정수연산방식을 적용한 저가 전자태그의 경량 인증기법 연구

곽민혜*, 김광조*

*한국정보통신대학교, 국제정보보호기술연구소

A Study on Lightweight Authentication Scheme using Integer Operation for lowcost RFID Tag

Min-Hea Kwak*, Kwangjo Kim*

*International Research Center for information security(IRIS),

Information and Communications University (ICU)

요 약

RFID(Radio Frequency IDentification)는 유비쿼터스 환경에서 가장 주목받고 있는 기술중의 하나로 사회전반에 걸쳐 다양하게 적용되고 있으나 있다. 그러나 RFID의 접근 편의성에도 불구하고 시스템 보안과 프라이버시 침해가 대두되고 있어 이를 해결하기 위한 암호기법과 인증기법의 개발이 필수적이다. 대부분의 경량 인증기법에서 사용하고 있는 난수발생기와 비트 연산방식은 효율성은 뛰어나나 외부 공격에 취약하다고 알려져 있다. 본 논문에서는 94비트의 저장공간(ROM)과 24비트의 메모리(RAM)로 구현 가능한 저가형 전자태그의 연산능력을 고려한 효율적이고 안정적인 정수연산방식 기반의 RFID 인증 기법을 제안한다. 이는 EPC 태그의 규격을 만족할 뿐만 아니라 중간자 공격이나 전수조사같은 외부공격에 대응할 수 있는 안정적인 기법이다.

I. 서론

현대사회에서 전자, 정보통신, 가전장비의 디지털 가속화와 통신 인프라의 확충은 사용자에게 언제 어디서나 다양한 서비스를 제공하는 유비쿼터스 기술로 발전하고 있다. 이런 유비쿼터스 환경에서 가장 주목받고 있는 기술중의 하나는 RFID(Radio Frequency IDentification, 전자태그)이다. 전자태그는 저전력의 작은 크기의 전자태그에 무선 주파수를 이용하여 물리적 접촉없이 정보를 저장하거나 읽는 무선 인식기술을 말하며 기존의 바코드를 대체하여 금융, 의료, 교통, 제조, 문화등 사회전반에 걸쳐 다양하게 적용되고 있다. 전자태그는 사용자 접근 편의성을 제공하지만 안전성과 프라이버시 문제을 야기시킨다. 그 예로, 리더와 태그사이의 무선통신을 제3자가 도청하여 식별정보를 분석하고 태그의 위치정보나 상품정보까지 확인할 수 있으며 이는 사용자의 프라이버시 침해와도 직결되어있다. 이러한 안전성 문제를 해결하기위하여 태그, 리더, 데이터베이스 서버간의 인증을 통한 정보제공에 대한 연구가 진행되고 있다[1][2][3][4].

그러나 저가의 태그는 제한적인 연산능력과 저장공간의 한계로 인해 전통적 암호기법인 대칭키, 공개키같은 암호기법의 사용이 힘들다. 이러한 저가형 전자태그를 위한 자원의 소모가 적은 안전한 암호 기법과 최소한의 자원을 사용하면서도 안전한 인증 기법의 개발은 필수적이다.

본 논문에서는 전자태그의 안전성 문제를 해결하기 위한 기존 경량 인증기법을 살펴보고 이들의 문제점을 분석, 저가형 전자태그의 연산 능력과 저장공간을 고려한 효율적이고 안전한 방식을 제안하고자 한다. 2장에서는 Stephane등[6]이 제안한 정수연산을

* 본 연구는 MIC/IITA의 IT R&D 프로그램 연구과제 (2005-S-106-02. RFID/USN용 센서 태그 및 센서노드 기술 개발) 지원으로 수행하였습니다.

응용한 경량 인증기법을 간략히 기술하고 취약점을 분석한다. 3장에서는 이를 바탕으로 Stephane등의 인증기법을 보완한 안정적이고 효율적인 인증기법을 제안한다. 4장에서는 기존 경량 인증기법과 제안된 인증 기법의 안정성과 효율성 측면에서 비교 분석한다. 마지막으로 5장에서는 결론을 맺는다.

