(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0098924
(43) 공개일자 2008년11월12일

(51) Int. Cl.
H04L 9/32 (2006.01) *G06K 17/00* (2006.01)
(21) 출원번호 10-2007-0044334
(22) 출원일자 2007년05월08일
심사청구일자 2007년05월08일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가1 고려대학교 내
(72) 발명자
최진영
서울 성북구 안암동5가 고려대학교 컴퓨터학과 정형기법연구실
김현석
서울 성북구 안암동5가 고려대학교 컴퓨터학과 정형기법연구실
(74) 대리인
현종철

전체 청구항 수 : 총 10 항

(54) RFID 태그에서의 보안 방법, RFID 태그의 보안장치 및 RFID 환경에서의 보안 방법

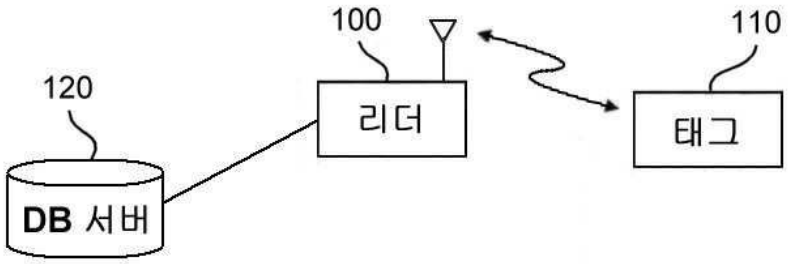
(57) 요 약

RFID 태그에서의 보안 방법, RFID 태그의 보안 장치 및 RFID 환경에서의 보안 방법이 개시된다.

본 발명에 따른 RFID 태그(tag)에서의 보안 방법은, RFID 태그(tag)에서의 보안 방법에 있어서, (a) RFID 리더(reader)로부터의 쿼리(query)를 수신하는 단계, (b) 상기 RFID 태그를 나타내는 ID와 RFID 태그에서 생성된 것임을 나타내는 태그 난수값을 상기 RFID 리더로 전송하여 소정의 데이터베이스 서버로 전달되게 하는 단계, (c) 상기 데이터베이스 서버로부터 전달되는 데이터베이스 서버에 대응하는 난수값 및 소정의 연산 결과를 수신하는 단계, 및 (d) 상기 (c) 단계에서 수신한 난수값과 연산 결과로부터 상기 데이터베이스 서버가 정상적인 서버임을 인증하고, 상기 RFID 리더에게 상기 쿼리에 대응하는 태그 정보를 전송하는 단계를 포함한다.

본 발명에 의하면, RFID 태그(tag)에서의 보안 방법에 있어서, RFID 리더(reader)로부터의 쿼리(query)를 수신하고, RFID 태그를 나타내는 ID와 RFID 태그에서 생성된 것임을 나타내는 태그 난수값을 RFID 리더로 전송하여 데이터베이스 서버로 전달되게 하며, 데이터베이스 서버로부터 전달되는 데이터베이스 서버에 대응하는 난수값 및 연산 결과를 수신하며, 수신한 난수값과 연산 결과로부터 데이터베이스 서버가 정상적인 서버임을 인증하고, RFID 리더에게 쿼리에 대응하는 태그 정보를 전송하게 하여, RFID 환경에서 신뢰성있는 보안 방법을 제공하여 사용자의 프라이버시 보호를 포함하는 편익을 대폭 제공한다.

대 표 도 - 도1



특허청구의 범위

청구항 1

RFID 태그(tag)에서의 보안 방법에 있어서,

(a) RFID 리더(reader)로부터의 쿼리(query)를 수신하는 단계;

(b) 상기 RFID 태그를 나타내는 ID와 RFID 태그에서 생성된 것임을 나타내는 태그 난수값을 상기 RFID 리더로 전송하여 소정의 데이터베이스 서버로 전달되게 하는 단계;

(c) 상기 데이터베이스 서버로부터 전달되는 데이터베이스 서버에 대응하는 난수값 및 소정의 연산 결과를 수신하는 단계; 및

(d) 상기 (c) 단계에서 수신한 난수값과 연산 결과로부터 상기 데이터베이스 서버가 정상적인 서버임을 인증하고, 상기 RFID 리더에게 상기 쿼리에 대응하는 태그 정보를 전송하는 단계;를 포함하는 것을 특징으로 하는 RFID 태그에서의 보안 방법.

청구항 2

제1항에 있어서,

상기 (a) 단계에서 수신하는 쿼리에는 상기 RFID 리더에 대응하는 키값이 포함되며,

상기 (b) 단계에서 상기 RFID 태그를 나타내는 ID는 상기 RFID 리더에 대응하는 키값과 상기 RFID 태그에서 생성된 난수를 입력으로 연산한 결과임을 특징으로 하는 RFID 태그에서의 보안 방법.

청구항 3

제2항에 있어서,

상기 RFID 태그를 나타내는 ID는 상기 RFID 리더로부터의 키값과 상기 RFID 태그에서 생성된 난수를 입력으로 배타적 합을 하고 그 결과를 해시 처리한 결과값인 것을 특징으로 하는 RFID 태그에서의 보안 방법.

