한국산학기술학회논문지
Vol. 7, No. 4, pp. 642-648, 2006

동적 ID 할당을 이용한 고기능 RFID 태그용 보안 프로토콜

박진성[1*]

A Secure Protocol for High-Performance RFID Tag using Dynamic ID Allocating

Jin-Sung Park[1*]

요 약 본 논문에서는 상호인증을 통하여 RFID에 안전하게 동적으로 ID를 부여하는 보안 프로토콜을 제안한다. 현재 RFID 태그의 보안을 위해 제안된 대부분의 방식들은 계산 능력과 메모리 저장 능력에 많은 제약을 가지고 있는 저가형 태그에 초점을 맞추고 있다. 그러나, 이러한 문제는 보다 우수한 계산 능력과 큰 저장 공간을 가지는 고가형/고기능 RFID 태그에서는 다른 접근 방식으로 해결될 수 있는 문제이다. 따라서, 본 논문에서는 저가형 RFID용 보안 프로토콜보다는 강력하지만 스마트카드의 보안 프로토콜보다는 간단하면서 동적인 ID 부여와 상호인증이 가능한 프로토콜을 제안한다. 이러한 고기능 RFID와 보안 프로토콜은 명품 브랜드 제품이나 고가 의약품의 관리, 문화재 관리 등과 같은 고가이면서 엄격한 보안과 관리가 필요한 분야에 적용할 수 있다.

Abstract In this paper, I have proposed a secure dynamic ID allocation protocol using mutual authentication on the RFID tag. Currently, there are many security protocols focused on the low-price RFID tag. The conventional low-price tags have limitation of computing power and rewritability of memory. The proposed secure dynamic ID allocation protocol targets to the high-performance RFID tags which have more powerful performance than conventional low-price tag by allocating a dynamic ID to RFID using mutual authentication based on symmetric encryption algorithm. This protocol can be used as a partial solution for ID tracing and forgery.

Key Words : Security, RFID, Protocol, Privacy

1. 서론

근래 들어 RFID 시스템은 공급망 관리를 시작으로 생산, 재고관리 분야는 물론 다양한 산업 전반에서 관심을 받고 있다. 현재 유통물류 분야에서는 바코드를 대체한다는 전제를 바탕으로 작고 값싼 태그에 대한 집중적인 연구가 진행되고 있으며 그에 따른 보안 문제와 프라이버시 보호 또한 해결해야할 문제이다[1]. 저가형 RFID 태그의 경우 단일의 고정된 ID를 저장하고 있기 때문에 이를 도청하면 위치 추적과 이동 경로를 파악할 수 있어 개인의 사생활 침해가 가능하며, 위조된 ID를 구별할 수 없는 등의 보안 문제가 그 대표적인 예이다[2]. 이러한 문제를 해결하기 위해 Kill 명령[3], 해쉬-락[2]등의 다양한 보안 방식이 연구되고 있으나, 모두 RFID 태그 가격을 5센트 미만으로 목표하고 있어 그 구현과 보안성에 상당한 제약이 따른다. 그러나, 이러한 문제는 보다 우수한 계산 능력과 큰 저장 공간을 가지는 고가형/고기능 RFID 태그에서는 다른 접근 방식으로 해결될 수 있는 문제이다. 따라서, 본 논문에서는 저가형 태그보다는 향후에 더 사용이 활성화될 고기능(High-Performance) RFID를 대상으로 그에 적합한 동적 ID 할당 프로토콜을 제안한다. 이미 RFID에 상호인증과 대칭형 암호화 알고리듬을 적용하는 연구가 진행되고 있으나, 고기능 RFID보다는 저가형 태그를 중심으로 하고 있다[4]. 고기능형 RFID는 현재의 스마트카드가 채택하고 있거나 그 이상의 계산능력을 가지는 RISC CPU에 대용량의 EEPROM 혹은 FRAM과 같은 재기록 가능한(rewritable) 메모리를 가지고 있으며, 저전력 문제를 해결하기 위해 자체 전원을 내장하는

