# Formal Analysis of RFID Authentication Protocol for Security System

Hyun-Seok Kim

Dept. of Research and Development, Mobile Technology Convergence Center,
Daegu Technopark, Dalseogu, Daegu, Republic of Korea
hyunseok.kim@gmail.com

***Abstract*- Recently, Radio frequency identification (RFID) has been an important and ubiquitous infrastructure technology for smart work system. As RFID tags are affixed to all items, they may be used to support various useful services. Security mechanisms for RFID systems, such as authentication and encryption, are therefore of utmost importance. However, there are many risks involved, for example user privacy violations and service interference. Therefore, security service is required to block these risk elements, and user authentication is an essential component for secure RFID system such as smart work system. In this paper, an authentication protocol for secure communications is proposed for secure RFID network environments and also for verified safety using GNY logic.**

***Keywords- RFID (Radio Frequency Identification); Formal Methods; Modal Logic; Security Protocol***

## I. INTRODUCTION

In the RFID security domain, various issues are related to data protection of tags, message interception over the air channel, and eavesdropping within the interrogation zone of the RFID reader [1, 2]. This topic has been so far been dominated by the topics of data protection associated with data privacy and authentication between tag and reader for smart work system. In this paper, when using RFID, two aspects on the risks imposed on the passive party are discussed.

Firstly, the data privacy problem is such that storing person-specific data in a RFID system can threaten the privacy of the passive party. This party may be, for example, a customer or an employee of the operator. The passive party uses tags or items that have been identified as tags, but the party has no control over the data stored on the tags.

Secondly, authentication is carried out when the identity of a person or program is verified. Then, on this basis, authorization takes place, i.e. rights, such as the right of access to data. In the case of RFID systems, it is particularly important for tags to be authenticated by the reader and vice-versa. In addition, readers must also authenticate themselves to the backend, but in this case, there are no RFID-specific security problems.

To satisfy the above requirements, 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 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, 4], and tool based methodology [5, 6]. In this paper, the [1] hash-based RFID authentication protocols which employs hash functions to secure RFID communication are specified and verified whether this protocol satisfies security properties such as secrecy and authentication using GNY logic (Gong L., Needham R., and Yahalom R.) [15] as the Modal logic [3] methodology. After verifying the protocols as GNY logic, the existence of known security flaws in the protocols is confirmed, and the problems of the hash based technique are described. 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 Ⅱ describes related work on RFID security and authentication schemes associated with hash functions. In Section Ⅲ, the use of modal logic (GNY) is outlined for analyzing security protocols. Section Ⅳ describes the analyzed result of the protocol. Section Ⅴ presents the proposed security scheme. Section Ⅵ addresses conclusions and future work.

## II. RELATED WORK

There has been much literature attempting to address the security concerns raised by the use of RFID tags.

### *A. The Hash Lock Scheme*

A reader defines a "Lock" value by computing lock = hash (key) [7], where the key is a random value. This lock value is sent to a tag and the tag stores this value in its reserved memory (i.e. a metaID value), the tag then enters into a locked state automatically. To unlock the tag, the reader transmits the original key value to the tag, and the tag performs 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 values match, the tag is unlocked. Once the tag is in an unlocked state, it can transmit its identification number, such as the Electronic Product Code (EPC) [2] to readers' queries in the forthcoming cycles. This approach is simple and straightforward in achieving data protection, i.e. the EPC code stored in the tag is being protected. An authorized reader is able to unlock and read the tag, then lock the tag again after reading the code. This scheme is analyzed in Section IV in detail.

### *B. The Randomized Hash Lock Scheme*

This is an extension of hash lock [7] based on pseudo random functions (PRFs). An additional pseudo-random number generator is required to be embedded into tags for this approach. Presently, tags respond to reader queries using a pair of values (r, hash(ID*k* ‖ *r*)), where *r* is the random number generated by a tag, ID*k* is the ID of the *k*-th tag among a number of tags in ID1, ID2, . . ., ID*k*, . . ., ID*n*. For reader queries, the tag returns two values. The first is the random number. The second is a computed hash value based on concatenation (‖) of its ID*k* and *r*. When the reader obtains these two values, it retrieves the current *N* number of ID (i.e. ID1, ID2, . . ., ID*n*) from the backend database. The reader will perform the above hash function on each ID from 1 to *n*, with *r*, until it finds a match. When the reader finds a match, the reader is able to identify the tag *k* is on its tag's ID list (i.e. tag authentication). The reader will then transmit the ID*k* value to the tag for unlocking. Once the tag is in an unlocked state, the reader can obtain its EPC code in the subsequent reading cycle.