청구항 4

제1항 또는 제3항에 있어서,

상기 (c) 단계에서의 소정의 연산 결과는 상기 데이터베이스 서버가 상기 RFID 태그로부터의 태그 ID와 태그 난수값을 비교하여 서로 대응되는 태그로부터의 것임이 확인되면 상기 데이터베이스 서버에 대응하는 난수값과 상기 RFID 리더에 대응하는 키 값을 배타적 합 연산하고, 그 결과를 해시 처리한 것임을 특징으로 하는 RFID 태그에서의 보안 방법.

청구항 5

제1항에 있어서,

상기 (d) 단계에서 상기 RFID 리더에게 전송하는 태그 정보는 해시 처리된 정보임을 특징으로 하는 RFID 태그에서의 보안 방법.

청구항 6

RFID 태그의 보안 장치에 있어서,

RFID 리더로부터의 데이터를 수신하는 수신부;

상기 태그에 대응하는 태그 난수값을 생성하는 난수발생부;

RFID 리더로부터의 쿼리(query)를 수신하면, 상기 태그를 나타내는 ID를 생성하는 ID생성부;

소정의 데이터베이스 서버로부터 데이터베이스 서버에 대응하는 난수값 및 소정의 연산 결과를 수신하여 상기 데이터베이스 서버가 정상적인 서버임을 인증하는 인증부; 및

상기 태그 난수값 및 태그를 나타내는 ID를 RFID 리더로 전송하며, 인증부에서 인증이 되면 상기 쿼리에 대응하

는 태그 정보를 RFID 리더로 전송하는 전송부;를 포함하는 것을 특징으로 하는 RFID 태그의 보안 장치.

청구항 7

제6항에 있어서,

상기 RFID 리더로부터 수신하는 쿼리에는 상기 RFID 리더에 대응하는 키값이 포함되며, 상기 RFID 태그를 나타내는 ID는 상기 RFID 리더에 대응하는 키값과 상기 RFID 태그 난수값을 입력으로 배타적 합을 하고 그 결과를 해시 처리한 결과값인 것을 특징으로 하는 RFID 태그의 보안 장치.

청구항 8

제6항에 있어서,

상기 데이터베이스 서버로부터의 소정의 연산 결과는 상기 RFID 태그로부터의 태그 ID와 태그 난수값을 비교하여 서로 대응되는 태그로부터의 것임이 확인되면 상기 데이터베이스 서버에 대응하는 난수값과 상기 RFID 리더에 대응하는 키 값을 배타적 합 연산하고, 그 결과를 해시 처리한 것임을 특징으로 하는 RFID 태그의 보안 장치.

청구항 9

제6항에 있어서,

상기 단계에서 상기 RFID 리더에게 전송하는 태그 정보는 해시 처리된 정보임을 특징으로 하는 RFID 태그의 보안 장치.

청구항 10

태그 정보를 요청하는 RFID 리더, 태그 정보를 제공하는 RFID 태그 및 RFID 태그에 대한 정보를 미리 저장하고 있는 데이터베이스 서버를 포함하는 RFID 환경에서의 보안 방법에 있어서,

(a) 상기 RFID 태그가 RFID 리더로부터의 쿼리(query)를 수신하는 단계;

(b) 상기 RFID 태그가 상기 RFID 리더로부터의 키값과 상기 RFID 태그에서 생성된 난수를 입력으로 배타적 합을 하고 그 결과를 해시 처리한 결과값인 상기 RFID 태그를 나타내는 ID와 상기 태그에서 생성된 난수값을 상기 RFID 리더로 전송하는 단계;

(c) 상기 RFID 리더는 상기 RFID 태그로부터 전송받은 데이터를 상기 데이터베이스 서버로 전달하는 단계;

(d) 상기 데이터베이스 서버는 상기 RFID 태그로부터의 태그 ID와 태그 난수값을 비교하여 서로 대응되는 태그로부터의 것임이 확인되면 상기 데이터베이스 서버에 대응하는 난수값과 상기 RFID 리더에 대응하는 키 값을 배타적 합 연산하고, 그 결과를 해시 처리한 값과 상기 데이터베이스 서버 난수값을 상기 RFID 리더로 전달하는 단계;

(e) 상기 RFID 리더가 상기 전달받은 데이터를 상기 RFID 태그로 포워딩하는 단계; 및

(f) 상기 RFID 태그가 상기 포워딩받은 상기 데이터베이스 서버 난수값과 연산 결과로부터 상기 데이터베이스 서버가 정상적인 서버임을 인증하고, 상기 RFID 리더에게 상기 쿼리에 대응하는 태그 정보를 해시 처리하여 전송하는 단계;를 포함하는 것을 특징으로 하는 RFID 환경에서의 보안 방법.

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<5> 본 발명은 네트워크 통신에 관한 것으로서, 특히 RFID 태그에서의 보안 방법, RFID 태그의 보안 장치 및 RFID 환경에서의 보안 방법에 관한 것이다.

<6> RFID를 위한 컴퓨팅 환경은 일반적인 인터넷 환경과는 달리 많은 제약사항을 갖는다. 이러한 제약사항은 셀룰러

폰(Cellular Phone) 등을 이용한 무선 인터넷보다 더욱 자원 측면적인 한계를 갖는다. 즉 유비쿼터스를 위한 RFID환경을 구축하기 위해서는 모든 상품이나 사람 등 객체에 설치되는 태그(tag)가격은 아주 저렴하게, 예를 들면 개당 5센트 이하로 구현되어져야 하며, 태그 리더(tag reader) 장비나 백엔드(Back End) 시스템에서 많은 성능, 자원 측면에서 열악한 태그장비의 자원적인 한계를 극복할 수 있도록 설계 운영되어져야 한다. 이에 보안 기술 적용에 대한 부분도 이러한 운영, 환경 측면을 충분히 고려해야 한다.

