논문 2010-47TC-1-12

커뮤니티 기반의 유비쿼터스 네트워크 환경에서 안전한 커뮤니티 생성 권한 위임 방안

(Authority Delegation Scheme for Secure Social Community Creation in Community-Based Ubiquitous Networks)

노 효 선*, 정 수 환**

(Hyosun Roh and Souhwan Jung)

요 약

본 논문은 커뮤니티 기반의 유비쿼터스 네트워크에서 대리 서명 기법을 적용한 커뮤니티 멤버들 간의 상호 인증 기법 및 안전한 커뮤니티 생성 권한 위임 기법을 제안한다. 커뮤니티 기반의 유비쿼터스 네트워크에서는 사용자가 필요로 하는 네트워크 서비스 및 응용서비스를 제공해주기 위해 다양한 상황인식 정보가 수집 되고 활용된다. 또한 사용자는 학업, 게임, 정보 공유, 사업, 회의 등의 다양한 목적에 의해 다양한 커뮤니티를 생성할 수 있다. 그러나 커뮤니티 기반의 유비쿼터스 네트워크의 경우 공격자에 의해 상황인식 정보가 위조 및 도용되어 악의적인 목적으로 사용될 수 있다. 또한 악의적인 목적으로 커뮤니티를 생성할 수 있다. 제안하는 기법은 위임 서명 기법을 이용하여 커뮤니티 멤버들 간의 상호 인증 및 비밀 키를 교환 할 수 있도록 지원하고, 안전하게 커뮤니티 생성 권한을 위임할 수 있도록 하였다. 또한 기존 RSA 서명 기법과의 비교 분석을 통해 전체 계산 시간이 감소함을 확인하였다.

Abstract

This paper proposes authority delegation for secure social community creation and mutual authentication scheme between the community members using proxy signature in community-based ubiquitous networks. In community-based ubiquitous network, User's context-awareness information is collected and used to provide context-awareness network service and application service for someone who need it. For the many reason, i.e. study, game, information sharing, business and conference, social community could be created by members of a social group. However, in community-based ubiquitous network, this kind of the context-awareness information could be abused and created by a malicious nodes for attack the community. Also, forgery community could be built up to attack the community members. The proposed scheme using the proxy signature provides a mutual authentication and secure secret key exchange between community members, and supports secure authority delegation that can creates social community. Also, when delegation of signing authority and mutual authentication, this scheme reduces total computation time compared to the RSA signature scheme.

Keywords: 커뮤니티, 유비쿼터스, 대리 서명, Diffie-Hellman, Ad-hoc

I. 서 론

유비쿼터스 네트워크 환경이 점차 다양한 사회 영역으로 확대되면서 사용자에게 지능적인 서비스를 제공해 주기 위한 기술 연구가 진행되고 있으며, 그중 사용자의 상황인식 정보를 수집 및 분석하여 필요한 응용 서비스 또는 네트워크 서비스를 제공하기 위한 연구가 활

* 정회원, ** 평생회원, 숭실대학교 정보통신전자공학부
(School of electronic Engineering, Soongsil University)

※ 본 연구는 지식경제 프론티어 기술개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅 및네트워크원천기반기술개발사업의 지원에 의한 것임

접수일자: 2009년8월10일, 수정완료일: 2010년1월18일

발하게 진행되고 있다. 또한 유비쿼터스 네트워크 환경은 사용자가 필요로 하는 응용 서비스 및 네트워크 서비스를 제공하기 위해 다양한 무선 및 유선 네트워크 환경이 공존한다. 때문에 사용자가 언제, 어디서든 네트워크에 접속하여 필요한 응용 서비스 및 네트워크 서비스를 사용할 수 있도록 지원하기 위한 지능적인 네트워크 기술이 개발되고 있으며, 이와 같은 기술로는 최근 연구되고 있는 커뮤니티 기반의 유비쿼터스 네트워크가 있다[1].

커뮤니티 기반의 유비쿼터스 네트워크는 사용자가 통신을 위해 사용하는 단말의 네트워크 상황인식 정보를 활용하여 사용자에게 최적의 네트워크 환경과 필요한 응용서비스를 제공할 수 있도록 지원한다. 이를 위해 커뮤니티 기반의 유비쿼터스 네트워크에서는 존 마스터 노드를 기반으로 커뮤니티를 형성하고, 커뮤니티에 따라 필요한 응용 서비스 및 네트워크 서비스를 제공한다. 또한 커뮤니티에 존재하는 사용자들의 네트워크 상황인식 정보 공유 및 서비스 실행 모듈, 네트워크 서비스 실행 모듈을 전달하기 위해 스마트 패킷 프로토콜이 사용된다[2]. 이와 같은 네트워크 환경으로 구성되는 커뮤니티 기반의 유비쿼터스 네트워크에서는 사용자들이 필요에 따라 자유롭게 커뮤니티를 형성할 수 있다. 물리적으로 동일한 커뮤니티에 존재하는 사용자들은 언제든지 학업, 게임, 정보 공유, 사업, 회의 그리고 네트워크에 연결된 장치를 공유하여 사용하는 등 다양한 목적으로 자유롭게 커뮤니티를 형성하고 취소할 수 있다. 때문에 비정상적인 사용자들 또한 자유롭게 악의적인 목적으로 커뮤니티를 생성하여 정상적인 사용자들의 네트워크 사용을 방해하거나, 커뮤니티 기반의 유비쿼터스 네트워크 전체를 혼란스럽게 할 수 있다. 뿐만 아니라 커뮤니티에 속한 사용자들은 ad hoc[3]으로 직접적인 통신을 하기 때문에 악의적인 공격자는 언제든지 정상적인 사용자들의 통신을 방해할 수 있으며, 정상적인 사용자처럼 가장하여 다양한 공격을 시도할 수 있다[4]. 따라서 이러한 악의적인 공격자로부터 커뮤니티에 존재하는 사용자들을 보호하고, 안전한 통신을 보장할 수 있는 보안 기술이 필요하다. 또한 사용자의 필요에 의해 커뮤니티가 생성될 때 커뮤니티를 생성하는 커뮤니티 마스터를 인증하고, 생성되는 커뮤니티의 안전성 및 신뢰성을 제공하기 위한 보안 기술이 필요하다.

