工學博士學位請求論文

RFID 시스템에서 다중 객체 접근을 위한 암호화 및 인증 기법

An Encryption and Authentication Scheme for Accessing Multiple Objects in RFID Systems

2008 年 8 月

仁荷大學校 大學院

情報工學科

金 芝 娟

工學博士學位請求論文

RFID 시스템에서 다중 객체 접근을 위한 암호화 및 인증 기법

An Encryption and Authentication Scheme for Accessing Multiple Objects in RFID Systems

2008 年 8 月

指導敎授 李 均 夏

이 論文을 博士學位請求論文으로 提出함

仁荷大學校 大學院

情報工學科

金 芝 娟

이 論文을 金芝娟의 博士學位請求論文으로 認定함

2008 年 8 月 日

主審 _____

副審 _____

委員 _____

委員 _____

委員 _____

요 약

최근 활발히 연구되고 있는 RFID 시스템은 여러 산업 분야와 개인의 생활에 있어서 그 응용의 범위를 점차 넓혀가고 있다. RFID 시스템은 비가시 거리에서 무선통신을 이용한 자동식별 기능을 제공하고, 읽기/쓰기를 가능하게 할 뿐만 아니라 다양한 환경 변화에 대한 적응력이 뛰어나 유비쿼터스 환경의 핵심기술로 평가되고 있다. 그러나, 이러한 RFID 시스템은 무선 통신의 불안전한 특성으로 인해 사용자 정보에 대한 추적과 접근이 용이하여 개인 정보 침해의 위험성 또한 증가하게 된다는 문제점을 내포하고 있다. 또한, 기존의 RFID 시스템은 각 응용 객체마다 별도의 태그를 사용하여 각각의 식별 정보를 저장하고 있으므로, RFID 기술을 적용하는 응용 분야가 광범위해짐에 따라 많은 태그들이 혼재하게 되고, 개인 사용자들은 많은 태그를 구분하여 사용해야 하는 불편함이 따르게 된다. 이는 사용자 관점에서의 RFID 시스템에 대한 이용 편의성을 감소시키는 결과를 초래한다. 따라서, RFID 기술을 응용한 시스템의 확산을 위해서는 안정적인 정보 보호를 바탕으로 한 여러 응용 객체들간의 효율적인 정보 공유에 대한 연구가 마련되어야 한다.

본 논문에서는 하나의 태그로 다수 개의 RFID 응용 객체들에 접근할 수 있도록 하는 다중 객체 접근을 위한 RFID 태그 구조를 설계하고, 이러한 태그 내에 저장되는 각 응용 객체에 대한 정보를 보호하기 위한 암호화 알고리즘을 제안한다. 제안하는 알고리즘은 기존의 블록 암호화 기법인 SEED 알고리즘을 RFID 시스템의 특성에 맞도록 경량화한 방법이다. 또한, 본 논문에서는 다중 객체 접근 방식의 태그를 포함하여

RFID 시스템의 구성 요소인 리더와 서버가 사용자 정보 침해를 목적으로 하는 여러가지 공격들로부터 안전하게 정보를 전송할 수 있도록 하는 인증 프로토콜을 제안한다. 제안하는 인증 프로토콜은 여러 응용 객체별로 다르게 요구되는 보안의 수준을 분류하여 보안 레벨을 정의하고, 보안 레벨에 따라 각각 다른 인증 절차를 수행하도록 함으로써, 저사양의 태그 구조 하에서도 인식 동작을 효율적으로 수행하고 다양한 정보 침해를 목적으로 하는 공격들로부터 강인하도록 설계하였다.

태그 정보를 보호하기 위한 SEED 변형 알고리즘의 성능은 암호화 및 복호화의 속도를 측정하여 기존의 SEED 알고리즘과 비교하여 평가한다. 그리고, RFID 시스템에 발생 가능한 다양한 정보 침해 공격들로부터 제안하는 인증 프로토콜의 안전성을 분석하고, 기존의 RFID 인증 프로토콜들과 연산량을 비교하여 효율성을 평가한다. 또한, 시뮬레이션을 통해 인증 세션에 소요되는 시간과 에러율을 측정하여 다중 객체 접근을 위한 RFID 시스템에서 보안 레벨에 따른 인증 프로토콜의 효율성을 평가한다.

Abstract

Recently, RFID systems are spreading in various industrial areas faster and now this time it is coming up to usual area covering individual life and environment. The RFID systems provide technologies of automatic object identification in invisible range, read/write function and adaptability against various circumstances through wireless communication among contactless devices. These advantages make RFID systems to be a core technology of ubiquitous environment. However, RFID systems allow easily expose of user information or tracking of user location because of its unstable wireless communication by radio frequency. Therefore, RFID systems often cause some serious problems such as violation of privacy and information security. Moreover, traditional RFID systems have a restriction that one tag per each application object. That is a tag is used to store identifying information just for a single object in common RFID applications. This restriction deteriorates their usability because it is difficult to distinguish many tags without some kind of effort. Therefore, efficient information sharing of objects based on information security has to be studied for more spreading of RFID technologies.

In this thesis, we design a new RFID tag structure for supporting multiple objects which can be shared by many different RFID applications. That is, the proposed RFID tag structure supports that a tag maintains several different objects and allows those applications to access them simultaneously. We also design an encryption algorithm to protect the identifying information of objects stored in our tag structure. This algorithm is a light revision of the existing SEED encryption algorithm which can be operated in RFID tag environment. In addition,

we propose an authentication protocol to support multiple objects RFID tag. In the proposed protocol, RFID elements process the authentication to prevent various attacks arising serious problems of security and privacy. Especially, we focus on efficiency of the authentication protocol by considering different security degrees in RFID applications. The proposed protocol includes two types of authentication procedures which are distinguished by security levels. In the proposed protocol, an object has its security level and goes through one of different authentication procedures suitable for its security level. This authentication protocol is designed to overcome the problems of security and privacty and has enough robustness against various attacks in low cost RFID systems.

We simulate the proposed schemes and make various experimental results. To evaluate the performance of our encryption algorithm, we measure the encryption and decryption times of this algorithm and compare the results with those of the original SEED algorithm. To evaluate robustness of the proposed authentication protocol, we analyze the safety of against attacks by comparing with the other schemes. We also evaluate the efficiency of the proposed protocol by measuring computation time and error rates during the authentication procedure.

목 차

요 약 i

Abstract iii

제 1 장 서 론 1

1.1 연구 배경 1

1.2 연구 목적 5

1.3 논문의 구성 7

제 2 장 RFID 기술 개요 8

2.1 RFID 시스템의 구성 8

2.1.1 태그 9

2.1.2 리더 18

2.1.3 서버 21

2.2 데이터 암호화 기법 24

2.2.1 DES 알고리즘 24

2.2.2 AES 알고리즘 26

2.2.3 SEED 알고리즘 29

2.3 RFID 인증 프로토콜 34

2.3.1 RFID 공격의 유형 35

2.3.2 RFID 시스템의 보안 요구사항 37

2.3.3 기존의 인증 기법들 38

제 3 장 다중 객체 접근 방식의 태그 및 암호화 메커니즘 47

3.1 다중 객체 접근을 위한 태그 구조 48

3.1.1 태그 구조의 설계 49

3.2 SEED 알고리즘을 변형한 태그 암호화 알고리즘 54

3.2.1 키 크기 변경 알고리즘(K-SEED) 56

3.2.2 라운딩 횟수 조정 알고리즘(R-SEED) 60

제 4 장 다중 객체 지원을 위한 인증 프로토콜 64

4.1 보안 레벨 개념 64

4.2 인증 프로토콜의 설계 65

4.2.1 다중 객체 정보 검색을 위한 인덱스 구조 66

4.2.2 저수준 인증 프로토콜 67

4.2.3 고수준 인증 프로토콜 69
제 5 장 실험 및 분석 73
5.1 실험 방법 및 환경 73
5.2 암호화 알고리즘의 성능 평가 80
5.3 인증 프로토콜의 안전성 평가 82
5.4 인증 프로토콜의 효율성 평가 85
제 6 장 결론 및 향후 연구과제 93
6.1 결론 93
6.2 향후 연구과제 95
참 고 문 헌 97

그림 목차

[그림 2-1] RFID 시스템의 구성 9
[그림 2-2] RFID 태그의 형태 10
[그림 2-3] RFID 태그 식별자의 개념 16
[그림 2-4] RFID 태그 식별자 18
[그림 2-5] RFID 리더와 정보 전송 범위 20
[그림 2-6] RFID 리더의 구성 21
[그림 2-7] RFID 서버와 RFID 시스템 구성 요소간 연결 22
[그림 2-8] DES 알고리즘의 구조 26
[그림 2-9] AES 알고리즘의 암호화 과정 29
[그림 2-10] SEED 알고리즘의 구조 32
[그림 2-11] 패러데이 케이지 기법을 이용한 제품들 40
[그림 2-12] 블로커 태그 기법의 트리 구조 41
[그림 2-13] 해쉬 락 프로토콜 42
[그림 2-14] 랜덤 해쉬 락 프로토콜 43
[그림 2-15] 해쉬 체인 메커니즘 45
[그림 2-16] 해쉬 체인 프로토콜 45
[그림 2-17] 해쉬 기반 ID 변형 프로토콜 46
[그림 3-1] 제안하는 태그 및 시스템 구조 49
[그림 3-2] 서버와 태그의 저장 구조 53
[그림 3-3] K-SEED 알고리즘의 구조 56
[그림 3-4] K-SEED 알고리즘의 입출력 블록 구조 57
[그림 3-5] K-SEED 알고리즘의 F 함수 구조 58
[그림 3-6] K-SEED 알고리즘의 G 함수 구조 59
[그림 3-7] K-SEED 의 키 스케줄 60
[그림 3-8] R-SEED 알고리즘의 구조 62
[그림 3-9] R-SEED 의 키 생성 알고리즘 63

[그림 4-1] 태그에 저장되는 인덱스 구조 67
[그림 4-2] 저수준 보안 레벨에서의 인증 절차 69
[그림 4-3] pID 생성 과정 71
[그림 4-4] 고수준 보안 레벨에서의 인증 절차 72
[그림 5-1] 저수준 인증 프로토콜 순서도 78
[그림 5-2] 고수준 인증 프로토콜 순서도 79
[그림 5-3] 암 · 복호화 성능 비교 82
[그림 5-4] 인증 단계별 연산 시간 90
[그림 5-5] 인증 프로토콜별 태그에서의 평균 에러율 92

표 목차

[표 2-1] RFID 태그의 분류 11
[표 2-2] RFID 태그의 주파수 대역별 특성 12
[표 2-3] RFID 관련 ISO 국제 표준화 그룹 13
[표 2-4] EPCGlobal 의 RFID 태그 분류 14
[표 2-5] AES 알고리즘의 라운드 수 27
[표 2-6] AES 알고리즘의 구성 함수 28
[표 2-7] SEED 관련 표준 제정 현황 30
[표 2-8] SEED 알고리즘의 특징 31
[표 2-9] RFID 보안 서비스의 종류 38
[표 3-1] 서버와 태그에 저장되는 정보 50
[표 5-1] 리더와 태그의 상세 정보 74
[표 5-2] 태그 메모리 블록 구조 76
[표 5-3] 다중 객체 정보 검색을 위한 인덱스 테이블 77
[표 5-4] 암호화와 복호화 처리 속도 81
[표 5-5] 인증 프로토콜의 안전성 비교 85
[표 5-6] 인증 프로토콜의 연산량 비교 87
[표 5-7] 시뮬레이션을 위한 개발 환경 88
[표 5-8] 인증 시간 측정 89
[표 5-9] 인증 프로토콜별 평균 에러율 측정 91

제 1 장 서 론

1.1 연구 배경

RFID(Radio Frequency IDentification)는 자동인식 기술의 한 종류로서, 마이크로 칩을 내장한 태그에 저장된 데이터를 무선주파수를 이용하여 비접촉식으로 읽어 내는 기술이다. 기존의 물류, 유통 분야에서 일반적으로 사용되던 바코드 기술과는 달리 물리적 접촉 없이도 인식이 가능하다는 장점을 비롯하여, 정보의 수정이 가능하고 원거리에서도 적용할 수 있다. 이러한 여러 장점으로 인하여 RFID 기술은 현재 교통카드, 고속도로 통행 시스템, 물류관리 및 출입통제 시스템 등 다양한 분야에서 사용되고 있다. 또한, 최근에는 RFID 태그 하드웨어에 대한 기술 개발과 구축 비용이 하락함에 따라 기업들이 RFID 기술을 새로운 비즈니스에 접목하려는 시도를 하고 있으며, 기존 기업의 응용 시스템들과 결합해 통합 응용됨으로써 그 가치가 배가되고 있는 상황이며, 미래에는 더욱 광범위한 목적으로 활용될 전망이다[Avoi04, Fink99].

최초의 RFID 기술은 2차 대전 중 연합공군에서 사용되었던 “ Identify Friend or Foe” 시스템[Weis03]이나 본격적인 기술 개발은 1970년대 탄도미사일 추적을 목적으로 미국에서 시작되었고, 칩 제조와 무선통신 기술의 발달로 여러 산업 분야에 적용되기 시작했다. 초기에는 비용 문제로 인해 널리 사용되지 못했지만, 1980년대에는 칩 기술의 발달

로 태그의 크기가 작아지고 가격이 낮아지면서 가축관리, 상품의 유통관리 등의 여러 산업분야에 사용되기 시작하였다. 1990년대에 들어오면서부터는 고주파(RF) 기술의 발전에 따라 저가격, 고기술의 태그가 개발되고, 카드나 동전 등의 다양한 형태의 제품도 출현하게 되었다. 2000년대 이후에는 언제 어디서나 간편하게 컴퓨터와 네트워크를 이용할 수 있는 사용자 중심 IT 환경 조성을 목적으로 하는 유비쿼터스(Ubiquitous) 환경이 도래하면서 무선 인식 기술의 중요성이 부각되고 있고, 이러한 컴퓨팅 환경의 핵심 기술 중 하나인 RFID 기술은 더욱 다양한 응용 솔루션으로 개발되고 있다.

RFID 시스템은 전자 태그를 비롯하여 리더와 서버로 구성된다. 태그는 정보 축적과 발신 기능을 가지는 매우 작은 칩으로 해당 상품의 세부 정보를 저장하고 있으며, 고주파 신호를 받으면 내장된 정보를 전송하는 방식으로 리더 및 서버와 조합되어 그 기능을 발휘한다. 리더는 태그를 향해 전파를 송수신하는 전자회로 부분을 가지고 있으며, 리더 내에 있는 마이크로프로세서를 이용하여 태그로부터의 신호를 바꾸거나 검증하면서 자체 메모리에 저장하거나 서버로 송신하는 역할을 수행한다. 또한, 서버에서는 리더로부터 전송되는 정보를 데이터베이스에 저장하거나 검색하고, 연산을 수행함으로써 태그와 리더에 대한 인증 절차를 거쳐 정당한 태그 혹은 리더인지를 판별한다.

현재 RFID 태그에는 하나의 응용 객체에 대한 식별 정보를 저장하고 있기 때문에 RFID 기술을 활용한 응용들이 광범위해짐에 따라 각종 물건들에 점점 더 많은 태그를 부착하게 되는데, 사용자 입장에서 보면 오히려 불편을 초래할 수도 있다. 즉, RFID 기술이 생활 전반에 적용되면 회사의

출입증이나 집 혹은 자동차의 키 등 개인의 생활 환경에 필요한 여러 물건들에 각기 다른 목적으로 태그가 부착되고, 사용자는 여러 개의 태그를 소유하게 되는데, 사용자의 입장에서는 점점 더 많은 태그를 소지하고 이러한 많은 태그들을 용도에 따라 구별하여 사용한다는 것이 불편하고 어려워질 수 있다. 더구나 RFID는 무선 통신을 이용한 기술이므로, 그에 대한 공격이 일반 네트워크 환경에 비해 용이하다. 기존의 바코드 기술에 비해 편리성은 향상되었지만 태그의 정보를 누구든지 항상 읽을 수 있다는 점 때문에 정당하지 않은 사용자에 의한 불법적인 접근이 가능하다. 따라서, 많은 태그들에 저장된 개인 정보는 곳곳에 설치되어 있는 리더를 통해 읽혀지게 되어 사용자의 정보 유출이나 위치 추적과 같은 프라이버시를 침해하는 심각한 문제점을 발생시킬 가능성이 있다[Sarma02, Yeo05].

RFID 시스템에서 발생 가능한 개인 정보 침해에 대한 공격들로는 도청, 위치 추적, 스푸핑(Spoofing), 재전송, 메시지 유실 등이 있으며, 이러한 공격들에 대해 사용자 프라이버시를 보호하기 위한 여러 가지 인증 기법들이 연구되고 있다. Kill 명령어 기법, 패러데이 케이지 기법, 블로커 태그 기법 등의 물리적인 기법들과 함께, 암호학적인 방법으로 해쉬 락 기법, 해쉬 체인 기법, Randomized 해쉬 기법, 해쉬 기반 ID 변형 기법 등이 있다. 기존의 인증 기법들은 하나의 태그를 하나의 특정 응용에 대한 객체로 동작하는 것을 기반으로 하고 있다. 추후 RFID 기술을 이용한 응용 시스템의 보급이 더욱 확대되고 기술의 발전으로 인해 고성능의 태그가 일반화된다면, 다수의 응용 객체에 대한 정보들을 하나의 태그에 저장할 수 있을 것이고, 이러한 경우 기존의 인증 기법들은 적절한 대응을 하지 못하고 또

한 현재 일부 공격들에 대해서도 여전히 약점들을 가지고 있다.

현재 RFID 시스템의 정보보호에 대한 연구는 계속되고 있으며, 많은 정보보호 기술이 개발되고 있다. 이를 해결하기 위한 기반 기술로서 암호화 알고리즘에 대한 기술 개발이 필수적이다. 그러나, RFID 기술의 상용화를 위해서는 태그의 가격이 낮아질 수밖에 없고, 이로 인해 태그의 IC(Integrated Circuit) 또한 적은 수의 게이트를 가질 수밖에 없는 실정이다. 따라서, 저가의 태그에서 기존의 DES, AES, SEED와 같은 암호화 알고리즘들의 사용 가능 여부는 여전히 불투명하다. 이를 해결하기 위해서는 낮은 가격에 좋은 성능을 가지는 IC를 개발, 생산하는 것도 중요하지만, 자원의 소모가 적으면서도 안전한 암호 알고리즘의 개발과 아울러 최소의 자원을 사용하면서도 안전한 프로토콜의 개발도 필수적이다[유성호04, Sarma01]. 결론적으로, 안전한 RFID 시스템 구축을 통한 유비쿼터스 환경을 이루기 위해서는 암호화 기술의 개발과 함께 RFID 시스템을 이용한 서비스의 보안 요구사항 수립과 인증 프로토콜의 개발이 동시에 진행되어야 할 것이다.

1.2 연구 목적

본 논문에서는 하나의 태그를 이용하여 다수의 RFID 응용 객체에 접근하기 위한 다중 객체 접근 방식의 RFID 시스템을 제안한다. 이러한 RFID 시스템은 다중 객체를 지원하는 태그 구조와, 객체의 식별 정보를 보호하기 위한 암·복호화 알고리즘 및 다양한 공격들로부터 사용자 프라이버시를 보호하기 위한 인증 프로토콜을 포함하며, 다음과 같은 요구사항을 만족시킨다.

첫째, RFID 기술이 광범위해짐에 따라 여러 가지 응용 객체들이 각각의 목적에 따라 많은 태그들을 필요로 하게 된다. 정보의 통합과 사용자의 이용 편의성을 위해서는 하나의 태그로 여러 응용 객체에 접근할 수 있는 구조를 가진 RFID 시스템이 요구된다.

둘째, RFID 태그 내에 저장되는 식별정보는 정당하지 않은 사용자의 여러 가지 공격들로 인해 침해의 소지가 있다. 따라서, 태그의 식별정보를 암호화하기 위한 알고리즘이 요구된다.

셋째, 무선통신을 이용하는 RFID 시스템의 특성상, 사용자의 정보보호와 올바른 동작을 위해 리더와 태그, 그리고 서버간의 인증 프로토콜이 요구된다.

본 논문에서는 다중 응용 객체에 접근할 수 있는 태그의 구조를 제안하고, 태그의 식별정보를 암호화하기 위해 기존의 SEED 알고리즘을 RFID 응용 환경에 적합하도록 수정하여 암호화 속도면에서의 성능을 향상

시킨 SEED 변형 알고리즘을 설계하였다. 그리고, RFID 시스템의 구성 요소들간의 인증을 위한 프로토콜은 저사양의 태그에서도 쉽게 구현이 가능하도록 연산량을 최소화하면서도 리더와의 통신 과정에서 발생 가능한 다양한 프라이버시 공격들에 대해 안전성을 보장하도록 설계하였다. 또한 인증 절차를 효율적으로 수행하기 위해 서로 다른 여러 응용 객체의 보안 요구 수준을 정의하였고, 이에 따라 인증 절차를 차별화하여 동작하도록 하였다.

1.3 논문의 구성

본 논문의 구성은 다음과 같다.

제 2 장에서는 RFID 기술을 소개하고, RFID 시스템의 구성 요소와 암호화 기법 및 RFID 시스템의 보안 요구사항에 대해 설명한다. 그리고 RFID 시스템의 각 구성 요소들 간에 이루어지는 인증 프로토콜에 대해 설명하고, 기존의 인증 프로토콜들에 대해 인증 절차와 특징을 살펴본다.

제 3 장에서는 다중 객체 접근 방식의 RFID 태그 구조와 암호화 메커니즘에 대해 소개한다. 다중 객체를 지원하기 위한 태그는 하나의 태그 내에 다수 개의 식별 정보를 포함하는 구조이며, 태그 내에 저장되는 식별 정보를 암호화하기 위한 암호화 알고리즘을 제안한다.

제 4 장에서는 다중 객체를 지원하는 태그 구조에 적합한 인증 프로토콜에 대해 소개한다. 제안하는 인증 프로토콜은 RFID 응용의 종류에 따라 달라지는 보안의 수준을 구분하여 동작하도록 설계하였으며, 이에 대해 자세히 설명한다.

제 5 장에서는 제안 시스템의 성능을 평가하기 위하여 암호화 알고리즘의 암·복호화 속도, 인증 프로토콜의 연산량, 공격으로부터의 안전성 및 인증 절차에 소요되는 시간과 에러율에 대해 기존의 기법들과 비교 분석한다.

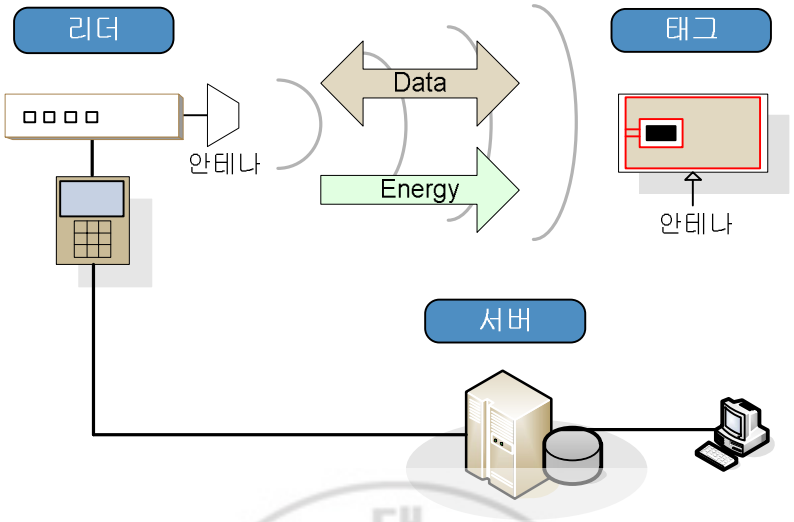
마지막으로 제 6 장에서는 본 논문의 연구 결과를 정리하고, 향후 추가 연구 분야에 대해 기술한다.

