J Korea Industr Inf Syst Res Volume 17 Number 6
http://dx.doi.org/10.9723/jksiis.2012.17.6.001

순열을 사용한 새로운 초경량 RFID 인증 프로토콜에 대한 보안 분석 및 개선†

(Cryptanalysis and Improvement of a New Ultralightweight RFID Authentication Protocol with Permutation)

전 일 수*, 윤 은 준**
(Il-Soo Jeon and Eun-Jun Yoon)

요 약 저가의 RFID 태그는 많은 응용에 이용되지만 매우 한정된 계산 및 저장 능력을 가지고 있기 때문에 다양한 보안공격에 견딜 수 있는 RFID 상호인증 프로토콜을 만들기가 쉽지 않다. 아주 최근에 Tian 등은 계산비용이 적은 XOR 연산, 회전연산, 그리고 순열 연산을 사용하여 다양한 보안 공격에 견딜 수 있는 저가의 RFID 태그를 위한 인증 프로토콜(RAPP)을 제안하였다. 본 연구에서는 RAPP가 비동기화 공격에 취약함을 보이고 아울러 그 취약점을 극복하여 개선된 RAPP를 제시한다.

핵심주제어 : 저가 RFID 태그, 인증, 초경량 인증 프로토콜, 순열

Abstract Low-cost RFID tags are used in many applications. However, since it has very limited power of computation and storage, it's not easy to make a RFID mutual authentication protocol which can resist from the various security attacks. Quite recently, Tian et al. proposed a new ultralightweight authentication protocol (RAPP) for low-cost RFID tags using the low computation cost operations; XOR, rotation, and permutation operations, which is able to resist from the various security attacks. In this paper, we show that RAPP is vulnerable to the de-synchronization attack and present an improved RAPP which overcomes the vulnerability of RAPP.

Key Words : Low-Cost RFID Tags, Authentication, Ultralightweight Authentication Protocol, Permutation

1. 서 론

RFID 시스템은 사물을 자동으로 식별하는 수단으로 뿐만 아니라 지불시스템이나 접근제어 시스템 등 다양한 응용에 사용되고 있으며, RFID 시스템이 가져다 주는 유용성과 편리함 때문에 이들을 사용하는 응용[1,2]이 갈수록 늘어나고 있다. RFID 시스템은 태그, 리더, 그리고 데이터베이스를 가진 서버로 구성된다. 리더는 태그의 정보를 읽어서 서버에 저장된 태그 관련 정보와 비교하여 태그를 식별한다. 이 때 일반적으로 리더와 서버는 안전한 채널을 통해 통신을 하지만

† 본 연구는 금오공과대학교학술연구비에 의하여 연구된 논문
* 금오공과대학교 전자공학부, 제1저자(isjeon@kumoh.ac.kr)
** 경일대학교 사이버보안학과, 교신저자

리더와 태그는 무선으로 통신을 하며 이 통신 채널에 다양한 보안 공격이 존재할 수 있다. 그러므로 많은 RFID 시스템의 응용에 있어서 보안과 사용자 프라이버시 문제는 해결해야 할 중요한 문제이다. 따라서 이러한 보안 위협과 프라이버시 침해로부터 안전한 RFID 시스템의 인증 기법이 반드시 필요하다.

RFID 시스템을 위한 인증 기법이 많이 존재하지만 대부분은 태그의 과도한 계산 오버헤드와 저장 공간을 요구하기 때문에 계산능력과 저장 공간이 부족한 저가의 RFID 태그에서는 그러한 기법을 적용하기가 어렵다. 그래서 최근에는 저가의 RFID 태그를 위한 초경량의 인증 프로토콜들이 제안되었다.

2006년에 Peris-Lopez 등은 저가의 RFID 태그를 위해 초경량 인증 프로토콜인 LMAP[3]와 M^2AP[4]를 제안하였다. 그들의 프로토콜은 비트연산인 XOR, AND, OR와 모듈러(modular) 연산을 사용한 것으로 매우 간단하였다. 그러나 그 프로토콜은 비동기화(de-synchronization) 공격과 적극적 및 수동적인 공격에 취약하였다[5]. 2007년에 Chien[6]은 상호인증과 태그 익명성을 제공하는 새로운 초경량 상호인증 프로토콜인 SASI를 제안하였다. 그러나 Sun 등[7]은 SASI가 비동기화 공격에 저항할 수 없음을 보였고, Cao 등[8]은 SASI에 대해 중간자 공격을 하여 리더와 태그가 비동기화됨을 보였으며, Phan[9]은 태그 추적 공격을 위해 SASI에서 사용한 비트단위 OR 연산의 불균형성을 이용하였다. 2009년에 Peris-Lopez 등[10]은 SASI의 영향을 받아 새로운 프로토콜인 Gossamer 프로토콜을 제안하였으나 2010년 Targa 등[11]은 Gossamer가 비동기화 공격에 취약함을 보였다. 아주 최근에 Tian 등[12]은 XOR 연산, 회전(rotation) 연산, 그리고 순열(permutation) 연산을 사용하여 저가의 RFID 태그를 위한 초경량 인증 프로토콜인 RAPP를 제안하고, RAPP가 다양한 보안공격으로부터 안전함을 주장하였다.