In addition to achieving RFID tag security, this scheme also provides location privacy. In the hash lock scheme, tags still disclose metaID values. However, this approach only discloses *r* and the hashed value.

### *C. The Chained Hash Scheme*

Ohkubo et al.[8][9] suggested the chained hash procedure as a cryptographically robust alternative. In every activation, the tag calculates a new metaID, using two different hash functions. First, the current metaID is hashed in order to generate a new metaID, which is then hashed again with the aid of the second function. It is this second metaID that is transmitted to the reader. For the purpose of decoding, the reader must hash until a match with the metaID transmitted from the tag has been found. The advantage of this procedure is that it is not sensitive to repeated attempts to eavesdrop the metaID during transmission via air waves.

### *D. Other Approaches*

Another hash-based approach is *Hash based Varying Identifier* proposed by Henrici and Müller [10]. Their scheme also adopts a hash function and a random number generator (RNG), but a pseudo random number is generated by a back-end server and transmitted to the tag every interrogation, to make the tag's queried identifier random and preserve location privacy.

Hwang et al. [11] proposed an improved authentication protocol of *Hash based Varying Identifier*. In their scheme, the main difference is that a reader has a random number generator to protect against a man-in-the-middle attack.

## III. FORMAL METHODS FOR SECURITY PROTOCOLS

Modal Logic: GNY (Gong L., Needham R., and Yahalom R.) [15] logic is used to reason about security protocols. GNY logic is a direct successor to BAN [3] logic and is quite powerful in its ability to uncover even subtle protocol flaws. Discussion of the virtues and limitations of the logic can be found in [12].

In GNY logic, message extensions are added to the protocol description during protocol formalization, so that principals can communicate their beliefs and thus reason about each other's beliefs. The use of message extensions enables the logic to deal with different levels of trust among protocol principals. As such, it is considered an improvement over BAN logic, which assumes that all principals are honest and competent. This development is noteworthy as many protocol attacks are performed by dishonest principals. As an example of a message extension, consider the following: P → Q: {K; P}Ks- is formally stated as Q ◁ *{*K, P}Ks- ~> S ⊨ P $\overset{K}{\leftrightarrow}$ Q. This means that principal Q is informed of a session key, K, and an identity, P, encrypted under the private key of principal S. The session key, K, is marked with a not-originated-here asterisk. Q is informed that S believes K is a suitable shared secret for P and Q.

The postulates of GNY logic are used to deduce whether protocol goals can be derived from the initial assumptions and protocol steps. If such a derivation exists, the protocol is successfully verified.

Logic-based formal verification involves the following steps (Fig. 1):

1. Formalization of the protocol messages;
2. Specification of the initial assumptions;
3. Specification of the protocol goals;
4. Application of the logical postulates.

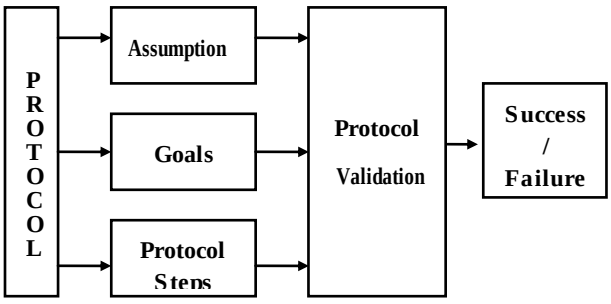


Fig. 1 The process of verification with modal logic

The first step in logic-based verification involves specifying the protocol in the language of the logic by expressing each protocol message as a logical formula. This step is known as protocol formalization (some authors also refer to it as idealization). A formal description of the protocol, obtained by formalization, does not simply list the components of each message but attempts to show the purpose of these components so as to avoid ambiguity.

The second step in the verification process involves formally specifying the initial protocol assumptions. These

assumptions reflect the beliefs and possessions of the involved principals at the beginning of each protocol run.

In the third step, the desired protocol goals are expressed in the language of the logic. These goals are specified in terms of the beliefs and possessions of the protocol participants at the end of a successful protocol run.