II. 관련연구

2.1 용어 정의

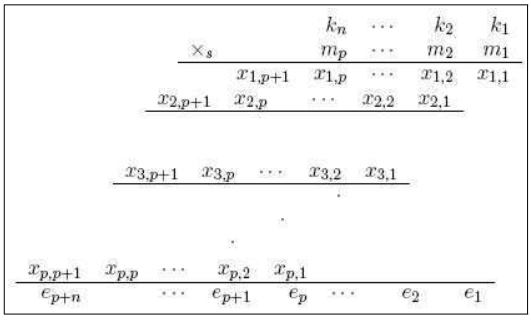
표 1은 본 논문의 인증기법에서 사용하는 용어들의 정의이다.

[표 1] 표기법

표기	내 용
K	비밀키, $K=\{k_1 k_2 \ldots\ldots k_n k_{n+1}\}$
K_r	초기 비밀키, $K_r=\{t_{n+1} t_n \ldots t_2 t_1\}$
$(K_r)_t$	K_r의 우측 t번째 비트
$(K_r)'$	K_r의 좌측 n-1 비트
IDS	$IDS=\{IDS_1, IDS_2 \ldots\ldots IDS_n\}$
r_i	랜덤값
$flag$	세션상태, 정상세션 0, 비정상적 세션완료 1
t_i	비밀키 K 와 랜덤값 r_i 내부연산 결과
X	리더와 태그의 메모리 공간
e_i	$(X+_{IDS} K_i)_1$

2.2 AIA(Abstractions of Integer Aritmetic)

Stephane등은 두 개의 이진 수열의 내부연산을 통한 특별한 정수 연산방식을 소개했다. 예를 들어, n 자리수 정수열 $k_n \ldots\ldots k_2 k_1$과 p 자리수 정수열 $m_n \ldots\ldots m_2 m_1$의 연산결과를 $e_{p+n+1} e_{p+n} \ldots\ldots e_2 e_1$이라고 하자. 자세한 내부연산과정은 그림 1을 참조한다.



[그림 1] 정수열 내부연산

예를 들어, 10진수 3과 7의 연산을 가정하면 이진 연산자 ⊗,⊕에 의한 연산결과는 0~9 사이의 한쌍으로 대응된다. 이진 곱셈 연산자 ⊗:(3,7)→(2,1) 또는 3⊗7=21로 표기, 이진 덧셈 연산자 ⊕:(3,7)→(1.0) 또는 3⊕7=10로 표기된다. 위의 이진 연산을 올림수와 나머지의 쌍으로 표기하면 ⊗:(3,7)→((3⊗7)c,(3⊗7)r)로 표기할 수 있다. 위의 방식으로 정수열 $k_n \ldots\ldots k_2 k_1$과 m_i의 연산결과 $x_{i,n+1} x_{i,n} \ldots\ldots x_{i,2} x_{i,1}$은 아래와 같이 표기한다.

$x_{i,1} = (k_1 \otimes m_i)_r$

$x_{i,2} = ((k_2 \otimes m_i)_r \oplus (k_1 \otimes m_i)_c)_r$

$x_{i,3} = ((k_3 \otimes m_i)_r \oplus ((k_2 \otimes m_i)_c \oplus ((k_1 \otimes m_i)_r \oplus (k_1 \otimes m_i)_c)_c)_r)$

최종 연산결과인 $e_{p+n+1} e_{p+n} \ldots\ldots e_2 e_1$은 중간 연산결과인$x_{i,j}$ 수직 열의 합이다.

Stephane등은 정수열 연산 AIA를 아래와 같이 정의하고 있다.

정의1 (Abstraction of Integer Arithmetic)

임의의 기수 b의 $4b^2$자리수 수열 s는 곱셈, 덧셈의 올림수와 나머지를 고려한 곱셈 알고리즘을 이용하여 기수 b의 수열 전체 집합 B에 대하여 이진연산 $\times_s$을 정의한다. $(b, \times_s)$쌍은 정수 연산방식(Abstraction of Integer Arithmetic, AIA)라 한다.