본 논문에서는 위와 같은 문제를 해결하기 위해 기존에 활발하게 연구된 대리서명 기법 (Proxy Signature)[5]을 적용하여 커뮤니티에 존재하는 사용자들 간의 상호 인증 및 비밀 키 교환 그리고 안전한 커뮤니티 생성을 지원할 수 있는 커뮤니티 생성 권한 위임 방법을 제안하였다. 제안하는 기법은 인증/등록 서버의 서명 생성 권한을 인증된 존 마스터에게 위임하여 인증/등록 서버가 생성하는 서명 정보를 대신 생성할 수 있도록 하였다. 서명 권한을 부여 받은 존 마스터는 자신의 커뮤니티에 존재하는 각 커뮤니티 멤버들을 위해 인증/등록 서버를 대신하여 서명을 생성 및 발급해주고, 각 커뮤니티 멤버들은 이 서명을 상호 인증하는 과정에서 사용한다. 서명 정보에는 각 커뮤니티 멤버들의 Diffie-Hellman[6] 공개 값을 포함시켜 상호 인증하는 과정에서 안전하게 비밀 키를 교환할 수 있도록 제안하였다. 또한 커뮤니티 멤버가 커뮤니티를 생성하기 원할 경우 인증/등록 서버를 대신하여 존 마스터가 커뮤니티 생성 권한을 커뮤니티 멤버에게 부여할 수 있도록 하였고, 커뮤니티 생성 권한에 대해 커뮤니티에 가입을 원하는 커뮤니티 멤버들은 누구나 검증할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 관련 기술들을 설명하고, Ⅲ장에서는 본 논문에서 제안하는 기법에 대해서 설명한다. 그리고 Ⅳ장에서는 제안하는 기법에 대한 안전성 분석 및 효율성을 비교하고, 마지막 Ⅴ장에서 결론을 맺는다.

Ⅱ. 관련 기술

다음은 커뮤니티 기반의 유비쿼터스 네트워크에 대

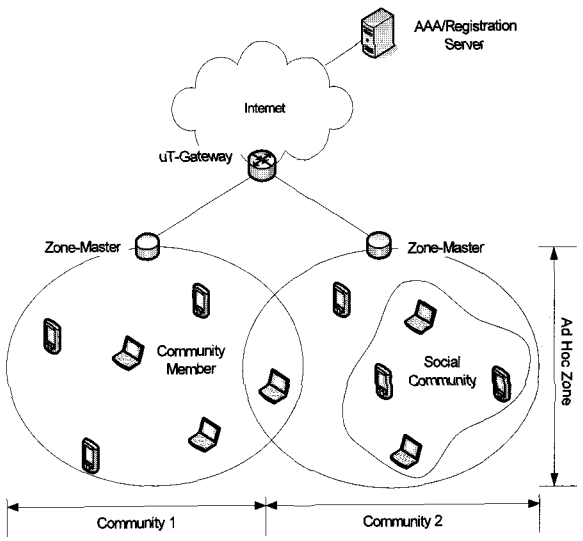

그림 1. 커뮤니티 기반의 유비쿼터스 네트워크 구조
Fig. 1. Community-Based Ubiquitous Network Architecture.

한 간략한 설명과 기존 서명 기법에 대해서 설명한다.

