# 유비쿼터스 컴퓨팅 시스템 정보 보호

주한규*

## 1. 서론

유비쿼터스 컴퓨팅이란 말 그대로 어디에서나 계산이 가능함, 즉 모든 사물에 컴퓨터 시스템이 내재되어 사용됨을 의미한다. 그리고 이들 컴퓨터 시스템은 독립적으로 존재하지 않고 네트워크에 의하여 서로 연결되어 필요한 정보를 주고받는다. 유비쿼터스 컴퓨팅은 아직 완전한 의미로 현실화된 개념은 아니다. 그러나 중요한 미래의 정보 기술로 여러 곳에서 연구가 진행되고 있다.

유비쿼터스 컴퓨팅이라는 용어는 Mark Weiser에 의하여 처음 사용되었으며 Scientific American에 기고되었던 그의 논문이 유비쿼터스 컴퓨팅의 개념을 제시하고 있다[1][2]. [1]에서 그는 "가장 심오한 기술은 사라지는 것이다. 그러한 심오한 기술은 일상의 생활과 구별될 수 없을 때까지 일상의 생활에 흡수가 된다." 라고 말하고 있다.

유비쿼터스 컴퓨팅의 개념이 일반화되면 우리 일상의 모든 생활이 컴퓨터에 의하여 도움을 받을 수 있을 것으로 보인다. 동시에 다양한 정보에 대한 정보보호의 문제점 또한 필요할 것으로 생각된다. 본 논문에서는 유비쿼터스 컴퓨팅 시스템을 위한 정보 보호의 기법에 대하여 기술한다. 2절에서는 정보 보호 서비스 및 기법에 대하여 기술한다. 3절에서는 유비쿼터스 컴퓨팅 시스템에서의 고려 사항에 대하여 알아본다. 4절에서는 유비쿼터스 시스템의 정보 보호를 위한 기법에 대하여 기술하고 5절에서는 결론을 맺는다.

## 2. 정보 보호 서비스 및 기법

정보보호는 기밀성(Confidentiality, Privacy), 인증(Authentication), 무결성(Integrity), 부인방지(Non-Repudiation), 접근제어(Access Control), 가용성(Availability) 등의 서비스를 제공한다. 이러한 서비스를 위하여 암호화, 전자서명, 인증정보 교환 등을 비롯한 다양한 기법이 사용된다. 이 절에서는 기밀성, 인증, 무결성, 부인 방지의 개념과 이들 서비스를 달성하기 위한 기법에 대하여 알아본다.

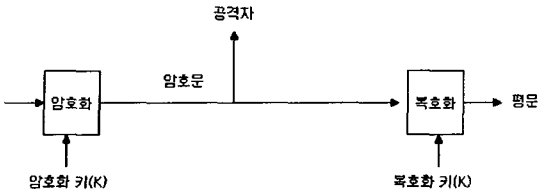
### 2.1 기밀성

서로 교신하는 객체들 간의 통신이 제 3자에 의하여 도청되지 않도록 하는 서비스가 기밀성이다. 기본적으로 암호화 기법을 이용하여 달성할 수 있다. 암호화 기법은 크게 대칭키 암호 기법과 공개키 암호 기법으로 나누어 생각할 수 있다.

#### 2.1.1 대칭키 암호 기법

교신하고자 하는 두 객체가 동일한 키를 공유하여 암호화와 복호화를 하는 경우 대칭키 암호 기법이다. 대칭키 암호화를 사용하는 경우 키를 공유하고 있는 객체들은 암호화된 내용을 복호화하여 볼 수 있지만 키를 가지지 않은 제 3자는 암호화된 내용을 알 수 없다. 이를 그림으로 나타내면 그림 1과 같다.

