情報保護學會論文誌
第19卷 第3號, 2009. 6

해쉬함수에 기반한 경량화된 RFID 인증 프로토콜

하 재 철,[1†‡] 백 이 루,[1] 김 환 구,[1] 박 제 훈,[2] 문 상 재[2]
[1]호서대학교, [2]경북대학교

Lightweight RFID Authentication Protocols Based on Hash Function

JaeCheol Ha,[1†‡] YiRoo Baek,[1] HwanKoo Kim,[1] JeaHoon Park,[2] SangJae Moon[2]
[1]Hoseo University, [2]Kyungpook National University

요 약

본 논문에서는 RFID 시스템에서 태그와 백엔드 서버간의 안전성과 태그의 구현 효율성을 높인 두 가지 형태의 상호 인증 방식을 제안하고자 한다. 첫 번째 제안 방법에서는 고정된 ID를 사용하여 분산 환경에 이용할 수 있는 방법이며, 두 번째 방법은 변동 ID를 사용한 것으로서 전방향 안전성(forward security)을 추가적으로 만족할 수 있도록 설계하였다. 제안 방식에서 태그가 랜덤 수 발생기를 사용하지 않으면서 최소 한번 혹은 최대 3번 정도의 해쉬 연산만으로 상호 인증을 수행할 수 있다. 제안된 프로토콜들을 RFID 겸용 스마트 카드 칩에서 구현하고 그 동작이 타당함을 검증하였다.

ABSTRACT

To guarantee security between the tag and back-end server and implementation efficiency in low power tag, we propose two typed mutual authentication protocols in RFID system. One is static-ID authentication scheme which is well suitable in distributed server environments. The other is dynamic-ID scheme which is additively satisfied forward security. In proposed scheme, it does not need any random number generator in tag and requires only one(maximally three) hash operation(s) in tag or server to authenticate each other. Furthermore, we implement the proposed schemes in RFID smart card system and verify its normal operations.

Keywords: RFID Mutual Authentication Protocol, Static-ID Scheme, Dynamic-ID Scheme, Random Number Generator

I. 서 론

유비쿼터스 시대를 맞이하여 주목을 받고 있는 RFID 시스템은 RFID 태그와 리더 그리고 데이터베이스를 갖춘 백엔드 서버(back-end server, 혹은 백엔드 DB)로 구성된 개체를 인식하는 시스템으로 태그가 소형, 저가, 내장성 등의 특징을 갖추고 있다[1,2]. 그러나 RFID 시스템은 태그와 리더간의 통신을 위해 RF 무선 신호를 이용하게 되고 이로 인해 여러 가지 보안 문제가 대두되었다. 특히, 태그를 소유한 사람이나 사물에 대한 프라이버시 노출, 위치 추적, 그리고 위조 공격 등이 위협적이다. 이와 같은 문제를 해결하는 방법으로 태그와 서버간의 상호 인증 기법들이 제안되었다[3-11]. 제안된 기존의 인증 방법들 중에서 해쉬 함수를 이용하는 방법이 안전도나 효율성 면에서 적합한 기술로 평가되고 있어 많이 연구되고 있다. 안전성과 더불어 RFID 인증 시스템에서는 구현의 효율성을 고려해야 한다. 특히, 태그가 서버에 비해 소형, 저용량, 저성능인 점을 감안하면 태그가 사용하는 알고리듬의 경량화와 무선 RF 통신량의 감소는 인증 프로토콜의 중요한 설계 요소가 된다.

기존에 제안된 해쉬 함수 기반의 상호 인증 방식은 크게 두 부류로 나눌 수 있는데 그 하나는 고정된 ID(static-ID)를 사용하는 방법으로서 Rhee 등의

접수일(2008년 12월 10일), 수정일(2009년 2월 27일), 게재확정일(2009년 3월 17일)
† 주저자, jcha@hoseo.edu
‡ 교신저자, jcha@hoseo.edu

CRAP(Challenge-Response Based Authentication Protocol)방식[6]과 Choi 등이 제안한 OHLCAP (One-way Hash Based Low-Cost Authentication Protocol)방식[7] 등이 있다. 다른 하나는 변동 ID(dynamic-ID)를 이용하는 방법으로서 Dimitriou의 LCRP(Lightweight Challenge-Response Protocol)[8], Lee 등이 제안한 LCAP(Low-Cost Authentication Protocol)[9], Ha 등의 LRMAP[10] 등이 있다. 고정 ID를 사용하는 방식은 태그의 ID가 변하지 않으므로 서버가 분산되어 있는 환경에는 적합하지만 ID를 변경할 수 없는 특징으로 인해 전방향 안전성 혹은 후방향 비추적성(forward security 혹은 backward untraceability)을 제공하기가 힘들다. 전방향 안전성은 태그가 어느 시점에서 그 비밀 정보가 노출되더라도 그 이전에 사용한 정보를 이용하여 위치 추적이나 프라이버시를 침해할 수 없도록 하는 성질이다. 따라서 전방향 안전성을 제공하기 위한 목적으로 변동 ID 기법을 많이 사용하는데 이 기법은 어느 시점에서 태그의 비밀 정보가 한번 노출된 경우에도 이전 세션에서 사용한 비밀에 대해 안전성을 보장받기를 원하는 인증 시스템에 사용할 수 있다. 최근에는 분산 환경에 적합하면서 태그나 데이터베이스의 검색시간을 줄이는 연구가 많이 진행되고 있다. 특히 문헌 [11]에서는 다음 세션에서 태그가 보낼 메시지를 이전 세션에서 데이터베이스가 미리 저장하는 방식을 이용하여 데이터베이스에서의 태그 검색 시간을 줄이면서 재동기화 기능을 갖는 방법을 제안한 바 있다.

본 논문에서는 [11]의 상호 인증 프로토콜을 개선하여 태그에서 랜덤 수 발생기를 사용하지 않으면서 인증에 필요한 태그의 연산량을 줄이는 방법을 제안하고자 한다. 여기에서는 고정 ID를 사용하는 경우와 변동 ID를 사용하는 경우로 나누어 상호 인증 모델을 제시한다. 또한, 이 RFID 인증 시스템을 상용 RFID 겸용 스마트 카드 칩에 구현하여 상호 인증이 실시간 내에 잘 수행될 수 있음을 검증하였다.

II. RFID 인증 프로토콜 관련 연구

2.1 용어 및 표기

기존 RFID 인증과 관련한 논문을 분석하거나 제안하는 프로토콜의 설명을 위해 사용될 용어 및 표기는 다음과 같이 정리할 수 있다.