The final verification step concerns the application of logical postulates to establish the beliefs and possessions of protocol principals. The objective of the logical analysis is to verify whether the desired goals of the protocol can be derived from the initial assumptions and protocol steps. If such a derivation exists, the protocol is successfully verified; otherwise, verification fails. A successfully verified protocol can be considered secure within the scope of the logic. On the other hand, even the results of failed verification are helpful, as these may point to missing assumptions or weaknesses in the protocol. If a weakness is discovered, the protocol should be redesigned and reverified. However, verification logic techniques have their limitations, not least of which is the likelihood of errors in protocol formalization. The number of opportunities to make such mistakes increases as the verification process becomes more complicated, requiring a thorough understanding of the logic used. During the verification process, the semantics of the protocol must be interpreted, in order to specify the meaning that a protocol message is intended to convey. This 'interpretation process' is somewhat controversial—different authors may interpret the same messages differently. If the formalized protocol does not properly represent the original design, then the proof demonstrates only that the protocol corresponding to this formal description is secure. However, no claims can be made on the security of the original design. Lack of clarity about protocol goals and initial assumptions is a further cause for concern.

In some cases the same protocol may be used for slightly different purposes. For example if a protocol is used to generate a new session key, each principal involved in the protocol run may require that the other principal believes the session key to be a shared secret. This property is known as second level belief. If a protocol is verified as secure for first level belief only and used in an application where second level belief is required, serious security breaches are likely. Hence, it is vital to note the assumptions and goals under which a security protocol is considered secure during its formal verification.

Despite these criticisms, different logic techniques have identified numerous protocol weaknesses and are considered as successful. Gligor et al. [13] summarize the virtues of authentication logic as follows:

- They help formalize reasoning about useful abstract properties of cryptographic protocols.
- They force designers to make explicit security assumptions.
- They achieve a reasonably well-defined set of authentication goals.

## IV. THE RFID AUTHENTICATION PROTOCOL AND ITS VERIFICATION

Firstly, the behavior of the hash unlocking protocols is modeled as hash unlocking of the hash lock scheme. The simple description of the hash locking is already described in Section II-A. The role of the reader simply writes the metaID as a keyed hash value in the tag.

The general overview of the authentication protocol (Table I, Fig. 2) is as follows.

TABLE I 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 Function |
| **ID** | Information Value of Tag |
| **Xn** | A Random Nonce Generated by X |
| **H** | Hash Function |
| $\mathbf{E_{key}(M)}$ | Encrypted Message with Key |

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. 2 The hash unlocking protocol overview

- Message 1: Request by the reader.
- Message 2: The tag transmits the metaID (locked value as hashed key) to the reader.
-Message 3: The reader forwards the metaID to the Database.
-Message 4: The database transmits the original key value and tag ID to the reader after checking the match between metaID from the reader and metaID in the database.
-Message 5: The reader transmits original key to the tag to ensure tag authentication.
-Message 6: The tag transmits its information value to the reader.

TABLE II NOTATION OF GNY LOGIC

| | |
|---|---|
| **(X,Y)** | Concatenation of two formulae |
| **{X}K, {X}K-** | Symmetric encryption and decryption |
| **#(X)** | The formula X is fresh. X has not been sent in a message at any time before the current run of the protocol |
| **□(X)** | Formula X is recognizable |
| **P◁ X** | P has a received a message containing X and P can read and repeat X, possibly after performing some decryption |
| **P◁ *(X)** | |
| **P ∋ X** | P is told formula X which he did not convey previously during the current protocol run |
| **P \|~ X** | P possesses or is capable of possessing formula X |
| **P \|≡ X** | P conveyed X |
| **X ~> C** | P believes X. That is, the principal P acts as if X is true |
| **P \|⇒ X** | Formula X has the extension C. The precondition for X being conveyed is represented by statement C |
| **$P \xleftrightarrow{K} Q$** | P has jurisdiction over X. The principal P is an authority on X and should be trusted on this matter. This construct is used when a principal has delegated authority over some statement |
| | K is a suitable secret for P and Q. They may use it as a key to communicate or as a proof of identity |

### A. Formalization of the Protocol Step

*M 1.* $R \triangleleft *metaID \sim> R \mid\equiv \xrightarrow{H(RKey)} T,$

$T \mid\equiv R \mid\sim H(RKey)$

*M 2.* $DB \triangleleft *metaID$

*M 3.* $R \triangleleft RKey, *ID \sim> R \mid\equiv \xrightarrow{RKey} DB,$

$R \mid\equiv \xrightarrow{ID} DB$

*M 4.* $T \triangleleft RKey$

*M 5.* $R \triangleleft ID$

Fig. 3 Formalization of the protocol step

A formalized version of the protocol is shown in Fig. 3 (from Table II). The asterisks denote the ability of each principal to recognize that it did not transmit the received message at an earlier stage in the protocol.

