OTP 를 이용한 RFID 상호인증 프로토콜

이영실*, 장원태**, 이훈재**
*동서대학교 일반대학원 유비쿼터스&IT 학과.
**동서대학교 정보공학부
e-mail : youngsil.lee0113@gmail.com, hjlee@dongseo.ac.kr

RFID mutual authentication protocol using OTP

Young Sil Lee*, Won Tae Jang**, Hoon Jae Lee**
*Dept. of Ubiquitous and IT, Graduate School of General, Dongseo University
**Div. of Computer and Information Engineering, Dongseo University

요 약

RFID(Radio Frequency Identification) 시스템은 비접촉식 무선 인식 기술로 유통 및 물류, 환경, 교통, 보안 분야 등 산업 전반에 걸쳐 다양하게 활용되고 있다. 그러나 태그의 정보가 전송과정에서 무선특성에 따른 과도한 정보 노출과 사용자의 위치정보 추적 등 심각한 프라이버시 침해를 유발시킨다. 본 논문에서는 해쉬된 ID 와 스트림 암호 알고리즘을 이용한 OTP 를 활용하여 리더와 태그 간 상호인증을 제공하는 프로토콜을 제안한다. 제안된 프로토콜의 OTP 생성에 사용된 NLM-128 알고리즘은 2^{128} 비도 수준(Security level)을 갖는 스트림 암호로써, 안전성 및 구현 용이성 등의 특징을 가지며 RFID/USN 등의 저전력, 제한된 메모리 및 컴퓨팅 사양에서 적용하기 용이한 알고리즘이다.

1. 서론

RFID 란 무선인식 기술 즉 Micro-chip 을 내장한 태그(Tag), 레이블(Lable), 카드(Card) 등에 저장된 데이터를 무선 주파수를 이용하여 리더(Reader)에서 자동으로 인식하게 하는 기술을 말한다. 또한 칩의 저장능력과 인식능력이 향상되면서 이 무선인식 기술은 유비쿼터스 환경에서 필수적인 기술로 각광받고 있다. RFID 기술은 단순한 바코드의 대체 수준을 넘어서 통신, 물류, 국방, 소방, 금융, 의료, 환경, 교육, 정보가전, 도로, 건설 등 다양한 인간의 생활 전반에 활용되어 무한한 부가가치를 창출 가능하여, 향후 전 세계적인 산업구조, 시장구조의 변화뿐만 아니라 인간의 삶의 형태까지 변화시키게 될 유비쿼터스 컴퓨팅의 기반 기술로서 인식되고 있다.

하지만 RFID 기술은 리더와 태그 사이에 물리적인 접촉 없이 인식 가능하고 태그의 정보가 전송과정에 무선특성에 따른 과도한 정보 노출과 사용자의 위치정보 추적과 같은 심각한 프라이버시 침해를 유발시킨다. 이러한 우려들이 RFID 의 상용화에 걸림돌이 되며, 성공적인 산업화를 위해서는 제반 프라이버시 문제를 해결해야 하는 것이 선결 과제로 되고 있다[1].

본 논문에서는 해쉬된 ID 와 NLM-128 스트림 암호 알고리즘[2]을 이용하여 생성된 OTP 를 활용하여 리더와 태그 간 상호인증을 제공하는 프로토콜을 제안한다.

논문의 구성은 다음과 같다. 2 장에서는 기존 인증 프로토콜[3]에 대하여 분석하고, 3 장에서는 제안 프로토콜에 대하여 기술, 4 장에서는 제안 프로토콜의 안전성과 효율성에 대하여 비교평가 하였으며 마지막 5 장에서 결론을 맺도록 한다.

2. 관련연구

2.1. 기존 RFID 인증 프로토콜.

현재까지 RFID 시스템에서 사용자의 프라이버시를 보호하기 위하여 여러 가지 기법들이 제안되었다. 이런 기법들은 크게 물리적 접근 방법과 암호학적 접근 방법으로 분류할 수 있으며, 물리적 접근 방법은 태그를 무효화시키는 Kill 명령어, Faraday Cage 기술, Blocker Tag 및 Active Jamming 기술 등의 방법이 있다. 암호학적 접근 방법으로는 비트연산(XOR), 재 암호화, 해쉬 함수 기반 및 전통적인 암호화(대칭키 및 공개키) 기반 기법 등이 있다. 최근의 홈 네트워크나 유비쿼터스 환경에서는 주로 암호학적 접근기법을 사용하고 있다.