그러나 RAPP는 저자의 주장과는 달리 비동기화 공격에 대한 취약점을 가지고 있다. 본 논문에서는 RAPP에 프로토콜에 존재하는 보안상의 문제점을 밝히고 또한 그 문제점을 극복할 수 있는 개선된 RAPP를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 Tian 등[12]이 제안한 RAPP와 프로토콜 기술에 필요한 기호 표기법을 소개한다. 3장에서는 RAPP에 대한 보안 취약점을 소개하고 4장에서는 RAPP의 보안 취약점을 극복할 수 있는 개선된 RAPP를 제시한다. 그리고 5장에서는 개선된 RAPP에 대한 안전성 분석과 성능을 평가하고 6장에서는 결론을 맺는다.

2. RAPP 소개

본 장에서는 저가의 RFID를 위한 초경량 인증 기법으로 아주 최근에 발표된 프로토콜인 RAPP를 소개한다. 그리고 본 논문에서 언급되는 프로토콜 기술에 사용되는 기호의 표기법을 소개하며 표 1에서 그 표기법을 정의하였다.

<표 1> 기호 표기법

기호	정의
K_1^o, K_2^o, K_3^o	리더와 태그가 공유하는 이전(old) 비밀키
K_1^n, K_2^n, K_3^n	리더와 태그가 공유하는 새로운(new) 비밀키
IDS^o, IDS^n	각각 태그의 이전 및 새로운 의사 아이디(pseudonym)
n_1, n_2, n_3	리더에 의해 생성된 난수
$Rot(A,B)$	A를 $w(B)$ 비트 왼쪽 회전, $w(B)$는 B의 해밍웨이트(Hamming weight)
$Per(A,B)$	B에 대한 A의 순열(permutation) 연산
$\oplus$	XOR 연산자
$\rightarrow$	메시지 전송

Tian 등[12]은 RAPP에서 초경량 RFID 인증 프로토콜 분야에서는 처음으로 순열(permutation) 연산인 $Per()$ 연산을 정의하고 이를 사용하였다. 그들이 정의한 $Per()$ 연산의 정의는 다음과 같다.

[정의] A와 B는 각각 다음과 같은 l 비트의 문자열이라 가정한다.

$$A = a_1 a_2 \cdots a_l,\ a_i \in \{0,1\},\ i = 1,2,\cdots,l$$

$$B = b_1 b_2 \cdots b_l,\ b_i \in \{0,1\},\ i = 1,2,\cdots,l$$

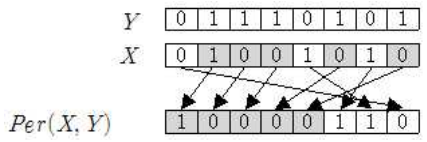
B의 해밍웨이트(Hamming weight)를 $w(B)$라고 표현하고, $w(B)$가 $m(0 \le m \le l)$이면

$$b_{k_1} = b_{k_2} = \cdots\ b_{k_m} = 1,$$
$$b_{k_{m+1}} = b_{k_{m+2}} = \cdots\ b_{k_l} = 0,\ \text{여기서}$$
$$1 \le k_1 < k_2 < \cdots < k_m \le l\ \text{이고}$$
$$1 \le k_{m+1} < k_{m+2} < \cdots < k_l \le l\ \text{이다.}$$

그렇다면 B에 대한 A의 순열 연산, $Per(A,B)$는 다음과 같다.

$$Per(A,B) = a_{k_1}a_{k_2}\cdots a_{k_m}a_{k_l}a_{k_{l-1}}\cdots a_{k_{m+2}}a_{k_{m-1}}$$

[예제] $X = 01001010$ 이고 $Y = 01110101$ 이라면 $Per(X,Y) = 10000110$ 가 된다. 그림 1은 예제의 연산과정을 알기 쉽게 도시하고 있다[12].



<그림 1> $Per(X,Y)$의 연산

그림 1에서 $Per(X,Y)$는 Y의 최상위비트에서 최하위비트까지 이동하면서 Y의 비트값이 1이면 X의 동일위치의 비트값으로 $Per(X,Y)$의 최상위비트에서부터 최하위비트 방향으로 채워나가고, Y의 비트값이 0이면 X의 동일 위치의 비트값으로 $Per(X,Y)$의 최하위비트에서부터 최상위비트 방향으로 채워나가면 계산된다.