<7> 보안프로토콜을 구현하기 전에 설계단계에서부터 사용자와 개발자에게 안전성과 신뢰성을 제공하기 위한 기술이 요구되고 있다. 예를 들면 정당하지 않은 자가 RFID 리더를 이용해서 RFID 태그로부터 무단히 정보를 가져가는, 특히 프라이버시와 관계되는 정보를 불법적으로 가져가는 상황이 발생할 수 있다.그러나 RFID와 같은 상당히 제한된 환경에서의 신뢰하고 안심할만한 보안 방법 혹은 보안 프로토콜이 아직은 미흡하다는 문제가 있다.

발명이 이루고자 하는 기술적 과제

<8> 따라서, 본 발명이 이루고자 하는 첫 번째 기술적 과제는, 유비쿼티스 환경에서 사용되는 RFID 환경에서 프라이버시 침해문제 등과 같은 문제를 차단할 수 있도록 하는 RFID 태그에서의 보안 방법을 제공하는 것이다.

<9> 그리고, 본 발명이 이루고자 하는 두 번째 기술적 과제는, RFID 리더, 태그 정보를 제공하는 RFID 태그 및 RFID 태그에 대한 정보를 미리 저장하고 있는 데이터베이스를 포함하는 RFID 태그의 보안 장치를 제공하는 것이다.

<10> 또한, 본 발명이 이루고자 하는 세 번째 기술적 과제는 RFID 환경에서의 보안 방법을 제공하는 것이다.

발명의 구성 및 작용

<11> 상기 첫 번째 기술적 과제를 달성하기 위하여 본 발명은,

<12> RFID 태그(tag)에서의 보안 방법에 있어서, (a) RFID 리더(reader)로부터의 쿼리(query)를 수신하는 단계; (b) 상기 RFID 태그를 나타내는 ID와 RFID 태그에서 생성된 것임을 나타내는 태그 난수값을 상기 RFID 리더로 전송하여 소정의 데이터베이스 서버로 전달되게 하는 단계; (c) 상기 데이터베이스 서버로부터 전달되는 데이터베이스 서버에 대응하는 난수값 및 소정의 연산 결과를 수신하는 단계; 및 (d) 상기 (c) 단계에서 수신한 난수값과 연산 결과로부터 상기 데이터베이스 서버가 정상적인 서버임을 인증하고, 상기 RFID 리더에게 상기 쿼리에 대응하는 태그 정보를 전송하는 단계;를 포함하는 것을 특징으로 하는 RFID 태그에서의 보안 방법을 제공한다.

<13> 상기 두 번째 기술적 과제를 달성하기 위하여 본 발명은,

<14> RFID 태그의 보안 장치에 있어서, RFID 리더로부터의 데이터를 수신하는 수신부; 상기 태그에 대응하는 태그 난수값을 생성하는 난수발생부; RFID 리더로부터의 쿼리(query)를 수신하면, 상기 태그를 나타내는 ID를 생성하는 ID생성부; 소정의 데이터베이스 서버로부터 데이터베이스 서버에 대응하는 난수값 및 소정의 연산 결과를 수신하여 상기 데이터베이스 서버가 정상적인 서버임을 인증하는 인증부; 및 상기 태그 난수값 및 태그를 나타내는 ID를 RFID 리더로 전송하며, 인증부에서 인증이 되면 상기 쿼리에 대응하는 태그 정보를 RFID 리더로 전송하는 전송부;를 포함하는 것을 특징으로 하는 RFID 태그의 보안 장치를 제공한다.

<15> 상기 세 번째 기술적 과제를 달성하기 위하여 본 발명은,

<16> 태그 정보를 요청하는 RFID 리더, 태그 정보를 제공하는 RFID 태그 및 RFID 태그에 대한 정보를 미리 저장하고 있는 데이터베이스 서버를 포함하는 RFID 환경에서의 보안 방법에 있어서, (a) 상기 RFID 태그가 RFID 리더로부터의 쿼리(query)를 수신하는 단계; (b) 상기 RFID 태그가 상기 RFID 리더로부터의 키값과 상기 RFID 태그에서 생성된 난수를 입력으로 배타적 합을 하고 그 결과를 해시 처리한 결과값인 상기 RFID 태그를 나타내는 ID와 상기 태그에서 생성된 난수값을 상기 RFID 리더로 전송하는 단계; (c) 상기 RFID 리더는 상기 RFID 태그로부터 전송받은 데이터를 상기 데이터베이스 서버로 전달하는 단계; (d) 상기 데이터베이스 서버는 상기 RFID 태그로부터의 태그 ID와 태그 난수값을 비교하여 서로 대응되는 태그로부터의 것임이 확인되면 상기 데이터베이스 서버에 대응하는 난수값과 상기 RFID 리더에 대응하는 키 값을 배타적 합 연산하고, 그 결과를 해시 처리한 값과 상기 데이터베이스 서버 난수값을 상기 RFID 리더로 전달하는 단계; (e) 상기 RFID 리더가 상기 전달받은 데이터를 상기 RFID 태그로 포워딩하는 단계; 및 (f) 상기 RFID 태그가 상기 포워딩받은 상기 데이터베이스 서버 난수값과 연산 결과로부터 상기 데이터베이스 서버가 정상적인 서버임을 인증하고, 상기 RFID 리더에게 상기 쿼리에 대응하는 태그 정보를 해시 처리하여 전송하는 단계;를 포함하는 것을 특징으로 하는 RFID 환경에서의 보안 방법을 제공한다.