암호학적 접근 방법 중 해쉬 함수 기반의 알고리즘은 일방향 해쉬 함수의 역함수 계산 어려움에 기반하여 프라이버시, 추적 등의 보안 문제를 해결하기 위한 적합한 알고리즘이나 표준 SHA 계열의 해쉬 함수를 하드웨어로 구현하면 8,000~10,000 gate 이상이 소요된다. 따라서 현실적으로 수천 게이트 정도의 자원만 사용 가능한 RFID 태그에서는 구현하기에는 커다란 도전이 되고 있다. RFID 태그에 구현 가능한 초경량 해쉬 함수가 있다면 구현 가능한 기술로 Weis 등[4]이 제안한 Hash-lock 프로토콜로 인하여 이후에 RFID 인증 프로토콜의 많은 프로토콜들이 초경량 해쉬 함수를 기반으로 하는 계기가 되었다. 한편 공개키 기반 알고리즘은 키 분배 및 키 관리 문제를 고려

할 때 적합한 암호 알고리즘으로, Rabin, NTRU, ECC 등의 공개키 기반으로 한 하드웨어 구현에 활발한 연구가 이루어지고 있다. 그리고 대칭키 암호 알고리즘은 미국 표준 블록 암호 AES 를 3,600 gate 구현에 성공한 사례가 발표되었고, 또 SHA1 과 같은 해쉬 함수보다 AES 대칭키 암호가 저전력 설계에 더 적합함이 다양한 논문을 통해 입증되었다. Feldhofer 등의 논문에서 대칭키 AES-128 은 어떤 해쉬 함수보다 더 RFID 시스템에 적합하다고 제안한 바 있다. 구본석은 Feldhofer 보다 진일보한 암•복호화가 가능한 초소형 AES 연산기를 3,992 gate 로 구현한 바 있다. AES 외에도 mCrtton, HIGHT[5] 및 PRESENT 등의 경량화 블록 암호화(대칭키 시스템) 시스템이 3,000 gate 이하에서 구현됨으로 자원 제약이 심한 저가(Low-cost)의 RFID 태그에 여유 있게 적용될 수 있음을 알 수 있다. 또한 Osaka 은 RFID 시스템의 소유권 이전 기법에 해쉬 함수와 대칭키 암호 시스템을 병행하여 사용하여, 인증 시에는 해쉬 함수를, 그리고 키 이전 시에는 대칭키 암호 기법을 활용하는 예를 보여주었다. Toiruul 은 랜덤 비밀키 k_1, k_2 을 데이터베이스와 태그가 공유하는 대칭키 AES 를 이용한 상호인증 기법을 제시한 바 있다.

한편, 물리적 공격에 의해 노출될 가능성을 방지하기 위해 PUF(Physically Unclonable Function)를 활용하여 태그의 복제를 방지하는 방법에 대한 연구가 활발히 진행되고 있다. Tuyls 은 RFID 태그의 복제 방지를 위해서 PUF 및 공개키 암호를 사용하여 인증 프로토콜을 제안한 바 있고, 최근 Kulseng 은 경량 검색 프로토콜에서 PUF 및 LFSR 을 활용함으로 태그의 경량화를 증진시켰다. 여기에 최근 인증프로토콜의 경향을 보면, 태그와 리더 간에 교환되는 메시지를 보호하기 위해 난수 생성기를 적극 활용하고, RFID 시스템의 보다 경량화된 인증기법을 도입하며, 최소한의 암호학적 접근 방식 등을 활용함을 알 수 있다.

2.2. NLM-128 Stream Cipher.

NLM-128 은 새로운 스트림 암호인 NLM 시리즈 중 하나로, 127 bit 의 LFSR 한 개와 129 bit 의 NFSR 하나로 구성된다. 또한 이들을 합쳐 256 bit 의 내부 메모리를 가지며, 128 bit 키와 128 bit 의 초기화 값으로 내부 메모리를 채우게 된다. 그리고 NLM 생성기는 LFSR, NFSR 수열과 carry 와 메모리 수열의 결합에 의해 키 수열이 출력되며, 원시 다항식 $P_a(x)$ 와 de Bruijn 의 기약 다항식인 $P_b(x)$, 두 개의 다항식을 가지게 된다.

