2010

# Efficient RFID authentication scheme for supply chain applications

Fei Bi
*University of Wollongong*, uow@bi.edu.au

Yi Mu
*University of Wollongong*, ymu@uow.edu.au

# Efficient RFID authentication scheme for supply chain applications

## Abstract

Radio Frequency Identification (RFID) technology has been widely used in supply chains to track and manage shipments. By tagging shipments with RFID tags, which can be remotely accessed by RFID readers, shipments can be identified and tracked in a supply chain. Security issues in RFID have been major concerns, since passive RFID tags have very weak computational power to support authentication. Sound authentication between tag and reader remains a challenging problem. In this paper, we provide a novel authentication scheme to protect tags from being tracked and identified by unauthorized readers and protect authorized readers against bogus tags. Our scheme can be applied to supply chain security. It also exhibits an additional feature that a supply chain can be dynamically updated.

## Disciplines

Physical Sciences and Mathematics

## Publication Details

Bi, F. & Mu, Y. (2010). Efficient RFID authentication scheme for supply chain applications. IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing, EUC 2010 (pp. 583-588). USA: IEEE Computer Society.

2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing

# Efficient RFID Authentication Scheme for Supply Chain Applications

Fei Bi and Yi Mu
Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Wollongong
NSW 2522, Australia
Email: {fb494,ymu}@uow.edu.au

***Abstract*—Radio Frequency Identification (RFID) technology has been widely used in supply chains to track and manage shipments. By tagging shipments with RFID tags, which can be remotely accessed by RFID readers, shipments can be identified and tracked in a supply chain. Security issues in RFID have been major concerns, since passive RFID tags have very weak computational power to support authentication. Sound authentication between tag and reader remains a challenging problem. In this paper, we provide a novel authentication scheme to protect tags from being tracked and identified by unauthorized readers and protect authorized readers against bogus tags. Our scheme can be applied to supply chain security. It also exhibits an additional feature that a supply chain can be dynamically updated.**

***Index Terms*—RFID Security, Authentication, Supply Chain.**

## I. Introduction

Radio Frequency Identification (RFID) is an automatic identification method based on reading and writing data remotely at certain frequency between devices. A typical RFID system consists of an RFID tag, an RFID reader and a back-end server. Most commonly used RFID tags are passive tags, which can only be powered by readers' interrogation signals. RFID readers can remotely retrieve data in RFID tags and submit data to the online back-end server. The useful information about RFID technology can be found in [1].

RFID tags are seemed as the next generation of bar-codes. Each product can be put into an RFID tag with an universally unique tag ID. This can cut off customers' waiting time, improve efficiency, and prevent shoplifting [2]. One of most important applications of RFID technology is supply chain management. By attaching passive RFID tags to shipments [3], RFID systems can improve supply chain visibility, which is useful to track and manage shipments in a supply chain. However, there are many security issues in current RFID supply chain systems. Mutual authentication is needed when shipments reach readers in the supply chain. Tags should be protected from being tracked by unauthorized parties. Security threats like eavesdropping, replay attack and malicious probing are real [4].

There are many existing approaches (e.g., [5], [6], [7], [8], [9], [10], [11]) aiming to solve the authentication problems, based on lightweight authentication. It is generally assumed that a passive RFID tag can only handle a lightweight hashing algorithm. Based on this assumption, many useful lightweight authentication protocols have been proposed. Unfortunately, those protocols cannot be applied to supply chains due to the requirements of dynamical node changes and routing authentication in a supply chain. In this paper, we provide a sound solution to supply chain security. Our scheme provides the mutual authentication, routing authentication, and dynamically node changes. Our scheme adopts lightweight cryptography, which only requires hashing for RFID tags. We assume that the readers have more computational power and therefore are able to handle public-key-based computations.

The remainder of the paper is organized as follows. We present the model of our RFID supply chain in section II, our solution in section III, security analysis in section IV, and the conclusion in section V.

## II. Model of the RFID Supply Chain

A supply chain consists of three parts: shipments, nodes equipped with readers and a back-end server. For a specific shipment, an item departs from the initial node and ends at the final node. All other nodes are called intermediate nodes.

In the initial node, each shipment is attached with an RFID tag which contains a unique identity of the shipment, some parameters used for authentication and a routing table. The routing table includes all the intermediate nodes in the route and some information about those intermediate nodes, which will be used for the tag-reader mutual authentication.