정의2 (Partial Abstraction of Integer Arithmetic)

임의의 기수 b의 $2b^2$자리수 수열 s는 곱셈, 덧셈의 올림수와 나머지를 고려한 곱셈 알고리듬을 이용하여 기수 b의 수열 전체 집합 B에 대하여 이진연산 $+_s$을 정의한다. $(b, +_s)$쌍은 정수 연산방식(Partial Abstraction of Integer Arithmetic, PAIA)라 한다.

2.3 Stephane등의 인증기법

2.3.1 인증기법

리더와 태그는 인증 시작전에 비밀키 *(K,PA,d)*를 생성하고 공유하는데 K는 각각 $n+1$길이의 b 자리수 수열 $K_1 K_2 \ldots\ldots K_b$의 집합, PA는 각각 $2b^2$길이의 n 자리수 수열 $PA_1 PA_2 \ldots\ldots PA_n$의 집합, d는 임의의 기수 b이다. 리더와 태그는 임의 기수 $d=m_1$와 n자리 수열 $k_n \ldots\ldots k_2 k_1$의 연산으로 $K_d = x_{i,n+1} x_{i,n} \ldots\ldots x_{i,2} x_{i,1}$를 공유하고 $K_{d=m_1}$의 좌측 n개의 수열를 메모리에 X로 저장한다. 리더는 임의의 기수 $m_2 \in 1, 2, \ldots, b$를 생성하고 K_{m_2}와 X를 PA를 이용한 정수연산 (AIA) 결과값 e_2를 태그로 전송한다. 리더는 수신한 (m_2, e_2)와 K_{m_2}와 X의 정수연산 결과 값과 수신한 e_2가 동일한지 확인하고

X를 갱신한다. 태그는 $m_3 \in 1,2,\ldots,b$를 생성하고 K_{m_3}와 X의 정수연산결과인 e_3를 리더로 전송한다. 사전에 지정된 횟수만큼 인증과정을 반복한 뒤 양측은 동일한 비밀키를 공유를 확신한다. 간략한 인증 알고리즘은 아래와 그림 2과 같다.

$Step$ 1 리더,태그: $X \leftarrow (K_d)'$
$Step2a$ 리더 : $m_2 \in \{1,2,\ldots,b\}$ 생성
$Step2b$ 리더 : $X +_{PA} K_{m_2}$ 계산
$Step2c$ 리더→태그 : (m_2, e_2) 전송
$Step$ 2d 태그 : $X \leftarrow (X +_{PA} K_{m_2})'$ 갱신
$Step3a$ 태그→리더 : (m_{i-1}, e_{i-1}) 수신
$Step3b$ 태그 : $(X +_{PA} K_{m_{i-1}})_1 \neq e_{i-1}$이면 인증실패
$Step3c$ 태그 : $X \leftarrow (X +_{PA} K_{m_{i-1}})'$ 갱신
$Step3d$ 태그 : $m_i \in \{1,2,\ldots,b\}$ 생성
$Step$ 3e 태그 : $X +_{PA} K_{m_i}$ 계산
$Step3f$ 태그→리더 : (m_i, e_i) 전송
$Step3g$ 리더 : $X \leftarrow (X +_{PA} K_{mi})'$ 갱신