<17> 이하에서 첨부된 도면을 참조하여 본 발명의 바람직한 일 실시예를 상세히 설명한다.

<18> 도 1은 본 발명에 따른 보안이 적용되는 RFID 환경의 예를 도시한 것이다. 도 1의 RFID 환경은 태그 정보를 요청하는 RFID 리더(100), 태그 정보를 제공하는 RFID 태그(110) 및 RFID 태그(100)에 대한 정보를 미리 저장하고 있는 데이터베이스 서버(120)를 포함한다. 도 1에 대한 것은 다음의 도면들에 대한 설명에서 같이 진행될 것이다.

<19> 도 2는 본 발명에 따른 보안 방법이 RFID 환경에서 적용되는 흐름을 도시한 것이다. RFID 리더(100), RFID 태그(110) 및 데이터베이스 서버(120)는 도 1의 것과 동일한 것이다.

<20> 리더(100)는 태그(110)로부터 정보를 얻기 위해 쿼리를 보낸다(200 단계).

<21> 태그(110)는 리더(100)에게 태그를 나타내는 ID(이하 metaID라고 한다)과 태그 난수값을 넘겨준다(210 단계). 태그(110)는 리더(100)로부터의 키값과 태그에서 생성된 난수를 입력으로 배타적 합을 하고 그 결과를 해시 처리한 결과값인 태그(110)를 나타내는 태그 ID와 상기 태그에서 생성되었던 난수값을 리더(100)로 전송한다

<22> 리더로부터의 키 값은 리더마다 미리 정의할 수도 있고, 리더로부터의 쿼리에 포함된 값을 이용할 수도 있다.

<23> 리더(100) 태그(110)로부터 받은 값들을 백엔드(back-end) 서버인 데이터베이스 서버(120)에게 포워딩한다(220 단계).

<24> 데이터베이스 서버(230)는 포워딩 받은 값인 태그의 난수값과 metaID 의 값을 비교하여 서로 대응되는 태그로부터의 것임이 확인되면 상기 포워딩 된 데이터가 어느 태그로부터 왔는지를 확인한다. 그리고 데이터베이스 서버(230) 자신이 생성한 난수값과 리더에 대응하는 키 값을 배타적 합 처리하고, 이 결과 값과 함께 데이터베이스 난수값을 리더(100)로 보내게 된다(230 단계). 이때에 배타적 합 연산을 한 후에 해시 처리할 수 있다.

<25> 리더(100)는 데이터베이스 서버(120)로부터 이 값을 태그(110)에게 포워딩한다(240 단계).

<26> 태그(110)는 데이터베이스 서버(120)로부터 리더(100)를 통해 포워딩된 값인 난수값과 해쉬처리된 값의 비교를 정상적인 데이터베이스 서버로부터의 데이터인 것을 인증한 후, 리더에게 자신의 태그 정보를 해쉬 처리하여 넘겨준다(250 단계).

<27> 본 발명은 종래의 기술에 비해 특히 다음과 같은 특징이 있다.

<28> 상기와 같이 태그 및 데이터베이스 서버의 난수가 도입된다. 이는 데이터 프라이버시와 태그 응답시의 악의적인 리더로부터의 재생공격을 막기 위해 도입된 것이다. 이 경우 태그에서의 난수는 태그의 장소추적을 막기 위해서 적어도 하나는 필요하다.

<29> 그리고 리더, 태그, 데이터베이스 서버 등과 같은 통신 참여자들 간의 기밀성을 보장하기 위해 배타적 합(Exclusive-or) 연산이 도입되었으며, 리더와 태그, 데이터베이스 서버와 리더간의 안전한 채널을 구축하기 위해 데이터베이스의 난수값과 리더의 키값으로 이루어진 데이터베이스 서버 연산 결과라는 값이 사용되었다. 이 경우 데이터베이스의 난수값을 metaID와 같은 형식으로 사용할 수도 있다.

<30> 도 3은 본 발명에 따른 RFID 태그에서의 인증 장치의 구성을 블록으로 도시한 것이다.

<31> 이 장치는, RFID 리더로부터의 데이터를 수신하는 수신부(300), 태그에 대응하는 태그 난수값을 생성하는 난수발생부(310), RFID 리더로부터의 쿼리를 수신하면, 상기 태그를 나타내는 ID를 생성하는 ID생성부(320), 데이터베이스 서버로부터 데이터베이스 서버에 대응하는 난수값 및 소정의 연산 결과를 수신하여 상기 데이터베이스 서버가 정상적인 서버임을 인증하는 인증부(330), 태그 난수값 및 태그를 나타내는 ID를 RFID 리더로 전송하며, 인증부(330)에서 인증이 되면 상기 쿼리에 대응하는 태그 정보를 RFID 리더로 전송하는 전송부(340)를 포함한다.

<32> 도 3의 각 구성부에 대한 설명은 도 2에 대한 설명에 다 포함되어 있으므로 더 이상의 설명은 생략한다.