- $h(\ )$: 해쉬 함수
- ID : EPC(Electronic Product Code)와 같은 태그의 고유 Identity. 64비트, 96비트, 혹은 256비트. 고정 ID 방식인 경우는 EPC가 곧 ID라고 볼 수 있음. 변동 ID 방식인 경우 서버에 고정된 EPC와 ID를 연결하는 데이터베이스 필드가 필요
- PID : 이전 세션에 사용하였던 ID
- K : 태그의 고유 비밀 키
- r_R : 리더가 발생하는 랜덤 수,
- r_T : 태그가 발생하는 랜덤 수
- $Q_{i/j}$: 메시지 Q를 j개의 블록으로 나눈 것 중에서 i번째 블록
- $||$: 두 정보의 연접(concatenation)
- $|A|$: 데이터 A의 길이를 나타내는 비트 수
- m : 인증을 위해 교환하는 정보의 길이, $m=|Q|=|R|=|r_R|$
- n : ID나 비밀 정보 K의 길이, $n=|ID|=|K|$

2.2 기존의 RFID 인증 방식 분석

해쉬 기반의 상호 인증 기법은 해쉬 함수의 일방향성을 이용하는 것으로 Weis 등이 제안한 해쉬 락(hash lock) 기법이 대표적이다[3]. 그러나 이 방법은 리더와 태그의 통신에서 고정된 $metaID=hash(key)$를 노출하므로 태그의 위치추적이 가능하며 사용자의 프라이버시도 침해되는 문제점을 가지고 있다. 또한 Ohkubo 등은 위치 추적 공격이나 전방향 안전성에 강인한 해쉬 체인 인증 기법을 제안하였다[4]. 그러나 이 방식은 서버가 ID를 검색하는데 많은 시간이 소비되는데, 특히 태그로부터 잘못된 응답이 오면 서버는 무한번의 해쉬 연산을 수행할 수도 있는 단점을 지니고 있다. Henrici와 Muller는 ID를 갱신함으로써 위치 추적 공격을 방어하는 인증 방식을 제안하였으나, 그 후에 비동기화 문제가 발생하면 위치 추적이 가능하며 스푸핑(spoofing) 공격에 취약함이 밝혀졌다[5].

고정 ID를 이용하는 방식 중에서 Rhee 등이 제안한 방식[6]은 안전성이 보장된 분산환경형 상호 인증 프로토콜이지만 데이터베이스에서는 태그의 인증을 위해 모든 ID를 검색하므로 서버에서 많은 연산이 필요하다. 즉, 전체 l개의 태그가 있다면 하나의 태그를 검색하는데 평균 $\lceil l/2 \rceil+1$번의 해쉬 연산이 필요하여 이는 서버에게 큰 검색 부하를 주게 된다. Choi 등이

제안한 해쉬 기반 인증 프로토콜 OHLCAP[7]는 태그가 한번의 해쉬 연산만 수행하므로 효과적인 인증 방법으로 제시되었으나 최근 위치 추적 공격을 비롯한 여러 공격에 취약함이 발견되었다[12,13].

변동 ID를 사용하는 방법 중에는 Dimitriou의 LCRP[8]는 정상적인 세션의 종료가 되지 않으면 다음 세션에서 이전 세션과 동일한 정보를 전송하게 되므로 위치 추적 공격에 안전하지 못하다. 또한 한번이라도 태그와 서버간의 접속 오류로 정보의 비동기화가 생기면 태그를 더 이상 사용할 수 없는 상황도 발생한다. Lee 등이 제안한 LCAP[9]은 공격자에 의해 스푸핑 공격이 가능함이 밝혀졌으며 데이터베이스에서 태그를 찾는데 평균 $\lceil l/2 \rceil + 2$번의 해쉬 연산이 필요하다[10]. Ha 등의 LRMAP[10]은 현재까지 안전한 것으로 밝혀져 있으며 태그에서의 계산량도 3번의 해쉬 연산과 1번의 랜덤 수 발생을 통해 인증을 수행할 수 있어 비교적 계산량도 적다.

최근에는 검색정보 사전 동기화를 이용한 분산환경에 적합한 RFID 인증 방식이 제안되었다[11]. 본 논문에서 제안하고자 하는 인증 프로토콜은 이 방식을 기초로 하므로 [그림 1]에 상세히 도시하였다. 여기서 태그는 사전에 ID와 자신의 비밀 키 K를 발급받아 저장해 두는데 ID는 고정 값이지만 K는 인증 세션마다 변하는 값이다.

- 1단계 : 리더는 질의(Query)와 랜덤 수 r_R을 태그에 보낸다.
- 2단계 : 태그는 랜덤 수 r_T를 발생하고 P와 Q를 계산하여 리더에게 보내고 K를 갱신한다.
- 3단계 : 리더는 데이터베이스에 관련정보를 안전하게 보내고 데이터베이스는 자신이 가진 K와 P를 비교하여 ID를 찾고 인증을 수행한다. 그리고 K를 갱신한 후 R'을 계산하여 보낸다. 이때 이전 세션에서 접속이 끊겨 현재 세션에서 비동기화(desynchronization) 현상이 발생했더라도 리더가 보낸 정보 P, Q, r_R, r_T를 이용하여 ID를 찾고 인증을 수행한다.
- 4단계 : 리더는 R'을 태그에게 보내고 태그는 이를 검증함으로써 데이터베이스를 인증한다.

이 프로토콜은 데이터베이스의 ID 검색을 최소화할 수 있는 방법으로 다음 세션에서 태그가 보낼 메시지를 이전 세션에서 데이터베이스가 미리 저장하는 방법으로 검색 정보를 사전에 동기화한 것이 특징이다. 만약, 이전 세션에서 인증이 정확하게 이루어졌다면, 다음 세션에서 태그는 이전 세션에서 저장되었던 태그 검색을 위한 정보 K를 보내고 데이터베이스는 K값을 비교하여 쉽게 ID를 찾도록 고안한 것이다. 또한 이 프로토콜에서는 세션 중간에 통신이 끊기는 비동기 현상이 발생할 경우일지라도 다음 세션에서 서버는 리더가 보낸 정보 P, Q, r_R, r_T를 이용하여 ID를 찾고 태그가 전송한 P를 이용하여 다음 세션을 위한 K를 복구하도록 하였다. 이 방식에서 태그는 1번의 랜덤 수 발생과 3번의 해쉬 함수 연산을 통해 상호 인증을 수행한다.