[1](주)씨이엔
*교신저자: 박진성(asicman7@yahoo.co.krr)

semi-active 혹은 active형 태그라고 상정한다[5]. Auto-ID 센터에 따르면, 현재의 저가형 태그는 클래스 0와 클래스 I에 해당하며, 향후 클래스 II에서 클래스 V 이상으로 진화하는 경우 암호화 기능과 메모리를 내장할 것으로 전망하고 있다[5]. 2006년 상반기 확정 예정인 국제규격 ISO 18000 Part 6 Type C가 바로 클래스 I의 Generation 2로, 클래스 II이상으로 진화하는 것도 머지않으리라 예상된다(그림 1 참조). 하지만, 고기능 RFID 태그는 저가형 RFID 태그를 교체하는 것이 아니라 저가형 RFID 보다는 좀 더 강력한 보안이 필요한 응용 분야에 사용될 것으로 전망된다. 이러한 고기능 RFID 태그에서는 단순한 수동형 동작뿐 아니라 능동적으로 외부 환경에 따라 동작하는 기능도 가능하기 때문에 이를 고려한 보안 방식이 요구되어진다. AES와 같은 대칭형 암호화 알고리듬으로 인증을 수행하는 알고리듬들이 제안되고 있지만, 아직까지는 단순한 인증에만 초점을 맞추고 있다[4]. 고기능형 RFID는 개방형 스마트카드가 채택하고 있는 GOP(Global Open Platform)의 상호 인증 프로토콜[6]을 수행하기에는 통신 시간의 제약[1]을 받기 때문에 그보다는 좀더 간단한 인증 프로토콜이 필요하다. 또, 고정된 ID 때문에 발생하는 보안 문제를 해결하기 위해 수시로 새로운 ID를 암호화하여 부여하고 서버는 항상 새로운 ID를 통해 태그를 인식하게 하면 설령 공격자가 ID를 추적한다 하더라도 이전 ID와 변경된 ID간의 상관관계를 알 수 없기 때문에 위치 파악이 불가능하다. 따라서, 본 논문에서는 저가형 RFID용 보안 프로토콜보다는 강력하지만 스마트카드의 보안 프로토콜보다는 간단하면서 동적인 ID 부여와 상호인증이 가능한 프로토콜을 제안한다.

본 논문의 2장에서는 기존 RFID의 보안 방식에 대해 알아보고, 3장에서는 제안한 프로토콜에 대하여 설명하였다. 4장에서는 제안된 프로토콜에 대한 보안 특성을 분석한다. 마지막으로 결론은 5장에서 논하였다.

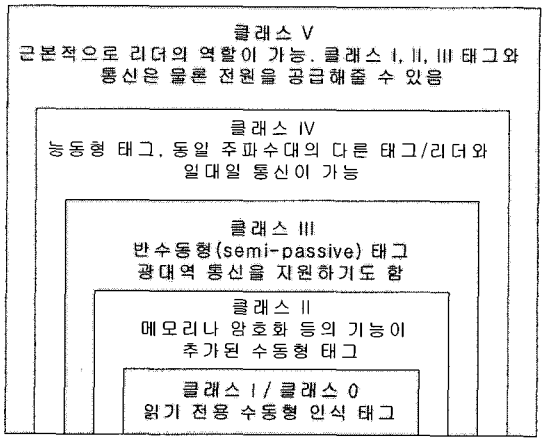


그림 1. RFID의 클래스 구분

2. RFID 보안

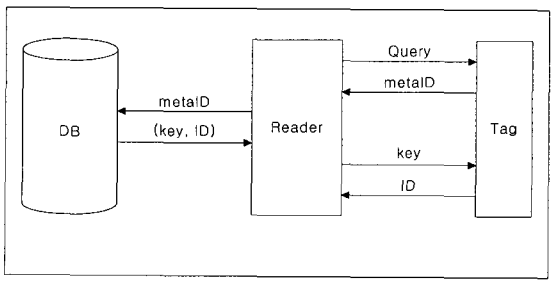
2.1 기존의 RFID 보안 방식

2.2.1 Kill 명령어[3]

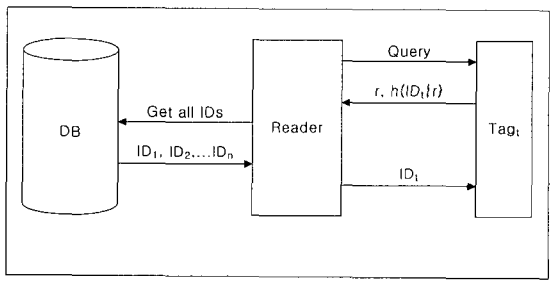
MIT의 Auto-ID 센터(현재 EPCglobal)에서 제안한 방식으로 태그에 8비트의 패스워드를 내장하고 있다가 동일한 패스워드와 Kill 명령이 전달되면 자신의 모든 기능을 중지시켜 다시는 사용할 수 없도록 하는 것이다. 이 명령을 실행하면 태그 내부의 회로들이 완전히 단락되며, 한 번 죽은 태그는 되살릴 수 없게 되어 태그를 재사용할 필요가 있는 분야에는 적용이 불가능하다. 이 방식은 현재 EPC 클래스 1 태그와 ISO 18000 Part 6 Type C 태그에 기본 기능으로 내장되고 있다.

