情報保護學會論文誌
第16卷 第2號, 2006. 4

유비쿼터스 컴퓨팅 환경에서의 익명성을 보장하는 사용자 인증 및 키 동의 프로토콜 설계

강 명 희,† 유 황 빈‡
광운대학교

Design of Anonymity-Preserving User Authentication and Key Agreement Protocol in Ubiquitous Computing Environments

Myung-hee Kang,† Hwang-bin Ryou‡
KwangWoon University

요 약

모바일 디바이스, PDA, 센서들의 확산은 실생활 공간이 지능적이고 향상된 서비스를 제공하는 스마트 공간으로 전환되는 유비쿼터스 컴퓨팅 환경이 구축될 수 있도록 하였다. 그러나 모바일 디바이스, PDA, 센서들과 같은 다양한 유비쿼터스 디바이스들의 초기 설계 단계부터 프라이버시 문제를 고려하지 않는다면 유비쿼터스 감시체제가 구축될 위험성이 있다. 또한 유비쿼터스 컴퓨팅 환경에서의 다양한 디바이스들은 대체로 컴퓨팅 연산 능력이 적기 때문에, 공개키 기반의 암호 기술을 적용하는 것은 부적절할 수 있다. 본 논문에서는 유비쿼터스 컴퓨팅 환경에서의 사용자 프라이버시를 위하여 익명성이 보장되고, 컴퓨팅 연산 능력이 적은 디바이스를 위한 효율적인 사용자 인증 및 키 동의 프로토콜을 제안한다.

ABSTRACT

The spread of mobile devices, PDAs and sensors has enabled the construction of ubiquitous computing environments, transforming regular physical spaces into "smart space" augmented with intelligence and enhanced with services. However, unless privacy concerns are taken into account early in the design process of various ubiquitous devices(e.g. mobile devices, PDAs, sensors, etc.). we will end up crating ubiquitous surveillance infrastructure. Also, it may inappropriate to use public key techniques for computational constrained devices in ubiquitous computing environment. In this paper, we propose efficient user authentication and key agreement protocol not only to preserve anonymity for protecting personal privacy but also to be suitable for computational constrained devices in ubiquitous computing environments.

Keywords : *User Authentication, Privacy, Key Agreement, Anonymity*

Ⅰ. 서 론

Mark Weiser에 의해 제시된 유비쿼터스 컴퓨팅은 일상 환경과 사물들에 컴퓨터가 보이지 않게 심어져 있어 서로간의 정보 교환이 가능하며, 사용자는 컴퓨터라는 거부감을 느끼지 않고 언제, 어디서나, 도처에 존재하는 컴퓨터를 편리하게 사용할 수 있는 컴퓨팅 환경이다. 그러나 언제 어디서나 컴퓨팅을 할 수 있다는 것은 언제 어디에서든지 정보

접수일: 2005년 8월 24일; 채택일: 2006년 3월 6일
† 주저자, mhkang@kw.ac.kr
‡ 교신저자, ryou@kw..ac.kr

가 누출되거나 왜곡될 위험성이 존재하기 마련이다. 특히, 프라이버시 문제가 해결되지 않고서는 유비쿼터스 감시 체제가 구축되는 심각한 역기능을 초래하게 될 것이다. 한편, 유비쿼터스 컴퓨팅 디바이스들은 크기와 모양이 다양하고 컴퓨팅 연산 능력이 적은 소형 휴대 장치들이 많기 때문에, 기존의 공개키 암호 기술을 소형 휴대 장치에 적용하기가 어렵다. 본 논문에서는 유비쿼터스 컴퓨팅 환경(스마트 공간)상에서의 다양하고 컴퓨팅 연산 능력이 적은 소형 디바이스들이 안전하게 암호 통신을 수행하고 사용자 개인의 프라이버시를 보호할 수 있는 익명성이 보장되는 사용자 인증 및 키 동의 프로토콜을 제안한다. 본 논문의 구성은 다음과 같다. Ⅱ장에서는 관련 연구 내용을 기술하며, Ⅲ장에서는 본 논문에서 제안한 익명성이 보장되는 인증 및 동의 프로토콜을 다루며, Ⅳ장에서는 제안 프로토콜에 대한 시스템 분석을 하고, Ⅴ장에서는 결론을 맺는다.

Ⅱ. 관련 연구

Kinderg [3, 4] 논문에서는 두 디바이스간에 안전하면서도 자발적으로 접속 연결할 수 있는 프로토콜을 제안하였다. 이 논문에서 제안한 접속 프로토콜에서는 레이저를 이용한다. Initiator 디바이스에 레이저 장치가 있고, Responder 디바이스에 레이저를 읽을 수 있는 장치가 마련되었다고 가정한다. Initiator 디바이스에서 레이저를 이용하여, 초기 세션키를 Responder 디바이스에 전송한 다음, 이후 통신에서는 초기 세션키를 이용하여, 암호 통신을 하는 방식이다. Kinderg 논문에서 제안한 방식은 신뢰할 수 있는 3자와의 통신 없이 디바이스들 자체적으로 안전한 통신 채널을 구성할 수 있는 편리성이라는 장점을 가지지만, 디바이스 간에 인증 과정이 없기 때문에, 서비스 거부 공격과 같은 보안 취약점을 가지고 있다. 따라서 편리성보다 안전성이 보다 중요한 환경에서는 TTP(Trust Third Party)를 두어, 디바이스에 대한 인증 과정을 반드시 수행하고 키를 공유하는 것이 바람직할 것으로 생각된다.

