효율적인 식별 기능을 가진 위조 불가 RFID Tag 가변 ID 방식

최 재 귀[†] · 박 지 환[††]

요 약

본 논문에서는 효율적인 식별 기능을 갖는 위조 불가 RFID 가변 정보화 방식을 제안한다. RFID 태그의 프라이버시 보호를 위해 제안된 대부분의 기존 방식들은 태그의 ID를 식별하기 위해 모든 태그에 대한 정보를 가지고 식별 과정을 수행해야 하는 비효율성을 가지고 있다. 또한 서버의 정보가 공격자에게 노출될 경우 태그의 위조도 가능하다는 문제가 있다. 본 논문에서는 2번의 지수 연산만으로 해당 태그의 ID를 식별할 수 있고, 서버의 태그에 대한 정보 노출에도 태그에 대한 위조가 불가능한 안전한 방식을 제안한다.

Unforgeable RFID Tag Variable ID Scheme with Efficient Identification

JaeGwi Choi[†] · JiHwan Park[††]

ABSTRACT

This paper proposes unforgeable RFID variable ID scheme with efficient identification. The existing schemes on privacy protection are inefficient because a server should execute identification process with all Tag ID's information in order to identify a certain Tag. Moreover these schemes have the serious problem that an attacker can forge special tags if he can know tag's secret information stored in the server's database. Our scheme is required only 2 times exponent computation to identify a tag. The proposed scheme is also secure against leakage of tags information stored in a database, because an attacker cannot forge special tag even if he knows secret information of the server(database).

키워드 : RFID, Privacy, 위조 불가(Forgery Identification)

1. 서 론

교통난과 유류비, 인건비의 상승으로 유통비용이 갈수록 증가하고 있는 요즘, 무선 인식 기술을 통한 유통비용과 재고관리 비용을 대폭 개선할 수 있는 기술인 RFID(Radio Frequency IDentification)가 개발되어, 사회 여러 분야로부터 큰 관심의 대상이 되고 있다. RFID란 초소형 반도체에 식별정보를 넣어 무선 주파수를 이용해 이 칩을 지닌 물체나 동물, 사람 등을 판독, 추적, 관리할 수 있는 기술로, 주차관리, 고속도로 요금 징수, 출입 통제, 원격 제어, 재고 관리 등 다양한 분야에 적용이 가능하다.

그러나 RFID의 많은 장점에도 불구하고, RFID의 프라이버시 침해적 요소는 이것의 사용 자체를 막고 있다[1]. 왜냐하면, 각종 물건들에 내장된 RFID 칩들은 거리 곳곳에 설치된 RFID 리더로 개인 정보를 전송할 것이고, 이러한 전송은 주파수라는 인간의 눈이나 귀로 인식되지 않는 매개체를 통하여 이루어지므로, 정부나 기업들은 정보주체의 동의 없이 언제, 어디서나 개인정보를 수집하거나 개인들의 일거수 일투족을 감시하는 것이 가능하게 되었기 때문이다. 이러한 환경에서 '익명성'이라는 말은 더 이상 실현 불가능한 말이 될 것이고, RFID 기술은 결국에는 'NO 프라이버시(No Privacy)' 사회를 만들어 낼 것이다. 이에 RFID의 프라이버시 침해적 요소들을 최소화하기 위한 여러 방법들이 제안되고 있다[2-7].

본 논문에서는 RFID의 프라이버시 문제를 해결하기 위한 기존의 방식들을 살펴보고, 이들의 문제점 분석을 통해 보다 효율적이고 안전한 방식을 제안하고자 한다. 본 논문의

† 준 회 원 : 부경대학교 대학원 정보보호학과
†† 종신회원 : 부경대학교 전자컴퓨터정보통신공학부 교수
논문접수 : 2004년 3월 22일, 심사완료 : 2004년 5월 12일

1) 의류회사인 베네통은 자사의 의류제품에 RFID 태그를 내장시켜 칩 속에 고객의 이름과 신용카드 정보, 그리고 의류 제품의 고유 번호를 함께 저장하여 회사가 지속적으로 자사 제품을 추적, 관리할 수 있도록 한다는 계획을 세웠으나, 프라이버시 운동가들의 극심한 반대에 부딪혀 재철회된 상태이다[1].

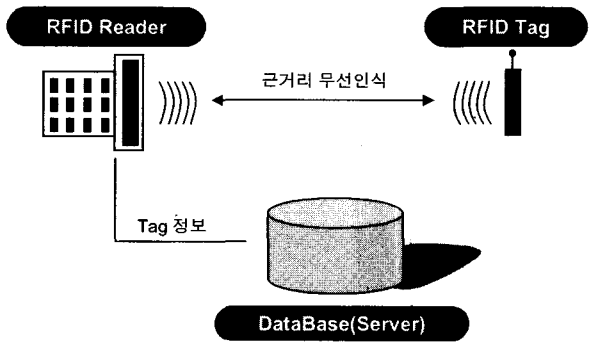
구성은 다음과 같다. 먼저 2장에서는 RFID 시스템 구성과 요구 조건을 기술하고, 3장에서 RFID 프라이버시에 관련된 기존 방식을 알아보고, 이의 안전성을 분석한다. 4장에서는 제안 방식의 프로토콜을 소개하고, 5장에서는 제안 방식의 안전성 및 특성 분석과 기존 방식과의 비교·분석을, 끝으로 6장에서 결론을 내리고자 한다.

