RFID Privacy Protection Scheme for Secure Ubiquitous Computing

Hyun-Seok Kim, Jung-Hyun Oh and Jin-Young Choi

Dept. of Computer Science and Engineering, Korea University
Seoul, Korea

Abstract. A Radio-Frequency-Identification (RFID) tag is a small and inexpensive device that consists of an IC chip and an antenna which communicate by radio frequency. It emits an ID in response to a query from a radio communication device called as a reader. For this reason, the RFID tag is used for management of goods and it is used as a substitute for a bar code. However, RFID system may infringe on a consumer's privacy because it has a strong tracing ability. In this paper we describe problems of previous works on RFID security protocols and specify several known attacks and introduce PPP(Privacy Protection Protocol) for a RFID security protocol which serves as a proof of concept for authentication an RFID tag to a reader device using the vernam and standard encryption as a cryptographic primitive. To verify our protocol, we use model checking methodology, that is, Casper(A Compiler for Security Protocol), CSP(Communicating Sequential Processes) and then verify security properties such as secrecy and authentication using FDR(Failure Divergence Refinement) tool.

1 Introduction

Recently, the mass deployment of Radio Frequency Identification systems (RFID)[1][2] has taken place. These systems comprise of Radio Frequency (RF) tags or transponders, and RF readers or transceivers. Tag readers broadcast an RF signal to access resistant data stored in tags. One of the main differences with barcodes is that RFID tags provide an unique identifier, or a pseudonym that allows accessing to this unique identifier. The use of RFID tags offers several advantages over barcodes: data can be read automatically, without line of sight, and through a non-conducting material such as cardboard or paper, at a rate of hundreds of times per second, and from a distance of several meters. Despite all the advantages RFID technology offers there are serious concerns about security and privacy as well. To minimize the above concerns, security protocols play an essential role. As with any protocol, the security protocol comprises a prescribed sequence of interactions between entities, and is designed to achieve a certain end. A diplomatic protocol typically involves a memorandum of understanding exchange, intended to establish agreement between parties with potentially conflicting interests. Security protocols are, in fact, excellent candidates for rigorous analysis techniques: they are critical components of distributed security architecture, very easy to express, however, extremely difficult to evaluate by hand. They are deceptively simple: literature is full of protocols that appear to be secure but have subsequently been found to fall prey

Kim H., Oh J. and Choi J. (2007).
RFID Privacy Protection Scheme for Secure Ubiquitous Computing.
In *Proceedings of the 1st International Workshop on RFID Technology - Concepts, Applications, Challenges*, pages 117-124
DOI: 10.5220/0002411201170124

to a subtle attack, sometimes years later. Cryptographic primitives are used as building blocks to achieve security goals such as confidentiality and integrity authentication.
Formal methods play a very critical role in examining whether a security protocol is ambiguous, incorrect, inconsistent or incomplete. Hence, the importance of applying formal methods, particularly for safety critical systems, cannot be overemphasized. There are two main approaches in formal methods, logic based methodology [3], and tool based methodology [5][6][7]. In this paper, we specify the hash[1] based RFID authentication protocols as the previous works which employs hash functions to secure the RFID communication using Casper[6], CSP[5]. Then we verify whether or not it satisfies security properties such as secrecy and authentication using FDR model checking tool[7]. After running FDR tool, we reconfirm the existence of known security flaws in this protocol and propose the scheme of PPP(Privacy Protection Protocol) based on vernam and standard encryption for secure RFID communication. The contribution of this paper is designing and verifying the secure authentication protocol, which is widely researched in RFID systems using formal methods. This paper is organized as follows. In brief, Section 2 describes related work on RFID security and authentication schemes. In Section 3, the use of model checking is outlined for analyzing security protocols. Our analyzed result of the protocol will be described in Section 4. The proposed security scheme associated with encryption are presented in Section 5. Finally, the conclusion and our future work are addressed in the last section.