When a shipment arrives at an intermediate node, the RFID tag will require the reader in this node to provide its authentication information. During the reader authentication process, the tag checks whether this reader is legitimate according to the routing table by the proposed authentication protocol. If the reader is indeed the next reader in the routing table, then the authentication of the reader is successful. After the authentication of the reader, the reader will verify the preceding node of the shipment. If there is an online back-end server connected, the current reader will submit its identity and the preceding position of the shipment to seek further authentication from the server. This authentication process is done by the reader using public-key algorithms. Namely, when the goods is about to leave the current node, the reader will

978-0-7695-4322-2/10 $26.00 © 2010 IEEE
DOI 10.1109/EUC.2010.94

insert its digital signature into the tag. This signature is used to authenticate the preceding node by the next node.

Our protocol allows dynamical routing changes. It is quite common that the pre-defined route is not suitable due to an unseen situation at a node. In this case, an alternative node has to be selected. We provide a solution for this case.

Malicious RFID readers can remotely probe tags trying to get some valuable information. In this situation, authentication is needed. In case of supply chains, there are two kinds of authentication choices that require to authenticate the route. The first one does not require a routing table stored in a tag [7]. In this case, the authentication is usually done by checking the hash value and the pre-defined value shared among tags and all readers. Once the hash values match, the authentication is regarded to be successful. However, this kind of authentication is very weak, because tags can only know that the corresponding reader is within the same system without knowing the sequence. Moreover, once the sole shared value is compromised, the system is broken [12]. We adopt the second choice which utilizes a routing table stored in a tag. We also adopt a novel lightweight cryptographic approach that assures much stronger mutual authentication.

## III. Proposed Solution

### A. Initialization

All notations that will be used in this paper are presented in Table I.

TABLE I
Notations of our protocol.

| Notation | Meaning |
|---|---|
| $x$ | a secret shared among all readers and back-end server. |
| $i$ | index to the routing table. |
| $G$ | preceding position index. |
| $k_\alpha$ | reader $\alpha$ private key. |
| $k_l$ | preceding reader's private key. |
| $H(m)$ | hash value of $m$. |
| $S$ | preceding reader's signature, $Sig(H(m))$, where $m$ is not included. We use Schnorr signature scheme. |
| $tid_j$ | ID for tag $j$. |
| $ID_l$ | ID of the reader for the preceding position. |
| $ID_\alpha$ | ID of reader $\alpha$. |
| $v_j$ | special value used for add-new-stop modes, where $j$ is an index associated with tag $j$. |
| $S_\alpha$ | the symmetric key shared between the server and reader $\alpha$. |
| $r$ | a random number. |
| $(m)_{S_\alpha}$ | $m$ encrypted with symmetric key $S_\alpha$. |
| $z \leftarrow y$ | $y$ is assigned to $z$. |

We require that all nodes in the system share a security parameter $x$. Every reader and the back-end server share a secret symmetric key. All readers in the system have a public/private key pair.

In the initial node, a tag $j$ is initialized with the following values: a routing table, an index $i$, a preceding position index $G$, a preceding reader's signature $S$, and a special value $v_j$. A routing table indicates the sequence of nodes this tag should reach. Each column in the table has three values: a number, a reader ID, and tag ID encrypted by the symmetric key shared between the back-end server and the reader. The routing table should be provided by the back-end server.

Table II is an example of a routing table. The Table II indicates that the route is $C, B, A, E, ....$

TABLE II
Example of a routing table.

| 1 | 2 | 3 | 4 | ... |
|---|---|---|---|---|
| $ID_C$ | $ID_B$ | $ID_A$ | $ID_E$ | ... |
| $(tid_j)_{S_C}$ | $(tid_j)_{S_B}$ | $(tid_j)_{S_A}$ | $(tid_j)_{S_E}$ | ... |

An index $i$ is used for a tag to find which column in the routing table contains the next reader's information. $i$ is initialized to 1. If $i$ is equal to 3, then the next reader is reader $A$ according to the routing table. After a tag passes a reader, the value of $i$ should be updated to $i+1$. A preceding position index $G$ can show which reader is a tag's last passed reader. Set $G = g^{xk_\alpha} \bmod p$, where $k_\alpha$ is the private key of reader $\alpha$ and $g^{k_\alpha} \bmod p$ is the corresponding public key, $g$ is a generator of $Z_p^*$, $p \in Z_p$ is a safe prime. $g^{k_\alpha} \bmod p$ is also used as the Schnorr signature verification key. $S = Sig_{in}(H(tid_j, ID_\alpha))$ denotes the Schnorr signature from the node $\alpha$. The form of the signature is predefined. We will describe it in the protocol. For simplicity, we will omit the modulo $p$ in the rest of presentation.

