한국항행학회 논문지 제12권 제1호 2008년 2월

임베디드 컴퓨팅 환경에서 브로드캐스트 암호화를 위한 효율적인 키 분배

An Efficient Key Distribution for Broadcast Encryption at Embedded Computing Environment

이덕규*, 김태훈**, 여상수***, 김석수**, 박길철**, 조성언****

Deok-Gyu Lee*, Tai-Hoon Kim**, Sang-Soo Yeo***, Seok-Soo Kim**, Gil-Cheol park** and Seong-Eon Cho****

요 약

브로드캐스트 암호화 기법은 공개된 네트워크상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다. 사용자가 디지털 정보를 획득하기 위해서는 브로드캐스터가 키를 생성하고 분배하는 과정이 필요하며, 사용자가 탈퇴나 새로운 가입 시에 효율적인 키 갱신이 필요하게 된다. 임시적인 컨퍼런스 환경은 각 이동 디바이스들 사이에 대하여 전송이 가능하고, 잦은 위치 변화에 따라 키 정보가 유동적으로 변하는 특성으로 인해 키 관리의 어려움이 있다. 이에 본 논문에서는 임베디드 컴퓨팅 환경에 적용하여 특정한 공간에서 이동 디바이스가 이용될 때 효율적인 키 생성과 키 갱신을 하도록 제안한다.

Abstract

Broadcast encryption schemes are applied to transmit digital informations of multimedia, software, Pay-TV etc. in public network. User acquires message or session key to use key that broadcaster transmits, broadcaster need process that generation and distribution key in these process. Also, user secession new when join efficient key renewal need. In this paper, introduce about efficient key generation and distribution, key renewal method. The temporary conference environment base structure against an each mobile device wild gap. Without the transmission possible, it follows infrequent location change and with the quality where the key information change flow. Thus, in this paper, in order to apply to the embedded computing environment and the key generation and the efficient key renewal are done when the mobile device is used of the specify space it proposes.

Key words : Imbedded Computing, Broadcast Encryption, Key Distribution

I. 서 론

최근 IT 기술은 디바이스들의 저가, 소형화, 고성능화에 따라 제품 경쟁력의 핵심이 H/W에서 뿐만 아니라 S/W로 이동하는 대변혁이 진행 중에 있다. 임베디드 소프트웨어는 마이크로프로세서 위에 내장되어

* 한국전자통신연구원 보네트워크보안연구팀(ETRI, HomeNetwork Security Research Team)
** 한남대학교 멀티미디어학부(Dept. of Multimedia, Hannam Univ.)
*** 일본 큐슈대학교 (Dept. of Computer Science and Communication engineering, Kyushu Univ.)
**** 순천대학교 정보통신공학부(Division of Iformation Communication, Sunchon National Univ.)
· 교신저자 (Corresponding Author) : 조성언
· 접수일자 : 2008년 1월 18일

산업 및 군사용 제어기기, 디지털정보 가전기기, 자동센서장비 등의 기능을 다양화하고 부가가치를 높이는 핵심 소프트웨어로서, 다시 말해, 임베디드 소프트웨어는 우리가 일상에 쉽게 접하는 휴대폰, TV, 세탁기, 기차, 비행기, 엘리베이터 등의 제품 안에 내장된 임베디드 시스템에서 하드웨어를 제외한 나머지 부분이라고 말할 수 있다. 특히 이러한 임베디드 컴퓨팅에서는 보안의 문제가 중요하게 대두 될 수 있는데, 최근 브로드캐스트 암호화 기법은 공개된 네트워크상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다. 디지털 기술이 발달함에 따라 많은 수의 디지털 콘텐츠가 생겨나고 보급되고 있다. 이러한 디지털 정보들은 컴퓨터 및 기타 장치를 사용하여 쉽게 복제할 수 있고 그에 따른 피해가 발생하고 있다. 예를 들어, CD나 디스켓 안에 저장되어 있는 디지털 정보들은 CD writer나 디스켓을 통해 쉽게 복제할 수 있으며, 인터넷을 통해 디지털 정보에 관련하여 쉽게 다운로드 받고 이 정보들을 쉽게 다른 사람에게 줄 수 있다. 이러한 상황에 대해 브로드캐스트 암호화 기법은 공개된 네트워크상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다[1][3].

키를 분배하는 방식 중에 공개키 방식을 이용하는 방식은 세션키를 암호화하기 위한 그룹 암호키는 하나이고 복호하기 위한 키는 여러 개의 키를 이용하는 것으로, 서버는 공개키를 이용하여 세션키를 암호화하고, 각 사용자는 서로 다른 개인키를 이용하여 복호화 할 수 있도록 되어 있다. 브로드캐스트 암호화 기법에서 중요한 것은 오직 사전에 허가받은 사용자만이 디지털 정보를 얻을 수 있어야 한다는 것이다. 브로드캐스트 메시지가 전달되면 권한이 있는 사용자들은 자신이 사전에 부여받은 개인키를 이용하여 디지털 정보를 얻게 된다. 브로드캐스트 암호화에 있어 가장 중요한 것은 키 생성, 분배, 갱신이다.

제안 방식에서는 임베디드 컴퓨팅의 특성에 의해 회의장 등과 같은 특정한 공간에서 이동 디바이스를 사용하여 소규모 그룹 생성하는 환경에서 키를 생성하고, 분배하는 기법을 제안한다. 브로드캐스트 암호화 모델은 두 가지 모델로 나눠볼 수 있는데, 서버가 사용자를 예측하여 키를 생성하는 모델과 사용자가 자신의 정보를 제공하여 키를 생성하는 모델이 존재한다. 이때 전자의 방식인 서버가 예측하여 개인의 키를 생성하는 것이 아니라 사용자로부터 정보를 제공받아 중앙 역할을 하는 디바이스가 전체적으로 키를 생성하여 분배하고 이를 이용하여 통신하는 방식을 제안하도록 한다.

본 논문은 2장에서 기존방식을 살펴보고, 이를 바탕으로 3장에서 제안 방식의 각 단계에 관하여 살펴본다. 또한 4장에서는 비교 분석을 통하여 제안 방식에 대해 고찰하며, 마지막 5장으로 결론을 맺도록 한다.

Ⅱ. 기존방식

A. Narayanan[2]등이 2003년에 RSA 기반을 둔 악의적인 사용자의 추적 능력을 가진 실용적인 유료 TV 스킴을 제안하였다. 악의적인 사용자를 추적하는 방식은 다음의 원리를 이용하여 제안하고 있다.

> 임의의 $s(<t)$ 벡터들의 선형 조합으로 주어진 n 개의 $(t+1)$ 차 벡터 $X_1, X_2, \ldots, X_n$ 을 구성하면, 사용된 정확한 벡터들을 높은 확률로 알아낼 수 있다.

m 개의 채널을 브로드캐스트 하는 콘텐츠 제공자와 n 명의 사용자가 있다고 가정한다. 프로토콜은 다음의 단계로 나뉘어져 있다. Setup, AddStream, AddUser, Subscribe, Unsubscribe, Broadcast, Receive 7개의 알고리즘으로 구성된다.