RAPP는 저가의 태그, 리더, 그리고 백엔드(back-end) 데이터베이스의 세 개체를 포함한다. 리더와 백엔드 데이터베이스는 유선이든 무선이든 안전한 통신 채널에 의해 통신하는 반면에 리더와 태그의 통신 채널은 무선이며 보안 공격에 취약하다. RAPP에서 태그 내의 연산을 위해 $Per()$ 연산과 더불어 XOR 연산 및 회전연산인 $Rot()$ 연산을 사용하였다. 표 1에서 $Rot()$을 정의하였는데 $Rot(A,B)$는 A를 $w(B)$ 비트만큼 왼쪽 회전하는 것을 말하며, 여기서 $w(B)$는 B의 해밍웨이트(Hamming weight)이다. 태그는 L 비트의 고유한 아이디, ID와 네 개의 요소, $\{IDS, K_1, K_2, K_3\}$를 가지고 있다. 여기서 IDS는 태그의 의사아이디(pseudonym)이며 K_1, K_2, K_3는 비밀키이다. 비동기화(de-synchronization) 공격을 막기 위해 백엔드 데이터베이스는 각 태그의 의사아이디와 비밀키에 대해 이전(old) 값, $\{IDS^o, K_1^o, K_2^o, K_3^o\}$와 새로운(new) 값, $\{IDS^n, K_1^n, K_2^n, K_3^n\}$를 유지한다. RAPP는 크게 인증단계와 업데이트 단계로 구분되며 그 세부적인 수행 과정을 다음에 기술하였고, 이를 그림 2에 요약하였다.

[인증 단계]

- Step1. 리더는 태그에게 "Hello" 메시지를 보낸다.
- Step2. 리더의 메시지를 받고나서 태그는 리더에게 IDS를 보낸다.
- Step3. 태그로부터 IDS를 받고나서 리더는 데이터베이스에서 IDS를 찾아 대응하는 K_1, K_2, K_3를 얻고 난수 n_1을 생성한 후 (A,B)를 계산하여 태그에게 보낸다.
  $A = Per(K_2, K_1) \oplus n_1$
  $B = Per(K_1 \oplus K_2, Rot(n_1, n_1)) \oplus Per(n_1, K_1)$
- Step4. 태그는 수신한 A로부터 n_1'를 구하고 그것을 이용하여 B'를 구한다.
  $n_1' = A \oplus Per(K_2, K_1)$
  $B' = Per(K_1 \oplus K_2, Rot(n_1', n_1')) \oplus Per(n_1', K_1)$
  만약 $B_i' = B_i$이면 태그는 C를 계산하여 리더에게 보낸다.
  $C = Per(n_1' \oplus K_1, n_1' \oplus K_3) \oplus ID$
- Step5. 리더는 C를 받고나서 C'를 계산한다.
  $C' = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID$
  만약 $C' = C$이면 리더는 태그를 인증하고 난수 n_2를 생성한 후 (D,E)를 계산하여 태그에게 보내고 리더는 업데이트 단계로 넘어간다.
  $D = Per(K_3, K_2) \oplus n_2$
  $E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2)$
- Step6. 태그는 수신한 D로부터 n_2'를 구하고 그것을 이용하여 E'를 구한다.
  $n_2' = D \oplus Per(K_3, K_2)$
  $E' = Per(K_3, Rot(n_2', n_2')) \oplus Per(n_1', K_3 \oplus K_2)$
  만약 $E' = E$이면 태그는 리더를 인증하고 업데이트 단계로 넘어간다.

[업데이트 단계]

인증 단계를 마친 후 리더와 태그는 의사아이디와 비밀키를 갱신한다. 리더는 태그로부터 받은 IDS가 IDS^o와 같으면 $\{IDS^n, K_1^n, K_2^n, K_3^n\}$를 다음처럼 갱신한다.

$$IDS^n = Per(IDS^o, n_1 \oplus n_2) \oplus K_1^o \oplus K_2^o \oplus K_3^o$$

$$K_1^n = Per(K_1^o, n_1) \oplus K_2^o$$

$$K_2^n = Per(K_2^o, n_2) \oplus K_1^o$$

$$K_3^n = Per(K_3^o, n_1 \oplus n_2) \oplus IDS^o$$

만약 리더가 태그로부터 받은 IDS가 IDS^n과 같으면 리더는 $\{IDS^o, K_1^o, K_2^o, K_3^o\}$와$\{IDS^n, K_1^n, K_2^n, K_3^n\}$를 다음처럼 갱신한다.