We define a special value $v_j$ for a tag $j$ to cope with route changes. The back-end server has an $ID$-$v$ table to keep a copy of $v$ of every tag. The update of the value of $v_j$ for a tag $j$ and the back-end server should be synchronized.

TABLE III
Example of a public key table.

| $g^{k_A}$ | $g^{k_B}$ | $g^{k_C}$ | $g^{k_D}$ | $g^{k_E}$ | ... |
|---|---|---|---|---|---|
| $ID_A$ | $ID_B$ | $ID_C$ | $ID_D$ | $ID_E$ | ... |

TABLE IV
Example of a route-change table.

| $v_b$ | $v_e$ | $v_m$ | ... |
|---|---|---|---|
| $tid_b$ | $tid_e$ | $tid_m$ | ... |

Every reader has a public key table which can be used to find the $ID$ of a reader corresponding to a public key. This table has the $ID$ and public key of every reader in the system.

An example of public key table is in Table III.

The back-end server keeps an $ID$-$v$ table which stores all tag's IDs and the value of $v$ of a tag. An example of $ID$-$v$ table is given in Table V.

TABLE V
Example of an $ID$-$v$ table.

| $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | ... |
|---|---|---|---|---|---|
| $tid_a$ | $tid_b$ | $tid_c$ | $tid_d$ | $tid_e$ | ... |

Every reader keeps a route-change table which stores information of route changes for some tags. Each column of the table is obtained from the server's route-change table.

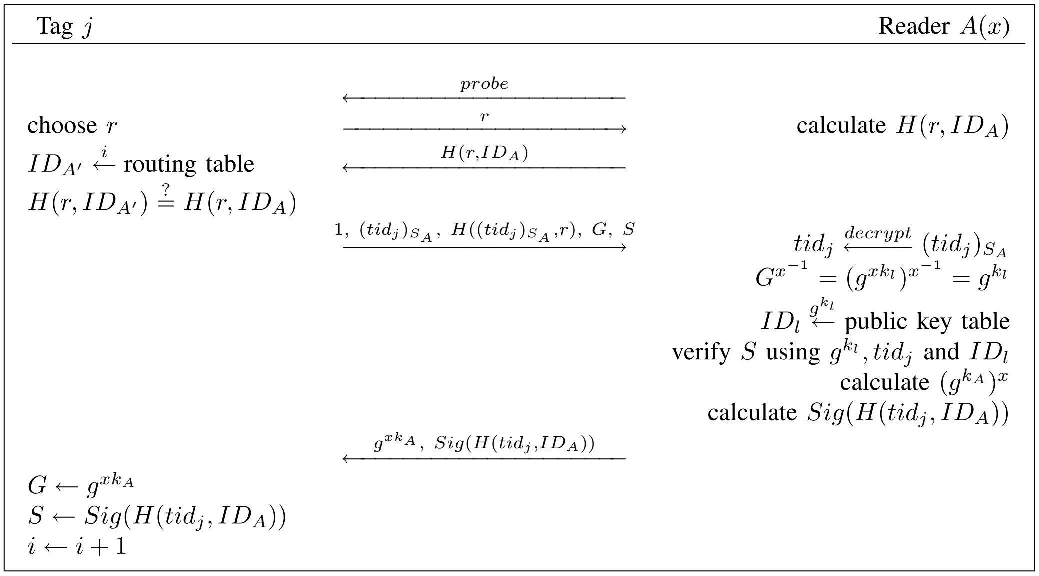
Tag $j$
Reader $A(x)$
$\xleftarrow{probe}$
choose $r$
$\xrightarrow{r}$
calculate $H(r, ID_A)$
$ID_{A'} \overset{i}{\leftarrow}$ routing table
$\xleftarrow{H(r,ID_A)}$
$H(r, ID_{A'}) \overset{?}{=} H(r, ID_A)$
$\xrightarrow{1,\ (tid_j)_{S_A},\ H((tid_j)_{S_A}, r),\ G,\ S}$
$tid_j \xleftarrow{decrypt} (tid_j)_{S_A}$
$G^{x^{-1}} = (g^{xk_l})^{x^{-1}} = g^{k_l}$
$ID_l \overset{g^{k_l}}{\leftarrow}$ public key table
verify $S$ using $g^{k_l}, tid_j$ and $ID_l$
calculate $(g^{k_A})^x$
calculate $Sig(H(tid_j, ID_A))$
$\xleftarrow{g^{xk_A},\ Sig(H(tid_j, ID_A))}$
$G \leftarrow g^{xk_A}$
$S \leftarrow Sig(H(tid_j, ID_A))$
$i \leftarrow i + 1$

Fig. 1. Normal Mode

Different readers have different route-change tables. Table IV shows an example of a route-change table. As we can see, the route-change table in a reader is actually part of the $ID$-$v$ table in the server.