$$P_a(x) = x^{127} \oplus x^{109} \oplus x^{91} \oplus x^{84} \oplus x^{78} \oplus x^{67} \oplus x^{66} \oplus x^{63} \oplus x^{56} \oplus x^{28} \oplus x^{25} \oplus x^{23} \oplus x^{21} \oplus x^{20} \oplus x^{19} \oplus x^{16} \oplus x^{13} \oplus x^{12} \oplus x^{7} \oplus x^{6} \oplus x^{2} \oplus x^{1} \oplus 1$$

$$P_b(x) = x^{128} \oplus x^{122} \oplus x^{121} \oplus x^{117} \oplus x^{113} \oplus x^{109} \oplus x^{102} \oplus x^{101} \oplus x^{97} \oplus x^{93} \oplus x^{89} \oplus x^{82} \oplus x^{77} \oplus x^{73} \oplus x^{69} \oplus x^{62} \oplus x^{57} \oplus x^{53} \oplus x^{49} \oplus x^{42} \oplus x^{41} \oplus x^{37} \oplus x^{33} \oplus x^{29} \oplus x^{18} \oplus x^{9} \oplus \left(\prod_{i=1}^{128} x^{i} \right)$$

두 개의 키 값 k 와 초기화 벡터 iv 는 각각 128 bit 의 크기를 가지고 함께 256 bit 의 내부 메모리를 가지게 되며 또한, 초기화 프로세스는 키 재생성에 사용된다. 키 수열 생성기의 초기 상태를 생성하기 위하여 자체적으로 생성기를 두 번 사용하며, L_a 의 연산을 시작하게 되면 $L_a = (k \oplus iv) \bmod 2^{127}$ 식에 따라 128 bit 의 키(Key) 값 k 와 128 bit 의 초기화 값 iv 를 XOR 하여 L_a 의 값을 얻게 된다.

초기화시킨 NLM-128 알고리즘을 통하여 256 bit 의 출력 문자열을 생성한다. 암호의 재사용을 위하여, 이 출력 문자열의 처음 128 bit 를 L_a 의 초기 상태를 채우는데 사용하고 나머지 129 bit 는 L_b 의 초기 상태를 채우는데 사용한다. 두 번째로 실행되는 암호 알고리즘은 257 bit 길이의 문자열을 출력하며, 이를 이용하여 다음 번 암호화 실행 시 새로운 키 수열 생성을 위하여 키 수열의 초기 상태로 사용한다. 이때 사용된 NFSR B 레지스터는 키 수열 발생기의 비선형성을 높이기 위하여 De Bruijn 수열을 사용하였으며, Park & Jang 이 제시한 효과적인 de Bruijn 수열 발생기를 사용하였다.

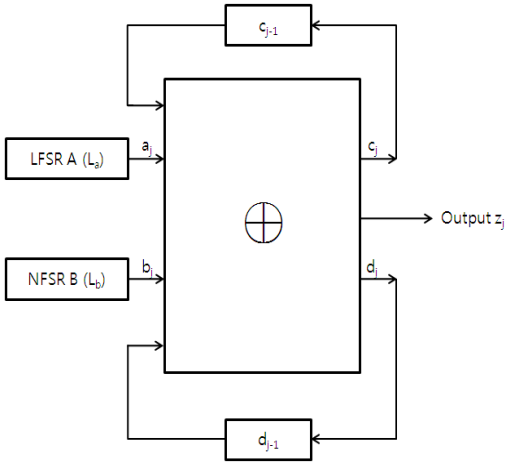

그림 1. NLM-128 암호 알고리즘

3. 제안 기법.

본 논문에서 제안하는 사용자 인증 프로토콜은 해쉬된 ID 인 H(ID)와 공유 ID(ID_s)를 이용하여 NLM-128 암호 알고리즘을 이용하여 생성된 OTP 를 인증에 사용한다.

3.1. 제안하는 프로토콜 설명.

아래의 그림 5 는 제안하는 인증 프로토콜의 인증 과정을 설명한 것으로 각 단계별 설명은 다음과 같다.

[1 단계] 리더는 태그에게 Query 와 함께 자신이 생성한 랜덤한 값 R 을 연접하여 전송한다.

[2 단계] 태그는 전송받은 랜덤 값 R 과 자신의 해쉬된 태그 ID 인 H(ID)를 키 값으로 NLM-128 알고리즘을 이용하여 OTP 값을 생성한다.

[3 단계] 생성된 OTP 를 리더에게 전송하다.

