되는 것이기 때문에 사전에 정보 교환을 하거나 사업협약이 필요한 기존의 인프라를 이용하는 것이 어렵다. U2IM에서는 동적신뢰 형성을 위한 방법으로 Diffie-Hellman (DH) 키설정 방식에 동의정보 a를 결합한 방식을 제안한다. DH 방식은 정보교환을 하려는 상호 간에 동일한 세션키를 형성하는 방식으로 양쪽이 각각 키설정을 위한 비밀정보와 공개정보를 생성하여 공개정보를 서로 교환하고, 비밀정보와 상대방으로부터 받은 공개정보를 이용하여 키 계산을 하면 암호학적 계산식에 의해 같은 세션키가 생성되는 원리를 이용한 방식으로 안전성은 Discrete Logarithm 문제의 복잡도에 기반을 두고 있다. 그러나 DH 방식은 상호 간에 교환되는 공개정보에 대한 인증 장치가 없어 공개정보를 중간에서 변조할 수 있는 중간자 공

격 (Man-in-the-Middle- Attack)에 취약하다[3]. 이를 보완하기 위해서 U2IM에서는 공개정보와 a를 해쉬한 값을 서로 교환하여 확인함으로써 공개정보와 공개정보를 보낸 상대방을 인증한다. a를 보유하고 있다는 것은 사용자의 동의를 통해 동적신뢰가 형성된 상대방이라는 인증이 되기 때문에 상대방과 상대방이 보낸 공개정보를 신뢰할 수 있게 되어 중간자 공격의 취약점이 보안되는 것이다. 동적신뢰를 통한 MSPG 형성 프로토콜의 세부적인 내용은 다음과 같다. (DH 키설정을 위한 공개정보들인 소수 p와 α (〈p, p의 primitive root) 값들은 사전에 미리 정의돼 있는 것으로 가정한다.)



[그림 2] 동적신뢰 형성

1) 사용자

① ID정보 관리가 필요한 SP를 선택한 후, IDP에게 해당 SP를 MSPG에 포함하라고 요청

2) IDP

② DH 키설정을 위한 비밀정보 r1과 공개정보 u1(=α^{r1}mod p)을 생성

③ u1과 사용자 동의정보 a를 해쉬하여 h(=H(u1||a))를 생성하고, u1, h를 SP에 전송하면서 동적신뢰 요청

- 전송방식: HTTP POST Redirection (IDP->사용자->SP)

3) SP

④ IDP로부터 u1, h가 포함된 동적신뢰 요청을 받고, 사용자에게 동의정보 a의 제공을 요청

4) 사용자

⑤ SP의 요청을 받고 웹브라우저 상에서 a를 입력하고 SP에 전송

- 전송방식: HTTPS (사용자->SP)

5) SP

⑥ 사용자로부터 받은 a와 IDP부터 받은 u1로 h′(=H(u1||a))를 계산

⑦ h와 h′가 같으면 IDP로부터의 동적신뢰 요청을 수락

i) DH 키설정을 위한 비밀정보 r2와 공개정보 u2(=α^{r2}mod p, ≠u1)를 생성

ii) u2와 사용자 동의정보 a를 해쉬한 값 h(=H(u2||a))를 생성하고, u2, h를 IDP에 전송

- 전송방식: HTTP POST Redirection (SP->사용자->IDP)

iii) IDP를 사용자의 IDP로 등록

⑧ h와 h′가 다르면 IDP로부터의 동적신뢰 요청을 수락하지 않고 오류상황을 IDP에 전송

6) IDP

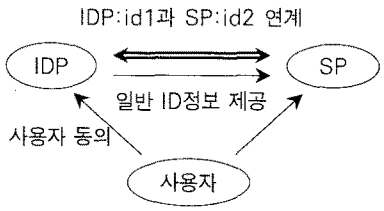
⑨ SP로부터 받은 u2와 저장하고 있던, 사용자 동의정보 a로 h′(=H(u2||a))를 계산하여 SP로부터 받은 h와 비교하여 동일하면 SP를 MSPG에 포함

⑩ h와 h′가 다르면 오류상황을 사용자에게 알림

3.4.3 id연계와 일반 ID정보 제공

U2IM id연계의 기본적인 방법은 한 사용자의 IDP의 id가 id1, SP의 id가 id2라고 가정할 때, IDP가 연계를 위한 매개정보 m을 생성하여 SP에 전송하면 SP는 m을 id2로 연계 시켜 같은 사용자로 인식하는 것이다. 이 때 사용자의 매개정보 m은 연계되는 SP마다 각기 다르다. 즉 IDP가 동일 사용자에 대해서 SP1에 연계 시키는 매개정보와 SP2에 연계 시키는 매개정보는 서로 다르다. 이렇게 하는 이유는 SP들끼리의 정보담합으로 사용자를 식별하여 프라이버시를 침해하거나 IDP 가장(Spoofing) 공격[13] 등을 방지하기 위해서이다. id연계와 함께 일반 ID정보 제공이 이루어지는데 IDP가 SP에 제공하는 일반 ID정보의 범위는 사용자가 미리 지정하여 정한다. 사용자가 SP에 처음 가입하는 경우는 IDP가 제공하는 일반 ID정보를 SP는 그대로 저장하고 사용자가 이미 SP에 가입되어 있는 경우는 기존에 저장하고 있는 일반 ID정보를 IDP가 제공하는 일반 ID정보로 대체한다. id연계와 일반 ID정보 제공은 단독으로 이루어지는 것이 아니라 동적신뢰를 통해 특정 SP가 MSPG에 포함되

면서 연속적으로 이루어진다. 따라서 아래에서 설명하는 프로토콜은 3.4.2의 동적신뢰를 통한 MSPG 형성 프로토콜의 연속적인 과정이다. IDP가 SP에게 제공하는 일반 ID정보를 G로 표기한다.