2. RFID 기술 개요

2.1 RFID 구성

RFID는 (그림 1)과 같이 리더를 통하여 무선 통신에 의해서 접촉하지 않고 태그의 정보를 판독하거나 기록하는 일종의 무선 통신 시스템으로, 크게 안테나가 포함된 리더, 무선 자원을 송수신할 수 있는 안테나, 정보를 저장하고 프로토콜로 데이터를 교환하는 태그(Tag), 서버로 구성 된다 [8]. 각 부분의 기능은 아래와 같다.

- 태그 : 데이터를 저장하는 RFID 핵심 기능
- 리더 : 태그에 읽기와 쓰기가 가능하도록 하는 장치
- 안테나 : 정의된 주파수와 프로토콜로 태그에 저장된 데이터를 교환
- 서버 : 태그에서 전송된 정보의 복호 및 해독



(그림 1) RFID 시스템 구성도

Passive 태그 시스템의 리더는 RF 캐리어 신호를 태그에 송신하고 태그는 RF 신호가 들어오면 진폭 또는 위상을 변조하여 태그에 저장된 데이터를 캐리어주파수 신호로 리더로 되돌려준다. 되돌려 받은 변조 신호는 리더에서 복호되어 태그 정보가 해독되는데 리더는 보통 컴퓨터(서버)에 연결되어 운용되며 응용목적에 따라 운용 소프트웨어에 의해 RFID 시스템을 제어한다. 이러한 RFID는 이동 중에도 인식이 가능하고 장애물의 투과 기능도 가지고 있을 뿐만 아니라, 여러 개의 태그를 동시에 인식할 수 있고, 데이터의 인식속도도 타 매체에 비해 빠른 장점이 있다.

2.2 RFID 요구 조건

RFID가 보편화되기 위해서는 낮은 비용의 생산과, 빠른 인식 속도, 다중 태그의 인식 등의 시스템적 측면과 프라이버시 보호라는 안전성 측면이 동시에 만족되어져야 한다. 시스템적 측면은 RFID 시스템의 1차적 해결과제로 기술 개발과 시장 확산에 따라 만족되어 질 것으로 기대되나, 안전성 측면은 이것이 만족되지 못할 경우 RFID 기술의 사용 자체가 어려워지므로 시스템적 측면과 함께 반드시 해결해야 할 문제이다. 아래에 RFID 시스템이 갖추어야 할 기본적인 요구 사항을 기술한다.

2.2.1 시스템 측면

- 저비용(Low cost)

시장의 활성화를 위해서는 낮은 비용으로 RFID 태그를 제조할 수 있어야 한다.

- 효율적인 인식(Efficient Identification)

인식 속도가 빨라야 하며, 이동 중에도 인식이 가능해야 한다.

- 다중 태그 인식(Multiple Tag Identification)

다른 물체에 부착된 각기 다른 태그에 대해서도 동시에 인식할 수 있어야 한다.

2.2.2 안전성 측면

- 위조불가(Unforgeability)

태그에 부여된 유일한 ID에 대해서는 위조가 불가능해야 한다. 예를 들어 출입 통제 시스템에 이용될 RFID 시스템에서 인증된 태그와 동일한 정보를 전송하는 또 다른 태그(인증되지 않은)가 위조 가능하다면, 위조된 태그를 지닌 자동차도 출입이 허락되어 시스템 자체의 안전성을 위협할 것이다.

- 추적불가(Non Tracking)

리더가 읽어 들이는 태그 정보, 즉 태그로부터 나오는 정보로 소비자의 정보 즉, 소비 경향 또는 위치 추적이 불가능해야 한다. 예를 들어 제품 관리를 위해 타이어에 RFID 태그를 내장시킨다고 생각해보자. 이 시스템은 제품관리라는 원래의 의도와는 달리 RFID 태그 내장 타이어가 부착된 자동차의 위치 추적에 악용될 가능성이 있다. 즉 RFID 리더만 있으면, 추적하고자 하는 자동차가 언제, 어느 곳에 있는 지를 파악할 수 있으며, 궁극적으로는 자동차 소유주의 위치까지도 파악이 가능하게 된다. 이를 해결하기 위해서는 같은 태그라도 매회 마다 다른 정보를 전송해 주어야 하며, 현재의 전송된 정보를 이용하여 이전의 정보를 추적할 수 없어야 한다.