In M1, the reader is told the metaID (locked value as hashed key) from the tag and the message extension in the first message indicates that if a reader transmits a H(RKey) to lock a tag, then the tag believes that RKey contained in that metaID belongs to the reader. In M2, the DB is told the metaID from the reader and it means the metaID is forwarded from the reader to DB. In M3, the reader is told the original key value and tag ID from the database to the reader after checking the match between metaID from the reader and metaID in the database and the message extension in the third message indicates that if the reader receives RKey and ID from some principal, then the reader believes that RKey contained in that metaID belongs to the DB. In M4, the tag is told the original key from the reader and in M5, the reader is told the tag ID from the tag.

### B. Specification of the Initial Assumptions

The initial assumptions for the hash unlocking protocol are as follows:

$T \ni metaID ; T \ni RKey ; T \ni ID;$
$DB \ni RKey ; DB \ni ID ;$
$R \mid\equiv □(RKey); R \mid\equiv □(ID);$
$T \mid\equiv \xrightarrow{RKey} DB; T \mid\equiv \xrightarrow{ID} DB;$
$T \mid\equiv \overrightarrow{DB} \mid\Rightarrow DB \mid\equiv *; R \mid\equiv DB \mid\Rightarrow DB \mid\equiv *;$

The first two rows state the possessions of both principals. Each principal possesses its information, its symmetric key and its identification data. The next row states the recognizability assumptions. Reader recognizes the symmetric key and other's identification data. The final two rows concern beliefs regarding the database server. Tag believes that RKey is the symmetric key between DB and Reader; ID is a secret value for DB and Tag, that DB is honest and competent, and that DB has jurisdiction over the other principal's symmetric key.

### C. Specification of the Protocol Goal

The goals of the hash unlocking protocol are as follows:

$R \mid\equiv \#H(RKey); T \mid\equiv \#H(RKey);$
$T \mid\equiv R \mid\sim RKey; R \mid\equiv T \mid\sim ID;$
$R \ni ID$

The goals in the first row state that both principals believe it to be fresh. The next row concerns authentication: each principal should believe that its counterpart conveyed the respective identification data. The goal on the remaining row describes the confidentiality of the information.

### D. Application of the Logical Postulates (from Appendix A)

*M 1.* $R \triangleleft *metaID \sim> R \mid\equiv \xrightarrow{H(RKey)} T, T \mid\equiv R \mid\sim H(RKey)$

- Applying T1 to M 1 yields $R \triangleleft metaID$. R is told T's metaID without not-originated-here asterisk.
- Applying P1 yields $R \ni metaID$. The reader possesses the metaID value of the tag.
- Since R recognizes RKey, by R1 $R \mid\equiv □(H(RKey))$. R recognizes the H(RKey).
- However, R cannot believe that metaID is the valid current value of the tag. The preconditions of J2 are not achieved and the freshness of H(RKey) is not satisfied. An intruder could use an old compromised hash value belonging to the tag in order to masquerade as the reader.

*M 2.* $DB \triangleleft *metaID$

- Applying T1 to M 2 yields *DB ◁ metaID*. DB is told T's metaID without not-originated-here asterisk.
- Applying P1 yields *DB ∋ metaID*. The database possesses the metaID value of the tag.
- However, R still cannot believe that metaID is the valid current value of the tag. The preconditions of J2 are not achieved as in *M 1*. An intruder still could use an old compromised hash value belonging to the tag in order to masquerade as the reader.

*M 3. R ◁ RKey, *ID ~> R* $|\equiv \xrightarrow{RKey} DB, R |\equiv \xrightarrow{ID} DB$

- Applying T1 and P1 yields *R ∋ (RKey, ID)*. The reader possesses the (RKey, ID). By T2, *R ∋ RKey, R ∋ ID*.
- However, R cannot believe that RKey is the valid current value from the tag's metaID. Since the freshness of RKey is not satisfied, the reader cannot transmit RKey to the tag.

*M 4. T ◁ RKey*

*M 5. R ◁ ID*

- Applying T1 and P1 to M4 and M5 yields T *∋ RKey*, R *∋ ID*.
- However, by I4, J2, the tag cannot believe that the reader transmits RKey to the tag. The reader cannot believe that the tag transmits the ID to the reader.