사용자의 채널 수신여부는 $m \times n$ 행렬인 $Subsc$ 로 나타내며 사용자 U_i가 S_i에 등록되어 있으면 $Subsc[i, j]$ 에 해당하는 값은 1을 가지고 , 등록되어 있지 않으면 $Subsc[i,j]$의 값은 0을 가진다.

Step 1. Setup : 콘텐츠 제공자는 다음과 같은 변수값을 생성한다. $N = pq, R, d_r \le R\{1,2,\ldots,\varphi(N)\}$ 이때 $1 \le r \le 4+t$ 이고 p 와 q 는 큰 값의 소수이며, R 은 랜덤값이다. p , q , d 는 콘텐츠 제공자의

비밀키로 구성되며, 콘텐츠 제공자는 공개키 N을 공개한다.

Step 2. AddStream : 시스템에 새로운 채널 스트림 S_i를 추가하기 위해 콘텐츠 제공자는 큰 위수를 갖는 임의의 $g_i \in Z_N^*$를 선택한다. $Subsc[i,j]$는 모든 j에 대해 0의 값을 가지도록 설정하고 g_i 값은 공개하지 않는다.

Step 3. AddUser : 새로운 사용자 U_j 가입시키려면 콘텐츠 제공자는 $\sum_{r=1}^{t+4} e_{rj} d_r = R\Phi(N)+1$를 만족하는 $(e_{1j}, e_{2j}, \ldots, e_{(t+4)j})$을 선택한다. 이때 U_j는 비밀키를 안전한 메모리에 저장한 복호기(Set-Top Terminal)를 받게 되고 U_j의 비밀키는 $(e_{1j}, e_{2j}, \ldots, e_{(t+4)j})$이 된다.

Step 4. Subscribe : 사용자 U_j가 서비스 S_i를 구독하면 콘텐츠 제공자는 사용자 U_j에게 g_i^{e1j}를 전송하고, 이 때 사용자는 등록되었음을 알리는 $Subsc[i,j]$=1로 변경한다.

Step 5. Unsubscribe : 사용자 U_j가 서비스 S_i의 구독을 중지하면 콘텐츠 제공자는 $Subsc[i,j]=0$으로 설정한다. AddStream 알고리즘에서 했던 것과 같이 새로운 g_i 값을 선택하여 $Subsc[i,j]=1$인 모든 사용자들에게 g_i^{e1j}를 전송한다.

Step 6. Broadcast : 메시지 M을 채널 스트림 S_i에 전송하려면 콘텐츠 제공자 $\Phi(N)$과 서로 소인 랜덤값 x를 선택한다. 암호화된 데이터 $C=(x, C_1, C_2, \ldots, C_{t+4})$를 전송하며, 이때 $C_1 = M^{d1} g_i^x, C_2 = M^{d2}, C_{t+4} = M^{dt+4}$이다.

Step 7. Receive : 채널 스트림 S_i로 전송되는 암호화된 데이터 $C=(x, C_1, C_2, \ldots, C_{t+4})$로 복호화하기 위하여 사용자 U_j는 비밀키 $(e_{1j}, e_{2j}, \ldots, e_{(t+4)j})$를 이용하여 다음 $\left(\prod_{r=1}^{t+4} C_r^{e_{rj}}\right) / g_i^{xe_{1j}}$을 계산한다. 사용자 U_j는 다음과 같은 과정을 거쳐 콘텐츠 M을 다음과 같이 복원한다. $\left(\prod_{r=1}^{t+4} C_r^{e_{rj}}\right) / g_i^{xe_{1j}} = M^{R\Phi(N)+1} = M$

Narayanan 스킴의 문제점은 하나의 채널당 $(x, C_1, C_2, \ldots, C_{t+4})$의 통신량이 필요하다는 것이다. 통신량은 채널의 개수와 연관되어 있기 때문에 채널이 늘어나면 통신량도 증가하는 문제가 발생할 수 있게 된다. 또한 콘텐츠 제공자는 불법 사용자 U_j를 찾는다하더라도 U_j를 제외한 모든 가입자에게 다시 새로운 비밀키를 배포해야만 U_j를 탈퇴시킬 수 있다

Ⅲ. 임베디드 컴퓨팅 환경에 적합한 키 분배 방식 제안

임베디드 컴퓨팅 환경에서 키를 생성하고 분배하는 경우, 사용자의 탈퇴 혹은 신규 사용자의 가입에 따른 효율적인 키 갱신을 위해 다음과 같은 두 가지 모델을 기술하고, 두 가지 중에서 임베디드 컴퓨팅 환경에 안전한 효율적인 키 분배 방식을 제안한다. 우선 첫 번째 모델은 서버가 사용자의 동의 없이 사용자를 예측하여 키를 생성하고 분배하여 암호화 통신이 이루어지는 모델이며, 두 번째 모델은 서버가 사용자의 동의를 받아야만 브로드캐스팅 암호화키를 생성할 수 있는 모델이다. 그림 1과 2에서는 최초 사용자 정보를 제공하여 키를 생성하는 부분에서만 상이하며, 이후 브로드캐스트 암호화 메시지 전달과 신규 요청 및 탈퇴에 대해서는 동일하게 진행된다. 본 논문에서는 임베디드 컴퓨팅 환경의 특성을 고려하여 서버가 사용자의 정보를 획득하여 키를 분배하는 모델을 기본으로 삼는다. 이와 같은 두 가지 적용 모델에 대해 살펴보고 임베디드 컴퓨팅 환경에서 효율적이고 안전한 키 분배를 위한 요구사항에 대해 기술한다. 두 가지 모델과 요구 사항으로부터 임베디드 컴퓨팅 환경에서 키를 생성, 분배하는 방식을 제안한다.

3-1 적용 모델

브로드캐스트 암호화는 다음과 같이 두 가지 모델을 기반으로 할 수 있지만, 본 방식에서는 서버가 사용자로부터 정보를 제공받아 키를 생성하는 방식으로 제안하도록 한다.