### B. Protocols

*Normal Mode*. In this mode, a shipment process is performed normally. A tag firstly authenticates the corresponding reader and then sends the information about the tag itself to allow the reader to authenticate it. The protocol is given in Figure 1 and described in the following steps:

1: Reader $A$ probes the tag remotely.
2: The tag chooses a random value $r$.
3: The tag sends the random value $r$ to reader $A$.
4: Reader $A$ calculates $H(r, ID_A)$. $ID_A$ is reader $A$'s ID.
5: Reader $A$ sends the $H(r, ID_A)$ to the tag.
6: The tag finds the corresponding value $ID_{A'}$ to the index $i$ in the routing table, calculates $H(r, ID_{A'})$, and compares this value with the value of $H(r, ID_A)$. If the two values agree, reader $A$ will be the legitimate reader of this tag. Once the tag successfully authenticates reader $A$, step 7 is initialized. If two values are not equal, the authentication fails.
7: The tag sends $(1, (tid_j)_{S_A}, H((tid_j)_{S_A}, r), G, S)$ to reader $A$. $(tid_j)_{S_A}$ is the value in the third row in the routing table corresponding to $i$. $1$ indicates that the authentication is successful. $G$ is equal to $g^{xk_l}$ where $k_l$ is the preceding reader's private key . $S$ is the preceding reader's signature and is equal to $Sig(H(tid_j, ID_l))$, where $ID_l$ is the preceding reader's ID. It proceeds to step 8.
8: Reader $A$ decrypts $(tid_j)_{S_A}$ to get the $tid_j$, calculates $G^{x^{-1}} = (g^{xk_l})^{x^{-1}} = g^{k_l}$, and uses $g^{k_l}$ to get $ID_l$ by checking the public key table. As reader $A$ has $ID_l$ and $tid_j$ now, it can verify the signature. By verifying the signature of the preceding reader, reader $A$ obtains the previous position of the tag. Reader A logs $tid_j$ and $ID_l$ or submits these two values to online server. Reader A computes $(g^{k_A})^x = g^{xk_A}$ and its signature $Sig(H(tid_j, ID_A))$, where $k_A$ is the private key of reader $A$.
9: Reader $A$ sends $g^{xk_A}$ and $Sig(H(tid_j, ID_A))$ to the tag.
10: The tag updates $G \leftarrow g^{xk_A}$ and $S \leftarrow Sig(H(tid_j, ID_A))$ and increments $i \leftarrow i + 1$.

In this protocol, we use the routing table for the authentication of readers by checking whether reader $A$ is the next reader in the routing table. If $H(r, ID_A)$ equals to $H(r, ID_{A'})$, the tag authenticates reader $A$. We utilized digital signatures for the verification of the route. Notice that position's IDs are not sent in clear and is in the form of $G^{x^{-1}} = (g^{xk_l})^{x^{-1}} = g^{k_l}$ that is also the public key used to verify the signature.

The structure of the signature is pre-defined. Reader $A$ learns $tid_j$ by decrypting $(tid_j)_{S_A}$ and $ID_l$ by checking $g^{k_l}$ in the public key table. So there is no need to send the signed plaintext of the signature. After authenticating the tag, the reader will calculate a new preceding position index $g^{xk_A}$ and its signature. The tag updates $G$ and $S$ which will be forwarded to the next node by the tag. The tag increments the counter $i$ to $i + 1$ pointing to the next node in the routing table.

*Add-New-Node Mode*. We divide this mode into cases: (1) the added new node is in the same supply chain, which means it has the parameter $x$ but is not in the routing table; (2) the new node is not in the routing table and does not belong to the same supply chain system (without $x$).