익명성이나 프라이버시를 보호하기 위한 기존의 연구들은 크게 사용자 익명성과 통신의 익명성을 제공하는 방법을 다루고 있다. 사용자의 익명성은 통신 상대방에게 자신의 신원을 숨기면서 네트워크를 사용함으로써, 사용자 익명성을 제공한다. [9, 11] 그러나 사용자 인증을 요구하지 않기 때문에 서비스 거부 공격과 같은 보안상의 취약점이 존재할 수 있다. 한편 Freedom.net 사의 Zero Knowledge System은 사용자의 단말 호스트에 클라이언트 프로그램을 설치하고, 암호학적으로 보호되는 가명을 사용하여 Telnet, FTP, E-mail 등의 서비스를 받도록 하고 있다. [10] Freedom.net의 사용자들은 서비스를 사용하기 전에 사용자 인증을 먼저 받아야 한다. 또한, 유비쿼터스 컴퓨팅 환경에서의 위치 정보에 대한 프라이버시 문제를 해결하기 위한 연구인 Gaia 프로젝트에서는 Mist라는 통신 프로토콜을 제안하였다. [5-7] Mist의 각 사용자는 Light House라고 하는 특별한 Mist 라우터에 라우팅에 필요한 최소한의 정보를 등록하도록 하고 있다. Mist 프로토콜은 통신 인프라 레벨에서 사용자의 위치 정보의 프라이버시를 보호하기 위하여, 새로운 통신 패킷을 정의하고, 사용하고 있다.

Ⅲ. 제안 프로토콜

3.1. 인증 및 키 동의 프로토콜

본 논문에서는 유비쿼터스 컴퓨팅 환경에서의 사용자 정보에 대한 익명성을 보장하며, 다양한 디바이스들과의 안전한 암호 통신을 위한 익명성을 보장하는 인증 및 키 동의 프로토콜을 제안한다. 본 논문에서 제안한 프로토콜은 Leighton-Micali [8] 논문의 제안 기법을 응용하였다. 그림 1은 본 논문에서 제안하는 인증 및 키 동의 프로토콜을 나타낸 것이다.

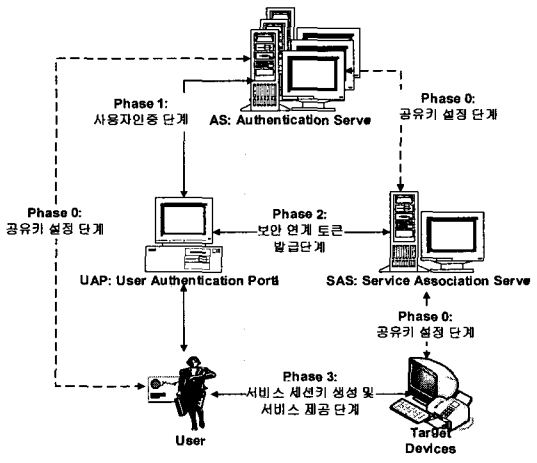

그림 1. 사용자 인증 및 키 동의 프로토콜 구조

본 논문에서 제안한 인증 및 키 동의 프로토콜의 시스템 구성요소 및 역할은 다음과 같다.

- **AS(Authentication Server)**: 사용자와 SAS(Service Association Server)에 대한 인증을 수행하는 TTP(Trust Third Party)이다. 유비쿼터스 컴퓨팅 환경과 같은 다양한 종류의 서비스를 제공하는 분산 환경에서는 TTP인 AS도 분산된 경우가 일반적이기 때문에 본 논문에서는 AS가 분산된 경우를 고려한다. 따라서 사용자와 SAS가 신뢰하는 TTP가 서로 다를 수 있으며, 본 논문에서는 사용자가 신뢰하는 TTP인 AS_1와 SAS가 신뢰하는 TTP인 AS_2로 분산되어 있다고 가정한다.

- **SAS(Service Association Server)**: 스마트 공간상에서 사용자에게 보안 연계 토큰(SAT: Security Association Token)을 발급하며, Target Device에 대한 전반적인 관리를 담당한다.

- **UAP(User Authentication Portal)**: 스마트 공간에 사용자가 입장하였을 때, 사용자가 사용자 인증을 받기위해 접촉하는 AS와 사용자 사이의 중간 매개 역할자로서, 익명 인증서(AC: Anonymous Certificate)를 AS에 요청하고, 발급받아, 사용자에게 전달하는 역할을 수행한다. 또한 SAS와 사용자 사이의 중간 매개 역할자로서, 보안 연계 토큰(SAT: Security Association Token)을 발급받아, 사용자에게 전달하는 역할을 수행한다.

- **Target Device**: 스마트 공간상의 사용자에게 유비쿼터스 서비스를 제공하는 디바이스이다.

본 논문에서 제안한 인증 및 키 동의 프로토콜은 크게 3단계로써, 사용자 인증 단계, 보안 연계 토큰 발급 단계, 서비스 세션키 생성 및 서비스 제공 단계로 구성되며, 전처리 과정으로써, 각 시스템 구성요소의 공유키 설정 단계가 있다.