우선 사용자와 서버간의 정보를 이용하여 키를 생성하고 분배하는 모델(그림 1 참조)은 전송되는 방식에서 차이가 존재하지만 제공되는 메시지가 그룹이 결정되고 난 후에 사용자들에게 메시지가 전송된다는 점에서는 기존 멀티캐스트 방식과 유사하다. 키 생성 과정에서 사용자가 참여하므로 키 생성 시간에 사용자의 참여 시간이 포함될 수 있으며, 기존 사용자의 탈퇴와 신규 과정에 따라 사용자의 최초 참여 때 부여받은 키에 부가된 갱신 정보를 이용하여 이뤄지게 된다. 사용자 정보를 이용하는 모델에 있어 문제점은 키를 생성하는데 있어 참여하고자 하는 모든 사용자의 정보가 수집되어야 올바른 키를 생성할 수 있으므로 정보를 수집하는 소요시간이 많이 발생하게 된다.

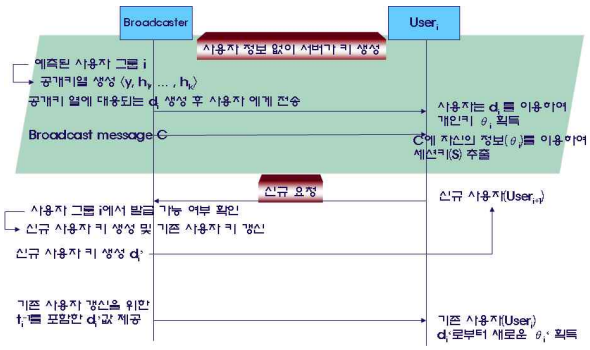

그림 1. 사용자 정보 제공 모델
Fig. 1. Providing User information model

또 다른 모델은 앞에서 살펴본 그림 1의 적용 모델과는 다르게 서버가 키를 예측하여 생성하는 모델로 서버가 단독으로 예측한 키를 사용자가 요청하면 전송하게 된다.(그림 2 참조) 이 모델은 사용자의 동의 없이 서버가 모든 사용자의 키를 생성하게 됨으로써 빠른 생성과 빠른 갱신이 가능하다. 하지만 서버가 악의적인 목적 혹은 서버가 공격의 대상이 되었을 경우 많은 취약점을 내포하고 있다. 많은 취약점을 내포하고 있음에도 불구하고 사용자에 대해 빠르게 키를 생성 분배할 수 있다는 면에서 DMB나 위성 TV등에서 많이 활용되고 있는 모델이다.

홈 네트워크와 같은 임베디드 컴퓨팅 환경에서 분배하는 경우, 환경의 특성상 참여자의 인원이 한정되어 있고, 정해진 사용자가 입장한다. 따라서 이러한 특성에 따라 임베디드 컴퓨팅 환경에 적합한 모델은 중앙에 위치한 서버(중앙 디바이스)가 사용자의 정보를 받아 키를 생성 분배하는 방식을 이용하면 서버는 사용자에 대해 예측할 필요가 없고, 사용자에 대한 정확한 키 생성이 이뤄질 수 있다.

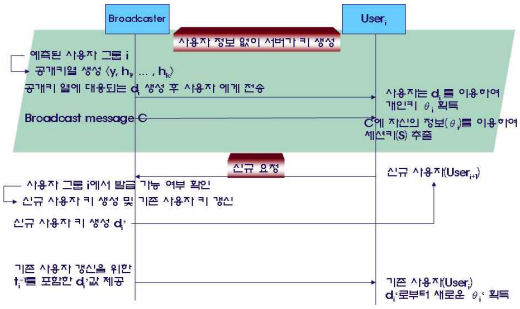

그림 2. 사용자 정보 비제공 모델
Fig. 2. Non-providing User information model

3-2 보안 요구 사항

다음은 본 방식에서 임베디드 컴퓨팅 환경에서 효율적이고 안전한 키 분배를 위한 보안 요구 사항에 대해 살펴본다.

- 사용자 참여: 임베디드 컴퓨팅 환경 구성에 있어 사용자가 정해져 있는 특정한 키 생성을 위해 사용자의 참여는 이뤄져야 한다.
- 키 갱신: 각 사용자에게 제공된 키는 다른 사용자가 탈퇴하더라도 자신의 키를 갱신하여 사용할 수 있어야하며, 갱신은 전체 사용자에 있어 원활히 이뤄질 수 있어야 한다.
- N회 탈퇴: 사용자의 탈퇴는 자유롭게 이뤄져야 하며, 전체 사용자 중 N명에 대한 탈퇴도 가능해야 한다.
- 키의 연속성: 사용자의 키는 연속적으로 사용될 수 있어야 하며, 새로운 사용자의 참여, 기존 사용자의 탈퇴 등과 같은 여러 사항에서도 키는 사용될 수 있어야 한다.
- 초기 예측 오류에 따른 재연산: 초기 사용자의

예측에 따른 전체 키 생성은 필요하지 안하야 한다.

- 불법 사용자 추적(Traitor Tracing): 불법적인 사용자가 발생하였을 경우 불법 사용자에 대한 추적이 이뤄져야 한다.

3-3 능동적 비밀 분산(Proactive Secret Sharing)

능동적 비밀 분산은 기존의 공격자 모델보다 좀 더 강력한 공격자 모델인 이동 공격자(Mobile Adversary) 모델에 대하여 비밀 분산 작업을 안전하게 수행하게 하기 위해 제안된 개념이다. 이동 공격자 모델은 공격자가 특정 주기를 두어 매 주기마다 프로토콜에 침입하는 것이다. 다라서 프로토콜 내에서 특정 주기 내에 현재 주기 안에 있는 공격자를 고려해 주지 않는다며, 다음 주기에서의 공격자들은 이전 주기에서 수집한 프로토콜에 관한 정보를 이용하여서 좀 더 강력한 공격을 할 수 있다.

능동적 비밀 분산은 이러한 이동 공격자가 존재하는 상황에서 안전한 비밀 분산을 수행할 수 있도록 제안된 개념이다. 1995년 Jarecki는 능동적인 비밀 분산을 이루기 위해서 주기적인 공유의 갱신을 통한 해결 방법을 제시하였다. 이것은 임의의 비밀 정보에 대하여 이미 비밀 분산을 수행하여 공유들을 분배받은 상태에서 수행하는 것으로 새로운 공유 갱신값을 더하여 기존 공유에 대한 안전도를 높이는 방법이다. 공유 갱신 방법을 간단히 요약하면 딜러가 없는 Pederson의 비밀 분산 프로토콜에서 각 참여자가 만드는 다항식의 상수항을 0으로 하는 것이다. 만약 공유 갱신 이전의 비밀 분산에 사용된 다항식을 $f_{old}(x)$라 한다면, 공유 갱신의 결과로 생긴 새로운 공유는 결과적으로 다음의 다항식 $f_{\neq w}(x)$에서 공유를 분배한 결과와 같다.