〔그림 3〕 id연계와 일반 ID정보 제공

7) IDP

⑪ r1과 u2를 사용하여 세션키 $k(=\alpha^{r1r2} \bmod p)$와 id연계를 위한 매개정보 m을 생성

⑫ m, 일반 ID정보 G를 해쉬하여 $h(=H(m||G))$를 생성하고 m, G, h를 세션키 k로 암호화하여 $M(=E_k(m||G||h)$을 생성

⑬ m과 M을 SP에 전송

- 전송방식: HTTP POST Redirection 방식 (IDP -〉 사용자 -〉 SP)

8) SP

⑭ r2와 u1을 사용하여 IDP와 동일한 세션키 $k(=\alpha^{r2r1} \bmod p)$를 생성하고, k를 사용하여 IDP로부터 받은 M정보를 복호화

⑮ m, G를 해쉬하여 $h'(=H(m||G)$를 생성하고, h와 같은지를 비교하여 정보 무결성 확인

⑯ h와 h′가 같으면 m을 id연계를 위한 사용자의 인증 ID정보로, G를 사용자의 일반 ID정보로 저장하고 IDP에게 정상적인 완료 통보

i) IDP와는 별도의 사용자 id나 부가 ID정보가 필요하면 사용자로부터 id/password와 부가 ID정보를 수집

⑰ h와 h′가 다르면 무결성이 깨진 것으로 판단하고 오류상황을 IDP에 전송

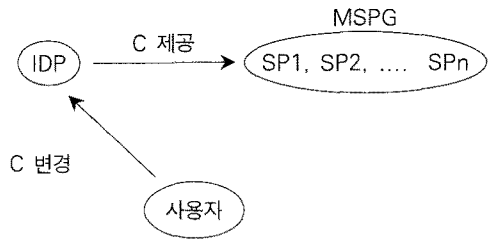
9) IDP

⑱ SP로부터 정상적인 완료통보를 받으면 id연계와 일반 ID정보 제공과정이 종료

⑲ SP로부터 오류상황 통보를 받으면 해당상황을 사용자에게 알림

3.4.4 일반 ID정보 변경

사용자의 일반 ID정보 중의 일부가 변경되면 변경된 내역은 사용자의 동의 하에 IDP가 MSPG의 모든 SP에게 전송하여 반영되게 한다. 변경되는 일반 ID정보를 C로 표기한다.



〔그림 4〕 일반 ID정보 변경

1) 사용자

① 사용자는 IDP에 저장된 일반 ID정보 C를 변경하고, IDP에게 C를 MSPG에 반영하라고 요청

2) IDP

② IDP는 DH 키설정을 위한 비밀정보 r1과 공개정보 $u1(=\alpha^{r1} \bmod p)$을 생성

③ u1, id연계를 위한 매개정보 m, 사용자 동의정보 a를 해쉬하여 $h(=H(u1||m||a))$를 생성하고 u1, m, h를 MSPG의 SP에 전송

- 전송방식: HTTP POST (IDP -〉 SP)

3) SP

④ IDP로부터 받은 정보 중 매개정보 m으로 SP에서의 사용자를 식별하여 사용자의 IDP가 맞는지와 동의정보 a를 확인

⑤ 해쉬값 $h'(=H(u1||m||a))$를 생성하여 h와 같은지를 비교

⑥ h와 h′가 같으면 동적신뢰에 기반한 IDP 인증과 전송 정보들의 무결성이 확인

i) DH 키설정을 위한 비밀정보 r2와 공개정보 $u2(=\alpha^{r2} \bmod p, \neq u1)$를 생성

ii) u2와 동의정보 a를 해쉬한 값 $h(=H(u2||a))$를 생성하여 u2와 h를 IDP에 전송

- 전송방식: HTTP POST (SP-〉IDP)

⑦ h와 h′가 다르면 동적신뢰와 무결성이 확인되지 않은 것이므로 오류상황을 IDP에 통보

4) IDP

⑧ SP로부터 받은 u2와 갖고 있는 동의정보 a의 해쉬값 h′(=H(u2||a))를 계산

⑨ h와 h′가 같으면 동적신뢰에 기반한 SP 인증과 u2의 무결성이 확인

i) 세션키 $k(=\alpha^{r1r2} \bmod p)$를 생성하고, m과 C의 해쉬값 h(=H(m||C))을 생성

ii) m, C, h를 세션키 k로 암호화 한 $M(=E_k(m||C||h))$을 생성하여 SP에 전송

- 전송방식: HTTP POST (IDP->SP)