[그림 2] 리더와 태그간 인증 알고리즘

2.3.2 취약점 분석

Stephane등은 인증세션의 반복을 통해 리더와 태그간의 상호 인증하는 방식을 채택하고 있는데 b=4, n=10일때 40라운드 이상 상호인증 과정을 반복해야 최소한의 전수 조사공격을 예방할 수 있다.(2^{80}). 태그는 각각 $b(n+1)\log_2(b)+2b$비트의 저장공간(ROM)과 $(n+2)\log_2(b)$비트의 휘발성 메모리(RAM)를 필요로 한다. b=4, n=10일때 태그는 90 비트의 저장공간과 24비트의 휘발성 메모리가 필요하다. 이는 표준 EPC(Electronic Product Code)태그의 요구조건을 만족한다. 그러나 상호인증의 초기세션에서는 초기 비밀키 K(2^{44})가 누출된 경우 (m_i, e_i)가 노출될 확률 $1/b$이다. 즉, 악의적인 제3자가 인증세션에 개입하여 정상적인 멤버로 참여해 메시지의 위변조가 가능한 중간자 공격이 가능하다. 또한, 리더측은 메시지(m_i, e_i)를 전송 뒤 레지스터 X값을 갱신하는데 악의적인 제3자에 의한 세션 종료시 태그측은 레지스터 X값을 갱신할 수 없으므로 비동기화 문제가 발생한다. Stephane등의 인증기법은 상호 인증세션의 반복을 통해 리더와 태그측이 메시지생성에 참여하고 세션마다 메시지가 변하기 때문에 제3자가 메시지를 탈취해 추후 사용이 불가능하다. 즉, 재전송 공격의 발생의 위험을 방지할 수 있다.

III. 제안기법

제안기법은 Stephane등이 제시한 AIA와 난수발생기를 이용한 상호인증 방식으로 구현된다.

리더는 각 태그에 대응하는 모든 초기 비밀키 *(K,IDS)*를 저장하고 태그는 비밀키 K만 메모리에 저장하고 나머지 비밀키 IDS_i는 논리게이트로서 하드웨어 형태로 구현한다. 인증세션은 태그와 리더의 비밀키와 랜덤값의 AIA 연산결과 양측에서 동일하게 발생하는지 확인하는 과정으로 리더가 지정한 횟수만큼 인증이 성공할 경우 태그와 리더는 동일한 비밀카를 공유하고 있음을 확인하는 기법이다. 제안기법은 사전 비밀키 계산과 인증세션의 반복으로 구성되어있다. 인증시작전 리더와 태그는 초기 비밀키 *(K, IDS)*를 아래와 같이 구성하고 세션상태를 나타내는 *flag*값을 0으로 설정한다. *flag*가 0 일때는 정상세션, 1 일때는 비정상적 세션완료를 의미한다.

1. K는 각각 n+1자리수 수열인 $K = k_1 k_2 \cdots k_n k_{n+1}$의 집합이며, $i \neq j$일때 $(K_i)_1 \neq (K_j)_1$이다.
2. IDS는 각각 $4b^2$자리수 수열인 $IDS_1, IDS_2 \ldots\ldots IDS_n$의 집합으로 각 태그에는 IDS_i가 AIA으로서 하드웨어적으로 구현되어있다.

그림 3과 같이 리더와 태그가 공유한 비밀키 K와 랜덤값 r의 연산으로 초기 비밀키 $K_i = t_{n+1} t_n \cdots t_2 t_1$는 그림 1의 정수연산과 동일한 방식으로 생성하여 인증 시작 전 $(K_i)'$를 메모리 X에 저장한다.

입력 : $K = k_n k_{n-1} \cdots k_2 k_1, random\ number\ i$
$Step1 : t_1 \leftarrow (k_1 \otimes r)_r, carry \leftarrow (k_1 \otimes r)_c$
$Step\ 2 : For\ j = 2$
$t_j \leftarrow ((k_j \otimes r)_r \oplus carry) r$
$carry \leftarrow ((k_j \otimes r)_c \oplus ((k_j \otimes r)_r \oplus carry)_c)_r$
$End\ For$
$Step3 : t_{n+1} \leftarrow carry$
출력 : $K_i = t_{n+1} t_n \cdots t_2 t_1$

[그림 3] 초기 비밀키 생성

리더는 랜덤값 r_2을 생성하여 메모리 $X = (K_r)'$에 태그와 동일한 IDS_i를 이용하여 K_{r_2}를 AIA연산을 한다. 연산결과에서 $e_2 = (x_{1,2} \otimes x_{2,1})_r$와 리더가