* 한림대학교 정보통신공학부




[그림 1] 대칭키 암호화 기법

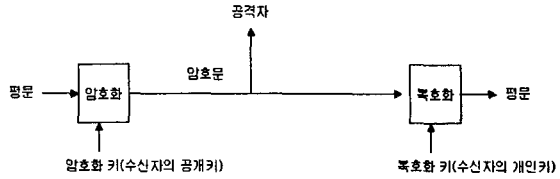
즉 암호화 알고리즘을 E, 복호화 알고리즘을 D, 평문 메시지를 m, 공유된 키를 k, 그리고 암호화된 메시지를 c 라고하면 다음의 관계가 성립한다.

$$c = E(m, k)$$
$$m = D(c, k)$$

이러한 대칭키 암호화 기법으로는 DES[3], AES[4] 등이 있다. 대칭키 암호화 기법은 수행되는 속도가 빠르다. 그러나 교신하는 두 객체가 동일한 키를 가지고 있어야 한다는 어려움이 따른다. 또한 교신하고자하는 상대가 복수일 경우, 교신 대상자 수만큼 키를 유지해야 하므로 관리해야하는 키 관리에 어려움이 따른다.

### 2.1.2 공개키 암호 기법

공개키 암호 기법은 송신자의 암호화 키와 수신자의 복호화 키가 서로 다른 암호 기법이다. 메시지의 수신자는 공개키와 개인키라고 불리는 한 쌍의 키를 생성한다. 공개키는 일반에게 공개하고 개인키는 외부에서 접근할 수 없도록 비밀스럽게 보관한다. 공개키와 개인키는 연관성을 가지고 있으나 공개키로부터 개인키를 생성할 수는 없다. 메시지 송신자는 수신자로부터 획득한 수신자의 공개키를 이용하여 보호되어야 할 메시지를 암호화 한 후 전송한다. 메시지의 수신자는 자신의 개인키를 이용하여 암호문 메시지로부터 평문 메시지를 복원하여 내용을 확인한다. 이를 그림으로 나타내면 그림 2와 같다.




[그림 2] 공개키 암호화 기법

즉 암호화 알고리즘을 E, 복호화 알고리즘을 D, 평문 메시지를 m, 수신자의 공개키를 $k_P$, 수신자의 개인키를 $k_S$, 그리고 암호화된 메시지를 c 라고하면 다음의 관계가 성립한다.

$$c = E(m, k_P)$$
$$m = D(c, k_S)$$

공개키 암호 기법을 사용하는 경우 대칭키 암호 기법의 키 교환 및 관리 문제를 쉽게 해결할 수 있다. 그러나 현재 안전하게 사용될 수 있는 공개키 암호 기법은 모두 암호화와 복호화에 많은 시간을 필요로 한다. 따라서 많은 양의 자료를 공개키 암호 기법을 이용하여 암호화하는 것은 현실적으로 불가능하다. 따라서 대칭키의 키만 공개키를 이용하여 암호화 한 후 전송하고 메시지는 대칭키를 이용하여 암호화하거나, Diffie-Hellman에 의하여 제안된 키 교환 방법[5]에 의하여 대칭키를 공유한 후 대칭키 암호 기법을 사용하는 것이 일반적이다. 대표적인 공개키 암호 기법에는 RSA[6], ElGamal[7] 등이 있으며 Diffie-Hellman 키 교환 기법은 키 공유만을 위하여 사용 된다[5].

## 2.2 무결성

무결성이란 메시지가 송신자로부터 수신자에게 전송되는 동안 의도되지 않은 변경이 발생하지 않음을 보장하는 것이다. 무결성을 지원하는 기법에는 CRC(Cyclic Redundancy Check), 해쉬(Hash)함수, MAC(Message Authentication Code), 전자서명 등이 있다.

CRC는 전송 중 의도되지 않은 변경이 일어나는 경우를 이를 탐지하기 위하여 개발된 기법이

다. 의도적으로 메시지를 변경하고자 하는 경우 가능 할 수도 있으므로 악의적인 공격에는 취약함을 보인다.

