RFID Tag를 위한 개선된 인증 프로토콜 설계

이 광호°, 손 명진, 강 민섭
안양대학교 컴퓨터공학과
e-mail:mskang@anyang.ac.kr

Design of Modified Authentication Protocol for RFID Tags

Kwang-Ho Lee°, Myoung-Jin Son, Min-Sup Kang
Department of Computer Engineering, Anyang University

요 약

본 논문에서는 RFID Tag을 위한 개선된 인증 프로토콜의 설계 및 검증에 관하여 기술한다. 제안한 프로토콜은 ISO/IEC 18000 standard를 기본으로하고 있으며, 강인한 인증을 위해 표준 프로토콜 frame format 을 수정한다. 상호 인증을 위해 three-way challenge response 프로토콜을 사용하며, 인증 알고리듬은 SHA-1이 추가되었다. 제안한 프로토콜의 검증을 위해 Xilinx ISE 6.2i 툴을 사용하여 RFID Tag의 디지털 part를 설계하였고, Virtex Xcv4000 FPGA를 타겟으로 합성을 수행하였다. RFID Tag의 디지털 part는 Mentor's Modelsim을 이용하여 시뮬레이션을 수행하였고, 동작속도는 약 75MHz를 가지며, 1290개의 슬라이스가 사용되었다.

1. 서론

RFID는 초소형 IC 칩에 식별정보를 입력하고 무선주파수를 이용하여 이 칩을 지닌 물체나 동물 또는 사람 등을 인식·추적·관리할 수 있는 기술이다. IC 칩은 무선 안테나와 함께 물체에 쉽게 부착될 수 있도록 다양한 모양과 크기의 RFID 태그에 내장된다. 이러한 태그의 정보는 판독기(Reader)라 불리는 무선 단말기에 의하여 읽혀지고 네트워크에 연결된 컴퓨터에서 데이터 처리가 이루어진다[1].

RFID 환경에서 안전하지 않는 Tag를 사용하는 경우 물리적 공격, 위조, Spoofing, 도청, 트래픽 분석, DOS 공격 등에 의한 보안적 취약점에 노출되어진다. 이러한 취약점을 방지하기 위해 RFID Tag에 저장 관리되는 식별 정보를 보호하기 위한 다양한 방법이 제안되고 있다[2].

초기 Tag 식별정보로 Key를 해쉬한 MetaID 정보만을 제공함으로 실제적인 ID 정보의 유통을 방지하기 위한 Hash Lock scheme이 제안되었으나, MetaID정보 역시 식별 정보로서 추적이 가능함으로 실제적인 서비스에 적용하기 어려운 단점이 있다[3].

이러한 문제점을 개선하여 해쉬 알고리즘과 함께 pseudo random 생성 기법을 적용한 Randomized Hash Lock scheme이 제안되었다. 그러나 이 방법은 만약 Tag의 비밀 정보가 노출되어진다면 Tag에 대한 위치 정보가 노출되기 용이한 문제점이 지적되고 있다[3].

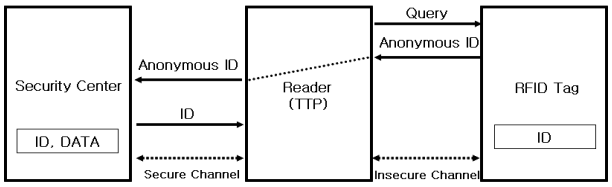
이러한 문제점을 해결하기 위해 본 논문에서는 종래의 인증 프로토콜을 개선한 RFID Tag을 위한 Robust 인증프로토콜을 제안하고, 제안한 인증 프로토콜의 검증에 관하여 기술한다. 제안한 프로토콜은 ISO/IEC 18000 standard를 기본으로 하고 있으며, 강인한 인증을 위해 표준 프로토콜 frame format 을 수정한다. 상호 인증을 위해 three-way challenge response 프로토콜을 사용하며, 인증 알고리듬은 SHA-1[7]이 추가되었다.

2. 관련 연구

2.1 표준 통신 프로토콜

RFID시스템은 크게 3가지, 즉 고유 정보를 저장하는 RFID Tag, Tag의 판독 및 해독 기능을 하는 Reader, 그리고 Reader와 연결되어 필요한 정보를 받아 가공 및 처리를 행하는 Back-end Server(Security Center)로 구성된다. 그림 1은 일반적인 RFID 시스템의 개념도를 나타낸다[4].

*본 연구는 중소기업청 2005년도 산학연 공동기술개발 컨소시엄사업 IDEC 지원으로 수행되었음.