3. RFID Privacy 관련 연구

3.1 기존 방식

3.1.1 Kill command feature[2]

유일한 8비트 패스워드를 가지는 각 태그는 패스워드를 전송하자마자, 태그 스스로 자신의 정보를 삭제시키는 방식이다. 즉 어떤 물건에 부착된 태그는 고객이 물건을 구입한 이후에는 RFID 칩이 더 이상 작동되지 않도록 하는 방식이다. 이 방법은 완벽하게 프라이버시를 보호할 수는 있으나, 이 속성으로 인해 사실상 태그를 이용한 정보 관리나 태그의 재사용 등의 RFID 시스템의 장점을 제거해 버린다. 게다가 패스워드 길이가 8비트이므로, 공격자로 하여금 쉽게 패스워드 추측을 가능하게 만드는 문제가 있다.

3.1.2 Hash lock 방식[3]

각 태그는 다음과 같이 리더를 확인한다. 리더는 각 태그마다 유일한 키 (k)를 가지고 있으며, 이에 해당하는 태그는 $meta\ ID = H(k)$를 가지고 있다. 이 때 $H()$는 해쉬함수를 의미한다. 태그가 ID 접근 신호를 받으면,

- 태그는 자신의 $meta\ ID$를 리더에 보낸다.
- 리더는 이에 해당하는 키 (k)를 태그에 보낸다.
- 태그는 리더로부터 받은 키 (k)를 해쉬한 값과 자신의 $meta\ ID$를 비교하여, 그 값이 동일하면 자신의 ID를 전송한다.

이 방식은 해쉬 함수만을 요구하므로 저비용으로 구현될 수 있는 장점을 지니나, $meta\ ID$가 고정되어 있으므로, 공격자는 $meta\ ID$를 이용하여 해당 태그의 위치를 추적할 수 있는 문제가 있다.

3.1.3 Randomized hash lock 방식[4]

이 방식은 hash lock 방식을 개선한 것으로, 태그는 해쉬함수와 난수 생성기를 가져야 한다. 방식은 다음과 같다.

- 각 태그는 난수 생성기로부터 생성된 난수값 (r)과 자신의 ID를 연접하여 해쉬값, $c = H(ID \| r)$을 계산한 후, c, r을 리더에 전송한다.
- 리더는 이 값 (c, r)을 데이터베이스(서버)에 전송한다.
- 데이터베이스(서버)에는 각 태그의 ID가 저장되어 있으므로, 서버는 전송받은 c와 같은 값이 나올 때까지 모든 태그의 ID와 전송받은 r을 해쉬하여 c와 관련된 ID를 검색하고, 이것을 리더에 보낸다.

이 방식은 접근 때마다 태그에서 다른 출력 값이 나오므로, 태그에 대한 추적은 불가능하다. 그러나 서버에서 태그의 ID를 식별할 경우, 만족하는 ID가 나올 때까지 해쉬함수를 반복·수행해야 하므로, 평균 $N/2$번의 해쉬 함수 수행이 요구되는 단점이 있다. 이 때 N은 서버에 저장된 모든 태그의 수이다. 게다가 이 방식은 태그 안에서 해쉬함수와 함께 난수 생성이 이루어져야 하므로 저비용으로 구현하기도 어렵다.

3.1.4 Anonymous ID 방식[5]

이 방식은 익명의 ID를 출력함으로 태그의 실제 ID를 공격자가 알 수 없도록 하는 방법이다. 그러나 이 방식 역시 동일한 익명 ID가 출력되므로, 추적 가능이라는 문제는 해결할 수 없다.

3.1.5 Hash-chain 방식[6]

이 방식은 randomized hash lock 방식의 문제점, 즉 태그 안의 정보가 노출된다면 이전의 위치 경로가 추적된다는 것과 태그 안에서의 해쉬 함수와 난수 생성으로 인한 높은 비용을 개선한 것으로 프로토콜은 다음과 같다.

- 서버는 각 태그 ($T_t, t = 1, \cdots, m$)에 대해 랜덤 값 $s_{t,1}$을 생성하여 태그 (T_t)에 $s_{t,1}$을 저장하고, 자신의 데이터베이스에도 각 태그의 $(ID_t, s_{t,1})$를 저장해 둔다.
- 태그 (T_t)는 $a_{t,1} = G(s_{t,1})$를 계산한 후 $(a_{t,1}, 1)$을 리더에 보내고, 이전의 값 $s_{t,1}$를 $s_{t,i+1} = H(s_{t,i})$로 갱신해 둔다. 이 때, G, H는 다른 해쉬 함수이다.
- 리더는 태그로부터 받은 정보 $a_{t,1}$을 서버에 보내고, 서버는 $a_{t,1}$값이 나올 때까지 자신의 데이터베이스에 저장된 모든 태그의 $s_{t,1}$을 해쉬한 후, 해당 태그의 ID_t를 검출하여 리더에 보낸다.
- 두번째 접근부터 태그는 $(a_{t,k}, k)$ $\{a_{t,k} = G(s_{t,k}), k = 2, \cdots, n$를 보내고, 이전의 값 $s_{t,k}$를 $s_{t,k+1} = H(s_{t,k})$로 갱신해 둔다. 서버에서의 태그 ID 검출은 위와 동일하다. 단 이때 서버는 각 태그의 정보를 k번 해쉬하여야 한다.