$$f_{\neq w}(x) = f_{old}(x) + \sum_{i \in QUAL} f_i(x) \qquad (1)$$

이때 새로 더한 $f_i(x)$의 계수의 합은 프로토콜에 참여한 모든 참여자가 각각 새로 만든 다항식의 계수를 알아야만 알 수 있는 값이므로 $f_{old}(x)$에 관한 정보를 갖고 있는 공격자도 새로운 다항식 $f_{\neq w}(x)$에 관하여 정보를 알 수 없게 된다. 따라서 공격자는 새로운 공유에 대한 정보를 얻을 수 없다. Jarecki의 공유 갱신 프로토콜은 Pederson의 검증 가능한 비밀 분산 프로토콜에서 전체 참여자가 모두 딜러의 역할을 한 번씩 수행한 것과 연산량이 갖게 된다. 따라서 위에서 설명한 바와 같이 각 참여자는 공유 갱신을 위해 전체 $O(kn)$번의 모듈라 멱승 연산을 수행해야 한다. 일반적으로 k는 $O(n)$의 값이므로 결국 $O(n^2)$의 모듈라 멱승 연산을 수행해야 한다. 이것은 매우 많은 연산량이며 참여자가 많아지게 될 경우 공유 갱신 프로토콜을 수행하기 위하여 많은 수행시간이 소요된다

3-4 개 요

제안 방식의 전체적인 개요에 대하여 살펴본다. 그림 3은 본 제안 방식에서 나타날 수 있는 브로드캐스트 암호화 흐름이다. 단계 별로 살펴보면 기본 흐름, 갱신 흐름, 신규 과정 흐름, 탈퇴 흐름, 사용자 예측 오류 흐름으로 볼 수 있다. 그림 3의 오른편은 각각의 흐름에 대해 자세히 기술한 것으로, 기본 흐름을 살펴보면, 사용자로부터 정보를 제공받아 키를 생성하는 단계, 브로드캐스트 메시지 생성하는 단계, 마지막으로 콘텐츠를 사용하는 단계로 이뤄진다. 제안 방식은 각각 동일하게 적용되며, 초기 키 생성 및 분배 부분에서 사용자 정보를 제공받고, 키를 생성/분배하여 브로드캐스트 암호화 메시지를 생성/전송하게 된다.

또한 본 방식은 다음과 같은 특징을 가질 수 있도록 제안한다. 사용자의 개인키는 사용자의 동의 과정을 거쳐 생성하며, 참여하고자 하는 사용자의 정보가 수집되었을 때 공개키를 생성하고, 이를 통해 브로드캐스팅되는 메시지를 암호화하여 전송하게 된다. 사용자 이외의 사람은 브로드캐스팅되는 메시지에 대해 복호할 수 없다. 사용자 정보를 제공받아 키를 생성할 때, 사용자 정보에 갱신 정보를 포함하고 있게 됨으로써 후에 사용자가 탈퇴할지라도 사용자의 갱신 정보만을 제거함으로써 쉽게 사용자 탈퇴와 같은 키 갱신이 용이하다. 위와 같은 장점을 임베디드 컴

퓨팅 환경에서의 요구사항을 만족할 수 있도록 제안한다.

3-5 시스템 계수

다음은 본 방식에서 사용되는 시스템 계수를 기술한 것이다.

- p : 소수 $\geq 512bit$
- q : 소수 $\geq 160bit\,(q|p-1)$
- n : $n=pq$
- $d_1,\ldots,d_k$: 개별 복호화키 리스트
- $d_i=\theta_i \bullet \gamma_i \pmod n$ $(\gamma_i\in\Gamma)$
  - $\Gamma=\gamma_1,\ldots,\gamma_k \quad (\Gamma\in Z_q)$
  - θ_i : 각 개인의 키 인자
  - l : 참여 사용자
- M : 메시지
- S : 세션키
- k : 예측 사용자의 수
- i : 사용자 $(i=1,\ldots,k)$
- j : 탈퇴자
- b : 서버가 공개한 공개값
- r_i : 랜덤 수 집합$(r_i\in Z_p)\rightarrow(r_1,\ldots,r_k)$
- $h_i=g^{\gamma_i} \pmod n$

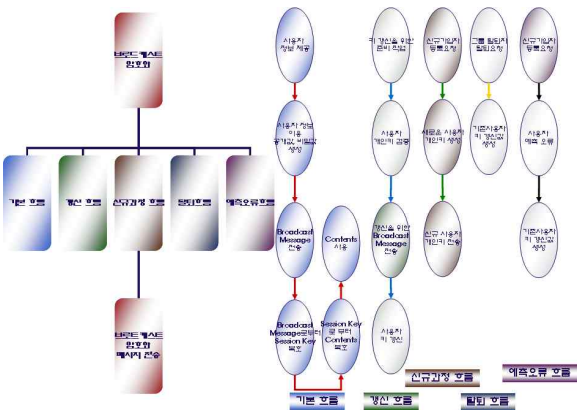

그림 3. 브로드캐스트 암호화 흐름
Fig. 3. Flow of broadcast encryption

- $\langle y,h_1,\ldots,h_k\rangle$: 공개키
- $y=\prod_{i=1}^{k} h_i^{\alpha_i} \bmod n$
- α_i : 랜덤수 $(\alpha_i\in Z_q)\{\alpha_1,\ldots,\alpha_k\}$
- $(h_1 \bullet \ldots \bullet h_k)^{(r_1,\ldots,r_k)a(d_i/\gamma_i)}=y^{aT} \pmod n$
- a: 랜덤 인자$(a\in Z_q)$
- C: 방송 메시지(Broadcast message)
- $C=\langle M(\text{or } S)y^{aT},h_1^a,\ldots,h_k^a\rangle=\langle B,H_1,\ldots,H_k\rangle$
  $=\langle B,H_1^{r_1},\ldots,H_k^{r_k}\rangle$
- $B=M(\text{or } S)y^{aT} \pmod n$
- $H_i=h_i^a \bmod n$
- t_i : 키 갱신을 위한 인자 $(t_1,\ldots,t_k\in Z_q)$
- $T=t_1 \bullet \ldots \bullet t_k \pmod n$
- Θ_i, U_i: 사용자가 등록에 참여하기 위한 사용자 정보
- ζ_i: 사용자가 임의로 선택한 값
- Ξ_i: 사용자의 ID를 보관한 값
- CT: 세션키로 암호화된 메시지 $CT=E_S(M)$

3-6 제안 프로토콜

본 제안 방식은 사용자의 수가 정해져 있고, 사용자가 서버에게 정보를 제공하면 이를 바탕으로 사용자의 키를 생성하고 분배하는 방식이다. 본 방식은 사용자의 정보가 제공된 후 서버가 이를 바탕으로 키 생성과 분배뿐만 아니라 사용자에 대한 인증 문제와 부정 사용자 적발에 대한 문제점을 해결할 수 있는 방식이다. 또한 사용자의 정보를 포함함으로써 사용자에 대한 인증과 함께 서버에 대한 인증도 한 번에 처리할 수 있도록 제안한 것이 특징이다.