In case (1), the tag arrives at reader $B$, which is in the system but not in the routing table of the target tag. So the tag fails to authenticate reader $B$ in the normal mode and requires

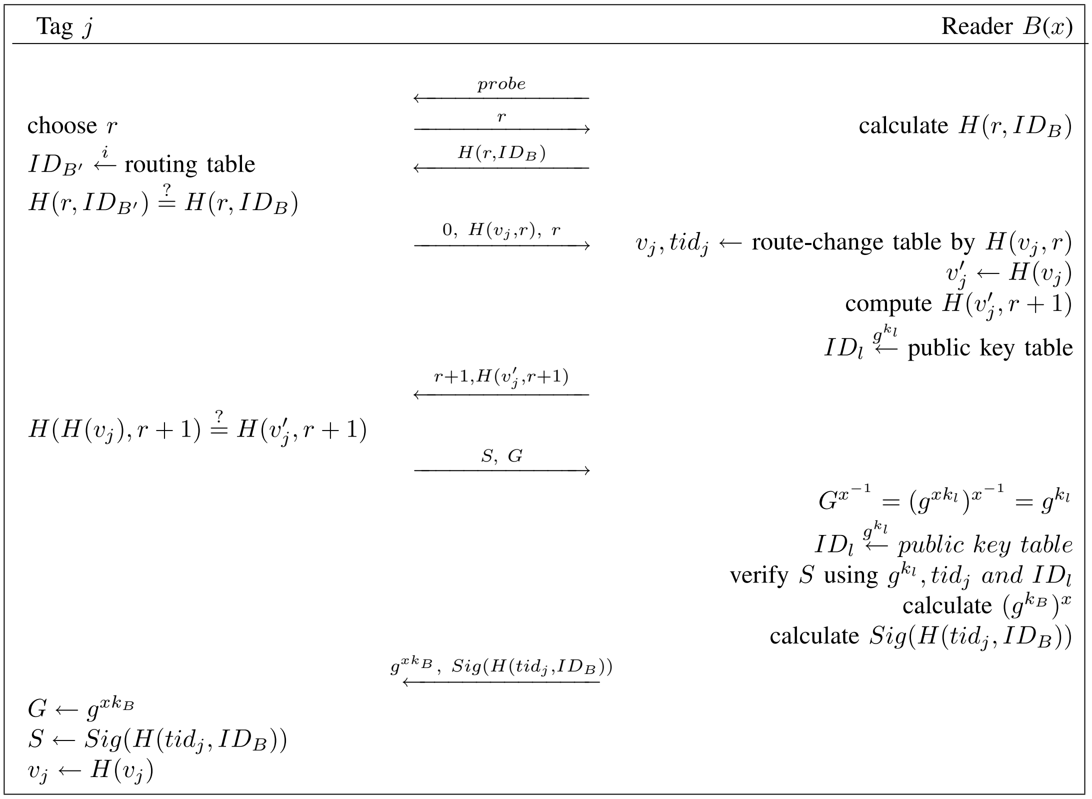
Tag $j$
Reader $B(x)$
$probe$
choose $r$
$r$
calculate $H(r, ID_B)$
$ID_{B'} \overset{i}{\leftarrow}$ routing table
$H(r,ID_B)$
$H(r, ID_{B'}) \overset{?}{=} H(r, ID_B)$
$0,\ H(v_j,r),\ r$
$v_j, tid_j \leftarrow$ route-change table by $H(v_j, r)$
$v'_j \leftarrow H(v_j)$
compute $H(v'_j, r+1)$
$ID_l \overset{g^{k_l}}{\leftarrow}$ public key table
$r+1, H(v'_j, r+1)$
$H(H(v_j), r+1) \overset{?}{=} H(v'_j, r+1)$
$S,\ G$
$G^{x^{-1}} = (g^{xk_l})^{x^{-1}} = g^{k_l}$
$ID_l \overset{g^{k_l}}{\leftarrow} public\ key\ table$
verify $S$ using $g^{k_l}, tid_j\ and\ ID_l$
calculate $(g^{k_B})^x$
calculate $Sig(H(tid_j, ID_B))$
$g^{xk_B},\ Sig(H(tid_j, ID_B))$
$G \leftarrow g^{xk_B}$
$S \leftarrow Sig(H(tid_j, ID_B))$
$v_j \leftarrow H(v_j)$

Fig. 2. Protocol of Add-New-Node Case (1).

reader $B$ to provide the evidence that $B$ has the route-change token for the tag. The protocol is given in Figure 2.

The first six steps are the exactly same as the normal mode, therefore omitted.

7: The tag sends 0, $H(v_j, r)$, $r$ to the reader $B$. 0 indicates that additional information is required for authentication. The tag sends $H(v_j, r)$ to $B$ and is expecting $H(H(v_j), r+1)$ from $B$.