1. 커뮤니티 기반의 유비쿼터스 네트워크

커뮤니티 기반의 유비쿼터스 네트워크는 그림 1과 같이 외부 네트워크와의 연결을 지원하는 uT-Gateway, 커뮤니티 구성을 위해 사용되는 존 마스터, 그리고 커뮤니티 멤버 노드들인 사용자 단말들로 구성된다. 존 마스터를 기반으로 생성되는 커뮤니티는 커뮤니티 성격에 따라 여러 개의 존 마스터가 그룹을 이루어 하나의 커뮤니티로 구성될 수 있으며, 하나의 존 마스터가 관리하는 커뮤니티로도 구성할 수 있다. 커뮤니티에 속한 이동 단말들은 ad hoc 라우팅 프로토콜을 이용하여 단말들 간에 직접 통신하며, 커뮤니티 네트워크 상황에 따라 pro-active 및 re-active 라우팅 프로토콜[7]이 사용된다. 따라서 이동 단말들은 커뮤니티가 변경될 경우 해당 커뮤니티의 네트워크 환경에 따라 네트워크 설정을 자동적으로 변경해야 하고, 환경에 맞는 라우팅 프로토콜을 사용해야 한다. 이를 지원하기 위해 커뮤니티에 속한 존 마스터 및 이동 단말들에는 스마트 패킷 에이전트가 사전에 설치되고, 스마트 패킷 에이전트들 간에 주고받는 스마트 패킷을 통해 커뮤니티 상태 정보 및 필요한 네트워크 서비스 모듈 등을 제공받아 추가적인 동작 없이 자동으로 실행 및 삭제할 수 있도록 지원한다. 또한 존 마스터가 관리하는 물리적인 커뮤니티에 속한 이동 노드들은 커뮤니티 멤버의 필요에 따라 자유롭게 커뮤니티를 생성 및 취소할 수 있다.

2. 기존 서명 기법과 대리서명 기법

디지털 서명은 서명자의 인증과 전달되는 데이터의 변경여부 확인 및 서명 정보를 생성하여 전달한 서명자의 부인 방지 등을 제공하는 기술이다. 이러한 디지털 서명 방식은 서명하는 서명자에 따라 간접서명 방식 과 직접서명 방식으로 분류된다. 간접서명 방식은 서명자와 서명을 검증하는 사용자 사이에 제 3의 중재 노드가 개입하여 서명의 정당성을 제공해주는 방식이고, 직접 서명 방식은 서명자의 서명을 수신자가 직접 수신하여 서명의 정당성을 확인하는 방법이다. 1976년 Diffie와 Hellman이 개발한 공개키 암호방식 기반의 디지털 서명 방식이 제안된 이후 다양한 디지털 서명 기법이 제안되었다. 대표적인 디지털 서명 기법으로는 RSA[8], ElGamal[9], Schnorr[10], Nyberg-Rueppel[11], DSS[12] 등이 있다. 다음은 이러한 대표적인 서명 기술 중 다양한 제안 기법에서 적용되고 있는 RSA와 Schnorr 서명 기법에 대해 설명한다.

가. 기존 서명 기법

RSA 서명 기법은 우선 RSA 기반의 공개키 기반구조를 가정하며 각 사용자는 공개키 (e,n)와 개인키 (p,q,d)를 가진다.

- 서명 과정: 서명자 A는 서명할 메시지 m에 대해서 잘 알려진 일방향 해쉬 함수를 이용하여 $H=h(m)$을 계산하고, 서명자의 개인키 (p_A,q_A,d_A)를 이용하여 다음과 같이 서명을 생성하여 서명 s와 m을 수신자인 B에게 전달한다.

$$s = H^{d_A} \bmod n_A \tag{1}$$

- 검증 과정: 서명자로부터 서명과 메시지를 수신한 B는 서명자와 동일한 일방향 해쉬 함수를 이용하여 $H' = h(m)$을 계산하고 다음 식과 같이 서명 정보를 검증한다.

$$H' = s^{e_A} \bmod n_A, \quad H' = H \tag{2}$$

다음은 Schnorr의 서명 기법에 대해서 설명한다. 먼저 서명을 생성하기 위해서 서명자는 충분한 길이의 두 개의 임의의 소수 p와 q를 선택하고, 위수 q를 갖는 $g \in Z_p - 1$을 구한다. 그리고 p,q,g는 일방향 해쉬 함수 h와 함께 공개한다.

- 서명 과정: 서명자 A는 메시지 m을 서명하기 위해서 $k \in_R Z_q^*$를 선택한 다음 $r = g^k$를 생성한다. 그리고 사전에 안전하게 보관하고 있는 자신의 개인키 x_A로 다음과 같이 서명을 생성한다.

$$s = x_A h(m,r) + k \tag{3}$$

- 검증 과정: 서명자 A는 수신자 B에게 s,m,r을 전달한다. 수신자 B는 서명자 A가 전달한 메시지를 수신하면 서명자와 동일한 일방향 해쉬 함수를 이용하여 $H' = h(m,r)$을 계산한 후 서명자의 공개된 공개키 y_A를 이용하여 서명을 검증한다.