(그림 1) RFID 시스템의 개념도

ISO/IEC 18000은 RFID의 Tag와 Reader의 표준 통신 인터페이스를 기술하고 있다. 표준 통신 인터페이스는 변조, 프레이밍, 충돌방지 메커니즘, 프로토콜 파라메터, 기타 정보의 제공 등으로 구성되어 있다. 표준 통신 인터페이스에 따르면, 변조 방식은 ASK 방식을 많이 사용하며, 데이터 전송은 SOF(Start-of-frame)와 EOF(End-of-frame)의 구분자 사이에 데이터를 첨부하여 전송한다.

데이터 전송은 "Reader talks first"를 기본 개념으로, Reader가 Tag에게 요청 데이터를 전송하고, Tag는 이 요청 데이터를 분석하여 응답 데이터를 재전송하는 형태이다. 데이터는 크게 4 부분으로 구성되는데 각 각은 다음과 같다[5].

- Flags : 데이터의 전송 타입, Tag의 접근 방식, data rate 등의 정보를 기술
- Command code : Tag의 동작을 나타냄. 크게 3가지의 그룹으로 구성되어지며, Mandatory Command는 반드시 구현되어져야 한다.
- Parameters and data fields : Command code의 데이터로 Command code의 동작에 필요한 데이터
- CRC : 통신상의 에러 체크

표준 통신 인터페이스는 이외에도 Custom Command에 대한 기술 방법 등에 대하여도 기술하고 있는데, 인증이나 보안을 위한 부분은 기술이 되어 있지 않다.

2.2 인증 메커니즘

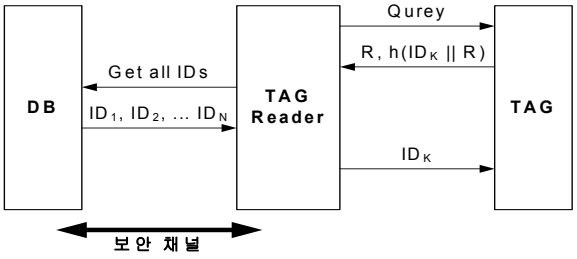
MIT에서 제안한 Hash Lock scheme[3]은 Hash 알고리듬을 기반으로 하며 MetaID 정보를 보관할 수 있는 저장 공간을 보유하고 있다. Tag 소유자는 임의의 키를 선택하고 키에 대한 해쉬 값을 계산하여 MetaID로 지정한다. 그리고 MetaID 정보를 Tag에 저장하고 Lock 상태로 설정한다.

Lock 상태의 Tag는 오직 외부의 요청에 대한 응답으로 MetaID 정보만을 제공하며 다른 정보는 제공하지 않는다. Tag는 Key 정보를 통한 Reader의 요구에 따라 UnLock 상태가 되며 Reader에 Tag의 정보를 제공하게 된다.

허락되지 않은 Reader는 DB로부터 Key 정보를 획득할 수 없으며, Key 정보가 없으면 Tag의 Unlock 이 불가능하다. 그러나 MetaID 정보는 각 Tag의 식별 정보로 사용되어짐으로 추적 공격에는 취약할 수밖에 없다.

Randomized Hash Lock 메커니즘[3]은 Hash Lock 메커니즘의 MetaID 정보를 추적 공격할수 있는 취약함을 극복하기 위한 목적으로 제안된 기술이다. Tag는 단방향 해쉬 알고리즘 및 PRF(Pseudo Random Function) 기능을 갖는다.

그림 3은 Randomized Hash Lock scheme을 나타낸다. Reader는 Tag에게 식별을 위한 Query를 송신하게 되며, 이 응답으로 Tag는 임의의 난수(R)을 생성하게 된다.