<33> 이하는 본 발명에 대한 효과를 설명하기 위한 것이다.

<34> 설계단계에서부터 사용자와 개발자에게 안전성과 신뢰성을 제공하기 위한 보안프로토콜 기술이 요구되고 있는데, 그러한 요구를 만족시키기 위해 진행되는 노력 중 대표적으로 정형 기법이라는 연구가 있으며 이는 정형 명세와 정형 검증의 두 가지 방법으로 구분된다.

<35> 정형 명세는 개발하고자 하는 시스템의 동작 및 시스템이 만족해야 하는 특성을 정형적인 표현방법을 이용해 모

델링하는 방법이고, 정형 검증은 정형적으로 명세된 시스템을 대상으로 그 시스템이 정확한지 혹은 그 시스템의 요구사항으로 주어지는 특성을 만족하는지를 논리적으로 증명하는 방법이다.

<36> 그 중 정형 검증은 정리증명과 모델체킹 기법으로 구분되며, 전자는 보안 로직을 이용하여 특정한 논리식으로 시스템을 명세하고 정확한 논리 증명단계로써 정확성을 증명하는 방식이고, 후자는 프로토콜의 인증과정을 유한 상태기계의 형식으로 모델링하고 그 모델이 만족해야 하는 요구사항이나 특성을 모델에서 만족되는지를 검증도구를 이용해 자동으로 증명하는 방식으로 ESTEREL, Murphi, FDR과 같은 방법이 있다. FDR에 대해서는 Formal Systems Ltd. FDR2 User Manual, Aug. 1999.과 같은 자료를 참조할 수 있다.

<37> 이하에서는 정형검증 도구 중 FDR이라는 모델체킹 도구를 이용, RFID 보안프로토콜인 해쉬기반 프로토콜들의 취약성을 분석하여 보안 프로토콜의 안전성을 향상시키고자 한다. RFID 보안프로토콜인 해쉬기반 프로토콜들에 대해서는 S. Weis, S. Sarma, R. Rivest and D. Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems , In 1st Intern. Conference on Security in Pervasive Computing (SPC), 2003 과 같은 자료를 참조할 수 있다.

<38> **Casper와 FDR 도구**

<39> **CSP(Communicating Sequential Process)**

<40> CSP는 프로세스 알제브라 언어로서, 병렬성을 갖는 통신프로토콜의 동작을 효율적으로 명세하기 위한 언어이다. CSP 혹은 CSP 언어에 대해서는 C.A.R. Hoare, Communicating Sequential Processes, Prentice-Hall, 1985 와 같은 자료를 참조할 수 있다. CSP는 최초 일반 통신 프로토콜 및 제어 시스템의 명세를 위해 사용되어졌으나, 점차 보안 프로토콜의 명세를 위한 영역으로 확대되어 가고 있다. CSP에서 제공하는 pure synchronization(|||)과 Interleaving parallelism(||) 개념을 사용하여 분산 시스템 환경하에서 동작하는 클라이언트 서버와 공격자 모델을 정형적으로 표현할 수 있는 장점을 갖고 있다. 예를 들어, 분산시스템 환경 하에서 동작하는 보안 시스템은 다음과 같이 간략히 표현할 수 있다.

<41> SYSTEM = CLIENT1 ||| CLIENT2||| SERVER || INTRUDER

<42> **Casper(A Compiler for the Analysis of Security Protocols)**

<43> CSP(Communication Sequential Process)언어를 이용하여 보안프로토콜 행위를 명세하고 FDR 정형검증 도구를 이용하여 보안속성을 검증하는 연구가 진행되었다. 하지만, CSP 언어를 이용한 정형명세과정은 정형적 설계 방법에 익숙치 않은 보안프로토콜 설계자에게는 매우 복잡한 명세언어라는 단점을 갖고 있었다. 이에 따라, 보안프로토콜의 행위를 간략히 명세할 수 있도록 Casper 도구가 개발되었다. Casper 도구로 보안프로토콜의 행위와 검증속성을 명세하게 되며, 자동변환기능을 이용해 CSP 명세코드를 생성할 수 있다. 결국, 자동 생성된 CSP 명세코드를 FDR 정형검증도구에 입력하여 보안프로토콜을 검증하게 된다. Casper에 대해서는 G. Lowe, Casper: A compiler for the analysis of security protocols , In Proceeding of the 1997 IEEE Computer Security Foundations Workshop X, IEEE Computer Society, Silver Spring, MD, pp. 18-30, 1997 과 같은 자료를 참조할 수 있다.

<44> **FDR(Failure Divergence Refinement)**

<45> FDR 도구는 CSP 명세언어를 입력으로 받아들이는 모델체킹 도구로서 옥스포드 대학에서 개발되었다. 이 도구는 CSP 명세언어로 기술된 보안프로토콜 모델이 보안성 및 인증속성과 같은 보안속성들을 만족하는지 검증하게 되며, 만일 만족하지 않을 경우에는 CSP 이벤트로 기술된 반례(counterexample)을 보여주어 보안상 취약점 분석을 도와준다.

<46> FDR 도구는 3가지의 검증방법을 지원하고 있다.