Reader $\{IDS^o, K_1^o, K_2^o, K_3^o, IDS^n, K_1^n, K_2^n, K_3^n\}$	**Tag** $\{IDS, K_1, K_2, K_3\}$
$Hello$ ——————→	
	←—————— IDS
Search for IDS in the DB & obtain K_1, K_2, K_3 *Generate* n_1 $A = Per(K_2, K_1) \oplus n_1$ $B = Per(K_1 \oplus K_2, Rot(n_1, n_1)) \oplus Per(n_1, K_1)$	
A, B ——————→	$n_1' = A \oplus Per(K_2, K_1)$ $B' = Per(K_1 \oplus K_2, Rot(n_1', n_1')) \oplus Per(n_1', K_1)$ *Verify* $B' = B$ $C = Per(n_1' \oplus K_1, n_1' \oplus K_3) \oplus ID$
	←—————— C
$C' = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID$ *Verify* $C' = C$ *Generate* n_2 $D = Per(K_3, K_2) \oplus n_2$ $E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2)$	
D, E ——————→	$n_2' = D \oplus Per(K_3, K_2)$ $E' = Per(K_3, Rot(n_2', n_2')) \oplus Per(n_1', K_3 \oplus K_2)$ *Verify* $E' = E$
Updating: 1) if IDS^o is received: $IDS^n = Per(IDS^o, n_1 \oplus n_2) \oplus K_1^o \oplus K_2^o \oplus K_3^o$ $K_1^n = Per(K_1^o, n_1) \oplus K_2^o$ $K_2^n = Per(K_2^o, n_2) \oplus K_1^o$ $K_3^n = Per(K_3^o, n_1 \oplus n_2) \oplus IDS^o$ 2) if IDS^n is received: $IDS^o = IDS^n$ $K_1^o = K_1^n$ $K_2^o = K_2^n$ $K_3^o = K_3^n$ $IDS^n = Per(IDS^o, n_1 \oplus n_2) \oplus K_1^o \oplus K_2^o \oplus K_3^o$ $K_1^n = Per(K_1^o, n_1) \oplus K_2^o$ $K_2^n = Per(K_2^o, n_2) \oplus K_1^o$ $K_3^n = Per(K_3^o, n_1 \oplus n_2) \oplus IDS^o$	**Updating:** $IDS^* = Per(IDS, n_1 \oplus n_2) \oplus K_1 \oplus K_2 \oplus K_3$ $K_1^* = Per(K_1, n_1) \oplus K_2$ $K_2^* = Per(K_2, n_2) \oplus K_1$ $K_3^* = Per(K_3, n_1 \oplus n_2) \oplus IDS$ $IDS = IDS^*$ $K_1 = K_1^*$ $K_2 = K_2^*$ $K_3 = K_3^*$

<그림 2> 프로토콜 RAPP

$IDS^o = IDS^n$

$K_1^o = K_1^n$

$K_2^o = K_2^n$

$K_3^o = K_3^n$

$IDS^n = Per(IDS^o, n_1 \oplus n_2) \oplus K_1^o \oplus K_2^o \oplus K_3^o$

$K_1^n = Per(K_1^o, n_1) \oplus K_2^o$

$K_2^n = Per(K_2^o, n_2) \oplus K_1^o$

$K_3^n = Per(K_3^o, n_1 \oplus n_2) \oplus IDS^o$

한편 태그는 자신의 의사아이디와 비밀키를 다음처럼 갱신한다.

$IDS^* = Per(IDS, n_1 \oplus n_2) \oplus K_1 \oplus K_2 \oplus K_3$

$K_1^* = Per(K_1, n_1) \oplus K_2$

$K_2^* = Per(K_2, n_2) \oplus K_1$

$K_3^* = Per(K_3, n_1 \oplus n_2) \oplus IDS$

$IDS = IDS^*$

$K_1 = K_1^*$

$K_2 = K_2^*$

$K_3 = K_3^*$

3. RAPP의 보안 취약점 분석

Tian 등이 제안한 프로토콜 RAPP는 리더의 데이터베이스 갱신 오류에 의한 서비스 거부(Denial of Service) 공격에 취약할 뿐만 아니라 비동기화(de-synchronization) 공격에도 취약하다.

먼저 리더의 데이터베이스 갱신 오류에 의한 서비스 거부 공격을 살펴본다. 리더가 태그로부터 받은 IDS가 IDS^n과 같을 때 새로운 의사아이디와 비밀키 계산에 RAPP에서는 IDS^o와 K_1^o, K_2^o, K_3^o를 사용한다. 반면에 IDS가 IDS^n과 같을 경우 태그측은 의사아이디와 비밀키를 리더 측의 데이터베이스 내에 있는 IDS^n, K_1^n, K_2^n, K_3^n과 같은 값을 사용하여 갱신한다. 따라서 그러한 갱신 후 태그의 의사아이디와 비밀키는 리더의 데이터베이스 내에는 존재하지 않으므로 그 태그는 더 이상 리더의 인증을 받을 수 없게 되어 서비스 거부 공격을 당하게 된다. 이러한 공격을 방어하기 위해서는 IDS와 IDS^n이 서로 같을 때 리더측 데이터베이스에서는 태그의 새로운 의사아이디와 비밀키 계산에 IDS^o와 K_1^o, K_2^o, K_3^o를 사용하는 대신에 IDS^n과 K_1^n, K_2^n, K_3^n를 사용하도록 변경하면 위와 같은 서비스 거부 공격을 막을 수 있다.