$$g^s = y_A^{h(m,r)} r \tag{4}$$

나. 대리 서명 기법

Mambo, Usuda, Okamoto는 앞서 설명한 전자 서명 기법을 활용하여 새로운 대리 서명 기법을 1996년 제안하였다[13]. 대리 서명 기법은 원서명자가 지정한 대리서명자에게 자신의 서명 권한을 부여하여 원서명자를 대신하여 서명을 할 수 있도록 하였다. 이들이 제안한 대리 서명 기법은 서명 권한의 위임 형태에 따라 완전 위임 (Full Delegation), 부분 위임 (Partial Delegation), 보증 위임 (Delegation by Warrant) 등으로 분류하였다. 완전 위임은 대리 서명자에게 원서명자가 자신의 서명용 비밀 키를 생성하여 전달하는 방식이고, 부분 위임은 대리 서명자가 원서명자가 전달한 서명용 비밀 키를 이용하여 자신의 대리 서명용 비밀 키를 생성하는 방식이다. 마지막으로 보증 위임은 원서명자가 대리서명자에게 보증 값 (warrant)을 서명과 함께 제공하여 검증하는 사용자들이 대리 서명자를 검증할 수 있도록 하는 방식을 말한다. 이러한 대리 서명 방식은 다시 대리 서명자 비보호형 (Proxy-unprotected) 방식과 대리 서명자 보호형 (Proxy-protected) 방식으로 구분된다. 대리 서명자 비보호형 방식은 대리 서명자가 원 서명자의 서명을 생성할 수 있고, 이와 동일한 서명을 원서명자 또한 생성할 수 있는 방식을 의미하고, 대리 서명자 보호형 서명 방식은 대리 서명자가 생성한 대리 서명을 원서명자가 동일하게 생성할 수 없도록 하는 방식을 말한다. 따라서 원서명자가 지정한 대리 서명자만이 유일하게 서명을 생성할 수 있다.

Mambo가 제안한 대리 서명 방식은 다음과 같다. 먼저 원서명자 A는 임의의 큰 소수 p를 선택하고 $g \in Z_p^*$를 구한다. 다음으로 서명에 사용되는 비밀 키 x_o에 대한 공개 키 $y_o = g^{x_o} \bmod p$를 계산한 후 다음과 같이 서명을 생성한다.

- 서명 과정: 원서명자 A는 서명을 생성하기 전에 $k \in_R Z_q^*$를 선택하고, $K = g^k \bmod p$를 계산한 후 대리서명자 B에게 전달할 서명 σ를 다음과 같이 생성한다.

$$\sigma = x_A + kK \tag{5}$$

- 검증 및 대리서명 생성 과정: 원 서명자 A는 σ와 K를 대리서명자 B에게 전달하고 이를 수신한 대리 서명자 B는 $g^{\sigma} = y_A K^K$를 계산하여 원서명자 A의 서명을 검증한다. 검증이 성공하면 대리서명자 B는 σ를 이용하여 대리 서명용 비밀 키 $x_P = \sigma + x_B y_B$와 공개 키 $y_P = g^{x_P} \bmod p$를 생성한 후 대리 서명용 비밀 키로 서명한다. 서명된 메시지 $S_{x_P}(m)$는 수신자에게 m, K, y_B와 함께 전달된다.
- 대리서명 검증 과정: 대리서명자의 서명은 누구든지 검증이 가능하다. 대리서명자의 서명을 수신한 노드는 원서명자로부터 정상적으로 권한을 위임 받았는지를 $y_P = y_o K^K y_B^{y_B}$를 계산하여 검증한 다음 y_P를 이용하여 서명을 검증한다.

Ⅲ. 제안기법

본 장에서는 앞서 설명했던 서명 기법을 이용하여 Mambo가 최초 제안했던 대리 서명 방식처럼 커뮤니티 기반의 유비쿼터스 네트워크 환경에서 커뮤니티 멤버 간의 상호 인증 및 안전한 비밀 키 교환 기법과 커뮤니티를 자율적으로 생성하기 원하는 커뮤니티 멤버에게 안전한 방법으로 커뮤니티 생성 권한을 부여하는 방법에 대해서 설명한다.

표 1에서는 본 논문에서 제안하는 기법에서 사용되는 주요 용어들을 정리하였다. 제안하는 기법은 앞서 설명한 그림 1과 같은 커뮤니티 기반의 유비쿼터스 네트워크 환경을 가정한다. 그리고 존 마스터 노드와 커뮤니티 멤버들의 최초 인증 및 등록을 위해 인증/등록 (AAA/Registration) 서버를 가정한다. 또한 존 마스터와 커뮤니티 멤버들의 초기 인증을 위해 EAP-TLS[14]

표 1. 용어 정리
Table 1. Definition.

표 기	정 의
s_o	원서명자의 서명 (인증/등록서버)
s_{ZM}	대리서명자의 서명 (존 마스터)
p	충분히 큰 소수
g	$g \in Z_P$의 생성자로 유한체 $GF(p)$상의 원시근
k	$o < k < q-1$에서 임의로 선택
$h(\ )$	일방향 해쉬 함수 SHA_1
ID_x	x의 식별자
X_i, Y_i	Diffie-Hellman 비밀 및 공개 값
x_i, y_i	비밀 키와 공개 키
t	타임 스탬프 값

와 대리 서명 생성을 위해 Schnorr의 서명 기법을 가정한다. 인증/등록 서버와 존 마스터 간, 존 마스터들 간에는 안전한 채널을 가정한다. 본 논문의 제안 기법에서 존 마스터 노드는 최초 부팅을 하면 인증/등록 서버를 통해 인증을 받고, 성공적으로 인증과정이 완료되면 인증/등록 서버로부터 대리 서명 권한을 부여받는다.