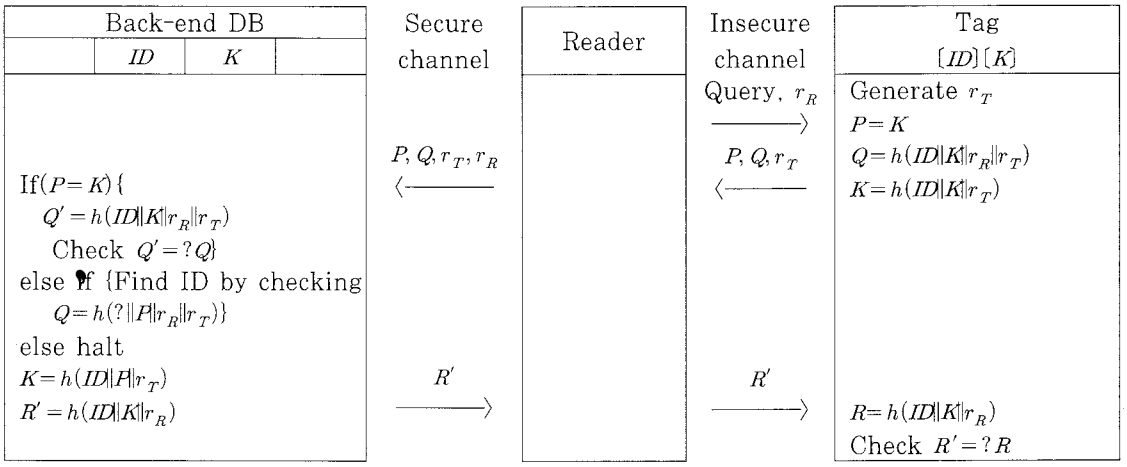
III. 경량화된 RFID 인증 방식 제안

3.1 고정 ID를 사용하는 인증 방식 분석

본 논문에서 제안하고자 하는 인증은 [그림 1]을 비롯한 대부분의 인증 프로토콜에서 사용했던 태그의 랜덤 수 발생기를 사용하지 않도록 설계하고자 한다. 랜덤 수 발생기는 일정한 시드(seed)를 가지고 특정 길이의 랜덤 수를 발생하는 것으로 잡음과 같은 정보를 수집하여 이를 이진화시키는 하드웨어적인 회로를 이용하거나 블록 암호, 해쉬 함수 혹은 전용 랜덤 수 발생 함수를 이용한다. 따라서 이 랜덤 수 발생기는 하드웨어적인 구현 요소가 필요하거나 특정 시드를 저장해서 암호학적으로 안전한 별도의 함수를 이용해야 하므로 이를 구현해야 하는 태그로서는 부하가 커질 수 있다.

[그림 1]에서 보면 태그가 생성한 랜덤 수 r_T는 전송정보 Q값을 매번 변동시키거나 동기화를 위한 비밀정보 K를 갱신하는데 사용된다. 이때 K는 매 세션마다 갱신됨을 알 수 있다. 따라서 본 논문에서의 아이디어는 태그가 세션을 수행할 경우마다 발생하는 랜덤 수 r_T의 역할을 세션마다 갱신되는 K와 동시에 사용하자는 것이다. 즉, 태그와 서버에서 발생하는 K를 매 세션마다 다르게 발생하게 함으로써 태그의 랜덤 수 r_T의 역할을 할 수 있도록 프로토콜을 설계하였다. 이를 위해 서버와 태그는 다음 세션에 사용할 비밀 정보 K를 동시에 저장하되 매 세션마다 갱신될 수 있도록 설계하면 된다.

따라서 태그에서 랜덤 수 r_T 의 발생이 없는 방법을 제안한 것이 [그림 2]의 프로토콜 I이다. [그림 2]에서 보면 태그는 서버와의 동기화나 세션의 성공여부와 관계없이 매 세션이 시작되면 K를 발생하도록 하였다. 따라서 $K = H(ID \| Q)$와 같이 생성하여 태그의 랜덤 수 역할을 하면서 새로운 비밀 정보로 재설정되는 것이

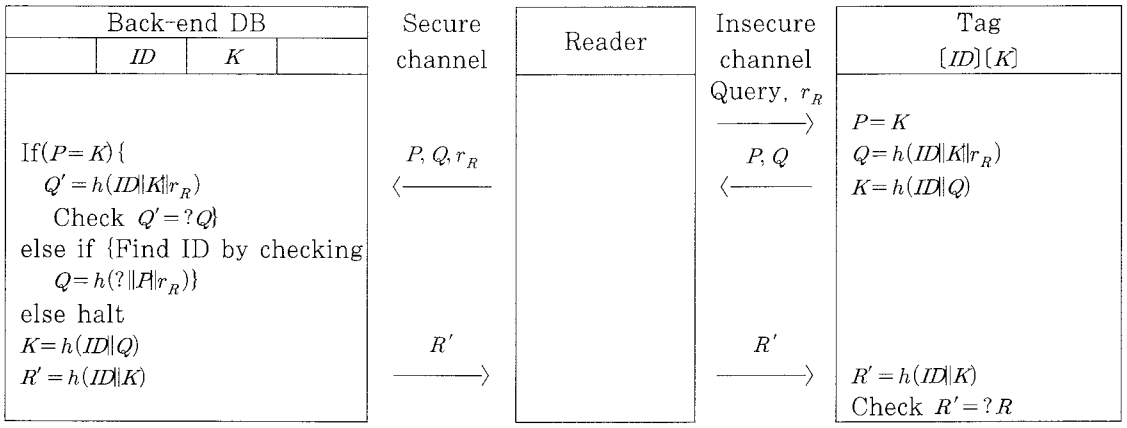


[그림 1] 상호 인증 프로토콜[11]

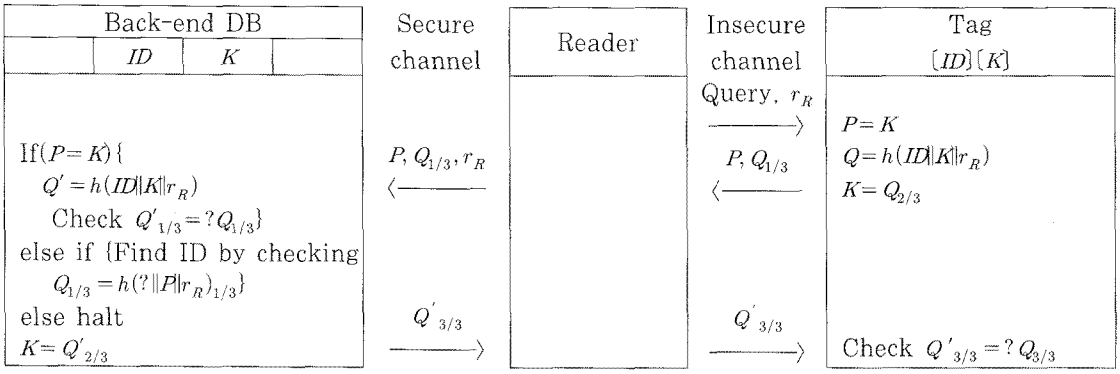
다. 그리고 태그는 서버로부터 $R'=H(ID||K)$를 전달받아 이를 검증함으로써 서버를 인증한다.

[그림 2]에서 제안하는 프로토콜 I에서 보면 태그에서는 최대 3번의 해쉬 연산이 필요한데 여기에서 일반적인 각 정보의 길이를 살펴 볼 필요가 있다. ID는 EPC표준에서는 96비트 혹은 256비트 등을 사용할 수 있다[14]. 문헌 [9]에서는 Q나 R은 해쉬 함수 출력의 절반으로도 충분히 안전도를 보장한다고 하였는데 Q나 R은 m비트 정도면 충분한 안전도를 보장할 수 있다고 가정하자. 예를 들어 $m=80$로 둘 수 있다. 또한 비밀 정보 K는 ID와 같은 길이인 $n=96$비트라 가정하자. 따라서 해쉬 함수의 출력이 크다면 태그에서 3번의 해쉬 연산을 하지 않더라도 충분히 인증 프로토콜을 수행할 수 있을 것이다.