3.1.1. 기호 설명

- $||$: A $||$ B라고 했을 때, A와 B를 연접하는 기호
- $|A|$: A의 길이를 나타내는 기호
- $\oplus$: A $\oplus$ B라고 했을 때, A와 B를 XOR 연산하는 기호
- $\doteq$: A $\doteq$ B라고 했을 때, A와 B의 값을 비교하는 기호
- CMAC(K, M): 키를 K로 하여, 메시지 M에 대한 CBC-MAC 연산을 하는 기호
- E(K, M): 키를 K로 하여, 메시지 M을 암호화 하는 기호
- D(K, M): 키를 K로 하여, 메시지 M을 복호화 하는 기호
- $K_{AS1,AS2}$: AS_1과 AS_2간의 공유키
- $KeyID_{AS1,AS2}$: $K_{AS1,AS2}$의 식별자
- $K_{AS1,U}$: 사용자 U와 AS_1간의 공유키
- $KeyID_{AS1,U}$: $K_{AS1,U}$의 식별자
- $K_{AS2,SAS}$: SAS와 AS_2간의 공유키
- $K_{SAS,Ti}$: Target Device T_i와 SAS간의 공유키
- $PK_{U,SAS}$: 사용자와 SAS간의 pair-wise public key
- $BK_{U,SAS}$: 사용자와 SAS 간의 공유하는 스마트 공간 기본키
- $PK_{U,Ti}$: 사용자와 Target Device T_i 사이의 pair-wise public key
- $SK_{U,Ti}$: 사용자와 Target Device T_i 사이의 서비스 세션키
- R_{AS}: AS가 생성한 nonce 값
- R_{SAS}: SAS가 생성한 nonce 값
- R_U: 사용자 U가 생성한 nonce 값

3.1.2 공유키 설정 단계

- **사용자와 AS_1간의 공유키 설정**: 사용자와 AS_1간의 사용자 인증을 위해 공유하는 마스터키($K_{AS1,U}$)를 설정하며, $K_{AS1,U}$는 AS가 발행한 익명 인증서에 대한 pair-wise digital signature를 사용자가 검증하는데 사용된다.

- **SAS와 AS_2간의 공유키 설정**: SAS와 AS_2간의 공유하는 마스터키($K_{AS2,SAS}$)를 설정하며, $K_{AS2,SAS}$는 AS가 발행한 익명 인증서에 대한 pair-wise digital signature를 SAS가 검증하는데 사용된다.

- **Target Device T_i와 SAS와의 공유키 설정**: SAS와 Target Device T_i와 최초에 공유

하게 되는 마스터키($K_{SAS,Ti}$)를 설정하며, $K_{SAS,Ti}$는 SAS에 대한 pair-wise digital signature를 T_i가 검증하는데 사용된다.

3.2. 사용자 인증 단계

유비쿼터스 컴퓨팅 환경이 구축되어 있는 스마트 공간에 사용자가 입장하였을 때, 맨 처음 사용자 인증을 수행하는 단계이다. 그림 2는 사용자가 신뢰하는 TTP인 AS_1과 SAS가 신뢰하는 TTP인 AS_2로 AS가 분리 되어 있고, AS_1과 AS_2간의 상호 신뢰관계가 형성되어 있을 경우의 사용자 인증 및 익명 인증서의 발급 과정을 나타낸 것이다. 사용자는 AS_1_ID, U_ID, AS_2_ID, SAS_ID, R_U를 연접하여, ID_Info 메시지를 생성하고, $K_{AS1,U}$를 이용하여 ID_Info 메시지에 대한 CBC-MAC 값을 계산하여 $AUTH_U$를 생성한다. ID_Info 메시지와 $AUTH_U$를 연접한 후, $K_{AS1,U}$를 이용하여 암호화하여, REQ_ACInfo 메시지를 생성한다. $KeyID_{AS1,U}$와 REQ_ACInfo 메시지를 연접하여 익명 인증서 요청 메시지 REQ_AC를 생성한다. AS_2는 R_{AS2}를 생성하고, KM_{SAS} = CMAC($K_{AS2,SAS}$, R_{AS2}) 계산한다. 그리고 R_{AS2}와 KM_{SAS}를 연접하고, $K_{AS1,AS2}$를 이용하여 암호화하여 E($K_{AS1,AS2}$, R_{AS2} || KM_{SAS})를 생성한 다음, REQ_AC || $KeyID_{AS1,AS2}$ || E($K_{AS1,AS2}$, R_{AS2} || KM_{SAS})를 AS_1에 전송한다. AS_1은 $KeyID_{AS1,AS2}$를 이용하여, $K_{AS1,AS2}$를 검색한 후, R_{AS2}, KM_{SAS}를 복호화 한다. 또한 AS_1은 REQ_AC 메시지에서 $KeyID_{AS1,U}$를 이용하여, $K_{AS1,U}$를 검색하고, $K_{AS1,U}$를 이용하여 REQ_ACInfo를 복호화하고, $AUTH_U$에 대한 유효성을 검증하여, 사용자 인증을 수행한다. $AUTH_U$가 유효하면, toBeSignedACInfo, authAS, authUser 필드를 생성하여, AS_2에 전송한다. AS_2는 $K_{AS1,AS2}$를 이용하여 authAS 필드를 검증하고 authAS가 유효하면, $K_{AS2,SAS}$를 이용하여 사용자 익명 인증서의 authSAS 필드를 생성한다. toBeSignedACInfo, authAS, authUser,