해쉬함수는 CRC와 유사하나 조금 더 암호학적으로 안전한 개념을 이용하여 설계되었다. 가변길이의 메시지로부터 고정길이의 해쉬값을 계산한다. 일방향성의 특성을 가져서 메시지로부터 해쉬값을 계산하기는 쉬우나 해쉬값으로부터 메시지를 찾는 것은 불가능하다. 이론적으로 복수의 메시지로부터 동일한 해쉬값이 생성될 수 있으나 현실적으로 동일한 해쉬값을 생성하는 복수의 메시지를 불가능하다. 원본 메시지와 해쉬값이 모두 안전하지 못한 경롤 전달되는 경우, 악의적인 공격자가 원본 메시지를 변경한 다음 해쉬값을 변경된 메시지에 의하여 계산하면 그 변경을 탐지하는 것은 불가능하다. MD5[8], SHS[9] 등이 이에 해당한다.

MAC은 해쉬 함수에 송신자와 수신자가 공유하는 공유키를 추가 입력으로 사용하는 경우라고 생각할 수 있다. 송신자와 수신자가 키를 공유하고 그 키를 가진 사람만이 MAC 값을 생성하고 확인할 수 있으므로 악의적인 공격에 안전하다. 대칭키 암호에서와 같이 송신자와 수신자가 MAC 키를 공유하는 기법과 키를 안전하게 관리할 필요가 있다.

전자 서명은 공개키 암호 기법의 응용이다. 메시지의 송신자는 공개키와 개인키 쌍을 생성하여 개인키는 비밀스럽게 보관하고 공개키는 일반에 공개한다. 메시지의 송신자는 원본 메시지의 해쉬값을 구한 후, 개인키를 이용하여 전자 서명을 한 후 메시지와 전자 서명을 함께 전송한다. 수신자는 메시지 수신 후 송신자의 공개키를 이용하여 올바른 서명인지 확인하여 메시지의 변경이 있었는지 확인한다. 전자서명 기법에는 RSA[6], DSS[10] 등이 있다.

## 2.3 인증

인증은 교신하고 있는 상대방이 올바른 상대방인가하는 상대 인증과 수신한 메시지가 실제 전송자가 전송한 메시지인가를 확인하는 메시지 근원 인증으로 나누어 생각할 수 있다. 메시지 근원 인증은 무결성과 유사하다. MAC과 전자서명을 이용하여 메시지 근원 인증을 할 수 있다. 우리가 흔히 사용하는 (user_id/password) 사용이 전형적인 상대 인증의 방법이다. (user_id/password)를 이용한 인증은 사전 공격(dictionary attack) 등에 취약하다. 도전-응답 기법(Challenge-response)이 (user_id/password)기법보다는 안전하다고 할 수 있다. 도전-응답 기법은 대칭키를 이용하는 방법과 공개키를 이용하는 방법이 있을 수 있다. 대칭키를 이용하는 경우는 키 공유 및 키 관리의 어려움이 있으므로 공개키를 사용하는 것이 일반적이다.

공개키는 암호화와 인증, 그리고 전자서명 등에 매우 유용하게 사용된다. 그러나 공개키의 사용은 중간자 공격 (man-in-the-middle attack)에 취약하다. 예를 들어 암호화를 이용한 기밀성 제공에서의 문제점을 살펴보면 다음과 같다. A가 B에게 비밀스러운 메시지를 보낼 필요가 있다. 이 경우 B가 공개키/개인키 쌍($Pub_B$, $Priv_B$)을 선택하여 공개키를 공개하고 A는 B의 공개키를 이용하여 메시지를 암호화한 후 B에게 전송하면 된다. 그런데 공격자 C가 B의 공개키를 획득하고 가공의 공개키/개인키 쌍($Pub_C$, $Priv_C$)을 생성하여 $Pub_C$를 B의 공개키라고 알리는 경우 A는 $Pub_C$를 B의 공개키로 여겨 $Pub_C$를 이용하여 메시지를 암호화하여 전송할 것이다. C는 $Ppriv_C$를 이용하여 암호화된 문서를 복호화하여 읽은 후 $Pub_B$를 이용하여 암호화하여 B에게 전달할 수 있다. 이러한 경우 기밀성을 달성할 수 없게 된다. 공개키의 중간자 공격 (man-in-the-middle attack)은 모든 공개키를 사용하는 서비스에 적용될 수 있다. 즉 상대방의 공개키를 획득한 사람은 획득한 공개키가 실제 상대방의 공개키인지 확인하는데 어려움이 있다.