⑩ h와 h′가 다르면 동적신뢰에 기반한 SP 인증과 u2의 무결성이 확인되지 않은 것이므로 오류상황을 사용자에게 알림

5) SP

⑪ 세션키 $k(=\alpha^{r2r1} \bmod p)$를 생성하여 M을 복호화하고 h′(=H(m||C))를 계산

⑫ h와 h′가 같으면 C를 사용자의 일반 ID정보에 반영하고 IDP에 변경완료 통보

⑬ h와 h′가 다르면 오류상황 통보

-전송방식: HTTP POST (SP->IDP)

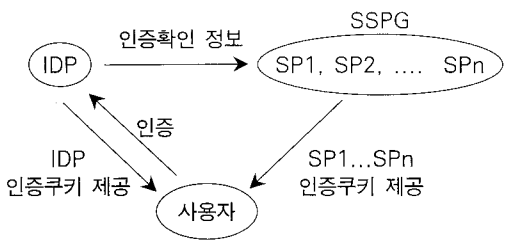
6) IDP

⑭ SP로부터 정상적으로 변경되었다는 통보를 받으면 MSPG의 모든 다른 SP에 반복 수행하고 결과를 사용자에게 알림

⑮ SP로부터 중간에 오류상황 통보를 받으면 해당 상황을 사용자에게 알림

3.4.5 SSO

U2IM에서 SSO가 이루어지는 기본적인 방법은 사용자가 IDP에 성공적으로 인증을 하면 IDP는 IDP 인증쿠키를 발행하고, IDP가 사용자의 SSPG의 개별 SP들에게 IDP 인증확인 정보를 전달하면 개별 SP들은 인증확인 정보를 검증하여 자신들의 인증쿠키를 발행한다. 이렇게 하면 사용자는 결국 IDP와 SSPG의 SP들의 모든 인증쿠키를 보유하게 돼서 SSO가 가능하게 된다. SSO 프로토콜에서는 공격자의 재사용 공격을 방지하기 위해서 IDP가 nonce 정보를 생성하여 SP에 보내는 정보에 추가한다.



〔그림 5〕 SSO

1) 사용자

① IDP에 인증을 하여 IDP 인증쿠키를 확보

2) IDP

② id 연계를 위한 매개정보 m, 인증확인 정보 A, nonce n 및 사용자의 동의정보 a를 해쉬한 값 h(=H(m||A||n||a))를 계산하여 m, A, n, h를 SSPG의 SP에 웹 Redirection하여 전송

- 전송방식: HTTP POST Redirection (IDP->사용자->SP)

3) SP

③ IDP로부터 받은 정보 중에 매개정보 m으로 SP에서의 사용자를 식별해서 사용자의 IDP와 사용자 동의정보 a를 확인하고 h′(=H(m||A||n||a))를 계산하여 h와 비교

④ h와 h′가 같으면 동적신뢰를 기반으로 한 IDP 인증과 전송정보들의 무결성이 확인 됐다고 판단하여 SP 인증쿠키를 발행하고 IDP에 웹 Redirection하여 정상완료 통보

- 전송방식: HTTP POST Redirection (SP->사용자->IDP)

⑤ h와 h′가 다르면 오류상황을 IDP에 전송

- 전송방식: HTTP POST Redirection (SP->사용자->IDP)

4) IDP

⑥ 정상적인 통보를 받으면 SSPG의 모든 SP에 대해서 위의 과정을 반복하고, 오류 통보를 받으면 해당상황을 사용자에게 알림

3.4.6 동적신뢰 해제와 사용자 동의정보 a의 변경

동적신뢰 해제는 사용자가 IDP를 통한 특정 SP의 ID정보 관리가 필요 없다고 판단할 때 요청하게 된다. IDP와 특정 SP 간에 동적신뢰가 해제되면 IDP는 SP를 MSPG에서 제거하고, SP는 사용자의 동의정

보 a와 IDP 정보를 삭제해야 한다. 동적신뢰 해제 프로토콜은 IDP에서 SP로 전달되는 정보의 내용만 인증확인에서 동적신뢰 해제로 바뀔 뿐 기본적으로 SSO 프로토콜과 같다. 동적신뢰 해제가 정상적으로 이루어지면 해당 SP는 사용자 동의정보 a를 삭제하는 것이 원칙이지만 악의적인 SP의 경우 a를 삭제하지 않을 가능성이 있다. 이런 가능성에 대비해서 사용자가 새로운 MSPG를 대상으로 동의정보 a를 변경할 수 있어야 한다. 동의정보 a의 변경은 일반 ID정보 변경 프로토콜에서 변경되는 내역을 C에서 새로운 a′로 바꿔주면 된다. 일반 ID정보 변경 프로토콜이 변경되는 정보에 대한 비밀성과 무결성을 보장하기 때문에 동의정보 a의 변경에도 적용할 수 있는 것이다.

IV. 모델분석과 고려사항

본장에서는 U2IM을 사용자 편리성, 안전성, 효율성 및 적용성의 관점에서 기존 ID정보 관리 모델들과 비교/분석을 하고 이를 통해 U2IM의 제안 타당성을 입증한다.

