논문 2004-42CI-4-4

유비쿼터스 컴퓨팅 환경을 위한 익명성을 보장하는 사용자 인증 및 접근제어 모델

(An Anonymity-Preserving User Authentication and Authorization Model for Ubiquitous Computing Environments)

강 명 희*, 유 황 빈*

(Myung-Hee Kang and Hwang-Bin Ryou)

요 약

모바일 디바이스, PDA, 센서들의 확산은 실생활 공간이 지능적이고 향상된 서비스를 제공하는 스마트 공간으로 전환되는 유비쿼터스 컴퓨팅 환경이 구축될 수 있도록 하였다. 그러나 보안문제 특히 적절한 인증 및 접근 제어 기술의 부재로 말미암아, 실생활에서의 이러한 컴퓨팅 패러다임의 변화에 있어, 방해 요소가 되고 있으며, 또한 유비쿼터스 컴퓨팅 환경에서는 보안 대책을 마련하고, 사용자의 프라이버시 보장 또한 매우 중요하다. 본 논문에서는 유비쿼터스 컴퓨팅 환경을 위한 사용자 프라이버시를 위한 익명성이 보장되는 효율적인 사용자 인증 및 접근 제어 모델을 제안한다. 본 논문의 제안 모델은 공개키 암호 기술이 아닌 MAC 기반의 익명 인증서와 보안 연계 토큰을 이용하여, 사용자 인증 및 접근 제어를 수행함으로써, 컴퓨팅 연산 능력이 컴퓨팅 연산 능력이 비교적 떨어지는 디바이스들에 적합한 모델이다. 또한 암호 연산 처리 측면에서, Kerberos 시스템과 비교하였을 때, 전반적으로 우수함을 알 수 있었다.

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, the deployment of this computing paradigm in real-life is disturbed by poor security, particularly, the lack of proper authentication and authorization techniques. Also, it is very important not only to find security measures but also to preserve user privacy in ubiquitous computing environments. In this paper, we propose efficient user authentication and authorization model with anonymity for the privacy-preserving for ubiquitous computing environments. Our model is suitable for distributed environments with the computational constrained devices by using MAC-based anonymous certificate and security association token instead of using public key encryption technique. And our proposed protocol is better than Kerberos system in sense of cryptographic computation processing.

Keywords: Anonymity, Authorization, Privacy, User Authentication

I. 서 론

IT 환경이 빠른 속도로 발전하면서 대두되고 있는 유비쿼터스 환경은 기존의 컴퓨터 통신 환경과는 다른 형태를 취할 것이다. 그러므로 기존의 컴퓨팅 환경에서는 사용자가 기계를 사용하기 위한 방법들을 학습하여 정보를 습득하고 이를 제 3자에게 전달하는 방법이 아니라 기계들이 스스로 정보를 수집하고 분석하여 처리할 수 있는 환경이 도래할 것으로 예상된다. 이러한 유비쿼터스 컴퓨팅 환경은 우리의 생활 속으로 스며들어 사용자들이 의식하지 못하는 주위의 장치들의 도움을

* 정회원, 광운대학교 컴퓨터 과학과
(Dept. of Computer Science, Kwangwoon University)
※ 이 논문은 2004년도 광운대학교 교내연구비에 의하여 연구되었음
접수일자: 2005년5월12일, 수정완료일: 2005년7월6일

받으며 좀 더 편리한 생활을 추구할 수 있도록 하는 순기능이 있다. 그러나 우리들의 일상생활과 컴퓨터가 융합됨에 따라, 컴퓨터가 악의적인 서비스 거부 공격을 받았을 경우에는 우리의 일상생활 모두에 악영향을 미치는 역기능을 초래할 수 있다. 또한 유비쿼터스 컴퓨팅 환경에서의 사용자 프라이버시 보호 문제가 해결되지 않고서는 유비쿼터스 감시 체제 Ubiquitous Surveillance Infrastructure)가 구축되는 위험성에 노출되어 있다. 본 논문에서는 유비쿼터스 컴퓨팅 환경(스마트 공간)상의 다양한 디바이스들로부터 안전하게 서비스를 제공받으면서도 사용자 개인의 프라이버시를 보호할 수 있는 익명성이 보장되는 사용자 인증 및 접근제어 모델을 제안한다.