8: Reader $B$ uses every $v_\beta$ where $\beta$ is an index of a tag ID in the route-change table to calculate the value of $H(v_\beta, r)$ and compares it to the value of $H(v_j, r)$ until it finds a $v_\beta$ that satisfies $H(v_\beta, r) = H(v_j, r)$. The tag ID corresponding with $v_\beta$ in the route-change table is set as $tid_j$. Then, $B$ calculates $v'_j = H(v_\beta)$ and $H(v'_j, r+1)$. Note that the route-change table can be updated according to the back-end server's information.

9: Reader $B$ returns $r+1$ and $H(v'_j, r+1)$.

10: The tag computes $H(H(v_j), r+1)$ and compares the value with $H(v'_j, r+1)$. If they are equal, the authentication is successful. This means that reader $B$ has received the route-change token of this tag from the server.

11: The tag sends $G$ and $S$ to reader $B$.

12: Reader $B$ calculates $G^{x^{-1}} = (g^{xk_l})^{x^{-1}} = g^{k_l}$ and can use $g^{k_l}$ to get $ID_l$ by checking the public key table. As reader $B$ has $ID_l$ and $tid_j$ now, it can verify the signature. By verifying the signature of the preceding reader, reader $B$ can find out the previous position. Reader $B$ can log $tid_j$ and $ID_l$ or submit these two values to the online server. Reader $B$ computes $(g^{k_B})^x = g^{xk_B}$ and $Sig(H(tid_j, ID_B))$, where $k_B$ is the private key of reader $B$.

13: Reader $B$ sends $g^{xk_B}$ and $Sig(H(tid_j, ID_B))$ to the tag.

14: The tag updates $G \leftarrow g^{xk_B}$, $S \leftarrow Sig(H(tid_j, ID_B))$, $v_j \leftarrow H(v_j)$.

As we can see, the only difference of this protocol from the normal mode protocol is the authentication process of reader $B$. If the authentication in the step 6 is not successful, the tag needs to know whether reader $B$ has the route-change token from the server by sending the hashed value of challenge $v_j$ and $r+1$. If reader $B$ has already got the route-change token $(v_j, tid_j)$ from the server and added these two values into the route-change table, reader $B$ will be able to find the value of $v_j$ by searching the route-change table and return $H(H(v_j), r+1)$.

For case (2), a back-end server is online to support reader $C$ in the new node. The protocol is in Fig 3.

Because the new reader $C$ in this mode is originally not in this supply chain, reader $C$ doesn't have the parameter $x$ and route-change token from the server. It has to let the online server to generate the $H(v'_j, r+1)$ which will be passed to the tag for the authentication. So at step 9, reader $C$ passes $H(v_j, r)$ and $r$ to the server with a signature.