4.1 사용자 편리성

사용자 편리성은 ID정보 관리 모델이 얼마나 사용자의 요구사항을 반영하고 있는지와 직접적으로 관련이 있다. [표 1]은 사용자 편리성을 ID정보가 관리되는 SP그룹의 형성방식과 IDP가 제공하는 ID정보 관리 기능으로 나누어 U2IM과 기존의 모델들을 비교/분석한 것이다. SP그룹 형성방식 관점에서 보면 기존 모델들은 모두 사업자들 간의 사업협약에 기반 하거나 혹은 사업협약이 없더라도 사업자의 결정에 의해서 형성되는 것으로 사용자의 요구와는 무관하다. 이에 반해서 U2IM은 사용자의 요구에 의해서 SP그룹이 형성되기 때문에 사용자의 요구를 정확하게 반영한다고 할 수 있다. IDP가 제공하는 ID정보 관리 기능에 있어서도 U2IM만이 모든 기능을 전부 제공한다. U2IM에서 의미하는 사용자 중심 개념은 이렇게 사용자의 요구사항을 완전하게 수용하는 모델이라는 의미이다.

4.2 안전성 분석과 고려사항

안전성 분석은 ID정보 관리 모델의 각 구성요소 간의 정보 전달이 정보보호의 기본 요소인 인증, 비밀성, 무결성 등을 만족하는지의 관점에서 이루어져야 한다[3]. U2IM에서 각 구성요소 간의 정보 흐름은 사용자〈--〉IDP, 사용자〈--〉SP, IDP〈--〉SP의 3가지 구간에서 존재한다. 이중 사용자〈--〉IDP, 사용자〈--〉SP 구간은 사용자 웹브라우저와 IDP/SP 웹서버 구간에 해당하는데 이 구간은 현재 전 세계적으로 HTTPS의 사용이 표준으로 되어 있어 U2IM에서도 이 구간은 HTTPS을 사용하여 인증, 비밀성, 무결성 등의 필요한 안전성을 확보한다. 따라서 U2IM 프로토콜의 안전성분석은 IDP〈--〉SP 구간에서 주요 기능 단계 별로 하였으며 이 구간에서 HTTPS 사용과 관련하여 추가로 언급하였다.

4.2.1 동적신뢰를 통한 MSPG 형성

IDP와 MSPG의 SP간의 동적신뢰는 HTTPS라는 안전한 채널을 통해 사용자로부터 직접 동의정보 a를 제공받은 IDP와 SP가 a의 보유 여부를 상호 확인함으로써 이루어진다. 사용자 동의정보 a의 상호 확인은 결국 사용자의 동의를 상호 확인하는 의미이기 때문에 프라이버시 보호의 기본 특성[1]을 만족하는 것이며 이를 프로토콜에 반영한 효과를 갖게 된다.

[표 1] 사용자 편리성 관점의 비교/분석

구분	SP그룹 형성방식	ID정보 관리 기능			
		id연계	일반 ID정보 제공	일반 ID정보 변경	SSO
.NET Passport	MS와 SP간의 사업협약	×	○	×	○
Liberty Alliance	IDP와 SP간의 사업협약	○	×	×	○
OpenID	IDP와 SP의 선택	×	×	×	○
CardSpace	기존 관리 모델 수용/통합	○	○	×	○
U2IM	사용자 요구에 의한 IDP와 SP간의 그룹 형성	○	○	○	○

○:제공, ×: 없음

1) DH 방식 취약점 보완

DH 방식에 대한 중간자 공격 취약점의 본질은 키설정을 위해 상호 교환하는 공개 정보에 대한 정보 인증이 없다는 것인데 U2IM에서는 이를 a 보유 여부의 확인으로 해결한다. 즉 IDP는 DH를 위한 공개정보 u1과 a를 해쉬한 h(=H(u1||a))를 u1과 함께 보내고, SP는 수신 받은 u1과 사용자로부터 받은 a로 h′(=H(u1||a))를 계산하여 h와 h′를 비교하면 IDP의 a 보유 여부 확인과 함께 u1에 대한 정보인증이 가능하게 된다. 공격자인 중간자는 동의정보 a를 모르기 때문에 u1을 변조한다 하더라도 정확한 해쉬값을 계산할 수 없다[8]. u2의 경우도 마찬가지로 적용된다.

2) 재사용 공격

동적신뢰 과정에서 nonce 정보나 Time Stamp 등의 정보를 사용하지 않았기 때문에 동의정보 a를 모르는 악의적인 중간자가 키설정을 위해서 IDP와 SP 간에 전송되는 정보들을 가로채 재사용할 수 있다. 그러나 동적신뢰 과정에서의 재사용 공격은 공격자가 세션키 설정을 위한 비밀정보를 알 수 없어 결과적으로 세션키 설정을 할 수 없기 때문에 무의미하다.

3) IDP 가장 공격

MSPG에 속한 특정 SP가 동의정보 a를 알고 있는 것을 이용하여 사용자 요청과 무관하게 IDP를 가장하여 MSPG에 속하지 않은 다른 SP에 동적신뢰 요청을 할 수가 있다. 그러나 이 경우에는 사용자의 실질적인 요청이 없는 상태이므로 다른 SP가 사용자로부터 동의정보 a를 받을 수 없기 때문에 공격 SP는 동적신뢰를 형성할 수 없다.