본 논문의 구성은 다음과 같다. Ⅱ 장에서는 관련 연구 내용을 기술하며, Ⅲ 장에서는 본 논문에서 제안한 익명성이 보장되는 인증 및 접근제어 모델을 다루며, Ⅳ 장에서는 제안 모델에 대한 시스템 분석을 하고, Ⅴ 장에서는 결론을 맺는다.

Ⅱ. 관련 연구

1. Secure Device Association Mechanism

Kinderg[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 가정하고, 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 편리성에 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 바람직 할 것으로 생각된다.

2. Authenticated Key Agreement Protocol

공개키 방식을 이용한 키 공유 프로토콜은 분명히 대칭키 방식의 키 공유 프로토콜에 비해서 여러 가지 좋은점이 있는 것은 분명하지만, 공개키 방식의 키 공유 프로토콜은 구현이 대칭키 방식에 비하여, 어려우며, 유비쿼터스 컴퓨팅 환경에서의 다양한 tiny device(컴퓨팅 연산 능력이 적은)에는 탑재되기가 힘든 상황이다. Leighton-Micali[6] 논문에서는 대칭키 방식의 암호 기술을 이용하면서도, 공개키 방식의 암호가 갖는 장점을 살릴 수 있는 효과적인 키 공유 프로토콜을 제안하였는데, 이 논문에서 제안한 키 공유 프로토콜을 tiny device에 적용하여 보안 시스템을 구성한다면 유용할 것으로 보인다.

Ⅲ. 제안 모델

1. 인증 및 접근제어 모델

본 논문에서는 유비쿼터스 컴퓨팅 환경에서의 다양한 디바이스들과의 안전한 통신을 위하여, 보안 연계 토큰 Security Association Token)을 제안하고, 익명성을 보장하는 인증 및 접근제어 모델을 제안한다. 본 논문에서 제안한 보안 연계 토큰은 Kerberos 시스템에서 사용되는 티켓 개념과 유사하지만, Leighton-Micali[6] 논문에서의 제안기법을 응용하여, 사용자 인증 및 접근제어를 수행한다. 다음 그림 1은 본 논문에서 제안하는 인증 및 접근 제어 모델을 나타낸 것이다.

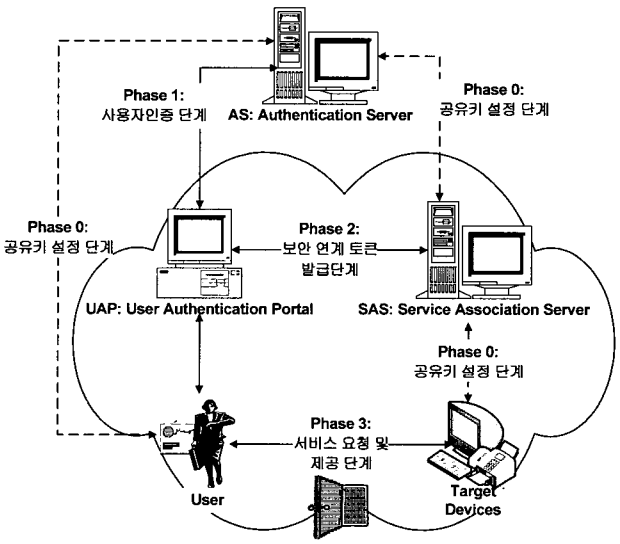

그림 1. 유비쿼터스 환경을 위한 사용자 인증 및 접근 제어 모델

Fig. 1. User authentication and authorization model for ubiquitous computing environments.