2 Related Work

Several researchers have attempted to resolve the security concerns related to the use of RFID tags and have proposed protocols that claim either to achieve secure authentication or to prevent unauthorized traceability. Most of these solutions only apply for weak adversary model (see e.g., [1][4]). In particular, those protocols for which a backend server is a trusted third party and the channel between the reader and the server is insecure, are susceptible to man-in-the-middle attacks. Weis-Sarma-Rivest-Engels [1] propose an RFID system as follows; A reader defines a “Lock” value by computing lock = hash(key)[1] where the key is a random value. This lock value is sent to a tag and the tag will store this value into its reserved memory location (i.e. a metaID value), and automatically the tag enters into the locked state. To unlock the tag, the reader needs to send the original key value to the tag, and the tag will perform a hash function on that key to obtain the metaID value. The tag then has to compare the metaID with its current metaID value. If both of them are matched, the tag unlocks itself. Once the tag is in unlocked state, it can respond its identification number such as the Electronic Product Code (EPC)[2] to readers’ queries in the forthcoming cycles.

3 Formal Methods for Security Protocol

3.1 Casper and FDR

Over the last few years, a method for analyzing security protocol that first models communication security protocol using CSP[5], then verifies its secrecy, authentication and other properties using FDR(Failure-Divergence Refinement)[7]. In this method, the main difficulty is specifying the security protocol's behavior using CSP. Creating the description of the security model with CSP is a very error-prone and difficult task. To simplify the expression of the security protocol, and render this process more error free, Casper(A Compiler of Security Protocol Analyzer)[6] was developed by Gavin Lowe[8]. This tool enables a non-expert who is unfamiliar with CSP to express the security protocol's behavior more easily, without being familiar with the notation used by CSP notation, using various key types, messages, security properties and intruder knowledge descriptions contained in Casper. In brief, Casper is a compiler that translates a more simple and concise description of a security communication model into CSP code.

3.2 CSP

CSP(Communicating Sequential Processes)[5] is a language for process specification specially designed to describe communication processes, and it can describe both a pure parallelism and interleaving semantics. In CSP, the former(a pure parallelism) is expressed as " || "and the latter(interleaving semantics) as " |||". The combination of a client, server and intruder are regarded as a process. The use of two different concurrency concepts is well suited to the description and analysis of network protocols. For example, security communication systems operated in distributed networks can be modeled briefly as follows.

```
SYSTEM =(CLIENT1 ||| CLIENT2 ||| SERVER) || INTRUDER
```

4 The Modeling and Analysis of the RFID Authentication Protocol using Casper and FDR Tool

4.1 The Specification of Hash Unlocking Protocol

Firstly, we model the behavior of hash unlocking protocol at the hash lock scheme and attacker in Casper script. The general overview of above protocol(Fig.1) was already described in section 2[1].

Table 1. The Hash Lock Scheme Notation.

T	RF tag's identity
R	RF reader's identity
DB	Back-end server's identity that has a database
Xkey	Session Key generated randomly from X
metaID	Key generated from reader using hash functioon
ID	Information value of tag
Xn	A random nonce generated by X
H	Hash function

Message 1.	R	$- >$ T	: Query
Message 2.	T	$- >$ R	: metaID
Message 3.	R	$- >$ DB	: metaID
Message 4.	DB	$- >$ R	: RKey, ID
Message 5.	R	$- >$ T	: RKey
Message 6.	T	$- >$ R	: ID

Fig. 1. The hash unclocking protocol.

```
#Protocol description
0.      -> T  : R
1.  T   -> R  : (H(Rkey)) % metaID
2.  R   -> DB : metaID % (H(Rkey))
3.  DB -> R   : Rkey, Id
4.  R   -> T  : Rkey
5.  T   -> R  : Id
```