이러한 문제를 해결하기 위하여 현재 가장 널

리 사용되는 방법은 공개키 기반 구조(PKI: Public Key Infrastructure)이다. 즉, 인증서(certificate) 및 인증기관(certification authority)의 사용을 위한 기반 구조를 갖추는 것이다. 공개키/개인키 쌍 생성자가 자신의 공개키/개인키 쌍을 생성한 후, 자신의 공개키를 인증기관으로부터 인증을 받는다. 인증기관은 공개키의 생성자와 공개키를 확인하여 인증서를 발부한다. 인증서를 획득하여 현재의 공개키 소유자는 자신이 가지고 있는 공개키가 실제 자신의 공개키임을 확신시킨다. 공개키를 획득한 사람도 공개키의 인증서를 확인하여 실제 당사자의 공개키임을 확인한다.

### 2.4 부인 방지

메시지를 송신한 사람이 메시지의 송신을 부인할 수 있다. 또한 메시지의 수신자도 메시지의 수신을 부인할 수 있다. 송신자가 메시지를 송신할 때 전자 서명을 하도록 요구하여 송신자의 부인은 방지할 수 있다. 제 3자의 공증을 거쳐 수신자의 부인을 방지할 수 있다.

## 3. 유비쿼터스 컴퓨팅 시스템에서의 고려 사항

유비쿼터스 컴퓨팅 시스템은 기존의 컴퓨팅 시스템과 다른 특성들을 가지고 있다. 이러한 특성들을 고려하여 그에 따른 정보보호 해법이 필요하다. 유비쿼터스 컴퓨팅 시스템은 매우 다양한 응용 분야를 가질 수 있으며 각각은 자신의 특성을 가진다. 그리고 각각의 응용에 따른 정보보호 정책과 기법이 필요하다. 그러나 대부분의 유비쿼터스 시스템이 가지는 일반적인 특성도 있으며 이와 관련한 일반적인 정보보호 고려 사항에 대한 고찰이 먼저 필요하다.

기본적으로 유비쿼터스 컴퓨팅 시스템에 참여하는 대부분의 장치는 초소형 프로세서를 장치 내에 포함하게 된다. 이러한 초소형 프로세서의 특성 때문에 이들 개체는 저성능 수행 속도를 가지며 소규모의 저장 장치 그리고 한정적인 전지를 전원으로 가지게 된다.

유비쿼터스 컴퓨팅 시스템은 ad hoc 네트워크의 특성을 포함한다. 반드시 ad hoc 네트워크만 존재하는 것은 아니나 많은 개체가 ad hoc 네트워크를 구성하며 일부 개체는 인터넷과 같은 기존의 확립된 네트워크의 노드로 참여하기도 한다.

## 4. 유비쿼터스 컴퓨팅 시스템의 정보 보호

유비쿼터스 컴퓨팅 시스템의 기존의 컴퓨팅 시스템과 다른 특성들을 갖는다. 이들을 고려하여 그에 따른 정보보호 기법이 필요하다.