2.2.2 해쉬-락 방식[2]

이 방식에서 리더는 각 태그에 대한 키를 가지고 있으며, 태그는 기본적으로 잠김 상태에 있어서 그 키에 대한



(a) 해쉬-락 방식



(b) 랜덤화된 해쉬-락 방식

그림 2. 해쉬-락 방식과 랜덤화된 해쉬-락 방식

해쉬 값 metaID를 저장하고 있다가 리더에게 접근하면 이 metaID를 전송한다. 리더는 이 metaID로부터 키를 유추하여 키 값을 태그로 보내고, 태그는 키에 대한 해쉬 값을 계산하여 자신의 metaID와 일치하는 경우에만 잠김 상태에서 빠져나와 자신의 ID를 리더에게 전송한다. 이 방식은 태그가 가지는 metaID가 항상 일정하기 때문에 추적이 가능한 단점이 있으며, 공격자가 태그의 metaID를 입수하여 정당한 리더에게 재생하여 보내는 경우 리더는 올바른 키를 공격자에게 보내게 되는 위험이 있다.

2.2.3 랜덤화된 해쉬-락 방식[2]

위의 해쉬-락 방식에 가지는 문제를 해결하기 위해 태그는 의사난수생성기를 이용한다. 태그는 자신의 ID와 자신이 생성한 난수로 해쉬를 계산하여 리더로 전송하기 때문에 항상 해쉬 값이 변하게 된다. 리더는 태그로부터 전달된 해쉬값과 난수를 가지고, 서버로부터도 모든 태그의 ID를 받아 수신한 난수로부터 해쉬값을 계산하여 일치되는 태그 ID를 찾은 후 태그로 전송한다. 따라서, 추적을 피할 수는 있으나 태그에 의사난수 생성기를 내장하여야 하며 서버/리더기의 계산량이 많아진다는 부담이 있다.

2.2.4 대칭형 암호화 기반 인증 방식[4]

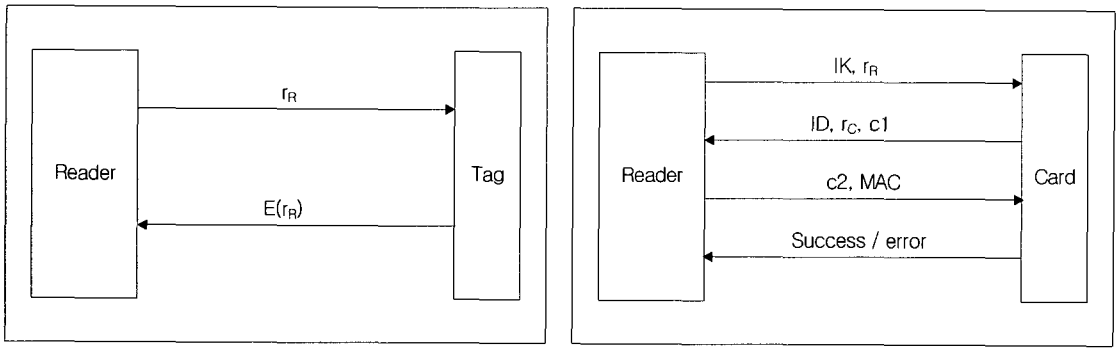
M.Feldhofer 등은 그림 3(a)에 나와있는 것과 같이 대칭형 암호화 알고리듬인 AES를 RFID에 내장하여 리더가 보낸 난수를 태그가 AES로 암호화하여 인증하는 방식을 구현하였다. 또, 이보다 진보된 상호인증 방식도 제안하고 있으나 저가의 수동형 태그에 기반하고 있으며, 13.56MHz 대역에 초점을 맞추고 있어 비접촉식 스마트카드와 유사한 통신환경을 배경으로 하고 있다.