Before explaination of *# Protocol description*, we will describe % notation to show specific notation. The % notation is used so that the metaID can be forwarded to other participants. This is why a reader can not construct the metaID, since the other reader does not know the value of hash function where m is a message and v is a variable, denoting that the recipient of the message should not attempt to decrypt the message m, but should instead store it in the variable v. Similarly, *v % m* is written to indicate that the sender should send the message stored in the variable v, and the recipient should expect a message of the form given by m. Therefore, *metaID* is the certain not knowing result value of hash function for T. In *# Protocol description* header, to unlock the tag, at the first line, Message 0 means that T(Tag) must communicate with R(Reader). The reader needs to send query to the tag and the tag sends the metaID to authenticate with reader.(Message 1). The reader forwards this metaID to DataBase to be ensured his identity.(Message 2). The DataBase has to compare the metaID with its current metaID value and ,if both of them are matched, lets the reader know the key and Id of tag.(Message 3). The reader authenticates his identity with the tag sending key received by database. (Message 4). As a result, if both of them are matched, the tag unlocks itself. Once the tag is in unlocked state, it can respond its identification number(*Id*) to queries of readers in the forthcoming cycles.(Message 5).

```
#Specification
Secret(R, Rkey, [T])
Secret(R, Id, [T])
Agreement(T, R, [Id, Rkey])
```

In hash unlocking protocol Casper script, *#Specification* description represents secrecy and authentication properties. The line starting with *Secret* expresses *secrecy property* associated with data privacy in RFID system. For example, the first statement is interpreted as " R believes that Rkey is a secret which should be known only to R and T" and the second statement is “ R believes that Id is a secret which should be known only to R and T”. If R, T or DB is an intruder in this protocol, secret information will be leaked to him, in which case a man-in-the-middle attack is considered to have occurred. The line starting with *Agreement* define that *authentication property* associated with authentication between a tag and a reader. For example, the third line means that “ T is authenticated to R with Id, Rkey”

```
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag, Reader, DataBase}
```

The above shows the intruder definition (*#Intruder Information*).

4.2 Protocol Goals

Using CSP[5], we describe the properties, i.e. secrecy property associated with data privacy, authentication property associated with authentication between a tag and a reader. The following predicate is implemented in CSP language.

```
SECRET_SPEC_0(s_) =
 signal.Claim_Secret?T_!s_?Rs_ -> (if member(Mallory,Rs_)
 then SECRET_SPEC_0(s_)
 else SECRET_SPEC_1(s_)) []leak.s_-> SECRET_SPEC_0(s_)
```

The *SECRET _ SPECT _ 0* and *SECRET _ SPECT _ 1* represent secret property of above *#Specification* section meet in the system. Formally speaking, if T has completed a protocol run apparently with R(*signal.Claim _ Secret?T _ !s _ ?Rs _*), and R is honest and uncompromised, then the key accepted during that run by T is not known to anyone other than R(*SECRET _ SPECT _ 1*), otherwise the key is known by someone in the system(*leak.s _*). Similarly, if R has completed a run with the honest and uncompromised T, then the key accepted by R is not known to anyone other than T.

```
AuthenticateINITIATORToRESPONDER
 Agreement_0(T) =
signal.Running1.INITIATOR_T.R
-> signal.Commit1. RESPONDER_R.T -> STOP
```

Formally speaking, the events of the form *Running1.INITIATOR _ T.R* in T’s run of the protocol are introduced to mark the point that should have been reached by the time that R performs the *Commit1.RESPONDER _ T.R event*. Occurrence of *Running1.INITIATOR _ T.R* run means simply that Agent T is following a protocol run apparently with R.

4.3 The Result of Verification

In this paper, we show verification results of the safety specification in hash unlocking scheme, we use traces refinement provided in FDR tool. Through debugging the counter-example trace events, we reconfirm that hash unlocking protocol may be susceptible to a sniff and spoof attack by an intruder due to unsecured communication channel between reader and tag. A general attack scenario, which could be found in this protocol is described as below; *I _ Agent* means an intruder who can sniff messages and spoof his identity.

```
1.    Tag     ->  I_Reader  : H(RKey)
2. I_Mallory  ->  DataBase  : H(RKey)
3.  DataBase  -> I_Mallory  : RKey, Id
```

The notation *I _ x* represents the intruder I imitating some participant to fake or intercept a message. Through the man-in-the-middle attack of the hash unlocking protocol, an intruder masquerading as Reader in Message 1, 2 could forward the message *H(Rkey)*and in Message3, an intruder masquerading as Reader could intercept the *RKey, ID*.