제 2 장 RFID 기술 개요

본 장에서는 논문에서 제안하는 다중 객체 접근 방식의 RFID 시스템을 설명하기에 앞서 이와 관련된 연구를 설명한다. 우선, 연구의 기반이 되는 RFID 시스템의 구성 요소들에 대하여 설명한다. 그리고 데이터의 안전성을 보장하기 위한 기존의 암호화 기법들에 대하여 고찰한다. 다음으로는 RFID 시스템이 동작하기 위해 각 구성 요소들 간에 이루어지는 인증 프로토콜에 대해 설명하고, 기존의 RFID 인증 프로토콜들의 특징 및 장단점을 분석한다.

2.1 RFID 시스템의 구성

RFID 시스템은 무선 주파수를 이용한 비접촉식 자동 인식 기술로서 태그와 리더 및 서버로 구성된다. 태그에는 식별 정보가 저장되며, 안테나가 부착된 리더에서는 무선 통신에 의해 접촉하지 않고 태그의 정보를 판독하거나 기록할 수 있다. 서버의 기능은 리더에 의해 수집된 정보를 해독하기 위한 연산을 수행하고, 수집한 정보를 데이터베이스에 저장한다. [그림 2-1]은 이러한 RFID 시스템을 이루고 있는 구성 요소들과 각 요소들 간의 연결 상태를 나타낸 것으로서, 이러한 RFID 시스템의 각 구성 요소들에 대해 다음과 같이 상세히 살펴보기로 한다.

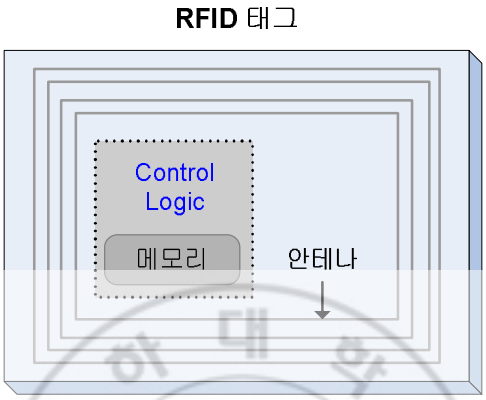


[그림 2-1] RFID 시스템의 구성

2.1.1 태그

RFID 시스템의 구성 요소들 중에서도 RFID의 핵심 기능이라 할 수 있는 태그는 정보 저장과 발신 기능을 가지는 매우 작은 칩으로 해당 상품의 세부 정보(ID)를 담고 있으며, 고주파 신호를 받으면 내장된 정보를 전송하는 방식으로 동작한다. [그림 2-2]와 같이 태그의 칩에는 Control Logic과 메모리가 포함되는데, Read-only 칩의 경우 mask ROM이 사용되며, Read/Write 칩은 EEPROM 또는 flash 메모리를 구성하여 태그를 제작한다. 태그에는 이러한 정보를 저장하기 위한 칩과 연결된 안테나, 그리고 전파동조를 위한 콘덴서가 내장되어 있다. 최근의 기술 추세는 단순한 인식소자 이상의 데이터 저장 기능을 갖는 태그가 본격적으로 개발되고 있

다[변상기04].



[그림 2-2] RFID 태그의 형태

가. 태그의 분류

태그는 배터리의 내장 여부와 저장 기능 및 주파수 대역에 따라 [표 2-1]과 같이 분류될 수 있다. 전력을 공급받는 방법에 따라 분류된 능동형(Active) 태그는 내장된 자체 배터리로부터 전원을 공급받으며 원거리 정보 전송이 가능하나, 태그의 가격이 높고 배터리의 수명이 태그의 수명을 좌우한다는 단점이 있다. 반면 수동형(Passive) 태그는 자체 배터리를 내장하고 있지 않아 리더로부터 수신한 전자기파에 의한 유도 전류를 전원으로 사용하며, 근거리 정보 전송에 주로 이용된다. 따라서, 태그의 가격이 저렴하며, 수명이 반영구적이라는 장점이 있어서 현재 RFID 시스템에서 널리 사용되고 있다[김상태03]. 그러나, 수동형 태그

는 연산 능력과 통신 능력의 제한으로 인해 통신 과정에서의 정보 보호 능력이 낮은 단점이 있다.

[표 2-1] RFID 태그의 분류

분류 기준	종류
전원	Active Passive
저장 기능	Read-only WORM(Write Once Read Many) Read/Write
사용 주파수	125~135kHz 13.56MHz 433.92MHz 860~960MHz 2.4GHz

태그의 데이터 저장 기능에 따라서는 Read-only, WORM, Read/Write 태그로 구분할 수 있는데, 태그는 단순한 인식소자 기능에서 데이터의 읽기·쓰기가 가능한 태그로 발전하는 추세이다. 현재 개발되고 있는 태그는 약 10Mbps의 데이터 전송 속도와 1Mbyte 이상의 메모리 용량을 실현할 수 있는 수준으로서, 초소형 사이즈의 칩에 고용량 정보 저장 수준의 메모리 구조를 채용한 태그를 활용하고 있으며, 향후에는 프로세서를 포함한 태그로까지 발전 가능할 것이라 예측되고 있다[Sarma03].

RFID 태그의 사용 주파수 대역은 저주파, 고주파, 극초단파 및 마이크로파로 분류할 수 있으며, 주파수 대역별로 태그의 특성과 동작 및 적용

분야를 비교하면 [표 2-2]와 같다[표철식04].

[표 2-2] RFID 태그의 주파수 대역별 특성

주파수 (Hz)	저주파(LF)	고주파(HF)	극초단파(UHF)		마이크로파
	125/135K	13.56M	433.92M	860~960M	2.45G
인식 거리	60cm 이하	60cm	50~100m	3.5~10m	1m
일반 특성	환경에 의한 성능 저하 없음	다중태그 인식	실시간 추적, 환경 센싱	다중태그 인식, 성능 우수	환경에 영향 받음
동작 방식	수동형	수동형	능동형	능동/수동형	능동/수동형
적용 분야	공정자동화, 출입/보안, 동물관리	수하물 관리, 교통카드, 출입/보안	수하물관리, 실시간 위치 추적	공급망 관리, 자동통행시스템	위조방지
인식 속도	저속 ←----------→ 고속				
환경 영향	강인 ←----------→ 민감				
태그 크기	대형 ←----------→ 소형				

시스템 가격이 저렴한 135kHz 이하의 RFID 태그는 주로 가축식별이나 공정관리, 재고관리 및 차량 잠금장치 등 짧은 인식거리를 가지는 분야에 활용되고 있다. 그리고, 13.56MHz 주파수 대역을 사용하는 RFID 태그는 1m 이내의 인식거리를 가질 수 있으며, 데이터의 전송 신뢰도가 높은 편이라 교통카드나 도서관리 시스템 등에 적용되고 있다. 433.92MHz 주파수 대역의 RFID 시스템은 50~100m의 비교적 긴 인식거리를 가지며,

실시간 추적 및 컨테이너 내부 습도 등의 환경을 감지할 수 있는 특성을 가진다. 그리고, 860~960MHz 주파수를 사용하는 RFID 태그는 인식 거리와 속도 및 성능 면에서 가장 우수하며, 다중 태그를 인식할 수 있다. 또한, 2.45GHz 주파수 대역의 태그는 시스템 가격이 높은 편이며 차폐물이 있는 경우에는 인식이 불가능하다는 단점이 있으나, 위조방지 시스템 및 톨게이트 자동화 분야 등에 적용되고 있다.

나. 태그의 표준화

RFID 태그의 표준화는 ISO와 IEC가 공동 설립한 JTC1과 GSI 기구에서 설립한 EPCGlobal Inc.[1]에서 이루어지고 있다. JTC1에서는 세부적으로 4개의 SG(Sub Group)으로 나뉘어져 분야별로 표준화를 추진하고 있는데, 다음의 [표 2-3]은 각 SG가 담당하는 표준화에 대한 현황을 나타낸다.

[표 2-3] RFID 관련 ISO 국제 표준화 그룹

분류	그룹명	분야
SG1	Data 구문 표준	데이터 프로토콜 표준화
SG2	Tag 식별	태그의 식별을 위한 번호부여 방법에 대한 표준화
SG3	Air Interface	태그·리더 간 주파수 대역별 통신 규약에 대한 표준화
APR	적용 요건/기술	RFID 활용을 위한 요구사항 정의

[1] MIT 의 Auto-Id Center 가 2003 년 EPCGlobal Inc.과 Auto-ID Labs 으로 분리

또한, MIT의 Auto-ID 센터를 전신으로 하는 EPCGlobal Inc. 에서는 [표 2-4]와 같이 태그를 분류하고 있다[EPC03]. Class0 형태의 RFID 태그는 제조사에 의해 식별 정보를 태그에 저장하는 반면, Class1 태그는 사용자에 의해 정보를 입력할 수 있다. Class2 이상의 RFID 태그는 읽기·쓰기가 가능한 메모리를 포함하고 있으며, 이 태그 메모리에는 리더와 태그 간의 통신 과정에서 필요한 정보들을 저장할 수 있다.

[표 2-4] EPCGlobal 의 RFID 태그 분류

분류	저장 기능	전원	특징
Class0	Read-only (64bit[1])	Passive	수명이 길고, 도달거리가 짧다.
Class1	WORM (96bit)		
Class2	Read/Write (128/256bit)	semi-Active	수명이 짧고, 도달거리가 길다.
Class3	Read/Write	Active	센서 태그
Class4			다른 태그와 통신
Class5			네트워크 구성 가능

국내에서도 정부와 산업계 주도로 RFID 육성 방안을 마련하고 있다. 정통부는 900MHz 대역 등 신규 주파수 확보와 기술 기준 제정 연구 개발, 응용 표준화, 테스트 베드 구축 지원 RFID 센터 설립과 산업협의회 구성 등의 세부 실행 방안을 마련하여 RFID를 유비쿼터스 컴퓨팅 인프라로 적

[1] 메모리 블록의 크기

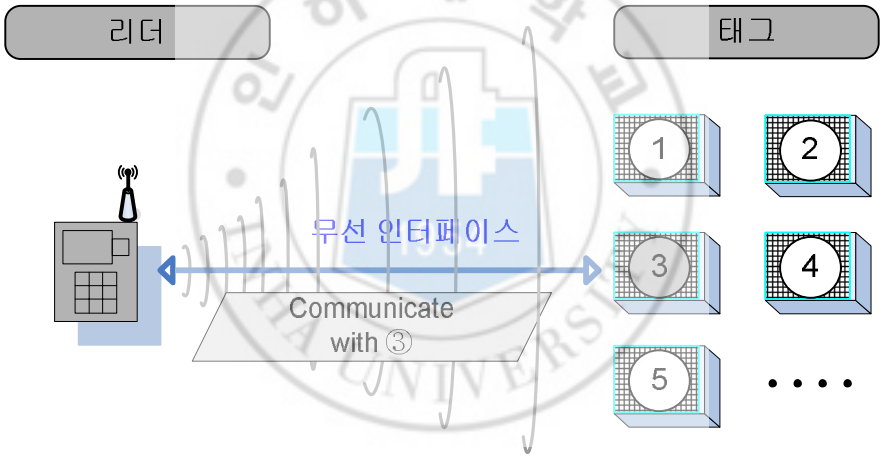
극 육성키로 했다[윤현철05]. 또한, 2004년도에는 RFID/USN 협회가 결성되어 국내 RFID/USN 산업 발전 및 보급을 위해 표준화 및 인증, 기술 개발 및 사업 적용, 각종 대회홍보 지원 등의 업무를 담당하고 있다. RFID/USN 협회는 표준화 분과, 하드웨어 분과, 네트워크 분과, 응용 분과, 정책/홍보 분과의 5개 분과로 나누어져 있다.

한편, 모바일 환경에서 RFID 기술을 적용함에 있어 RFID 단말기 규격 및 메시지 전송 프로토콜 규격, RFID 단말기 제어를 위한 WIPI 확장규격, 응용 서비스 규격 등에 대한 표준화 작업을 담당하기 위한 모바일 RFID 포럼도 구성되어 있다. 모바일 RFID 시스템에서는 보안 기술이 매우 중요한 항목이며, 특히 구매나 금융 서비스와 관련된 모바일 RFID 응용의 경우에는 사용자의 프라이버시 정보와 결합된 정보가 불법으로 유출될 수 있으므로 모바일 RFID 보안 기술이 가장 큰 이슈가 될 것으로 보인다[김말희06].

다. 태그의 식별 정보

RFID 태그는 데이터를 저장할 수 있는 논리적인 메모리를 가지고 있으며, 전파 신호를 이용하여 RFID 리더와 데이터를 주고 받는다. RFID 리더는 태그와 데이터를 주고 받는 동안, 정확한 태그와 통신을 하고 있는지 확인하기 위한 수단으로서 해당 태그의 고유한 식별 번호를 필요로 한다. 이러한 태그 식별을 위한 고유 번호를 마련하기 위한 표준화된 번호화 방안[오세원05]은 개별 RFID 태그 각각을 고유하게 구별할 수 있는 RFID 태그 식별자를 규정하고 있다.

RFID 태그 식별자는 무선 인터페이스를 통해 특정 RFID 태그를 고유하게 구별하기 위한 식별자이다. 다음의 [그림 2-3]에서는 RFID 리더가 인식 범위 내의 ①~⑤의 태그들 중에서 특정 태그 ③을 구별하는 개념을 보여준다. 즉, 리더는 태그 식별자가 ③인 태그를 구별하여 통신을 수행함으로써 통신 자원 및 전력의 소모를 최소화할 수 있다. 또한, 특정 태그를 유일하게 구별할 수 있으므로, 특정 태그의 이력 관리 및 추적을 수행할 수 있는 수단으로 사용 가능하다.



[그림 2-3] RFID 태그 식별자의 개념

이와 같이 각각의 태그를 구별하기 위한 RFID 태그 식별자는 다음과 같이 불변 유일 식별자와 사물 식별자, 가상 식별자로 구분할 수 있으며, 이 세 가지 식별자의 하나 또는 그 이상의 번호 체계를 선택 및 조합하여 만들어질 수 있다.

■ 불변 유일 식별자 (Permanent Unique Identifier)

태그의 IC 생산자 또는 태그 제조자에 의해 태그 내에 고정 기록되는 식별 번호로서, 칩 ID(Chip Identifier) 또는 태그 ID(Tag Identifier)라고도 한다. 즉, 태그의 IC 제조 시점이나 태그 제품의 제조 시점에서 결정되는 고유한 식별 번호로서, 태그가 활용되는 응용 환경이나 시간 및 공간에 구애받지 않고 해당 태그를 고유하게 식별할 수 있는 수단이 된다. 이는 국제 표준인 ISO/IEC 15963에 명시된 규정 사항을 준수한 번호 형식을 갖추어야 한다.

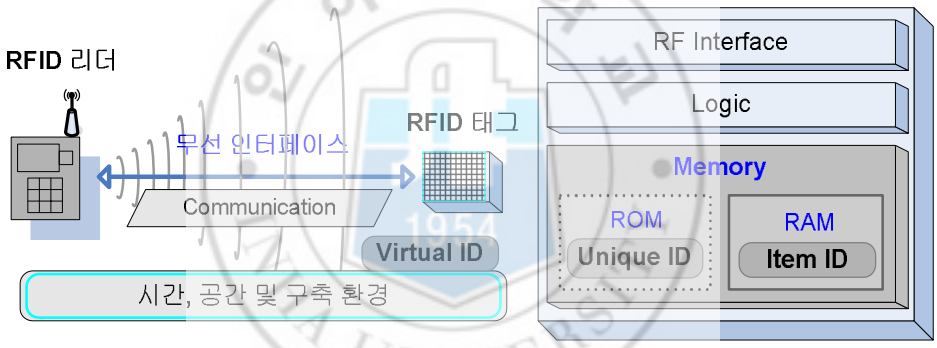
■ 사물 식별자 (Item Identifier)

사물 식별자는 태그를 부착한 사물 자체를 응용 분야별로 구별하기 위한 식별자로서 태그 사용자의 결정에 따라 태그의 데이터 기록 공간에 저장되는 정보이다. 태그의 이용 시점에서 사물에 대한 구별을 위해 기록되므로, 사물 ID 또는 아이템 ID로 불리며, 태그 이용자가 자율적으로 정한 번호 체계를 사용할 수도 있다. 그러나, 여러 기관 간의 협업 환경이나 정보의 원활한 교환과 공유를 위해 사물 식별을 위한 공식적인 표준 번호 체계의 사용이 권장되며, 현재 ISO/IEC JTC1 SG31과 EPCGlobal Inc. 등 국제 표준화 단체에서 사물 식별자의 표준화 작업이 활발히 진행되고 있다. 본 논문에서 제안하는 RFID 태그는 사물 식별자를 이용하여 다중 객체에 접근할 수 있는 구조로 설계된다.

■ 가상 식별자 (Virtual Identifier)

가상 식별자는 특정 무선 인터페이스 및 통신 세션, 시·공간 환경,

RF 리더 제품 특성에 따라 리더가 태그에게 임시적으로 부여하는 식별 번호로서, 임시 식별자(Temporary Identifier)라 불리기도 한다. 즉, 리더가 특정 태그와 데이터를 주고 받는 동안에만 임시적으로 부여되는 번호이므로 동일한 태그에 대해서도 시간이나 공간, 무선 인터페이스 특성에 따라 다른 번호가 부여될 수 있고, 식별자의 유일성을 보장할 수도 없다. [그림 2-4]는 태그의 구조와 태그를 고유하게 구별할 수 있는 식별자 정보를 태그 메모리에 저장하는 논리적인 메모리 구조를 보여준다.



[그림 2-4] RFID 태그 식별자

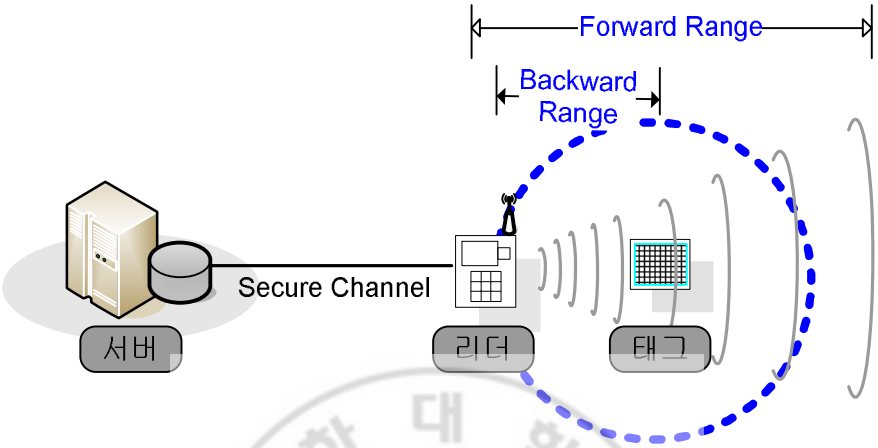
2.1.2 리더

RFID 시스템에서 리더는 무선 채널을 통하여 각각의 태그들과 통신을 하는데, 태그들은 리더가 보낸 신호를 듣고 리더의 전송 요구에 응답을 한다. 이 때, 리더의 전송 요구에 대해 무선 채널의 전송 반경 내에 있는

모든 태그들은 동시에 신호를 듣게 되고 리더의 전송 요구에 응답을 하게 되므로 태그 충돌(Tag Collision) 문제가 발생한다. 따라서, 리더는 동시에 응답한 여러 개의 태그를 구별할 수 있도록 충돌 방지(Anti-collision) 알고리즘를 탑재하여 태그를 인식해야 하며, 이는 RFID 리더의 핵심 기술이다[Her01, 이근호03, 최호승05]. 특히, 수동형 태그 시스템의 리더는 RF 신호를 통해 태그에게 전원을 공급하는 역할을 한다.

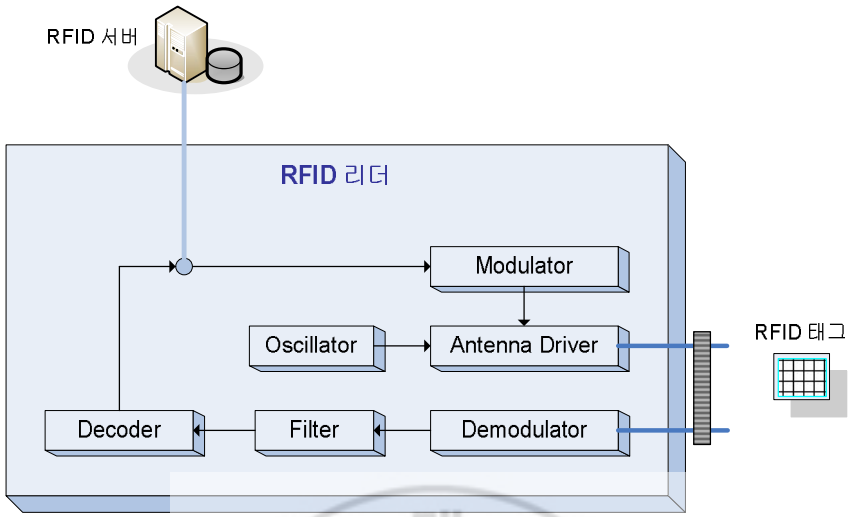
또한, 리더는 태그로부터 수신한 데이터를 서버의 데이터베이스로 전송하는 기능을 한다. 일반적으로 리더는 RFID 시스템 서버에 연결되어 운용되며, 응용 목적에 따라 운용 소프트웨어에 의해 RFID 시스템을 제어한다.

[그림 2-5]에서 전방위 영역(Forward Range)은 리더가 RF 신호를 태그로 전송할 수 있는 범위를 나타내며, 후방위 영역(Backward Range)은 태그가 리더의 요청에 대하여 자신의 정보를 전송할 수 있는 범위를 의미한다. 예를 들어, 915MHz의 주파수를 이용하는 RFID 시스템의 경우, 태그의 전송 반경은 약 3m 정도이며 리더가 전송할 수 있는 영역은 반경 약 100m 정도라 할 수 있다[양형규05]. 태그가 배터리를 내장하지 않는 경우, 리더는 태그에게 에너지를 전달해야 하는 필요성에 의하여 필드의 범위가 넓어질 수밖에 없고, 태그는 리더에게 전달받은 에너지를 이용하여 리더에게 응답하기 때문에 범위가 작을 수 밖에 없다. RFID 시스템에서는 기본적으로 리더와 태그 사이의 통신이 불안전한 채널에서 이루어진다고 가정하며, 상대적으로 서버와 리더 간의 통신 경로는 안전하다고 가정할 수 있다[Aign05, Goll04].



[그림 2-5] RFID 리더와 정보 전송 범위

RFID 리더의 형태는 고정형, 이동형, PC 카드형 등 다양하며, [그림 2-6]과 같이 안테나 및 RF 회로, 변복조기, 실시간 신호처리 모듈 등으로 구성된다. RFID 리더는 안테나의 성능 및 주변의 환경에 의해 인식 거리, 검출 정확도가 영향을 받아 적용 범위가 제한되는 특성이 있다. 인식 성능을 높이기 위해 2~4개의 안테나를 사용하고 있으나, 향후 주변 환경에 적응하여 빔을 제어할 수 있는 빔 성형 안테나 기술이 개발될 전망이다. 또한, 여러 주파수 대역의 RFID 시스템이 혼합하여 사용될 수 있으므로 다중 대역 RF 안테나가 필요하게 될 것이고, 향후 다양한 정보기기와 리더가 통합되는 방향으로 발전하리라 예상된다[변영기06].