<47> Trace refinement : 안전성 (safety) 검증

<48> Failures refinement : 교착상태 (deadlock) 검증

<49> Failures - Divergence : 라이브락 (livelock) 검증

<50> **Casper/FDR을 이용한 본 발명에 따른 보안 방법**

<51> 제안하고자 하는 해쉬-배타적 합 기반 프로토콜에서는 3가지 새로운 기술이 적용된다.

<52> 이하에서는 본 발명에 따른 보안 방법을 Casper 도구를 이용해 모델링하였는데, 7가지 항목을 표현하였고, 그

중 자유변수 영역(#Free variables)과 프로토콜 기술영역(#Protocol description), 침입자 영역(#Intruder Information)에 대한 표현을 설명한다.

<53> #Free variables

<54> R, T : Agent

<55> key: Nonce

<56> tn, dbn : Nonce

<57> Id : Text

<58> DB: Server

<59> InverseKeys = (key, key), (tn, tn),

<60> (dbn, dbn)

<61> H : HashFunction

<62> #Processes

<63> INITIATOR(T, R, DB, key, tn, Id)

<64> RESPONDER(R, DB, T)

<65> SERVER(DB, T, key, dbn)

<66> #Protocol description

<67> 0. -> T : R

<68> 1 . T -> R : tn, H(key (+) tn)%metaID

<69> 2. R -> DB : tn, metaID%H(key (+) tn)

<70> 3. DB -> R : dbn, H(key (+) dbn)%auth

<71> 4. R -> T : dbn, auth%H(key (+) dbn)

<72> 5. T -> R : Id, H(Id)

<73> #Actual variables

<74> Tag, Reader, Mallory : Agent

<75> Key, MKey : Nonce

<76> TN, DBN, MN : Nonce

<77> ID : Text

<78> DataBase : Server

<79> InverseKeys = (Key, Key), (MKey, MKey),

<80> (TN, TN),

<81> (DBN, DBN), (MN, MN)

<82> #System

<83> INITIATOR(Tag, Reader, DataBase, Key, TN, ID)

<84> RESPONDER(Reader, DataBase, Tag)

<85> SERVER(DataBase, Tag, Key, DBN)

<86> #Specification

<87> StrongSecret(DB, key, [T])

<88> StrongSecret(T, key, [DB])

<89> StrongSecret(T, key, [R])

<90> Agreement(T, DB, [tn, key])

<91> #Intruder Information

<92> Intruder = Mallory

<93> IntruderKnowledge = {Tag, Reader, MKey, MN}

<94> 먼저 자유변수 영역(#Free variables)에서, R은 리더, T는 태그로서 각각 Agent로 나타내고, DB는 백엔드 서버의 역할을 한다. key는 Session 키로 난수형으로 표현, 마찬가지로 tn, dbn 또한 난수형으로 표현하며, Id는 Tag의 정보를 표현하고, InverseKeys는 Session 키 및 난수값에 대한 암호화 복호화를 표현하며, H는 해쉬함수를 뜻한다.

<95> 다음으로 프로토콜 기술 영역(#Protocol description)은 해쉬-배타적 합 기반 프로토콜을 명세한 부분으로 여기서 % 표현은 메세지1에서 T가 H(key(+)tn) 값을 metaID로서 수신자인 R에게 복호화의 목적이 아닌 단지 다른 수신자 DB에게 전달하는 목적을 지니고 있다. 따라서 메세지2에서 이 메세지가DB에게 전달되어 복호화된다. 메시지 3과 4, 5의 의미는 동일하게 표현될 수 있다. 마지막으로 침입자 영역(#Intruder Information)에 대한 정보가 제시되어 있다.

<96> **본 발명에 따른 보안 방법 검증 결과**

<97> 기존의 해쉬-언락킹 프로토콜에서는 metaID의 값을 중간자 공격 및 재생 공격에 이용함에 따라 태그 정보의 노출 및 추적이 가능하게 하였을 뿐만 아니라 리더기와 태그의 인증에 실패하는 결과를 초래하였다. 이를 수정하여 Casper script를 이용, 제안된 프로토콜을 검증하기 위해 명세하는 단계에서 해쉬-배타적 합 기반 프로토콜의 두 개체간 사용된 정보에 대한 비밀성과 개체간 상호ID에 대한 인증을 만족해야 하며 이는 다음과 같이 표현할 수 있다.

<98> StrongSecret(T, key, [DB])

<99> StrongSecret(T, key, [R])

<100> StrongSecret(DB, key, [T])

<101> Agreement(T, DB, [tn, key])

<102> 첫번째 표현은 'T는 key 정보를 오직 DB와만 알고 있다'라고 풀이할 수 있고 두번째 표현은 'T는 key 정보를 오직 R과만 알고 있다'로 풀이할 수 있다. 세번째 표현은 동일하며 네번째 표현은 'T는 tn, key 정보를 통해 DB로부터 자신의 개체를 인증받는다'라고 풀이할 수 있다

<103> 모델 체커를 이용해 비밀성과 개체인증 속성의 만족여부를 확인한 결과 첫번째 표현에서 T가 전달하는 key에 대해 DB와의 비밀성 속성을 만족하였고 이에 따라 결국 두 개체간의 데이터는 안전하게 유지되었다. 또한 T는 나머지 R에 대해서도 key 데이터를 안전하게 유지할 수 있었고 DB 입장에서 또한 T와 함께 통신함에 있어 key 정보가 누설되지 않았음을 알 수 있었다. tn과 key를 통한 개체 인증에서도 이 두 정보를 이용해 두 개체간의 인증에 성공했다.