본 논문에서 제안한 인증 및 접근 제어 모델의 시스템 구성요소 및 역할은 다음과 같다.

- AS (Authentication Server): 사용자와 SAS (Service Association Server에 대한 개체를 인증하는 서버이다.
- SAS (Service Association Server): 스마트 공간상에서 사용자에게 보안 연계 토큰 (SAT: Security Association Token)을 발급하여, 사용자의 Target Device에 대한 접근제어를 수행하며, 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 공간상에 사용자에게 유비쿼터스 서비스를 제공하는 디바이스이다.

본 논문에서 제안한 인증 및 접근 제어 모델은 크게 3단계로써, 사용자 인증단계, 보안 연계 토큰 발급 단계, 서비스 요청 및 제공 단계로 구성되며, 전처리 과정으로써, 각 시스템 구성요소의 공유키 설정단계가 있다.

가. 기호 설명

기호	설명
$\|\|$	A $\|\|$ B라고 했을 때, A와 B를 연접하는 기호
$\oplus$	A $\oplus$ B라고 했을 때, A와 B를 XOR 연산 기호
$\doteq$	A $\doteq$ B라고 했을 때, A와 B의 값을 비교하는 기호
CMAC(A, B)	키를 A로 하여, 메시지 B를 CBC-MAC 연산을 하는 기호 (CMAC 함수의 키 길이는 16바이트 이며, 메시지의 길이는 16의 배수가 되도록 패딩 한다.)
K_U	사용자 U와 AS간의 공유하는 마스터키
K_{SAS}	SAS와 AS간의 공유하는 마스터키
K_{Ti}	Target Device T_i와 SAS간의 공유하는 마스터키
$PK_{U,SAS}$	사용자와 SAS간의 Pair-wise Public Key
BK_U	사용자와 SAS간의 공유하는 스마트 공간 기본키
$PK_{U,Ti}$	사용자와 Target Device T_i 사이의 Pair-wise Public Key
$SK(U, T_i)$	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

나. 공유키 설정 단계

1. 사용자와 AS간의 공유키 설정: 사용자와 AS간의 사용자 인증을 위해 공유하는 마스터키(K_U)를 설정하며, K_U는 AS가 발행한 익명 인증서에 대한 pair-wise digital signature를 사용자가 검증하는데 사용된다.

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

3. Target Device T_i와 SAS와의 공유키 설정: SAS와 Target Device T_i와 최초에 공유하게 되는 마스터키(K_{Ti})를 설정하며, K_{Ti}는 SAS에 대한 pair-wise digital signature를 T_i가 검증하는데 사용된다.

2. 사용자 인증 단계

유비쿼터스 컴퓨팅 환경이 구축되어 있는 스마트 공간에 사용자가 입장하였을 때, 맨 처음 사용자 인증을 수행하는 단계이다.

1. 사용자는 자신의 인증 디바이스(예: PDA, Smart Badge, Mobile Phone, Password, etc.)를 이용하여, 스마트 공간상의 가장 가까우면서, 사용가능한 UAP에 접근하여, 그림 2와 같이 스마트 공간의 SAS 서버 ID와 자신의 사용자 ID를 전송하고, 그에 대한 인증값 $AUTH_U$를 계산하여, AS에 사용자 인증을 요청한다.

2. AS는 사용자 U와의 공유키 K_U를 이용하여, $AUTH_U$에 대한 유효성을 검증하여 $AUTH_U$가 유효하면 해당 사용자에게 익명 인증서를 발급하여, UAP에

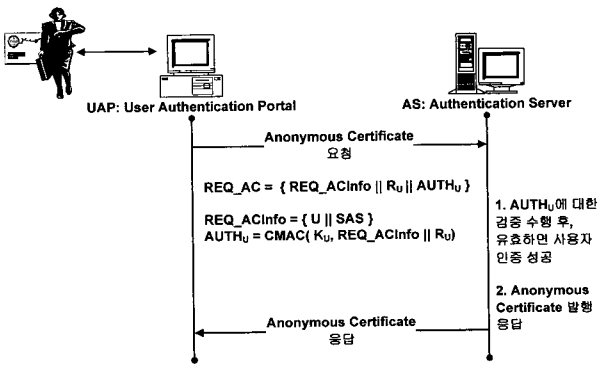

그림 2. 1단계: 사용자 인증
Fig. 2. Phase 1: User authentication.