[그림 2-6] RFID 리더의 구성

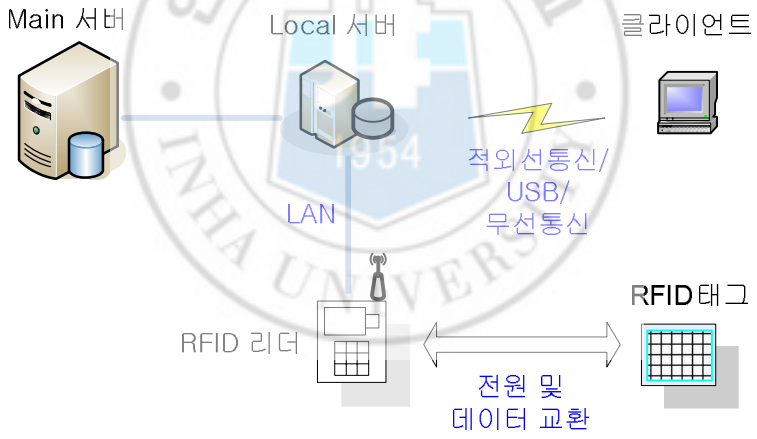
현재 RFID 프로토콜에 대한 표준이 통일되지 않아 여러 가지 프로토콜이 사용되고 있어 멀티 프토코콜 리더가 요구되고 있는 상황이며, 이러한 기능의 리더를 구현하기 위해 디지털 RF 및 SDR(Software Defined Radio) 기술이 적용되어 지능형 리더가 출현될 것으로 예상된다[표철식 04]. RFID 리더는 궁극적으로 소형화되어짐과 동시에 모든 정보 기기에 내장되어 다양한 정보를 수집하는 수단이 되고, 여러 가지 통신 서비스와 연계되어 부가적인 서비스를 창출할 것으로 전망된다.

2.1.3 서버

RFID 시스템의 서버는 [그림 2-7]과 같이 리더와 연결되어 리더에서 수신하는 태그의 식별 정보 및 여러 데이터들을 수집하고 제어, 관리하는 기능을 수행한다. 또한, RFID 시스템의 다른 구성 요소들과 분산된 구

조의 네트워크를 구성하여 서로 통신할 수 있다. 서버는 다양한 형태의 RFID 리더와의 인터페이스 및 다양한 코드 및 망 연동, 그리고 여러 응용 플랫폼에 대해 상호 운용성을 보장할 수 있어야 한다.

서버는 리더로부터 수집한 정보를 데이터베이스에 저장하여 관리할 뿐만 아니라, 연산 능력이 낮은 태그 또는 리더를 대신하여 복잡한 연산을 대신 수행하기도 한다. 또한, 데이터베이스에 저장된 태그의 식별 정보를 이용하여 리더를 통해 태그로부터 수집한 정보의 정당성을 판별하는 역할을 수행한다.



[그림 2-7] RFID 서버와 RFID 시스템 구성 요소간 연결

그러나, 서버와 리더 사이의 통신은 대부분 무선 환경으로 구축될 가능성이 크다. 특히, 대형 쇼핑몰이나 할인점 등의 계산대에 적용되는 리더를 제외한 다수의 리더는 이동성 있는 PDA 타입으로서, 이동성을 지원하

기 위해서는 무선 통신의 적용이 필수적이다. 서버와 리더간에 사용 가능한 대표적인 무선통신 방식으로는 IEEE 802.11, IEEE 802.15.3, IEEE 802.15.4 등의 기술이 있다.

2.2 데이터 암호화 기법

RFID 시스템을 이용한 응용 분야에서는 다양한 보안 요구 사항이 발생하게 된다. 따라서 각 응용 분야에 따라 데이터 암호화의 수준이 다양하게 나타날 수 있다. 비록 각 응용 분야별로 암호화의 수준에 차이는 있지만, RFID 시스템에서의 암호화 동작은 RFID 시스템을 구성하는 서버와 리더 및 태그 사이의 정보 전송 과정에서 발생할 수 있는 여러 가지 형태의 공격들로부터 전송되는 정보를 보호하기 위해 반드시 필요한 과정이다.

암호 알고리즘은 암·복호화에 사용되는 키의 특성에 따라 암·복호화 키가 같은 대칭키 암호알고리즘과 암·복호화 키가 다른 공개키 암호알고리즘으로 크게 구분할 수 있으며, 대칭키 암호알고리즘은 메시지 처리 형식에 따라 스트림 암호알고리즘과 블록 암호알고리즘으로 나누어 볼 수 있다. 이 중에서도 블록 암호알고리즘은 고정된 크기의 입력 블록을 고정된 크기의 출력으로 변형하는 알고리즘에 의해 암호화 및 복호화 과정을 수행하는 방식이다. 따라서, RFID 태그 식별 정보와 같이 고정된 크기의 값을 암호화하기 위한 방식으로는 블록 암호알고리즘이 가장 적합하다고 할 수 있다. 따라서, 기존의 대표적인 블록 암호알고리즘인 DES, AES 및 SEED 알고리즘의 특징과 구조에 대해 알아보기로 한다.

2.2.1 DES 알고리즘

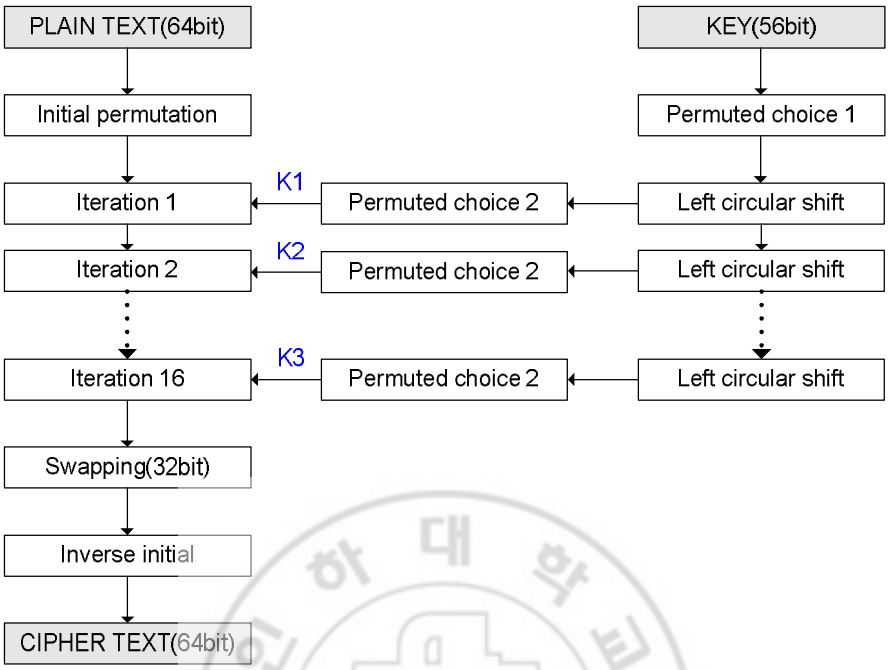
DES(Data Encryption Standard) 알고리즘은 1977년 미국

NBS(National Bureau of Standards)[1]에서 채택된 표준으로서, 컴퓨터와 통신 정보를 보호할 목적으로 IBM이 개발한 알고리즘이다[Juel03a].

DES 알고리즘은 56비트 비밀키 하에서 변환(Permutation)과 치환(Substitution)을 사용하여 64비트의 입력 블록을 수행하는 블록 암호알고리즘이다. 이 알고리즘은 미국 연방 정부의 데이터 보호용으로 출발하여 ANSI(American National Standards Institute)의 표준 암호알고리즘으로 사용되었으며, ABA(American Bankers' Association)에서는 미국 내 금융 정보의 보호 표준으로 사용하기에 이르러 그 사용 범위가 널리 확산되었다.

DES 알고리즘에서는 64비트의 평문을 64비트의 암호문으로 만드는데, 64비트의 키를 사용한다. 이 중 56비트는 비밀키가 되고, 나머지 8비트는 검사용 비트로 사용된다. 또한, 안전성을 증가시키기 위해 키의 길이를 두 배인 128비트를 키로 하는 변형된 알고리즘도 있다. DES 알고리즘은 16라운드의 반복적인 암호화 과정을 가지고 있으며, 각 라운드마다 변환 및 치환의 과정을 거친 평문과 56비트의 비밀키에서 나온 48비트의 키가 섞여 암호문을 만든다. 복호화는 암호화 과정과 동일하나, 사용하는 키만 역순으로 작용한다. DES 알고리즘에서 수행되는 유일한 산술은 비트 문자열의 XOR(Exclusive-OR)이기 때문에, 하드웨어적으로 또는 소프트웨어적으로 매우 효율적으로 수행할 수 있다. 다음의 [그림 2-8]은 DES 알고리즘의 기본 구조를 나타낸 것이다.

[1] NBS 은 현재 NIST(National Institute of Standard and Technology)로 변경되었다.



[그림 2-8] DES 알고리즘의 구조

2.2.2 AES 알고리즘

컴퓨터 시스템의 발달에 따른 계산능력 향상으로 DES 알고리즘의 안전성을 보장받을 수 없게 되자, NIST(National Institute of Standard and Technology)에서는 1997년에 이를 대신할 차세대 블록 암호알고리즘(Advanced Encryption Standard, AES)[Elbi01]을 공모하였고, Rijmen과 Daemen이 만든 Rijndael 알고리즘을 선정하게 되었다[KISA01]. 많은 후보들 중 AES로 선정된 Rijndael 알고리즘은 안전성, 성능, 구현의 간단함 그리고 유연성의 결합이 장점이다. 특히, 이 알고리즘은 광범위한 컴퓨팅

환경에서 하드웨어와 소프트웨어에서 일정하게 매우 좋은 성능을 보인다. 또한, 키 설정 시간의 우수성과 낮은 메모리 요구는 제한된 환경에서 매우 잘 적응할 수 있기 때문에, 메모리 용량이 극히 제한적인 RFID 태그에 AES 알고리즘을 구현하기 위한 연구가 시도되기도 하였다[Feld04b].

AES 알고리즘은 가변 블록 길이와 가변 키 길이를 갖는 반복 구조의 블록 암호 방식이며, 128/192/256비트 크기의 블록과 키를 독립적으로 지정할 수 있다[최병윤01]. 라운드 수는 블록과 키 크기에 따라 다음의 [표 2-5] 와 같이 결정된다.

[표 2-5] AES 알고리즘의 라운드 수

Nr[1]	Nb = 4	Nb = 6	Nb = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

AES 알고리즘의 암호화 과정은 BS(ByteSubstitution), SR(ShiftRow), MC(MixColumn), ARK(AddRoundKey) 함수들의 연산으로 구성된다. 최종 라운드를 제외한 (Nr-1)번의 각 라운드에서 [표 2-6]의 각 함수들이 동작함으로써, 외부에서 입력되는 암호키를 확장하여 각 라운드에 필요한 라운

[1] Nr 은 라운드 수, Nk 는 키 길이, Nb 는 블록 길이이며, Nk 와 Nb 의 4, 6, 8 값은 각각 128bit, 192bit, 256bit 를 나타낸다.

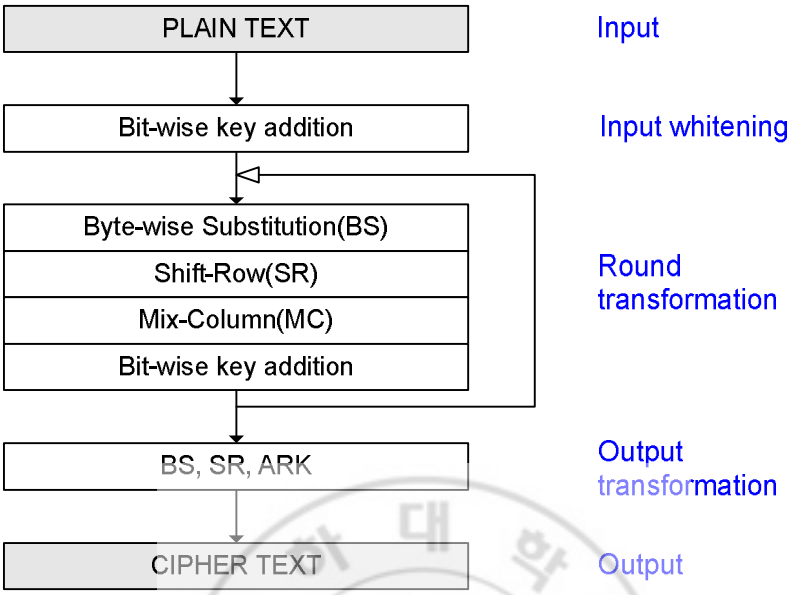
드 키를 생성하게 된다.

[표 2-6] AES 알고리즘의 구성 함수

이름	동작
ByteSubstitution	데이터의 각 바이트 단위로 치환을 수행
ShiftRow	각 행에 대하여 바이트 단위로 우측으로 순환
MixColumn	열에 적용되는 선형 변환으로 행렬 곱셈을 연산
AddRoundKey	각 라운드 키를 XOR 연산

(Nr-1)번의 라운드 연산을 수행하고, MixColumn 함수를 제외한 마지막 라운드의 모든 연산을 수행하고 나면 암호화된 데이터를 출력하게 된다. AES 알고리즘은 복호화 연산 시 암호화 연산의 역 연산을 사용하는 non-Feistel 구조를 바탕으로 하고 있다. 따라서, AES 알고리즘의 복호화 과정은 각 변환 연산의 역변환 연산으로 대체되어 암호화 과정의 역순으로 수행된다.

AES 알고리즘의 암호화 과정을 [그림 2-9]와 같이 나타낼 수 있다. [그림 2-9]에서는 (Nr-1)번의 반복되는 라운드에서의 함수 연산에 대해서는 Round transformation 과정으로, 그리고 마지막 라운드에서의 함수 연산은 Output transformation 과정으로 표시된다.



[그림 2-9] AES 알고리즘의 암호화 과정

2.2.3 SEED 알고리즘

1999년 2월 한국정보보호센터에서 개발한 SEED 알고리즘은 전자상거래, 금융, 무선 통신 등의 분야에서 전송되는 개인 정보와 같은 중요한 정보를 보호하기 위해 한국정보보호진흥원과 국내 암호전문가들이 순수 국내기술로 개발한 블록 암호알고리즘이다[KISA99]. SEED는 1999년 9월 정보통신단체표준(TTA)으로 제정되었으며, 2005년에는 국제 표준화 기구인 ISO/IEC 국제 블록 암호알고리즘 표준으로 제정되었다. 또한, 같은 해 IETF 표준으로도 제정되었고, SEED 알고리즘 자체에 대한 표준 외에도 [표 2-7]과 같이 SEED를 사용하기 위한 다양한 국내/외 표준들이 제정되었다.

[표 2-7] SEED 관련 표준 제정 현황

분류	표준명	내용
국내	TTAS.KO-12.0004	128 비트 블록 암호알고리즘(SEED)
	TTAS.KO-12.0025	블록 암호알고리즘 SEED 의 운영모드
국제	ISO/IEC 18033-3	Information technology - Security techniques - Encryption - Part 3 : Block ciphers
	IETF RFC 4269	The SEED Encryption Algorithm
	IETF RFC 4010	보안전자우편에서의 메시지 암호화를 위한 SEED 사용표준
	IETF RFC 4162	TLS 를 위한 SEED 알고리즘 사용표준
	IETF RFC 4196	IPsec 을 위한 SEED 알고리즘 사용표준

SEED는 대칭키 블록 암호알고리즘으로서, 이 알고리즘은 블록의 크기가 n개의 비트로 고정된 평문을 같은 길이의 n비트 암호문으로 바꾸는 함수를 말하며, 이러한 변형 과정에 암·복호키가 작용하여 암호화와 복호화를 수행하게 된다. 블록 단위로 메시지를 처리하는 대부분의 블록 암호 알고리즘은 Feistel 구조로 설계된다. Feistel 구조란 각각이 $n/2$비트인 L_0, R_0 블록으로 이루어진 n비트 평문 블록 (L_0, R_0)이 r라운드($r \geq 1$)를 거쳐 암호문 블록(L_r, R_r)으로 변환되는 반복 구조를 말한다. 반복 구조란 평문 블록이 몇 번의 라운드를 거쳐 암호화를 수행하는 것을 말하고, 라운드 i($1 \leq i \leq r$)란 암호키 K로부터 유도된 각 서브키 K_i(또는 라운드 키라 불림)를 입력으로 (L_{i-1}, R_{i-1})->(L_i, R_i)로 바꾸어 주는 함수이다. 또한, 전체 알

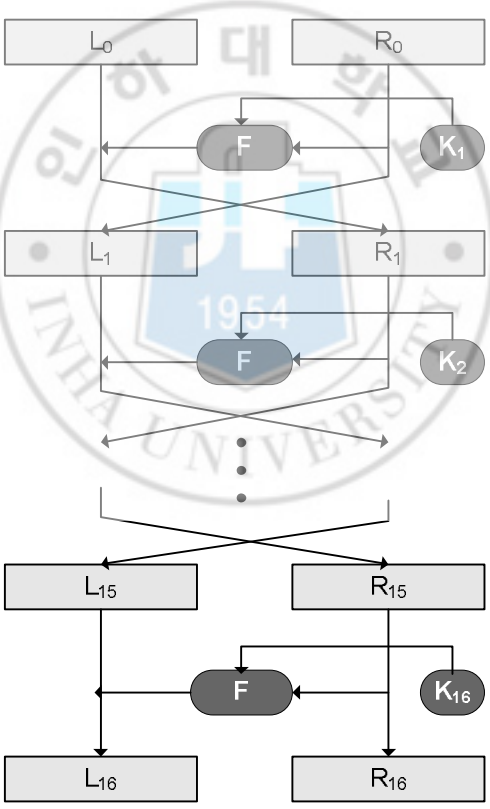
고리즘의 라운드 수는 요구되는 보안 강도와 수행 효율성의 상호 절충적 관계에서 결정된다. 보통 Feistel 구조는 3라운드 이상이며, 짝수 라운드로 구성된다. Feistel 구조의 장점은 라운드 함수에 관계없이 역변환이 가능하며, 알고리즘의 수행속도가 빠르다는 것이다.

SEED 알고리즘에서 한 번의 암호화 과정은 하나의 블록을 반으로 나누고, 이들 중 오른쪽 블록을 F함수를 이용한 연산을 수행한 후, 그 값을 왼쪽의 블록과 비트 단위로 XOR 연산을 수행한다. 이 때, 왼쪽의 블록을 다음 라운드의 오른쪽 입력값으로 사용하고, F함수를 통해 나온 결과와 XOR 연산 결과는 다음 라운드의 왼쪽 입력값으로 사용한다. 그리고 이와 같은 방법을 r번 반복함으로써, 전체를 한꺼번에 암호화하거나 왼쪽 또는 오른쪽을 바꾸지 않고 계속 암호화할 때 생기는 안전성 문제를 해결할 수 있다. 다음 [표 2-8]은 SEED 알고리즘의 일반적 특징을 나타낸다.

[표 2-8] SEED 알고리즘의 특징

데이터 처리 단위	8/16/32 비트
암·복호화 방식	블록암호 방식
입·출력문의 크기	128 비트
입력키의 크기	128 비트
라운드 수	16 라운드
구조	Feistel 구조
키 생성 알고리즘	라운드 동작과 동시에 암·복호화 라운드 키 생성

[표 2-8]에 나타나 있듯이 SEED 알고리즘의 전체 구조는 대부분의 블록 암호알고리즘과 같이 Feistel 구조이다. SEED 알고리즘에서는 128비트의 평문 블록이 128비트의 키로부터 생성된 64비트의 라운드 키 16개를 입력받아 총 16번의 라운드를 거치면서, 128비트의 암호문 블록을 출력하게 된다. 다음의 [그림 2-10]은 SEED 알고리즘의 전체 구조를 도식화한 것이다.



[그림 2-10] SEED 알고리즘의 구조

SEED 알고리즘의 속도는 DES와 비슷하나, 키 생성 알고리즘은 상당히 빠른 편이며 효율적이다. 또한, 안전성과 성능에 대한 다양한 검사와 분석을 통해 SEED 알고리즘이 암호화 과정의 효율성을 지원하면서도 데이터의 안전도를 충분히 지원하고 있음을 증명하고 있다[KISA03].

2.3 RFID 인증 프로토콜

RFID 시스템의 리더와 태그 간의 통신 과정에서 리더를 이용하면 태그에 저장되어 있는 정보를 수집하는 것이 가능하며, 이는 태그 소유자의 프라이버시 침해와 밀접하게 연관되어 있다. 즉, RFID 태그에는 유일한 식별자(ID) 정보가 설정되어 있으므로, ID 정보를 입수하면 태그를 부착한 대상의 정보를 얻는 것이 가능하다. 예를 들어, 소비자의 양복 사이즈나 가격, 상점의 상품 재고 정보 등을 아는 것이 가능하며, 이로 인해 소비자의 프라이버시가 침해된다. 또한, 특정 태그를 추적함으로써 태그 소유자의 위치를 파악하는 것도 가능하다. 이러한 위치 정보에 대한 프라이버시를 위치(Location) 프라이버시라 부른다[남택용05].

RFID 시스템을 적용하게 되는 경우 발생 가능한 프라이버시 침해 문제는 RFID 기술의 많은 장점에도 불구하고 부정적으로 인식되어 RFID 기술의 발전과 확산을 저해하는 요소가 된다. 따라서, RFID 시스템에서는 프라이버시 보호 대책이 아주 중요하며, 이를 개선하기 위한 연구가 계속 진행되고 있다[Yang05].

본 절에서는 프라이버시 침해와 관련하여 RFID 시스템에 발생 가능한 공격의 유형을 알아보고, 이러한 프라이버시 침해 문제를 해결하기 위한 기존의 RFID 인증 프로토콜들을 소개하고 장·단점을 분석한다.

2.3.1 RFID 공격의 유형

RFID 시스템은 물리적 접촉 없이 인식이 가능하므로 개인 정보보호 즉, 개인의 프라이버시와 안전성 측면의 여러 문제점들을 야기시킬 수 있다. 특히 리더와 태그 간의 데이터 전송 과정에서 사용되는 무선 채널에서 발생 가능한 여러 가지 공격들로 인해 RFID 시스템의 안전성이 위협받을 수 있으며, 이 점은 RFID 시스템의 사용 자체를 부정적으로 인식하게끔 할 가능성이 있다. 따라서, 무선 채널 상에서 발생 가능한 공격들의 유형을 살펴 보고, 그에 따라 RFID 시스템이 갖추어야할 안전성 측면의 요구사항을 살펴보면 다음과 같다[황영주04].

■ 도청 (Eavesdropping)

일반적으로 RFID 태그에 저장되어 있는 정보는 일련번호나 코드번호로서 본 사물의 제조사나 가격 등의 세부 정보를 알 수는 없으나, 리더나 태그로부터 도청을 통해 획득한 이러한 정보를 이용하여 다른 세부 정보를 알아내는 데 사용할 수 있다. 즉, 도청한 태그의 정보를 재전송함으로써 RFID 시스템의 인증 과정에 참여하여 서버에 접속할 수 있으며, 지속적인 도청을 통하여 사용자에 대한 위치 추적을 할 수도 있다. 따라서, RFID 시스템은 도청된 정보로부터 어떠한 비밀 값도 유추할 수 없도록 설계되어야 하며, 이를 전방위 보안성(Forward Security)이라 한다[Feld04a]. 또한, 도청한 정보를 다시 사용하여 재전송하거나 사용자의 위치를 추적할 수 없도록 하는 인증 프로토콜을 가져야 한다.

■ 위치 추적 (Location Tracking)

위치 추적이란 RFID 태그로부터 리더에게 전송되는 정보를 가로채고 검사함으로써 이를 이용하여 태그의 위치나 소비자의 경향 등 트래픽 분석이 가능한 공격 유형을 말한다. 이것은 암호화된 메시지를 복호화할 수 없을 때에도 수행될 수 있으며, 일반적으로 많은 수의 메시지를 얻을수록 많은 정보를 트래픽으로부터 추론할 수 있다. 따라서, 안전한 RFID 시스템은 동일한 태그로부터 수신되는 정보를 매 세션마다 변경함으로써 정보를 발송하는 태그가 동일한 태그임을 구분할 수 없도록 설계되어야 한다.