본 논문에서는 이러한 점을 고려하여 태그의 연산량을 더 줄이기 위해 해쉬를 1회만 수행하는 프로토콜을 제안하고자 한다. 제안 방식에서는 해쉬 함수의 출력 길이에 따라 태그나 서버에서 3번의 해쉬 연산을 모두 수행할 필요가 없다. 이를 위해서 $Q=h(ID||K||r_R)$값을 먼저 구하고 이를 LSB부터 3개의 블록으로 나누어서 이용하고자 한다. 이때 해쉬 함수의 출력은 최소 $2m+n$비트 이상이어야 한다. 먼저 Q의 LSB부터 m비트를 $Q_{1/3}$, 다음 n비트를 $Q_{2/3}$, 그리고 다음 m비트를 $Q_{3/3}$이라고 하자. 만약 $m=80$이고 $n=96$이라면 해쉬 함수의 출력은 최소 256비트가 필요하므로 SHA-256[15]와 같은 알고리듬을 사용하면 [그림 3]과 같이 한번의 해쉬 연산으로 상호 인증을 수행할 수 있다. 그림에서 $Q_{1/3}$은 인증을 위해 태그가 데이터베이스로 보내는 정보이고 $Q_{2/3}$는 비밀 정보 K를 갱신하는



[그림 2] 태그의 랜덤 수 발생기가 없는 인증 프로토콜 I



[그림 3] 한번의 해쉬 함수를 사용하는 경량화된 상호 인증 프로토콜 II

데 사용하고 $Q_{3/3}$은 태그가 데이터베이스로부터 받은 정보를 검증하는데 사용한다. 따라서 [그림 2]에서 태그는 3번의 해쉬 연산으로 상호 인증을 실현할 수 있지만 [그림 3]의 프로토콜 II에서는 한번의 해쉬 함수 연산으로 상호 인증을 수행할 수 있다.

제안하는 고정 ID를 이용하는 상호 인증 프로토콜 II의 특징을 다음과 같이 요약할 수 있다.

① 고정 ID를 사용하는 분산환경 적합형
② 구별 불가능성(indistinguishability)을 비롯한 프로토콜의 안전성 제공
③ 태그에서 랜덤 수 발생기가 없음
④ 데이터베이스 및 태그에서의 연산이 용이(1번의 해쉬 연산)
⑤ 비동기(desynchronization) 발생 시 동기복구

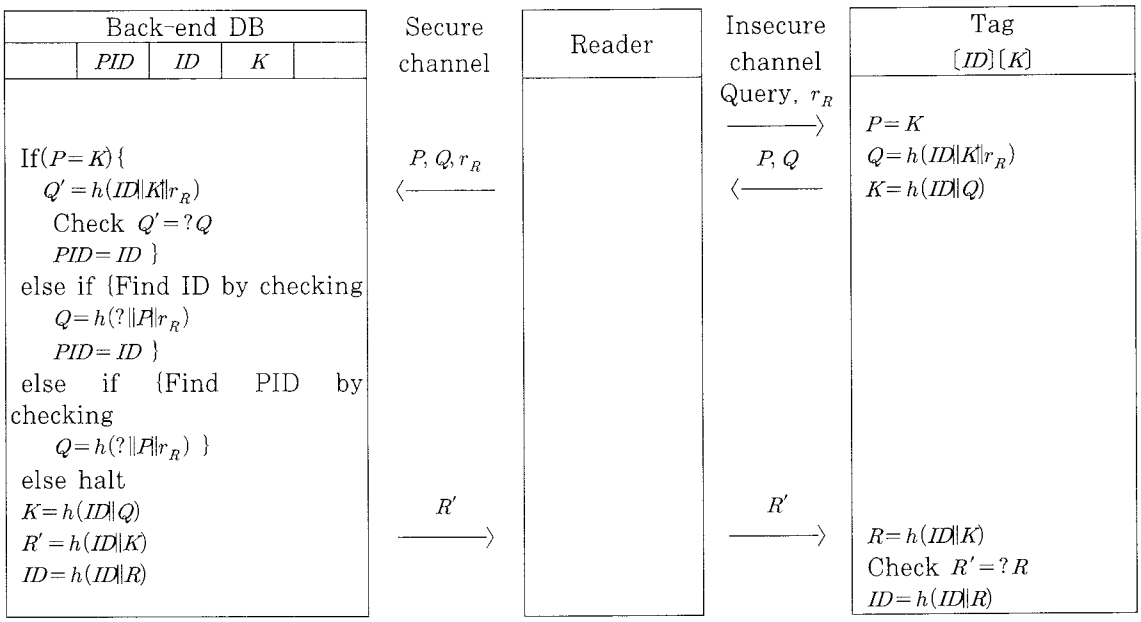
제안하는 인증 프로토콜 I과 II는 고정 ID를 사용하므로 분산환경에 유용하게 적용할 수 있다. 즉, 분산환경하에서는 지역적으로 떨어져 있어 독립된 여러 대의 데이터베이스 서버를 가지고 있을 수 있는데 이 경우 데이터베이스 A에서 인증을 수행한 태그가 데이터베이스 B가 있는 지역에 가서 인증을 수행한다고 가정하자. 이 경우 중앙 서버를 두지 않는다면 두 데이터베이스가 가지고 있는 K값이 다를 수 있다. 하지만 이 경우에도 태그 ID를 가진 정당한 서버 B라면 리더가 보낸 정보 P, Q, r_R(프로토콜 II에서는 $P, Q_{1/3}, r_R$)를 이용하여 $Q=h(?\|P\|r_R)$(프로토콜 II에서는 $Q_{1/3}=h(?\|P\|r_R)_{1/3}$)를 검사하여 해당 ID를 찾고 K를 다시 재동기화하여 인증을 계속 수행할 수 있다.

3.2 변동 ID를 사용하는 인증 방식 분석

프로토콜 I과 II의 경우, 일정한 시점에서 태그의 ID와 K값이 노출되면 이전 세션에서의 통신 정보를 알고 있는 공격자는 Q를 계산할 수 있어 위치 추적이 가능하다. 즉, 전방향 안전성을 만족하지 못한다. 일반적으로 RFID 인증 프로토콜에서 전방향 안전성까지 만족하는 안전도를 제공하기 위해서는 변동 ID 기법을 사용하기도 한다. 그러나 고정 ID 기법은 ID를 EPC로 사용할 수 있지만 변동 ID를 사용하는 경우에는 변하는 ID와 EPC의 연관성을 나타내는 필드를 서버가 유지해야 한다.

본 논문에서는 변동 ID를 사용하여 전방향 안전성을 제공하면서 인증 프로토콜을 경량화할 수 있는 방법을 제안하고자 한다. 따라서 [그림 4]에 기술한 프로토콜 III과 같이 매 세션마다 ID를 변동시키면 전방향 안전성을 만족할 수 있다. 이 경우 ID를 매 세션마다 변경시키면 특정 시점에서 ID와 비밀 정보 K를 알았고 이전의 통신한 내용을 알고 있다하더라도 해쉬 함수의 일방향성에 의해 이전에 사용된 ID 값을 알 수 없으므로 위치추적과 같은 공격은 피할 수 있다.

