情 報 保 護 學 會
영남지부 2008년도 정보보호 학술논문 발표회

비트 스크램블을 통해 개선된 RFIDsec07의 상호 인증 프로토콜

신 승 목*, 이 현 록*, Divyan M. Konidala*, 김 광 조*

요 약

2006년 ISO 국제 표준화 회의에서 'EPC Class1 Gen2'[1](이하 Gen2) 태그 방식이 RFID 세계 표준으로 확정되면서 그동안 여러 가지 통신방식이 혼재되어 사용됨으로써 혼란이 불가피 했던 RFID 산업계가 상호 호환성을 가질 수 있게 되었다. 그러나 Gen2 표준에서 사용되는 저가형 태그의 한정된 자원으로는 현재까지 제안된 보안 장치들에 사용되는 해쉬함수나 공개키 알고리즘 등의 암호학적 장치들을 사용할 수 없다. 또한 Gen2 표준에서는 태그-리더 간의 상호 인증을 제공하고 있지 않아, 중간자 공격 및 서비스 거부 공격이 가능하고, 태그 복제 공격 또한 가능하여 RFID 태그가 부착된 상품의 무분별한 복제로 인한 막대한 경제적 피해를 입을 수 있다. 이에 Divyan등[3]은 Gen2 표준에 적용가능하고 안전한 통신을 위한 리더-태그-생산자간의 상호 인증 프로토콜을 제안하였으나, 해당 프로토콜의 취약점을 제시한 논문이 나오면서 안전성에 문제가 제기되고 있다. 본 논문에서는 해당 프로토콜의 취약점에 대해 분석해보고 이를 극복할 수 있는 방안을 포함한 개선된 상호 인증 프로토콜을 제시하고 구현을 통한 검증을 실시하였다.

Ⅰ. 서 론

RFID(Radio Frequency Identification, 전파식별, 이하 RFID)는 초소형 반도체에 식별정보를 넣고 무선주파수를 이용해 이 칩을 지닌 물체나 동물, 사람 등을 판독·추적·관리할 수 있는 기술로, 현재 물류·유통뿐 아니라 전자지불·보안 등 다양한 분야에도 적용되고 있다. RFID 기술은 2차 세계대전 당시 레이더에서 발산되는 신호로 적과 아군을 식별할 목적으로 연합군에 의해서 처음으로 사용된 것으로 알려져 있다.

지금까지의 RFID 시스템은 국제 표준 부재로 인해 시스템 오류 및 업무 혼선 등의 문제가 야기되어왔다. 하지만 2006년 6월, ISO 국제 표준화 회의에서 EPC Class-1 Generation-2 UHF-RFID(이하 Gen2) 표준에 기초한 ISO 18000-6의 신규타입(타입 C)이 사실상의 RFID 국제 표준으로 확정됨으로써 RFID 시스템 시장이 일대 전기를 맞게 되었다.

그러나 Gen2 표준은 저가형 태그에 초점을 두어 RFID 시스템에서 필요한 보안 요구사항들을 충분히 만족시키지 못하고 있다. Kill 및 Access 패스워드[1]를 이용하여 태그 기능을 영구히 정지시키거나 태그의 특정 메모리 영역 쓰기 기능을 제한(Lock)하여 고객 프라이버시 보호 및 안전성이 향상되었다고 분석하고 있지만, 실제 Gen2 표준은 복제 공격·도청 공격·고객 프라이버시 침해[4]·중간자 공격·서비스 거부 공격 등의 공격에 매우 취약하다. 또한 Gen2 표준의 태그는 기존에 제안된 RFID 보안 기법에서 사용한 대칭키 암호 알고리즘 및 암호학적 해쉬함수가 지원되지 않아, Gen2 표준 기반의 RFID 시스템에 직접 사용하는 것은

* 한국정보통신대학교(Information and Communications University, ICU)
국제정보보호기술연구소(International Research Center for Information Security, IRIS)
({cabin15,tank,divyan,kkj}@icu.ac.kr)

불가능하다. 이에 Ari Juels[2]는 Gen2 표준에서 명시하고 있는 Kill 및 Access 패스워드를 이용한 태그-리더 간의 인증 기법을 제안하여 복제 공격을 방지하고자 하였다. 하지만 이 기법 역시 도청 공격 등에 취약하다는 문제점이 있다.