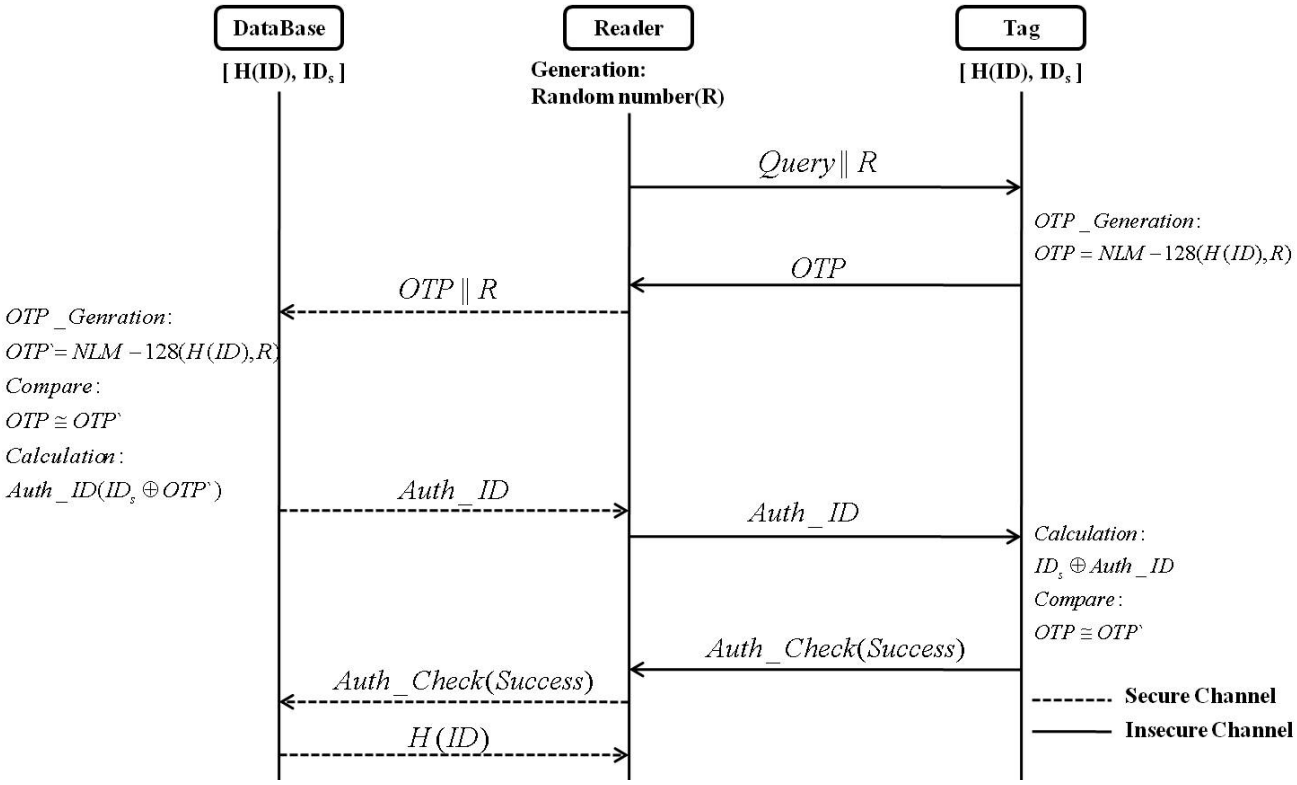

그림 2. 제안 프로토콜

[4 단계] 리더는 전송 받은 OTP 와 자신이 생성한 랜덤한 값 R 을 연접하여 DB 에게 전송한다.

[5 단계] DB 는 전송 받은 랜덤 값 R 과 자신이 가진 해쉬된 태그의 ID 인 H(ID)를 키 값으로 NLM-128 알고리즘을 사용하여 OTP 값인 OTP`을 생성하다.

[6 단계] 자신이 생성한 OTP`과 리더로부터 전송 받은 OTP 를 비교한다. 비교한 값이 일치한다면 자신이 가진 공유 ID 인 $ID_s$ 와 생성한 OTP 값인 OTP`을 XOR 연산하여 Auth_ID 를 생성하고 리더에게 전송한다. 비교한 값이 일치하지 않는다면 인증을 중단한다.

[7 단계] 리더는 전송 받은 Auth_ID 를 태그에게 전송한다.

[8 단계] 태그는 리더로부터 전송 받은 Auth_ID 와 자신이 가진 공유 ID 인 ID_s를 XOR 연산하여 OTP`값을 얻고, 이 값을 자신이 계산한 OTP 값인 OTP 와 비교한다. 비교한 값이 일치한다면 리더에게 Auth_Check(Success) 를 전송하고, 일치하지 않는다면 인증 과정을 중단한다.