(그림 2) Randomized Hash Lock scheme

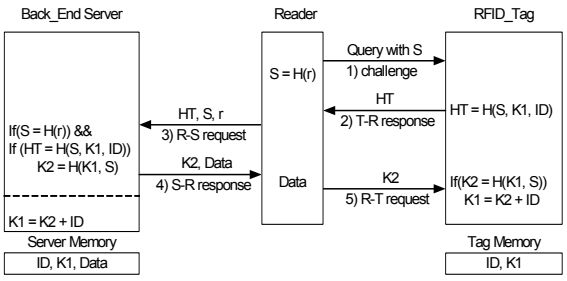
또한 생성된 난수(R)와 자신의 식별자인 ID_K 를 조합하여 생성된 해쉬 값을 함께 Reader에 전송한다. Reader는 저장된 모든 ID 정보를 기반으로 h(ID || R)을 생성한 후 Tag로부터 받은 해쉬 값과 비교하여 일치하는 ID를 찾게 된다. 일치하는 ID를 찾게 되면 태그에 ID 값을 전송하며, ID를 수신한 Tag는 자신의 ID와 동일하면 Unlock 상태가 된다. 그러나 이 기술은 대량의 Tag 환경에서는 Reader에서 처리해야 하는 부하의 한계로 인하여 부적합하고 소량의 Tag을 갖는 RFID환경에서 적합한 기술이다. 또한 Tag가 난수(R)를 생성하기 위한 회로를 추가적으로 가지고 있어야 한다는 부담이 있다[3, 6].

3. 개선된 Robust 인증 메커니즘

3.1 Robust 인증 프로토콜

본 논문은 Randomized Hash Lock 메커니즘을 기반으로 Randomized Hash Lock의 문제점인 Tag이 난수 생성부를 추가적으로 가져야 하는 단점을 개선하기 위하여, Reader가 난수를 생성시키도록 개선하였다. 또한 인가되지 않은 Reader나 공격자의 replay-attack을 방지하기 위하여 Server와 Tag이 가진 Key를 변경하는 기법을 적용한 인증 메커니즘을 제안한다.

그림 3은 제안한 개선된 인증 프로토콜의 흐름도를 나타낸다.



(그림 3) 제안한 인증 프로토콜의 흐름도

그림 3에서 Back_End Server는 데이터베이스를 가진 서버를, H()는 해쉬 알고리즘을 나타낸다. RNG는 Random Number Generator를 r은 Random Number를 나타내고, K1은 Tag와 Server만 보관하는 Key를 나타낸다. 그리고 K2는 Key1의 변경을 위한 Key를 ID는 Tag의 고유 ID를 나타낸다. 제안된 인증 프로토콜의 흐름은 아래와 같이 6단계로 나누어진다.

[Step 1] (challenge): Reader에서는 RNG를 통하여 r을 만들고, 해쉬 알고리즘 H()를 이용하여 S를 구한다. 구해진 S는 Tag으로 보내는 쿼리에 포함되어진다. 이때 Reader는 Inventory request Command(IRqC)를 사용하여 데이터를 전송한다.

[Step 2] (T->R response): Tag(T)은 메모리에 저장되어있는 해쉬함수를 통하여 해쉬된 값(HT)를 생성하여 Reader 에 재전송을 하게 된다. Tag은 Inventory response Command(IRsC)를 통하여 데이터를 전송한다.

[Step 3] (R->Server request): 수신된 HT, Reader가 생성한 r과 S를 Server로 전송한다. 이는 Reader를 통하여 발생할 수 있는 man-in-the-middle attack을 방지하기 위함이다. 이 단계를 통하여 Server 와 Reader, 그리고 Tag은 다음의 과정을 통하여 인증하게 된다.

1) Server와 Reader는 해쉬 알고리즘 H()를 사용하기 때문에 Reader에서 생성된 r을 Server에서 같은 해쉬 알고리즘 H()를 통하여 S와 일치하는지 검증한다.
2) Server와 Reader사이의 인증이 확인되면, Server는 Reader로부터 전송받은 HT를 통하여 Server의 Database에 존재하는 HT와 일치하는 ID, K1, Data를 추출한다.

[Step 4] (Server->R response): 이전 단계를 통하여 인증이 완료되면, Server는 Tag과 다음 통신에 사용할 새로운 키를 생성하는데 쓰일 K2를 생성하여 전송한다. 이때, Reader에 필요한 정보인 Data도 함께 Reader에 보내준다.

[Step 5] (R->T request): Reader는 Server 로부터 받은 K2와 Data에서 Reader가 필요로 하는 Data만을 취하고 K2를 Tag에 전송한다. 이때 Reader는 Select request 커맨드를 사용하여 데이터를 전송한다. Reader로부터 K2를 받은 Tag은 Server와 같은 해쉬 알고리즘을 통하여 K2를 계산하여 일치하는지 확인하고, 새로운 K1을 생성하여 기록한다. 이 K1은 다음 통신에서 쓰이게 되며 이를 통하여 replay attack을 방지할 수 있다.

[Step 6] (Server-R-T check): Tag으로부터 에러가 발생하지 않으면 통신은 종료되며, 인증이 완료된다. 이때, Server도 K1을 변경하여 다음 통신 시에 발생할 수 있는 replay attack을 방지한다.