다음으로 RAPP가 비동기화 공격에 취약함을 살펴본다. RAPP는 앞에서 지적한 데이터베이스 갱신 오류가 바르게 고쳐진 상태로 가정하며, 비동기화 공격 시나리오는 다음과 같다. 공격자는 리더와 태그가 주고받는 메시지, (A,B)와 (D,E)를 몰래 캡쳐함과 동시에 메시지 (D,E)가 태그로 전송되는 것을 차단시킨다. 그렇게 하면 리더의 IDS^n과 K_1^n, K_2^n, K_3^n는 갱신되지만 태그의 정보는 변하지 않는다. 그리고 바로 다음 세션에서 공격자는 리더가 태그에게 전송하는 메시지 (D,E)를 차단시킨다. 그러면 리더는 IDS^n과 K_1^n, K_2^n, K_3^n를 또 갱신하지만 태그의 정보는 여전히 바뀌지 않는다. 이러한 상태에서 공격자가 바로 리더로 위장하여 "Hello" 메시지를 태그로 전송한다. 태그는 IDS로 공격자에게 응답하고 공격자는 태그에게 몰래 캡쳐한 (A,B)로 응답한다. 이 때 IDS는 공격자가 (A,B)와 (D,E)를 몰래 캡쳐할 당시의 IDS이므로 태그의 검증을 통과하게 된다. 그러므로 태그는 공격자에게 메시지 C를 전송하고 공격자는 캡쳐한 (D,E)로 응답한다. 이 경우도 태그의 검증을 통과하게 되고 태그는 IDS와 K_1, K_2, K_3를 갱신한다. 그런데 갱신된 IDS와 K_1, K_2, K_3는 리더의 데이터베이스에 저장된 IDS^n과 K_1^n, K_2^n, K_3^n과는 다르다. 실제로 리더 측의 데이터베이스에 있는 IDS^n과 K_1^n, K_2^n, K_3^n은 현재 태그가 가지고 있는 IDS와 K_1, K_2, K_3를 이용하여 한 번 더 계산한 값으로 대체되어 있고 IDS^o와 K_1^o, K_2^o, K_3^o는 현재 태그가 가지고 있는 IDS와 K_1, K_2, K_3의 이전 값을 가지고 있다. 그러므로 현재 태그가 소유한 IDS와 K_1, K_2, K_3는 리더의 데이터베이스에 없는 값이므로 다음 세션부터 그 태그는 리더의 인증을 받을 수 없는 태그가 된다.

4. 개선된 RAPP

RAPP에 대해 지적한 내용 중 데이터베이스 갱신 시 오류로 인한 서비스 거부 공격에 대한 수정은 이미 앞 절에서 소개되었으므로 여기서는 언급을 생략하고, 본 절에서는 비동기화 공격에 대한 취약성을 극복할 수 있는 방법을 설명하고 이를 프로토콜에 반영한 개선된 RAPP를 제시한다. RAPP가 지적한 비동기화 공격에 저항하기 위해서는 태그에도 리더의 데이터베이스와 마찬가지로 이전 값, $\{IDS^o, K_1^o, K_2^o, K_3^o\}$와 새로운 값, $\{IDS^n, K_1^n, K_2^n, K_3^n\}$를 유지한다. 그리고 SASI[6]에서처럼 리더가 "Hello" 메시지를 보내면 태그는 새로운(new) IDS로 응답한다. 리더가 수신한 IDS를 데이터베이스에서 찾지 못하면 태그에게 "Hello"를 재전송하고, 태그는 저장된 이전(old) IDS로 응답한다. 이렇게 RAPP를 수정하면 앞 절에서 언급한 비동기화 공격으로부터 안전한 프로토콜을 만들 수 있다.