표 1. 익명 인증서 구조
Table 1. Anonymous certificate profile.

필드		내용
toBeSignedACInfo	Anonymous_ID	AS가 관리하는 사용자에 대한 Random ID (4바이트)
	SAS_ID	AS가 관리하는 SAS ID (4바이트)
	Validity	사용자 익명 인증서의 유효기간 (8 바이트)
	Nonce_AS(R_{AS})	AS가 생성한 Nonce (8바이트)
	$PK_{U,SAS}$	U와 SAS간의 Pair-wise Public Key (16 바이트) < 생성방법은 먼저 KM_U = CMAC(K_U, R_{AS}), KM_{SAS} = CMAC(K_{SAS}, R_{AS}) 계산 후, $PK_{U,SAS}$ = KM_U ⊕ KM_{SAS} 계산 >
$AUTH_U$		사용자 U가 K_U를 가지고 익명인증서의 유효성을 검증할 수 있도록 AS가 CMAC(K_U, toBeSignedACInfo)를 계산하여, 생성한 인증 값 (16 바이트)
$AUTH_{SAS}$		SAS가 K_{SAS}를 가지고 익명인증서의 유효성을 검증할 수 있도록 AS가 CMAC(K_{SAS}, toBeSignedACInfo)를 계산하여, 생성한 인증 값 (16 바이트)

전송하고, UAP는 사용자 익명 인증서를 사용자에게 전달한다. 익명인증서 구조는 표 1과 같다.

3. AS로부터 발급받은 익명 인증서를 사용자는 AS와 공유하고 있는 K_U를 이용하여 익명 인증서의 유효성을 검증할 수 있다. 유효성 검증 과정은 다음과 같다. CMAC(K_U, toBeSignedACInfo)를 먼저 계산한 다음, $AUTH_U$와 비교하여 값이 같으면 유효한 것으로 본다.

$$CMAC(K_U, toBeSignedACInfo) \stackrel{?}{=} AUTH_U$$

4. 사용자는 익명 인증서의 $PK_{U,SAS}$로부터 SAS와 사용자간의 스마트 공간 기본키 BK_U를 생성한다. BK_U의 생성과정은 다음과 같다. KM_U = CMAC(K_U, R_{AS})를 계산한 후, KM_{SAS} = $PK_{U,SAS}$ ⊕ KM_U를 계산한 다음, KM_U의 상위 8바이트와 KM_{SAS}의 상위 8바이트를 서로 연접하여 BK_U를 생성한다.

KM_U = $KM_{U[0..7]}$ || $KM_{U[8..15]}$ = CMAC(K_U, R_{AS})
KM_{SAS} = $KM_{SAS[0..7]}$ || $KM_{SAS[8..15]}$ = $PK_{U,SAS}$ ⊕ KM_U
BK_U = $KM_{U[0..7]}$ || $KM_{SAS[0..7]}$

3. 보안 연계 토큰 발급 단계

SAS가 사용자의 익명 인증서를 검증한 후, 보안 연계 토큰을 발급하는 단계이다. 보안 연계 토큰 발급 과정은 다음과 같으며, 표 2는 보안 연계 토큰의 프로파일을 나타낸 것이다.

1. UAP는 SAS에 사용자에 대한 익명 인증서를 이용하여, 사용자의 보안 연계 토큰을 발급 요청한다.

표 2. 보안 연계 토큰 프로파일
Table 2. Security association token profile.