3-6-1 키 생성 및 분배 단계

키 생성은 서버(중앙 디바이스)의 담당이며, 개인키와 공개키를 생성하고 전달하기 위해 다음의 일련의 과정을 거친다. 사전에 서버는 각 사용자에게 번호를 부여하고 이를 바탕으로 다음의 단계를 따르도록 한다.

Step 1. 서버(중앙 디바이스)는 사용자의 정보 획득을 위하여 집합 Γ를 생성하여 공개한다.

$$\gamma_i\in\Gamma,\ (i=1,2,\ldots,k) \qquad (2)$$

Step 2. 사용자는 공개된 집합 정보 Γ를 이용하여 자신의 ID_i 값이 포함된 Ξ_i값을 계산한다. 이때 선택된 γ_i값은 사용자에 할당된 번호와 동일한 값으로 서버가 전송하게 된다.

$$\Xi_i \equiv (ID_i)^{1/\gamma_i} (\mathrm{mod}\ n) \quad (3)$$

Step 3. 사용자는 생성된 Ξ_i를 이용하여 다음을 계산한다.

$$U_i \equiv \Xi_i \bullet \zeta (\mathrm{mod}\ n) \quad (4)$$

$$\Theta_i \equiv \zeta^b (\mathrm{mod}\ n) \quad (5)$$

Step 4. 사용자는 생성된 (Θ_i, U_i)를 서버에게 전송한다.

Step 5. 서버는 사용자로부터 제공받은 (Θ_i, U_i)를 이용하여 사용자 정보 ID_i'를 획득한다. 즉, Θ_i로부터 ζ'을 추출하면 추출한 값을 이용하여 Ξ_i'를 획득한다. 획득한 Ξ_i'값을 이용하여 ID_i'값을 계산한다. 획득한 ID_i'는 사용자에게 제공된 γ_i와 함께 보관한다.

$$(\Theta_i)^{1/b} \equiv (\zeta^b)^{1/b} = \zeta' \ (\mathrm{mod}\ n) \quad (6)$$

$$(U_i / \zeta') \equiv (\Xi_i \bullet \zeta) / \zeta' = \Xi_i' \ (\mathrm{mod}\ n) \quad (7)$$

$$(\Xi_i')^{\gamma_i} = \left(ID_i^{1/\gamma_i}\right)^{\gamma_i} = ID_i' \ (\mathrm{mod}\ n) \quad (8)$$

Step 6. 서버는 사용자 정보 ID_i'로부터 사용된 γ_i를 확인하고 다음을 계산한다.

$$h_i \equiv g^{\gamma_i} \mathrm{mod}\ q \quad (9)$$

계산한 값을 보관하고, 모든 사용자의 $h_i\ (i = 1, \cdots, n)$로부터 공개키를 생성하여 공개한다.

$$\langle y, h_1, ..., h_k \rangle \quad (10)$$

Step 7. 서버는 공개키 생성에서 사용된 정보를 포함하여 개인키 인자 θ_i를 계산한다. 이때 갱신을 위해 t_i 인자를 각 개인키에 삽입하여 생성하지만, 사용자 l에 해당하는 갱신정보 t_l는 θ_i에 포함하지 않고 생성한다. 생성된 각 $t_i\ (i = 1, \ ...\ , \ k)$값은 전체의 곱인 T와 같다.

$$T = t_1 \bullet \ ... \ \bullet t_k (\mathrm{mod}\ n) \quad (11)$$

$$\theta_i = \left(\prod_{i=1}^{k} \gamma_i \alpha_i t_i\right) / \left(\prod_{i=1}^{k} r_i \gamma_i\right) \mathrm{mod}\ q \quad (12)$$

Step 8. 서버는 생성된 개인키 d_i를 계산할 때 사용자 l의 갱신인자 t_l를 포함하여 사용자에게 전송한다.

$$d_i = \theta_i \cdot \gamma_i \cdot t_l \ (\mathrm{mod}\ n) \quad (13)$$

Step 9. 사용자는 전송받은 d_i에서 θ_i'를 획득한다.

$$d_i / \gamma_i = (\theta_i \cdot \gamma_i \cdot t_l) / \gamma_i (\mathrm{mod}\ n) \quad (14)$$

$$\theta_i' = \left(\prod_{i=1}^{k} (\alpha_i \cdot t_i) / (r_i)\right) \cdot t_l \quad (15)$$

3-6-2 브로드캐스트 메시지 생성 단계

브로트캐스트 메시지를 전송하는데 있어 메시지 M을 암호화한 세션키 S를 암호화하여 전송할 수 있고, 또 다른 방법은 메시지 M 자체를 암호화하여 전송할 수 있다. 다음에서는 두 가지에 대해 모두 고려하여 기술한다.

첫 번째는 메시지 M만을 암호화하여 전송하는 방식으로 메시지가 짧을 경우와 1회 전송으로 끝나는 경우에는 효율적으로 전송할 수 있지만 메시지의 길이가 길거나 많은 수를 전송하게 되는 경우에는 적합하지 않다.

Step 1. 서버는 다음과 같은 브로드캐스트 메시지를 작성하여 전송한다. 이때 브로드캐스트 메시지에 메시지 M을 암호화하여 전송하는 경우는 다음과 같다.

$$C=\langle My^{aT},h_1^a,...,h_k^a\rangle=\langle B,H_1,...,H_k\rangle \quad (16)$$
$$=\langle B,H_1^{r_1},\ ...\ ,H_k^{r_k}\rangle$$

Step 2. 사용자는 전송받은 메시지에 개인키를 이용하여 메시지 M을 획득한다.

$$M=B/U^{\theta_i'},U=\prod_{i=1}^{k}H_i^{r_i}\ (\mathrm{mod}\ n) \quad (17)$$

$$U^{\theta_i'}=\left(\prod_{i=1}^{k}H_i^{r_i}\right)^{\theta_i'}=\left(\prod_{i=1}^{k}g^{ar_i\gamma_i}\right)^{\theta_i'}=\left(\prod_{i=1}^{k}g^{r_i\gamma_i}\right)^{\theta_i'a} \quad (18)$$
$$=\left(\prod_{i=1}^{k}g^{\gamma_i\alpha_i t_i}\right)^{a}=\left(\prod_{i=1}^{k}h_i^{\alpha_i t_i}\right)^{a}=y^{aT}\ (\mathrm{mod}\ n)$$

$$M=(M\bullet y^{aT})/y^{aT}\ (\mathrm{mod}\ n) \quad (19)$$

다음은 세션키 S로 메시지 M을 암호화하여 전송하는 경우로써 이는 메시지의 길이가 길거나 여러 회 전송해야하는 경우에 가장 효율적인 전송을 제공할 수 있다. 따라서 위의 방식이나 다음 기술하는 방식은 혼용하여 사용 가능하다.