4.2.2 id연계와 일반 ID정보 제공

IDP에서 SP로 전달되는 매개정보 m과 일반 ID정보 G의 비밀성과 무결성 보장은 m, G와 해쉬값 h(=H(m||G)) 전체를 세션키 k로 암호화(M(=E_k(m||G||h))하여 전송함으로써 가능하다. 공격자는 세션키 k를 알 수 없기 때문에 M의 내용을 알 수 없으며 m이나 G의 내용을 변조한다 하더라도 해쉬값 h의 검증을 통해서 변조여부를 확인할 수 있다.

4.2.3 일반 ID정보와 사용자 동의정보 a의 변경

일반 ID정보와 사용자 동의정보 a의 변경 프로토콜은 세션키 설정 과정과 변경되는 일반 ID정보 전송 과정으로 나누어진다. 후자의 안전성은 id연계와 일반 ID정보 제공의 안전성과 같다. 그러나 세션키 설정 과정의 안전성 분석은 앞에서 설명한 동적신뢰를 통한 MSPG 형성 시의 세션키 설정 과정과는 약간 다르다. 동적신뢰 과정에서 SP는 동의정보 a를 사용자로부터 직접 받았지만 일반 ID정보와 a 변경 과정에서 SP는 이미 저장 돼 있는 사용자 a 정보를 토대로 세션키 설정을 해야 하기 때문에 a 정보의 검색을 위해서 세션키 설정 과정에서 IDP가 SP로 보내는 매개정보 m은 암호화하면 안 된다.

1) 매개정보 m의 무결성

매개정보 m은 그자체로는 의미 없는 문자열이기 때문에 비밀성은 그다지 중요하지 않으며 공격자가 m을 알게 되더라도 사용자에게 큰 위협이 되지 않는다. 그러나 m의 무결성은 정확한 사용자의 식별과 관련되기 때문에 매우 중요하다. IDP가 m과 해쉬값 h(=H(u1||m||a)를 SP에 전송하면 SP는 m을 통해서 사용자의 동의정보 a를 알아낸 후, h와 h′(=H(u1||m||a)의 비교를 통해서 m의 무결성을 확인할 수 있다. 공격자가 m을 갈취해서 변조를 한다 하더라도 동의정보 a 값을 모르기 때문에 SP가 검증할 수 있는 정확한 h를 계산할 수 없다.

2) IDP 가장 공격

MSPG 내의 특정 SP가 사용자의 동의정보 a를 공유하는 것을 이용하여 다른 SP에게 사용자의 요구와 상관없이 일반 ID정보 변경을 요구하는 경우를 말한다. 그러나 동일 사용자의 IDP와 각 SP 간의 id연계를 위한 매개정보 m이 모두 다르기 때문에 가장 공격이 성립하지 않는다. 즉 공격 SP는 동의정보 a를 알고 있는 사용자의 다른 SP에서의 매개정보 m를 알 수가 없기 때문에 정확한 해쉬값 h(=H(u1||m||a))를 계산할 수 없다.

4.2.4 SSO와 동적신뢰 해제

SSO와 동적신뢰 해제 프로토콜에서는 IDP에서 SP로 보내는 정보의 인증 즉 무결성 보장과 재사용 공격에 대한 분석이 필요하다. 이 단계에서 정보의 비밀성은 큰 문제가 되지 않는다. 즉 공격자가 정보의 내용을 안다 하더라도 특별한 위협이 되지 못한다.

1) 정보의 무결성

정보의 무결성 보장은 사용자 동의정보 a와 무결성 보장이 필요한 정보들을 해쉬하여 전송함으로써 가능하다. SSO를 예로 들면 해쉬값 h(=H(m||A||n||a))를 통해서 m과 A의 무결성이 보장되는 것이다. 즉 공격자는 a를 모르기 때문에 m이나 A를 변조했다 하더라도 정확한 h'값을 계산할 수 없다. 또한 특정 사용자의 동의정보 a를 알고 있는 MSPG 내의 SP에 의한 IDP 가장 공격도 해당 사용자의 매개정보를 알 수 없기 때문에 성립하지 않는다.

2) 재사용 공격

SSO는 인증에 관련된 기능이기 때문에 재사용 공격에 대한 방어가 필요하다. 즉 IDP가 SP에게 보내는 정당한 인증확인 정보를 중간에 가로채서 재사용하면 SSPG의 SP들의 인증쿠키를 공격자가 획득할 가능성이 있다. 이를 위해 재사용할 수 없는 nonce 정보 n를 추가해서 전송하면 재사용 공격을 막을 수 있다.