3.2 Robust 인증 프로토콜의 설계

통신에 사용되는 Command code는 Inventory와 Select이며, 이 Command code의 통신 포맷을 변경하여 인증 메커니즘을 수행한다. 일반적으로 통신은 Reader에서 Tag를 향하여 Request 데이터를 보내고, Tag는 수신한 Request에 따라 그에 맞는 Response 데이터를 전송한다.

그림 4는 General request format이다.

SOF	Flags	Command code	Parameters	Data	CRC	EOF

(그림 4) General request format

Flags는 현재 보내는 데이터의 옵션들에 대한 내용을 기술하고 있다. Command code는 보내는 데이터의 종류를 나타낸다. Parameters는 보내는 데이터에 따라 주어지는 추가적인 옵션을 나타내고, Data는 Parameters에 따른 데이터를 나타낸다. CRC은 SOF와 EOF를 제외한 데이터에 대한 통신상의 오류가 발생하는 것을 체크하기 위하여 계산하여 기록한다.

General response format 은 Reader의 request에 따라 달리 구성되며, 에러의 유무에 따라 flags만 기록되기도 한다. CRC은 request format 와 마찬가지로 SOF와 EOF를 제외한 데이터에 대하여 CRC체크 섬을 계산하여 기록하게 된다.

제안하는 인증 메커니즘(그림 3)에 따라, Inventory request format은 Reader에서 생성된 S 데이터를 추가로 삽입, 구성하였다. S는 데이터 전송시 많은 부하를 가지지 않도록 16 bit를 가지도록 구성하였다.

Original Inventory response format의 구조는 UID 데이터 field를 가지고 있으나, 제안하는 프로토콜의 구조에서는 UID 데이터field 대신하여 HT 데이터 field로 대치된 형식을 가지게 된다. HT는 제안하는 인증 메커니즘에 의하여 생성되는 데이터로서 해쉬 함수를 거쳐 생성하게 되므로 HT를 통하여 UID를 알 수 없게 된다.

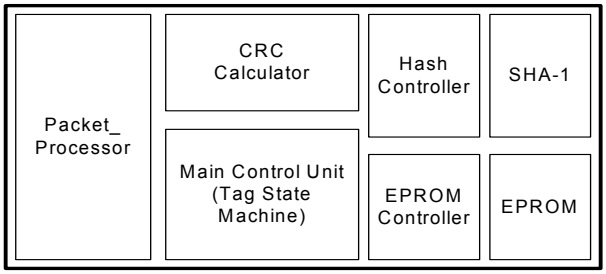
또한, 제안하는 인증 메커니즘에 따라 Select request format의 구조는 Back_End Server에서 생성한 K2 데이터를 추가하여 함께 전송한다.

4. RFID Tag의 디지털 part 설계

RFID-Tag의 구조는 RF/Analog 부분과 디지털 부분으로 크게 구분된다. Digital Part는 RF/Analog Part에서 넘어온 패킷의 Command를 분석하는 Command Decoder와 CRC 체크값을 계산하는 CRC Calculator, Tag의 고유 ID등을 저장하는 EPROM과 Controller, 이 블록들의 동작을 관리하고 Tag의 상태를 관리하는 Main Control Unit으로 구성된다.

그림 5는 제안한 인증 메커니즘을 적용한 딧털 part의 설계 블록도이며, 크게 Packet_Processor와 CRC Calculator, SHA-1, EPROM 블록 그리고 제

어 블록으로 구성된다. Packet_Processor 블록에서는 아날로그 부분에서 넘어온 데이터를 입력받고, 데이터를 분석하여 Tag의 동작에 필요한 데이터를 분리하는 일을 수행한다. 또한 각 제어 블록 으로부터 처리를 마치고 Reader에게 보내어져야하는 데이터들의 조립도 이곳에서 수행된다.