5 The Design and Verification of Privacy Protection Protocol for RFID System

5.1 The PPP (Privacy Protection Protocol)

In the previous scheme [9], they assumed that R(reader) is a TTP(Trusted Third Party) and the communication channel between R(reader) and DB(database) is secure. However, we assume that R is not a TTP and the communication channel is insecure like the current wireless network. The PPP(Privacy Protection Protocol) for establishing a session key involves the exchange of four messages; it is illustrated in Figure.2. below. When the initiator Tag transmits the information to the responder Reader, he transmits

Message 1. T $->$ R : Anonymous(T)$*$
Message 2. R $->$ DB : Anonymous(T)$*$, E$_{DBpublic}$ $\{T, Privacy(R), SkeyR\}$
Message 3. DB $->$ R : SkeyR (+) SkeyT
Message 4. R $->$ T : E$_{SkeyT}$ $\{R\}$

Anonymous(T)$*$: E$_{DBpublic}$ $\{R, Privacy(T), SkeyT\}$: will be just forwarded to DataBase

Fig. 2. The privacy protection protocol for secure RFID system.

a anonymous value that contains reader's identity(*R*) he can receive in this system, privacy value(*Privacy(T)*) of Tag, and session key(*SkeyT*) of Tag (for Reader which the server will later transmit the session key to be decrypted by authenticated Reader), then sends it to the server DB (Message 1). Since the responder Reader can not acknowledge the *Anonymous(T)*, he forward the message to the Server(DataBase) with encrypted message that contains tag's identity(*T*) he want to access in this system, privacy value(*Privacy(R)*) of Reader, and session key(*SkeyR*) of Reader, then encrypt it with DataBase's public key and transmit the all messages to DataBase(Message 2). The server DB forms the Vernam encryption of the two keys(*SkeyT, SkeyR*) he has received, and returns these to Reader. When Reader receives this message, he can decrypt the *SkeyT* using the *SkeyR*(Message 3). Finally, Reader can transmit the his real identity to

Tag using session key(*SkeyT*) securely(Message 4). After all steps finishing, Tag can transmit his information to the authenticated Reader.

5.2 Modeling the Privacy Protection Protocol using Casper

In this section we give brief description of how we can model the PPP(Privacy Protection Protocol) in Casper. We give a brief overview instead of whole script due to the limitation of space.(you can find the whole script at[10])

```
#Protocol description

0.      -> R  : T
1.  T   -> R  : {R, privacy(T), kT}{pkdb}%AnonyT
2.  R   -> DB : {T, privacy(R), kR}{pkdb},
                AnonyT%{R, privacy(T), kT}{pkdb}
3.  DB -> R   : kT (+) kR
4.  R   -> T  : {R}{kT}
```

In *# Protocol description*, to authenticate the reader from database and tag, at the first line, Message 0 means that R(Reader) must communicate with T(Tag). In Message 1, T transmits the encrypted value with DB's public key(*pkdb*) to R and the encrypted value include R's identity(R), anonymous value(privacy(T)) of T's identity using privacy function, and session key(kT) generated from T. This message would be used to forward from T to DB using % notation. In Message 2, R forwards the *AnonymousT* with the message that contains tag's identity(*T*), privacy value(*Privacy(R)*) of Reader, and session key(*kR*) of Reader, then encrypt it with DB's public key(*pkdb*) and transmit the all messages to DB. In Message 3, we introduce *exclusive-or(+) technique* called Vernam encryption into this protocol. After DB received the two session keys from R and T, he forms the Vernam encryption and transmits these to the R. As a result, if R get the another session key, they will use it to communicate their information such as EPC[2] and thus mutual authentication between them can be satisfied. In message 4, R can transmit the his real identity(*R*) to T using session key(*kT*) securely.