### *E. Weakness in the Hash Unlocking Protocol*

The above verification of the hash unlocking protocol identifies the following failed goals:

1. R cannot derive that the H(RKey) is fresh;
2. T cannot derive that the H(RKey) is fresh;
3. T cannot derive that R conveyed RKey;
4. R cannot derive that T conveyed ID;
5. R cannot derive that ID is valid;

## V. THE PROPOSED STRONG AUTHENTICATION PROTOCOL FOR RFID SYSTEMS

### *A. Analysis of the Strong Authentication Protocol Using GNY Logic*

In the previous schemes [7-11], it is assumed that database is a TTP (Trusted Third Party) and the communication channel between reader and database is secure. However, this paper assumes that database is not a TTP and the communication channel is as insecure as current wireless networks. It is also assumed that k is the secret session key shared between reader and database, and reader and database have enough capability to manage the symmetric-key crypto-system and sufficient computational power for encryption and decryption.

To satisfy security requirements, the most effective protective measure against an attack involving eavesdropping at the air interface is not to store any contents on the tag itself and instead to read only the ID of the tag that database has transmitted to be scanned from reader. This measure, which is most often recommended in the technical literature and which is assumed by EPC global [2], offers the additional advantages that less expensive tags can be used; the memory for the associated data in the database is practically unlimited. The main idea of this framework is based on the security algorithm employed in the Yahalom protocol [14, 15].

The proposed protocol must guarantee the secrecy of the session key: in Messages 4, 5, the value of the session key must be known only by participants playing the roles of T and R. R and T also must be properly authenticated to the DB.

Message 1. R -> T : Query
Message 2. T -> R : Tn
Message 3. R -> DB: $E_{ServerKey(R)}$ ( T, Tn, Rn )
Message 4. DB -> T : $E_{ServerKey(T)}$ ( R, DBkey, Tn, Rn, ID)
Message 5. DB -> R : $E_{ServerKey(R)}$ ( T, DBkey )
Message 6. T -> R : $E_{DBkey}$ ( ID )

Fig. 4 Overview of the proposed strong authentication protocol

The main idea of the proposed protocol is that the ***ServerKey*** and ***Tag's Nonce(Tn)*** is used to minimize the burden of the Tag and to ensure authentication between Tag and Reader. The definition of a function called ***ServerKey*** that takes in the name of a Server and returns a ***ServerKey*** could be regarded as shared: ***Agent -> ServerKey***. If reader would like to transmit any messages to database, then he would use the ***ServerKey*** with his identity as parameter. This description resembles a functional programming language.

The general description of the proposed protocol is described as follows;

- Message 1: Query request by the reader.
- Message 2: T is defined to take a random nonce Tn and transmit R. This makes simple challenge-response easy.
- Message 3: Through ***T, Tn,*** and ***Reader's Nonce (Rn)*** with ***Server Key***, R can ensure database authentication.
- Message 4: DB encrypts all of the ***R, DBkey, Tn, Rn, and ID*** received from R and transmits these to T to allow R to authenticate securely using the server key.
- Message 5: DB also transmits ***T, DBkey*** to R to decrypt Tag's ID.
- Message 6: T can transmit ***ID*** securely using the DBkey received in Message 4.

In addition, Messages 4, 5 mean the protocol step that can be transmitted from database to other participants simultaneously to decrypt the tag's ID in Message 6.

*1) Formalization of the Protocol Steps:*

*M 1. R ◁ *Tn*

*M 2. DB ◁ *{T, Tn, Rn}K(R)*

*M 3. T ◁ {*R, *DBKey, Tn, *Rn, *Id}K(T)*

*M 4. R ◁ {T, *DBKey}K(R)*

*M 5. R ◁ {*Id}DBKey*

Fig. 5 The formalization of the protocol step

A formalized version of the protocol is shown in Fig. 5. The asterisks denote the ability of each principal to recognize that it did not transmit the received message at an earlier stage in the protocol. The protocol step in message 1 (Fig. 4.) was omitted in Fig. 5.