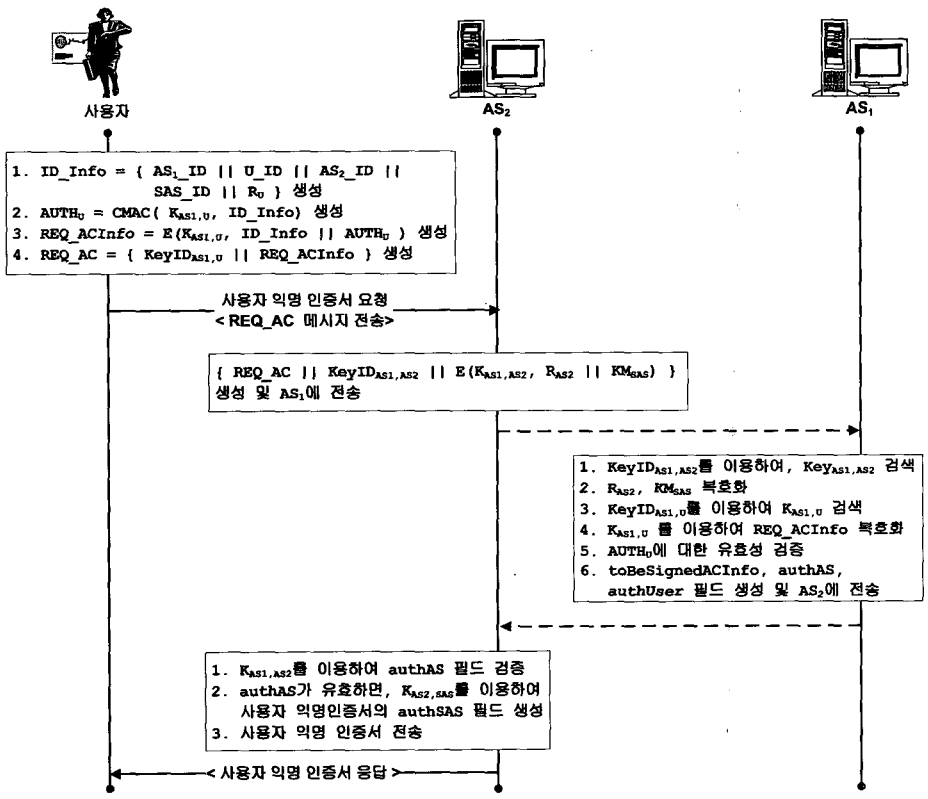

그림 2. 사용자 인증 및 익명 인증서 발급 과정

표 1. 익명 인증서 구조

필 드		내 용
ToBeSignedACInfo	AS_1ID	사용자가 신뢰하는 TTP인 AS_1의 ID
	anonymousID	AS_1이 관리하는 사용자에 대한 Random ID
	AS_2ID	사용자가 신뢰하는 TTP인 AS_2의 ID
	sasID	AS_2가 관리하는 SAS ID
	validity	사용자 익명 인증서의 유효기간
	nonceAS(R_{AS2})	AS_2가 생성한 Nonce
	pwPubKey($PK_{U,SAS}$)	U와 SAS간의 pair-wise public key (생성방법은 먼저 KM_U = CMAC($K_{AS1,U}$, R_{AS2}), KM_{SAS} = CMAC($K_{AS2,SAS}$, R_{AS2}) 계산 후, $PK_{U,SAS}$ = $KM_U \oplus KM_{SAS}$ 계산)
ACSignature	authAS	AS_2가 AS_1이 생성한 익명 인증서의 유효성을 검증할 수 있도록 CMAC($K_{AS1,AS2}$, toBeSignedACInfo)를 계산하여, 생성한 인증 값
	authUser	사용자 U가 익명 인증서의 유효성을 검증할 수 있도록 AS_1가 CMAC($K_{AS1,U}$, toBeSignedACInfo)를 계산하여, 생성한 인증 값
	authSAS	SAS가 익명 인증서의 유효성을 검증할 수 있도록 AS_2가 CMAC($K_{AS2,SAS}$, toBeSignedACInfo)를 계산하여, 생성한 인증 값

authSAS 필드를 연접하여 사용자 익명 인증서를 완성한 후, 사용자에게 전송한다. 표 1은 인증서의 세부 필드에 대한 구조를 나타낸 것이다.

AS_1, AS_2로부터 발급 받은 익명 인증서를 사용자는 AS_1과 공유하고 있는 $K_{AS1,U}$를 가지고 익명 인증서의 유효성을 검증할 수 있다. 유효성 검증 과정은 CMAC($K_{AS1,U}$, toBeSignedACInfo)를 먼저 계산한 다음, authUser 필드와 비교하여 값이 같으면 유효한 것으로 본다.

$$CMAC(K_{AS1,U}, \text{toBeSignedACInfo}) \stackrel{?}{=} \text{authUser}$$

사용자는 익명 인증서의 $PK_{U,SAS}$로부터 SAS와 사용자간의 스마트 공간 기본키 $BK_{U,SAS}$를 생성한다. $BK_{U,SAS}$의 생성 과정은 다음과 같다. 먼저 KM_U = CMAC($K_{AS1,U}$, R_{AS2})를 계산한 후, KM_{SAS} = $PK_{U,SAS} \oplus KM_U$를 계산한다. $PK_{U,SAS}$의 크기가 16 바이트(즉 암호 알고리즘의 블록의 크기가 16바이트)이면 KM_U의 상위 8바이트와 KM_{SAS}의 상위 8바이트를 서로 연접하여 $BK_{U,SAS}$를 생성한다. $PK_{U,SAS}$의 크기가 8바이트(즉 암호 알고리즘의 블록의 크기가 8바이트)이면 KM_U와 KM_{SAS}를 서로 연접하여 $BK_{U,SAS}$를 생성한다.

```
if (|PK_{U,SAS}| == 16)
{
  KM_U = KM_U[0..7] || KM_U[8..15] = CMAC(K_{AS1,U}, R_{AS2})
  KM_SAS = KM_SAS[0..7] || KM_SAS[8..15] = PK_{U,SAS} ⊕ KM_U
  BK_{U,SAS} = KM_U[0..7] || KM_SAS[0..7]
}
else if (|PK_{U,SAS}| == 8)
{
  KM_U = KM_U[0..3] || KM_U[4..7] = CMAC(K_{AS1,U}, R_{AS2})
  KM_SAS = KM_SAS[0..3] || KM_SAS[4..7] = PK_{U,SAS} ⊕ KM_U
  BK_{U,SAS} = KM_U || KM_SAS
}
```