제안 프로토콜 III에서 태그는 상호 인증을 위해 최소 4번의 해쉬 연산이 필요하지만 랜덤 수 발생은 없도록 설계하였다. [그림 4]와 [그림 2]와의 차이는 태그와 서버가 상호 인증과정의 맨 마지막에 ID를 동일하게 갱신하는 $ID=h(ID\|R)$ 과정이 추가된 것이다. 또한, 본 논문에서는 태그와 데이터베이스의 연산량을 더 줄이기 위해 해쉬를 한번만 수행하는 프로토콜을 제시하고자 한다. 이를 나타낸 것이 [그림 5]이다. 한번의 해쉬 연산만을 사용하기 위해서 $Q=h(ID\|K\|r_R)$ 값을 먼저 구하고 이를 LSB부터 4개의 블록으로 나누어서 이용하고자 한다. 이때 해쉬 함수의 출력은 최소 $2(m+n)$비트 이상이어야 한다. 먼저 Q의 LSB부터 m비트를 $Q_{1/4}$, 다음 n비트를 $Q_{2/4}$, 다음 m비트를

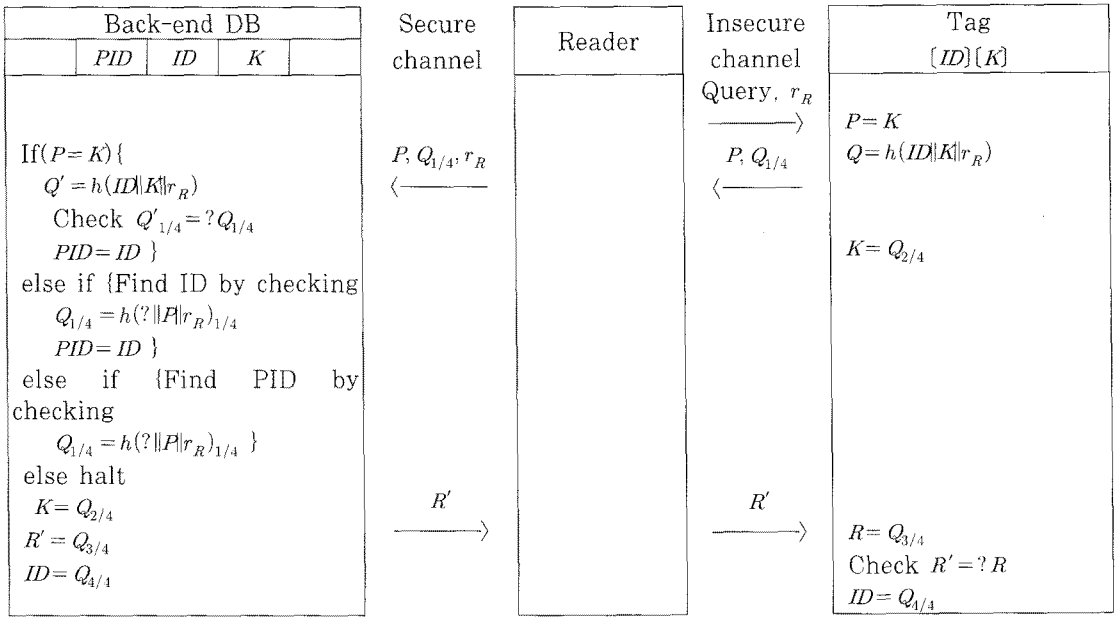


〔그림 4〕 전방향 안전성을 만족하는 인증 프로토콜 III

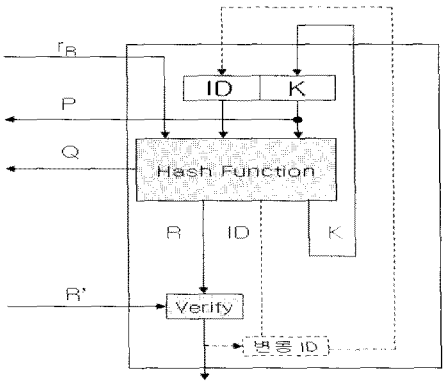
$Q_{3/4}$, 그리고 다음 n비트를 $Q_{4/4}$라고 하자. 만약 $m=80$이고 $n=96$이라면 해쉬 함수의 출력은 최소 352비트가 필요하므로 SHA-384〔15〕와 같은 알고리듬을 사용하면 〔그림 5〕와 같이 한번의 해쉬 연산으로 상호 인증을 수행할 수 있다. 이 중에서 $Q_{1/4}$은 인증을 위해 태그가 데이터베이스로 보내는 정보이고 $Q_{2/4}$는 비밀 정보 K를 갱신하는데 사용한다. 그리고 $Q_{3/4}$은 태그가 데이터베이스로부터 받은 정보를 검증하는데 사용하고, $Q_{4/4}$는 ID를 갱신하는데 사용한다. 따라서 〔그림 4〕에서 태그는 4번의 해쉬 연산으로 상호 인증을 실현할 수 있지만 〔그림 5〕의 프로토콜 IV에서는 단 한번의 해쉬 함수 연산으로 상호 인증을 수행할 수 있다. 제안 프로토콜 III과 IV를 비교해 보면 비교해 보면 프로토콜 III에서 태그나 데이터베이스에서 4번의 해쉬 연산을 하던 것을 프로토콜 IV에서 한번의 해쉬 연산으로 줄여 경량화하였다.

만약 여기서 사용하는 해쉬 알고리듬이 한번에 $2(m+n)$비트 보다 작게 출력하는 알고리듬이라면 해쉬 함수를 두번이상 반복 사용을 함으로써 $2(m+n)$비트 이상을 출력해 내야 한다. 위의 예에서 만약 $m=80$이고 $n=96$이라면 해쉬 함수의 출력은 최소 352비트가 필요하지만 SHA-256을 사용한다면 2번의 해쉬 연산을 수행해야 한다. 이때 2회이상 연속적인 해쉬 값이 필요한 경우는 $Q^i=h(ID||K||r_R)$라 두고 $Q^{i+1}=h(ID||Q^i)$와 같은 방법으로 ID와 이전 Q^i를 연접하여 해쉬할 수 있다. 비슷한 예로 SHA-1을 사용한다면 3번의 해쉬 연산을 수행해야 한다. 즉, $Q^1=h(ID||K||r_R)$의 출력 160비트를 두 부분으로 나누어 각각 80비트인 〔그림 5〕의 $Q_{1/4}$와 $Q_{3/4}$로 사용할 수 있으며, $Q^2=h(ID||Q^1)$의 출력 중 96비트은 $Q_{2/4}$로, $Q^3=h(ID||Q^2)$의 출력 중 96비트는 $Q_{4/4}$로 사용할 수 있다. 결론적으로 상호 인증에 필요한 모든 정보의 출력 비트를 가장 고속으로 출력해 낼 수 있는 해쉬 함수를 반복적으로 사용하는 것이 가장 효과적이다.

지금까지의 제안된 프로토콜을 정리하여 경량화된 태그 내부의 인증 모델을 간략히 도식화한 것이 〔그림 6〕이다. 여기서 점선은 변동 ID를 사용하는 경우를 나타낸다. 여기서 사용하는 해쉬 연산은 해쉬의 출력 크기와 태그의 성능과 보안도를 고려하여 회수를 조절할 수 있으며 해쉬 함수의 출력이 충분히 크다면 한번의 해쉬로도 충분히 상호 인증을 수행할 수 있다. 이 모델에서 해쉬 함수는 리더로부터 받은 랜덤 수 r_R와 ID 그리고 K를 입력으로 $|P|+|Q|+|K|$ 비트(변동 ID 방식에서는 $|P|+|Q|+|K|+|ID|$ 비트)의 출력을 낼 수 있어야 한다. 그리고 P와 Q는 서버로 보내고 서버로부터 받는 R'을 검증하게 된다. 이 과정에서 K는 인증의 종료와 관계없이 갱신을 하며 ID는 태그에서 서버 인증이 종료된 후 갱신하게 된다.