Reader $\{IDS^o, K_1^o, K_2^o, K_3^o, IDS^n, K_1^n, K_2^n, K_3^n\}$	**Tag** $\{IDS^o, K_1^o, K_2^o, K_3^o, IDS^n, K_1^n, K_2^n, K_3^n\}$
Hello ⟶	**if** *Hello received a second time* **then** $IDS = IDS^o, K_1 = K_1^o, K_2 = K_2^o, K_3 = K_3^o$ **else** $IDS = IDS^n, K_1 = K_1^n, K_2 = K_2^n, K_3 = K_3^n$
	⟵ IDS
if *IDS doesn't exist in the DB* **then** *resend Hello* **else** *set IDS and* K_1, K_2, K_3 *as the value of matched IDS's record;* *Generate* n_1 $A = Per(K_2, K_1) \oplus n_1$ $B = Per(K_1 \oplus K_2, Rot(n_1, n_1)) \oplus Per(n_1, K_1)$	
A, B ⟶	$n_1' = A \oplus Per(K_2, K_1)$ $B' = Per(K_1 \oplus K_2, Rot(n_1', n_1')) \oplus Per(n_1', K_1)$ *Verify* $B' = B$ $C = Per(n_1' \oplus K_1, n_1' \oplus K_3) \oplus ID$
	⟵ C
$C' = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID$ *Verify* $C' = C$ *Generate* n_2 $D = Per(K_3, K_2) \oplus n_2$ $E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2)$	
D, E ⟶	$n_2' = D \oplus Per(K_3, K_2)$ $E' = Per(K_3, Rot(n_2', n_2')) \oplus Per(n_1', K_3 \oplus K_2)$ *Verify* $E' = E$
Updating: $IDS^o = IDS$ $K_1^o = K_1$ $K_2^o = K_2$ $K_3^o = K_3$ $IDS^n = Per(IDS, n_1 \oplus n_2) \oplus K_1 \oplus K_2 \oplus K_3$ $K_1^n = Per(K_1, n_1) \oplus K_2$ $K_2^n = Per(K_2, n_2) \oplus K_1$ $K_3^n = Per(K_3, n_1 \oplus n_2) \oplus IDS$	**Updating:** $IDS^o = IDS$ $K_1^o = K_1$ $K_2^o = K_2$ $K_3^o = K_3$ $IDS^n = Per(IDS, n_1' \oplus n_2') \oplus K_1 \oplus K_2 \oplus K_3$ $K_1^n = Per(K_1, n_1') \oplus K_2$ $K_2^n = Per(K_2, n_2') \oplus K_1$ $K_3^n = Per(K_3, n_1' \oplus n_2') \oplus IDS$

<그림 3> 개선된 RAPP

개선된 RAPP에 대한 구체적인 수행 절차를 다음에 기술하였고, 이를 그림 3에 요약하였다.

[인증 단계]

- Step1. 리더는 태그에게 "Hello" 메시지를 보낸다.
- Step2. 리더의 메시지를 받고나서 태그는 IDS^n의 값을 IDS로 해서 리더에게 보낸다. 그리고 K_1, K_2, K_3의 값을 K_1^n, K_2^n, K_3^n으로 각각 배정한다.
- Step3. 태그의 IDS를 받고나서 리더는 데이터베이스에서 IDS를 검색한다. 만약 검색에 실패하면 태그에게 "Hello"를 재전송하고 나서 Step4로 가고, 검색에 성공하면 Step5로 간다.
- Step4. 태그는 두 번째 "Hello" 메시지를 받으면 IDS^o의 값을 IDS로 해서 리더에게 보낸다. 그리고 K_1, K_2, K_3의 값을 K_1^o, K_2^o, K_3^o으로 각각 배정한다.
- Step5. 리더는 IDS를 데이터베이스에서 검색하여 IDS가 존재하면 그 IDS의 비밀키 값을 대응하는 K_1, K_2, K_3에 각각 배정한다. 그리고 난수 n_1을 생성한 후 (A, B)를 계산하여 태그에게 보낸다.

  $A = Per(K_2, K_1) \oplus n_1$

  $B = Per(K_1 \oplus K_2, Rot(n_1, n_1)) \oplus Per(n_1, K_1)$
- Step6. 태그는 수신한 A로부터 n_1'를 구하고 그것을 이용하여 B'를 구한다.

  $n_1' = A \oplus Per(K_2, K_1)$

  $B' = Per(K_1 \oplus K_2, Rot(n_1', n_1')) \oplus Per(n_1', K_1)$

  만약 $B_i' = B_i$이면 태그는 C를 계산하여 리더에게 보낸다.

  $C = Per(n_1' \oplus K_1, n_1' \oplus K_3) \oplus ID$
- Step7. 리더는 C를 받고나서 C'를 계산한다.

  $C' = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID$

  만약 $C' = C$이면 리더는 태그를 인증하고 난수 n_2를 생성한 후 (D, E)를 계산하여 태그에게 보내고 업데이트 단계로 넘어간다.

  $D = Per(K_3, K_2) \oplus n_2$

  $E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2)$
- Step8. 태그는 수신한 D로부터 n_2'를 구하고 그것을 이용하여 E'를 구한다.

  $n_2' = D \oplus Per(K_3, K_2)$

  $E' = Per(K_3, Rot(n_2', n_2')) \oplus Per(n_1', K_3 \oplus K_2)$

  만약 $E' = E$이면 태그는 리더를 인증하고 업데이트 단계로 넘어간다.

[업데이트 단계]

리더는 데이터베이스 내의 태그 관련정보를 다음처럼 갱신한다.

$IDS^o = IDS$

$K_1^o = K_1$

$K_2^o = K_2$

$K_3^o = K_3$

$IDS^n = Per(IDS, n_1 \oplus n_2) \oplus K_1 \oplus K_2 \oplus K_3$

$K_1^n = Per(K_1, n_1) \oplus K_2$

$K_2^n = Per(K_2, n_2) \oplus K_1$

$K_3^n = Per(K_3, n_1 \oplus n_2) \oplus IDS$

한편 태그는 자신의 의사아이디와 비밀키를 다음처럼 갱신한다.