생성한 r_2값, 그리고 세션상태를 나타내는 *flag*′를 태그에 전송한다. 태그와 리더가 공유한 초기 비밀키 $(K_i)'$와 K_{r_2}의 연산 결과가 수신한 e_2와 동일한지 비교하고 동일값일때는 메모리의 $X=(X+_{IDS_i}K_r)'$을 갱신하고 아닐경우 인증실패로 간주한다. 수신값 (r_2,e_2)이 리더로부터의 정상적인 메시지로 확인된 경우 태그는 랜덤값 r_3를 생성하고 K_{r_3}를 계산하여 기존 X와 K_{r_3}의 AIA연산을 한다. 연산결과값 e_3와 r_3, *flag*를 리더측에 전송한다. 세션이 비정상적으로 종료되었을 경우 *flag*는 1로 변경되고 태그는 수신한 *flag*′과 태그측의 flag값을 비교하여 태그는 인증실패로 종료한다.

$Step1$ 리더, 태그 : $X=(K_i)'=t_{n+1}t_n\cdots t_2$
$Step2$ 리더 : 랜덤값 r_2생성
$Step3$ 리더 : $X\leftarrow(X+_{IDS_i}K_{r_2})'$ 갱신
$Step4$ 리더→태그 : $(flag',e_2,r_2)$
$Step5$ 태그 : $(X+_{IDS_i}K_{r_2})$계산
$Step6$ 태그 : $(X+_{IDS_i}K_{r_2})'_1=e_2, flag'=flag$이면 인증성공,
$Step7$ 태그 : $X\leftarrow(X+_{IDS_i}K_{r_3})'$ 갱신
$Step8$ 태그 : 랜덤값 r_3 생성
$Step9$ 태그 : $(X+_{IDS_i}K_{r_3})'_1=e_3$계산
$Step10$ 태그→리더 : $(flag',e_3,r_3)$전송

[그림 4] 초기 상호인증 과정

*r*번째 인증세션의 알고리즘은 아래 그림 5와 같다.

$Step1$ $(flag',r_{r-1},e_{r-1})$수신
$Step2$ $(X+_{IDS_i}K_{r_{r-1}})_1\neq e_{r-1}, flag'\neq flag$면 인증
$Step3$ $X\leftarrow(X+_{IDS_i}K_{t_{r-1}})'$ 갱신
$Step4$ 랜덤값 r_r생성
$Step5$ $X+_{IDS_i}K_{r_r}$
$Step6$ $(flag',r_r,e_r)$ 전송
$Step7$ $X\leftarrow(X+_{IDS_i}K_{r_r})'$ 갱신

[그림 5] *r*번째 인증세션

IV. 제안기법 분석

4.1 안전성

제안기법은 상호인증과정의 반복과 초기 비밀키(K,IDS)분석의 어려움을 기반으로 안전성을 제공한다. 초기 비밀키 (K, IDS)의 전수조사공격을 하기 위해서는 각각 2^{n+1}추측과 $(2^{12}*(r!)^r)^n$추측이 필요하다. Stephane등의 기법에서는 한정된 PA를 생성하여 동일한 *PA*를 가지는 태그발생 위험이 있는 반면 제안기법에는 각각의 태그는 단일한 논리게이트 *IDS*i가 하드웨어적으로 구현돼 있어 비밀키 *K*가 노출되어도 동일한 논리게이트를 가지는 태그를 만들어야 하기 때문에 태그복제가 불가능하다.

리더와 태그는 세션별 랜덤값 r_1를 생성, *X*값을 갱신하기 때문에 중간자 공격자 공격을 예방할 수 있다. 태그와 리더는 Tag ID를 전송하는 대신 인증세션을 반복을 통해 키 공유를 확신함으로 익명성을 보장한다. Stephane등의 기법에서 발견되었던 비동기화 문제는 세션상태를 나타내는 *flag*를 설정해 악의적인 세션차단이나 네트워크 단절로 인한 불안정한 종료시 *flag*값을 변경해 비동기화 문제를 해결한다. 기존의 경량 인증 기법간의 안정성 비교는 아래의 표 2과 같다. 표 2의 안정성 비교는 Chien[7]의 연구결과를 바탕으로 하였으며 기존 기법의 취약점이었던 재전송공격과 중간자 공격에 대한 대응 가능하고 Chien등[7]의 취약점인 비동기화 문제 또한 해결하였다.