Step 1. 서버는 다음과 같은 브로드캐스트 메시지를 작성하여 전송한다. 이때 브로드캐스트 메시지에 세션키 S를 암호화하여 전송하는 경우 우선 세션키 S로 메시지를 암호화하고, 메시지 암호화에 사용된 세션키 S를 브로트캐스트 메시지에 암호화하여 전송한다.

$$CT=E_S(M) \quad (20)$$

$$C=\langle Sy^{aT},h_1^a,...,h_k^a\rangle=\langle B,H_1,...,H_k\rangle \quad (21)$$
$$=\langle B,H_1^{r_1},\ ...\ ,H_k^{r_k}\rangle$$

Step 2. 사용자는 전송받은 메시지에 개인키를 이용하여 세션키 S를 다음과 같이 획득한 후 세션키 S로 복호하여 메시지 M을 획득하게 된다.

$$S=B/U^{\theta_i'},U=\prod_{i=1}^{k}H_j^{r_i}\ (\mathrm{mod}\ n) \quad (22)$$

$$U^{\theta_i'}=\left(\prod_{i=1}^{k}H_i^{r_i}\right)^{\theta_i'}=\left(\prod_{i=1}^{k}g^{ar_i\gamma_i}\right)^{\theta_i'}=\left(\prod_{i=1}^{k}g^{r_i\gamma_i}\right)^{\theta_i'a} \quad (23)$$
$$=\left(\prod_{i=1}^{k}g^{\gamma_i\alpha_i t_i}\right)^{a}=\left(\prod_{i=1}^{k}h_i^{\alpha_i t_i}\right)^{a}=y^{aT}\ (\mathrm{mod}\ n)$$

$$S'=(S\bullet y^{aT})/y^{aT}\ (\mathrm{mod}\ n) \quad (24)$$
$$M'=D_{S'}(E_S(M)) \quad (25)$$

3-6-3 키 갱신 단계

임베디드 컴퓨팅 환경에 참여하는 사용자에게 제공된 개인키가 분배되고 전송된 후, 서버는 암호화된 브로드캐스트 메시지를 생성하여 전송한다. 그러나 사용자의 탈퇴 혹은 신규 가입자가 발생한 경우는 다음과 같은 과정을 통하여 사용자의 키를 갱신하고 새롭게 갱신된 키를 이용하여 브로드캐스트 메시지를 전송하게 된다. 다음의 과정에서 마지막 단계에서 제공되는 브로드캐스트 메시지의 두 가지 변화에 대해서 기술한다. 이것은 서버 측에서 메시지 M을 암호화하여 전송하거나 메시지 M을 암호화한 세션키 S를 전송하는 방법에 대해 단계별로 나눠 기술한다.

다음은 서버가 갱신된 키를 이용하여 메시지 M을 전송하는 경우로써 브로드캐스트 메시지 전송 단계에서 살펴보았듯이 본 전송의 장점은 1회 전송 혹은 짧은 메시지 전송에 있다.

Step 1. 사용자 j가 탈퇴 요청하는 메시지를 전송한다.

Step 2. 서버는 기존 사용자의 개인키를 갱신하기 위해 갱신 인자인 T에서 사용자 j의 갱신 인자인 t_j를 t_j^{-1}로써 제거하고, 서버가 보관하는 갱신 인자도 변경하여 T'로 보관한다.

$$T\cdot t_j^{-1}=(t_1\cdot t_2\cdot\cdots\cdot t_k)\cdot t_j^{-1}=T'\ (\mathrm{mod}\ n) \quad (26)$$

Step 3. 서버는 각 사용자의 개인키에 탈퇴 사용자 정보 t_j^{-1}를 갱신하여 갱신된 개인키 d_i'를 전송한다.

$$\theta_i\cdot\gamma_i\cdot t_l\cdot t_j^{-1}=d_i'\ (\mathrm{mod}\ n) \quad (27)$$

Step 4. 사용자는 서버로부터 전송받은 갱신된 키 정보를 이용하여 자신의 개인키를 획득한다. 이때 개인키에는 사용자의 갱신 정보가 포함된 값인

$\theta_i \cdot t_l \cdot t_j^{-1}$이 된다. 서버는 갱신된 키를 이용하여 브로드캐스트 메시지를 전송하고, 사용자는 암호화된 브로드캐스트 메시지 C를 복호하여 메시지 M'을 획득하게 된다.

$$d_i'/\gamma_i = (\theta_i \cdot \gamma_i \cdot t_l \cdot t_j^{-1})/\gamma_i = \theta_i \cdot t_l \cdot t_j^{-1} \pmod n \quad (28)$$

$$\theta_i'' = \left(\prod_{i=1}^{k}(\alpha_i \cdot t_i)/(r_i)\right) \cdot t_l \cdot t_j^{-1} \pmod n \quad (29)$$

$$C = \langle B', H_1, \dots, H_k \rangle = \langle M \bullet y^{aT'}, h_1^a, \dots, h_k^a \rangle \quad (30)$$
$$= \langle B, H_1^{r_1}, \dots, H_k^{r_k} \rangle$$

$$M = B/U^{\theta_i''}, U = \prod_{i=1}^{k} H_i^{r_i} \pmod n \quad (31)$$

$$U^{\theta_i''} = \left(\prod_{i=1}^{k} H_i^{r_i}\right)^{\theta_i''} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i}\right)^{\theta_i''} = \left(\prod_{i=1}^{k} g^{r_i\gamma_i}\right)^{\theta_i''a} \quad (32)$$
$$= \left(\prod_{i=1}^{k} g^{\gamma_i\alpha_i t_i}\right)^{at_j^{-1}} = \left(\prod_{i=1}^{k} h_i^{\alpha_i t_i}\right)^{at_j^{-1}} = y^{aT'} \pmod n$$

$$M' = (M \bullet y^{aT'})/y^{aT'} \pmod n \quad (33)$$

또 다른 방식으로 서버는 갱신된 키를 제공하고 갱신된 키로 메시지 M을 암호화하여 전송하는 것이 아닌 메시지 M은 세션키 S로 암호화하고 이 세션키 S를 브로드캐스트 암호화하여 전송하는 방법이다. 이 방법도 앞 단계에서 기술한 장단점을 가진다. 이때 사전에 이뤄지는 Step 1. ~ Step 3.까지는 동일한 단계로 이뤄지며 마지막 단계만 다음과 같이 다르게 계산하여 전송하게 된다.