2.2.5 개방형 스마트카드 인증 방식[6]

개방형 스마트카드는 카드 메모리에 저장되는 내용물(프로그램과 데이터)의 추가/삭제가 가능한 스마트카드로, 카드와 리더기 간에 3-DES 암호화 알고리듬에 기반한 보안채널 프로토콜을 통해 상호인증을 수행하여 불특정 다수가 카드 내용물을 함부로 변경하는 것을 방지하고 있다. 이 인증방식은 GOP(Global Open Platform)에 의해 규정되었으며 현재 국제적인 산업규격으로 대부분의 개방형 스마트카드에 채택되고 있다. 그림 3(b)에 나타나 있듯이, 먼저 리더가 카드로 사용할 키 인덱스 *IK*와 자신이 생성한 난수 r_R을 보내면, 카드는 난수 r_C를 생성하고 리더의 난수와 조합하여 세션키를 생성한다. 생성된 세션키로 r_R과 r_C에 대한 암호화를 수행하여 그 최상위 일부만을 인증값 *c1*으로 리더기에 응답하면, 리더는 카드의 난수와 ID를 이용해 동일한 세션키를 생성하고 *c1*을 검증한 다음, 카드에 제시할 *c2*와 MAC(Massage Authentication code)을 생성하여 카드로 전달한다. 카드는 *c2*와 MAC을 세션키로 검증하고, 올바르면 성공코드를 응답하는 것으로 상호인증을 완료하게 된다.

3. 제안 프로토콜

위 2장에서 살펴본 다양한 보안 방식은 저가형 태그에 구현하기 위해 계산량과 간단한 구조를 우선적으로 고려하고 있다. 본 장에서 제안하는 보안 프로토콜은 고기능 태그를 위한 것으로 스마트카드보다는 간단하지만 저가형 태그에 비해서는 강력한 보안 성능을 제공하면서 ID를 부여하는 것을 목표로 한다. 이 방식은 리더 측에서 태그에게 ID를 부여할 때 암호화된 ID를 전송하기 때문에 공격자는 태그의 이전 ID와 부여되는 ID 간의 상관



(a) 간단한 인증 프로토콜 (b) GOP의 상호 인증 프로토콜

그림 3. 암호화 알고리듬을 이용한 인증 방식

관계를 파악할 수 없어 ID를 이용한 위치 추적이 불가능하다. 백엔드 서버는 발급할 ID들을 미리 생성하여 두었다가 새로운 ID를 할당할 필요가 있을 때마다 이를 부여하게 된다. 하나의 태그에 대한 원래의 ID와 새로 부여된 ID의 상관관계와 변경 이력은 모두 백엔드 서버가 관리하므로 오로지 백엔드 서버만이 위치추적에 필요한 정보를 가지게 된다. 부여된 ID는 일반적인 질의(query)의 응답으로 사용되며, 더 이상 필요가 없거나 일정 질의 횟수가 넘어가면 파기되고 새로운 ID를 부여받도록 할 수도 있다. 따라서, 공격자가 동일한 ID를 추적하고 있다 하더라도 얼마안가 새로운 ID로 변경되기 때문에 위치 추적이 불가능해진다. 또한, 재생공격(replay attack)을 방지하기 위해 인증이 일정 회수 이상 실패하는 경우, 태그가 잠김 상태로 전환되어 더 이상 응답하지 않도록 하고 있다.