이와 같은 공격에 대하여 현재까지 다양한 리더-태그 상호 인증 프로토콜들이 제안되었다. 특히 RFID Security-07에서 Divyan등[3]이 발표한 "A Simple and Cost-effective RFID Tag-Reader Mutual Authentication Scheme"은 이미 저장하고 있는 비밀정보 APwd, KPwd와 난수 생성기, XOR연산만을 사용하여 실제 시스템에 적용시킬 수 있을 만한 효율적인 프로토콜이다. 하지만 특정 경우에 발생할 수 있는 취약점을 기술한 관련 논문들이 발표되면서 수정, 보완될 필요가 있다.

본 논문에서는 Divyan등이 제안한 방식의 취약점에 대해 알아보고 발견된 취약점을 보완할 수 있는 개선된 방식을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 Ⅱ장에서는 취약점을 지적한 관련 논문들에 대해 살펴보고, Ⅲ장에서는 제안된 프로토콜의 핵심인 PadGen함수의 수정과 추가적인 보안 장치의 삽입을 통해 좀 더 안전하고 효율적인 상호인증 프로토콜을 제시하고자 한다. Ⅳ장에서는 실제 프로토콜의 시뮬레이션을 통해 안전성 여부를 확인하고 Ⅴ장에서 본 논문의 결론을 내리도록 한다.

Ⅱ. 관련 연구

1. 리더-태그 간 상호 인증 프로토콜의 필요성

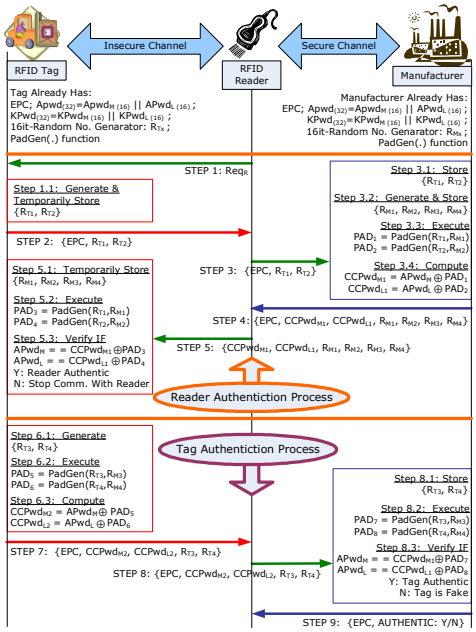
Gen2 표준에서는 특별한 태그-리더 간의 상호 인증을 제공하지 않고 있다. 이로 인해 중간자 공격 및 서비스 거부 공격이 가능하여 원활한 RFID 서비스 제공에 큰 지장을 주게 된다. 또한 복제 공격이 가능하여 RFID 태그가 부착된 상품의 무분별한 복제로 인한 막대한 경제적 피해를 입을 수 있다. 따라서 태그-리더 상호 개체 간의 유효성 검증을 위한 상호 인증이 필수적으로 제공되어야 한다.

리더-태그 간 상호 인증 프로토콜 연구에서 한 가지 더 고려해야할 사항은 프로토콜의 경량화이다. Gen2 태그의 특성상 저가 태그 사용에 따라 자원이 한정되어 있어 기존의 암호학적 함수들의 사용이 어려우므로 가벼운 연산들을 이용하여 보안을 달성하는 것 또한 중요한 고려 사항이다.

2. RFIDsec07에 제안된 상호 인증 프로토콜

Gen2 프로토콜 표준에서는 단순히 난수와 평문을 XOR 연산하는 단순하며 취약한 인증 프로토콜만이 제안되었다.

Divyan 등의 방식에서는 이러한 기존의 프로토콜에 PadGen이라는 함수를 추가하여 보안성을 높였다. PadGen함수는 패딩기법의 일종으로, 난수를 인자로 받아서 PAD값을 생성하는 함수이다. Pad와 APwd(Access Password)는 XOR 연산을 거쳐 cover-coded된다. [그림1]에서 프로토콜의 전개과정을 각 Step에 따라 간략하게 설명해본다.



[그림 1] Divyan등의 상호인증 프로토콜

Step 1~3. 태그의 난수 생성

: 리더의 Req_R 요청 메시지에 태그는 난수 $\{R_{T_1}, R_{T_2}\}$를 생성하여 EPC와 함께 리더로 전송하고, 리더는 다시 생산자(Manufacturer)로 이를 전송한다.

Step 4. Pad 생성 및 Cover-Coding APwd 생성

: 태그에서 받은 난수와 생산자에서 생성된 난수는

PadGen함수를 통과하여 Pad를 출력한다. 이 Pad와 생산자 DB에 저장된 해당 태그의 APwd를 XOR 연산하여 CCPwd가 생성되고, 이를 생산자에서 생성된 난수와 함께 리더에서 태그로 전송한다.