가. 대리 서명 발급 및 비밀 키 교환 과정

- 대리 서명 권한 위임 과정: 존 마스터가 성공적으로 초기 인증을 수행하고 나면 인증/등록 서버는 서명 권한을 존 마스터에게 부여한다. 먼저 인증/등록 서버는 $k \in_R Z_q^*$를 선택한 후 $r_o = g^k \bmod p$을 계산한다. 이후 존 마스터에게 대리 서명 권한을 위임하는 서명 s_o를 생성하여 r_o, t_o 값과 함께 존 마스터로 전달한다. 이때 전달되는 시간 t_o는 서명 생성 시간을 의미하고, 서명 생성은 다음 식과 같다.

$$H = h(Y_{ZM}, ID_{ZM}, ID_o, r_o, t_o)$$
$$s_o = Hx_o + k \bmod q \quad (6)$$

- 검증 및 대리 서명 생성 과정: 인증/등록 서버로부터 서명 정보를 전달 받은 존 마스터는 g^{s_o}를 계산함으로써 서명을 검증한 후, 검증이 성공하면 수신한 서명 정보와 자신의 Diffie-Hellman 개인 값 X_{ZM}을 이용하여 대리 서명에 사용할 대리 서명용 개인키 $x_{ZM} = s_o + X_{ZM}$과 공개키 $y_{ZM} = g^{x_{ZM}} \bmod p$을 생성한다. 이후 존 마스터는 인증/등록 서버를 대신해서 대리 서명용 개인키 x_{ZM}으로 서명 정보를 요청하는 커뮤니티 멤버의 식별정보 ID_{MN}과 Diffie-Hellman 공개 값 Y_{MN}을 Schnorr 서명 기법으로 서명한 다음 커뮤니티 멤버에게 s_{ZM}, t, r_{ZM}과 자신의 공개 값 Y_{ZM}을 함께 전송한다. 이때 함께 전송하는 r_{ZM}은 존 마스터가 임의로 선택하는 $k_{ZM} \in_R Z_q^*$을 $g^{k_{ZM}}$을 통해 계산된 값이다. 다음 식 7은 인증/등록 서버가 생성한 서명 검증과 대리 서명을 생성하는 과정을 보여준다.

$$g^{s_o} = y_o^H r_o \bmod p$$
$$s_{ZM} = h(Y_{MN}, ID_{MN}, ID_{ZM}, t_{ZM}, r_{ZM}) x_{ZM} + k_{ZM} \quad (7)$$

- 대리 서명 검증 과정: 서명을 수신한 커뮤니티 멤버는 인증/등록 서버가 정상적으로 서명 생성 권한을 위임 받은 존 마스터가 생성한 서명 인지를 검증하기 위해 존 마스터의 서명용 공개키 y_{ZM}을 검증한다. 이를 검증함으로써 정상적으로 대리 서명 위임을 받았는지를 확인할 수 있다. 검증이 성공하면 공개 키 y_{ZM}를 이용하여 대리서명 s_{ZM}을 확인한다. 다음 수식은 정상적인 대리 서명 위임 여부에 대한 검증 과정을 보여준다.

$$y_{ZM} = y_o^H r_o Y_{ZM} \bmod p$$
$$g^{s_{ZM}} = y_{ZM}^{h(Y_{MN}, ID_{MN}, ID_{ZM}, t_{ZM}, r_{ZM})} r_{ZM} \bmod p \quad (8)$$

- 커뮤니티 멤버 간 상호 인증 과정: 존 마스터로부터 서명을 발급받은 커뮤니티 멤버는 서명을 인증서처럼 사용하여 동일한 커뮤니티에 존재하는 커뮤니티 멤버들 간에 상호 인증을 수행한다. 통신을 원하는 커뮤니티 멤버는 상호 인증 요청 메시지를 상대 노드로 전달한다. 상호 인증 요청 메시지에는 S_{ZM1}, Y_{MN1}, t 등이 포함되어 전달된다. 이 메시지를 수신한 커뮤니티 멤버는 존 마스터의 서명용 공개 키 y_{ZM}로 확인 한다. 검증이 성공하면 커뮤니티 멤버는 Diffie-Hellman 공개 값을 이용하여 새로운 비밀 키 SK를 생성하고, 상호 인증 응답 메시지에 MAC_{SK}와 Y_{MN2}, S_{ZM2}, t 등을 함께 전달한다. 비밀 키 SK와 MAC_{SK}은 다음 식 9와 같이 생성한다. 이후 상호 인증 응답 메시지를 수신한 커뮤니티 멤버는 포함된 Diffie- Hellman 값을 이용하여 생성된 비밀 키로 MAC_{SK}를 검증함으로써 상호 인증 및 안전한 키 교환을 수행한다.

$$SK = (Y_{MN1})^{X_{MN2}} \bmod p$$
$$MAC_{SK} = h(SK, Y_{MN1}, Y_{MN2}, t, r_o, S_{ZM1}, S_{ZM2}) \quad (9)$$

나. 커뮤니티 생성 권한 부여 과정 (과정 추가)

커뮤니티 기반의 유비쿼터스 네트워크에서는 커뮤니티 멤버들이 필요에 따라 자유롭게 커뮤니티를 생성할 수 있다. 존 마스터로부터 서명을 발급받은 커뮤니티 멤버가 필요에 의해 커뮤니티를 생성하고자 할 때 존 마스터로부터 커뮤니티 생성 권한이 포함된 새로운 서명을 발급 받아야 한다.