[그림 5] 전방향 안전성을 고려한 경량화된 프로토콜 IV



[그림 6] 상호 인증 모델에서 태그의 동작 과정

IV. 제안 인증 프로토콜의 분석 및 구현

본 장에서는 제안한 상호 인증 프로토콜의 안전성과 효율성을 알아보고 이를 실제로 구현한 결과를 제시하고자 한다. 안전성 분석을 위해 공격자의 공격 능력을 가정할 필요가 있는데 공격자는 태그와 리더간의 주고 받는 정보를 무선으로 도청할 수 있으며 정보에 대한 전파 방해, 인터럽트 혹은 블로킹 공격이 가능하다고 가정한다. 또한 도청한 정보를 변조하여 태그나 리더에게 보낼 수 있는 능동적인 공격도 가능하다고 본다.

4.1 안전성과 효율성

4.1.1 도청

전송되는 정보는 해쉬 함수의 결과이므로 이 정보로부터 ID나 비밀 정보 K를 유추해 낼 수 없다. 프로토콜 II에서는 $Q=h(ID\|K\|r_R)$와 같이 계산되는데 다음 세션에서는 이전 세션에서 사용하였던 Q, K, r_R을 모두 알 수 있다. 그러나 이 경우에도 해쉬 함수의 일방향성으로 인해 ID값은 계산할 수 없다. 프로토콜 IV에서는 같은 원리로 Q, K, r_R만 도청해서는 ID값을 계산할 수 없다. 더구나 프로토콜 IV에서는 매번 ID 값이 변하기 때문에 ID를 찾는 공격은 더 어렵게 된다. 그러나 도청 정보로부터 ID나 다음 세션의 비밀 정보 K를 직접적으로 찾는 것은 어렵지만 도청은 다른 공격을 위한 정보 수집 차원의 공격이므로 이에 기반한 다른 제반 공격에 주의해야 한다.

4.1.2 스푸핑(Spoofing) 공격

공격자가 올바른 리더나 서버로 가장하여 태그를 속이기 위해서는 올바른 R'값을 계산해서 전송해야 하지만 ID나 비밀 키 K를 알지 못하면 이를 계산할 수 없어 스푸핑 공격을 할 수 없다. 역으로 공격자가

태그로 위장하고자 하는 경우에는 올바른 P나 Q 값을 계산해야 하지만 이 역시 ID나 비밀키 K를 알지 못하면 계산할 수 없어 안전하다.

4.1.3 위치 추적

제안된 프로토콜들은 매 세션마다 K 값이 갱신되므로 매번 전송되는 P값들이 랜덤하게 변경되어 이전 세션과 동일한 값을 전송하지 않는다. 따라서 공격자는 매번 다른 세션 정보로 인해 태그의 위치를 추적할 수 없어 위치 프라이버시가 보장된다. 즉, 인접한 두 태그를 구별해 낼 수 없는 구별 불가능성(indistinguishability)를 만족한다. 그러나 제안된 프로토콜 Ⅰ, Ⅱ를 비롯하여 고정 ID를 가지는 인증 방식은 대부분 전방향 안전성를 만족하지 못한다. 즉, 이전의 모든 통신 메시지와 현재 시점에서 특정 태그의 ID와 K를 알고 있다면 이전의 태그 위치를 추적할 수는 있다. 그러나 이전 세션의 모든 통신 메시지를 연속적으로 수집하는 것은 공격자의 많은 능력이 요구되므로 실제로 실현되기에는 어려움이 있다.

4.1.4 비동기(Desynchronization) 공격

공격자가 메시지 전송을 방해하여 정상적인 인증 과정이 실패하였을 경우, 태그와 데이터베이스는 비동기 상태에 빠질 수 있다. 즉, 어떤 세션에서 태그는 K 값을 갱신했지만 인증에 필요한 정보가 차단되어 데이터베이스에서는 갱신되지 못했다고 가정하자. 그러나 이 경우에도 제안된 인증 프로토콜들은 리더가 보낸 정보 P, r_R, Q의 부분정보를 이용하여 $Q=H(?\|P\|r_R)$과 같은 연산을 통해 ID를 찾고 K를 다시 복구할 수 있다. 따라서 제안된 프로토콜들은 비동기 공격에도 자동 복구 기능을 가지게 된다.

Ⅲ장에서 제안된 상호 인증 프로토콜의 안전성과 효율성을 정리한 것이 〔표 1〕이다. 제안된 프로토콜 Ⅰ과 Ⅲ은 정상적으로 인증 과정이 완료되었을 경우, 데이터베이스와 태그에서 각각 3번과 4번(단, 접속이 상으로 비동기 상태에서는 평균 $\lceil l/2 \rceil+2$번 혹은 $\lceil l/2 \rceil+3$번이 필요)의 해쉬 연산을 수행한다. 그러나 제안된 프로토콜 Ⅱ와 Ⅳ는 정상적으로 인증 과정이 완료되었을 경우, 데이터베이스와 태그에서 각각 단 한번(단, 비동기 상태에서는 $\lceil l/2 \rceil$번 혹은 l번)의 해쉬 연산만을 수행한다. 그리고 제안 방식에서는 기존의 다른 방식과 달리 랜덤 수 발생기를 사용하지 않고도 인증을 수행할 수 있는 장점이 있다.

물론 해쉬 함수의 출력 크기, 비밀 키 K 나 ID의 크기에 따라 안전도와 효율성은 상호보완(trade-off) 관계가 있으므로 가급적이면 큰 길이의 출력을 내는 해쉬 함수의 구현이 필요하다. 그러나 구현의 제한으로 인해 인증에 필요한 전체 정보 길이보다 해쉬 함수의 출력이 작으면 해쉬를 한번이상 수행하도록 구현하

〔표 1〕 안전성과 효율성 비교

구 분	안전성			계산량			고정/변동ID	장·단점
	스푸핑 공격	구별 불가능성	전방향 안전성	*DB* 해쉬	태그 해쉬	태그 랜덤 수 발생기		
CRAP〔6〕	○	○	×	$\lceil l/2 \rceil+1$	2	1	고정	*DB*의 과중한 계산량
OHLCAP〔7〕	×	×	×	1	1	1	고정	안전성 부족
LCRP〔8〕	○	×	○	4	4	1	변동	추적공격 위협
LCAP〔9〕	×	○	○	$\lceil l/2 \rceil+2$	3	1	변동	위장공격 위협 *DB*의 과중한 계산량
LRMAP〔10〕	○	○	○	3	3	1	변동	랜덤 수 발생기가 필요
Ha et al.〔11〕	○	○	×	3	3	1	고정	랜덤 수 발생기가 필요
제안 Ⅰ	○	○	×	3	3		고정	랜덤 수 발생기 필요없음
제안 Ⅱ	○	○	×	1	1		고정	초경량화 가능
제안 Ⅲ	○	○	○	4	4		변동	전방향 안전성
제안 Ⅳ	○	○	○	1~3*	1~3*		변동	전방향 안전성 초경량화 가능

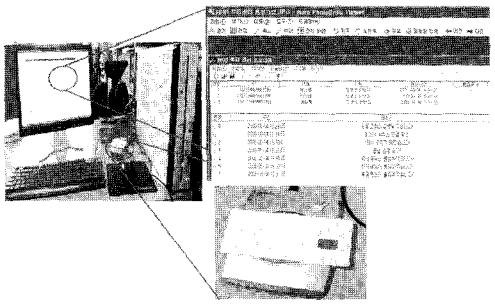
○: 안전, ×: 불안전, l : 시스템의 전체 태그 수

* : 해쉬 함수의 출력 길이와 인증에 필요한 전체 정보 길이에 따라 가변

여야 한다. 그리고 큰 길이의 출력을 내는 해쉬 함수 수행 시간이 상당히 큰 경우에는 이보다 빠른 해쉬 함수를 여러 번 사용하는 것도 효율성을 높이는 대안이 될 수 있다.