Step 5. 사용자는 서버로부터 전송받은 갱신된 키 정보를 이용하여 자신의 개인키를 획득한다. 서버는 갱신된 키를 이용하여 브로드캐스트 메시지를 전송하고, 사용자는 다음과 같이 암호화된 세션키 S를 복호하여 세션키를 획득한 후에 메시지를 복호하게 된다.

$$d_i'/\gamma_i = (\theta_i \cdot \gamma_i \cdot t_l \cdot t_j^{-1})/\gamma_i = \theta_i \cdot t_l \cdot t_j^{-1} \pmod n \quad (34)$$

$$\theta_i'' = \left(\prod_{i=1}^{k}(\alpha_i \cdot t_i)/(r_i)\right) \cdot t_l \cdot t_j^{-1} \pmod n \quad (35)$$

$$C = \langle B', H_1, \dots, H_k \rangle = \langle S \bullet y^{aT'}, h_1^a, \dots, h_k^a \rangle \quad (36)$$
$$= \langle B, H_1^{r_1}, \dots, H_k^{r_k} \rangle$$

$$CT = E_S(M) \quad (37)$$

$$S = B/U^{\theta_i t_j^{-1}}, U = \prod_{i=1}^{k} H_i^{\gamma_i} \pmod n \quad (38)$$

$$U^{\theta_i''} = \left(\prod_{i=1}^{k} H_i^{r_i}\right)^{\theta_i''} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i}\right)^{\theta_i''} = \left(\prod_{i=1}^{k} g^{r_i\gamma_i}\right)^{\theta_i''a} \quad (39)$$
$$= \left(\prod_{i=1}^{k} g^{\gamma_i\alpha_i t_i}\right)^{at_j^{-1}} = \left(\prod_{i=1}^{k} h_i^{\alpha_i t_i}\right)^{at_j^{-1}} = y^{aT'} \pmod n$$

$$S' = \left(S \bullet y^{at_j^{-1}}\right)/y^{at_j^{-1}} \bmod n \quad (40)$$

$$M' = D_{S'}(E_S(M)) \quad (41)$$

표 1. 기존 방식과 제안 방식 비교
Table 1. Comparison with previous methods

분석 / 방식	사용자 참여	키 갱신	N회 탈퇴	키의 연속성	Traitor Tracing	초기 예측 오류에 따른 재연산	암호 블록길이
기존 KPS[3]	X	X	X	O	X	필요	
Broadcast Encryption[1]	X	O	X	△	X	필요	O(Z+1)
IKPS[4]	O	O	X	△	X	불필요	
Narayanan[2]	X	O	△	O	O	필요	O(2z)*|p|
제안방식	O	O	O	O	X	불필요	O((n+1)/2)

(O: 제공, △: 일부 제공, X: 제공 못함)

Ⅳ. 제안 방식 고찰

본 논문에서는 임베디드 컴퓨팅 환경에서 기존의 방식보다 효율적인 키 생성과 키 갱신을 위한 브로드캐스트 암호화 방식을 제안하였다. 본 제안 방식의 안전성은 이산대수의 문제에 기반을 두고 있다. 기존의 방식에 비해 사용자의 참여, 키 갱신, 사용자의 탈퇴 혹은 연산량에 있어 효율성을 나타내고 있다. 본 장에서는 제안 방식과 기존 방식을 비교하여 제안 방식의 효율성을 보이도록 한다.

4-1 사용자 참여

기존의 방식에서 보면 서버는 사용자의 참여없이 사용자를 예측하여 키를 미리 생성하고 새로 가입하는 사용자에게 키를 제공하여 분배하는 형태를 가지고 있다. 4.1절에서 기술한 기존의 사용자를 예측하여 전체키를 사전에 생성하는 모델에서는 서버 자체에 대하여 공격이 이뤄 질 경우 서버가 생성한 키 전체에 대하여 악의적인 행위가 발생할 수 있다. 그러나 본 논문에서는 문제점을 해결하고자 키를 생성하

고 분배하기 이전에 사용자의 그룹을 먼저 형성하고, 사용자의 정보를 제공받아 이용함으로써 서버에 대한 공격을 대비하였다. 서버에 사용자의 정보가 전송될 때 사용자와 서버가 정보가 다르게 함으로써 사용자들의 정보를 수집하여 키를 유도하지 못하도록 제안하였다.

4-2 키 갱신

기존 KPS(Key Predistribution Scheme)에서는 사전에 키가 생성/분배된 후에 브로드캐스트 메시지를 전송한다. 전송된 메시지를 사용자가 확인한 후 한 세션이 종료되면 키를 새롭게 생성하고 전송된다. 또한 키에 대하여 공격이 발생한 경우 키를 갱신하지 않고, 전체적으로 재생성하게 된다. 하지만 본 제안 방식에서는 사용자의 가입 혹은 탈퇴가 발생하면 기존 사용자들의 키를 갱신하고 사용이 가능한데 이는 초기 키 생성 시에 키 갱신 인자인 T인자를 삽입하게 되며, 차후 사용자 탈퇴/강제 탈퇴 등과 같은 상황이 발생되면 서버는 탈퇴자의 키 갱신 인자인 t_i^{-1}를 제공함으로써 사용자는 간단한 연산으로 키 갱신을 이뤄지게 됨으로써, 본 제안 방식은 기존 방식보다 빠르고 효율적으로 키 갱신이 되도록 제안하였다.

4-3 초기 예측에 따른 재연산

기존 방식과 서버가 사용자를 예측하여 키를 생성하는 경우 키 관리와 시스템 전체를 관리하여야 하며 사용자의 불법 사용 시 키에 대한 사용자 정보를 가지고 있지 않기 때문에 많은 문제를 발생할 수 있다. 또한, 만일 서버가 유동적인 사용자를 관리한다면 사용자에 대한 예측이 올바르게 이뤄져야 함에도 서버는 초기 예측에 대한 오류가 발생하였을 경우 재연산 혹은 추가 연산을 실시해야 한다. 이러한 문제점을 해결하고자 본 논문에서는 서버가 시스템을 설정하는데 사용자에 대한 예측 연산을 지원하지 않고 사용자의 정보를 제공받아 이를 통해 인가된 사용자에게만 키 분배가 가능하다. 또한 랜덤한 수 r_i에 대해서는 Z_p상에서 생성하게 되며 γ_i에 대해서 사전에 예측 사용자보다 많은 수를 만들면 해결할 수 있다. 제안 방식은 초기 사용자 예측이 아닌 사용자 정보로부터 인가된 그룹을 형성하고 키를 생성하고 분배함으로 4.1절에서 언급했던 서버 예측에 따른 위협 요소를 감소시킬 수 있다. 하지만 동절에서 기술한 사용자 정보로부터 제공받을 경우 사용자 정보의 지연에 따른 전제척인 통신지연을 발생시키는 원인이 되기도 한다.