4.2.5 HTTPS 사용과 관련한 고려사항

앞에서도 언급했듯이 인터넷에서 사용자 웹브라우저와 IDP/SP 웹서버 간의 신원/인증 ID정보 전송 시에 HTTPS의 사용은 국제 표준 의무사항이기 때문에 이 인프라를 그대로 IDP<-->SP 구간에서 사용하는 것을 고려할 수도 있다. 그러나 사용자<-->IDP, 사용자<-->SP 구간에서 HTTPS가 적용될 때는 IDP/SP 웹서버 인증서만 사용해도 되지만 IDP<-->SP 구간에서는 IDP와 SP간에 상호 인증을 해야 하기 때문에 양쪽의 인증서가 모두 필요하다. 따라서 인증서 관리와 함께 공개키 사용의 부담이 생긴다. 또한 동적신뢰 형성 시나 ID정보 관리 기능에서 사용자의 동의 여부, 즉 프라이버시 특성을 시스템적으로 반영하는 사용자 동의정보 a의 적용이 불가능해진다. 상대적으로 무거운 HTTPS를 사용하지 않고도 결합하여 충분한 안전성을 보장하면서 동시에 효율성도 높일 수 있기 때문에 IDP<-->SP 구간에서 HTTPS의 사용을 고려하지 않았다.

4.3 효율성

ID정보 관리 모델의 효율성은 주로 안전성 확보를 위해 사용하는 암호기법에 의해 좌우된다. ID정보 관리모델은 속도에 민감한 인터넷 환경에 적용되는 것이기 때문에 안전성을 너무 고려한 무거운 암호기법의 사용은 바람직하지 않다. 정보의 인증, 비밀성 및 무결성을 위해 사용하는 암호기법으로는 크게 해쉬, 대칭키, 공개키 3종류가 있는데 일반적으로 해쉬가 가장 빠르고 다음이 대칭키, 가장 느린 것이 공개키이다[13]. 해쉬와 대칭키의 속도 차이는 크게 없으나 대칭키와 공개키의 속도 차이는 상당히 크다. 물론 타원곡선 등의 개선된 공개키 방식이 등장했지만 속도가 우선 시 되는 환경에서는 되도록이면 공개키 방식은 배제해야 한다. 이러한 점을 고려해 U2IM은 정보의 인증을 위해 사용자 동의정보 a, 비밀성을 위해 대칭키, 무결성을 위해 해쉬를 사용함으로써 최대한 경량화된 암호기법으로만 구성되었다. 또한 대칭키 생성을 위해서 a와 동적인 DH 키설정 방식을 결합함으로써 따로 키 관리를 할 필요가 없다. 따라서 U2IM 프로토콜은 필요한 안전성은 확보하면서 최대한 효율성이 높은 구조라고 할 수 있다.

4.3.1 기존 ID정보 관리 모델과의 비교/분석

[표 2]는 U2IM과 기존의 ID정보 관리 모델들이 정보보호의 기본 요소를 만족 시키는데 사용하는 암호기법을 비교/분석한 것이다.

[표 2] 암호기법 비교/분석

구분	무결성	비밀성	인증	키관리
.NET Passport	없음	대칭키	대칭키	대칭키 관리
Liberty Alliance (PKI)	디지털 서명	공개키	디지털 서명	인증서 관리
OpenID (DH)	해쉬	없음	없음	없음
CardSpace (SAML)	디지털 서명	공개키	디지털 서명	공개키 관리
U2IM	해쉬	대칭키	동의정보	없음

.NET Passport의 경우 MS의 서비스 형태이기 때문에 MS가 제공하는 대칭키로 인증과 비밀성을 보장하고 무결성 보장은 별도로 정의 돼 있지 않다. Liberty Alliance나 CardSpace는 특정 암호기법으로 한정되는 모델은 아니지만 Liberty Alliance의 표준 규격이나 CardSpace 관련 문서에서 적용 예로서 많이 언급한 PKI[5]와 SAML(Security Assertion Markup Language)[8]의 예로 비교

하였는데 모두 무거운 인증서와 공개키 기술을 사용하는 것을 알 수 있다. OpenID는 해쉬를 이용한 메시지 인증으로 무결성을 해결하는데 메시지 인증을 위한 비밀키 전달을 IDP와 SP 간의 DH 키설정으로 형성된 세션키로 암호화 해서 전달한다. 그러나 앞서 언급했듯이 DH는 중간자 공격에 취약한데 이에 대한 보완책으로 OpenID에서는 HTTPS를 고려할 수 있다고만 언급할 뿐[9] 구체적인 규격을 제시하지 못하고 있다. OpenID를 신뢰성이 보장된 관리모델로 보기 힘든 이유가 이 때문이다. 결국 U2IM은 기존 사례들의 이런 문제점을 효율성과 안전성의 조화를 이룬 방법으로 해결한 것이다.

4.3.2 HTTPS 인증과의 비교/분석

[표 3]은 U2IM의 ID정보 관리 기능과 현재 널리 사용되고 있는 id/password HTTPS 보안 인증에서 사용되는 암호기법을 빈도수 관점에서 비교/분석한 것으로서 각 기능 별로 사용되는 암호기법의 빈도수와 기능 자체가 사용되는 빈도수를 표시하였다. [표 3]을 보면 알 수 있듯이 U2IM의 ID정보 관리 기능에서 사용되는 암호기법의 빈도수는 HTTPS 보안 인증에서의 암호기법에 비해서 해쉬/대칭키 빈도수가 비슷하거나 적은 것을 알 수 있다. 더구나 HTTPS 보안 인증은 서버 인증서의 확인을 위해서 공개키를 사용하는 반면 U2IM에서는 이를 동의정보 a의 사용으로 해결하기 때문에 훨씬 효율적이다.