*2) Specification of the Initial Assumption:*

The initial assumptions for the proposed protocol are as follows;

$$T \ni Tn;\ T \ni K(T);\ R \ni Rn;\ R \ni K(R);$$
$$DB \ni Id;\ DB \ni DBKey;\ DB \ni K(T);\ DB \ni K(R);$$
$$T \mid\equiv \sharp(Id);\ T \mid\equiv \sharp(T, DBKey);$$
$$R \mid\equiv \sharp(Id);\ R \mid\equiv \sharp(DBKey);$$
$$T \mid\equiv \#Tn;\ R \mid\equiv \#Rn;\ DB \mid\equiv \#DBKey;$$
$$T \mid\equiv \xrightarrow{DBKey} DB;\ T \mid\equiv \xrightarrow{K(T)} DB;\ T \mid\equiv (DB \Rightarrow \xrightarrow{DBKey} R);$$
$$R \mid\equiv \xrightarrow{DBKey} DB;\ R \mid\equiv \xrightarrow{K(R)} DB;\ R \mid\equiv (DB \Rightarrow \xrightarrow{DBKey} T);$$

The first two rows mean that each principal possesses its random nonce, symmetric key and information data. The next two rows state that the tag and reader recognize the other's symmetric key and information data. The next row means that each principal believes its nonce or key freshness. The final two rows concern beliefs regarding the database server that DB has jurisdiction over its own key and the other principal's symmetric key.

*3) Specification of the Protocol Goal:*

The goals of the proposed protocol are as follows;

$$DB \mid\equiv \#\{T, Tn, Rn\}K(R);$$
$$T \mid\equiv \#\{R, DBKey, Tn, Rn, Id\}K(T);$$
$$R \mid\equiv DB \mid\sim \{DBKey\}K(R);\ R \mid\equiv T \mid\sim \{ID\}DBKey;$$
$$T \mid\equiv T \xleftrightarrow{DBKEY} R;\ R \mid\equiv T \xleftrightarrow{DBKEY} R;$$
$$R \ni Id$$

The first three rows concern authentication: each principal should believe that its counterpart is conveyed in the respective identification data. The goals in the fourth row describe key agreement: both principals should possess the shared key through a challenge-response process. The goal on the remaining row describes the confidentiality of the information.

*4) Application of the Logical Postulates (from Appendix A):*

*M 1. $R \triangleleft *Tn$*

- Applying T1 and P1 yields $R \ni Tn$. The reader possesses the T's random nonce.

*M 2. $DB \triangleleft *\{T, Tn, Rn\}K(R)$*

- Applying T1 and T3 yields $DB \triangleleft T, Tn,$ and $Rn,$ by T2 and P1 $DB \ni T, DB \ni Tn, DB \ni Rn.$
- Applying F1 yields $DB \mid\equiv \#\{T, Tn, Rn\}K(R)$ and satisfies the goal at the first row in V.A.3.

*M 3. $T \triangleleft \{*R, *DBKey, Tn, *Rn, *ID\}K(T)$*

- Applying T3 yields $T \triangleleft (*R, *DBKey, Tn, *Rn, *ID).$
- Applying T2 and T1, P1 yields $T \ni DBKey, T \ni Tn, T \ni Rn,$ *and* $T \ni ID.$
- Applying F1 yields $T \mid\equiv \#\{R, DBKey, Tn, Rn, ID\}K(T)$ and satisfies the goal at the second row.

*M 4. $R \triangleleft \{T, *DBKey\}K(R)$*

- Applying T3 yields $R \triangleleft \{T, *DBKey\}.$
- Applying T2, T1 and P1 yields $R \ni DBKey.$
- Applying I4 yields $R \mid\equiv DB \mid\sim \{DBKey\}K(R)$ and satisfies the first goal at the third row.
- Applying R ∋ DBKey and I4, yields $R \mid\equiv T \mid\sim \{ID\}DBKey$ and satisfies the second goal at the third row.

*M 5. $R \triangleleft \{*ID\}DBKey$*

- Applying T3 and P1 yields $R \ni ID$ and satisfies the goal at the last row.

  Through $T \ni DBKey$ in M3. and $R \ni DBKey$ in M4., the goals $(T \mid\equiv T \xleftrightarrow{DBKEY} R;\ R \mid\equiv T \xleftrightarrow{DBKEY} R;)$ at the fourth row.

TABLE III COMPARISON AMONG PROTOCOLS (O: SECURE, -: INSECURE)

| *Lists* | *H.L. (Hash Lock)* | *R.H. (Randomized Hash)* | *C.H. (Chained Hash)* | *Proposed* |
|---|---|---|---|---|
| Data Confidentiality | - | - | - | O |
| Tag Anonymity | - | - | - | O |
| Data Integrity | - | O | O | O |
| Reader Authentication | - | O | O | O |
| DB Authentication | O | O | - | O |
| Mitm Attack | - | - | - | O |
| Replay Attack | - | O | - | O |

From Table III, it can be seen that the proposed protocol meets all security requirements listed above. These protocols were primarily designed to provide link security to protect against passive and active attacks over the air interface. Due to the limitation of the space, all result that been analyzed the vulnerabilities about other protocols, randomized protocol and chained hash protocol were described in brief in Table III.