4.2 구현 및 검증

본 논문에서는 제안 방식 II와 VI를 RFID 태그로 쓰이는 스마트 카드에 실제로 구현하여 그 동작이 정확함을 검증하였다. 실험에 사용된 RFID용 비접촉식 스마트 카드 칩은 STMicroelectronics사의 ST19WR66 마이크로프로세서이며 [그림 7]은 실험 환경을 보인 것이다.

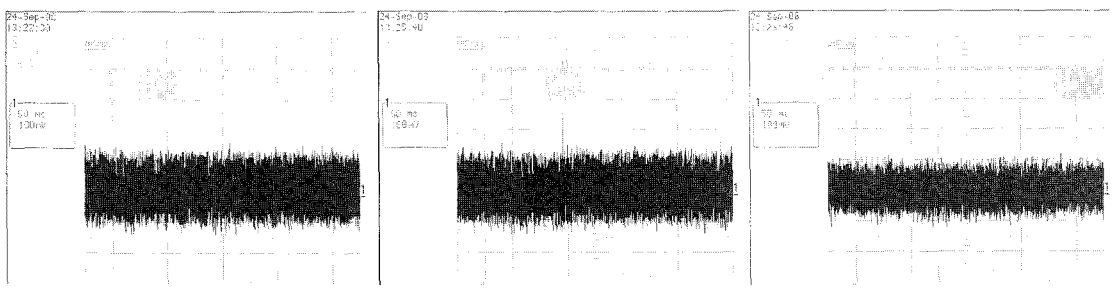


[그림 7] RFID 인증 프로토콜의 구현 환경

제안방식 II에서는 해쉬 함수 SHA-256을 태그에 구현하여 사용하였으며 제안 방식 VI에서는 SHA-1, SHA-256, SHA-384를 구현하여 실험하였다. 여기서 SHA-1과 SHA-256은 512비트를 하나의 블록으로 처리하며 SHA-384는 1024 비트를 하나의 블록으로 처리한다. 따라서 해쉬 함수의 입력 정보가 처리 블록보다 큰 경우에는 두 번 이상의 해쉬 연산이 필요하다. 구현에는 GIN(General Identifier)-96 코드 체계를 가정하여 ID와 K의 크기는 $m=96$비트로 사용하였으며 Q, R 그리고 r_R은 모두 $n=80$비트를 사용하였다. 따라서 제안방식 II에서 사용한 해쉬 함수는 입력 256비트에 SHA-256의 출력을 모두 사용하였으며($|Q|+|R|+|K|=2n+m=256$), 제안방식 VI에서는 해쉬 입력 256비트에 출력 352비트만을 사용하였다($|Q|+|R|+|K|+|ID|=2(n+m)=352$). 태그에서 한번의 해쉬 함수를 연산하는데 걸린 시간은 SHA-1이 50ms, SHA-256이 95ms, SHA-384가 360ms의 시간이 소요되었다. 그 이유는 해쉬 함수의 수행 시간은 연산과정에서 처리하는 입력 블록이나 데이터 길이 혹은 라운드 수에 의해 좌우되기 때문이다.

[그림 8]은 SHA-1, SHA-256, SHA-384를 수행하는데 걸리는 시간을 측정하기 위해 오실로스코프로 그 파형을 관측한 것이다. 그림에서 위쪽 파형은 해쉬 함수의 시작과 끝을 나타나내는 글리치 신호를 표시한 것이며 아래는 소비 전압을 표시한 것이다. 파형에서 X축의 눈금 한 칸은 50ms를 나타낸다. 따라서 오실로스코프 위쪽 파형을 보면 SHA-1, SHA-256, 그리고 SHA-384의 수행시간이 각각 50ms, 95ms, 360ms임을 볼 수 있다. SHA-384은 SHA-256보다 약 3.7배의 수행시간이 필요하였다. 따라서 제안 방식 VI의 경우는 한번의 SHA-384를 수행하여 [그림 5]의 $Q=h(ID\|K\|r_R)$를 수행하는 것보다 두번의 SHA-256을 수행하는 것이 효과적이다. 즉, $Q^1=h(ID\|K\|r_R)$를 계산하여 $Q_{1/4}$와 $Q_{2/4}$를 구하고 $Q^2=h(ID\|Q^1)$를 계산하여 $Q_{3/4}$와 $Q_{1/4}$를 구하면 SHA-256 연산 시간의 2배인 190ms 이내에 인증 기능을 수행할 수 있어 SHA-384를 한번 수행하는 것 보다 훨씬 효과적인 결과를 나타내었다. SHA-1을 이용한 경우에는 $Q^1=h(ID\|K\|r_R)$과 $Q^2=h(ID\|Q^1)$, $Q^3=h(ID\|Q^2)$을 구하여 인증에 필요한 정보로 사용했을 때 수행시간이 약 150ms로 가장 빠른 결과를 보였다.

결론적으로 제안 프로토콜에서는 상호 인증에 필요



[그림 8] SHA-1, SHA-256, SHA-384 수행 시간 측정

한 정보 길이에 따라 전체적인 수행 속도가 결정되었지만 프로토콜 II에서는 SHA-256 1번으로 처리할 수 있었다. 반면 프로토콜 VI은 한번의 SHA-384로 구현할 수도 있지만, 2번의 SHA-256 그리고 3번의 SHA-1으로도 구현할 수 있고, 이 중에서 3번의 SHA-1으로 처리하는 것이 효율성 면에서 가장 효과적이었다.

기존의 검색정보 사전 동기화를 이용한 인증 방식[11]에서는 태그에서 1번의 랜덤 수 발생과 3번의 해쉬 연산이 필요하다. 이때 랜덤 수 발생기는 이전 세션에서 발생한 랜덤 수를 저장해 두었다가 이를 입력으로 하는 해쉬 연산을 수행하여 처리할 수도 있으므로 한번의 랜덤 수 발생기는 한번의 해쉬 연산으로 대체할 수도 있다. 따라서 태그는 최소 SHA-1을 4번 연산하게 되므로 최소 200ms정도의 연산 시간이 필요하게 된다.