■ 스푸핑 (Spoofing)

스푸핑 공격은 공격자가 중간자 공격(Man-in-the-Middle Attack) 등의 방법으로 정당한 리더나 태그로 가장하여 인증 프로토콜에 참여하는 것으로, 수집한 정보를 바탕으로 다른 세부 정보를 획득할 수 있는 공격의 유형이다. 따라서, 안전한 RFID 시스템은 태그에 저장된 정보를 암호화하거나 태그 접근 권한을 관리할 수 있는 매커니즘을 가짐으로써, 공격자가 정당한 리더나 태그로 가장하여 인증 과정에 참여하는 것이 불가능하도록 해야 한다.

■ 전송 방해 (Interference)

전송 방해 공격은 RFID 시스템의 인증 과정에 참여하지 않고 어떠한 정보도 수집하지 않지만, 정상적인 정보의 전송을 방해하여 메시지의 유실을 유발할 수 있다. RFID 인증 과정 중에 태그나 리더에서 전송되는 메시지를 유실하는 경우, 현재의 세션 또는 다음 세션의 인증 과정에서 비정상

적인 상태가 발생할 수 있다. 따라서, 안전한 RFID 시스템은 정보 전송 방해에 대한 공격 탐지 기능을 설계하도록 해야 한다.

2.3.2 RFID 시스템의 보안 요구사항

RFID 시스템의 보안 문제를 해결하기 위해서는 태그와 리더 및 서버 등 구성 환경에 대해 다음과 같은 사항을 고려하여야 한다[변영기06].

- 태그는 태그 소유자의 프라이버시를 위협하지 말아야 한다.
- 정보는 인증이 되지 않은 리더로 유출되어서는 안되며, 태그와 그 소유자 사이에 긴 기간 동안의 추적이 불가능해야만 한다.
- 추적을 막기 위해서 소유자는 보유한 태그를 감지하거나 사용 불가로 만들 수 있어야 한다.
- 공개적으로 사용 가능한 태그의 결과는 랜덤화되거나, 태그와 소유자 사이의 장기간 관련성을 회피하기 위해 쉽게 수정 가능해야 한다.
- 비공개적인 태그의 내용은 접근 제한 기법에 의해 채널이 안전하지 않다고 예상된다면 암호화되어야 한다.
- 태그와 리더는 모두 상호 신뢰할 수 있어야만 한다.
- 태그와 리더 어느 쪽이든 스푸핑이 어려워야 한다.
- 전원 중단으로 인한 프로토콜의 가로채기 공격 등의 시도에 대한 대책을 강구하여야 한다.
- 태그와 리더 모두 재전송 공격에 대한 저항력이 있어야 한다.

RFID 시스템에서 제공하고자 하는 보안 서비스 및 보안 서비스별 정보 보호에 대한 요구 사항을 다음의 [표 2-9]와 같이 정의할 수 있다. 이러한 서비스에 대한 요구 사항을 만족시키기 위해서 보안 서비스별로 적절한 보안 메커니즘을 구성할 수 있다.

[표 2-9] RFID 보안 서비스의 종류

보안 서비스	요구 사항
기밀성	권한이 있는 사용자에 의해서만 태그가 읽혀져야 하며, 태그 사용자는 태그에 기록된 데이터를 암호화할 수 있어야 한다.
익명성	태그 식별 정보 및 태그 내에 저장된 정보에 대한 익명성이 보장되어야 한다.
무결성	태그는 잠김 데이터를 사용하여 알려진 데이터의 변경이나 삭제를 막을 수 있어야 한다.
인증성	태그 데이터의 저장 장소와 전송 프로토콜은 태그 데이터를 읽기 전의 인증 요구에 대해 사용자가 제어 가능한 옵션을 제공한다.
침해 대응성	서비스 거부 공격 대응, 시스템 보호 제공, 네트워크 보호 제공

2.3.3 기존의 인증 기법들

RFID 시스템의 안전성을 위협하는 여러 공격들에 대한 문제점을 해결하기 위한 기존의 방법들로는 Kill 명령어 기법, 패러데이 케이지(Faraday Cage) 기법, 블로커 태그(Blocker Tag) 기법 등의 물리적 보안 기법들과 암호학적인 방법을 이용한 해쉬락(Hash Lock) 기법, 해쉬체인

(Hash Chain) 기법, Hash 기반 ID 변형 기법 등이 있다. 이러한 인증 기법들에 대해 상세히 살펴보기로 한다.

가. Kill 명령어 기법[AID03]

Auto-ID 센터와 EPCGlobal Inc. 에서 제안한 Kill 명령어 기법은 태그를 제조할 때 유일한 패스워드를 저장해 놓고, 태그에 올바른 패스워드가 전송되면 태그 스스로 자신의 정보를 삭제시키는 방식이다. 즉, 태그가 부착된 물건을 고객이 구입하고 물품 계산대에서 계산을 완료하는 순간 Kill 명령어가 적용되어 RFID 태그의 기능은 정지된다. 이 방법은 사용자의 프라이버시를 완벽하게 보호할 수는 있지만, RFID 기술을 이용한 정보 관리나 태그의 재사용을 불가능하게 하기 때문에 넓은 응용환경을 지원하지 못하는 단점이 있다.

나. 패러데이 케이지 기법[Juel03b]

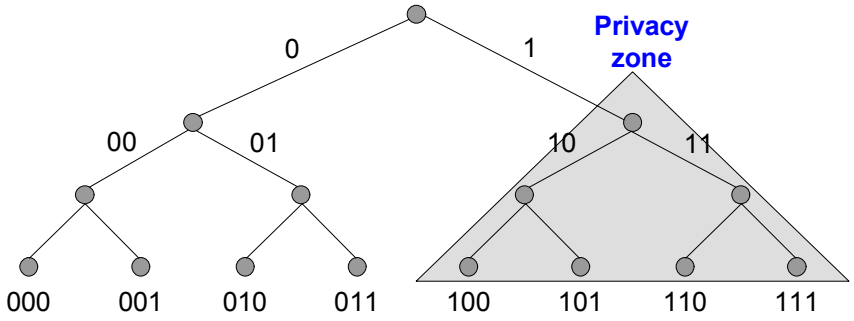
패러데이 케이지 기법에서는 RF 신호가 투과되지 않는 금속의 컨테이너를 이용하여 불법적인 리더가 태그의 정보를 읽어내는 것을 방지하도록 하는 방법이다. 미국의 mobileCloak 회사에서는 이 기술을 이용하여 [그림 2-11]과 같은 형태의 mCloak과 medCloak 제품을 판매하고 있다 [mCloak05].

[그림 2-11] 패러데이 케이지 기법을 이용한 제품들

다. 블로커 태그 기법[Juel03b]

블로커 태그 기법은 RFID 태그 위에 블로커(Blocker)를 부착하여 불법적인 리더로부터의 태그의 데이터 송신을 방해함으로써 개인의 프라이버시나 위치 추적을 불가능하도록 하는 통신 방해 기술의 일종이다. 블로커가 태그에 부착되어 있을 때는 프라이버시를 보호할 수 있으며, 블로커가 태그에서 제거되면 태그가 원래의 목적으로 사용될 수 있다.

이 기법은 태그의 응답에 대한 충돌 회피 기법으로 제안된 트리 구조[EPC03]를 사용함으로써 프라이버시가 요구되는 태그와 그렇지 않은 태그를 구분하고 특정 영역을 할당함으로써 효율성을 가질 수 있다는 것이 장점이다. 예를 들어 [그림 2-12]와 같이 ID prefix가 1로 지정된 프라이버시 보호 영역에 있는 태그는 읽지 못하도록 하는 반면, 0으로 지정된 영역은 읽을 수 있도록 선택적으로 설계할 수 있다. 이 방법은 적극적인 방식의 전파방해(Jamming) 형태는 아니며, 일종의 소극적인 전파방해 형태로 태그의 정보전송을 방해하는 방법이다.



[그림 2-12] 블로커 태그 기법의 트리 구조

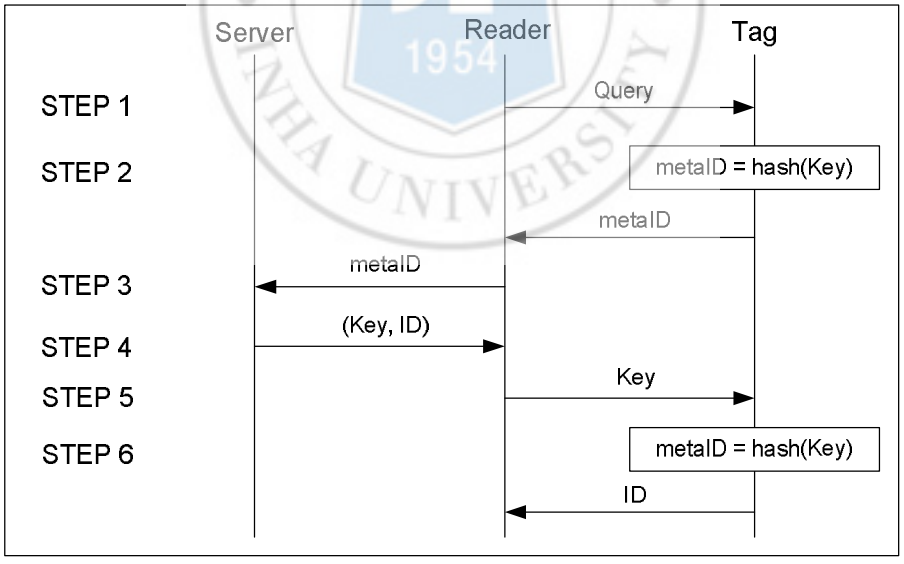
그러나, 블로커 태그는 악의적인 도구로 사용될 수도 있다. 즉, 일련번호의 모든 스펙트럼을 읽지 못하도록 차단하거나, 특별한 영역(예를 들어, 특별한 제조 회사에 할당된 일련 번호의 집합)을 리더가 읽지 못하도록 차단시킬 수 있다. 이런 형태의 블로커 태그는 비즈니스를 방해하거나 물품 명세서 제어 메커니즘으로부터 상품을 차폐시킴으로써, 좀도둑이 침입하는데 도움을 줄 수 있다.

라. 해쉬 락 기법[Weis04]

해쉬 락 기법은 태그의 실제 ID를 보호하기 위해 metaID를 이용하는 방식으로 해쉬 함수에 의해 metaID를 생성한다. 이 기법은 저가(Low-cost) 태그의 자원 제한 문제를 해결하기 위해 하드웨어적으로 최적화되어 구현된 해쉬 함수를 태그에 포함하고 있다.

[그림 2-13]과 같이 동작하는 해쉬 락 기법은 시스템 초기화 과정에서 태그 소유자가 태그의 키와 해쉬함수에 의해 생성된 metaID(metaID = hash(Key)) 값을 RFID 서버의 데이터베이스에 저장하는데, metaID 값이

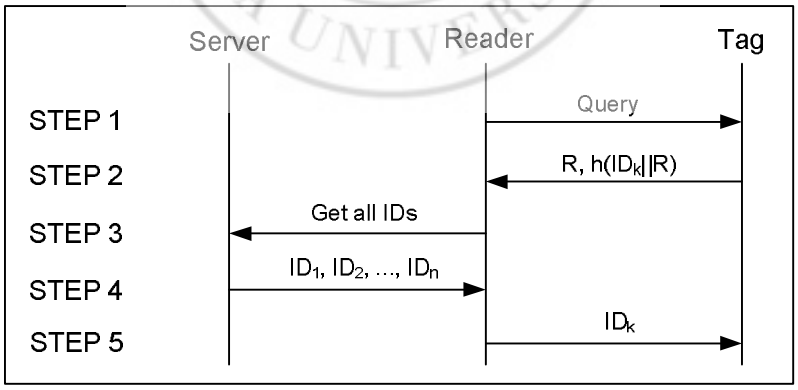
할당되자마자 태그는 잠금(Lock) 상태가 된다. 태그의 잠금 상태를 풀기 위해 리더는 metaID 값을 서버에 전송하고, 서버의 데이터베이스에 저장된 metaID에 대한 키를 찾아 리더에게 전송한다. 전송받은 키 값이 리더를 통해 태그에게 전송되면 태그에서는 해쉬함수를 이용하여 metaID 값과 일치하는지 검증한 후, 일치하는 경우 잠금 상태를 풀고(Unlock) 리더에게 자신의 ID 및 모든 기능을 제공하게 된다. 그러나, 해쉬 락 기법에서 metaID의 값은 고정되어 있으므로 공격자가 태그의 위치를 추적할 수 있다. 또한, 리더와 태그 사이의 통신 채널은 안전하지 않기 때문에 공격자가 metaID를 도청으로 알아낸 다음, 이 값을 이용하여 태그로 가장하는 스푸핑 공격도 가능하다.



[그림 2-13] 해쉬 락 프로토콜

마. 랜덤 해쉬 락 기법[Weis04]

랜덤 해쉬 락 기법은 해쉬 락 기법을 개선한 방식으로 태그가 난수를 생성하여 매 세션마다 다른 응답을 하도록 함으로써, 공격자에 의한 위치 추적을 방지할 수 있다는 장점이 있다. 이 프로토콜에서 태그는 매 세션마다 난수생성기와 해쉬함수를 이용하여 새로운 인증 정보를 생성하게 된다. 즉, 해쉬 락 프로토콜에서의 metaID 대신 난수값 R을 이용함으로써 태그의 정보를 추적하는 공격자에게 태그의 출력을 임의의 가변적인 정보 (R, hash(ID_k||R))를 사용하여 추적 문제를 해결하고 있다. 그러나, 현재까지 난수생성기의 구현에는 많은 자원이 필요하다고 알려져 있으며, 인증 프로토콜의 마지막 단계에서 리더가 태그에게 ID를 전송하는 것은 공격자에 의해 ID가 노출될 가능성이 있다는 단점이 있다. [그림 2-14]는 랜덤 해쉬 락 프로토콜의 과정을 나타낸 것이다.



[그림 2-14] 랜덤 해쉬 락 프로토콜

바. 해쉬 체인 기법[Ohku03]

해쉬 체인 기법은 서로 다른 두 개의 해쉬함수를 이용하여 리더의 질의에 대한 태그의 응답을 매 세션마다 다르게 전송하도록 하는 방식이다. 동일한 태그의 응답이 매번 달라지므로 공격자가 정보를 도청하더라도 어떤 태그의 응답인지 알 수 없으며, 서로 다른 응답을 하는 태그가 동일한 태그인지 아닌지 알 수 없다. 즉, 태그의 안전성 요구 사항을 구별 불가능성(Indistinguishability)[1]과 순방향 안전성(Forward Security)[2]으로 정의하고, 이를 만족시키는 방법으로 [그림 2-15]와 같이 해쉬 체인 메커니즘을 이용하는 것이다.

[그림 2-15]에서 H와 G는 각각 다른 두 개의 해쉬함수를 나타낸다. 태그에서는 현재의 인증 세션에서 a_{i+1}와 비밀값 S_{i+1}를 리더에게 전송하기 위하여 이전 세션에서의 a_i와 S_i 값을 해쉬함수에 적용하여 값을 생성한다. 또한, 해쉬 체인 기법은 리더에 의해 태그 정보가 변경되지 않으므로 외부 환경으로부터 태그가 안전하다고 할 수 있다.

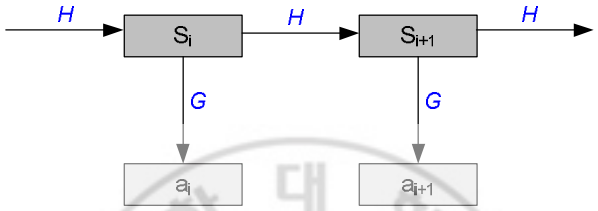
그러나, 해쉬 체인 기법은 공격자가 태그의 응답을 재전송하여 정당한 태그로 가장할 수 있는 스푸핑 공격에 취약하며, 서버에서 태그를 인증하기 위해 데이터베이스에 저장된 모든 정보에 대한 전수조사(Exhaustive Search)를 수행해야 할뿐더러 해쉬함수의 계산이 추가된다. 또한, 태그가

---

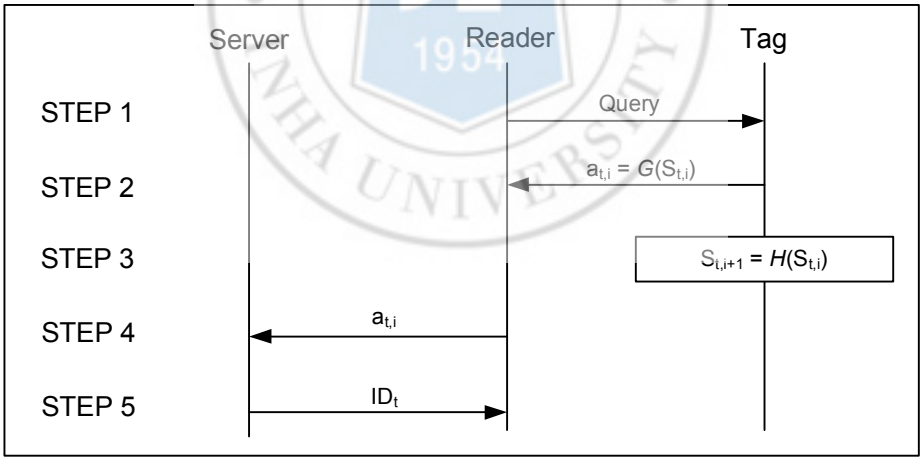
[1] 태그의 출력 값과 태그의 ID를 연결시킬 수 없어야 한다는 개념이다. 즉, 태그 ID의 익명성에 대한 개념에 포함된다.

[2] 태그의 출력을 도청하는 공격자가 현재의 출력 값와 이전의 출력 값을 연계시킬 수 없어야 한다는 개념이다.

서로 다른 두 개의 해쉬함수를 내장하고 있어야 하므로 태그의 가격이 상승된다는 문제점이 있다. 이러한 해쉬 체인 프로토콜의 전체 과정은 [그림 2-16]과 같다.



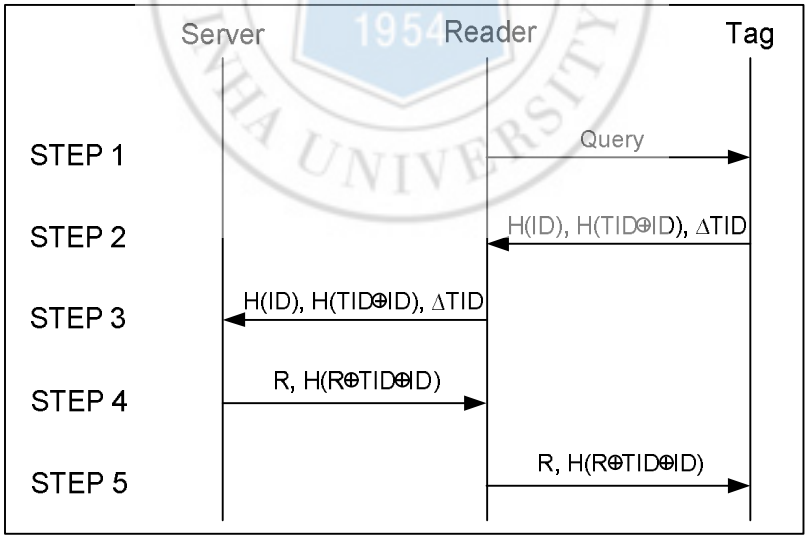
[그림 2-15] 해쉬 체인 메커니즘



[그림 2-16] 해쉬 체인 프로토콜

사. 해쉬 기반 ID 변형 기법[Henr04]

해쉬 기반 ID 변형 기법에서는 태그 ID를 난수에 의해 매번 변경함으로써 안전성을 보장하도록 한다. [그림 2-17]과 같이 매 세션마다 *TID*(Transaction ID)와 *LST*(Last Successful Transaction)을 변경시켜 프로토콜에 사용함으로써 다음 세션에서 재전송하여 공격할 수 없도록 하는 방법이다. 이 기법에서는 프로토콜의 마지막 단계에서 이루어지는 리더로부터의 메시지에 의해 태그를 인증하는데, 메시지 유실이 발생하는 경우에는 이전 세션에서의 값을 변경하지 않는다. 따라서, 공격자가 이전 세션에서의 *TID* 값을 도청한 후 태그로 가장하는 스푸핑 공격이 가능하게 된다는 단점이 있다.



[그림 2-17] 해쉬 기반 ID 변형 프로토콜

제 3 장 다중 객체 접근 방식의 태그 및 암호화 메커니즘

RFID 시스템이 적용되는 분야는 물류 관리나 제조·유통 등의 다양한 산업 환경뿐만 아니라 가정 및 직장 등과 같은 개인 생활환경에 걸쳐 넓게 분포하고 있으며, 다가올 유비쿼터스 컴퓨팅 환경에서는 그 활용도가 더욱 높아질 것으로 예상되므로 국방, 의료, 금융 등의 더욱 많은 분야에서 RFID 시스템을 적용하게 될 것이다.

현재 활용되고 있는 RFID 응용 시스템은 해당 분야의 특성과 목적에 따라 예를 들면, 개인의 가정 및 직장, 생산 및 물류 등으로 분류할 수 있는데, 이러한 각 응용 시스템들은 각각 별도의 RFID 태그를 부착하여 사용하고 있다. 한 개인이 자동차를 타고 고속도로를 이용하여 회사에 출근한다고 가정할 경우, 타고 갈 자동차의 문을 열고 시동을 걸 때 필요한 자동차용 태그와 고속도로 통행시스템을 위한 태그, 그리고 회사 출입 시스템을 통과하기 위해 또 다른 태그가 필요하다. 더구나 퇴근하여 집에 들어가기 위해 아파트 입구 통행 시스템용 태그와 주차장 및 현관문을 열기 위한 태그까지 포함한다면 한 개인이 부착해야할 RFID 태그의 종류와 수는 헤아리지 못할 정도로 많아질 수 있다. 따라서, 유비쿼터스 환경이 모든 생활 전반에 걸쳐 점차 확대되고 있는 현재 시점에서 개인 사용자의 편의성을 위해서는 하나의 태그를 사용하여 용도에 따라 여러 종류의 RFID 시스템에 접근할 수 있도록 하는 것이 더욱 적합할 것이다.

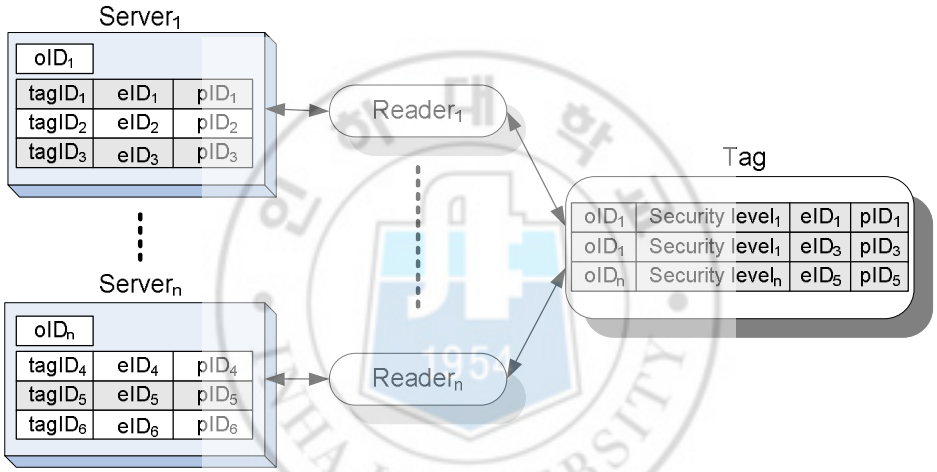
본 논문에서는 이러한 사용 편의성이라는 목적을 위해 다중 객체 접근 방식의 RFID 태그 구조를 설계하였고, 이를 이용하면 유비쿼터스 환경에서 여러 가지 종류의 정보를 통합하면서도 RFID 시스템 활용의 용이성을 실현할 수 있을 것이라 생각된다.

본 장에서는 다음과 같이 하나의 태그 내에 사용 분야와 목적에 따라 정의되는 다수 개의 ID를 가지는 태그 구조와 태그 내에 저장되는 정보의 암호화 메커니즘을 설명한다.