*B. The Result of Verification*

After verifying the protocols using GNY logic, it is confirmed that the proposed protocol solves the security weakness in previous hash-based protocols.

- Secrecy: Spoofing, Replay Attack, Tracking, Eavesdropping on communication between tag and reader are attacks that threaten all participants. To protect from these attacks, the countermeasures are therefore essentially identical in this protocol as follows. Firstly, all data are shifted except ID to the backend. This is also to be recommended for reasons of data management (i.e. the ID for the tag existing at the backend database will be shifted to protect spoofing and eavesdropping attacks to the tag through the database when the reader sends a request).
- Secondly, data transmission is encoded. Encryption of the data transmission is supported to ensure authorized access to the data of concern and to protect replay attacks and tracking.
- Authentication: When a tag receives a "get challenge (query)" command from a reader, it generates a random number Tn and sends this number to the reader. The reader

in turn generates a random number Rn with it and the random number Tn generates an encrypted data block (token T) on the basis of an encryption algorithm and server key (R). The data block is then returned to the database to authenticate the reader. The reader and tag both use the same encryption algorithm and since the server key is stored on the tag, the tag is capable of decrypting the server key (T). If the original random number Tn and the random number Tn, which has now been decrypted, are identical, then the authenticity of the tag vis-a-vis the reader is demonstrated.

### C. *Availability Problem of Proposed Protocol*

In this paper, we propose the strong symmetric key algorithm based RFID authentication protocol. Regarding performance of the protocol at an application level, our assumption is that CPUs are now faster and memory and network speeds have also increased, but not nearly as much as CPU speeds. Pure computation, such as is used in a block cipher, is cheaper in both absolute terms and relative to other tasks, such as writing the data to disc. Unlike DES, nearly all AES candidates are designed for high performance in software.

It could be argued that for most applications, nearly all AES algorithms are fast enough. Some literature [16, 17] reached the point where cryptography is not a significant portion of the total CPU burden, and the relative speed of the algorithms no longer matters very much. Therefore, our proposed protocol can be available for light-weight tags in the RFID system.

## VI. DISCUSSION AND CONCLUSIONS

Smart work is defined as environments where users can receive smart work system services for anytime and anywhere access through any device, connected with a wired and wireless network to home information appliances including the PC. In this environment, there are many security threats that violate user privacy and interfere with smart work services. Especially, the smart work consists of several networks with RFID system therefore authentication between the reader and the appliance devices affixed tag is required.

In this paper, the RFID security requirements in smart work environments are defined, and authentication mechanism among reader, tag and database is proposed. The focus is to analyze the vulnerabilities of the protocol using formal methods and to design and verify the secure authentication protocols, which is widely researched in RFID systems. In verifying these protocols using GNY logic, it is possible to confirm some of the known security vulnerabilities likely to occur in RFID systems.

Finally, a strong authentication protocol based encryption algorithm, is proposed for guarding against man-in-the-middle, and replay attacks, and also for verifying safety using GNY logic.

## ACKNOWLEDGEMENT

This work is supported by Business of 2009 Strategy Planning Technology Development for “Development of vehicle convergence Devices based on DSRC/WAVE”

## REFERENCES

[1] S. Sarma, S. Weis, and D. Engels, “RFID systems and security and privacy implications”, *Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2002*, LNCS No. 2523, pp. 454-469, 2003.

[2] EPCGLOBAL INC.: http://www.epcglobalinc.org.

[3] M. Burrows, M. Abadi, and R. Needham, “A Logic of Authentication”, *ACM Operating System Review*, Vol.23, No.5, pp.1-13, 1989.

[4] C.A.R. Hoare, Communicating Sequential Processes, Prentice-Hall, Englewood Cliffs, NJ, 1985.

[5] G. Lowe, “Casper: A compiler for the analysis of security protocols”, *The 1997 IEEE Computer Security Foundations Workshop X, IEEE Computer Society*, Silver Spring, MD, pp. 18-30, 1997.

[6] Formal Systems Ltd. FDR2 User Manual, Aug. 1999.

[7] 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.

[8] M. Ohkubo, K. Suzuki, and S. Kinoshita, “Cryptographic Approach to Privacy-Friendly Tags”, *RFID Privacy Workshop*, Massachusetts Institute of Technology, Cambridge, MA, USA.