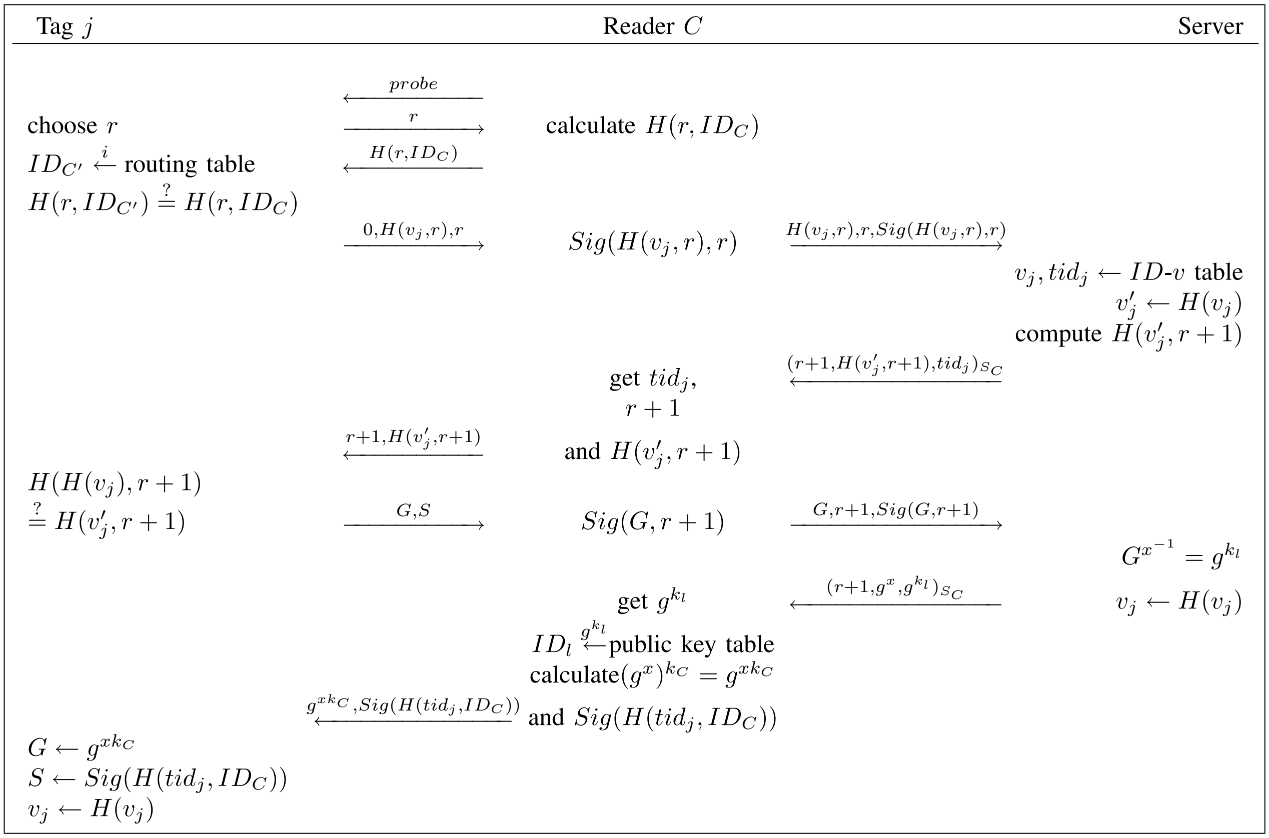
Tag $j$
Reader $C$
Server
$probe$
choose $r$
$r$
calculate $H(r, ID_C)$
$ID_{C'} \overset{i}{\leftarrow}$ routing table
$H(r,ID_C)$
$H(r, ID_{C'}) \overset{?}{=} H(r, ID_C)$
$0,H(v_j,r),r$
$Sig(H(v_j, r), r)$
$H(v_j,r),r,Sig(H(v_j,r),r)$
$v_j, tid_j \leftarrow ID\text{-}v$ table
$v'_j \leftarrow H(v_j)$
compute $H(v'_j, r+1)$
$(r+1,H(v'_j,r+1),tid_j)_{S_C}$
get $tid_j$,
$r+1$
$r+1,H(v'_j,r+1)$
and $H(v'_j, r+1)$
$H(H(v_j), r+1)$
$\overset{?}{=} H(v'_j, r+1)$
$G,S$
$Sig(G, r+1)$
$G,r+1,Sig(G,r+1)$
$G^{x^{-1}} = g^{k_l}$
get $g^{k_l}$
$(r+1,g^x,g^{k_l})_{S_C}$
$v_j \leftarrow H(v_j)$
$ID_l \overset{g^{k_l}}{\leftarrow}$public key table
calculate$(g^x)^{k_C} = g^{xk_C}$
$g^{xk_C},Sig(H(tid_j,ID_C))$
and $Sig(H(tid_j, ID_C))$
$G \leftarrow g^{xk_C}$
$S \leftarrow Sig(H(tid_j, ID_C))$
$v_j \leftarrow H(v_j)$

Fig. 3. Protocol of Add-New-Node Case (2).

The server searches the $ID$-$v$ table to find the value of $v_j$ and $tid_j$, then calculates $v'_j = H(v_j)$. The server creates an authentication token $(r+1, H(v'_j, r+1))$ and sends it and $tid_j$ to reader $C$ in a public-key encrypted form. So reader $C$ gets the $tid_j$ and passes the authentication token to the tag. After authenticating reader $C$, the tag returns $G$ and $S$ to reader $C$. Without $x$, reader $C$ is unable to get the public key of the preceding reader by computing $G^{x^{-1}}$. So reader $C$ forwards $G$ to the online server that will in turn compute $g^{k_l}$ for reader $C$. The server returns $g^{k_l}$ and $g^x$ to reader $C$. With $g^{k_l}$, reader $C$ can find $ID_l$ by searching the public key table. Reader $C$ can use $g^x$ to compute $(g^x)^{k_C} = g^{xk_C}$ without finding out the value of $x$. Reader $C$ also calculates the signature $Sig(H(tid_j, ID_C))$ and sends $g^{xk_C}$ and $Sig(H(tid_j, ID_C))$ to the tag. The tag finally updates $G \leftarrow g^{xk_C}$ and $S \leftarrow Sig(H(tid_j, ID_C))$.