- 커뮤니티 생성 권한 요청: 커뮤니티를 생성하기

원하는 커뮤니티 멤버는 존 마스터에게 커뮤니티 생성 권한을 요청한다. 이때 사용자는 존 마스터로부터 발급 받았던 서명 s_{ZM}과 자신이 생성하고자 하는 커뮤니티 관련 정보가 포함된 메시지 M, 그리고 시간 정보 t_D를 비밀 키 SK_{Z-M}으로 암호화하여 존 마스터로 전달한다. 이때 사용되는 비밀 키 SK_{Z-M}는 존 마스터의 Diffie-Hellman 공개 값 Y_{ZM}을 통해 커뮤니티 멤버가 생성한다. 요청 메시지는 다음과 같다.

$$Req = E_{SK_{Z-M}}(M, t_D), MAC_{SK_{Z-M}}, s_{ZM} \quad (10)$$

- 커뮤니티 생성 권한 위임: 존 마스터는 암호화된 커뮤니티 생성 권한 요청 메시지를 수신한 다음 메시지 복호화를 위해 커뮤니티 멤버의 Diffie-Hellman 공개 값 Y_{MN}을 이용하여 SK_{Z-M}을 생성한 후 메시지를 복호화 한다. 이후 커뮤니티 멤버의 서명을 확인하여 정상적인 커뮤니티 멤버임을 인증한 다음 커뮤니티 생성 권한이 포함된 새로운 서명 S_{ZM}을 생성하여 전달한다. 새롭게 생성되는 서명 S_{ZM}에는 커뮤니티를 생성할 수 있는 권한인 CM_{ZM} 정보와 권한 위임 시간 정보가 포함되어 생성된다. 전달되는 응답 메시지는 다음과 같다.

$$S_{ZM} = x_{ZM}(M, Y_{MN}, ID_{MN}, CM_{ZM}, t_{CM}, t_o, r_o)$$
$$Rep = E_{SK_{Z-M}}(S_{ZM}, t_{CM}), MAC_{SK_{Z-M}} \quad (11)$$

- 커뮤니티 생성 과정: 커뮤니티 생성 권한을 부여 받은 커뮤니티 멤버는 주변 노드들에게 커뮤니티 생성 권한이 포함된 서명을 전달하여 자신이 생성한 커뮤니티에 가입하도록 요청한다. 이때 존 마스터로부터 커뮤니티 생성 권한을 부여 받은 커뮤니티 멤버를 커뮤니티 마스터라고 하고, 커뮤니티 마스터는 커뮤니티에 가입하는 커뮤니티 멤버들과 자신이 생성한 커뮤니티의 다른 커뮤니티 멤버들 간에 안전한 통신을 위해 그룹 키를 생성 및 분배할 수 있다.

Ⅳ. 분석 및 비교

1. 제안 프로토콜의 안전성 분석

본 논문에서 제안하는 기법은 기존의 대리서명 프로토콜들에서 요구되는 안전성과 부인 방지기능을 만족하고, 상호 인증 및 비밀 키를 교환하는 과정에서 Diffie-Hellman 키 분배 기법을 사용하므로 안전한 비밀 채널의 가정 없이 이동 노드 간 상호 인증 및 비밀 키를 공유한다. 또한 제안하는 프로토콜의 각 수행 단계에서 원 서명자와 대리 서명자, 이동 노드들은 상호 간에 위임 정보 및 서명 정보를 검증 할 수 있기 때문에 오용 및 악의적인 사용을 검사하고 예방할 수 있다.