3.1 준비 단계

그림 4와 같이 태그는 사용되기 전에 발급자가 미리 보안에 사용할 키(Key_1, Key_2, ... Key_n)들과 그 키를 나타내는 키 인덱스(IK_1, IK_2, ... IK_m)를 태그에 저장하며, 백엔드 서버의 데이터베이스에도 동일한 키들과 그 인덱스를 저장하여 둔다. 이 때 키의 개수는 n < m으로 DB는 많은 수의 키들을 준비하고, 개별 태그에는 3~5개 정도의 키를 부여한다. 또한 서버는 태그에 할당할 ID들(Dynamic ID_1, ... Dynamic ID_x)을 미리 준비하여 둔다. 태그의 Fail counter는 외부로부터의 명령이 완성되지 않거나 암호화 데이터 검증에 실패한 경우 증가하는 카운터로 외부 공격을 방지하는데 사용된다. 태그는 자신의 고유 ID(fixed ID)와 동적 ID를 따로 저장하고 있으며, 모드에 따라 고정 ID를 사용하거나 동적 ID를 사용하게 된다. 본 단계에서 리더는 충돌방지 알고리듬을 이용하여 태그를 인식한다. 이 과정에서 태그는 자신이 가진 동적 ID를 계속 사용할지, 아니면 새로운 ID를 부여 받을지에 대해 결정하고 새로운 ID를 부여받기로 결정하였다면, 다음 절의 프로토콜을 통해 ID를 부여받는다고 가정한다. 따라서, 리더가 태그를 인식하는 과정에서 태그가 자신의 ID를 변경하기를 리더에 요청하게 되면 리더가 새로운 ID를 부여하게 된다. 현재의 ID와 새로 부여되는 ID 간에는 코드 체계만 동일하며, 일련 번호 등은 랜덤하게 생성하여 사용하게 되면 그것을 생성하고 관리하는 백엔드 서버 이외에는 상관 관계를 알 수 없게 된다. 예를 들어 ID가 EPC(Electrical Product Code) 체계라면, 64비트의 ID 중 마지막 24 비트가 일련번호로 사용된다. 이 일련번호를 백엔드 서버가 체계적으로 고정 ID와 동적 ID에 할당하여 관리한다면 여기서 제안하는 프로토콜을 통한 ID 부여가 가능할 것이다. 태그가 고정 ID와 동적 ID 중 어떤 ID를 사용할지 결정하는 방법과 동적 ID의 갱신이 어떤 주기나 계기에 의해서 이루어질 것인지는 본 제안에서 다루지 않기로 한다. 본 제안은 단지 동적 ID를 갱신하게 되었을 때, 어떤 방법으로 할당하게 되는지에 초점을 맞추고 있다.

3.2 ID 할당 단계

그림 5에 할당 프로토콜에 대한 흐름도가 나타나 있다. ①과 ③에 대해 GenerateRandom과 ChnageID 명령이라 이름 붙였으며, 이에 대한 자세한 데이터 구조는 Ⅳ장에서 설명한다.

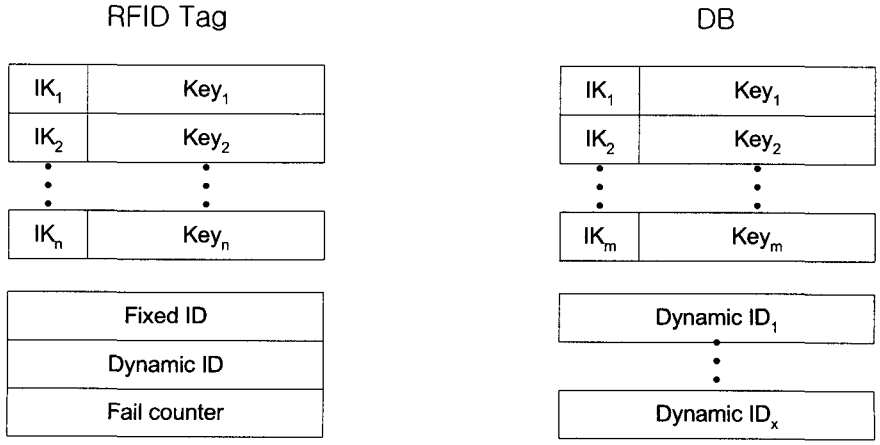


그림 4. 태그와 데이터베이스 저장 정보

① 리더는 태그 인식 과정에서 ID 변경 요청을 받으면 새로 부여할 ID를 준비하고, ID를 변경하고자 하는 태그에게 GenerateRandom 명령을 보낸다. 여기에는 태그의 ID를 함께 전송하여 이 명령을 받은 태그 중에 자신의 현재 ID와 일치하는 태그만 이 명령에 응답하게 된다.

② 태그는 난수 *R*을 생성하여 이를 응답한다.

③ 리더 혹은 백엔드 시스템은 태그의 현재 ID로부터 그 태그에 저장된 키 셋(IK_1....IK_n)을 알 수 있으며, 그 키 중에서 하나를 임의로 선택하여 암호화하여 *EID*를 생성하고, 그에 따른 인증값 *M1*을 생성한다. *EID*를 생성함에 있어 단순히 ID만을 암호화하지 않고, ID와 R의 XOR 연산 결과를 암호화한 이유는, 스푸핑 공격과 재생공격을 막기 위해서이다. 계산이 끝나면, ChangeID라는 명령을 통해 태그로 암호화에 사용된 키의 인덱스 IK, *EID*와 *M1*을 전송한다. 여기서 *E()* 함수는 T-DES나 SEED와 같은 대칭형 암호화 연산이며, *MAC()* 함수는 *E()* 함수를 CBC(Cipher Block Chaining) 모드로 연산하여 그 최종 결과의 최상위 일부분만을 취하는 함수이다.