3.3. 보안 연계 토큰 발급 단계

SAS가 사용자의 익명 인증서를 검증한 후, 보안 연계 토큰을 발급하는 단계이다. UAP는 SAS에 사용자의 익명 인증서를 전송하여, 사용자의 보안 연계 토큰을 발급 요청하면, SAS는 $K_{AS2,SAS}$를 이용하여, AS_1과 AS_2가 생성한 사용자의 익명 인증서에 대한 유효성을 검증한다. 유효성 검증 과정은 다음과 같다. CMAC($K_{AS2,SAS}$, toBeSignedACInfo)를 먼저 계산하고, authSAS 필드 값을 비교하여, 동일하면 유효한 것으로 간주한다.

$$CMAC(K_{AS2,SAS}, \text{toBeSignedACInfo}) \stackrel{?}{=} \text{authSAS}$$

사용자의 익명 인증서가 유효하면, SAS는 익명 인증서의 $PK_{U,SAS}$로부터 SAS와 사용자간의 스마트 공간 기본키 $BK_{U,SAS}$를 생성한다. $BK_{U,SAS}$의 생성과정은 사용자가 생성한 방법과 거의 동일한 방법으로 계산된다. 먼저 KM_{SAS} = CMAC($K_{AS2,SAS}$, R_{AS2})를 계산한 후, KM_U = $PK_{U,SAS} \oplus$

KM_{SAS}를 계산한다. $PK_{U,SAS}$의 크기가 16 바이트 이면 KM_U의 상위 8바이트와 KM_{SAS}의 상위 8바이트를 서로 연접하여 $BK_{U,SAS}$를 생성하고, $PK_{U,SAS}$의 크기가 8바이트이면 KM_U와 KM_{SAS}를 서로 연접하여 $BK_{U,SAS}$를 생성한다.

$BK_{U,SAS}$를 생성한 SAS는 보안 연계 토큰의 일련번호, issuerID(SAS ID), holderID(사용자의 익명 ID), 유효기간, SAS가 생성한 Nonce 값, targetDeviceID 정보와 $BK_{U,SAS}$를 이용하여, 사용자의 보안 연계 토큰을 생성한다. 표 2는 보안 연계 토큰의 세부 필드에 대하여 기술한 것이다.

사용자 U는 SAS로부터 발급 받은 자신의 보안 연계 토큰의 유효성 검증한다. 유효성 검증 과정은 CMAC($BK_{U,SAS}$, toBeSignedSATInfo)를 계산하고 보안 연계 토큰의 authHolder 필드와 비교하여, 동일하면 유효한 것으로 간주한다.

3.4. 서비스 세션키 생성 단계

본 단계에서는 사용자가 SAS로부터 발급 받은 보안 연계 토큰을 제공받고자 하는 서비스를 수행하는 Target Device T_i에 제출하여, 서비스를 요청하고 제공받는 단계이다.

1. 먼저, 사용자는 자신의 보안 연계 토큰과 스마트 공간 기본키 $BK_{U,SAS}$를 이용하여, 서비스 세션키 $SK_{U,Ti}$를 생성한다. 서비스 세션키 생성 방법은 스마트 공간 기본키 생성 방법과 거의 유사하다. $SK_{U,Ti}$의 생성과정은 먼저 $BKM_{U,Ti}$ = CMAC($BK_{U,SAS}$, R_{SAS})를 계산하고, KM_{Ti} = $PK_{U,Ti}$ ⊕ $BKM_{U,Ti}$를 계산한다. $PK_{U,Ti}$의 크기가 16바이트이면 $BKM_{U,Ti}$와 KM_{Ti}의 상위 8바이트를 연접하여 $SK_{U,Ti}$를 생성하고, $PK_{U,Ti}$의 크기가 8바이트이면 $BKM_{U,Ti}$와 KM_{Ti}를 서로 연접하여 $SK_{U,Ti}$를 생성한다.
2. 사용자는 서비스 요청 메시지 REQ_SERV를 Target Device T_i에 전송한다. REQ_SERV 메시지 생성 과정은 다음과 같다.

$$REQ\_SERV = \{ SAT \,||\, R_U \,||\, AUTH_U \}$$
$$AUTH_U = CMAC(SK_{U,Ti}, R_U)$$

3. Target Device T_i는 $K_{SAS,Ti}$를 이용하여 사용자 보안 연계 토큰에 대한 유효성을 다음과 같이 검증 한다.

$$CMAC(K_{SAS,Ti}, toBeSignedSATInfo) \stackrel{?}{=} authTarget$$

4. Target Device T_i는 사용자의 보안 연계 토큰을 이용하여 $SK_{U,Ti}$를 생성한다. $SK_{U,Ti}$의 생성 과정은 사용자가 생성하는 방법과 거의 동일하게 계산된다. 먼저 KM_{Ti} = CMAC($K_{SAS,Ti}$, R_{SAS})를 계산하고, $BKM_{U,Ti}$ = $PK_{U,Ti}$ ⊕ KM_{Ti}를 계산한다. $PK_{U,Ti}$의 크기가 16바이트이면 $BKM_{U,Ti}$와 KM_{Ti}의 상위 8바이트를 연접하여 $SK_{U,Ti}$를 생성하고, $PK_{U,Ti}$의 크기가 8바이트이면 $BKM_{U,Ti}$와 KM_{Ti}를 서로 연접하여 $SK_{U,Ti}$를 생성한다.
5. $SK_{U,Ti}$를 이용하여 $AUTH_U$의 유효성을 검증하여, 유효하면 해당 서비스를 제공한다. 유효성 검증 과정은 다음과 같다.