### 4.1 인증

유비쿼터스 시스템은 ad hoc 네트워크를 포함하는 것이 일반적이다. Ad hoc 네트워크에서는 온라인 서버의 존재를 가정하기 어렵다. 따라서 기존의 공개키 기반 구조를 이용하여 인증서 사용이 어려워지게 된다. 인증서를 사용하는 경우 그 객체의 인증은 어렵지 않으나 인증서를 사용하지 못함으로 객체의 인증에 어려움이 있다.

유비쿼터스 시스템에는 매우 많은 수의 다양한 개체가 존재한다. 대부분의 개체는 낮은 처리 능력, 적은 저장 공간, 저전력 등의 제약 사항을 가지고 있다. 이들이 모두 공개키를 사용하여 인증되는 것은 어려움이 있다.

이러한 제약사항을 염두에 두고 the resurrecting duckling 기법이 제안되었다[11]. The resurrecting duckling은 기본적으로 주종관계(master-slave, ownership)을 정의하는 방법으로 생각할 수 있다. 개체 A가 다른 개체 B와 주종관계에 있게 되면 개체 A는 주인인 개체 B에만 복종한다. 오리가 알에서 깨어 처음 본 개체를 자신의 어미로 인식하고(imprinting) 그 어미를 따르게 되는 것과 유사한 기법이다. 하나의 개체가 처음 생성된 후 비밀 키를 넘겨주는 개체를 주인으로 인식하고 그 개체에 복종한다. 이 기법은 개체가 두 개의 상태를 가지도록 한다. 하나는 imprintable(unborn) 상태이고 다른 하나는 imprinted(alive) 상태이다. 개체는 처음 생성되어

imprintable 상태에 있게 된다. imprintable 상태에 있는 개체가 다른 개체로부터 비밀키를 넘겨받으면 imprinted 상태가 된다(imprinting). imprinted 상태에 있는 개체는 자신에게 비밀키를 넘겨준 개체와 주종관계를 가지며 그 개체에만 복종한다. Imprinted 상태에 있는 개체는 죽음(death)에 의하여 다시 imprintable 상태로 돌아갈 수 있다. 죽음의 행위는 여러 이유에 있을 수 있으며 응용에 맞추어 죽음 정책을 만들어야 한다. 일반적으로 기본적으로 주인의 명령에 의하여 죽음이 일어날 수 있으며 그 외에도 특정 시간이 지난 후 또는 특정한 일의 완료 후에도 죽음이 일어날 수 있도록 정책을 정할 수 있다. 명시된 죽음 정책 외에 imprinted 개체가 imprintable 개체로 돌아가는 죽음이 일어나도록 하는 암살(assassination)은 비경제적이어야 한다.

The resurrecting duckling은 주종관계에 있는 개체간의 인증을 효율적으로 할 수 있는 기법으로 제안되었다. 그러나 이 기법은 주종관계 이외의 관계에서는 어떻게 인증하는지 기술되어있지 않다. 동일한 주인을 가지고 있는 개체들이 현재 네트워크에서 주인과 교신 가능하면 주인을 통하여 상호 교신 가능할 것이다. 그러나 주인과의 네트워크가 항상 연결되어 있다고 볼 수 없다. 또한 동일한 주인을 가지지 않는 개체들 간에 상호 인증할 수 있는 방법에 대한 언급이 없다.

또한 본인 증명(예, 인증서)이라는 하나의 속성에 의하여만 인증을 하고 인증된 개체에 대하여만 신뢰하던 기존에 방식 외에 다양한 신뢰 정보를 이용하는 신뢰 관리의 개념이 제안되었다[12]. 유비쿼터스 시스템 환경에서는 개체의 물리적인 문맥, 즉 개체의 위치 또는 시각, 또한 신뢰 관계에 중요한 요인이 된다. 따라서 기존의 본인 증명 외에 물리적 문맥을 추가한 속성 벡터를 유지한다. 속성 벡터를 종합하여 일정 정도를 넘으면 상대방을 신뢰한다. 제안된 신뢰 관리의 개념은 전체적인 신뢰도를 이용한다는 개념이다 특정 응용에 곧바로 적용될 수 있는 기법은 아니다.