4.4 적용성

적용성이란 ID정보 관리 모델의 실제 인터넷 환경에서의 적용 가능성을 의미한다. 적용성은 인터넷 환경을 움직이는 3개의 주체들, 즉 정부, 사업자, 사용자 관점에서 파악해야 하는데 결론적으로 현재의 인터넷 환경에서는 3개의 주체들이 모두 ID정보 관리 모델의 필요성을 갖고 있다고 볼 수 있다. 정부 입장에서는 ID정보의 관리 부실로 인한 프라이버시 문제가 심각해지면서 프라이버시 보호를 위한 더 나은 인터넷 서비스 환경을 필요로 하고 있고, 사업자 입장에서는 ID정보의 중요성이 증가하면서 이를 사업모델로의 이용 가능성 (IDP)과 아예 ID정보 관리 부담으로부터 벗어나는 방법(SP)을 필요로 할 수 있다. 사용자 입장에서는 여러 SP에 독립적으로 중복/분산 돼 있는 ID정보로 인한 서비스 이용의 불편함과 프라이버시 침해의 우려를 제거할 필요가 있다. ID정보 관리 모델의 적용성은 이러한 인터넷 정보 주체들의 필요성을 만족 시키는 정도에 의해 결정된다. 즉 정부 입장에서는 프라이버시 보호, 사업자 입장에서는 모델 도입 시의 구현 용이성, 사용자 입장에서는 정부 입장과 중복되는 프라이버시 보호와 서비스 이용의 편리성 관점에서 만족하는 수준이 적용성의 주요 결정 요인이 된다고 볼 수 있다. [표 4]는 앞에서 언급한 적용성과 관련한 요소들에 대해 U2IM과 기존의 사례들이 만족하는 수준을 비교/분석하였다. 프라이버시 보호 측면은 3가지 요인으로 구분하였다. ID정보 암호화 전송은 정보가 유출되더라도 정보의 비밀성 보장을 위한 것이며, ID정보 사용자 통제는 ID정보가 SP에 전송될 때 사용자의 동의가 시스템적으로 반영되는 가를 의미하고, SP 내역 사용자 제어는 ID정보가 제공되는 SP 내역을 사용자가 파악하고 관리할 수 있는지를 의미한다. 프라이버시 보호와 관련해서 SP들의 ID정보 관리 문제는 사실 SP들의 관리 능력과 책임성에 관련된 부분이지만 사용자가 자신의 ID정보를 저장하고 있는 SP들을 정확히 파악하고 언제든지 이를 회수할 수 있는 구조에서는 SP들의 관리 책임성이 높아질 가능성이 크기 때문에 프라이버시 보호와 관련된 요인으로

[표 3] U2IM의 ID정보 관리 기능과 HTTPS 인증의 암호기법 빈도수 비교/분석

구분	해쉬	대칭키	공개키	사용빈도
동적신뢰 형성	2	0	0	MSPG 형성 시 1회
id연계/일반 ID정보 제공	2	1	0	MSPG 형성 시 1회
일반 ID정보/a 정보 변경	2	1	0	일반 ID정보/a 정보 변경 시 MSPG의 SP 수
SSO/동적신뢰 해제	2	0	0	SSO는 SSPG의 SP 수/동적신뢰 해제 시 1회
HTTPS 인증	2	1	1	id/password 인증 시 1회

〔표 4〕 적용성 비교/분석

구분		.NET Passport	Liberty Alliance	OpenID	CardSpace	U2IM
프라이버시 보호	ID정보 암호화 전송	○	×	×	○	○
	ID정보 사용자 통제	○	×	×	○	○
	SP 내역 사용자 제어	×	×	×	×	○
구현 용이성	경량화 암호기법 사용	○	×	○	×	○
	오픈 소스 가능성	×	○	○	○	○
사용자 편리성		△	△	△	△	○

○:높음, △: 보통, ×: 낮음

파악하였다. 구현 용이성과 관련해서는 사용되는 암호기법이 경량화 되어야 하고, 오픈 소스들이 많이 존재하는 것이 중요하다. 사용자 편리성은 사용자의 요구가 얼마나 관리 모델에 반영되는 구조인가의 여부이다. 〔표 4〕에서 알 수 있듯이 U2IM은 적용성과 관련된 모든 요인을 만족함으로써 3개 주체들의 필요성에 가장 많이 부합하기 때문에 적용성이 가장 높다고 볼 수 있다.