이 방식은 태그 안에서 해쉬 함수만의 수행으로 태그 소유자의 프라이버시를 보호했다는 의의가 있다. 그러나 이 방식은 태그 안의 정보 노출이라는 최악의 상태를 고려하여 randomized hash lock 방식을 개선한 것으로, 태그의 정보가 노출될 경우 태그의 이전 위치는 추적 불가능하나, 노출 이후의 위치는 쉽게 찾을 수 있을 수 있으며, 태그의 위조도 가능한 문제가 있다. 또한, 서버에서 태그의 ID를 식별할 경우, 만족하는 ID가 나올 때까지

해쉬 함수를 계속해서 반복, 수행해야 되므로, randomized hash lock 방식보다 더 많은 식별 연산이 요구되는 문제도 있다.

3.1.6 Universal re-encryption 기반의 ID 가변 정보화 방식[7]

이 방식은 universal re-encryption 방식[9]과 one-time pad에 기반함으로 태그의 출력 값이 매회 다르게 출력되도록 만들고, 기존의 방식과는 달리 읽기 전용 리더와 one-time pad 갱신용 리더를 따로 두어태그 안의 계산량을 줄임으로 프라이버시 보호와 저비용의 태그 구현을 동시에 이루었다. 이 방식은 제안 방식의 배경이 되므로, 이 절의 나머지 장에서 보다 구체적으로 기술한다.

3.2 ID 가변 정보화 방식의 프로토콜[7]

[키 생성]

각 태그마다 비밀키 x_t와 이에 해당하는 공개키 $y_t = g^{x_t}$를 생성한다. 각 태그의 비밀키 x_t는 ID_t와 함께 서버의 데이터베이스에 저장해둔다. 본 방식은 El-Gamal 암호 방식에 기반하므로 g는 위수 $p-1$ (p : 소수)을 갖는 그룹 G상의 원시원소를 의미한다.

[암호화]

각 태그의 ID_t를 공개키 y_t, 난수 $r=(k_0, k_1)$을 이용하여 식 (1)과 같이 암호문 C를 계산하고, 데이터베이스에 (x_t, ID_t)을 저장해둔다.

$$C=[(\alpha_0, \beta_0);(\alpha_1, \beta_1)]$$
$$\alpha_0 = ID_t y_t^{k_0}, \beta_0 = g^{k_0}, \alpha_1 = y_t^{k_1}, \beta_1 = g^{k_1} \quad (1)$$

[One-time pad 생성]

데이터베이스(서버)는 각 태그의 암호문 C와 난수 $r=(l_1, \cdots, l_{2n})$에 의해 다음과 같이 one-time pad (Δ)를 생성해 둔다.

$$\Delta = [(\alpha_1^{l_1}, \beta_1^{l_1}), \cdots, (\alpha_1^{l_{2n}}, \beta_1^{l_{2n}})]$$

초기에는 생성된 one-time pad (Δ)와 암호문 C를 해당 태그에 저장시키고, 그 다음부터는 one-time pad 갱신용 리더를 통해 태그에 갱신, 저장시킨다.

[복호화]

판독용 리더를 통해 태그가 전송한 정보 C를 서버가 받으면, 서버는 해당 ID가 식별될 때까지 데이터베이스에 저장된 모든 태그의 비밀키를 이용하여 다음을 수행한다. 이때 해당 태그의 ID는 $\alpha_1/(\beta_1)^{x_t}=1$이면, $\alpha_0/(\beta_0)^{x_t}=ID_t$이다.

[재암호화]

다음 번에 리더로 자신의 신호를 보낼 때, 태그는 저장된 Δ에서 2개의 값 $(\alpha_1^{l_k}, \beta_1^{l_k}), (\alpha_1^{l_{k+1}}, \beta_1^{l_{k+1}})$을 선택하여 식 (2)와 같이 암호화한다.

$$C' = [(\alpha_0', \beta_0');(\alpha_1', \beta_1')]$$
$$\alpha_0' = \alpha_0 \alpha_1^{l_k}, \beta_0' = \beta_0 \beta_1^{l_k}, \alpha_1' = \alpha_1 \alpha_1^{l_{k+1}}, \beta_1' = \beta_1 \beta_1^{l_{k+1}} \quad (2)$$

one-time pad는 $2n$개이므로, 1번 갱신으로 n회 사용할 수 있으며, 이후에는 재사용하면 되나, 안전성을 위해 다음의 과정을 추가할 수도 있다.

[One-time pad 갱신]

One-time pad 갱신용 리더는 태그로부터 발생된 암호문 C를 서버에 보내고, 서버는 한 세션에 대한 count가 n번이 넘으면 그 때 one-time pad Δ'과 태그의 비밀정보 S, 세션 횟수 i를 해쉬한 값 $X=h(\Delta', i, S)$와 Δ'을 one-time pad 갱신용 리더를 통해 태그에 보낸다. 태그도 Δ'와 자신의 비밀정보와 세션 횟수를 이용한 해쉬 값과 전달받은 X값을 확인하여 같으면, one-time pad를 갱신하고, $i'=i+1$로 저장해 둔다. 이 경우에 서버의 데이터베이스에는 태그의 정보로 (x_t, ID_t) 외에도 (α_1, β_1)이 저장되어 있어야 한다.