## 4.2 기밀성

기밀성은 기본적으로 암호화를 통하여 달성된다. 암호화 알고리즘은 유비쿼터스 컴퓨팅 개념 이전부터 널리 사용되어 왔으며 이러한 암호화 알고리즘을 다양한 저성능 프로세서에 어떻게 적용하느냐가 기본적인 관건이다.

공개키 암호 알고리즘은 대칭키 암호 알고리즘에 비하여 매우 느리다. 그런 이유로 현재의 컴퓨팅 환경에서도 문서의 암호화 같은 경우에는 공개키 암호 알고리즘을 이용하지 않는 것이 일반적이다. 이러한 공개키 알고리즘을 유비쿼터스 컴퓨팅 환경의 초소형 저성능 프로세서에서 있는 그대로 사용할 수 없음은 당연하다. 초소형 프로세서를 사용하는 센서 네트워크에서의 공개키 사용에 대하여 많은 연구가 진행되고 있으며 효율적인 계산을 이용하면 공개키도 적어도 키 교환을 위해서는 사용 가능하다고 주장된다[13][14]. 공개키 암호를 사용하는 경우 빠른 연산을 위하여 타원곡선암호[15]를 사용함이 일반적이다.

공개키 암호가 반드시 필요한 경우 외에는 대칭키 암호를 사용한다. 대칭키 암호의 경우 특히 AES와 같은 블록 암호를 이용하여 대량의 자료를 암호화하는 경우 빠른 시간에 암호화 할 수 있다[16]. 한편 유비쿼러스 컴퓨팅에 사용되는 개체는 일반적으로 전지를 사용하므로 최소한의 전력을 사용할 필요가 있다. 이러한 필요성은 암호화를 위한 저전력의 전용 하드웨어와 그에 따른 암호 알고리즘의 개발을 요구할 수도 있다[11].

## 4.3 무결성

객체간의 일대일 통신을 위해서는 간단히 MAC(Message Authentication Code)을 이용하여 무결성과 메시지 근원 인증을 달성할 수 있다. 그러나 이 경우 교신 당사자가 카를 공유하고 있어야 하며 교신 당사자의 수가 많아지면 키 관리의 어려움이 따른다. 공개키를 이용한 전자 서명은 키 공유 및 관리의 어려움이 거의 없고 부인 방지의 효과도 포함하는 장점이 있으나 많은 처리

시간을 필요로 하므로 저성능의 소형 프로세서를 사용하는 유비쿼터스 컴퓨팅 환경에서의 사용은 부적절하다.

일대다 통신(멀티캐스트)의 경우 무결성 및 메시지 근원 인증을 위하여 TESLA (Time Efficient Stream Loss-tolerant Authentication) [17] 기법이 제안되었다. 이 기법은 송신자와 수신자의 시간 동기화 이후 가능하다. TESLA는 다섯 가지 방법이 제안되었으나 이해를 위하여 가장 기본적인 방법을 살펴보도록 한다. 기본적인 접근 방법은 송신자는 하나의 패킷($P_i$)을 전송할 때 자신만이 아는 키($K_i$)를 이용하여 $P_i$의 MAC을 계산하여 패킷과 MAC을 함께 전송한다. 그리고 이 MAC 키($K_i$)는 다음 순서의 패킷($P_{i+1}$)을 전송할 때 함께 전송된다. 따라서 수신자는 $P_{i+1}$ 패킷을 수신한 다음에야 $P_i$ 패킷의

### 4.4 익명성

유비쿼터스 시스템은 익명성(anonymity) 서비스를 필요로 하기도 한다. 예를 들어 본인 인식이 가능한 교통 카드를 사용하는 경우를 생각하여 보자. 이 경우 사용자는 교통 카드를 단순히 지불만을 위하여 사용하나, 사용자의 위치가 노출되어 사생활의 침해를 입을 수 있다. 이러한 문제를 해결하기위한 익명성 제공 기법이 필요하며 현재 널리 연구되고 있다. 위치 정보를 숨기기 위한 연구[18], 사용자와 행동, 장치, 그리고 위치의 연관관계를 숨기기 위한 연구[19] 등 다양한 연구가 진행되고 있다.