3.1 다중 객체 접근을 위한 태그 구조

일반적으로 RFID 시스템은 여러 개의 태그들을 서로 구별하기 위한 고유한 식별자 값으로 *tagID*를 이용한다. RFID 응용 시스템들을 그 용도에 따라 분류하여 객체의 형태로 정의할 수 있는데, 본 논문에서는 각 RFID 응용 시스템 객체에 대해 서로 구분하기 위한 객체 번호로 *oID*(object ID)를 정의한다. 제안하는 다중 객체 접근을 위한 RFID 태그는 이러한 객체를 분류하기 위한 *oID*를 메모리에 저장한다. 또한, *oID*와 *tagID*는 3.2절에서 설명할 암호화 과정에 사용되는데, 암호화 과정은 RFID 서버에서 시스템 설정 과정으로 수행된다. 암호화된 결과 값을 본 논문에서는 *eID*(encrypted ID)로 정의하고, 태그의 메모리에 *oID*와 함께 저장한다. *oID*와 *eID*는 3.3절에서 설명할 RFID 인증 프로토콜 수행 시, 응용 객체별로 각 태그를 구별할 수 있는 식별자로서 이용된다. 그리고, 인증 프로토콜의 효율적인 동작을 위해 각 응용 객체의 특성에 따라 보안 레벨(Security level)을 정하고, 태그에 객체별로 저장한다.

제안하는 다중 객체 접근을 위한 태그 구조와 RFID 구성 요소들은 [그림 3-1]과 같다. 그림에 나타나는 *pID*(partial ID)는 보안 수준에 따라 다르게 동작하는 인증 프로토콜에서 매 인증 세션마다 응답을 달리 하기 위해 태그에서 생성하는 랜덤한 값으로, 고수준의 보안을 요구하는 응용 객체의 인증 프로토콜에서만 이용한다.



[그림 3-1] 제안하는 태그 및 시스템 구조

이러한 태그 구조는 하나의 태그 내에 다수 개의 식별자 정보를 저장하도록 함으로써, 여러 종류의 RFID 응용 객체에 접근할 수 있다는 장점이 있다.

3.1.1 태그 구조의 설계

본 논문에서 제안하는 RFID 시스템의 서버와 태그에는 암호화된 값

*eID*를 저장하고 이를 태그의 식별 정보로 이용한다. 특정 리더로부터의 질의에 대해 태그에서는 태그 메모리에 저장되어 있는 *oID*의 값을 이용하여 인덱스 테이블로부터 *eID*를 검색한 후, 인증 프로토콜에 따라 리더에게 적절한 응답을 하게 된다. 제안하는 RFID 시스템은 하나의 태그 내에 여러 개의 응용 객체별 식별 정보를 포함하지만, 식별 정보 *eID*를 검색하기 위한 인덱스 구조를 이용하면 리더의 질의에 대한 태그의 응답 시간을 최소화할 수 있으며 효율적인 동작을 가능하게 한다.

RFID 시스템 서버의 데이터베이스와 태그에 저장되는 정보는 다음의 [표 3-1]과 같다.

[표 3-1] 서버와 태그에 저장되는 정보

	저장 정보
서버(DB)	oID, tagID, eID, pID
태그	oID, Security level, eID, pID

■ tagID

태그의 제조 시점에서 결정되어 ROM에 기록되는 태그별 고유한 식별 정보이다. 본 논문에서 제안하는 시스템에서는 다중 객체에 대한 정보가 하나의 태그 내에 저장되므로 한 개의 태그가 여러 식별 정보를 가져야 한다. 그러나, 본 논문에서는 *tagID*를 식별 정보로 사용하지 않고 암호화 알고리즘을 통해 암호화된 값을 각 응용 객체에 대한 식별 정보로 태그 메모리에 저장하며, *tagID*는 암호화 알고리즘의 입력 값으로 사용된다.

■ oID(object ID)

다중 응용 객체를 지원하는 RFID 시스템에서 각 객체의 종류를 구별하기 위해 정의한 ID로서, 다중 객체 접근 구조의 태그 메모리에 저장된다. 제안하는 RFID 시스템에서는 리더의 질의문에 *oID*를 포함시켜 다중 객체 정보를 저장한 태그 내에서 해당 리더의 객체 타입을 구별하게 하고 해당하는 객체의 식별 정보를 검색하게 된다.

■ eID(encrypted ID)

oID와 태그의 고유한 ID인 *tagID*를 사용하여 암호화 알고리즘을 수행한 후 생성된 값으로서, 시스템 설정 시 서버와 태그에 각각 저장된다. 하나의 태그 내에 여러 개의 *eID*가 저장될 수 있으며, 각 *eID*는 응용 객체별 태그의 고유한 식별자 정보가 된다. 또한, 여러 개의 *eID* 값이 하나의 태그에 저장되므로, 검색의 효율을 위해 태그 메모리에 인덱스 테이블을 만들고 각 *eID* 값이 저장된 블록의 번호를 기록한다.

■ pID(partial ID)

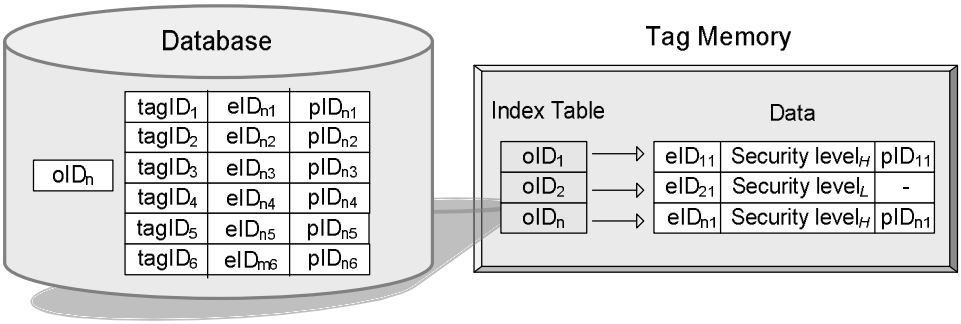
인증 프로토콜 수행 과정에서 위치 추적과 같은 공격으로부터 태그 정보를 보호하기 위해 매 인증 세션마다 생성하는 가변적인 ID 정보이다. *pID*를 만들기 위해 매 세션마다 리더의 랜덤 값 생성이 필요하며, 이러한 랜덤 값과 태그에 저장된 *eID*의 값을 연산에 적용하여 각 세션마다 다른 *pID*를 생성하게 된다. 또한, *pID*의 값은 전체 인증 과정이 성공적으로 수행되는 경우 서버와 태그에 저장된다. 그 이유는 공격자로부터 이전 세션에서 도청된 정보를 이용한 공격이 행하여졌을 경우, 서버에서는 저장되어

있는 *pID*와 비교함으로써 정상적인 인증 프로토콜이 수행되지 않았음을 감지할 수 있으며, 만약 공격자가 리더를 가장하여 태그에게 질의하는 경우에는 태그에 저장되어 있는 이전 세션에서의 pID_{n-1}와 현재 세션에서 전송된 랜덤 값을 이용하여 생성한 pID_n를 비교해 봄으로써 정당하지 않은 리더임을 알 수 있게 된다.

■ Security level

인증 프로토콜 절차를 효율적으로 설계하기 위해 각 응용 객체별로 요구되는 보안의 수준을 분류한 값이다. 하나의 태그 내에 다중 응용 객체 정보를 저장하고 있는 RFID 시스템의 인증 프로토콜은 각 응용들에 따라 요구되는 보안의 수준이 다양하게 나타날 수 있다. 따라서, 사용자 정보에 대한 보안이 매우 중요하게 처리되어야 하는 응용 분야와 그렇지 않은 응용 분야로 나누어 값을 정의하고, 이 값에 따라 인증 프로토콜이 각각 다른 절차로 동작하게 한다. 보안의 수준이 높은 응용 객체의 인증 프로토콜에서는 앞서 설명한 *pID*를 매 인증 세션마다 생성하지만, 낮은 수준의 보안이 요구되는 응용 객체의 인증 프로토콜에서는 암호화된 *eID*만을 이용하여 인증을 수행하도록 한다. 이렇게 함으로써, 요구되는 보안의 수준이 낮은 응용 객체의 태그에서 불필요한 연산을 수행하지 않도록 하여 인증 과정에 소요되는 시간을 감소시킬 수 있다.

제안하는 RFID 서버의 데이터베이스와 태그의 저장 구조는 다음 [그림 3-2]와 같다. 저수준(Security $level_L$)의 보안을 요구하는 응용 객체에서는 인증 프로토콜의 수행 과정에서 *pID*를 생성하지 않으므로, 태그 메모리에 *pID*의 값이 저장되지 않는다.



[그림 3-2] 서버와 태그의 저장 구조

3.2 SEED 알고리즘을 변형한 태그 암호화 알고리즘

본 절에서는 다중 객체 접근을 위한 RFID 태그와 서버의 데이터베이스에 저장되는 식별 정보를 암호화하기 위한 암호화 알고리즘에 대해 설명한다.

태그에 부여되는 유일한 식별 정보는 위조가 불가능해야 한다. 만약 태그를 위조하여 RFID 시스템에서 인증된 태그와 동일한 정보를 소지한다면, RFID 시스템 자체의 안전성을 위협할 수 있다. 그러나, RFID 시스템에서 사용하는 무선 통신의 특성상 태그 정보는 보안에 취약할 수밖에 없으며, 이로 인하여 태그 정보의 유출과 유통 가능성, 위변조 및 오동작과 같은 문제점이 발생하게 된다[최재귀04].

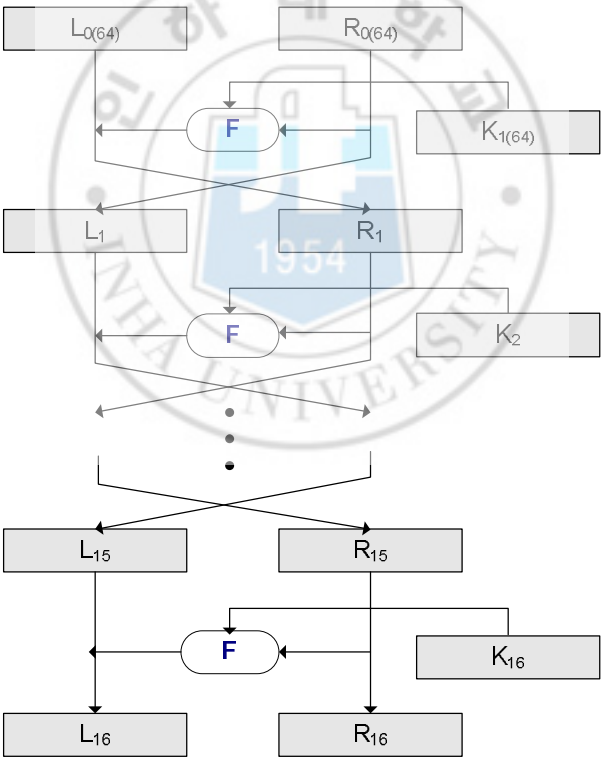
이러한 문제점들을 해결하기 위해 본 논문에서 제안하는 RFID 시스템에서는 태그에 저장되는 데이터의 암호화/복호화를 통한 보안 모듈을 적용한다. 그러나, 태그의 물리적인 특성으로 인해 RFID 태그에 적용 가능한 보안 모듈은 저전력과 고속의 성능을 가지도록 해야 한다[Park05]. 현재 국내에서 널리 사용되고 있는 SEED 알고리즘은 암호 사용을 촉진하기 위해 개발된 암호화 알고리즘이며, 본 논문에서는 이러한 SEED 알고리즘을 이용하여 다음과 같이 RFID 시스템 환경에 적합하도록 변형한다. 즉, SEED 알고리즘을 RFID 시스템과 같이 경량화를 유지하여야 하는 응용들에 적용하고자 하는 목적으로 SEED 변형 알고리즘을 설계하였다. 이러한

목적을 위해 키의 크기와 라운드 횟수를 조정하는 방법을 사용하고 있는데, 키의 크기를 이용한 방식을 K-SEED 알고리즘이라 정의하고, 라운드 횟수를 이용한 방식을 R-SEED 알고리즘이라 정의한다. K-SEED 알고리즘에서는 키의 크기를 확장하여 암호화 과정에 소요되는 전체 블록의 개수를 감소시킴으로써 암호화와 복호화 속도를 향상시키도록 한다. 그리고, R-SEED 알고리즘에서는 암호화/복호화 속도를 향상시키기 위해 라운딩 횟수를 감소시킨다. SEED 변형 알고리즘의 설계 기준으로 사용된 RFID 태그는 읽기/쓰기가 가능한 Class2 타입의 수동형 태그이며, 메모리 블록의 크기는 128비트로 한다. 암호화 알고리즘의 입력으로는 3.1절에서 정의한 *tagID* 값과 *oID*의 값을 이용한다. 다음은 제안하는 암호화 알고리즘에 사용되는 표기법이다.

- $X^{<<s}$: X를 s비트 만큼 왼쪽으로 순환 이동하는 연산
- $X^{>>s}$: X를 s비트 만큼 오른쪽으로 순환 이동하는 연산
- L_i : i 라운드에서 출력된 왼쪽 메시지 블록
- R_i : i 라운드에서 출력된 오른쪽 메시지 블록
- $K_i = (K_{i,0}, K_{i,1})$: i 라운드의 라운드키
- $K_{i,0}$: i 라운드 F함수의 오른쪽 입력키
- $K_{i,1}$: i 라운드의 F함수의 왼쪽 입력키
- $X = (X_3 \| X_2 \| X_1 \| X_0)$: G함수의 입력값
- $Y = (Y_3 \| Y_2 \| Y_1 \| Y_0)$: G함수에서 S-box(S_1, S_2)의 출력값
- $Z = (Z_3 \| Z_2 \| Z_1 \| Z_0)$: G함수의 출력값
- m_i : 상수
- KC_i : 라운드키 생성 과정에서 사용되는 i+1 라운드 상수

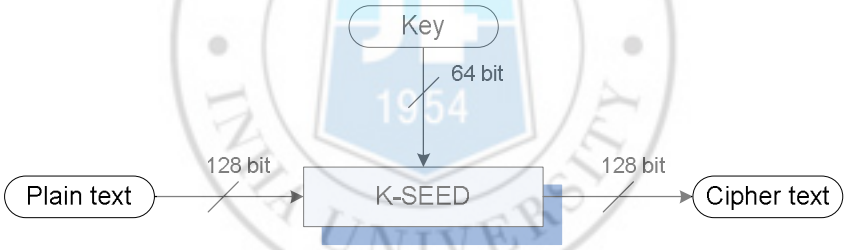
3.2.1 키 크기 변경 알고리즘(K-SEED)

SEED 알고리즘의 키 크기를 변경하여 RFID 시스템의 특성에 맞게 변형시킨 암호화 방식을 본 논문에서는 K-SEED 알고리즘으로 정의한다. K-SEED 알고리즘은 암호화 처리 속도를 높이기 위한 목적으로 키 크기를 변형하는 방식으로 설계되었다. K-SEED 알고리즘의 전체 구조는 다음의 [그림 3-3]으로 나타낼 수 있다.



[그림 3-3] K-SEED 알고리즘의 구조

[그림 3-3]에 나타난 K-SEED 알고리즘의 전체 구조는 Feistel 구조로 이루어져 있으며, 다중 객체 접근 태그의 *tagID*와 *oID*를 각각 평문과 라운드키로 하여 총 16라운드를 반복 수행하여 암호문 블록을 생성하게 된다. 태그 메모리 블록의 크기가 128비트인 경우 *tagID*는 128비트이고 *oID*는 64비트가 되므로, 128비트의 평문으로부터 생성된 64비트의 라운드 키를 입력으로 받아 총 16라운드를 거쳐 128비트의 eID값을 출력하게 된다. 즉, 태그 메모리 블록의 크기가 128비트인 경우, K-SEED 알고리즘은 [그림 3-4]와 같이 128비트의 태그 정보를 하나의 블록 단위로 인식하고, 64비트 크기의 키를 이용하여 128비트의 암호화 문서를 생성한다.

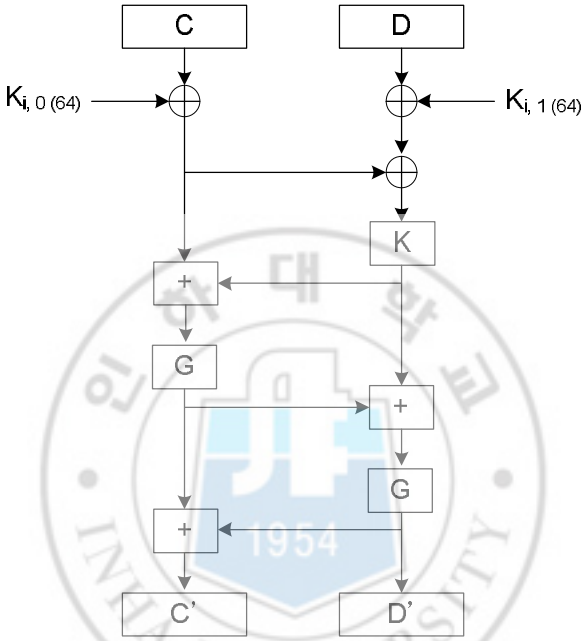


[그림 3-4] K-SEED 알고리즘의 입출력 블록 구조

가. F 함수

Feistel 구조의 블록 암호화 알고리즘은 F함수의 특성에 따라 구분할 수 있는데, 본 논문에서 제안하는 K-SEED 알고리즘의 F함수는 태그 메모리 블록의 크기를 128비트로 기준하였으므로, [그림 3-5]와 같이 64비트 크기의 블록을 처리하도록 한다. 즉, 64비트 크기의 블록 두 개(C, D)

를 입력으로 받아 64비트 크기의 블록 두 개(C’ , D’)를 출력하는데, 암호화 과정에서 C, D와 키 값으로 K_i를 F함수의 입력으로 처리한다.

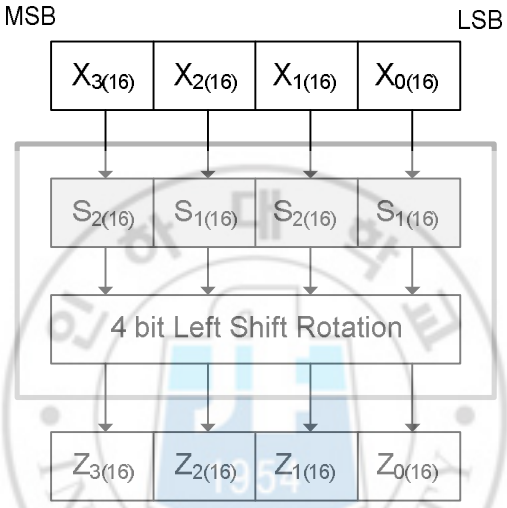


[그림 3-5] K-SEED 알고리즘의 F 함수 구조

나. G 함수

G함수는 기존의 SEED 알고리즘에서 사용하는 두 개의 16비트 S-box(S_1, S_2)를 이용하여 입력의 각 비트를 비선형 변환한 후, 그 결과인 64비트를 4비트 왼쪽으로 회전 이동한 후 출력한다. 즉, G함수의 입력 값(64비트)을 4개의 16비트 블록인 ($X_3||X_2||X_1||X_0$)으로 분할하여 2

개의 S-box에 ($S_2||S_1||S_2||S_1$) 순서로 적용시켜 ($Y_3||Y_2||Y_1||Y_0$)를 생성하고 4비트만큼 왼쪽으로 회전 이동한 후, 4개의 16비트 블록인 ($Z_3||Z_2||Z_1||Z_0$)을 생성한다. G함수의 구조는 [그림 3-6]과 같다.



[그림 3-6] K-SEED 알고리즘의 G 함수 구조

다. 키 생성 알고리즘

기존의 SEED 암호화에서 키 생성 알고리즘은 128비트의 암호 키를 64비트씩 좌우로 나누어 이들을 교대로 16비트씩 좌/우 회전 이동한 후에 생성된 64비트 결과 값에 대한 간단한 산술 연산과 G함수를 적용하여 라운드 키를 생성하고 있다. 본 논문에서 제안하는 K-SEED 알고리즘에서의 키 생성 알고리즘은 기본적으로 하드웨어나 제한된 자원을 갖는, 즉 RFID와 같은 응용에서의 효율성을 위하여 암호화나 복호화 시 암호 키로부터

필요한 라운드 키를 간단히 계산할 수 있도록 설계한다.

라운드 키를 생성하기 위한 키 스케줄(Key Schedule)은 [그림 3-7]과 같다. RFID 시스템의 적용과 속도 향상을 목적으로 암호화 동작을 경량화시키기 위해서 기존의 SEED 알고리즘에서 16비트 사용자 비밀 키를 확장시킨다. 즉, *UserKey*를 32비트로 변경하고 처리된 중간 라운드의 결과는 **AlgInfo*에 저장한다. 사용자 비밀 키를 확장시킴으로써, 암호화 과정에서 처리되는 전체 블록의 개수를 감소시키는 효과를 가져온다.

```
RET_VAL SEED_KeySchedule {
        BYTE            *UserKey,
        DWORD            UserKeyLen,
        SEED_ALG_INFO    *AlgInfo);
}
```

[그림 3-7] K-SEED 의 키 스케줄

3.2.2 라운딩 횟수 조정 알고리즘(R-SEED)

라운딩 횟수 조정에 의해 암호키를 생성할 경우, 기존 SEED 알고리즘의 라운드 처리 횟수를 줄임으로써, 각 라운드 당 처리되는 시간을 감소시킬 수 있다. 본 논문에서는 이러한 방법을 R-SEED 알고리즘으로 정의하고, 다음과 같이 설계한다.

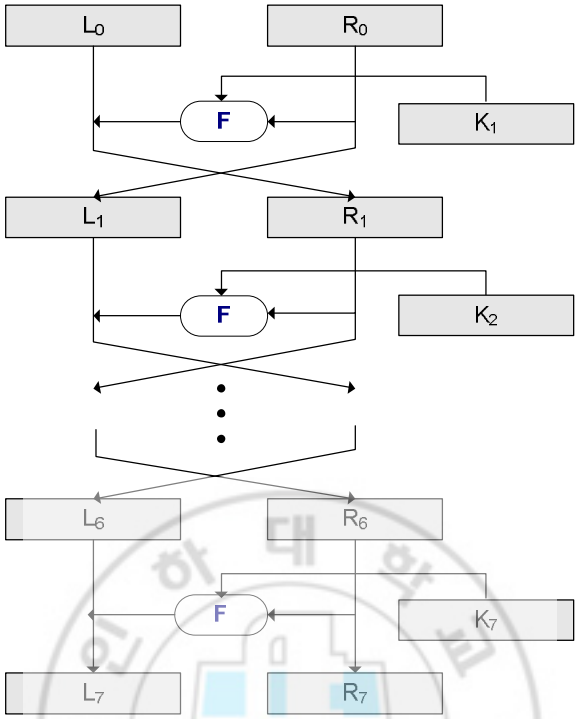
R-SEED 알고리즘은 Feistel 구조로 이루어지며, t 비트인 L_0, R_0 블록으로 이루어진 $2t$ 비트 크기의 평문 블록을 8라운드를 거쳐 암호문 L_r,

R_r을 생성해 내는 반복 구조이다. 반복 구조란 평문 블록이 몇 번의 라운드를 거쳐 암호화를 수행하는 것을 말하고, 라운드 i $(1 \le i \le r)$란 암호 키 K로부터 유도된 각 서브키인 K_i를 주요 입력으로 하는 $L_i = R_{i-1}$, $R_i = L_{i-1} \bigcirc f(R_{i-1}, K_i)$를 통해 $(L_{i-1}, R_{i-1}) \rightarrow (L_i, R_i)$로 변환해 주는 함수를 의미한다.