3.3 ID 가변 정보화 방식의 문제점

ID 가변 정보화 방식은 태그 내부에서의 계산을 곱셈 4회와 해쉬 함수 1번(이 때의 해쉬함수 수행은 one-time pad 갱신 시 Δ'의 위조 여부 확인 위해 필요)으로 간략화시킴으로 낮은 비용의 구현을 가능하게 만들었다. 또한 매회 태그의 출력값이 다르므로, 태그의 위치 추적을 불가능하게 하므로, 프라이버시 보호 또한 만족한다. 그러나 이 방식은 서버에서 해당 태그의 ID를 식별할 경우, 전체 태그 수 (N) 만큼, 즉 평균 $N/2$만큼의 지수 연산을 필요로 한다.

또한 기존에 제안된 다른 방식[4-7]과 마찬가지로, 서버에 저장된 정보를 공격자가 획득할 수 있다면 태그의 위조 역시 가능하게 된다. 공격자가 데이터베이스에 저장된 특정 태그의 값인 ID_t, x_t를 얻었다고 가정하자. 그러면 공격자는 임의의 난수 (r_0, r_1)를 생성하여 식 (3)과 같이 암호문 (C_{red})을 리더를 통해 서버에게 보내고, 이 암호문은 서버에 의해 아무런 문제없이 복호되게 된다.

$$C_{red} = [(\alpha_{red_0}, \beta_{red_0}); (\alpha_{red_1}, \beta_{red_1})]$$

$$\alpha_{red_0} = ID_t y_t^{r_0}, \beta_{red_0} = g^{r_0}, \alpha_{red_1} = y_t^{r_1}, \beta_{red_1} = g^{r_1} \quad (3)$$

$$\frac{\alpha_{red_1}}{(\beta_{red_1})^{x_t}} = \frac{y_t^{r_1}}{g^{r_1 x_t}} = 1, \quad \frac{\alpha_{red_0}}{(\beta_{red_0})^{x_t}} = \frac{ID_t y_t^{r_0}}{g^{r_0 x_t}} = ID_t$$

4장에서 위에 언급한 ID 가변 방식의 문제점을 개선하고자 한다.

4. 제안 방식

[매개변수 및 시스템 설정]

- p : $q=(p-1)/2$인 소수, q : 소수
- G : 위수 $(p-1)$를 갖는 그룹
- g : 그룹 G의 원시원소
- x_t : 태그 t의 비밀키
- y_t : 태그 t의 공개키, $y_t = g^{x_t} \bmod p$
- x_s : 서버의 비밀키
- y_s : 서버의 공개키, $y_s = g^{x_s} \bmod p$

4.1 키 생성 단계

각 태그마다 비밀키 x_t와 이에 해당하는 공개키 $y_t = g^{x_t}$를 생성한다.

4.2 암호화 단계

- 서버는 각 태그마다 $x_{t,1} \cdot x_{t,2} = x_t$를 만족하는 비밀 랜덤 값 $(x_{t,1}, x_{t,2})$를 선택한 후, 식 (4)를 계산한다.

$$y_{t,1} = g^{x_{t,1}} \bmod p, \quad y_{t,2} = g^{x_{t,2}} \bmod p \quad (4)$$

- 서버는 난수 k_0을 이용하여 식 (5)와 같이 암호문 C를 계산하고, 자신의 데이터베이스에는 $(x_{t,1}, y_{t,1}, ID_t)$를 저장해 둔다.

$$C = [(\alpha_0, \beta_0, \gamma_0, \zeta_0)]$$

$$\alpha_0 = ID_t y_t^{k_0}, \beta_0 = y_{t,2}^{k_0}, \gamma_0 = y_{t,1} y_s^{k_0}, \zeta_0 = g^{k_0} \quad (5)$$

4.3 One-time pad 생성

서버는 각 태그의 암호문 C와 난수 k_1, $r=(l_1, \cdots, l_n)$에 의해 식 (6)과 같이 one-time pad (Δ)를 생성해 둔다.

$$\alpha_1 = y_t^{k_1}, \beta_1 = y_{t,2}^{k_1}, \gamma_1 = y_s^{k_1}, \zeta_1 = g^{k_1}$$

$$\Delta = [(\alpha_1^{l_1}, \beta_1^{l_1}, \gamma_1^{l_1}, \zeta_1^{l_1}), \cdots, (\alpha_1^{l_n}, \beta_1^{l_n}, \gamma_1^{n}, \zeta_1^{l_n})] \quad (6)$$

초기에는 생성된 one-time pad (Δ)와 암호문 C를 해당 태그에 저장시키고, 그 다음부터는 one-time pad 갱신용 리더를 통해 태그에 갱신, 저장시킨다.