5.3 The Result of Verification

In this paper, we show verification results of the safety specifications in PPP(Privacy Protection Protocol) scheme, we use traces refinement provided in FDR tool. After running the FDR model checking tool, this protocol satisfies the Secret and Agreement requirements in Casper script and the testing result of the protocol can be described in CSP like below.

- $Secret_{R,T}$(tr) = $\forall$ m $\bullet$ signal.*Claim_Secret.R.T*.m in *tr* $\wedge$ R $\in$ Honest $\wedge$ T $\in$ Honest $\Rightarrow$ $\neg$ (*leak.kR.kT* in tr)
  For all message m, through trace specification(tr), the session key kR and kT were not leaked by an intruder. That is, at the message 1, 3, confidentiality of the kR and kT can be ensured through public key encryption scheme. It prevents a replay and man-in-the-middle attack between agents.

- $Secret_{T,R}$(tr) = $\forall$ m $\bullet$ signal.*Claim_Secret.T.R*.m in *tr* $\wedge$ T $\in$ Honest $\wedge$ R $\in$ Honest $\Rightarrow \neg$ (*leak.kR.kT* in tr)
  For all message m, through trace specification(tr), the session key kR and kT were not leaked by an intruder. T can believe the authentication through the kR and kT because all the messages was transmitted using exclusive-or technique from DB.
- R $\in$ Honest $\Rightarrow$ *signal.Running_Initiator.T.R.kT.Kr* precedes *signal.Commit_Responder.R.T.kT.kR*
  In this protocol, we can guarantee that the corresponding Running signal has occurred provided we assume that the initiator is honest: that *R* $\in$ *Honest*. This results in a successful key agreement between two agents through authenticated channel by kR and kT.

6 Conclusions

Mobile and Ubiquitous computing is defined as environments where users can receive network services for anytime and anywhere access through any device, connected with a wired and wireless network to information appliances including the PC. In this environment, there are many security threats that violate user privacy and interfere with services. In this paper, we focus on proposal of Ensuring Privacy Protocol which can be widely researched in RFID system and safety analysis of the protocol using Casper, CSP, and FDR. In verifying this protocol with FDR tool, we were able to confirm prevent this protocol from some of the known security vulnerabilities which are likely to occur in RFID system.

References

1. Sarma, S., Weis, S., Engels,D.: RFID systems and security and privacy implications. Workshop on Cryptographic Hardware and Embedded Systems(CHES) 2002. LNCS No. **2523** (2003) 454-469
2. EPCGLOBAL INC.: http://www.epcglobalinc.org.
3. Gong, L., Needham, R., Yahalom, R.: Reasoning about Belief in Cryptographic Protocols. The 1990 IEEE Symposium on Security and Privacy (1990) 18-36
4. Juels, A., and Weis, Stephen.: Authenticating Pervasive Devices with Human Protocols. Advances in Cryptology - CRYPTO (2005), LNCS, volume 3621, 293-308.
5. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall. Englewood Cliffs. NJ (1985)
6. Lowe, G.: Casper: A compiler for the analysis of security protocols. Proceeding of the 1997 IEEE Computer Security Foundations Workshop X. IEEE Computer Society. Silver Spring. MD (1997) 18-30
7. Formal Systems Ltd. FDR2 User Manual. Aug. (1999)
8. Ryan, P. Y. A., Schneider, S. A.: Modelling and Analysis of Security Protocols: the CSP Approach. Addison-Wesley (2001)
9. Ohkubo, M., Suzuki, K., Kinoshita, S.: Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID. Proc. of the SCIS 2004. (2004) 719-724
10. http://formal.korea.ac.kr/∼hskim/PPP