Step 5. 전송 받은 CCPwd값의 검증
: 생산자에서 받은 난수를 이용해 Step 4에서와 동일한 과정을 거쳐 Pad 생성 후, 태그에 저장하고 있는 APwd와의 XOR 연산을 통해 CCPwd를 생성한다. 리더가 올바른 생산자와 통신하여 Pad를 생성하였다면, 두 CCPwd는 같은 값을 가지게 되고 태그에 대한 리더의 인증과정이 완료된다.

Step 6~9. 리더에 대한 태그의 인증
: Step 6~9에서는 리더에 대한 태그의 인증이 이루어진다. 인증 방식은 Step 1~5에서 거쳤던 것과 동일하다.

3. Divyan 등의 기법에 대한 취약점 공격. [5]

Divyan 등이 제안한 기법의 취약점을 이용한 첫 번째 공격 방법[5]은 태그에서 생성된 $\{R_{T_1}, R_{T_2}\}$를 임의로 수정하여 APwd(Access Password)의 값을 알아내는 것이다. 공격자가 임의로 R_{T_1}의 값을 '0000'으로 만든다면 리더에서 계산 되어지는 PAD_1의 h_3와 h_4의 값을 0 혹은 F의 값으로 만들 수 있다.

[표 1] $CCPwd_M = APwe_M \oplus PAD_1$의 계산 결과

$[PAD_1]_{8\sim11}$	$[PAD_1]_{12\sim15}$	$[CCPwd_M]_{8\sim11}$	$[CCPwd_M]_{12\sim15}$
0000	0000	$[APwd_M]_{8\sim11}$	$[APwd_M]_{12\sim15}$
0000	1111	$[APwd_M]_{8\sim11}$	$\overline{[APwd_M]_{12\sim15}}$
1111	0000	$\overline{[APwd_M]_{8\sim11}}$	$[APwd_M]_{12\sim15}$
1111	1111	$\overline{[APwd_M]_{8\sim11}}$	$\overline{[APwd_M]_{12\sim15}}$

[표 1]에서 PAD_1의 마지막 4비트가 0 혹은 F임을 알 수 있으므로 $CCPwd_M = APwd_M \oplus PAD_1$식에서

$$[PAD_1]_{8-11} = 0 \rightarrow [CCPwd_M]_{8-11} = [APwd_M]_{8-11}$$
$$[PAD_1]_{8-11} = F \rightarrow [CCPwd_M]_{8-11} = \overline{[APwd_M]_{8-11}}$$
$$[PAD_1]_{12-15} = 0 \rightarrow [CCPwd_M]_{12-15} = [APwd_M]_{12-15}$$
$$[PAD_1]_{12-15} = F \rightarrow [CCPwd_M]_{12-15} = \overline{[APwd_M]_{12-15}}$$

결과적으로 75%의 확률로 4bit의 값을 얻을 수 있다. R_{T_2} 또한 '0000'으로 놓는다면 동일한 방법으로 공격이 가능하다.

Ⅲ. 제안 방식

1. 프로토콜 수정을 통한 APwd 유추 공격 방어

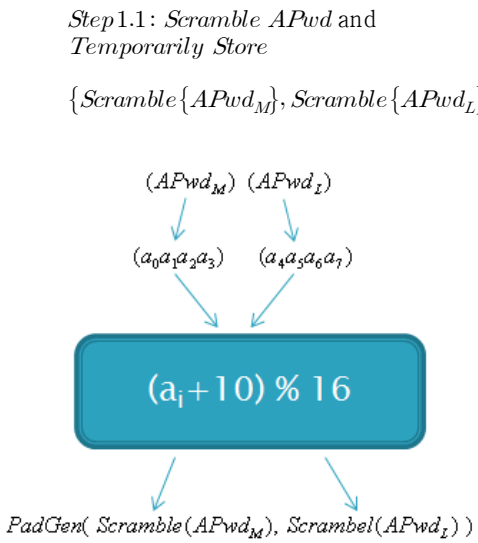
앞선, [5]의 공격은 XOR 연산의 특성을 이용한 것으로 PadGen함수의 비트 선택이 R_{T_1}의 비트 값에 의해 이루어진다는 것을 이용한 방법이다. R_{T_1}이 '0000'이라면 Pad값이 0 또는 F로 유추가능하게 되고 APwd의 특정 장소에 위치한 비트 값도 알 수 있게 된다. 이에 대해 [3]은 DES 알고리즘의 Weak Key와 같은 개념으로 R_{T_1}의 '0000'대입을 사전에 차단하는 방법을 간략히 언급한 바가 있다. 그러나 앞서 제시된 '0000' 등의 사전 설정된 목록 외 알려지지 않은 Weak key가 존재하는 경우 공격이 발생할 잠재적인 가능성이 존재한다.