그러나 제안방식 I 에서는 Ha 등의 방식[11]에 비해 태그의 랜덤 수 발생에 필요한 연산을 줄일 수 있어 약 150ms 연산 시간이 필요하게 되어 효율적이다. 또한 제안 방식 II는 태그의 모든 연산을 SHA-256 해쉬 함수 1번으로 줄일 수 있어 약 95ms 정도만 필요하게 되어 고속 인증이 가능하다. 제안방식 III은 Ha 등의 방식에 비해 해쉬 함수 연산수는 많아 약 200ms의 시간이 소요되지만 랜덤 수 발생기를 필요로 않으며 전방향 안전성을 제공할 수 있다. 또한 제안 방식 IV는 SHA-1을 3번만 사용하므로 약 150ms 연산 시간이 필요하며 전방향 안전성을 제공할 수 있다. 따라서 제안하는 인증 방식들은 기존 방식에 비해 효율성과 안전성면에서 향상된 성능을 보이며 전방향 안전성 제공 유무나 태그내의 해쉬 함수의 구현 가능성을 고려하여 RFID 인증 시스템에 유연하게 적용할 수 있다.

V. 결 론

본 논문에서 RFID 시스템의 보안 요구 사항을 만족하는 효율적이고 경량화된 상호 인증 프로토콜을 제안하였다. 본 논문에서 제안한 프로토콜은 태그에서 랜덤 수 발생기를 구현할 필요가 없는 것이 특징이다. 또한, 해쉬 함수 수행 결과를 분할하여 인증을 위한 전송 정보, 비밀 키 갱신을 위한 정보 혹은 ID를 갱신하는 정보로 사용함으로써 단 한번 혹은 최대 3번의 해쉬 연산으로 태그와 데이터베이스간의 상호 인증을 실현할 수 있는 경량화된 프로토콜을 제안하였다. 그리고 이를 RFID용 스마트 카드에 구현하여 실시간 내에 동작됨을 실험적으로 검증하였다.

참 고 문 헌

[1] Auto-ID Center, "860MHz-960MHz Class I Radio Frequency Identification Tag Radio Frequency and logical Communication Interface Specification Proposed Recommendation Ver. 1.0.0," Technical Report, MIT-AUTOID-TR-007 AutoID Center, MIT, pp. 1-19, Nov. 2002.

[2] International Standard ISO/IEC 18000-6: Information technology- Radio frequency identification for item management -Part 6: Parameters for air interface communications at 860MHz to 960MHz, 2004.

[3] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," Security in Pervasive Computing 2003, LNCS 2802, pp. 201-212, 2003.

[4] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Hash-Chain Based Forward Secure Privacy Protection Scheme for Low-Cost RFID," In proceedings of the SCIS'04, pp. 719-724, Sep. 2004.

[5] D. Henrici and P. Müller, "Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers," Proceeding of the Second IEEE Annual Conference on Pervasive Computing and Communication Security, pp. 149-153, Mar. 2004.

[6] K. Rhee, J. Kwak, S. Kim, and D. Won, "Challenge-Response Based on RFID Authentication Protocol for Distributed Database Environment," SPC'05, LNCS 3450, pp. 70-84, 2005.

[7] E. Choi, S. Lee, and D. Lee, "Efficient RFID Authentication Protocol for Ubiquitous Computing Environment,"

EUC-2005, LNCS 3823, pp. 945-954, 2005.

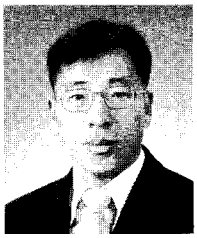
[8] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," Security and Privacy for Emerging Areas in Communications Networks-2005, pp. 59-66, Sep. 2005.

[9] S. Lee, Y. Hwang, D. Lee, and J. Lim, "Efficient Authentication for Low-cost RFID Systems," ICCSA'05, LNCS 3480, pp. 619-627, 2005.

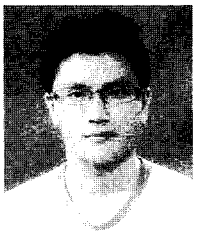
[10] J.C. Ha, J.H. Ha, S.J. Moon, and C. Boyd, "LRMAP: Lightweight and Re-synchronous Mutual Authentication Protocol for RFID System," ICUCT'06, LNCS 4412, pp. 80-89, 2006.

[11] 하재철, 박제훈, 하정훈, 김환구, 문상재, "검색정보 사전 동기화를 이용한 저비용 RFID 인증 방식," 정보보호학회논문지, 18(1), pp. 77-88, 2008년 2월.

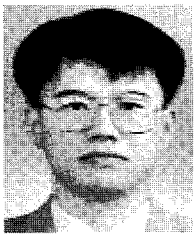
〈著者紹介〉



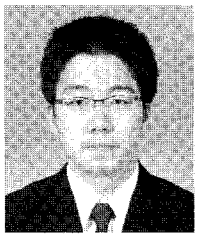
하 재 철 (Jae Cheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 졸업
1993년 8월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월~2006년 1월: 나사렛대학교 전자계산소장, 학술정보관장, 입시학생처장
1998년 3월~2007년 2월: 나사렛대학교 정보통신학과 부교수
2006년 7월~2006년 12월: QUT in Australia 연구 교수
2007년 3월~현재: 호서대학교 정보보호학과 부교수
2002년 3월~현재: 한국정보보호학회 이사, 논문지 편집위원
2009년 1월~현재: 한국산학기술학회 이사
〈관심분야〉 정보보호, 네트워크 보안, 스마트카드 보안



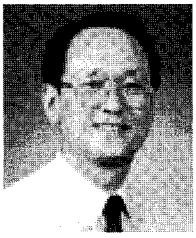
백 이 루 (Yi Roo Baek) 학생회원
2008년 8월: 호서대학교 정보보호학과 졸업
2008년 9월~현재: 호서대학교 정보보호학과 석사과정
〈관심분야〉 네트워크 보안, 프로토콜, 스마트 카드 보안



김 환 구 (Hwan Koo Kim) 종신회원
1987년 2월: 경북대학교 수학과 졸업
1991년 2월: 경북대학교 대학원 수학과 이학석사
1998년 5월: U. of Tennessee-Knoxville, 수학과, Ph. D.
2002년 3월~현재: 호서대대학교 정보보호학과 부교수
2004년 3월~현재: 한국정보보호학회 이사
〈관심분야〉 평가 및 인증, 암호학



박 제 훈 (Jea Hoon Park) 학생회원
2004년 2월: 경북대학교 전자·전기공학부 졸업
2006년 2월: 경북대학교 전자공학과 석사
2006년 3월~현재: 경북대학교 전자공학과 박사과정
〈관심분야〉 정보보호, 네트워크 보안, 스마트카드 보안



문 상 재 (Sang Jae Moon) 종신회원
1972년 2월: 서울대학교 공업교육(전자전공)과 학사
1974년 2월: 서울대학교 전자공학과 석사
1984년 6월: 미국 UCLA 전기공학과 박사
1984년 7월~1985년 6월: UCLA Postdoctor 근무
1984년 7월~1985년 6월: 미국 OMNET 컨설턴트
1997년 9월~1998년 8월: 경북대학교 전자전기공학부 학부장
1974년 12월~현재: 경북대학교 전자전기컴퓨터공학부 교수
2000년 8월~현재: 경북대학교 이동네트워크 정보보호기술 연구센터장
2002년 2월~현재: 한국정보보호학회 명예회장
〈관심분야〉 정보보호, 디지털 통신, 이동 네트워크