$$EID = E(ID \ xor \ R) \quad (1)$$

$$M1 = MAC(ID) \quad (2)$$

④ 태그는 IK가 가리키고 있는 키를 사용하여 전달된 *EID*를 복호화하고 자신이 생성한 *R*로 XOR 연산을 수행하여 ID를 추출한 후 인증값을 계산하고 전송되어온 *M1*과 비교한다. 동일한 인증값을 가지면 자신의 메모리 영역에 이 ID를 저장한다. 성공적으로 저장이 완료되면 *M2*를 생성하여 리더기에 응답으로 보낸다. 여기서 | 기호는 데이터 연접 연산을 나타낸다.

$$M2 = MAC(R \mid ID) \quad (3)$$

공격자가 위 과정을 반복하여 키 값을 알아내려는 시도를 하는 경우를 방지하기 위해, 암호화된 ID를 받지 못하거나 인증값 *M2*의 검증이 연속으로 실패하는 경우에는 카운터(Fail counter)가 증가하며, 이 카운터가 일정 한도를 넘어서면 태그는 스스로 잠김 모드로 들어가 특별한 인증 절차를 거치지 않는 한 외부에 응답을 하지 않게 된다. 리더는 전달된 *M2*를 검증하고 ID 부여가 완료되었다는 정보를 DB에 저장한다. 따라서, 리더와 DB를 가지는 백엔드 서버는 하나의 태그에 대해 그 태그의 ID가 언제 어떻게 부여되어 변경되었는지에 대한 이력을 모두 저장하기 때문에 특정 태그를 추적할 수 있는 유일한 주체이다.

4. 제안 프로토콜의 안전성

여기서는 보안 위협에 대해 제안한 프로토콜이 안전함을 설명한다[8].

4.1 도청에 대한 안전성

공격자가 태그의 응답 ②를 도청하여 얻을 수 있는 것은 난수 *R* 뿐이다. 리더의 전송 명령 ③은 키 인덱스(*IK*)

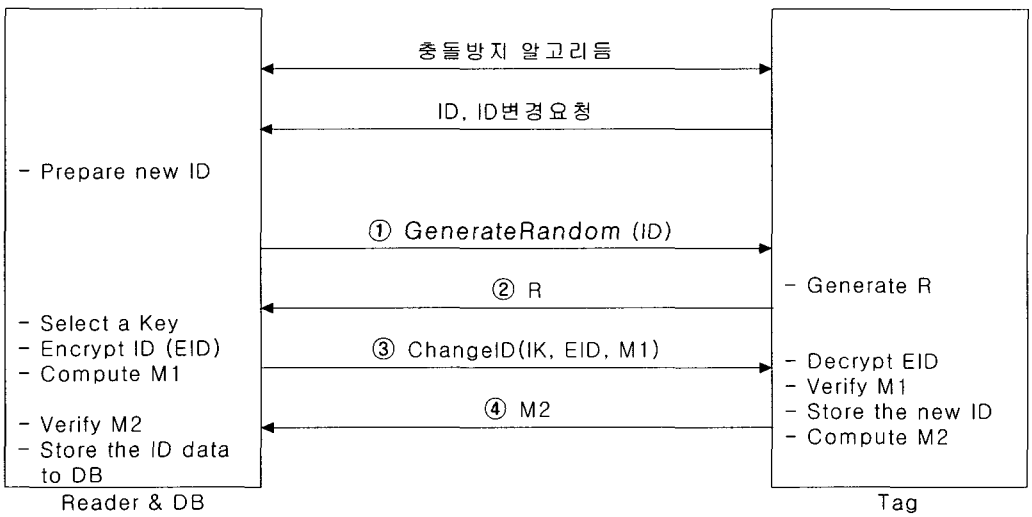


그림 5. 제안한 보안 프로토콜의 흐름도

와 암호화된 ID(*EID*), 그리고 인증값(*MI*)이 전달되므로 도청자는 새로 부여되는 ID를 전혀 알 수 없다. 암호화되지 않은 키 인덱스만을 이용하여 해독할 수 있는 정보는 거의 없으며, 암호화 키가 여러 개의 키 셋 중에서 임의로 선택되어 사용되기 때문에 장시간에 걸친 도청을 하여도 암호화된 *ID*를 해독하기에 충분한 데이터를 수집할 수 없다.