필드		내용
toBeSignedSATInfo	SAT_No	SAT 일련번호 (4 바이트)
	Issuer_ID	SAT 발급자의 ID (4 바이트)
	Holder_ID	SAT 사용자의 익명 ID (4 바이트)
	Validity	보안 연계 토큰의 유효기간 (8 바이트)
	Nonce_SAS(R_{SAS})	SAS가 생성한 Nonce (8 바이트)
	Target_ID	Target Device의 ID (4 바이트)
	$PK_{U,Ti}$	U와 T_i 간의 Pair-wise Public Key (16바이트) < KM_{Ti} = CMAC(K_{Ti}, R_{SAS}), $BKM_{U,Ti}$ = CMAC(BK_U, R_{SAS})를 계산 후, $PK_{U,Ti}$ = KM_{Ti} ⊕ $BKM_{U,Ti}$ 계산 >
AUTH_Holder		사용자 U와 SAS간의 스마트 공간 기본키 BK_U를 이용하여 SAS가 생성한 인증 값 (16 바이트)
AUTH_Target		T_i와 SAS간의 공유키 KT_i를 이용하여 SAS가 생성한 인증 값 (16 바이트)

2. SAS는 K_{SAS}를 이용하여, AS가 생성한 사용자의 익명 인증서에 대한 유효성을 검증한다. 유효성 검증 과정은 다음과 같다. CMAC(K_{SAS}, toBeSignedACInfo)를 먼저 계산하고, $AUTH_{SAS}$와 비교하여, 동일하면 유효한 것으로 간주한다.

$$CMAC(K_{SAS}, toBeSignedACInfo) \stackrel{?}{=} AUTH_{SAS}$$

3. 사용자의 익명 인증서가 유효하면, SAS는 익명 인증서의 $PK_{U,SAS}$로부터 SAS와 사용자간의 스마트 공간 기본키 BK_U를 생성한다. BK_U의 생성과정은 사용자가 생성한 방법과 거의 동일하게 다음과 같이 계산된다.

KM_{SAS} = $KM_{SAS[0..7]}$ || $KM_{SAS[8..15]}$ = CMAC(K_{SAS}, R_{AS})
KM_U = $KM_{U[0..7]}$ || $KM_{U[8..15]}$ = $PK_{U,SAS}$ ⊕ KM_{SAS}
BK_U = $KM_{U[0..7]}$ || $KM_{SAS[0..7]}$

4. toBeSignedSATInfo 필드는 SAT 일련번호, Issuer ID 정보(SAS ID), Holder ID(사용자의 익명 ID), 유효기간, SAS가 생성한 Nonce 값, Target Device ID와 사용자 U와 Target Device T_i간의 pair-wise public key $PK_{U,Ti}$ 필드로 구성되며, $PK_{U,Ti}$의 생성 과정은 다음과 같다.

KM_{Ti} = CMAC(K_{Ti}, R_{SAS})
$BKM_{U,Ti}$ = CMAC(BK_U, R_{SAS})
$PK_{U,Ti}$ = KM_{Ti} ⊕ $BKM_{U,Ti}$

5. SAS는 스마트 공간 기본키 BK_U를 이용하여

AUTH_Holder 필드를 계산한다. AUTH_Holder 필드의 계산 방법은 다음과 같다.

$$AUTH\_Holder = CMAC(BK_U, toBeSignedSATInfo)$$

6. SAS는 Target Device T_i와 공유하고 있는 K_{Ti}를 이용하여 AUTH_Target 필드를 계산한다. AUTH_Target 필드의 계산 방법은 다음과 같다.

$$AUTH\_Target = CMAC(K_{Ti}, toBeSignedSATInfo)$$

7. toBeSignedSATInfo 필드, AUTH_Holder필드, AUTH_Target 필드를 연접하여 사용자 보안 연계 토큰을 구성하고, UAP에 전달한다.