Ⅴ. 결 론

브로드캐스트 암호화 기법은 공개된 네트워크상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다. 브로드캐스트 암호화 기법에서는 중요한 것은 오직 사전에 허가된 사용자만이 디지털 정보를 얻을 수 있어야 하며, 브로드캐스트 메시지가 전송되면 권한이 있는 사용자들은 자신이 사전에 부여받은 개인키를 이용하여 디지털 정보를 얻게 된다. 이와 같이 사용자는 브로드캐스터가 전송하는 키를 이용하여 메시지나 세션키를 획득하게 되는데, 이러한 과정 전에 브로드캐스터가 키를 생성하고 분배하는 과정이 필요하게 된다. 또한 사용자가 탈퇴나 새로운 가입 시에 전체 공개키나 각 개인의 개인키에 대한 효율적인 키 갱신이 필요하다.

본 논문에서 크게 두 가지 모델에 대해 살펴보았다. 우선 첫 번째 모델은 사용자의 정보를 제공하여 사용자 정보로부터 공개키와 개인키를 생성하는 모델이고, 두 번째 모델은 사용자의 정보없이 서버가 접속할 사용자를 예측하여 키를 생성하는 모델에 대해 살펴보았다. 컨퍼런스와 같은 소규모 네트워크를 대상으로 하는 곳에서는 올바른 사용자에 대한 올바른 키가 분배되어야한다. 따라서 본 논문에서는 서버가 단독으로 키를 생성하여 분배하는 방식이 아닌, 서버가 사용자로부터 정보를 획득하여 이 정보로부터 각 사용자의 개인키를 생성하여 분배하도록 한다. 따라서 본 방식은 사용자에게 개인키의 생성, 분배와 갱신에 이르는 효율적인 방법을 제안하였다. 본 제안 방식의 특징은 키 갱신에 초점을 두어 키 갱신 인자인 T 를 통하여 효율적인 키 갱신이 이뤄지는 것이

며, 또한 기존 사용자의 동의 없이 키 정보를 생성하게 되면 서버 노출 시 사용자의 키가 모두 노출된다는 취약점에 대해 사용자가 자신의 정보를 생성하여 서버에 제공하는 방식으로 제안하였다. 본 제안 방식은 Ad-Hoc 상에서 컨퍼런스에 적합한 브로드캐스트 암호화 방법을 이용한 키 생성과 키 갱신을 제시하였다. 본 방식은 음악, 방송과 같은 콘텐츠 분배에 있어 효율적인 제공할 수 있으며, 소규모에서 암호통신을 전체적으로 실시할 경우에도 효율적으로 사용될 수 있을 것이다. 마지막으로 본 연구는 향후 세션에 대해 각각의 키를 다시 생성하고 분배하는 것은 서버나 사용자에 많은 부담을 전가시킬 수 있는데 키 주기에 관한 연구를 통해 효율적인 키 관리를 이룰 수 있을 것이라 본다.

참 고 문 헌

[1] A. Fiat and M. Naor, "Broadcast Encryption", *Crypto'93*, pp. 480-491, 1993

[2] A. Narayana, "Practical Pay TV Schemes", *to appear in the Proceedings of ACISP03*, July, 2003

[3] C. Blundo, L. A. F. Mattos and D.R. Stinson, "Generalized Beimel-Chor schemes for Broadcast Enryption and Interactive Key Distribution", *Theoretical Computer Science,* vol. 200, pp. 313-334, 1998.

[4] D. H. Lee, H. J.. Kim and J. I. Lim, "Efficient Public-Key Traitor Tracing in Provably Secure Broadcast Encryption with Unlimited Revocation Capability", *KoreaCrypto 02'*, 2003

[5] I. Gracia, S. Martin and C. Padro, "Improving the Trade-off Between Storage and Communication in Broadcast Encryption Schemes", *Discrete Applied Mathematics archive,* Vol. 143, Issue 1-3, pp 213 - 220, 2004

[6] 이덕규, 이임영, “브로트캐스트 암호화에서의 효율적인 키 생성과 갱신 방법”, *정보처리학회논문지 C*, 제 11-C권 제2호, pp 149-156, 2004

[7] 임채덕, 김홍남, 박승민, 김두헌, 김선자, 김채규, 임기욱, “임베디드 소프트웨어 기술동향 및 산업 발전 동향”, *정보통신연구진흥지*, 4권 3호, 2002

[8] 장정숙, 전용희, “임베디드 시스템 보안”, *한국정보통신학회지,* 22권 8호, pp 81-97, 2005

이 덕 규 (李悳揆)

2001년 순천향대학교 공학사
2003년 순천향대학교 공학석사
2006년 순천향대학교 공학박사
2006년~현재 한국전자통신연구원 정보보호연구단
관심분야 : 유비쿼터스 및 RFID보안, 임베디드 소프트웨어

김 태 훈 (金泰勳)

1995년 성균관대학교 공학사
1997년 성균관대학교 공학석사
1999년 (주)신도리코 기술연구소 연구원
2002년 성균관대학교 공학박사
2004년 한국정보보호진흥원 선임연구원
2006년 국군기무사령부 사무관
2007년 이화여자대학교 연구교수
2007년~현재 한남대학교 멀티미디어학부 조교수
관심분야 : 유비쿼터스 및 RFID보안, 임베디드 소프트웨어

여 상 수 (呂相壽)

1997년 중앙대학교 공학사
1999년 중앙대학교 공학석사
2005년 중앙대학교 공학박사
2006년 단국대학교 정보컴퓨터학부 강의전임강사
2007년~현재 Kyushu University 방문연구원
관심분야 : 유비쿼터스 및 RFID보안, 임베디드 소프트웨어

박 길 철 (朴吉綴)

1983년 한남대학교 공학사
1986년 숭실대학교 공학석사
1988년 성균관대학교 공학박사
1998년~현재 한남대학교 멀티미디어학부 정교수
관심분야 : Communication, HCI,VR

김 석 수 (金錫洙)

1989년 경남대학교 이학사
1991년 성균관대학교 공학석사
1991년 정풍물산(주)중앙연구소 주임연구원
1997년 한국 탑웨어 책임연구원
1998년 경남 도립 거창전문대학교 교수
2000년 동양대학교 컴퓨터공학부 교수
2002년 성균관대학교 공학박사
2003년~현재 한남대학교 멀티미디어학부 조교수
관심분야 : Ubiquitous, Healthcare, Multimedia Authoring

조 성 언 (趙誠彦)

1989년 한국항공대학교 항공통신정보공학과 공학사
1991년 한국항공대학교 대학원 항공통신정보공학과 공학석사
1997년 한국항공대학교 대학원 항공전자공학과 공학박사
1997년~현 재 순천대학교 정보통신공학부 부교수
관심분야 : 무선통신시스템, Wireless USN