## 5. 결론

유비쿼터스 컴퓨팅의 개념이 일반화되면 우리 일상의 모든 생활이 컴퓨터에 의하여 도움을 받을 수 있을 것으로 보인다. 또한 유비쿼터스 컴퓨팅 시스템은 동시에 다양한 정보보호 서비스 필요하게 된다. 본 논문에서는 일반적인 유비쿼터스 컴퓨팅 시스템이 공유하는 정보보호 서비스와 기법들에 대하여 기술하였다. 유비쿼터스 컴퓨팅은 매우 다양한 응용 분야를 가질 수 있다. 그리고 각각의 응용은 자신의 고유한 정보보호 서비스를 필요로 하게 될 것이다.

## 참고 문헌


[1] Mark Weiser, "The Computer for the Twenty-First Century", Scientific American, Vol. 265, No. 3, pp. 94-104, 1991.

[2] M. Weiser, R. Gold, and J. S. Brown, "The origins of ubiquitous computing research at PARC in the late 1980s", IBM Systems Journal, Vol. 38, No. 4, pp. 693-696, 1999.

[3] FIPS 46, "Data Encryption Standard", National Bureau of Standards, 1977.

[4] FIPS 197, "Advanced Encryption Standard", National Institute of Standards and Technology, 2001.

[5] W. Diffie and M. Hellman, "Multiuser Cryptographic Techniques", AFIPS Conference Proceedings, 1976.

[6] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Communications of the ACM. Vol. 21, No. 2, pp.120-126, 1978.

[7] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, Vol. 31, pp. 469-472, 1985.

[8] rfc 1321, "The MD5 Message-Digest Algorithm", ietf, 1992.

[9] FIPS 181-1, "Secure Hash Standard", National Institute of Standards and Technology, 1995.

[10] FIPS 186, "Digital Signature Standard", National Institute of Standards and Technology, 1994.

[11] F. Stajano, Security for Ubiquitous Computing, Wiley, 2002.
[12] N. Shankar and W. Arbaugh, "On Trust for Ubiquitous Computing", Workshop on Security in Ubiquitous Computing, 4th International UBICOMP, 2002.
[13] D. J. Malan, M. Welsh, and M. D. Smith, "A public Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography", First IEEE International Conference on Sensor and Ad Hoc Communications and Network, Oct. 2004.
[14] B. Arazi, I. Elhanany, O. Arazi, and H. Qi, "Revisiting Public Key Cryptography for Wireless Sensor Network," IEEE Computer Magazine, pp. 85-87, Nov. 2005.
[15] A. J. Menezes, Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, 1993.
[16] Brian Gladman, "The AES Algorithm in C++", http://fp.gladman.plus.com/AES/index.htm
[17] A. Perrig, R. Canetti, D. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels", in Proc. of IEEE Security and Privacy Symposium S&P2000, May 2000.
[18] U. Hengartner and P. Steenkiste, "Protecting People Location Information", Workshop on Security in Ubiquitous Computing, 4th International UBICOMP, 2002.
[19] A. Zugenmaier and A. Hohl, "Anonymity for Users of Ubiquitous Computing", Workshop on Security in Ubiquitous Computing, 5th International UBICOMP, 2003.

주한규
1988년 한림대학교 전자계산학과 졸업(학사)
2004년 Arizona State University 졸업(석사)
2008년 Arizona State University 졸업(박사)
1999년 ~ 2000년 한국전자통신연구원 선임연구원
2000년 ~ 현재 한림대학교 부교수
관심분야: 소프트웨어공학, 정보보호