$$SAT = \{ toBeSignedSATInfo \,||\, AUTH\_Holder \,||\, AUTH\_Target \}$$

8. UAP는 사용자에게 보안 연계 토큰을 전달하고, 사용자는 자신의 보안 연계 토큰을 스마트 공간 기본키를 이용하여 유효성을 검증한다. 유효성 검증과정은 스마트 공간 기본키 BK_U를 이용하여 $CMAC(BK_U, toBeSignedSATInfo)$를 계산하고, AUTH_Holder 필드와 비교하여, 동일하면 유효한 것으로 간주한다.

$$CMAC(BK_U, toBeSignedSATInfo) \stackrel{?}{=} AUTH\_Holder$$

4. 서비스 요청 및 제공 단계

본 단계에서는 사용자가 SAS로부터 발급받은 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_U를 이용하여, 서비스 세션키 $SK(U, T_i)$를 생성한다. $SK(U, T_i)$의 생성과정은 먼저 $BKM_{U,Ti} = CMAC(BK_U, R_{SAS})$를 계산하고, $KM_{Ti} = PK_{U,Ti} \oplus BKM_{U,Ti}$를 계산한 다음, $BKM_{U,Ti}$와 KM_{Ti}의 상위 8바이트를 연접하여, $SK(U, T_i)$를 생성한다.

$$BKM_{U,Ti} = BKM_{U,Ti[0..7]} \,||\, BKM_{U,Ti[8..15]} = CMAC(BK_U, R_{SAS})$$
$$KM_{Ti} = KM_{Ti[0..7]} \,||\, KM_{Ti[8..15]} = PK_{U,Ti} \oplus BKM_{U,Ti}$$
$$SK(U, T_i) = BKM_{U,Ti[0..7]} \,||\, KM_{Ti[0..7]}$$

2. 사용자는 서비스 요청 메시지 REQ_SERV를 Target Device T_i에 전송한다. REQ_SERV 메시지 생성 과정은 다음과 같다.

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

3. Target Device T_i는 자신의 ID 정보와 K_{Ti}를 이용하여 사용자의 SAT에 대한 유효성을 검증한다. SAT에 대한 유효성 검증 과정은 다음과 같다.

$$CMAC(K_{Ti}, toBeSignedSATInfo) \stackrel{?}{=} AUTH\_Target$$

4. Target Device T_i는 사용자의 SAT를 이용하여, $SK(U, T_i)$를 생성한다. $SK(U, T_i)$의 생성 과정은 사용자가 생성하는 방법과 거의 동일하게 다음과 같이 계산된다.

$$KM_{Ti} = KM_{Ti[0..7]} \,||\, KM_{Ti[8..15]} = CMAC(K_{Ti}, R_{SAS})$$
$$BKM_{U,Ti} = BKM_{U,Ti[0..7]} \,||\, BKM_{U,Ti[8..15]} = PK_{U,Ti} \oplus KM_{Ti}$$
$$SK(U, T_i) = BKM_{U,Ti[0..7]} \,||\, KM_{Ti[0..7]}$$

5. $SK(U, T_i)$를 이용하여, $AUTH_U$의 유효성을 검증하여, 유효하면, 해당 서비스를 제공한다. 유효성 검증 과정은 다음과 같다.

$$CMAC(\, SK(U, T_i),\, R_U \,) \stackrel{?}{=} AUTH_U$$

Ⅳ. 시스템 분석

1. 안전성 분석

본 논문에서 제안한 인증 및 접근제어 모델에서는 AS는 TTP(Trust Third Party)로서, 신뢰한다고 가정한다. 사용자의 익명 인증서(AC: Anonymous Certificate)는 사용자와 AS만이 공유하고 있는 키 K_U와 AS와 SAS만이 공유하고 있는 키 K_{SAS}를 가지고 생성되므로, TTP인 AS 만이 생성할 수 있다. 또한 사용자 익명 인증서의 검증은 사용자만이 $AUTH_U$를 검증할 수 있으며, SAS만이 $AUTH_{SAS}$를 검증할 수 있기 때문에, 사용자 익명인증서에 대한 생성 및 검증은 안전하다고 볼 수 있다. 그리고 익명 인증서 내부 필드인 pair-wise public key $PK_{U,SAS}$로부터 KM_U 와 KM_{SAS}를 계산해내는 것은 Leighton-Micali[6] 논문에서 제시된 바와 같이 사용자와 SAS만이 할 수 있기 때문에, 스마트 공간 기본키 BK_U 또한 안전하다.