기존의 SEED 알고리즘에서 처리되는 라운드의 조건은 키 전수 조사 공격에 필요한 계산 복잡도 및 평문과 암호문 쌍의 크기가 2^{128} 비트 이하가 되지 않아야 하며, 효율성 요구조건을 만족하여야 한다. 그리고, 키 생성 알고리즘은 2라운드마다 일정한 규칙으로 서브키를 생성하며, 서브키를 생성하기 위한 입력 값은 (8+i)라운드의 값과 i라운드에서의 값이 동일하다. 즉, 1라운드와 9라운드에서 동일한 입력 값을 이용하여 키를 생성하게 된다. 또한, 서브키 간의 관계를 이용하여 암호를 공격하는 Related Key 공격의 경우 5라운드 이상이 되면 공격이 거의 불가능하다[KISA03]. 따라서, 기존의 SEED 알고리즘에서 처리되는 16라운드를 8라운드로 감소시켜 RFID 시스템에 적용하게 되더라도 암호 공격으로부터 비교적 안전한 성능을 보장하면서, 암호화 속도를 향상시킬 수 있다.

가. R-SEED 알고리즘의 구조

R-SEED 알고리즘은 [그림 3-8]와 같이 128비트 단위의 평문 블록당 128비트 크기의 키로부터 생성된 8개의 64비트 라운드 키를 입력받아 총 8라운드를 거쳐 128비트 크기의 암호문 블록을 출력한다.



[그림 3-8] R-SEED 알고리즘의 구조

나. F 함수

R-SEED 알고리즘의 F함수는 128비트 크기를 단위로 하는 Feistel 암호 알고리즘으로 구성된다. F함수는 두 개의 64비트 크기의 블록을 입력받아 두 개의 64비트 크기의 블록을 출력한다. 즉, 암호화 과정에서 64비트의 블록(C, D)와 64비트의 키 $K_i(K_i = K_{i,0}, K_{i,1})$를 F함수의 입력으로 하고, 처리한 결과 ($C'$, D')를 출력한다.

다. G 함수

G함수는 기존의 SEED 알고리즘과 같이 두 개의 16비트 S-box를

이용하여 입력의 각 비트를 비선형 변환한 후, 그 결과인 32비트를 4비트 왼쪽으로 회전 이동한 후 출력한다. 즉, G함수의 입력 값(32비트)을 4개의 16비트 블록인 ($X_3||X_2||X_1||X_0$)으로 분할하여 2개의 S-box에 ($S_2||S_1||S_2||S_1$) 순서로 적용시켜 ($Y_3||Y_2||Y_1||Y_0$)를 생성하고 8비트 만큼 왼쪽으로 회전 이동한 후, 4개의 16비트 블록인 ($Z_3||Z_2||Z_1||Z_0$)을 생성한다.

라. 키 생성 알고리즘

R-SEED 알고리즘에서의 키 생성 알고리즘은 64비트의 암호화 키를 32비트씩 좌우로 나누어 이들을 교대로 16비트씩 좌/우로 회전 이동한 후, 그 결과인 32비트에 대해 간단한 산술 연산과 G함수를 적용하여 다음 [그림 3-10]과 같이 기존의 16라운드 키를 8라운드로 조정하여 생성한다.

```
for (i = 1; i = 8; i++) {
        K_{i,0} ← G(A + C − KC_{i-1})
        K_{i,1} ← G(B − D + KC_{i-1})
        if (i % 2 == 1) A||I ← (A||B)^{>>8}
        else C||I ← (C||D)^{<<8}
}
```

[그림 3-9] R-SEED 의 키 생성 알고리즘

제 4 장 다중 객체 지원을 위한 인증 프로토콜

3장에서는 다중 객체 접근을 지원하는 태그의 구조와 태그에 저장되는 식별 정보를 암호화하기 위한 SEED 변형 알고리즘에 대해 알아보았다. 본 장에서는 이러한 다중 객체 태그를 포함하는 RFID 시스템을 지원하기 위한 인증 프로토콜에 대하여 설명한다.

4.1 보안 레벨 개념

다중 객체를 지원하는 태그는 다양한 종류의 RFID 응용들에 의해 여러 가지 목적으로 사용되는데, 이러한 여러 종류의 응용들에 요구되는 보안의 수준은 다양하게 나타난다. 실제 환경에 적용되는 RFID 응용 객체들은 금융 시스템, 출입/보안시스템과 같이 사용자의 정보에 대한 보호가 극히 중요한 분야가 있다. 반면, 동물 관리 시스템과 같이 요구되는 보안의 수준이 상대적으로 낮은 응용 분야도 존재한다. 이처럼 각 응용 객체마다 요구되는 보안의 중요도에 기반하여 보안 레벨을 정의할 수 있으며, 이를 기준으로 RFID 응용 객체들을 분류할 수 있다. 분류된 RFID 응용들에 대해 레벨에 따라 다른 인증 프로토콜을 적용하게 되면, 인증 과정에 소요되는 시간과 연산량을 감소시킬 수 있다는 장점이 있다. 다음 절에서는 다중 객체를 지원하는 태그에 대해 보안 레벨에 따라 구분하여 동작하는 인증 프로토콜에 대해 상세히 설명한다.

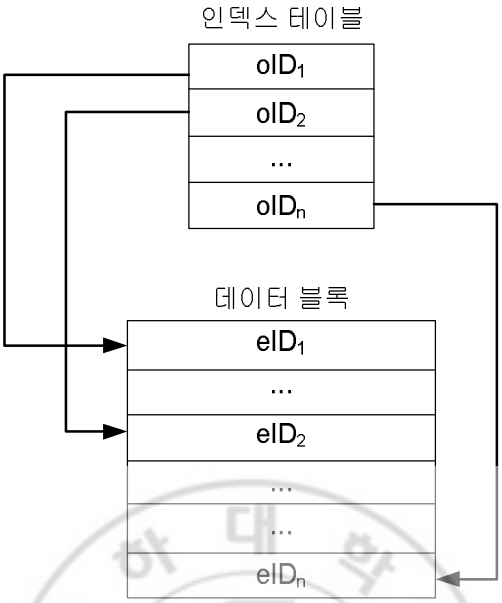
4.2 인증 프로토콜의 설계

본 절에서는 다중 객체를 지원하는 태그 구조에 적합한 인증 프로토콜에 대해 설명한다. 제안하는 인증 프로토콜은 효율적인 동작을 위해 각 RFID 응용 객체에서 요구되는 보안 레벨을 정의하고, 보안의 수준에 따라 RFID 시스템의 구성요소들이 다르게 동작하도록 설계하였다. 보안 레벨은 저수준 및 고수준의 두 단계로 정의하였으며, 각 단계에 해당하는 저수준 인증 프로토콜과 고수준 인증 프로토콜에 대해 상세히 설명한다. 다음은 제안하는 인증 프로토콜에 사용된 표기들이다.

- *oID* : RFID 응용 객체의 종류를 구별하기 위한 식별자
- *tagID* : 태그별 고유한 식별자
- *RNo* : 매 세션마다 리더의 질의에 포함되는 임의의 값
- *pID* : 매 세션마다 태그에서 생성되는 가변적인 값
- *eID* : SEED 변형 알고리즘에 의해 암호화된 값
- $R_{Forward\_no}$: *RNo*의 부분 값으로 왼쪽 비트 추출 시작 위치
- $R_{Backward\_no}$: *RNo*의 부분 값으로 오른쪽 비트 추출 시작 위치
- $R_{Forward}$: $R_{Forward_no}$를 eID의 비트 크기에 맞게 계산한 실제 왼쪽 비트 추출 위치
- $R_{Backward}$: $R_{Backward_no}$를 eID의 비트 크기에 맞게 계산한 실제 오른쪽 비트 추출 위치
- *S* : 현재 세션의 인증 과정에 참여하는 RFID 서버
- *R* : 현재 세션의 인증 과정에 참여하는 RFID 리더
- *T* : 현재 세션의 인증 과정에 참여하는 RFID 태그

4.2.1 다중 객체 정보 검색을 위한 인덱스 구조

다중 객체를 지원하는 태그에 접근하기 위한 메시지는 기존의 RFID 시스템에서의 구조와는 달리 객체 정보를 포함해야 한다. 즉, RFID 시스템의 리더는 객체 정보를 포함한 메시지를 태그에게 전송하여 태그에게 질의해야 한다. 특정 응용 객체의 리더로부터 전송되는 질의에 대해 태그에서는 태그 메모리에 저장되어 있는 객체 식별자(*oID*)의 값을 이용하여 해당하는 암호화된 값(*eID*)를 검색하는데, 검색의 효율성을 높이기 위해 인덱스 테이블을 사용한다. 이러한 인덱스 구조는 리더의 질의에 대한 태그의 응답 시간을 최소화할 수 있으며 효율적인 동작을 가능하게 한다. 인덱스 구조에서는 각 *oID*를 인덱싱하여 해당하는 *eID*가 저장된 블록 번호를 빠른 시간에 검색할 수 있도록 다음의 [그림 4-1]과 같이 설계한다.



[그림 4-1] 태그에 저장되는 인덱스 구조

또한, 본 논문에서 제안하는 RFID 시스템의 인증 프로토콜을 수행하기 위해서는 서버에서 SEED 변형 알고리즘으로 암호화한 값인 *eID*를 각 태그 메모리와 서버 데이터베이스에 시스템 설정 단계 수행 시 저장해야 한다. SEED 변형 알고리즘은 *oID*와 *tagID*를 입력 값으로 하여 암호화를 수행하고 그 결과 값으로 *eID*를 생성해 내는데, 이렇게 생성된 *eID*는 물리적인 복제 및 해독이 용이하지 않으므로 불법적인 사용자로부터 태그의 식별 정보를 안전하게 보호할 수 있다.

4.2.2 저수준 인증 프로토콜

저수준 보안 레벨에서의 인증 프로토콜은 저가의 물품에 대한 물류

관리나 고속도로 통행 시스템와 같이 빠른 응답을 필요로 하는 RFID 시스템에 적용될 수 있는 인증 절차이다. 저수준 인증 프로토콜에서는 랜덤 값 생성과 같은 과정들을 감소시켜 리더와 태그에서의 연산량을 최소화시킴으로써, 인증 절차를 수행함에 있어 저사양의 태그에서도 효율적인 동작이 가능하도록 설계하였다. 상세한 인증 절차는 다음과 같으며, [그림 4-2]와 같이 저수준 인증 프로토콜의 전체 과정을 표시할 수 있다.

STEP 1. RFID 리더는 해당하는 RFID 응용 시스템 객체의 *oID*를 질의문에 포함시켜 태그에게 전송한다.

$$R \rightarrow T : Query(oID)$$

STEP 2. RFID 태그는 전송받은 *oID*를 이용하여 태그 메모리에 저장되어 있는 인덱스 테이블을 검색한다. 인덱스 테이블에는 객체 번호별로 암호화된 태그 식별자 *eID* 값이 저장된 블록의 위치를 기록하고 있으므로, 저장된 블록을 읽어들여 검색된 *eID*를 리더에게 전송한다.

$$T \rightarrow R : eID$$

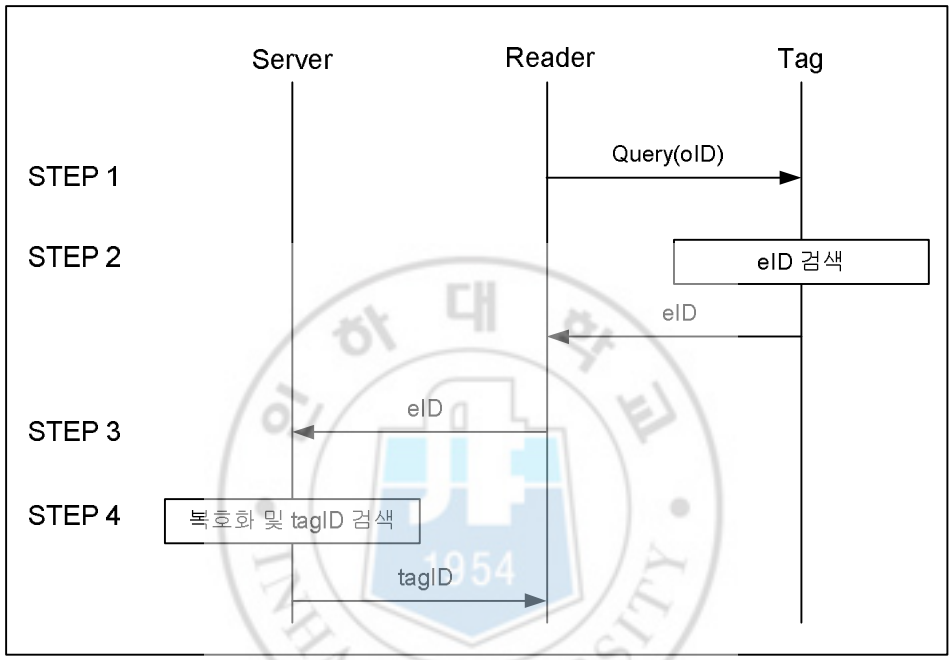
STEP 3. RFID 리더는 태그로부터 전송받은 *eID*를 해당하는 RFID 응용 객체의 서버로 전송한다.

$$R \rightarrow S : eID$$

STEP 4. RFID 서버에서는 리더로부터 전송된 *eID*를 SEED 변형 알고리즘에 적용시켜 복호화 과정을 수행한다. 복호화 과정의 결과 생성된 *tagID*를 이용하여 태그 사용자 및 해당 태그의 정보를 데이터베이스에서 검색할 수 있고, 검색 결과에 의해 현재의 세션에 참여한 태그가 정당한 태그인지 알 수 있다. 정당한 태그인 경우

서버는 리더에게 *tagID*를 전송함으로써 인증 절차를 종료한다.

$$S \rightarrow R : tagID$$



[그림 4-2] 저수준 보안 레벨에서의 인증 절차

4.2.3 고수준 인증 프로토콜

금융 시스템이나 고가의 물품에 대한 물류 관리 시스템에서는 사용자 정보나 물류 정보에 대해 고수준의 보안을 필요로 한다. 이러한 고수준의 보안 레벨을 요구하는 RFID 응용 시스템에서는 다음과 같이 고수준 인증 프로토콜을 적용하여 좀 더 안전한 인증 과정을 거치도록 설계하였다.

또한, 고수준 인증 프로토콜에서는 태그에서의 연산량을 감소시키기 위해 비트 추출 방식을 사용함으로써, 랜덤 값을 생성하거나 해쉬 함수를 이용하는 기존의 다른 RFID 인증 프로토콜보다 효율적으로 동작할 수 있다. 고수준 인증 프로토콜의 상세한 인증 절차는 다음과 같으며, 전체 인증 과정은 [그림 4-4]와 같이 표시할 수 있다.

STEP 1. RFID 리더는 랜덤 값 *RNo*를 생성하고, 해당하는 RFID 응용 시스템 객체의 *oID*와 함께 질의문에 포함시켜 태그에게 전송한다. *RNo*의 값은 매 세션마다 다르게 생성되며, 다음과 같이 두 개의 값 $R_{Forward_no}$와 $R_{Backward_no}$가 연접(concatenation)하여 구성된다.

$$R \rightarrow T : Query(oID, RNo)$$

$$RNo = R_{Forward\_no} \,||\, R_{Backward\_no}$$

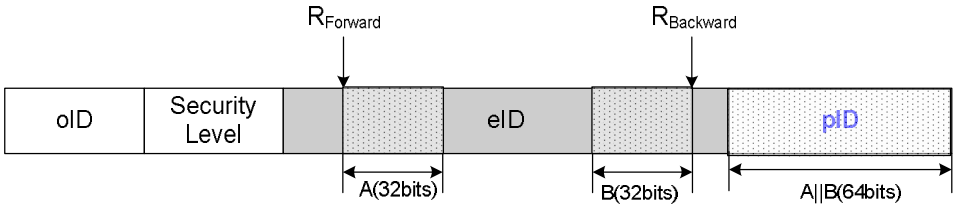
STEP 2. RFID 태그에서는 리더로부터 전송받은 *oID* 값을 이용하여 태그 메모리에 저장되어 있는 인덱스 정보에서 해당 *eID*를 검색한다. *eID*를 검색한 후, 함께 전송된 *RNo* 값을 이용하여 다음과 같이 $R_{Forward}$와 $R_{Backward}$를 계산한다.

$$R_{Forward} = R_{Forward\_no} \bmod (size\ of\ eID / 2)$$

$$R_{Backward} = R_{Backward\_no} \bmod (size\ of\ eID / 2)$$

이렇게 생성된 $R_{Forward}$ 와 $R_{Backward}$ 는 *eID*의 값에서 임의의 비트를 추출하는 위치를 의미하는데, [그림 4-3]과 같이 *eID*로부터 정해진 32비트 크기만큼 추출해 낼 때 사용된다. 추출된 부분을 각각 A와 B라고 할 때, A와 B를 연접한 결과로 64비트의 값이 생성되는데 이를 *pID*라 정의하고, *pID*의 값을 태그 메모리에

저장한다.



[그림 4-3] pID 생성 과정

STEP 3. 태그는 *STEP 2* 에서 생성된 *pID*를 해당 리더에게 전송한다. 리더에게 전송되는 *pID*는 같은 태그일지라도 매 세션마다 다른 값으로 생성된다.

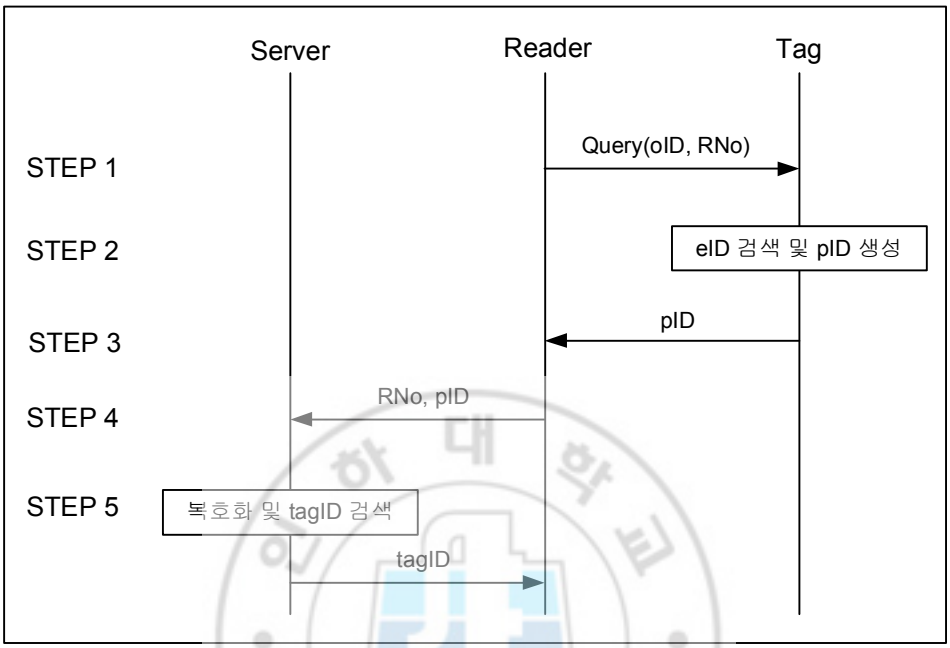
$$T \rightarrow R : pID$$

STEP 4. 리더는 *STEP 1* 에서 생성한 *RNo*와 태그에서 전송된 *pID*를 해당 RFID 응용 시스템 객체의 서버에게 함께 전송한다.

$$R \rightarrow S : (RNo, pID)$$

STEP 5. RFID 서버에서는 전송된 *RNo*와 *pID*를 이용하여 태그에서의 연산을 역 수행하여 해당하는 태그의 *eID*를 계산해 낸다. *eID*의 값이 생성되면 이를 이용하여 SEED 변형 알고리즘의 복호화 과정을 수행한다. 복호화 결과 생성된 *tagID*를 이용하여 서버의 데이터베이스에서 사용자 정보를 검색할 수 있고, 검색된 결과로 현재의 태그가 정당한 태그인지 알 수 있다. 정당한 태그인 경우 서버는 리더에게 *tagID*를 전송함으로써 인증 절차를 종료한다.

$$S \rightarrow R : tagID$$



[그림 4-4] 고수준 보안 레벨에서의 인증 절차

제 5 장 실험 및 분석

본 장에서는 4장에서 제안한 SEED 변형 암호화 알고리즘과 다중 객체를 지원하는 RFID 시스템의 인증 프로토콜에 대한 실험 방법과 환경에 대해 설명한다. 또한, 본 논문에서 제안한 암·복호화 알고리즘의 속도, 다중 객체 접근 구조의 태그를 포함한 RFID 시스템에 대한 공격으로부터의 안전성, 그리고 태그 내에서의 연산량 및 인증 프로토콜 수행시간과 에러율에 대해 실험하고, 그 결과에 대해 설명한다. 그리고, 실험 결과에 기반하여 본 논문에서 제안하고 있는 암호화 알고리즘과 인증 프로토콜의 특징과 성능에 대해 분석한다.

5.1 실험 방법 및 환경

SEED 알고리즘의 키 크기를 변형시킨 K-SEED 알고리즘과 라운드 횟수를 변경시킨 R-SEED 알고리즘의 성능 평가를 위해서는 암호화와 복호화에 걸리는 속도를 측정하여 기존의 SEED 알고리즘과 비교하여 분석하였다.

또한, RFID 인증 프로토콜에 대해서는 공격에 대한 안전성과 태그가 필요로 하는 연산량을 기준으로 기존의 인증 기법들과 비교하여 성능을 평가한다. 안전성에 대해서는 도청과 위치추적, 재전송 공격, 스푸핑 공격 및 정보 전송 방해 공격으로부터 사용자 프라이버시를 보호할 수 있는지의 여부를 기존의 인증 프로토콜과 비교한 후 분석하였다. 마찬가지로, 태그가

필요로 하는 연산량에 대해서도 기존의 기법들에서 필요로 하는 연산량과 비교함으로써 제안하는 프로토콜의 연산량이 기존의 기법들보다 향상된 성능을 보이는지를 알 수 있다.

그리고, 인증 절차에 소요되는 시간과 평균 에러율을 RFID 인증 프로토콜의 성능 평가 항목으로 하여 시뮬레이션을 통해 값을 측정하였다. 시뮬레이션에 사용된 RFID 시스템은 다음의 [표 5-1]과 같은 사양의 리더와 태그를 포함한다.

[표 5-1] 리더와 태그의 상세 정보

구 분	사 양	
Reader	Size	200 x 250 x 35(mm)
	Communication Buffer	250bytes
	Rated Voltage	DC 9~12V / 0.5~1A
	Interface	RS-232
	Operating Frequency	13.56MHz band
	Type	ISO15693, 14443A
	Read Range	5~20cm
Tag	Communication Standard	ISO15693
	Operating Range	1.5m
	Communication Speed	26kbps
	Memory Type	2kbit EEPROM, read/write
	Block Size	128 비트
	Write-once memory space for data protection	

[표 5-1]에서와 같이 실험에 사용한 태그는 읽기·쓰기 가능한 Class2(EPCGlobal Inc. 분류 기준) 타입의 수동형 태그이며, 인증 과정에서 필요한 정보를 메모리에 읽거나 기록할 수 있다. 따라서, 암호화 알고리즘에 의해 생성되는 *eID*를 기록하고, 인증 프로토콜의 수행 과정에서 생성되는 *pID* 값을 저장할 수 있다.

실험에 사용된 RFID 시스템은 현재 물류/유통 관리와 고속도로 통행 시스템 등에 주로 사용되고 있는 제품으로, 본 논문에서 제안한 다중 객체에 대한 접근 방식을 지원할 수 없다. 즉, 기존의 단일 객체 접근 방식으로 동작하며, 리더가 태그와의 무선 통신을 통해 태그 메모리의 특정 블록에 저장된 정보를 단순히 읽어오는 형태이다. 또한, 태그로부터 읽어들인 정보에는 하나의 특정 RFID 응용에 사용되는 한 개의 태그에 대한 식별정보가 포함된다.

이러한 단일 객체 접근 방식의 RFID 시스템 환경에서 다중 객체 접근 방식의 인증 프로토콜에 대한 실험을 수행하기 위한 목적으로 시뮬레이터를 설계하였다. 시뮬레이터의 동작은 제조사에 의해 지원되는 리더 프로그램(MxPanel)과 통신하면서 태그에게 질의하고, 태그의 정보를 읽어들여 데이터베이스에서 검색하거나 정보를 저장한다.