$IDS^o = IDS$

$K_1^o = K_1$

$K_2^o = K_2$

<표 2> 개선된 RAPP와 기존 프로토콜의 비교
* : L은 아이디나 비밀키의 길이

비교요소 \ 프로토콜	LMAP[3]	M^{2}AP[4]	SASI[6]	Gossamer[10]	RAPP[12]	개선된 RAPP
태그 추적 저항	×	×	×	○	○	○
비동기화공격 저항	×	×	×	×	×	○
디스클로즈공격 저항	×	×	×	○	○	○
요구되는 저장공간	6L*	6L	7L	7L	5L	9L
사용된 연산	$\oplus, +, \vee$	$\oplus, +, \vee, \wedge$	$\oplus, +, \vee, Rot$	$\oplus, +, Rot_2, Mixbits$	$\oplus, Rot, Per$	$\oplus, Rot, Per$

$K_3^o = K_3$

$IDS^n = Per(IDS, n_1' \oplus n_2') \oplus K_1 \oplus K_2 \oplus K_3$

$K_1^n = Per(K_1, n_1') \oplus K_2$

$K_2^n = Per(K_2, n_2') \oplus K_1$

$K_3^n = Per(K_3, n_1' \oplus n_2') \oplus IDS$

5. 안전성 분석 및 성능평가

5.1 안전성 분석

Tian 등[12]은 그들의 논문에서 RAPP가 상호인증과 태그 익명성(anonymity) 및 비추적성(untraceability)을 제공하고, 또한 재전송(replay) 공격, 디스클로즈(disclose) 공격, 비동기화 공격 등으로부터 안전함을 보였다. 비동기화 공격에 대한 안전성을 제외하고, RAPP에 대한 보안 공격으로부터의 안전성은 개선된 RAPP에서도 그대로 유지된다.

그러므로 여기서는 개선된 RAPP에서 비동기화 공격에 대한 안전성을 분석한다. 공격자가 리더와 태그가 주고받는 메시지, (A,B)와 (D,E)를 몰래 캡쳐하면서 그 세션과 연속한 다음 세션에서 메시지 (D,E)가 태그로 전송되는 것을 차단시키고 이어지는 다음 세션에서 공격자가 리더로 위장하여 캡쳐한 메시지로 태그를 속여서 태그의 정보가 갱신되고, 그 결과 태그의 IDS^n과 K_1^n, K_2^n, K_3^n가 데이터베이스의 그것과 다르게 되었다고 하자. 그렇지만 다음 세션의 인증과정에서 리더는 태그에게 "Hello"를 재전송하게 되고, 태그는 IDS와 K_1, K_2, K_3의 값으로 IDS^o와 K_1^o, K_2^o, K_3^o를 배정하고 IDS를 리더로 재전송하게 된다. 이 때 데이터베이스에 저장된 IDS^o와 K_1^o, K_2^o, K_3^o와 태그에 저장된 IDS^o와 K_1^o, K_2^o, K_3^o의 값은 서로 같으므로 인증을 통과하고 갱신과정을 거친다. 프로토콜 수행 후 데이터베이스의 $\{IDS^o, K_1^o, K_2^o, K_3^o\}$와 $\{IDS^n, K_1^n, K_2^n, K_3^n\}$의 값은 태그의 그것들과 서로 같게 된다. 그러므로 소개한 개선된 RAPP는 비동기화 공격으로부터 안전한 프로토콜이다.

5.2 성능평가

본 연구에서 제시한 개선된 RAPP의 성능평가는 다양한 보안공격으로부터의 안전성 평가에 중점을 두고 부가적으로 연산의 종류 및 요구된 저장공간의 크기를 고찰한다. 그리고 리더와 리더에 연결된 서버의 하드웨어 및 소프트웨어 능력은 프로토콜을 수행하는데 크게 문제될 것이 없으므로 태그쪽의 경우만 평가하여 표 2에 요약하였다.

비록 저가의 태그를 위한 프로토콜일지라도 보안공격에 대한 안전성이 보장되지 않는다면 그 프로토콜은 실제로 사용이 어려운 무용지물이 되므로 프로토콜의 안전성이 가장 중요한 평가 요소가 된다. 표 2에서 알 수 있듯이, 비교된 기존 프로토콜들이 비동기화 공격으로부터 모두 안전하지 않지만 본 논문에서 제안한 개선된 RAPP는 그 공격으로부터 안전한 프로토콜이다.