[표 2] 안정성 비교

Item	Chien [7]	Duc [8]	Karikeya-Nesterenko[9]	Stephane [6]	제안기법
기밀성	○	○	○	○	○
익명성	○	○	×	○	○
재전송공격 방어	○	×	×	○	○
중간자공격 방어	○	×	×	×	○
동기화	×	×	×	×	○

※ ○: 보안 요구사항 만족, × : 보안 요구사항 불만족

4.2 효율성

EPC 샘플 태그는 128-512비트의의 저장공간과 32-128비트의 휘발성 메모리를 요구한다.[5] 제안기법에서는 K_i는 n+1자리 수열이고 기수는 r일때 저장공간은 r(n+1)log_2(r)+4비트가 필요하고 메모리는 (n+2)log_2(r)비트를 요구한다. 리더측에서는 *IDS*i 구현을 위해 n(r^2-r)+$blog_2$(r!))를 비밀키 *K*와 *flag*의 메모리 r(n+1)log_{2r}+4를 필요로 한다. r=4, n=10일때 태그는 94비트의 저장공간과 24비트의 메모리, 그리고 300-400비트의 논리 게이트가 필요하고 리더측은 414비트의 메모리를 요구한다.

V. 결론

본 논문에서는 정수연산방식을 이용한 저가형 태그에 적합한 경량 인증기법을 제안하였다. 기존의 경량 인증기법에서 적용된 비트 연산와 난수발생을 이용한 방식은 전수조사 및 적극적 공격에 취약한 단점이 있다. 제안기법은 정수 연산방식과 난수발생을 기반으로 상호인증을 구현하였다. 또한, 세션상태값을 설정과 난수발생을 통해 Stephane등의 기법의 취약점인 비동기화 문제, 중간자공격, 전수조사공격등의 위험성을 해결하였다. 결론적으로 제안기법은 EPC의 규격 요건을 만족할 뿐만 아니라 보안 요구사항을 만족하는 경량화 기법으로 기존의 비트연산 방식의 대안이 될 것이라 예상한다.

[참고문헌]

[1] S.Weis, “Security and privacy in radio-frequency identification devices”, Master thesis MIT, 2003.

[2] M.Ohkubo, K.Suzuki and S.Kinoshita, "Cryptographic approach to privacy-friendly tag", RFID privacy workshop, MIT, 2003.

[3] M.Ohkybo, K.Suzuki and S.Kinoshita, "Forward-secure RFID privacy protocol using hash chain", NTT Laboratorie, 2003.

[4] D.Henrici and P.Muller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers", Second IEEE annual conference on pervasive computing and communications workshops, pp219-224, 2004.

[5] A.Juel and S.A. Weis, "Authenticating pervasive devices with human protocols", Advances incryptology-CRYPT2005, 293-308, Lecture notes in computer science, 3621, Springer, Berlin, 2005.

[6] L.Stéphane and T.L. Adrian, "Clone resistant mutual authentication for low-cost RFID technology", IACR Eprint, 2007.

[7] H.Y Chien and C.H. Chen, "Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards", Computer standards and interfaces, 2006.

[8] D.N. Duc, J. Park, H. Lee and K. Kim, " Enhancing security of EPC global GEN-2 RFID tag against traceability and cloning", The 2006 symposium on cryptography and information security, 2006.

[9] S. Karthikeyan and M. Nesterenko, "RFID security without extensive cryptography", proceedings of the 3rd ACM workshop on security of ad hoc and sensor networks, pp. 63 - 67, 2005.