4.4 복호화

- 판독용 리더를 통해 태그가 전송한 정보 C를 받으면, 서버는 자신의 비밀키 x_s를 이용하여 식 (7)을 계산하여, 해당 태그의 정보를 추출한다.

$$\frac{\gamma_0}{\zeta_0^{x_s}} = \frac{y_{t,1} y_s^{k_0}}{g^{k_0 x_s}} = \frac{y_{t,1} y_s^{k_0}}{y_s^{k_0}} = y_{t,1} \quad (7)$$

- 식 (7)에서 추출된 $y_{t,1}$에 해당하는 태그의 비밀정보 $x_{t,1}$을 이용하여 식 (8)을 계산한 후, 자신의 데이터베이스에 저장된 해당 태그의 ID_t와 같은 지 확인한다. 이 때 서버가 식 (8)을 통한 확인과정을 거치지 않고, $y_{t,1}$으로부터 바로 ID_t를 추출할 수도 있다. 그러나 이 경우에는 $y_{t,1}$을 얻은 공격자의 태그 위조가 가능하므로 이 과정은 반드시 수행되어야 한다.

$$\frac{\alpha_0}{\beta_0^{x_t}} = \frac{ID_t y_t^{k_0}}{y_{t,2}^{k_0 x_{t,1}}} = \frac{ID_t y_t^{k_0}}{y_t^{k_0}} = ID_t \quad (8)$$

4.5 재암호화

다음 번에 리더로 자신의 신호를 보낼 때, 태그는 저장된 Δ에서 1개의 set $(\alpha_1^{l_k}, \beta_1^{l_k}, \gamma_1^{l_k}, \zeta_1^{l_k})$을 선택하여 식 (9)와 같이 암호화한 후 C'를 보낸다.

$$C' = [(\alpha_0', \beta_0', \gamma_0', \zeta_0')]$$

$$\alpha_0' = \alpha_0 \alpha_1^{l_k}, \beta_0' = \beta_0 \beta_1^{l_k}, \gamma_0' = \gamma_0 \gamma_1^{l_k}, \zeta_0' = \zeta_0 \zeta_1^{l_k} \quad (9)$$

one-time pad는 n개의 쌍으로 이루어져 있으므로, 1번 갱신으로 n회 사용할 수 있으며, 기존의 방식과 마찬가지로 안전성을 위해 다음의 과정을 추가할 수도 있다.

4.6 One-time pad 갱신

One-time pad 갱신용 리더는 태그로부터 발생된 암호문 C를 서버에 보내고, 서버는 한 세션에 대한 count가 n번이 넘으면 그 때 one-time pad Δ'과 태그의 비밀정보 S, 세션 횟수 i를 해쉬한 값 $X = h(\Delta', i, S)$와 Δ'을 one-time pad 갱신용 리더를 통해 태그에 보낸다. 이 때 새로운 one-time pad, $\Delta' = [(\alpha_1^{m_1}, \beta_1^{m_1}, \gamma_1^{m_1}, \zeta_1^{m_1}), \cdots, (\alpha_1^{m_n}, \beta_1^{m_n}, \gamma_1^{m_n}, \zeta_1^{m_n})]$이다. 태그도 전송 받은 Δ'과 자신의 비밀정보, 세션 횟수를

이용한 해쉬 값과 전달받은 X값을 확인하여 같으면, one-time pad를 갱신하고, $i' = i+1$로 저장해 둔다. 이 경우에 서버의 데이터베이스에는 태그의 정보로 $(x_{t,1}, y_{t,1}, ID_t)$ 외에도 (α_1, β_1)이 저장되어 있어야 한다.

5. 제안 방식의 비교 · 분석

5.1 제안 방식의 특성

본 절에서는 제안 방식의 특성을 기술하며, 제안 방식에서 사용된 암호 알고리즘의 안전성은 universal re-encryption 방식[9]에 근거한다.

5.1.1 프라이버시 보호

제안 방식에서 태그의 출력이 항상 다르며, 어떤 일정한 시점의 출력 값으로부터 그 이전이나 이후의 값들을 유추할 없다. 따라서 태그와 리더 사이의 무선 통신 사이에 전달되는 값들로는 태그의 위치 추적이 불가능하므로 태그(소유자의 위치 추적)의 프라이버시를 보호할 수 있다.

5.1.2 태그의 적은 계산량

제안 방식에서 태그는 리더에게 정보를 보낼 때마다 4번의 곱셈만 수행하면 된다. 그리고 이에 추가적으로, one-time pad의 갱신을 위해 n번마다 1번씩 해쉬 함수를 수행하면 된다. 따라서 제안 방식은 적은 비용으로 태그를 생산할 수 있다.

5.1.3 효율적인 식별 기능

제안 방식에서 서버는 해당 태그의 ID를 검색하기 위해서 2번의 지수 연산만 수행하면 된다. 이는 기존 방식의 평균 $N/2$ 지수 연산(N : 태그의 수)을 2회로 줄임으로 태그의 식별 방식을 효율적으로 개선한 것이다.