한편 제안한 프로토콜에서 사용한 주요 연산은 RAPP에서와 동일한 XOR 연산, 회전(rotation) 연산, 그리고 순열(permutation) 연산만을 사용하였고, 이 연산들의 수행횟수는 RAPP와 같으나 비동기화 공격이 존재할 때 1번의 추가적인 의사아이디 전송이 필요하다. 그리고 태그가 비동기화 공격에 저항할 수 있도록 하기 위해 의사아이디와 비밀키에 대해 각각 두개의 값을 저장함으로 인해 요구되는 저장공간은 타프로토콜에 비해 크다. 그러나 전술한 바와 같이 비동기화 공격과 같은 보안공격에 대한 프로토콜의 안전성이 이러한 저장공간의 대소문제나 추가적인 의사아이디 전송과 같은 오버헤드보다 훨씬 더 중요한 요소이기 때문에 제안한 프로토콜은 기존 프로토콜들보다 우수한 프로토콜이라고 할 수 있다.

6. 결 론

본 논문에서는 Tian 등이 제안한 저가의 RFID를 위한 초경량 상호인증 프로토콜인 RAPP가 비동기화 공격에 취약함을 보이고, 이를 개선한 프로토콜을 소개하였다. 본 논문에서 소개한 개선된 RAPP 프로토콜은 RAPP에서와 동일한 저비용의 연산들만을 사용한 초경량 프로토콜이고, 저가의 RFID 태그에서 요구되는 다양한 보안요건들을 충족시키고 있다. 따라서 본 연구에서 제시한 개선된 RAPP는 저가의 RFID 태그를 위한 실질적인 프로토콜로 활용될 수 있을 것이다.

참 고 문 헌

[1] 문병현, 이태훈, 서용석, 황지영, 류정탁,. "RFID를 이용한 출입관리 로봇," 한국산업정보학회논문지, Vol. 13, No. 4, pp. 139-144, 2008.

[2] 최형림, 박병주, 신중조, 이정희, "RFID/OCR 기반의 자동화 게이트시스템 개발," 한국산업정보학회논문지, Vol. 12, No. 2, pp. 37-44, 2007.

[3] P. Peris-Lopez, J. C. Hernandez- Castro, J. M. E. Tapiador, and A. Ribagorda, "LMAP: a real lightweight mutual authentication protocol for low-cost RFID tags," in Proc. 2006 Workshop RFID Security. 2006.

[4] P. Peris-Lopez, J. C. Hernandez- Castro, J. M. E. Tapiador, and A. Ribagorda, "M2AP: a minimalist mutual-authentication protocol for low cost RFID tags," in Proc. 2006 International Conference on Ubiquitous Intelligence and Computing, pp. 912 - 923. 2006.

[5] T. Li and G. Wang, "Security analysis of two ultra-lightweight RFID authentication protocols," in Proc. 2007 IFIP RC-11 International Information Security Conference, pp. 109 - 120. 2007.

[6] H. Y. Chien, "SASI: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," IEEE Trans. Dependable and Secure Computing, vol. 4, no. 4, pp. 337 - 340, 2007.

[7] H. M. Sun, W. C. Ting, and K. H. Wang, "On the security of Chien's ultralightweight RFID authentication protocol," IEEE Trans. Dependable and Secure Computing, vol. 8, no. 2, pp. 315 - 317, 2011.

[8] T. Cao, E. Bertino, and H. Lei, "Security analysis of the SASI protocol," IEEE Trans. Dependable and Secure Computing, vol. 6, no. 1, pp. 73 - 77, 2009.

[9] R. C. W. Phan, "Cryptanalysis of a new ultralightweight RFID authentication protocol - SASI," IEEE Trans. Dependable and Secure Computing, vol. 6, no. 4, pp. 316 - 320, 2009.

[10] P. Peris-Lopez, J. Hernandez-Castro, J. Tapiador, A. Ribagorda, "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol," Information Security Applications, pp. 56 - 68, 2009.

[11] D. Tagra, M. Rahman, S. Sampalli, "Technique for preventing DoS attacks on RFID systems," 18th international conference on software telecommunications and computer networks - SoftCOM'10, IEEE Computer Society, 2010.

[12] Y. Tian, G. Chen, J. Li, "A New Ultralightweight RFID Authentication Protocol with Permutation," IEEE Communication Letters, Vol. 16, No. 5, pp. 702-705, 2012

전 일 수 (Il-Soo Jeon)

- 정회원
- 경북대학교 전자공학과 공학사
- 경북대학교 전자공학과 공학석사
- 경북대학교 전자공학과 공학박사
- 금오공과대학교 전자공학부 교수
- 관심분야 : 정보보호, 암호프로토콜

윤 은 준 (Eun-Jun Yoon)

- 정회원
- 경일대학교 섬유공학과 공학사
- 경일대학교 컴퓨터공학과 공학석사
- 경북대학교 컴퓨터공학과 공학박사
- 경일대학교 사이버보안학과 조교수
- 관심분야 : 암호학, 정보보호, 유비쿼터스보안, 네트워크보안, 인증, 융합보안, 스테가노그라피

논 문 접 수 일 : 2012년 08월 09일
1차수정완료일 : 2012년 10월 16일
2차수정완료일 : 2012년 11월 21일
게 재 확 정 일 : 2012년 11월 21일