$$CMAC(SK_{U,Ti}, R_U) \stackrel{?}{=} AUTH_U$$

표 2. 보안 연계 토큰 구조

필 드		내 용
ToBeSignedSATInfo	serialNumber	보안 연계 토큰 일련번호
	issuerID	보안 연계 토큰 발급자의 ID
	holderID	보안 연계 토큰 사용자의 익명 ID
	validity	보안 연계 토큰의 유효기간
	nonceSAS(R_{SAS})	SAS가 생성한 nonce
	targetDeviceID	Target Device의 ID
	pwPubKey($PK_{U,Ti}$)	U와 T_i 간의 pair-wise public key (KM_{Ti} = CMAC($K_{SAS,Ti}$, R_{SAS}), $BKM_{U,Ti}$ = CMAC($BK_{U,SAS}$, R_{SAS})를 계산 후, $PK_{U,Ti}$ = KM_{Ti} ⊕ $BKM_{U,Ti}$ 계산)
SATSignature	authHolder	사용자 U와 SAS간의 스마트 공간 기본키 $BK_{U,SAS}$를 이용하여 SAS가 생성한 인증 값
	authTarget	T_i와 SAS간의 공유키 $K_{SAS,Ti}$를 이용하여 SAS가 생성한 인증 값

Ⅳ. 시스템 분석

4.1. 안전성 분석

본 논문에서 제안한 사용자 인증 및 키 동의 프로토콜에서는 AS_1, AS_2는 TTP(Trust Third Party)이며, CBC-MAC 알고리즘과 Leighton- Micali 기법은 안전하다고 가정한다. 본 논문에서 제안한 프로토콜은 사용자 인증, 사용자 익명 인증서의 생성 및 검증, 스마트 공간 기본키 공유, 사용자 보안 연계 토큰의 생성 및 검증, 서비스 세션키 공유로 나누어 볼 수 있다.

- **사용자 인증**: 사용자 인증 과정은 사용자 U가 $K_{AS1,U}$를 가지고 R_U에 대한 CBC-MAC 값을 계산하여, AS_1에 전송하고, AS_1은 $K_{AS1,U}$를 가지고 R_U에 대한 CBC-MAC 값을 검증하는 방식으로 이루어지므로 안전하다고 볼 수 있다.

- **사용자 익명 인증서의 생성 및 검증**: 사용자 익명 인증서는 사용자 U와 AS_1만이 공유하고 있는 키 $K_{AS1,U}$와 AS_2와 SAS만이 공유하고 있는 키 $K_{AS2,SAS}$, 그리고 AS_1과 AS_2만이 공유하고 있는 키 $K_{AS1,AS2}$가 있어야만 생성할 수 있다. 따라서 AS_1, AS_2가 서로 협력하여야 사용자 익명 인증서를 생성할 수 있다. 그러므로 $K_{AS1,U}$, $K_{AS2,SAS}$, $K_{AS1,AS2}$ 모두가 노출되지 않는 한 사용자 익명 인증서의 불법적인 생성은 불가능하므로 사용자 익명 인증서의 생성 과정은 안전하다고 볼 수 있다. 또한 사용자 익명 인증서의 유효성 검증은 사용자 U, SAS만이 각각 authUser, authSAS 필드를 검증 할 수 있다. 따라서 사용자 익명 인증서의 검증 과정 또한 안전하다고 볼 수 있다.

- **스마트 공간 기본키 공유**: 사용자 익명 인증서 내부 필드인 pair-wise public key $PK_{U,SAS}$로부터 KM_U와 KM_{SAS}를 계산해내는 것은 Leighton-Micali[8] 논문에서 제시된 바와 같이 사용자 U와 SAS만이 할 수 있기 때문에, 스마트 공간 기본키 $BK_{U,SAS}$는 사용자 U와 SAS 간에 안전하게 공유되었다고 볼 수 있다.

- **사용자 보안 연계 토큰의 생성 및 검증**: 사용자 보안 연계 토큰은 사용자 U와 SAS가 안전하게 공유한 스마트 공간 기본키 $BK_{U,SAS}$와 SAS와 Target Device T_i 만이 공유하고 있는 키 $K_{SAS,Ti}$가 있어야만 ToBeSignedSATInfo 필드에 대한 CBC-MAC 값을 계산할 수 있기 때문에, SAS 만이 생성할 수 있다. 그러므로 $BK_{U,SAS}$, $K_{SAS,Ti}$, 모두가 노출되지 않는 한 사용자 보안 연계 토큰의 불법적인 생성은 불가능하므로 사용자 보안 연계 토큰의 생성과정은 안전하다고 볼 수 있다. 또한 보안 연계 토큰에 대한 검증은 사용자 U, Target Device T_i만이 각각 authHolder, authTarget 필드를 검증할 수 있기 때문에 사용자 보안 연계 토큰 검증 과정 또한 안전하다고 볼 수 있다.

- **서비스 세션키 공유**: 사용자 보안 연계 토큰의 내부 필드인 pair-wise public key $PK_{U,Ti}$로부터 $BKM_{U,Ti}$ 또는 KM_{Ti}를 계산하는 것은 Leighton-Micali[8] 논문에서 제시된 바와 같이 사용자와 T_i만이 할 수 있기 때문에, 서비스 세션키 $SK_{U,Ti}$ 또한 사용자 U와 Target Device T_i 간에 안전하게 공유되었다고 볼 수 있다.