4.2 위치 추적 방지

본 프로토콜의 수행 중에 태그의 ID는 전혀 전송되지 않으며. 그 용도가 동적으로 새로운 ID를 부여하는 것이다. 따라서, 백엔드 서버 이외에는 이전 ID와 새로운 ID와의 상관관계는 물론 ID 부여 이력도 알 수 없으므로 위치 추적은 불가능하다.

4.3 스푸핑 방지

본 프로토콜은 상호 인증을 기반으로 하고 있기 때문에 상대방과 동일한 키를 가지고 있지 않고서는 상대방을 속일 수 없다. 공격자가 태그인척 행동을 한다면 ③을 받고 *EID*를 복호화하여 ID를 복구하여야 하는데, 키를 모르고 있어 불가능하다. 또, 공격자가 리더인척 행동을 한다면, ②를 수신한 후 ID를 암호화하고 인증값을 생성하여 전송해야 하는데, 이 또한 키를 모르고 있으면 불가능하다. 공격자가 리더인척 ①을 발신하고 태그의 ② 응답을 받아도 얻을 수 있는 것은 태그의 난수 밖에 없다. 태그는 내부에 카운터(Fail counter)를 가지고 있어서 ②를 전송하였음에도 불구하고 ③ 응답이 없는 경우와 ③을 받았지만 인증값이 틀리면 카운터가 증가하고, 이 카운터가 누적되어 일정 한도를 넘어서는 경우 잠김 모드로 전환되어 외부에 응답하지 않게 된다. 이 카운터는 연속적인 경우에만 증가하며 성공적으로 이루어지는 경우에는 다시 0 값으로 되돌려진다. 따라서 공격자가 연속적인 공격을 시도하는 경우 태그는 잠김 모드로 전환되어 더 이상 응답하지 않게 된다. 잠겨진 태그에 대한 잠김 해제는 추가적인 보안 명령을 통해 이루어져야 할 것이다.

4.4 재생 공격 방지

공격자가 ②를 도청하였다가 정당한 태그인 척 리더에 전송하는 경우에도 ③을 받은 후 암호화 키를 알지는 못하기 때문에 ID를 추출해 낼 수 없으며, ④를 생성할 수 없거나 틀린 정보를 전송하게 되어 백엔드 측에서 감지할 수 있다. 공격자가 ③을 도청하였다가 정당한 리더인 척 다시 태그에 전송하여도 ③의 *EID*를 연산하는데 태그의 난수 *R*이 포함되어 있기 때문에 ② 단계에서 태그가 전송한 *R*과 도청한 ③에 포함된 R은 다르므로 태그의 인증 과정에서 드러나게 된다.

표 1은 GOP와 제안 프로토콜을 비교한 것으로, 처리 속도에 있어서도 제안 프로토콜이 약 37% 빠른 것으로 측정되었다. 측정 방법은 JCOP31, JATE 등과 같은 개방형 스마트카드(자바카드)에 GOP 인증 프로토콜과 제안 프로토콜을 JAVA 애플릿으로 프로그래밍하여 스마트카드에 다운로드하고, 스마트카드와 리더 간 프로토콜을 실행하여 그 수행 시간을 측정하였다. 이 수행시간은 스마트카드 전용 자바 플랫폼에서 수행된 결과이기 때문에 전용 마이크로프로세서에 최적화된 어셈블리나 C 언어로 구현하는 경우에는 표에 제시된 수치보다 훨씬 빠른 수행 속도가 나올 것이다. 현재의 자바카드는 전용 COS(Chip Operating System)를 사용하는 스마트카드보다 처리 속도면에서 5~10배정도 느리다는 점을 감안해야 할 것이다. GOP의 프로토콜은 단순한 상호인증용이고, 본 논문에서 제안하고 있는 프로토콜은 인증과 동시에 *ID* 부여를 목적으로 하고 있어 단순한 *1:1* 비교는 불가능하지만, 고기능 RFID 태그와 가장 유사한 제품인 스마트카드를 이용하여 제안 프로토콜을 구현하고 GOP 프로토콜과 비교하였다. 이는 GOP와 제안 프로토콜 각각의 수행속도에 대한 상대적인 비교를 하기 위한 것으로, 제안 프로토콜이 GOP 보다 빠르게 처리됨을 보여준다. 구현에 사용된 스마트카드는 8비트 마이크로프로세서를 CPU로 채용하고 있으며 3-DES 처리를 위해 Crypto-processor를 내장하고 있다.