The main difference between these two cases is that the online server does the calculation of $(r+1, H(v'_j, r+1))$ and $g^{k_l}$ instead of the reader $C$. This is because that reader $C$ does not have $x$ and the route-change token.

## IV. Security Requirements and Analysis

In this section, we discuss security requirements for our RFID supply chain system we proposed and explain how the system achieves the requirements.

### A. Resistance to Replay Attack

Resistance to Replay Attack means that if the adversary $\mathcal{A}$ can eavesdrop communication between tags and readers, it cannot replay any information transferred in the communication. In our protocol, we use a random number $r$ as nonce to protect the system from the replay attack. Different sessions have different values of $r$, so a message used in one session cannot be reused in another session.

### B. Forward Untraceability

Forward Untraceability means that given all communication flows between a tag and a reader at time $t$, the adversary $\mathcal{A}$ cannot use these information to trace the tag at a time $t' > t$. Our protocol only considers the authentication of a tag and the verification of the preceding location of the tag, with no information related to forward locations of the tag. Moreover, the identity of a tag is not revealed in the protocol.

### C. Backward Untraceability

Backward Untraceability means that given all communication flows between a tag and a reader at time $t$, the adversary $\mathcal{A}$ cannot use these information to trace the tag at a time $t' < t$. Our protocol requires the preceding location verification by tags that sends the verification information to readers. Every tag stores the information $G$ and $S$ about its preceding location. $G$ is protected by the hardness of the discrete logarithm and only authorized readers, which have the secret key $x$ can compute the preceding location's public key by applying $G^{x^{-1}}$. The preceding reader's public key is used to verify the signature $S$. So unauthorized readers will not be able to track a tag's last location.

### D. Resistance to Reader Impersonation

Resistance to Reader Impersonation means that the adversary $\mathcal{A}$ cannot impersonate a reader so that it can extract useful information from a legitimate tag. Reader impersonation is the primary attack in this system. When an unauthorized reader probes a tag, it should not be authenticated by the tag. It is easy to find that our protocols are secure against this attack.

## V. Conclusion

We proposed an RFID authentication scheme for supply chain applications. Our solution provides strong mutual authentication between tags and readers. We allow the authentication to be asymmetric, in the sense that tags are only required to carry out very basic computation based on hashing and readers can implement much more complex computations such as encryption and digital signatures. Apart from strong authentication, we allow the supply chain route to be authenticated and updated according to the need. In other words, we allow a supply chain to be dynamically updated by adding new nodes.

## References

[1] A. Juels, "RFID security and privacy: A research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24(2), pp. 381–394, February 2006.

[2] "Wal-mart details RFID requirement," November 6 2003, http://www.rfidjournal.com/article/articleview/642/1/1.

[3] A. Niemeyer, M. H. Pak, and S. E. Ramaswamy, "Smart tags for your supply chain," *The McKinsey Quarterly*, vol. 4, pp. 6–8, 2003.

[4] S. A. Weis, S. E. Sarma, and D. W. Engels, "Radio-frequency identification: Security risks and challenges," *Cryptobytes*, vol. 6, no. 1, pp. 6–8, 2003.

[5] A. Juels, "Minimalist cryptography for low-cost RFID tags," in *Security of Communication Networks(SCN) 2004, LNCS 3352*. Springer, 2004, pp. 149–164.

[6] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proceedings of SECURECOMM'05*. IEEE Press, September 2005, pp. 59–66.

[7] S. A. Weis, S. E.Sarma, R. L.Rivest, and D. W.Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in Pervasive Computing 2003, LNCS 2802*. Springer, 2004, pp. 201–212.

[8] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic approach to privacy-friendly tags," in *RFID Privacy workshop*. MIT, USA, 2003.

[9] I. Vajda and L. Buttyan, "Lightweight authentication protocols for low-cost RFID tags," *Second Workshop on Security in Ubiquitous Computing*, 2003.

[10] D. Henrici and P. Müller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers," in *Pervasive Computing and Communications Security*. IEEE Computer Society, 2004, pp. 149–153.

[11] G. Avoine and P. Oechslin, "RFID traceability: A multilayer problem," in *Financial Cryptography, LNCS 3570*. Springer, 2005, pp. 125–140.

[12] D. Molnar and D. Wagner, "Privacy and security in library RFID: issues, practices, and architectures," in *Proceedings of the 11th ACM conference on Computer and communications security*. IEEE Press, 2003, pp. 210–219.