4.2. 성능 분석

본 논문에서의 제안 프로토콜은 사용자의 보안 연계 토큰을 가지고 사용자 인증, 접근제어, 암호통신을 위한 키 공유까지 이루어지는데, 시스템 성능 분석을 위하여, Kerberos 시스템을 비교 대상으로 하였다.[2] Kerberos 시스템과 본 논문에서의 제안 프로토콜 모두 시스템 구성요소가 우선 4-Entity이고, 6-way 프로토콜로 인증 및 세션키가 공유된다. 또한 본 논문에서 제안한 프로토콜의 익명 인증서는 Kerberos 시스템의 TGT(Ticket Granting Ticket)와 대응되며, 보안 연계 토큰은 SGT(Service Granting Ticket)에 대응될 수 있다는 점에서 Kerberos 시스템과 유사성을 가진다. 또한 Kerberos 시스템과 제안 프로토콜 모두 동일한 암호 알고리즘(예: RC5, AES 등)을 적용할 수 있어, Kerberos 시스템과 성능 비교 대상으로 하였다. 표 3은 Kerberos의 메시지 교환 프로토콜을 요약 정리한 것이다.

성능 분석을 위하여, 본 논문에서는 AES-CBC

와 RC5-CBC로 메시지를 암/복호화 하였으며, MAC 알고리즘은 AES-CBC-MAC과 RC5-CBC-MAC을 사용하였다. 성능 측정을 위한 구현 환경은 사용자(클라이언트)를 위해서는 CPU는 ARM7 TDMI 24 Mhz, 개발 도구는 ADS(ARM Developer Suite) 1.2를 사용하였으며, AS, SAS, Target Device를 위해서는 Pentium Ⅳ 3.2Ghz, Windows XP SP2, MS-Visual C++ 6.0을 사용하였다. 표 4는 암호 알고리즘의 성능을 나타낸 것이며, '/' 좌우는 각각 ARM7TDMI와 Pentium IV 환경에서 1024바이트 데이터에 대한 암/복호화 및 MAC 연산에 대한 성능을 분석하였다.

표 3. Kerberos 메시지 교환 프로토콜 요약

(a) Authentication Service Exchange
(1) C → AS: $ID_c \parallel ID_{tgs} \parallel TS_1$ (2) AS → C: $E_{Kc}[K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]$ $Ticket_{tgs} = E_{Ktgs}[K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$
(b) Ticket-Granting Service Exchange
(3) C → TGS: $ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$ (4) TGS → C: $K_{c,tgs}[K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v]$ $Ticket_{tgs} = E_{Ktgs}[K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$ $Ticket_v = E_{Kv}[K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$ $Authenticator_c = E_{Kc,tgs}[ID_c \parallel AD_c \parallel TS_3]$
(c) Client / Server Authentication Exchange
(5) C → V: $Ticket_v \parallel Authenticator_c$ (6) V → C: $E_{Kc,v}[TS_5 + 1]$ $Ticket_v = E_{Kv}[K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$ $Authenticator_c = E_{Kc,v}[ID_c \parallel AD_c \parallel TS_5]$

표 5에서는 Kerberos 시스템과 본 논문의 제안 프로토콜의 시스템 구성 요소들이 암호 처리하는데 소요되는 시간을 측정하였으며, AS_1과 AS_2의 각각의 암호 처리 소요 시간을 합하여 나타내었다. 연산 ARM 7TDMI 환경에서는 암호 처리 모듈을 1,000회 반복 수행한 결과를 50회 측정하여, 평균시간 값을 계산한 것이며, Pentium Ⅳ 3.2Ghz 환경에서는 암호 처리 모듈을 1,000,000회 반복 수행한 결과를 50회 측정하여, 평균시간 값을 계산한 것이다. Kerberos 시스템에서의 ID_c, ID_{tgs}, AD_c, ID_v, TS_1, TS_2, TS_3, TS_4, TS_5는 각각 4바이트로 가정하였으며, $Lifetime_2$, $Lifetime_4$는 각각 8바이트, $Ticket_{tgs}$, $Ticket_v$는 40바이트를 암호화한 결과인 48바이트, $Authenticator_c$는 12바이트를 암호화한 결과인 16바이트로 가정하였다. 또한 암호화에 사용되는 키인 $K_{c,tgs}$와 $K_{c,v}$는 각각 16바이트로 가정하였다. Kerberos 시스템과 본 논문의 제안 프로토콜에서의 AS에서 수행하는 암호연산을 처리하는데 소요되는 시간은 각각 TGT(Ticket Grating Ticket)와 사용자 익명 인증서를 생성하는 시간이다. 또한 Kerberos 시스템의 TGS와 본 논문의 제안 프로토콜에서의 SAS에서 수행하는 암호연산을 처리하는데 소요되는 시간은 각각 SGT(Service Granting Ticket)와 사용자 보안 연계 토큰을 생성하는 시간이다.

표 5에서 볼 수 있듯이, 본 논문에서의 제안 프로토콜은 Kerberos System과 비교하여 성능면에서 다소 우수함을 알 수 있다. AES 알고리즘과 RC5 알고리즘을 사용했을 때의 속도 차이의 폭

표 4. 암호 알고리즘 성능

RC5-CBC	암/복호화 속도(Mbps)	암호용 라운드 키 생성 시간(msec)	복호용 라운드 키 생성 시간(msec)
	3.4 / 729	0.1816 / 0.002263	0.1816 / 0.002263
RC5-CBC-MAC	CBC-MAC 생성 속도(Mbps)		라운드 키 생성 시간(msec)
	3.86 / 756		0.1816 / 0.002263
AES-CBC	암/복호화 속도(Mbps)	암호용 라운드 키 생성 시간(msec)	복호용 라운드 키 생성 시간(msec)
	1.94 / 570	0.0291 / 0.000306	0.1037 / 0.000572
AES-CBC-MAC	CBC-MAC 생성 속도(Mbps)		라운드 키 생성 시간(msec)
	1.96 / 606		0.0291 / 0.000306

*CBC 암/복호화 보다 CBC-MAC 성능이 다소 좋게 나온 이유는 CBC 암/복호화 연산 수행시 암/복호화된 블록을 복사하는데 소요되는 오버헤드 때문이다.