<104> 위 비밀성 요구사항들에 대해 FDR을 통해 검증한 결과는 도 4와 같다.

<105> 상기에도 언급된 S. Weis, S. Sarma, R. Rivest and D. Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, In 1st Intern. Conference on Security in Pervasive Computing (SPC), 2003과 같은 자료에도 언급된 기존의 해쉬 언락킹 프로토콜에서는 T가 R에게 정상적인 데이터 전송을 했다고 간주했으나 I_Reader에 의해 H(key)정보가 노출되었다. 결과적으로 문제점은 T의 metaID 정보는 중간자 공격에 이용되었으며, 또 다른 공격시나리오로서 R입장에서 T로부터 정상적인 데이터를 전송받았다고 간주했으나, I_Tag나 I_DataBase와 같은 악의적인 개입이 가능했다.

<106> 이러한 해쉬-언락킹 프로토콜의 문제점으로 분석되었던 부분은 태그 내에 저장된 정보를 해쉬 기반기법으로 태깅할 수 있게 함으로써 발생되었으며, 본 발명에서는 태그의 식별자를 리더기에 통해 데이터베이스에 인증함으

로써 태그정보를 안전하게 리더에게 전달할 수 있었다는 것을 알 수 있다.

<107> 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 본 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 상기의 설명에 포함된 예들은 본 발명에 대한 이해를 위해 도입된 것이며, 이 예들은 본 발명의 사상과 범위를 한정하지 않는다. 상기의 예들 외에도 본 발명에 따른 다양한 실시 태양이 가능하다는 것은, 본 발명이 속한 기술 분야에 통상의 지식을 가진 사람에게는 자명할 것이다. 본 발명의 범위는 전술한 설명이 아니라 청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

<108> 또한 본 발명에 따른 상기의 각 단계는 일반적인 프로그래밍 기법을 이용하여 소프트웨어적으로 또는 하드웨어적으로 다양하게 구현할 수 있다는 것은 이 분야에 통상의 기술을 가진 자라면 용이하게 알 수 있는 것이다.

<109> 그리고 본 발명의 일부 단계들은, 또한, 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, CD-RW, 자기 테이프, 플로피디스크, HDD, 광 디스크, 광자기 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드로 저장되고 실행될 수 있다.

발명의 효과

<110> 본 발명에 의하면, RFID 태그에서의 보안 방법, RFID 태그의 보안 장치 및 RFID 환경에서의 보안 방법에 있어서, RFID 리더(reader)로부터의 쿼리(query)를 수신하고, RFID 태그를 나타내는 ID와 RFID 태그에서 생성된 것임을 나타내는 태그 난수값을 RFID 리더로 전송하여 데이터베이스 서버로 전달되게 하며, 데이터베이스 서버로부터 전달되는 데이터베이스 서버에 대응하는 난수값 및 연산 결과를 수신하며, 수신한 난수값과 연산 결과로부터 데이터베이스 서버가 정상적인 서버임을 인증하고, RFID 리더에게 쿼리에 대응하는 태그 정보를 전송하게 하여, RFID 환경에서 신뢰성있는 보안 방법을 제공하여 사용자의 프라이버시 보호를 포함하는 편익을 대폭 제공한다.

도면의 간단한 설명

<1> 도 1은 본 발명에 따른 보안이 적용되는 RFID 환경의 예를 도시한 것이다.

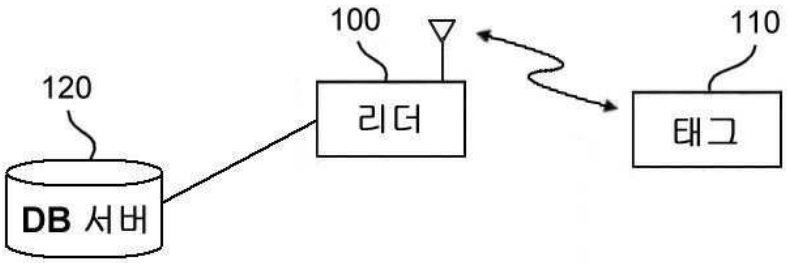
<2> 도 2는 본 발명에 따른 보안 방법이 RFID 환경에서 적용되는 흐름을 도시한 것이다.

<3> 도 3은 본 발명에 따른 RFID 태그에서의 인증 장치의 구성을 블록으로 도시한 것이다.

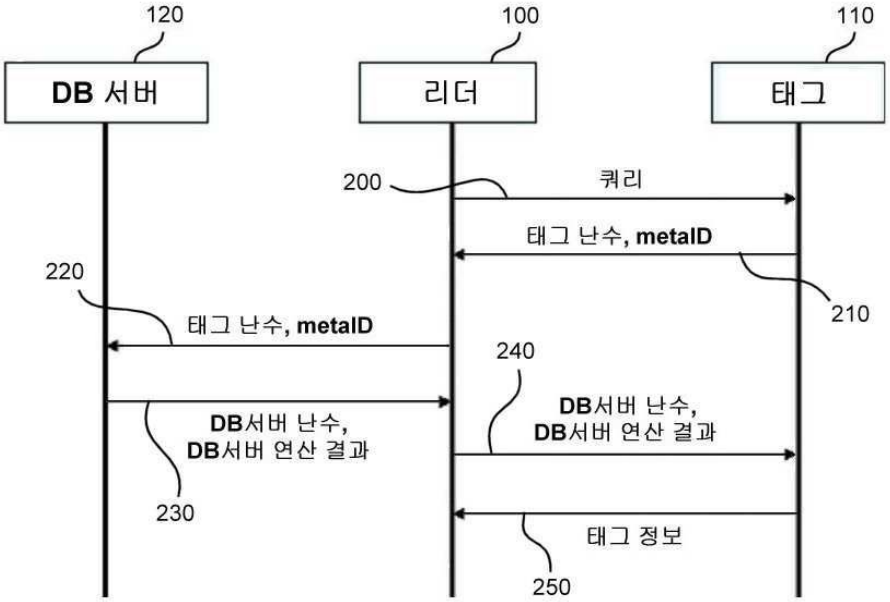
<4> 도 4는 본 발명에 따른 보안 방법의 검증결과를 도시한 것이다.