또한, 시뮬레이터를 이용하여 인증 프로토콜을 수행하기 위해서 태그의 메모리에는 [표 5-2]와 같이 다중 응용 객체들의 정보를 저장하기 위한 구조를 정의한다.

[표 5-2] 태그 메모리 블록 구조

Block No.	Description	Definition
0	Serial Number	Tag ID
1	Configuration Block	
2	Secured e-Purse Area	
3	Debit Key	
4	Credit Key	
5	Application Issuer Area	
6	Read/Write Area	Index Table
7	Read/Write Area	eID of oID_0
8	Read/Write Area	pID/Security level of oID_0
9	Read/Write Area	eID of oID_1
10	Read/Write Area	pID/Security level of oID_1
…	Read/Write Area	…
29	Read/Write Area	eID of oID_{12}
30	Read/Write Area	pID/Security level of oID_{12}
31	Read/Write Area	Undefined

위의 [표 5-2]에는 인덱스 테이블이 저장되는데(6번 블록), 이 인덱스 테이블에는 각 객체 번호(*oID*)별로 식별 정보(*eID*와 *pID*)의 값이 저장되는 블록의 위치가 다음의 [표 5-3]과 같이 기록되어 있다. 다중 객체 접근 방식의 RFID 태그에는 태그 메모리에 여러 응용 객체에 대한 정보와

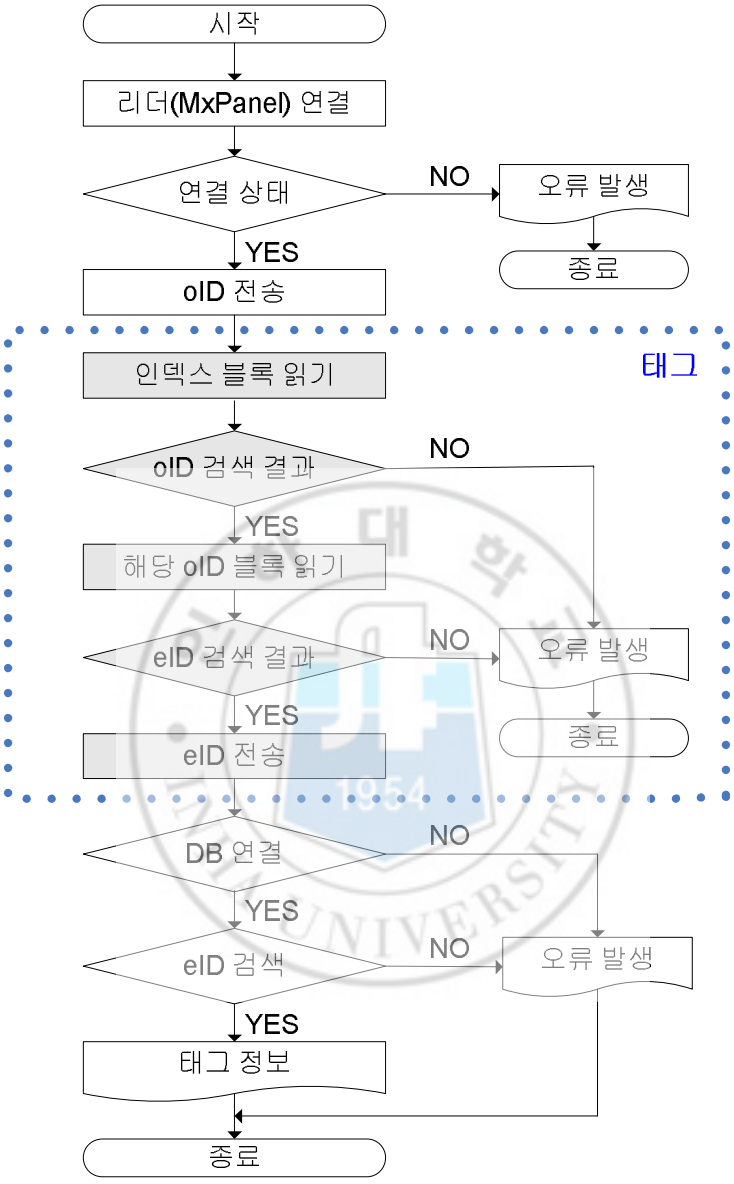
그에 따른 태그별 식별 정보가 저장되므로, 특정 객체에 해당하는 식별 정보를 효율적으로 검색하기 위해 이러한 인덱스 테이블을 이용한다.

[표 5-3] 다중 객체 정보 검색을 위한 인덱스 테이블

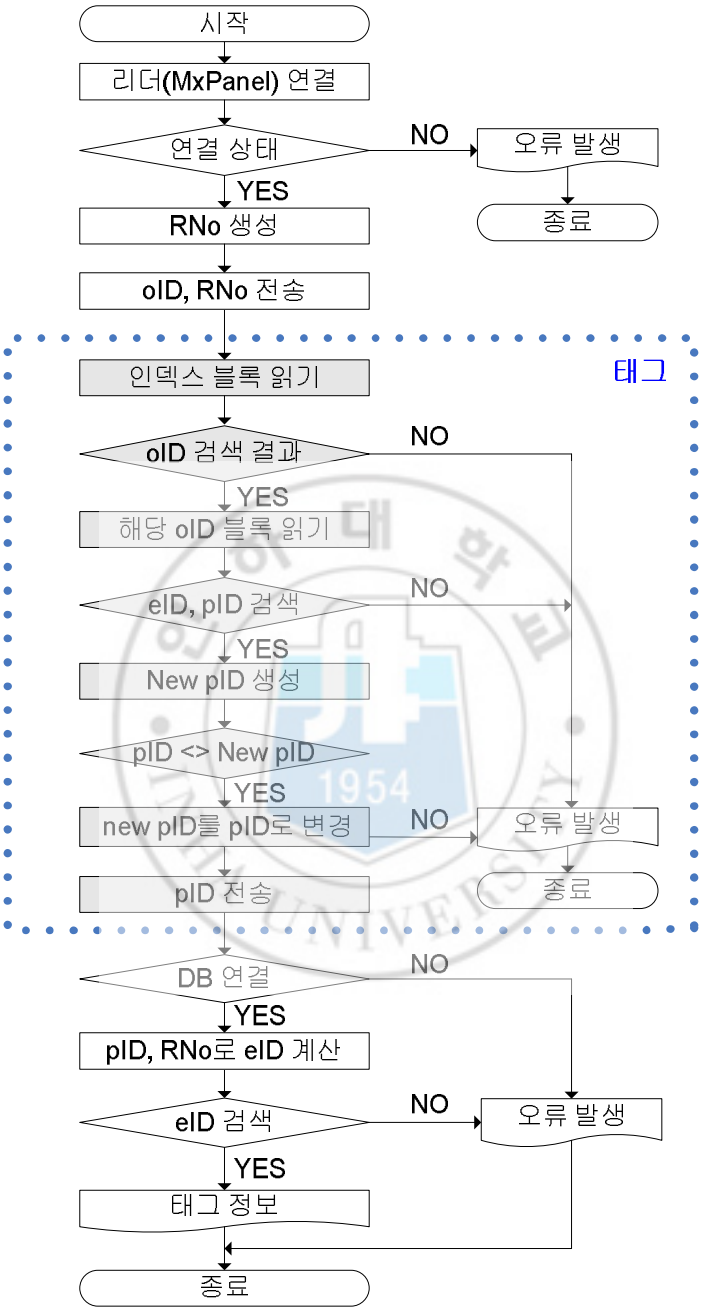
oID	해당 eID 의 Block No.	해당 pID 의 Block No.
oID_0	7	8
oID_1	9	10
oID_2	11	12
oID_3	13	14
…	…	…
oID_{11}	27	28
oID_{12}	29	30

시뮬레이션 수행을 위해 총 100개의 태그를 실험에 사용하였으며, 각 태그마다 여러 개의 객체 데이터 예제를 기록하고 동일한 데이터를 서버의 데이터베이스에도 저장하였다. 이 때, 태그와 서버에 저장되는 데이터에는 암호화 알고리즘에 의해 생성된 *eID*가 포함되며, 암호화와 초기 데이터 설정 시간은 연산 시간에서 제외하였다.

[그림 5-1]과 [그림 5-2]는 보안의 수준에 따라 각 인증 프로토콜을 시뮬레이션하기 위한 과정을 나타낸 그림이다.



[그림 5-1] 저수준 인증 프로토콜 순서도



[그림 5-2] 고수준 인증 프로토콜 순서도

5.2 암호화 알고리즘의 성능 평가

암호화에 사용되는 키의 크기를 변경한 K-SEED 알고리즘과 라운드 횟수를 변경한 R-SEED 알고리즘의 성능을 평가하기 위해, 두 알고리즘에 대해 각각 암·복호화 처리 속도를 측정하였다. 또한, 기존의 SEED 알고리즘에 대해서도 암·복호화 처리 속도를 함께 측정하여 그 결과를 비교, 분석한다.

각 암호화 방식에 따라 실험한 결과는 다음 [표 5-4]와 같으며, 실험에 사용된 컴퓨터는 Pentium IV 2.80, 메모리 크기는 2GB이다. 암호화와 복호화 처리 속도의 평균값은 최고와 최저 값을 뺀 나머지 값들을 대상으로 한 결과이다. [표 5-4]의 실험 결과를 바탕으로 기존의 SEED 알고리즘에 대해 제안한 K-SEED 알고리즘의 암호화와 복호화의 속도를 비교하여 성능을 계산하면 다음과 같다.

$$\begin{aligned} \textit{Encryption Performance} &= (\textit{K-SEED} - \textit{SEED}) / \textit{SEED} * 100 \\ &= (22.254 - 17.185) / 17.185 * 100 \\ &= 29.496\,(\%) \end{aligned}$$

$$\begin{aligned} \textit{Decryption Performance} &= (\textit{K-SEED} - \textit{SEED}) / \textit{SEED} * 100 \\ &= (23.001 - 17.077) / 17.077 * 100 \\ &= 34.689\,(\%) \end{aligned}$$

즉, K-SEED 알고리즘을 적용하여 암호화를 수행하는 경우, 기존의 SEED 알고리즘보다 29퍼센트 정도의 성능 향상을 보여주고 있으며, 복호화의 경우에도 약 34퍼센트의 성능 향상을 확인할 수 있다.

[표 5-4] 암호화와 복호화 처리 속도 (E:암호화, D:복호화, 단위:Mbps)

No.	SEED (E / D)	K-SEED (E / D)	R-SEED (E / D)
1	17.220 / 17.517	22.002 / 23.532	30.103 / 31.894
2	17.484 / 16.279	21.261 / 21.669	30.163 / 31.059
3	17.249 / 16.987	22.947 / 23.418	30.280 / 31.738
4	16.961 / 17.364	22.489 / 21.486	31.102 / 32.954
5	17.278 / 17.249	22.539 / 23.098	31.281 / 32.738
6	17.305 / 17.133	22.590 / 23.209	31.280 / 33.151
7	16.848 / 15.971	21.955 / 23.263	30.720 / 31.280
8	16.357 / 17.334	22.534 / 23.209	32.079 / 33.312
9	17.308 / 17.305	22.685 / 23.263	31.662 / 32.687
10	17.018 / 17.397	21.767 / 22.890	28.403 / 30.281
11	17.264 / 16.357	22.539 / 22.947	31.862 / 32.687
12	17.397 / 17.364	21.440 / 23.046	31.079 / 31.894
Avg.	17.185 / 17.077	22.254 / 23.001	30.953 / 32.208

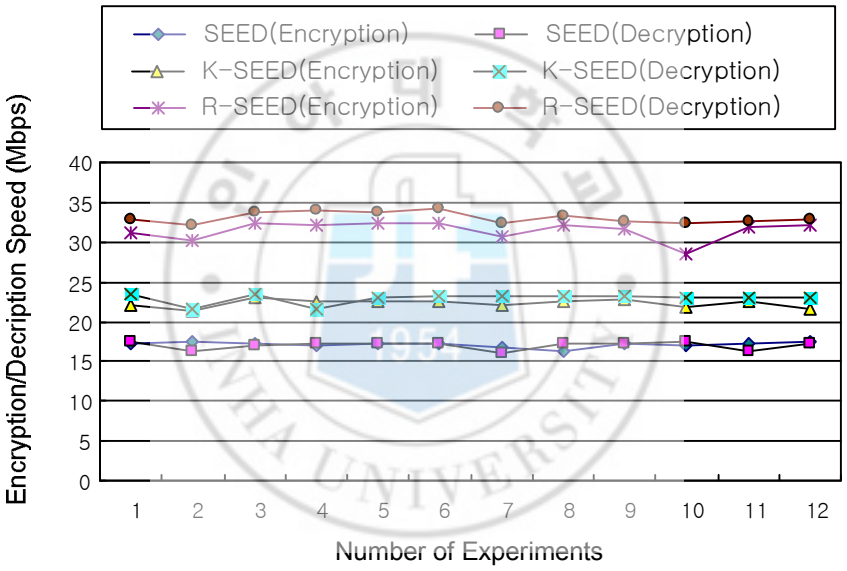
같은 방법으로 기존의 SEED에 대해 제안한 R-SEED 알고리즘의 암호화와 복호화의 성능을 비교, 계산하면 다음과 같다.

$$Encryption\ Performance = (R\text{-}SEED - SEED) / SEED * 100$$
$$= (30.953 - 17.185) / 17.185 * 100$$
$$= 80.116(\%)$$

$$Decryption\ Performance = (R\text{-}SEED - SEED) / SEED * 100$$
$$= (32.208 - 17.077) / 17.077 * 100$$

$= 88.604\,(\%)$

계산 결과, R-SEED 알고리즘의 경우 암호화와 복호화의 속도는 기존의 SEED 알고리즘에 비해 80퍼센트 이상의 성능 향상을 가져옴을 알 수 있다. 측정된 각 방법의 암호화와 복호화 처리 속도를 그래프로 표시하면 다음의 [그림 5-3]과 같다.



[그림 5-3] 암·복호화 성능 비교

5.3 인증 프로토콜의 안전성 평가

4장에서 제안한 RFID 인증 프로토콜은 다중 객체 접근을 지원하면서

다음과 같은 공격에 대해 안전성을 보장할 수 있다.

가. 도청 공격

RFID 리더와 태그 사이에 전송되는 정보가 공격자에 의해 도청에 의한 방법으로 획득되면 스푸핑 공격이나 태그 소유자의 위치 추적에 활용될 수 있다. 제안하는 인증 프로토콜에서는 도청된 내용을 그대로 이용할 수 없도록 하기 위하여 리더의 질의에 랜덤 값(*RNo*)을 포함시키고, 그에 따른 태그의 응답은 랜덤 값에 따라 매번 달라지는 *pID* 값을 생성하여 전송하도록 하였다. 또한, 태그에 의해 생성되는 가변적인 정보 *pID*는 암호화된 식별자(*eID*)를 임의로 추출한 값으로서, 공격자에게 어떠한 태그 식별 정보도 제공하지 않도록 한다.

나. 위치 추적과 재전송 공격

제안하는 인증 프로토콜에서는 리더의 질의에 대한 태그의 응답이 매번 달라진다. 따라서, 공격자가 태그의 실제 식별자는 물론 태그가 각각의 인증 세션에서 서로 다른 응답을 하게 되어, 이러한 서로 다른 응답이 동일한 태그에 의한 것인지 아닌지를 판별할 수 없도록 한다. 그러므로 제안하는 프로토콜은 위치 정보 노출에 대하여 ‘ 익명성(강)’ [이근우05][1]을 보장할 수 있다. 또한, 공격자가 이전 세션에서 획득한 정보를 이용하여 리더나 태그로 가장하여 재전송하는 경우, 현재 세션에서 만들어진 pID_n과

[1] 위치 정보 노출의 익명성(약) 보장은 리더의 질의에 대한 태그의 응답으로부터 태그의 실제 ID는 알 수 없으나, 하나의 태그가 각각이 인증 세션에서 서로 다른 응답을 하더라도 이러한 서로 다른 응답이 동일한 태그에 의한 것임을 알 수 있는 경우이다.

태그 메모리에 저장되어 있던 이전 세션에서의 pID_{n-1}의 값을 비교하여 보면 현재 세션에서의 리더나 태그가 정당하지 않은 상대임을 알 수 있으므로 재전송 공격을 불가능하게 한다. 즉, 공격자가 pID_{n-1} 값을 도청한 후, 정당한 태그로 가장하여 리더의 질의에 대한 응답으로 도청한 값을 재전송하는 경우, 매 세션마다 달라지는 랜덤 값(*RNo*)을 알지 못하고서는 올바른 응답이 아님을 서버에서의 연산을 통해 알 수 있게 된다.

다. 스푸핑 공격

본 논문에서 제안한 RFID 인증 프로토콜은 공격자가 리더로 가장하는 경우, 특정 *oID*의 값을 알아야 하고 임의의 *RNo*의 값 또한 랜덤하게 생성하여 태그에게 전송하여야 한다. 제안한 인증 프로토콜에서 리더는 랜덤 값을 생성할 때 그 값의 범위를 결정해야 하는데, 이는 태그 메모리의 데이터 블록 크기와 구조를 알아야만 가능하다. 만약 공격자가 이전 세션에서 도청한 정보를 그대로 사용한다 하더라도, 태그와 서버에 저장된 *pID* 값을 비교해 봄으로써, 정상적인 인증 절차가 아님을 알 수 있게 된다.

또한, 공격자가 태그로 가장하는 경우에는 태그에 저장되어 있는 암호화된 식별자인 *eID*를 알지 못하고서는 올바른 *pID*를 생성할 수 없고, 리더의 질의에 포함되는 랜덤 값은 매번 변경되므로 이전 세션에서 도청한 값을 이용할 수도 없게 된다.

라. 전송 방해 공격

전송 방해 공격은 정상적인 정보의 전송을 방해하여 메시지의 유실을 유발한다. 태그에 저장되는 식별 정보가 인증 과정에서 전송되는 값으

로 변경되는 가변 ID 방식의 인증 프로토콜[최재귀04]에서는 전송 방해 공격으로 인해 메시지를 유실하게 되면 이후의 모든 세션에서 인증 프로토콜이 정상적으로 동작할 수 없게 된다. 제안하는 RFID 인증 프로토콜에서는 태그 메모리에 암호화된 식별자 *eID*를 저장하고, 이것을 유일한 식별 정보로 유지한다. *eID*의 값은 고정적이므로 리더와 태그 간의 인증 프로토콜을 수행하는 과정에서 발생할 수 있는 정보 유실에 대한 복구 과정이 불필요하다. 다음의 [표 5-5]는 제안하는 인증 프로토콜의 안전성을 기존의 인증 기법들과 비교한 것이다.

[표 5-5] 인증 프로토콜의 안전성 비교

기법 \ 공격	위치추적	스푸핑	재전송	전송 방해
해쉬 락	약함	약함	약함	약함
랜덤 해쉬락	강함	약함	약함	강함
해쉬 체인	강함	강함	약함	강함
변형 ID	강함	약함	강함	약함
제안 기법	강함	강함	강함	강함

5.4 인증 프로토콜의 효율성 평가

다중 응용 객체의 접근을 지원하고, 각 응용 객체별로 보안 레벨에 따라 다르게 동작하는 인증 프로토콜의 효율성을 평가하기 위해 먼저, 기

존 인증 프로토콜과의 연산량을 비교하였다. 또한, 인증 프로토콜의 효율성을 평가하기 위한 다른 하나의 방법으로, 시뮬레이터를 구현하여 인증 절차를 수행하는 데 걸리는 시간과 평균 에러율을 측정하였다. 시뮬레이션에서는 본 논문에서 제안한 저수준 인증 프로토콜과 고수준 프로토콜에 해당하는 프로그램을 각각 구현하여 프로토콜 별로 인증 절차를 수행하고 해당하는 값을 측정하였다. 제안한 인증 프로토콜과의 비교를 위해, 실험에 사용된 RFID 시스템의 기본 프로토콜로 내장된 ISO15693 프로토콜에 따라 수행한 결과 값을 측정하고 비교해 보았다.

가. 연산량 비교

본 논문에서 제안한 RFID 인증 프로토콜에서 태그가 필요로 하는 연산은 리더에서 생성한 랜덤 값에 대한 임의의 비트들을 추출하는 것이므로, 기존의 인증 기법들에서 사용하는 해쉬 함수 수행이나 랜덤 값 생성과 같은 연산보다 매우 적은 비용으로 수행될 수 있다. 따라서, 제안한 인증 프로토콜은 저사양의 태그에서도 효율적으로 동작할 수 있다는 장점을 가진다. 또한, 제안한 인증 프로토콜에서 RFID 서버는 시스템 설정 단계에서만 SEED 변형 알고리즘에 의한 암호화 연산을 필요로 한다. 인증 세션 동안에는 데이터베이스 검색을 위해 요구되는 (n / 2) 회[1]의 비트 마스킹(Bit Masking) 연산만을 수행함으로써, 기존의 인증 프로토콜에서 수행하던 해쉬 함수 연산 등에 비해 서버에서의 연산량 또한 효율적으로 개선시킬 수 있다. [표 5-6]은 기존의 인증 기법들에서 인증 과정 중 필요로 하는 연산

[1] n = 태그의 평균 개수

량과 본 논문에서 제안한 인증 프로토콜의 연산량을 비교하고 있다. 연산량에 대한 비교 결과, 제안한 인증 프로토콜에서 인증 과정을 수행하는 동안 필요로 하는 연산량은 기존 기법들에 비해 크지 않아 더 효율적임을 알 수 있다.

[표 5-6] 인증 프로토콜의 연산량 비교

구성요소 / 기법	태그	리더	서버
해쉬 락	hashing=1	–	–
랜덤 해쉬 락	randomizing=1 hashing=1	hashing=(n/2)	–
해쉬 체인	hashing=2	–	hashing=(n/2)*i (i : update count)
변형 ID	hashing=3	randomizing=1	randomizing=1 hashing=3
제안 기법	bit masking=2	randomizing=1	bit masking=n/2

나. 연산 시간 및 평균 에러율 실험

인증 프로토콜의 효율성을 평가하기 위한 기준으로 인증 절차를 수행하는 시간과 인증 과정 중에 발생하는 에러율을 측정하였다. 이를 위하여 다중 객체 접근 구조의 시뮬레이터를 구현하여 실험하였으며, 시뮬레이터 구현을 위한 개발 환경은 [표 5-7]과 같다.

실험에 사용된 태그의 메모리에는 각 태그마다 다수 개의 응용 객체 정보를 저장하도록 하고, 이러한 태그의 제어를 위해 RFID 시스템 제조사에 의해 제공되는 MxPanel이라는 리더-태그 간 통신 프로그램을 시뮬레이터와 연결하여 태그 메모리 블록을 읽어들인다. 이러한 방식을 통해 기존의 단일 객체 구조의 RFID 시스템을 다중 응용 객체에 대해 동작할 수 있게 수정하였다.

[표 5-7] 시뮬레이션을 위한 개발 환경

H/W	CPU	AMD Athlon 64 X2 Dual Core 3800 + 2.0 GHz
	RAM	2 GB
S/W	O.S	Windows XP
	Language	Visual Basic
	Database	Oracle 10G

각 태그의 메모리에는 응용 객체 정보 외에도 암호화한 태그 식별 정보를 기록하는데, 서버의 데이터베이스에도 이와 동일한 데이터를 저장해야 한다. 태그 메모리와 서버의 데이터베이스에 저장되는 암호화한 태그 정보(*eID*)는 3장에서 설명한 SEED 변형 알고리즘을 이용하여 암호화하였다. 암호화 알고리즘의 수행 시간과 태그 메모리 및 데이터베이스 기록 과정은 초기화 작업으로 분류하여 연산 시간에서 제외하였다.