표 5. 제안 프로토콜과 Kerberos system간의 실험 결과

	Kerberos System (RC5-CBC)	제안 프로토콜 (RC5-CBC-MAC)	Kerberos System (AES-CBC)	제안 프로토콜 (AES-CBC-MAC)
클라이언트 / 사용자 (ARM 7TDMI 24Mhz)	0.960(msec)	0.823(msec)	1.106(msec)	0.743(msec)
AS (Pentium Ⅳ 3.2Ghz)	0.006025(msec)	0.005892(msec)	0.002354(msec)	0.002256(msec)
TGS / SAS (Pentium Ⅳ 3.2Ghz)	0.008919(msec)	0.008325(msec)	0.004721(msec)	0.003269(msec)
서버 / Target Device (Pentium Ⅳ 3.2Ghz)	0.005365(msec)	0.005284(msec)	0.002869(msec)	0.001681(msec)

이 속도가 차이가 나는 이유는 AES 알고리즘 특성상, 복호화용 라운드 키를 생성하는 시간이 암호용 라운드 키를 생성하는 시간보다 많게는 3배가량 더 걸리기 때문에, 복호화 연산을 여러 번 수행하는 Kerberos 시스템에서는 성능 저하 요인이 있어, AES 알고리즘을 적용하였을 때는 본 논문의 제안 프로토콜이 보다 나은 성능상의 이점을 갖는다.

Ⅴ. 결 론

본 논문에서는 유비쿼터스 컴퓨팅 환경(스마트 공간)상의 다양한 디바이스들로부터 안전하게 서비스를 제공받을 수 있는 사용자 인증 및 키 동의 프로토콜을 제안하였다. 본 논문에서의 제안 프로토콜은 사용자의 개인 정보가 수록되지 않은 익명 인증서를 제안하고, 사용함으로써, 유비쿼터스 컴퓨팅 환경에서의 사용자 프라이버시를 보장 받을 수 있도록 하였다. 또한 본 논문의 제안 프로토콜은 공개키 암호 기술이 아닌 CBC-MAC 기반의 익명 인증서와 보안 연계 토큰을 이용하여, 사용자 인증 및 키 동의 프로토콜이 수행되기 때문에, 컴퓨팅 연산 능력이 비교적 떨어지는 디바이스들에 적용하기가 용이한 장점을 갖고 있으며, 시스템 성능 측면에서도 Kerberos 시스템과 비교하였을 때, 전반적으로 우수함을 알 수 있었다.

참 고 문 헌

[1] 권태경, 천정희, 김용대, 정찬주, "공개키 기반 구조에서 조건부 추적가능한 RSA 기반의 익명 인증서 연구", 제 16회 정보보호와 암호에 관한 학술대회(WISC2004), pp.183-194, 2004.

[2] B. Neumann and T. Ts'o,, "Kerberos: An Authentication Service for Computer Networks", IEEE Communications Magazine, 32(9): 33-38, September 1994.

[3] Kindberg, T. and Zhang, K., "Secure Spontaneous Device Association", Proceedings of UbiComp 2003: Ubiquitous Computing, vol. 2864 of Lecture Notes in Computer Science, Seattle, WA, USA, October 12-15, 2003. Springer Verlag.

[4] Kindberg, T. and Zhang, K.,"Validating and Securing Spontaneous Associations between Wireless Devices", Proceedings of 6th Information Security Conference(ISC'03), October 2003.

[5] Jalal Al-Muhtadi, Anand Ranganathan, Roy Campbell and M. Dennis Mickunas, "A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments", Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW' 02), 2002.

[6] Jalal Al-Muhtadi, R.Campbell, A.Kapadia, D.Micunas P.Naldurg and S.Yi, "Routing Through the Mist: Privacy-Preserving Communication in Ubi-

quitous Computing Environments", Proc. of the International Conference of Distributed Computing Systems (ICDCS 2002), pp.65-74, Vienna, Austria, July 3, 2002.
[7] Jalal Al-Muhtadi, R.Campbell, A.Kapadia, D.Micunas and P.Naldurg, "Socializing in the Mist: Privacy in Digital Communities", Technical Report UIUCDCS-R-2002-2271, April, 2002.
[8] Tom Leighton and Silvio Micali, "Secret-Key Agreement without Public-Key Cryptography", Advances in Cryptology CRYPTO 1993, 456-479, 1994.
[9] Anonymizer, http://www.anonymizer.com
[10] Freedom.net, http://www.freedom.net
[11] SafeWeb, http://www.safeweb.com

〈著者紹介〉

강 명 희 (Myung-hee Kang) 정회원
1996년 2월: 광운대학교 대학원 컴퓨터 과학과 졸업(이학석사)
2006년 2월: 광운대학교 대학원 컴퓨터과학과 졸업(공학박사)
1996년 2월~1998년 5월: 백두정보기술/주 EP&C팀 주임연구원
1998년 6월~2006년 2월: (주) 퓨쳐시스템 정보통신연구소 선임연구원
<관심분야> 인터넷 보안, 무선 네트워크 보안, 유비쿼터스 보안

유 황 빈 (Hwang-bin Ryou) 정회원
1989년 2월: 경희대학교 대학원 전자공학과 졸업(박사)
1981년 2월~현재: 광운대학교 컴퓨터공학부 교수
2004년 ~현재: 한국정보보호학회 비상임 부회장
<관심분야> 네트워크 보안, 유비쿼터스 정보보호