사용자 보안 연계 토큰(SAT: Security Association Token)은 사용자와 AS만이 공유되는 스마트 공간 기본키 BK_U와 SAS와 Target Device T_i 만이 공유하고 있는 키 K_{Ti}를 가지고 생성 되므로, SAS만이 생성할 수 있다. 또한 보안 연계 토큰에 대한 검증은 사용자만이 AUTH_Holder 필드를 검증할 수 있으며, Target Device T_i만이 AUTH_Target 필드를 검증할 수 있기

때문에, 보안 연계 토큰에 대한 생성 및 검증은 안전하다고 볼 수 있다. 그리고 보안 연계 토큰의 내부 필드인 pair-wise public key $PK_{U,Ti}$로부터 $BKM_{U,Ti}$ 또는 KM_{Ti}를 계산하는 것은 Leighton-Micali[6] 논문에서 제시된 바와 같이 사용자와 T_i만이 할 수 있다. 따라서 사용자와 Target Device T_i간에 공유되는 세션키 $SK(U, T_i)$ 또한 안전하다.

2. 성능 분석

본 논문에서 제안한 인증 및 접근제어 모델은 사용자의 보안 연계 토큰을 가지고 사용자 인증, 접근제어, 암호통신을 위한 키 공유까지 이루어지는데, 시스템 성능 분석을 위하여, 본 논문에서의 제안 모델과 유사한 구조인 Kerberos 시스템[2]을 비교 대상으로 하였다. 다음 표 3은 Kerberos의 메시지 교환 프로토콜을 요약 정리한 것이다.

성능 분석을 위하여, 본 논문에서는 AES CBC 모드로 메시지를 암/복호화 하였으며, MAC 알고리즘은 AES-CBC-MAC을 사용하였다. 성능 측정을 위한 구현 환경은 사용자(클라이언트)를 위해서는 CPU는 ARM7TDMI 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. 다음 표 4는 암호 알고리즘의 성능을 나타낸 것이며, '/' 좌우는 각각 ARM7TDMI와 Pentium IV의 환경에서 1024바이트 데이터에 대한 암/복호화 및 MAC 연산에 대한 성능을 분석하였다.

표 5는 Kerberos 시스템과 본 논문에서의 제안 모델 간의 수행되어야할 암호연산 블록수를 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. 표 5에서는 Kerberos 시스템과 본 논문의 제안 모델의 시스템 구성 요소들이 암호 처리하는데 소요되는 평균 시간을 측정한 것이다. 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.

표 5에서 볼 수 있듯이, 본 논문에서 제안한 모델은 Kerberos System과 비교하여 성능 면에서 클라이언트/사용자는 41%, AS는 11%, TGS/SAS는 31%, 서버/Target Device는 41%가 우수함을 알 수 있다.

표 3. Kerberos 메시지 교환 프로토콜 요약
Table 3. Summary of Kerberos Message Exchanges.

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

표 4. 암호 알고리즘 성능
Table 4. Performance of cryptographic algorithms.

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
	1블록(16 바이트) 암/복호화 시 소요시간(msec)		
	0.0648 / 0.000233		
AES-CBC-MAC	CBC-MAC 생성 속도(Mbps)	라운드 키 생성 시간(msec)	
	1.96 / 606	0.0291 / 0.000306	
	1블록(16 바이트) CBC-MAC 연산 시 소요시간(msec)		
	0.0634 / 0.000213		

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

표 5. 제안 모델과 Kerberos system간의 실험 결과
Table 5. Experimental results between kerberos system and our model.