[9] M. Ohkubo, K. Suzuki, and S. Kinoshita, “Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID”, *Symposium on Cryptography and Information Security*, pp.719-724, 2004.

[10] D. Henrici and P. Müller, “Hash based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers”, *PerSec’04 at IEEE PerCom*, pp.149-153, 2004.

[11] Y. Hwang, S. Lee, D. Lee, and J. Lim, “An Authentication Protocol for Low-Cost RFID in Ubiquitous”, *CISC S’04,* pp.109-114, 2004.

[12] A. Mathuria, R. Safavi-Naini, and P. Nikolas, “Some Remarks on the Logic of Gong, Needham, and Yahalom”, *The International Computer Symposium*, Vol.1, pp.303-308, 1994.

[13] V. D. Gligor, R. Kailar, S. Stubblebine, and L. Gong, “Logics for Cryptographic Protocols – Virtues and Limitations”, *Computer Security Foundation Workshop*, pp. 219-226, 1991.

[14] C. Lawrence Paulson. “Relations between Secrets: Two Formal Analyses of the Yahalom Protocol”, *IEEE Computer Security*, 2001.

[15] L. Gong, R. Needham, and R. Yahalom. “Reasoning about Belief in Cryptographic Protocols”, *The 1990 IEEE Symposium on Security and Privacy*, pp. 18-36, 1990.

[16] M. Roe, “Performance of Protocols: Security Protocols”, *Lecture Notes in Computer Science 1796* , pp.140-146, 2000.

[17] H. S. Kim, J. H. Oh, and J. Y. Choi, “Security Analysis of RFID Authentication for Pervasive Systems using Model Checking”, *The thirtieth Annual International COMPSAC*, pp. 195-202, 2006.

**Hyun-Seok Kim** received the B.S. degree in the Department of Business Management from Korea Military Academy, Seoul, Korea in 2000 and MS, Ph.D degree in the Department of Computer Science and Engineering at Korea University, Seoul, Korea in 2006, 2009 respectively. He is currently a senior researcher of research and development department of Daegu Techno Park in Republic of Korea. His research interests include the areas of Formal methods (formal specification, verification, and model checking), security authentication system design & verification, key authentication & exchange scheme, mobile commerce anonymity, and smart card privacy protection.

## APPENDIX A. GNY LOGICAL POSTULATES

In this appendix we list the logical postulates of GNY logic

used throughout this paper.

$$\text{T 1}: \frac{P \lhd {*X}}{P \lhd X}$$

If a principal is told a formula is marked with a not-originated-here asterisk, then the principal is told that formula.

$$\text{T2}: \frac{P \lhd (X, Y)}{P \lhd X}$$

Being told a formula implies being told each of its concatenated components.

$$\text{T3}: \frac{P \lhd \{X\}K, P \ni K}{P \lhd X}$$

If a principal is told that he possesses a formula encrypted with a key, then he is considered to have been told the decrypted contents of that formula.

$$\text{P1}: \frac{P \lhd X}{P \ni X}$$

A principal is capable of possessing anything he is told.

$$\text{F1}: \frac{P \models \#(X)}{P \models \#(X, Y), P \models \#(F(X))}.$$

If a principal believes that a formula X is fresh, then it is believed that any formula of which X is a component is fresh and that a computationally feasible one-to-one function, F, of X is fresh.

$$\text{R1}: \frac{P \models \square (X)}{P \models \square(X, Y), P \models \square(F(X))}.$$

If a principal believes that a formula X is recognizable, then it is believed that any formula of which X is a component is recognizable and that a computationally feasible one-to-one function, F, of X is recognizable.

$$\text{I4}: \frac{P \lhd \{X\}K-, P \ni K+, P \models^{K+} Q, P \models \square(X), P \models \#(X, K+)}{P \models Q \mid\sim X, P \models Q \mid\sim \{X\}K-}.$$

If, for principal P, the following conditions hold: P receives a formula X encrypted under private key (K-), P possesses the corresponding public key (K+), believes the public key belongs to Q, and P believes that the formula X is recognizable that either X or K+ is fresh. Then, P believes that Q once conveyed the message X, and that Q once conveyed the message X encrypted under Q's private key (K-).

$$\text{J2}: \frac{P \models Q \mid\Rightarrow Q \models *, P \models Q \mid\sim (X \sim> C), P \models \#(X)}{P \models Q \models C}$$

If principal P believes that Q is honest and competent and P receives a fresh message X with the extension C, which he believes Q conveyed, then P believes that Q believes C.