[9 단계] 리더는 태그로부터 전송 받은 Auth_Check(Success)를 DB 에게 전송한다.

[10 단계] DB 는 리더에게 전송 받은 Auth_Check(Success)를 확인하고, 리더에게 태그의 해쉬된 태그의 ID 인 H(ID)를 전송한다.

3.2. 제안 프로토콜의 상호 인증.

태그와 리더간 상호인증은 그림 5 에서 제안 프로토콜의 동작과정에서 자동으로 이루어진다.

- 리더가 태그에 대한 인증: 위의 프로토콜 동작과정 [5 단계]에서 DB 는 전송 받은 OTP 와 랜덤 값 R 을 이용하여 OTP`을 생성하고, OTP 를 비교한다. OTP 값이 일치하지 않으면 위장된 태그로 인정하여 프로토콜을 중단하고, 일치한다면 정당한 태그로 인정한다.

- 태그가 리더에 대한 인증: 위의 프로토콜 동작과정 [8 단계]에서 태그는 자신이 가진 공유 ID 인 ID_s 를 이용하여 전송 받은 Auth_ID 와 XOR 계산하여 DB 에서 계산한 OTP`값을 구하고, 자신의 OTP 와 비교하여, 동일하다면 정당한 리더로, 아니면 위장된 리더로 인정한다.

4. 안전성 및 효율성.

본 장에서는 다양한 공격유형에 대하여 제안한 인증 프로토콜의 안전성에 대하여 기술한다.

4.1. 안전성 분석.

1) 재전송 공격(Replay Attack)

공격자는 리더로 위장한 재전송 공격과 태그로 위장한 재전송 공격 두 가지 경우가 있다. 리더로 위장한 경우 공격자는 리더에서 태그로 전송되는 메시지를 도청하여 재전송하는 경우인데, 제안 프로토콜에서는 OTP 를 이용하여 리더와 태그 간에 상호인증을 한다. 따라서 제안한 프로토콜은 재전송 공격에 안전하다.

2) 도청(Eavesdropping)

본 논문에서 제안하는 프로토콜은 서버와 리더 사이에 안전한 채널을 통한 통신을 가정하고, 리더와 태그 사이는 안전하지 못한 채널이라고 가정한다. 다라서 리더와 태그 사이의 모든 단계에서 도청이 가능

하다. 하지만 제안한 프로토콜에 의해 매 세션마다 항상 다른 값을 전송하기 때문에 공격자가 도청을 하여 얻은 정보만으로는 값을 유추하거나 다른 공격에 활용할 수 없다.

3) 위치추적(Location Traceability)

태그가 리더에게 보내는 값이 고정되어 있는 경우 태그를 소지하고 있는 사용자나 태그가 부착된 물품의 위치 추적이 가능하다. 제안한 프로토콜에서는 리더와 태그 사이의 모든 과정에서 태그는 OTP 를 사용하여 항상 다른 값을 리더에게 전송하므로 위치 추적의 문제를 해결할 수 있다.

4) OTP 생성 알고리즘의 안전성.

NLM-128 스트림 암호의 설계 기준 강도는 2256 이며, TMTO 공격에 대한 안전성을 고려하면 2128 이 된다. 여러 가지 공격에 대하여 기본적인 키 수열 특성은 큰 선형 복잡도 및 긴 주기 때문에 안전하다 볼 수 있으며, 선형 복잡도와 주기의 방정식은 아래의 수식과 같다.

$$P = (2^{L_1} - 1)(2^{L_2})$$
$$LC \approx (2^{L_1} - 1)(2^{L_2})$$

4.2. 효율성 분석.

기존의 제안된 프로토콜과 제안한 프로토콜의 효율성을 비교하기 위해 태그에서의 Hash 함수 작동 횟수, 태그에서의 암•복호화 작동 횟수, 태그에서의 XOR 연산 횟수, 태그에서의 랜덤 넘버 생성 횟수, 인증을 위한 서버의 필요 여부를 비교 분석한 결과를 <표 1>에 나타내었다.