5.1.4 태그에 대한 위조 불가

제안 방식에서 공격자가 서버의 데이터베이스에서 특정 태그의 정보, $ID_t, x_{t,1}, y_{t,1}$를 알 수 있다 하여도 이를 이용하여 태그를 위조할 수는 없다. 왜냐하면 태그를 위조하려면 식 (10)과 같은 암호문을 생성할 수 있어야 하는데, 공격자는 $ID_t, x_{t,1}, y_{t,1}$로부터 해당 태그의 $y_t, y_{t,2}$를 알 수 없기 때문이다.

$$C_{red} = [(\alpha_{red_0}, \beta_{red_0})]$$
$$\alpha_{red_0} = ID_t y_t^{r_0}, \beta_{red_0} = y_{t,2}^{r_0} \quad (10)$$

물론 공격자가 서버의 데이터베이스 안에 저장된 모든 태그의 정보를 알고 있고, 리더로부터 서버에 전송되는 모든 정보를 가져올 수 있다면, 제안 방식에서도 태그의 위조는 일어날 수도 있다. 하지만 리더와 서버간의 통신은 안전한 채널을 이용하므로, 이는 실현 불가능한 일이다. 따라서 제안 방식에서는 서버의 데이터베이스 정보가 노출된다 할지라도 공격자의 태그에 대한 위조는 불가능하다.

5.2 기존 방식과의 비교

본 절에서는 기존의 3가지 방식[4, 6, 7]과 제안 방식을 비교 분석하고자 한다. <표 1>은 제안 방식과 기존 방식을 비교, 분석한 것이다.

5.2.1 태그의 위조

제안 방식을 제외한 기존의 대부분의 방식[4-7]은 서버의 데이터베이스에 저장된 정보가 공격자에게 노출되면 태그의 위조가 가능하다.

Randomized hash lock 방식[4]에서는 공격자가 서버에 저장된 태그의 ID를 얻으면, 임의의 난수 (r')를 생성하여 $H(ID \| r'), r'$을 전송하면, 서버에서는 모든 태그의 ID와 전송받은 r'을 해쉬하여 태그를 인증하므로, 공격자는 쉽게 위조된 태그를 만들 수 있다.

Hash chain 방식[6]에서도 공격자가 서버에 저장된 태그의 비밀정보 $s_{t,1}$을 얻으면, 해쉬 함수 G, H를 이용해 쉽게 인증받을 수 있는 태그를 만들 수 있다. 왜냐하면 해당 태그의 출력 값은 $a_{t,1} = G(s_{t,1})$, $a_{t,2} = G(s_{t,2}) = G(H(s_{t,1}))$, …, 식으로 이루어지기 때문이다.

ID 가변 정보화 방식[7] 역시, 공격자가 서버에 저장된 태그의 비밀값 x과 ID를 안다면 3.3절에 언급되었듯이 제대로 복호되는 암호문을 생성할 수 있다. 이에 반해 제안 방식은 서버에 저장된 태그의 비밀 정보만으로는 제대로 인증되는 태그의 정보를 생성할 수 없으므로, 태그에 대한 위조가 불가능하다.

5.2.2 태그 ID 식별

기존의 대부분의 방식에서는 태그로부터 전송된 정보를 이용하여 해당 태그의 ID를 확인할 경우, 데이터베이스에 저장된 모든 태그에 대해 그 과정을 수행하여야 한다. 이에 반해 제안 방식은 2번의 지수 연산만 수행하면 해당 태그의 ID를 추출할 수 있으므로, 빠르게 식별할 수 있는 효율성을 가진다.

5.2.3 태그의 계산

제안 방식에서는 태그가 4번의 곱셈만 수행하면 되므로, ID 가변 정보화 방식[7]과는 태그의 계산량이 같고,

해쉬 함수를 수행하는 다른 2가지 방식[4,6]에 비해서는 보다 효율적으로 수행될 수 있다. 이 외에 제안 방식과 ID 가변 정보화 방식은 n번의 수행마다 해쉬 함수도 1번씩 수행해야 된다. 결과적으로 태그가 정보를 100번 출력할 경우 2가지 방식은 100번의 해쉬 함수 수행이 요구되는 반면, ID 가변 정보화 방식과 제안 방식은 400번의 곱셈과 $100/n$번의 해쉬 함수가 요구된다. 따라서 태그의 계산량면에서는 제안 방식이 효율적이라 볼 수 있다. 그러나 제안 방식과 ID 가변 정보화 방식은 태그 안의 계산량을 줄이기 위해, one-time pad 갱신용 리더를 이용하므로, 다른 2가지 방식에 비해 부가적 비용이 요구되는 것도 사실이다.