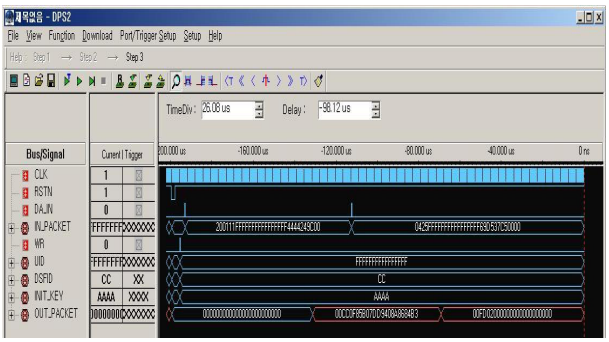
(그림 5) 디지털 Part의 설계 블록도

CRC Calculator 블록에서는 Packet_Processor로부터 넘어온 데이터를 이용하여 CRC check와 제안한 인증 메커니즘에 맞는 동작을 수행한다. 이때, 메모리(EPROM) 블록으로부터 ID와 K1 데이터를 받아 해쉬 알고리즘 SHA-1을 수행하는 SHA-1 블록으로 데이터를 전송하는 역할도 함께 한다. 모든 동작을 마치고 각각의 데이터들은 Packet_Processor로 전송되어 조립된다. EPROM 블록은 Tag의 고유 정보를 저장하는 블록이다. SHA-1 블록은 제안하는 인증 메커니즘의 해쉬 알고리즘을 수행하는 블록으로 해쉬 알고리즘 SHA-1을 사용하고 있다.

설계한 Hash engine의 동작 과정은 초기 Init를 기준으로 입력된 커맨드의 에러가 발생하지 않으면 CRC check를 통하고, 이때 커맨드의 종류에 따라 다른 동작들을 수행하도록 구성되어있다.

5. 구현 및 성능평가

설계한 인증 프로토콜의 검증을 위해 RFID Tag의 디지털 Part의 구현은 Xilinx ISE 6.2i 툴을 사용하였고, 타이밍 시뮬레이션에는 Modelsim 을 사용하였다. FPGA 검증을 위하여 리버트론사의 EDA-PRO 키트를 사용하였고, 타겟 디바이스는 Xilinx Virtex2 xc4000 을 사용하였다. 그림 6은 키트를 사용한 검증 결과를 나타낸다.



(그림 6) FPGA 키트의 검증 결과

검증에 사용한 테스트 벡터는 Mentor Modelsim을 통해 시뮬레이션한 벡터를 이용하였다. EDA-PRO FPGA 키트는 Xilinx Virtex Xcv4000 디바이스로 구성되었다. 검증에는 1MHz의 동작주파수를 적용하여 검증하였으며, 타이밍 시뮬레이션 시 사용한 테스트 벡터를 사용하였다. <표 1>은 구현된 디지털 Part에 대한 성능평가 결과를 나타낸다.

<표 1> Tag의 성능 평가

	# gates (Silces)	Frequency
Xilinx 합성	34,830 (1290)	75 Mhz
Synopsys 합성	13,000	45 Mhz

Xilinx ISE 로 합성한 결과는 슬라이스 약 1290개, 총 게이트 수는 약 34830 정도였고, 약 75Mhz의 동작 주파수로 나타났다. Synopsys Hynix 0.25 공정으로 합성한 결과는 게이트 수는 약 13000 정도였고, 약 45Mhz의 동작 주파수로 나타났다.

6. 결론

본 논문에서는 Tag의 보안상의 문제를 해결하기 위해 Robust 인증 메커니즘을 제안하였다. 제안된 프로토콜은 한 번사용된 키값을 변경하는 기법을 사용하여 인가되지 않은 Reader나 공격자의 replay-attack을 방지할 수 있다. 제안된 인증 메커니즘의 검증을 위해 디지털 Part를 FPGA로 구현하였다. 설계 검증을 위해 Xilinx ISE 6.2i 툴을 사용하였고, Xilinx Virtex Xcv4000 디바이스의 FPGA를 타겟으로 합성하였다.

참고문헌

[1] 이근호, “RFID 기술과 시스템”, 정보산업 민간백서, 2004.

[2] 주학수, “RFID 시스템의 보안 및 프라이버시 보호를 위한 기술 분석”, 전자정보센터, 2004.

[3] 서운석, 신순자, 구자동, 임진수, “유비쿼터스 컴퓨팅 환경에서 보안 및 인증서비스 방향연구”, 한국전산원, 2004.

[4] Jeongkyu Yang, Kui Ren, SuGil Choi, Kwangjo Kim, “Privacy Preserving Mutual Authentication Protocol for Low-cost RFID”.

[5] Martin Feldhofer, "A Proposal for an Authentication Protocol in a Security Layer for RFID Smart Tags", Graz University of Technology.

[6] J. G. Yang and K. J. Kim, “Security and Privacy on Authentication for Low-cost RFID,“ Proc. of SCIS 2005, Jan.25~28, 2005.

[7] NIST, Secure hash standard, FIPS 180-1, US Department of Commerce, Washington D.C., April 1995.