	Kerberos System (암/복호화)	제안 모델 (CBC-MAC)
클라이언트 / 사용자 (ARM 7TDMI 24Mhz)	1.106(msec)	0.658(msec)
AS (Pentium IV 3.2Ghz)	0.002354(msec)	0.002106(msec)
TGS / SAS (Pentium IV 3.2Ghz)	0.004721(msec)	0.003269(msec)
서버 / Target Device (Pentium IV 3.2Ghz)	0.002869(msec)	0.001681(msec)

V. 결 론

본 논문에서는 유비쿼터스 컴퓨팅 환경(스마트 공간) 상의 다양한 디바이스들로부터 안전하게 서비스를 제공받을 수 있는 사용자 인증 및 접근제어 모델을 제안 하였다. 본 논문에서의 제안 모델은 사용자의 개인 정보가 수록되지 않은 익명 인증서를 제안, 사용함으로써, 유비쿼터스 컴퓨팅 환경에서의 사용자 프라이버시를 보장 받을 수 있도록 하였다. 또한 본 논문의 제안모델은 공개키 암호 기술이 아닌 MAC 기반의 익명인증서와 보안 연계 토큰을 이용하여, 사용자 인증 및 접근 제어를 수행함으로써, 컴퓨팅 연산 능력이 비교적 떨어지는 tiny device들에 적용하기가 용이한 장점을 갖고 있으며, 시스템 성능 측면에서도 Kerberos 시스템과 비교하였을 때, 전반적으로 우수함을 알 수 있었다.

참 고 문 헌

[1] 권태경, 박해룡, 이철수 "공개키 기반 구조에 기반한 익명게시판 기술 현황" 정보보호학회학술지, 제 14권, 제6호, 1-13쪽, 2004년 12월

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

[3] T. Kwon, J. Cheon, and Y. Kim, "Anonymous Certificate and its Application", in preparation and available form http://dasan.sejong.ac.kr/~tkwon/research/pseudonym2.pdf, 2004.

[4] 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.

[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] Tom Leighton and Silvio Micali, "Secret-Key Agreement without Public-Key Cryptography", Advances in Cryptology CRYPTO 1993, 456-479, 1994.

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

[8] C. H. Lim, "Authenticated Key Distributed for Security Services in Open Networks", Future system Technical Report, May 1997.

[9] T. Kindberg and A. Fox, "System Software for Ubiquitous Computing", IEEE Pervasive Computing, January-March, 2002, pp. 70-81.

[10] Weiser, M., "The Computer for the Twenty-First Century", Scientific American, vol.256, No. 3, pp. 94-104, September 1991.

[11] Ha, W., et at. "Confusion of Physical Space and Electronic Space: Ubiquitous IT Revolution and the Third Space", Korean Electronic Times, 2002.

[12] G. Banavar and A. Bernstein, "Software infrastructure and design challenges for ubiquitous computing applications", Communications of the ACM, vol. 45(12), pp. 92-6, 2002.

저 자 소 개

강 명 희(정회원)

1996년 광운대학교,컴퓨터과학과 이학석사

2001년~광운대학교,컴퓨터과학과 박사과정

1998년 5월 백두정보기술/주, 주임연구원

1998년 6월~(주)퓨쳐시스템, 선임연구원

<주관심분야 : 무선네트워크 보안, 유비쿼터스 보안>

유 황 빈(정회원)

1975년 인하대학교 전자공학과 공학사 졸업

1977년 연세대학교 대학원 공학석사 졸업

1989년 경희대학교 대학원 공학박사 졸업

1981년~현재 광운대학교 컴퓨터소프트웨어 교수

<주관심분야 : 멀티미디어통신 및 응용, 네트워크 보안, RFID>