〈표 1〉 기존 방식과 제안 방식의 비교 분석

	Randomized hash lock	Hash chain	ID 가변 정보화	제안 방식
구성 개체	태그, 리더, 서버	태그, 리더, 서버	태그, 리더 2개, 서버	태그, 리더 2개, 서버
프라이버시 보호	보호(위치추적불가)	보호(위치추적불가)	보호(위치추적불가)	보호(위치추적불가)
태그의 계산량	해쉬 1번, 난수 생성 1번	해쉬 1번	곱셈 4번, n 회마다 해쉬 1번	곱셈 4번, n 회마다 해쉬 1번
식별 과정의 계산량	평균 $N/2$ 해쉬	평균 $iN/2$ 해쉬	평균 $N/2$ 지수연산	2번의 지수연산
위조 여부	가 능	가 능	가 능	불가능

N : 전체 태그의 수 , i : i번째 태그의 출력, n : one-time pad의 구성 세트 수

6. 결 론

본 논문에서는 태그 소유자(태그)의 프라이버시를 보호하기 위한 기존 방법들을 분석함으로, 기존 방식들이 비효율적인 식별과 태그의 위조 가능이라는 문제가 있음을 지적하였고, 이를 해결한 방식을 제안하였다. 기존의 방식들이 태그의 ID를 식별할 경우 모든 태그에 대해 연산을 수행한 반면, (<표 1> 참조), 제안 방식은 2번의 지수 연산으로 해당 태그를 식별할 수 있다. 또한 제안 방식은 서버에 저장되어 있는 태그의 비밀정보가 공격자에게 노출된다 할지라도 태그의 위조가 불가능하므로, 기존의 방식에 비해 안전성이 높은 방식이라 할 수 있다. 그러나 제안 방식은 one-time pad 갱신용 리더와의 교신이 추가적으로 필요하므로, 이에 대한 개선이 요구된다. 또한 제안 방식을 포함한 기존의 대부분의 방식이 태그로부터 리더로 전송되는 정보 유출에는 안전하지 않다는 취약점을 가지므로, 태그의 리더에 대한 인증 문제도 차후 연구 과제로 요구된다.

참 고 문 헌

[1] 한국정보보호진흥원, 정보보호 뉴스 2003년 10월호 통권 73호, 2003.

[2] Auto-ID Center, "860MHz-960MHz Class I Radio Frequency Identification Tag Radio Frenquency & Logical Communication Interface Specification Proposed Recommendation Version 1.0.0," Technical Report MIT-AUTOID-TR-007, 2002.

[3] Stephen A. Weis, "Security and Privacy in Radio-Frenquency Identification Devices," Maters thesis, MIT. http://theory.lcs.mit.edu/~sweis/masters.pdf, 2003.

[4] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest and Daiel W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," First International Conference on Security in Pervasive Computing, LNCS 2802, Springer-Verlag, pp.201-212, 2003.

[5] S. Kinosita, F. Hoshino, T. Komuro, A. Fuhimura and M. Ohkubo, "Non-identifiable Anonymous ID-Scheme for RFID Privacy Protection," Computer Security Symposium 2003, IPSJ Symposium Series, Vol.2003, No.15, pp.497-502. 2003.

[6] M. Ohkubo, K. Suzuki and S. Kinoshita, "Cryptographic Approach to "Privacy-Friendly" Tags," RFID Privacy Workshop, http://www.rfid.edu.com, 2003.

[7] J. Saito and K. Sakurai, "Variable ID Scheme of Anonymity in RFID tags," The 2004 Symposium on Cryptography and Information Security, Vol.I, pp.713-718. 2004.

[8] 장동원, 조평동, "RFID 기술기준 도입을 위한 기술 분석", 전자통신동향분석, 제18권 제6호, pp.59-67, 2003.

[9] P. Golle, M. Jakobsson, A. Juels and P. Syverson, "Universal Re-encryption for Mixnets," CT-RSA 2004, LNCS 2946, Springer-Verlag, pp.163-178, 2004.

최 재 귀

e-mail : jae@mail1.pknu.ac.kr

1998년 부경대학교 전자계산학과(학사)

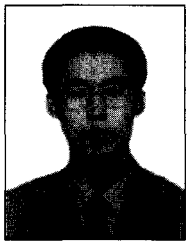
2001년 부경대학교 교육대학원 전산교육 전공(교육학석사)

2002년~2003년 Kyushu Univ. 교환학생, JSPS 지원

2002년~현재 부경대학교 대학원 정보보호학과 박사과정

2004년~현재 Tokyo Univ. Internship, KOSEF 지원

관심분야 : 정보보호, 저작권 보호, 디지털 핑거프린팅



박 지 환

e-mail : jpark@pknu.ac.kr

1990년 요코하마국립대학 전자정보공학 (공학박사)

1994년~1995년 동경대학 생산기술연구소 방문연구

1998년~1998년 일본 전기통신대학 방문 연구

1999년~1999년 Monash University, Australia, 방문연구

2001년, 2003년 Communication Research Lab. Japan, JSPS Fellowship

1990년~현재 부경대학교 전자컴퓨터정보통신공학부 교수

1996년~현재 동경대학 생산기술연구소 협력연구원

1997년~현재 정보보호학회 이사, 논문지 편집위원

1998년~현재 멀티미디어학회 운영위원, 논문지 편집위원

1999년~현재 정보처리학회 논문지 편집위원

2004년~현재 방송공학회 논문지 편집위원

관심분야 : 정보보호 및 암호학, 멀티미디어 압축 및 응용