- 구별가능성 (Distinguishability): 서명 s_{ZM}에는 원 서명자의 ID와 대리 서명자인 존 마스터의 ID가 포함되어 있다. 또한 서명 검증 시 원 서명자의 공개 키 y_o와 대리 서명자인 존 마스터의 공개 키 y_{ZM} 모두 필요하다. 따라서 대리 서명 s_{ZM}은 원서명자가 지정한 대리인에 의해 서명된 대리서명임을 확인할 수 있다.
- 검증가능성 (Verifiability): 제안기법은 원서명자가 대리서명자에게 서명 권한을 정상적으로 위임한 것을 임의의 노드가 검증할 수 있다. 대리 서명자인 존 마스터는 원 서명자가 전달해준 서명 s_o를 기반으로 대리 서명용 개인 키 x_{ZM}를 생성하고, $g^{x_{ZM}}$하여 대리 서명용 공개 키 y_{ZM}를 생성한다. 따라서 대리 서명자의 공개 키 y_{ZM}을 검증해보면 원 서명자가 정상적으로 서명 권한을 위임 했는지 확인할 수 있다. 대리 서명용 공개 키 y_{ZM}의 검증은 식 $y_{ZM} = y_o^H r_o Y_{ZM}$를 통해 검증 할 수 있다. 즉 원 서명자의 서명 권한의 위임이 없이는 공개 키 y_{ZM}을 생성할 수 없으며 공개 키 검증을 통해 진위 여부를 확인할 수 있다.
- 강한 위조 방지 (Strong non-forgeability): 제안기법은 서명에 원서명자의 ID와 대리 서명자의 ID가 원서명자의 개인키로 서명되어 전달된다. 따라서 임의로 원서명자의 서명을 수정 및 위조 할 수 없다. 그리고 이 서명은 정상적으로 인증/등록 서버와 초기인증을 수행한 존 마스터와 공유하게 되는 공유 비밀 키를 통해 암호화되어 대리 서명자인 존 마스터에게로 전달되기 때문에 악의적인 공격자는 원서명자의 서명 s_o를 위조할 수 없다. 또한 대리 서명자에 의해 생성되는 대리 서명의 경우도 원 서명자의 서명 s_o를 기반으로 생성하는 대리 서명자의 개인 키로 서명되기 때문에 임의로 위조할 수 없다. 특히 원서명자의 서명은 대리 서명자에게 암

호화되어 전달되기 때문에 임의로 공개 되지 않으며, 대리 서명자의 공개 키를 검증할 때 필요한 원 서명자의 공개 키 y_o는 커뮤니티 멤버 노드가 인증/등록 서버와의 초기 인증 과정에서 공유하게 되는 비밀 키로 암호화되어 전달되기 때문에 공격자가 위조할 수 없다.

- 강한 신원 확인 (Strong Identifiability): 대리 서명자가 생성하는 대리 서명에는 서명자의 ID 정보와 Diffie-Hellman 공개 값이 포함되어 전달된다. 또한 원서명자의 서명에도 대리 서명자의 ID와 원서명자의 ID가원서명자의 개인 키로 서명되어 전달되고, 대리 서명에도 대리서명자의 ID 정보와 Diffie-Hellman 공개 값이 포함되어 있기 때문에 대리 서명자의 신원을 확인 할 수 있다.
- 강한 부인 방지 (Strong non-deniability): 원서명자는 자신이 대리 서명자에게 서명 권한을 위임하기 위해 생성하는 서명 s_o는 원서명자의 개인 키로 서명되어 대리 서명자와 공유하는 비밀 키로 암호화되어 전달되기 때문에 원서명자는 자신의 서명에 대해 부인 할 수 없다. 또한 대리 서명자는 자신이 생성하는 대리 서명을 자신의 대리 서명용 개인 키로 서명한다. 이 개인 키는 원서명자의 서명 s_o와 대리 서명자의 Diffie-Hellman 개인 값 X_{ZM}을 통해 생성하기 때문에 서명에 대해 부인 할 수 없다.

2. RSA 서명 기법과의 효율성 비교

다음은 본 논문에서 제안하고 있는 Schnorr의 서명 기법에 기반 한 제안 기술과 RSA 서명 기법을 본 논문에서 제안하는 기법에 적용하였을 경우를 비교하여 효율성을 분석한다. 두 기법에 대한 효율성 비교는 각 기법들에서 서명을 생성하는 과정에서 발생하는 계산량을 비교하여 분석하였다.

RSA 서명 기술을 이용하여 대리 서명을 제공할 경우 다음 식 12와 같이 대리 서명을 생성할 수 있다.

$$s_{ZM} = h(y_{ZM}, ID_{MN}, ID_{ZM}, t, e_{MN})^{d_{ZM}} \bmod n_{ZM} \quad (12)$$

위의 식과 같이 생성된 서명은 서명을 요청한 이동 노드로 전달되고, 다음 식 11과 같은 방법으로 서명을 검증할 수 있다.

$$(s_{ZM})^{e_{ZM}} \bmod n_{ZM} = (h(y_{MN}, ID_{MN}, ID_{ZM}, t, e_{MN})^{d_{ZM}})^{e_{ZM}} \bmod n_{ZM} \quad (13)$$

표 2. 계산량 비교
Table 2. Computation Time Comparison.

	제안 기법	RSA 서명 기법
서명자 계산량	$t_H + t_{MUL} + t_{ADD} + t_{MOU}$	$t_H + t_{\mathrm{EXP}} + t_{MOU}$
검증자 계산량	$t_H + t_{\mathrm{EXP}} + t_{MUL} + t_{MOU}$	$t_H + 2t_{\mathrm{EXP}} + t_{MOU}$

다음의 표 2에서 제안기법과 RSA 서명을 이용한 대리 서명 방식에서 대리 서명을 생성하고 검증하는 과정에서의 계산량을 비교한 것을 정리하였다. 표 2에서 계산량을 비교하기 위해 사용된 t_{MUL} 은 곱셈 연산 시간, t_{MOU}는 모듈러 연산 시간, t_{EXP}는 지수승 연산 시간, t_{ADD}는 덧셈 연산 시간, t_H는 해쉬 연산을 수행하는데 걸리는 시간을 의미한다. 표 2에서 정리한 것처럼 제안 기법의 경우 서명을 생성하고 검증하는 과정에서 지수승 연산을 한번 필요로 하는 반면 RSA 서명 기법의 경우 세 번의 지수승 연산을 요구한다. 따라서 RSA 서명 기법에 비해 제안하는 대리 서명 기법의 계산량이 상대적으로 적음을 알 수 있고, 이는 일반 컴퓨터 환경보다 연산 능력이 빈약한 소형 단말 환경에는 제안 기법이 더 효과적임을 알 수 있다.