표 1. GOP와 제안 프로토콜의 비교

	GOP 프로토콜	제안 프로토콜
태그 암호화/복호화 횟수	4회	3회
리더 암호화/복호화 횟수	5회	3회
교환 데이터 길이	64 bytes	45 bytes
사용 알고리듬	3-DES	3-DES
처리 시간	Init. Update : 330 ms	GenRandom : 70 ms
	Ext. Auth. : 240 ms	ChnageID : 390 ms
	Total : 570 ms	Total : 460 ms

6. 결 론

RFID의 보안 문제를 해결하기 위해 다양한 프로토콜이 제안되고 있으나, 대부분 저가형 RFID 태그를 대상으로 하고 있다. 본 논문에서는 고기능 RFID를 대상으로 상호인증을 통하여 안전하게 동적인 ID를 부여하는 프로토콜을 제안하였다. 이 프로토콜은 저가형 태그보다는 강력한 보안을 제공하면서, 스마트카드의 상호 인증 방식보다는 간단하고 동시에 ID 부여가 가능한 기능을 목표로 하였다. 또한, 제안 프로토콜이 현재의 국제 규격에 어떻게 활용될 수 있는지 보여주기 위해 ISO 18000 Part 6 Type C 규격에 맞추어 데이터 구조를 구성하였다. 이러한 상호인증 기능을 가지는 고기능 RFID는 향후 명품 브랜드 제품이나 고가 의약품의 관리, 문화재 관리 등과 같은 고가이면서 엄격한 보안과 관리가 필요한 분야에 적용할 수 있으며, 센서를 내장하여 유비쿼터스 환경의 센서 네트워을 구축하는 근간으로 활용될 수 있다[9].

참고문헌

[1] S.E.Sarma, S.A.Weis, and D.W.Engels, "RFID Systems, Security & Privacy Implications", White Paper, Auto-ID Center, MIT, 2002.

[2] S.A.Weis, S.Sarma, R.Rivest, and D.Engels, "Security and Privacy Aspect of Low-Cost Radio Frequency Identification Systems", Springer-Verlag, First International Conference on Security in Pervasive Computing, LNCS 2802, pp.201-212, 2004.

[3] Auto-ID Center, "860-930MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical Communication Interface Specification Candidate Recommendation Version 1.0.1", Technical Report, Auto-ID Center, MIT, 2002.

[4] M.Feldhofer, "A Proposal for an Authentication Protocol in a Security Layer for RFID Smart Tags", MELECON 2004 IEEE Proceedings, pp.759-762, 2004.

[5] S.Sarma, D.W.Engels, "On the Future of RFID Tags and Protocols", Technical Report, Auto-ID Center, MIT, 2003.

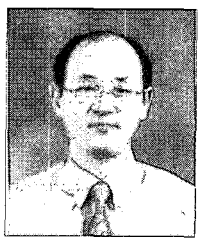
[6] GlobalPlatform, "Card Specification Version 2.1.1", GlobalPlatform, 2003.

[7] ISO/IEC, "Information technology - Radio-frequency identification for item management - Part 6 : Parameters for air interface communications at 860 MHz to 960 MHz", International Standard, ISO, 2005.

[8] 박진성, 최명렬, "고기능 RFID 태그를 위한 동적 ID 할당 프로토콜", 한국정보보호학회 논문지, 제15권 제6호, pp. 49-58, 12월, 2005.

[9] S.Haller, S.Hodges, "The Need for a Universal Smart Sensor Network", Auto-ID Center, White Paper, MIT, 2002.

박 진 성(Jin-Sung Park) [정회원]



- 1995년 2월 : 한양대학교 제어계측공학과 (공학사)
- 1997년 2월 : 한양대학교 제어계측공학과 (공학석사)
- 2006년 2월 : 한양대학교 제어계측공학과 (공학박사)
- 2000년 ~ 2002년 : (주)마니네트웍 개발팀장
- 2003년 ~ 2004년 : 노틸러스효성(주) 개발팀 과장
- 2005년 5월 ~ 현재 : (주)씨이엔 연구소장

<관심분야>
스마트카드, RFID, 정보보호