V. 결 론

본 논문에서는 인터넷 서비스 패러다임이 사업자 중심에서 사용자 중심의 웹2.0으로 바뀌어 가는 흐름 속에서 사용자의 실체와 관련된 정보인 ID정보를 사용자가 실질적으로 중심이 되어 관리할 수 있는 모델인 U2IM을 제안하였다. U2IM에서는 사용자의 요청에 의해서 형성된 동적신뢰 그룹인 MSPG을 대상으로 ID정보 관리의 모든 기능이 제공되기 때문에 완전한 사용자 중심의 ID정보 관리 모델이라고 할 수 있다. U2IM은 기존의 ID정보 관리 모델이 인터넷 사업자들이 중심이 되어 형성된 신뢰그룹을 대상으로 했기 때문에 사용자의 실질적인 요구가 반영되지 않는 점과 ID정보 관리 기능이 SSO 등으로만 한정되어 사용자의 기대수준에 못 미치는 점 등의 한계를 극복하였다. 또한 사용자의 동의정보 a를 이용하여 프라이버시 특성을 시스템적으로 반영하였으며 이와 함께 해쉬와 대칭키 등의 경량화된 암호기법만으로 필요한 안전성을 확보하여 속도에 민감한 인터넷 환경에의 적용성과 효율성을 높였다. 이러한 장점들은 정부, 사업자, 사용자들의 필요성을 유발시켜 정부의 정책으로 반영되거나 Open Source 그룹에서의 U2IM 구현 동기를 촉발 시켜 사업자들의 자발적 적용을 유도할 수 있다. 결과적으로 U2IM은 사용자의 인터넷 이용 편리성과 프라이버시 보호 측면에서 발전된 결과를 도출하였다. 인터넷 환경은 이제 사용자가 중심이 되는 세상으로 바뀌고 있으며 이런 상황에서 필요한 것이 바로 안전성과 효율성의 필요 수준을 만족하면서 사용자의 이용 편리성과 프라이버시 보호가 보장되는 ID정보 관리 모델의 적용이다. U2IM은 인터넷 환경의 이런 변화의 흐름에 부합하는 ID정보 관리 모델이라고 할 수 있다. 앞으로는 인터넷 환경이 Pervasive 환경으로 변화할 것으로 예상되고 있으며 이미 이러한 변화의 사례들이 속속 나타나고 있다. Pervasive 환경에서도 유선 인터넷 환경과 마찬가지로 ID정보 관리가 매우 중요하다. Pervasive 환경에서는 단말기 컴퓨팅 능력과 정보보호의 취약점이 유선환경 보다 훨씬 열악하기 때문에 사용자 편리성과 프라이버시 보호와 함께 효율성과 안전성의 조화에 대한 고려가 더욱 필요하다. 따라서 Pervasive 환경에 맞는 ID정보 관리 모델의 연구가 계속 되어야 한다고 본다.

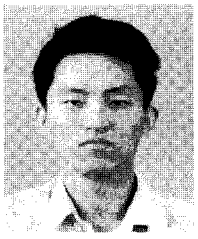
참 고 문 헌

〔1〕 Abhilasha Bhargav-Spantzel, "Establishing and Protection Digital Identity in Federation Systems," Journal of Computer Security, vol. 14, no. 3, pp. 269-300, Nov. 2005.

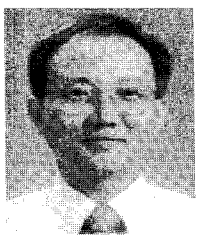
〔2〕 D.W. Shin, "Ensuring Information Assurance in Federated Identity Management," Performance, Computing, and Communications 2004 IEEE International Conference on, pp. 821-826, Apr. 2004.

[3] D.R. Stinson, "CRYPTOGRAPHY," CRC Press, 1995.
[4] K. Bhargavan, "Secure Session for Web Services," ACM Workshop on Secure Web Services, pp. 56-66, Oct. 2004.
[5] "Liberty Alliance Project: Introduction to the Liberty Alliance Identity Architecture," 2003.
[6] "Liberty Alliance Project: Liberty Trust Models Guidelines," 2003.
[7] Microsoft, "Microsoft.NET Passport," 2004.
[8] Microsoft, http://msdn.microsoft.com/en-us/library/aa480189.aspx, 2008.
[9] "OpenID Authentication 1.1," 2006.
[10] T. Tsiakis and G. Sthephanides, "The concept of security and trust in electronic payments," Computers & Security, vol. 24, no. 1, pp. 10-15, Feb. 2005.
[11] V. Samar, "Single Sigh-On Using Cookies for Web Applications," Proceedings of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises, pp. 158-163, June 1999.
[12] W. Diffie and M.E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, Nov. 1976.
[13] W. Stallings, CRYPTOGRAPHY AND NETWORK SECURITY, Prentice Hall, 1999.
[14] W.G. Shieh, "Efficient remote mutual authentication and key agreement," Computers & Security, vol. 25, no. 1, pp. 72-77, Feb. 2006.

〈著者紹介〉



이 해 규 (Haegyu Rhy) 정회원
1989년 2월: 서울대학교 컴퓨터공학과 학사
1991년 2월: 서울대학교 컴퓨터공학과 석사
현재: 서울대학교 컴퓨터공학과 박사과정, KT 책임연구원
〈관심분야〉 인증, ID관리, 정보보호



신 현 식 (Hyeonshik Shin) 정회원
1973년: 서울대학교 응용물리학과 학사
1980년: 미국 텍사스 대학교 의공학과 석사
1985년: 미국 텍사스 대학교 전기/컴퓨터공학과 박사
1986년~현재: 서울대학교 컴퓨터공학부 교수
〈관심분야〉 실시간 계산, 분산 시스템, 모바일 컴퓨팅