V. 결 론

본 논문은 커뮤니티 기반의 유비쿼터스 네트워크에서 커뮤니티 멤버들 간의 상호 인증과 안전한 비밀 키교환 기법을 제안하였다. 또한 커뮤니티 멤버들이 필요에 의해 생성하는 자율적인 커뮤니티를 생성할 수 있도록 커뮤니티 생성 권한을 안전하게 위임할 수 있도록 제안하였다. 제안 기법은 인증/등록 서버를 대신하여 커뮤니티에 존재하는 커뮤니티 멤버들에게 대리로 서명을 생성할 수 있는 권한 위임을 위해 Schnorr의 서명 기법 기반의 대리 서명 기술을 사용하였고, 대리 서명을 존 마스터로부터 전달 받은 커뮤니티 멤버들이 안전하게 비밀 키를 공유할 수 있도록 Diffie-Hellman 키 교환 기법을 적용하였다. 이렇게 함으로써 인증/등록 서버의 도움 없이 커뮤니티 멤버들 간에 대리 서명 기반의 상호 인증 및 안전한 키 교환을 가능하게 하였다. 또한 기존 RSA 서명 기반의 대리 서명 방식에 비해 계산량이 비교적 작아 다양한 소형 단말이 존재하는 커뮤니티 기반의 유비쿼터스 네트워크에 적용하기에 효과적임을 보였다.

참 고 문 헌

[1] K. Namhi, P. Ilkyun, and K. Younghan, "Ubiquitous zone networking technologies for multi-hop based wireless communications," *IWSOS 2006, LNCS 4124*, September 2006.

[2] C. Jaeduck, R. Hyosun, J. Souhwan, and K. Younghan, "Support of Context-awareness in Ubiquitous Networks using Smart Packet," *IPSJ SIG 2005*, pp. 275-280, 2005.

[3] S. Corson and J. Macker, "Mobile ad-hoc networking (MANET)," *IETF RFC 2051*, January 1999.

[4] P. Argyroudis and D. O'Mahony, "Secure routing for mobile ad hoc networks," *Communications surveys & Tutorials, IEEE volume 7*, pp. 2-21, 2005.

[5] F Zhang, R Safavi-Naini and CY Lin, "New Proxy Signature, Proxy Blind Signature and Proxy Ring Signature Schemes from Bilinear Pairings," *Advances in Cryptology-Asiacrypt 2002, LNCS 2510*, 2002.

[6] W. Diffie and M. E. Hellman, "Multi cryptographic techniques," *AFIPS Conference Proceedings*, pp. 109-112, 1976.

[7] C. Siva Ram Murthy and B. S, Manoj, "Ad Hoc Wireless Networks Architectures and Protocols," *Prentice Hall PTR*, 2004.

[8] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem," *Commun. ACM*, pp. 120-126, 1978.

[9] T. ElGamal, "A Public Key Cryptosystem and a signature scheme based on discrete logarithms," *Advances of Cryptology-CRYPTO '84, LNCS 196*, pp. 10-18, 1985.

[10] C. P. Schnorr, "Efficient Identification and Signatures for Smart cards," *Advances of Cryptology-CRYPTO '89, LNCS*, pp. 239-251, January, 1989.

[11] K. Nyberg and R. A. Rueppel, "Message recovery for signature scheme based on the discrete logarithm problem," *Eurocrypt '94 proceedings*, 1995.

[12] FIPS PUB XX, "Digital Signature Standard (DSS)," February, 1993.

[13] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation pf the power to sign message," *IEICE Teans, E79-A*, pp. 1338-1354, 1996.

[14] D. Simon, B. Aboba and R. Hurst, "The EAP-TLS Authentication Protocol," *IETF RFC 5216*, March 2008.

저 자 소 개

노 효 선(정회원)

2005년 숭실대학교 정보통신전자공학부 학사

2007년 숭실대학교 정보통신전자공학과 석사

2007년~현재 숭실대학교 전자공학과 박사과정

<주관심분야 : 네트워크 보안, 이동 네트워크 보안, IPTV 보안>

정 수 환(평생회원)-교신저자

1985년 서울대학교 전자공학과 학사

1987년 서울대학교 전자공학과 석사

1996년 University of Washington 박사

1996년~1997년 Stellar One SW Engineer

1997년~현재 숭실대학교 정보통신전자공학부 부교수

2009년~현재 지식경제부 지식정보보안 PD

<주관심분야 : 이동 네트워크 보안, 차량 네트워크 보안, VoIP 보안, RFID/USN 보안>