이에 본 논문에서는 Divyan 등이 제시한 Weak key 필터링과 함께 사용될 수 있는 대안을 제시하였다. 기존 방식에서는 태그에서 생성되는 난수를 PadGen함수의 인자로 사용하였지만 제안 방식에서는 APwd에 비트 스크램블 과정을 더하여 공격자가 원하는 비트 값을 얻어 내지 못하도록 기존 방식을 수정하였다.

R_{T_1}이 '0000'인 경우 PadGen함수 내에서 비트의 영향은 8비트까지 전파되게 된다. 그러므로 R_{T_1}값이 Weak Key의 값을 가지지 않도록 기존 방식을 수정한다면 본래의 목적인 패딩을 수행할 수 있게 된다. 제안 방식은 다음과 같다.

기존 방식에서 Step 1 과정은 리더의 요청에 따라 태그가 R_{T_1}과 R_{T_2} 두 난수를 생성하는 과정이다. 제안 방식에서는 따로 난수를 생성하지 않고 APwd값의 MSB와 LSB을 비트 스크램블한 값을 R_{T_1}, R_{T_2} 값으로 입력받는다. 따라서 비트 스크램블 과정을 거친

R_{T_1}, R_{T_2}는 Weak Key를 가질 가능성이 없다. 〔그림 2〕는 비트 스크램블 과정을 보여준다.



〔그림 2〕 비트 스크램블 함수의 과정

Ⅳ. 구현 및 분석

1. 안전성 분석

1.1 태그의 Access Password 노출 문제

Divyan 등의 원래 프로토콜과 달리, 제안 방식은 태그의 APwd를 cover-code하기 위한 패드를 생성하는 과정에서 암호화되지 않은 난수를 사용하지 않는다. 대신에, 그 난수들은 태그의 APwd와 KPwd와 연계되어 Pad를 생성하는데 사용된다. 이 Pad들은 태그와 생산자에게만 알려져 있다. 그러한 Pad들을 사용하여 APwd를 cover-code함으로써 태그의 실제 APwd에 대한 보안성을 제공한다.

1.2. 재전송(Replay) 공격 문제

Pad를 계산하기 위해서, 우리는 태그와 생산자로부터 생성된 각각 두 개의 난수를 사용한다. 따라서 특정 세션에서의 재전송 공격은 공격자에게 어떠한 이점도 주지 못한다. 왜냐하면 매번 다른 세션마다 유일한 난수가 사용되기 때문에 계산되어지는 Pad는 항상 유일하다.

2. 성능 분석

저가형 Gen2 태그의 특성상 무거운 암호학적 장치를 사용하는 것은 불가능하다. 본 논문에서 개선한 프로토콜은 태그에 이미 저장되어있는 APwd, KPwd를 그대로 사용하고 랜덤넘버 생성기, XOR 연산만을 이용하여 태그의 복제 공격에 대하여 인증을 수행하고 올바른 태그만을 인증할 수 있게 해준다.

그러므로 제안 방식은 Divyan 등의 상호 인증프로토콜에서 보여준 효율성을 유지하면서 취약점으로 지적된 부분들을 저가의 연산을 통해 개선했다고 말할 수 있다. 〔표 2〕는 기존 방식과 제안 방식의 안전성 및 효율성을 비교하였다.

〔표 2〕 비교 분석

	기존 방식	제안 방식
R_{T_1}수정 공격[6]	○	×
연산량	효율적	효율적
사용된 연산	XOR, 난수생성기	XOR, 난수생성기

○:가능 ×:불가능

3. 구현 결과

본 논문에서 구현한 상호인증 프로토콜은 PC를 기반으로 구현되었으며 구현 환경은 다음과 같다.