도면

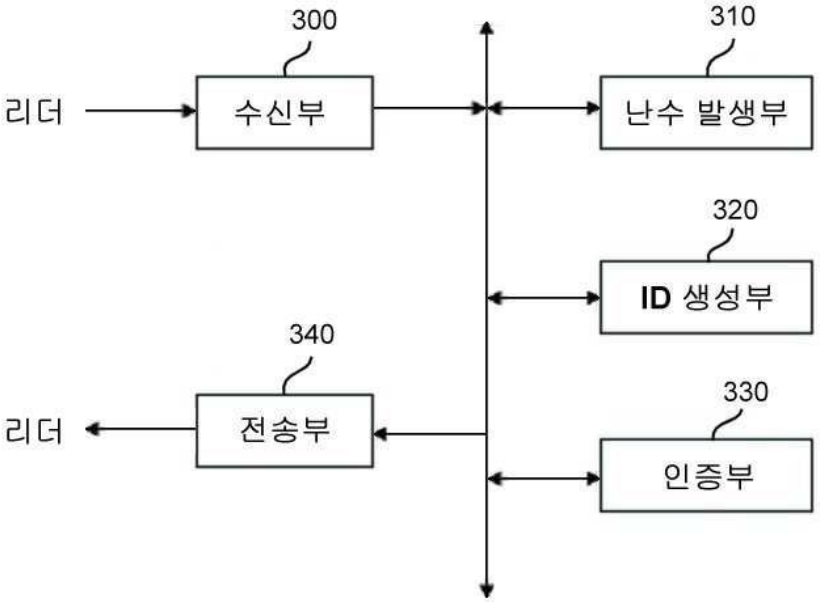
도면1



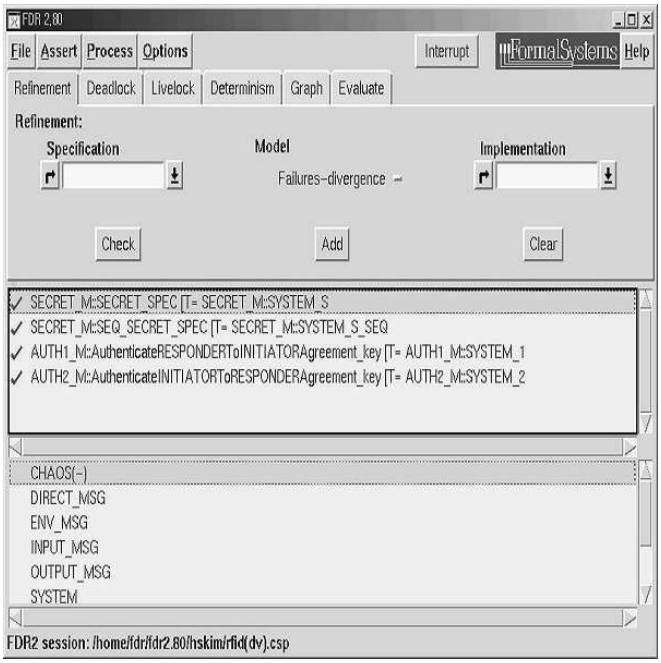
도면2


120
100
110
DB 서버
리더
태그
200
쿼리
태그 난수, metaID
210
220
태그 난수, metaID
240
DB서버 난수,
DB서버 연산 결과
DB서버 난수,
DB서버 연산 결과
230
태그 정보
250

도면3


300
310
리더
수신부
난수 발생부
320
ID 생성부
340
330
리더
전송부
인증부

도면4


FDR 2.80
File
Assert
Process
Options
Interrupt
FormalSystems
Help
Refinement
Deadlock
Livelock
Determinism
Graph
Evaluate
Refinement:
Specification
Model
Implementation
Failures-divergence
Check
Add
Clear
SECRET_M::SECRET_SPEC [T= SECRET_M::SYSTEM_S
SECRET_M::SEQ_SECRET_SPEC [T= SECRET_M::SYSTEM_S_SEQ
AUTH1_M::AuthenticateRESPONDERToINITIATORAgreement_key [T= AUTH1_M::SYSTEM_1
AUTH2_M::AuthenticateINITIATORToRESPONDERAgreement_key [T= AUTH2_M::SYSTEM_2
CHAOS(-)
DIRECT_MSG
ENV_MSG
INPUT_MSG
OUTPUT_MSG
SYSTEM
FDR2 session: /home/fdr/fdr2.80/hskim/rfid(dv).csp