Ref[4]의 경우 태그에서의 연산량은 한번의 Hash 함수 계산과 랜덤 넘버 생성뿐이지만 안전성 측면에서 상당히 취약함이 밝혀졌으며, Ref[6]의 경우 태그에서 Hash 함수 계산을 2 번 수행하나 역시 안전성 측면에서 취약함이 밝혀졌다. Ref[3]와 Ref[5]의 경우 대칭키 기반의 프로토콜을 사용하고 있으며, Ref[4]와 Ref[6]보다 안전성을 제공하고 있다. 따라서 Ref[3]와 Ref[5]와 본 논문의 제안한 프로토콜을 비교한다. 태그에서 수행되는 암•복호화 횟수가 각각 3E+1D:3E 이다. 본 논문에서 제안하는 프로토콜은 OTP 생성을 위한 1 번의 암호 알고리즘의 사용으로 복호화 또한 필요치 않으므로 1E 라 할 수 있다. 한편 태그의 XOR 연산의 횟수를 비교해보면 Ref[3]와 Ref[5]의 경우 각각 1 번과 4 번, 제안된 프로토콜의 경우 1 번의 XOR 연산을 수행한다. 그리고 제안된 프로토콜의 경우 태그에서는 랜덤 넘버 생성을 하지 않으며, 인증을 위해 서버의 자원을 사용하나, 효율성 측면에서 기존의 프로토콜보다 뛰어남을 알 수 있다.

표 1. 기존 프로토콜과의 효율성 비교분석

구분	Ref [3]	Ref [4]	Ref [5]	Ref [6]	Our Protocol
Hash 함수 작동 수(tag)	-	1	-	2	-
암호 알고리즘 작동 수(tag)	3E+1D	-	3E	-	1E
XOR 연산(tag)	1	-	4	2	1
랜덤 넘버 생성 수(tag)	2	1	0	0	0
인증 시 서버 필요 여부	X	X	○	○	○

5. 결론

RFID 기술은 기존 바코드 기술에 비해 인식속도가 빠르고, 저장 공간이 크며 무선인식 등 장점을 갖고 있어 사회의 각 분야에 많이 활용되고 있다. 그러나 기존 RFID 시스템은 비용문제, 무선 환경의 보안 취약성 및 프라이버시 침해와 같은 새로운 보안문제점이 발생하고 있다.

본 논문에서는 해쉬된 태그 ID 와 스트림 암호 알고리즘을 이용하여 생성한 OTP 를 사용해 리더와 태그 간 상호인증 가능한 RFID 프로토콜을 제안하였다. 제안한 프로토콜의 OTP 생성에 사용된 암호 알고리즘은 2^{128} 비도 수준(Security Level)을 갖는 스트림 암호로써, 안전성 및 구현 용이성의 특징을 가지며 RFID/USN 등의 저전력, 제한된 메모리 및 컴퓨팅 사양에서 적용하기 용이한 알고리즘이다. 이 인증 프로토콜은 도청, 위장 및 프라이버시 침해와 같은 문제점을 해결하였으며, 기존 프로토콜과의 비교분석을 통해 효율성 측면에서 우수함을 확인할 수 있다.

감사의 글

이 논문은 2010 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(과제번호:20100010488).

참고문헌

[1] S. A. Weis, “Radio-frequency identification security and privacy”, Master’s thesis, M.I.T, 2003.

[2] H.J. Lee, S.M. Sung, H.R. Kim, “NLM-128, An Improved LM-type Summation Generator with 2-bit memories”, ICCIT09, pp.577-582, 2009.

[3] 박용수, 신주석, 최명실, 정경호, 안광선, “해쉬된 태그 ID 와 대칭키 기반의 RFID 인증 프로토콜”, 정보처리학회 논문지, 제 16-C 권, 제 6 호, 2009.12.

[4] S. A. Weis, S. E. Sarma, R. L. Rivest, D. W. Engels, “Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems”, Security in Pervasive Computing-SPC2003, LNCS 2802, pp.719-724, 2004.

[5] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonsek Koo, Changoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyung Kim, Jongsung Kim, and Seongtaek Chee, “HIGHT: A New Block Cipher Suitale for Low-Resource Device”, CHES 2006, LNCS 4249, pp.46-59, 2006.

[6] S. Kang, D. Lee, and I. Lee, “A study on secure RFID mutual authentication scheme in pervasive computing environment”, Computer Communications 31, pp.4248-4254, 2008.

[7] Jin-Oh Jeon, Min-Sup Kang, “Security Enhancing of Authentication Protocol for Hash Based RFID Tag”, 한국인터넷 정보학회, 제 11 권 4 호, pp.23-32, 2010.08.