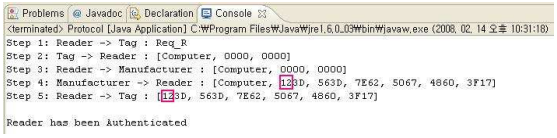
- CPU : Intel Pentium Dual Core 3.0GHz
- RAM : 1GByte
- OS : Windows XP. Service Pack 2
- Language : JDK 1.5

구현을 통한 프로토콜 시뮬레이션에서 임의적으로 랜덤넘버를 수정하여 공격한 결과 APwd와 KPwd가 패딩과정을 거치고도 CCPwd에 그대로 반영되는 결과가 나타났다. 그러나 비트 스크램블 후에는 그러한 현상이 사라진 것을 볼 수 있었다. 〔그림 3〕는 프로토콜의 스크램블 함수를 구현한 코드를 보여준다.

```
// APwd를 입력으로 받아 Scramble 과정을 거쳐 난수를 생성한다.
public static String scramble(String hex) {

    StringBuffer sb = new StringBuffer();

    for(int i = 0;i < hex.length();i++) {
        int index = ((Character.digit(hex.charAt(i), 16) + 10) % 16);
        sb.append(Integer.toHexString(index));

    }
    return sb.toString().toUpperCase();

}
}
```

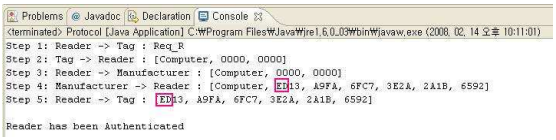
〔그림 3〕 스크램블 함수 구현 코드

〔그림 4〕의 시뮬레이션 결과에서 기존 프로토콜을 그대로 사용하고, APwd를 특정 값 '12345678'로 두어 결과를 출력했을 때, APwd의 초기 8비트가 그대로 노출되는 것을 볼 수 있다.



〔그림 4〕 기존 방식의 결과 출력화면

그러나 〔그림 5〕에서 제안 방식에 의해 출력된 결과에는 기존 APwd값이 노출되지 않음을 볼 수 있다.



〔그림 5〕 제안 방식의 결과 출력화면

V. 결 론

본 논문을 통해서 Gen2 태그 표준의 상호 인증 프로토콜과 그 취약성, 개선방안에 대해서 알아보았다. 저가형 태그라는 제한 때문에 강한 보안 함수들을 사용하지 못하는 Gen2의 특성 상 완벽한 보안을 제공하는 일은 매우 어려운 일이다. 그러나 가까운 미래에 RFID 기술이 우리 생활에 완전히 접목 되는 가운데 태그에 대한 복제, 위조 등이 이루어진다면 우리 삶뿐만 아니라 경제 전반적으로도 큰 파장을 불러 올 수 있다. 그러므로 Gen2 표준을 만족하면서도 현실세계에서 사용될 수 있을 만한 경량화된 인증 프로토콜의 개발이 시급하다.

본 논문에서는 RFIDsec07에서 발표된 Divyan 등의 상호 인증 프로토콜에 대해 알아보고 또한 취약점을 지적한 논문〔5〕에 대해서 살펴보았다. 이를 통해 Divyan 등의 방식에 도청 위협과 XOR 연산의 특성에 대한 취약점이 남아있다는 것을 알 수 있었다.

이에 〔5〕에서 발견된 취약점에 대해서 난수 생성과정에서 APwd를 비트 스크램블하는 과정 추가를 통해 Weak Key의 생성을 막아 패딩 과정이 올바르게 수행될 수 있도록 해당 방식을 개선한 방식을 제안하였다.

향후에는 〔6〕에서 제안된 태그 도청을 통한 선형방정식 공격에 대해 그 타당성을 분석하고 그에 따른 개선 방안을 연구해볼 것이다.

참 고 문 헌

〔1〕 EPCglobal, *EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFIDProtocol for Communications at 860 MHz - 960 MHz Version 1.0.9*

〔2〕 A. Juels, "Strengthening EPC Tags Against Cloning", ACM Workshop on Wireless Security (WiSe), 2005 (To appear)

〔3〕 Divyan M. Konidala, Zeen Kim and Kwangjo Kim, "A Simple and Cost-effective RFID 태그-Reader Mutual Authentication Scheme", Pre-Proc. of International Conference of RFID Security 2007(RFIDsec 07), pp. 141-152, Jul, 11-13, 2007, Malaga, Spain.

〔4〕 Choi, D. H., Kim, T. S., and Kim, H. W., "Privacy protection for secure mobile RFID service", International Symposium on Wireless Pervasive Computing, 2006

〔5〕 윤주승, 김일희, 김희문, 이길주, 박용수, "RFID 상호 인증 프로토콜에 대한 취약성 분석", 2007년도 정보보호학회 동계학술대회, pp. 76-79, 2007. 12.1, 상명대학교, 서울.

〔6〕 구본욱, 한대완, 권대성, "RFIDsec07에 제안된 경량 상호인증 프로토콜의 안전성 분석", 2007년도 정보보호학회 동계학술대회, pp. 80-84, 2007. 12.1, 상명대학교, 서울.