시뮬레이션을 통한 인증 프로토콜 수행 시간과 평균 에러율 측정은 다음과 같이 세 가지 방식의 프로토콜로 나누어 그 값을 측정하였다. 첫

번째 방식은, 실험을 위해 사용된 RFID 시스템의 기본 프로토콜로 설정되어 있는 ISO15693 통신 프로토콜을 사용하여 리더와 태그 간의 인증을 수행 하는 방법이고, 이 방법을 기본 프로토콜(Basic Protocol)이라 정의한다. 두 번째 방식은 4장에서 제안한 저수준 보안 레벨에서의 인증 프로토콜 방식이며, 이를 저수준 프로토콜(Low-Level Protocol)이라 정의한다. 마지막으로 고주순 보안 레벨에 해당하는 인증 프로토콜을 사용하는 경우를 고수준 프로토콜(High-Level Protocol)로 정의한다.

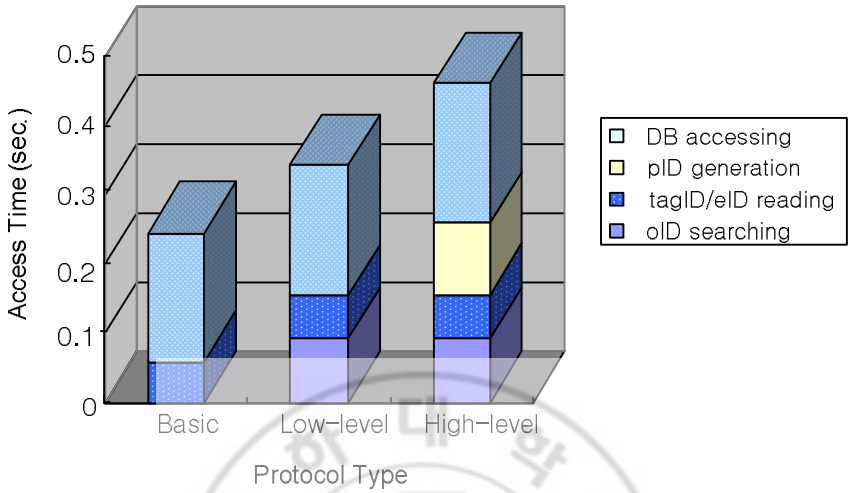
인증 절차에 소요된 시간을 측정한 결과는 [표 5-8]과 같다. 인증 과정의 각 단계에서 소요되는 시간을 100회씩 반복 수행하여 측정하고, 각 프로토콜별로 전체 인증 시간을 구하여 [그림 5-4]와 같이 비교하였다.

[표 5-8] 인증 시간 측정 (sec.)

수행 동작	프로토콜		
	기본	저수준	고수준
oID searching[1]	-	0.092	0.092
tagID/eID reading[2]	0.059	0.063	0.064
pID generation	-	-	0.104
DB accessing	0.186	0.188	0.202
Total	0.245	0.343	0.462

[1] 해당하는 RFID 응용 객체에 대한 인덱스 테이블 검색 과정을 의미한다.

[2] 기본 프로토콜의 경우 tagID 를 읽는 시간을 의미하고, 저수준과 고수준 프로토콜에서는 eID 를 읽는 시간을 의미한다.



[그림 5-4] 인증 단계별 연산 시간

평균 에러율에 대한 실험은 인증 프로토콜을 수행하는 동안 태그에서 발생하는 여러 종류의 에러에 대해 발생 횟수를 측정하였다. 기본 프로토콜, 저수준 프로토콜, 고수준 프로토콜의 세 경우에 대해 각각 100회, 200회, 500회, 1000회의 반복 수행하고, 각 프로토콜에서 발생한 오류의 종류에 따라 발생 횟수를 집계하여 평균 에러율을 구하였다. 평균 에러율의 값은 (총 에러 발생 횟수 / 실험 회수)를 백분율로 환산한 것이다.

인증 프로토콜을 수행할 때 태그에서 발생 가능한 오류에는 *tagID* 혹은 *eID* 읽기 오류, *oID*에 따른 인덱스 테이블 검색 오류 및 *pID* 생성 오류가 있다. 다음의 [표 5-9]에는 이러한 기준으로 평균 에러율에 대해 실험한 결과를 프로토콜별로 나타내었으며, 세 가지 프로토콜의 평균 에러율

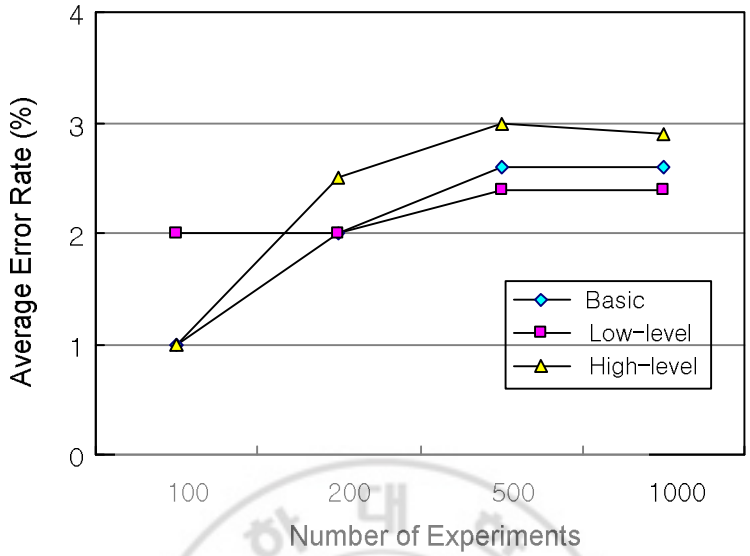
을 서로 비교하기 위해 [그림 5-5]와 같이 그래프로 표시하였다.

[표 5-9] 인증 프로토콜별 평균 에러율 측정 (number)

에러 유형	기본 프로토콜			
	100 회	200 회	500 회	1000 회
tagID reading error	1	4	13	26
Average error rate(%)	1.0	2.0	2.6	2.6

에러 유형	저수준 프로토콜			
	100 회	200 회	500 회	1000 회
Index table error	0	1	7	14
eID reading error	2	3	5	10
Average error rate(%)	2.0	2.0	2.4	2.4

에러 유형	고수준 프로토콜			
	100 회	200 회	500 회	1000 회
Index table error	1	3	8	15
eID reading error	0	1	4	9
pID generation error	0	1	3	5
Average error rate(%)	1.0	2.5	3.0	2.9



[그림 5-5] 인증 프로토콜별 태그에서의 평균 에러율

[그림 5-5]를 보면 평균 에러율은 어떤 프로토콜의 경우에도 비슷한 수치를 보여준다. 앞서의 인증 시간 측정값과 평균 에러율 실험을 통해 여러 가지 공격에 대해 안전성을 보장하기 위한 고수준 보안 레벨의 인증 프로토콜을 사용하였을 경우에도 인증 과정에 소요되는 시간이 비교적 높지 않고, 이 프로토콜이 오류 발생 측면에서도 비교적 안정적임을 알 수 있다. 이러한 사실은 실생활에 적용되는 RFID 시스템에서 리더를 통한 서버와 태그 간의 통신에 본 논문에서 제안한 다중 객체를 지원하는 인증 프로토콜을 적용하였을 경우, 시스템의 성능과 여러 종류의 공격에 대해 안전성을 보장할 수 있다는 것을 보여준다.

제 6 장 결론 및 향후 연구과제

6.1 결론

본 논문에서 제안한 RFID 시스템의 태그 구조는 다양한 목적에 적용 가능한 다중 객체를 지원할 수 있는 방식으로 구성되며, 이는 RFID 시스템의 기능과 목적에 따라 사용자가 별도의 태그들을 소유하지 않고 하나의 태그 내에 인증 가능한 정보를 통합 관리할 수 있다는 장점을 내포하고 있다. 즉, 하나의 태그 내에 여러 개의 ID들을 가짐으로써 사용자의 입장에서 여러 가지 편리한 기능을 제공할 수 있게 된다. 이러한 다중 객체를 지원하는 태그와 리더, 그리고 서버간의 통신 선로에서 발생할 수 있는 여러 가지 공격들로부터 사용자 프라이버시를 보호하기 위한 인증 프로토콜을 제안하였다. 또한, 기존의 암호화 방식인 SEED 알고리즘을 수정하여 RFID 환경에 맞게 경량화시킨 암호화 알고리즘을 설계하고, 각 태그에 대한 식별 정보의 암호화에 SEED 변형 알고리즘을 적용하였다.

현재 국내 표준으로 자리잡고 있는 SEED 암호화 알고리즘은 암호화에 소요되는 연산량과 속도 문제로 인해 RFID 시스템에 그대로 적용할 수 없다. 따라서, 본 논문에서는 기존의 SEED 알고리즘을 변형하여 키 크기와 라운드 횟수를 감소시켜 RFID 태그의 식별 정보를 암호화하는데 사용하였다. 암호화 과정에서의 속도를 향상시키기 위한 방법으로 암호화 과정의 입력 값으로 사용되는 사용자 키 길이를 변경시킴으로써 암호화 과정

중 입력에 필요한 블록의 개수를 감소시켜, 전체 암호화 단계에 필요한 시간을 감소시키는 효과를 얻을 수 있었다. 또 다른 방법으로, 기존의 SEED 알고리즘에서 수행하는 16라운드의 암호화 동작을 8라운드로 동작하도록 수정하였고, 실험 결과 기존의 방법보다 성능이 향상되었음을 확인할 수 있었다.

기존의 암호화 알고리즘에 비해 향상된 결과를 보여주는 SEED 변형 알고리즘을 적용한다하더라도 암호화된 ID를 생성하기 위해서는 암호화 과정에 적지 않은 비용이 요구된다. 따라서, 본 논문에서는 RFID 시스템의 인증 프로토콜이 수행되기 전인 시스템 설정 단계에서 태그 식별 정보에 대한 암호화 동작을 수행하도록 하여 효율성을 높일 수 있도록 설계하였다. 그리고, 매 세션마다 이루어지는 인증 과정 중에는 랜덤 값에 의한 비트 추출 방식을 사용함으로써 태그에서의 연산 시간을 감소시키도록 설계하였다. 특히, 본 논문에서 제안한 인증 프로토콜은 여러 종류의 RFID 응용들이 요구하는 서로 다른 수준의 보안 레벨에 따라 각기 다른 인증 절차를 수행하도록 함으로써, 저사양의 태그에서도 좀더 효율적인 동작이 가능하도록 하였다.

제안한 인증 프로토콜의 효율성과 안전성을 평가하기 위하여 인증 과정에 소요되는 연산 시간과 평균 에러율에 대한 시뮬레이터를 구현하여 실험하였다. 실험을 수행한 결과, 본 논문에서 제안하고 있는 인증 프로토콜은 안정적이면서도 현재의 RFID 응용 환경에 널리 적용 가능한 수준임을 확인할 수 있었다.

6.2 향후 연구과제

다중 객체를 지원하는 RFID 시스템은 향후 도래할 유비쿼터스 환경에서 사용자의 프라이버시를 보호하기 위한 목적으로 연구되었다. 본 연구와 관련하여 다음과 같은 추가적인 연구들이 필요하다.

본 논문에서 제안한 K-SEED 및 R-SEED 알고리즘을 포함한 암호화 방식은 기존의 SEED 암호화 알고리즘을 RFID 시스템에 적용해 보려는 목적으로 연구되었으며, RFID 시스템에 적용 가능하도록 암호화 속도의 성능 향상에 그 중요성을 두었다. 제안한 두 알고리즘을 구현하여 실험한 결과 기존의 SEED 알고리즘보다 향상된 성능을 보여주고 있다. 그러나, 이러한 두 가지 변형 알고리즘에 대해서는 암호화된 결과 값에 대해 여러 가지 암호 해독법 공격들로부터 안전한지에 대한 안전성 분석이 필요하다. 향후 연구에서는 제안한 R-SEED 알고리즘 및 K-SEED 알고리즘을 안전성 측면에서의 성능을 실험하고 분석해 보아야 할 것이며, 안전성을 보장하면서도 성능을 향상시킬 수 있는 방법에 대한 연구 또한 추가되어야 할 것이다.

본 연구에서는 또한, 다중 객체를 지원하는 RFID 시스템의 태그 구조와 인증 프로토콜을 제안하였고, 인증 프로토콜의 안전성과 효율성을 분석하였다. 특히, 효율성에 대한 분석은 시뮬레이터를 이용한 실험에 의해 수행되었는데, 측정된 실험 결과 값은 실제 RFID 환경에 적용하는 경우 다소 차이가 발생할 가능성이 있다. 구현된 시뮬레이터는 현재 일반적으로 사용되는 단일 객체 방식의 리더에 연결되어 태그의 인증 절차를 수행하므

로, 여러 RFID 응용 객체의 리더들이 동시에 하나의 태그와 통신하는 경우 인증 과정 중 필요로 하는 메시지의 양이 많아지게 됨으로써 인증 프로토콜의 효율성에 영향을 미치게 될 것으로 보여진다. 따라서, 향후 태그를 비롯한 RFID 시스템의 다른 구성 요소들에 대해서도 다중 객체 접근을 지원하는 소프트웨어 모듈 개발에 대한 추가적인 연구가 필요할 것으로 생각된다.

참 고 문 헌

[AID03] AutoID Center, “900MHz class 0 radio frequency (RF) identification tag specification. Draft,” Internet Draft, MIT AutoID Center, Mar. 2003.

[Aign05] M. Aigner, and M. Feldhofer, “Secure Symmetric Authentication for RFID Tags,” *Telecommunication and Mobile Computing*, 2005.

[Avoi04] G. Avoine, and P. Oechslin, “RFID Traceability: A Multilayer Problem,” *EPFL*, Oct. 2004.

[Elbi01] A. J. Elbirt, W. Yip, B. Chetwynd, and C. Parr, “An FPGA-based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists,” *IEEE Transactions on Very Large Scale Integration System*, vol.9, no.4, 2001.

[EPC03] EPCGlobal Inc., “Draft protocol specification for a 900 MHz class 0 radio frequency identification tag,” http://www.epcglobalinc.org, Feb. 2003.

[Feld04a] M. Feldhofer, “An Authentication Protocol in a Security Layer for RFID Smart Tags,” *Proc. MELECON 2004*, pp.759-762, May. 2004.

[Feld04b] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, “Strong authentication for RFID systems using the AES algorithm,” *Workshop on Cryptographic Hardware and Embedded Systems*, LNCS 3156, pp.357-370, Aug. 2004.

[Fink99] K. Finkenzeller, “RFID Handbook,” John Wiley & Sons, 1999.

[Goll04] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, “Universal re-encryption for mixnets,” *RSA Conference Cryptographics’ Track ’04*, pp.163-178, 2004.

[Henr04] D. Henrici, and P. MÜller, “Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers,” *PerSec’04*, pp.149-153, Mar. 2004.

[Her01] P. Hernandez, J. D. Sandoval, F. Puente, and F. Perez, “Mathematical Model for a Multiread Anticollision Protocol,” *Proc. 2001 IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, pp.647-650, Aug. 2001.

[ISO04] ISO/IEC/JTC 1/SC 31, “Information technology—Radio frequency identification for item management—Unique identification for RF tags,” ISO/IEC 15963, 2004.

[Juel03a] A. Juels, and R. Pappu, “Squealing euros: Privacy Protetction in RFID-Enabled Banknotes,” *FC’03, LNCS 2742*, pp.103-121, 2003.

[Juel03b] A. Juels, R. Rivest, and M. Szydlo, “The blocker tag: Selective blocking of RFID tags for consumer privacy,” *ACM CCS’03*, pp.103-111, Oct. 2003.

[KISA01] 한국정보보호진흥원, “AES, NESSIE, CBYPTREC 후보 블록암호알고리즘의 안전성 비교 분석,” 2001년 11월.

[KISA03] 한국정보보호진흥원, “128비트 블록 암호알고리즘(SEED) 개발 및 분석 보고서,” 2003년 10월.

[KISA99] 한국정보보호진흥원, “SEED 알고리즘 상세 명세서,” 1999년.

[Ohku03] M. Ohkubo, K. Suzuki, and S. Kinoshita, “Cryptographic Approach to “Privacy-Friendly” Tags,” *RFID Privacy Workshop*, 2003.

[Park05] H. Y. Park, B. Y. Sohn, and Y. T. Shin, “Safe Authentication Method for Security Communication in Ubiquitous Environment,” *IICSA2005 LNCS*, pp. 442-448, May. 2005.

[Sarma01] S. E. Sarma, “Towards the fivecent tag,” MIT AutoID Center,

2002.

[Sarma02] S. E. Sarma, S. A. Weis, and D. W. Engels, “RFID Systems and Security and Privacy Implications,” *Workshop on Cryptographic Hardware and Embedded Systems*, 2002.

[Sarma03] S. E. Sarma, and D. W. Engels, “On the Future of RFID Tags and Protocols,” Technical Report, MIT AutoID Center, 2003.

[Weis03] S. A. Weis, “Security and Privacy in Radio-Frequency Identification Devices,” M.S. Thesis, MIT, May. 2003.

[Weis04] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, “Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems,” *Security in Pervasive Computing*, *LNCS 2802*, pp.201-212, 2004.

[Yang05] J. Yang, J. Park, H. Lee, K. Ren, and K. Kim, “Mutual Authentication Protocol for Low-Cost RFID,” *Encrypt Workshop*, Jul. 2005.

[Yeo05] S. S. Yeo and S. K. Kim, “Scalable and Flexible Privacy Protection Scheme for RFID System,” *Proc. the 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks(ESAS 2005)*, *LNCS 3813*, pp.153-163, Jul. 2005.

[강수영07] 강수영, 이임영, “난수를 이용하여 동기화를 제공하는 RFID 프라이버시 보호 기법에 관한 연구,” *멀티미디어학회 논문지*, 제10권 제5호, pp.623-630, 2007년 5월.

[김말희06] 김말희, 이용준, “모바일 RFID 서비스를 위한 QoS 및 보안 모델,” *한국통신학회논문지*, vol.31, no.5C, pp.562-567, 2006년 5월.

[김상태03] 김상태, “RFID 기술개요 및 국내외 동향분석,” 전자부품연구원 전자정보처리센터, 2003년 8월.

[남택용05] 남택용, 장종수, 손승원, “유비쿼터스 환경에서의 개인정보 보호 기술”, *전자통신동향분석*, 제20권 제1호, pp.54-62, 2005년 2월.

[변상기04] 변상기, “RFID Tag 기술,” *한국전자파학회지*, 제15권 제2호, pp.32-43, 2004년 4월.

[변영기06] 변영기, RFID 시스템에 적합한 암호화 알고리즘 연구, 상명대학교대학원 석사학위논문, 2006년.

[송석현04] 송석현, “RFID 서비스 전망,” *한국전자파학회지*, 제15권 제2호, pp.80-95, 2004년 4월.

[양형규05] 양형규, 안영화, “유비쿼터스 컴퓨팅 환경에 적합한 RFID 인증 프로토콜에 관한 연구”, *전자공학회논문지*, 제42권 CI 제1호, pp.45-50, 2005년 1월.

[여상수06] 여상수, 김순석, 김성권, “안전한 RFID 프라이버시 보호 프로토콜을 위한 백엔드 서버의 태그 판별 시간 절감 기법,” *정보보호학회논문지*, 제16권 제4호, pp.13-25, 2006년 8월.

[오세원05] 오세원, “RF 태그 식별을 위한 번호화 방안”, 한국전자통신연구원, 2005년 5월.

[유성호04] 유성호, 김기현, 황용호, 이필중, “상태기반 RFID 인증 프로토콜,” *정보보호학회논문지*, 제14권 제6호, pp.57-68, 2004년 12월.

[윤현철05] 윤현철, 김재권, 박주용, 범진욱, “Passive RFID Sensor Tag,” *한국전자파학회지*, 제16권 제3호, pp.16-25, 2005년 7월.

[이근우05] 이근우, 오동규, 곽진, 오수현, 김승주, 원동호, “분산 데이터베이스 환경에 적합한 Challenge-Response 기반의 안전한 RFID 인증 프로토콜”, *정보처리학회논문지 C*, 제12-C권 제3호, pp.310-316, 2005년 6월.

[이근호03] 이근호, “무선식별(RFID) 기술,” *TTA저널*, 제89호, pp.124-129, 2003년 10월.

[이승구05] 이승구, 여상수, 조정식, 김성권, “RFID 시스템에서 안전하고 효율적인 프라이버시 보호 기법,” *한국컴퓨터종합학술대회 2005 논문집*, vol.32, no.1(A), 2005년.

[정민화04] 정민화, 전자태그(RFID) 글로벌 표준과 산업적용 전망, 산업자원부, 2004년 6월.

[최병윤01] 최병윤, “AES Rijndael 암호 프로세서의 설계,” *한국통신학회논문지*, vol.26, no.10B, pp.1491-1500, 2001년.

[최재귀04] 최재귀, 박지환, “효율적인 식별 기능을 가진 위조 불가 RFID Tag 가변 ID 방식,” *정보처리학회논문지* C, 제11-C권 제4호, pp.448-454, 2004년 8월.

[최호승05] 최호승, 김재현, “RFID 시스템에서의 태그 인식 알고리즘 성능 분석,” *전자공학회논문지*, 제42권 TC 제5호, pp.47-54, 2005년 5월.

[표철식04] 표철식, 채종석, 김창주, “RFID 시스템 기술”, *한국전자파학회지*, 제15권 제2호, pp.21-31, 2004년 4월.

[황영주04] 황영주, 이수미, 이동훈, 임종인, “유비쿼터스 환경의 Low-Cost RFID 인증프로토콜,” *한국정보보호학회 하계학술대회*, pp.109-114, 2004년.

부 록 - 약어표

Abbreviation	Meaning	Chapter
D	Decryption Speed Rate	5
E	Encryption Speed Rate	5
eID	encrypted ID, 암호화된 식별자	3, 4, 5
LST	Last Successful Transaction	2
Nb	AES 알고리즘의 블록 길이	2
Nk	AES 알고리즘의 키 길이	2
Nr	AES 알고리즘의 라운드 수	2
oID	object ID, RFID 응용 객체에 대한 식별자	3, 4, 5
pID	partial ID, 태그에서 생성하는 가변적인 식별자	3, 4, 5
S	RFID Server	4
R	RFID Reader	4
RNo	Random Number	3, 4, 5
T	RFID Tag	4
TID	Transaction ID	2

감사의 글

학부시절부터 대학원까지 20여년이란 오랜 시간을 인하대학교의 캠퍼스를 보아왔습니다. 주변은 많이도 변했지만, 여전히 화사한 봄날과 매서운 칼바람의 겨울날도 정들어 뒤로 하기 아쉽습니다. 그리고, 그 오랜 시간을 변함없이 지도해 주신 이균하 교수님께 진심으로 감사드립니다.

바쁘신 가운데에서도 논문 심사를 맡아주시고 배려해주신 조근식 교수님과 권구인 교수님, 세심하게 지도해 주신 차영환 교수님과 권오형 교수님께 감사드리고, 인하대 정보공학과의 모든 교수님들께도 감사의 말씀드리고 싶습니다.

결실을 맺기까지 거쳐야 했던 많은 과정이 있었지만, 논문의 실험 과정을 적극적으로 지원해 주시고 여러 도움을 주신 장윤석 교수님께 감사드리고, 대진대학교 컴퓨터공학과의 모든 교수님들과 류욱재 선생님, 그리고 멀리 계신 고훈 교수님께도 감사드리고 싶습니다.

비록 지금은 그 이름도 찾아볼 수 없으나, 많은 사람들이 밤잠 설쳐가며 연구하며, 즐겁고 유쾌한 추억을 만들었던 인공지능 연구실의 모든 선후배님들과 마지막 졸업생으로서의 기쁨을 함께 나누고 싶습니다.

그리고, 항상 곁에서 격려해 주신 부모님과 모든 가족들에게 고마움을 전하고 싶습니다. 특히, 저의 동반자이자 선배로서 시작부터 마지막까지 도움을 아끼지 않은 정종진 교수님과 생각만으로도 행복해지는 마음의 보물 은상, 은수에게 무한한 사랑을 전합니다.