12-2009

On the Untraceability of Anonymous RFID Authentication Protocol with Constant Key-Lookup

Bing LIANG
Singapore Management University

Yingjiu LI
Singapore Management University, yjli@smu.edu.sg

Tieyan LI
Institute for Infocomm Research, Singapore

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

DOI: https://doi.org/10.1007/978-3-642-10772-6_7

On the Untraceability of Anonymous RFID Authentication Protocol with Constant Key-Lookup

Bing Liang[1], Yingjiu Li[1], Changshe Ma[1,3], Tieyan Li[2], and Robert Deng[1]

[1] School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore, 178902
[2] Institute for Infocomm Research, A*STAR Singapore
[3] School of Computer, South China normal University, Guangzhou, China, 510631
liangb02@gmail.com, {yjli,changshema}@smu.edu.sg,
litieyan@i2r.a-star.edu.sg, robertdeng@smu.edu.sg

Abstract. In ASIACCS'08, Burmester, Medeiros and Motta proposed an anonymous RFID authentication protocol (BMM protocol [2]) that preserves the security and privacy properties, and achieves better scalability compared with other contemporary approaches. We analyze BMM protocol and find that some of security properties (especial untraceability) are not fulfilled as originally claimed. We consider a subtle attack, in which an adversary can manipulate the messages transmitted between a tag and a reader for several continuous protocol runs, and can successfully trace the tag after these interactions. Our attack works under a weak adversary model, in which an adversary can eavesdrop, intercept and replay the protocol messages, while stronger assumptions such as physically compromising of the secret on a tag, are not necessary. Based on our attack, more advanced attacking strategy can be designed on cracking a whole RFID-enabled supply chain if BMM protocol is implemented. To counteract such flaw, we improve the BMM protocol so that it maintains all the security and efficiency properties as claimed in [2].

Keywords: RFID, Anonymous, Authentication, Privacy.

1 Introduction

Radio Frequency Identification (RFID) technology has been applied in a range of industries such as libraries [12], automatic payment [15], animal tracking [15], supply chains [8] and E-passport [16]. An RFID system generally incorporates three components: tag, reader and back-end database. Typically, a reader can interrogate with a tag and send the tag's information to database for verification. There are two main kinds of tags: active tags which are battery-powered [14] and passive tags without battery, which are powered by the electromagnetic field established by the reader's antenna. As the cost of active tags is much higher than the passive ones, only passive tags are considered to be suitable for large-scale applications such as supply chain management.

Privacy and scalability are two important perspectives in RFID protocols. On the aspect of privacy, if the tag is not managed carefully, the privacy of its carrier will be inferred by a malicious party. In some cases, the tags can release the information about an individual's medication record, banknote's serial number, culture preference,

location information, and etc.. In other cases, a company's sensitive information such as product price, and supply chain routine can be obtained by the company's opponent, which may lead to the financial loss of the company. In all, privacy is one of the most essential security consideration in RFID system.

Besides the privacy concern, scalability is another important issue in designing an RFID authentication protocol. RFID users usually have a high requirement of the proceeding time. In the survey of [4], more than half the people in the investigation consider efficiency of the RFID authentication process quite important, far more important than those people who consider security important. In [2], Burmester, Medeiros and Motta (BMM) proposed an RFID authentication protocol with constant key-lookup to balance the privacy requirement and scalability. To the best of our knowledge, this protocol is one of the most scalable solutions that preserve privacy as claimed (please see Section 7 for more details about related works). In this paper, we identify the shortcoming in BMM protocol [2] and propose an improved protocol accordingly. We argue that the improved protocol provides stronger privacy than the BMM protocol, while the performance of the improved protocol is the same as the BMM protocol. Our contributions in this paper are summarized below:

1. We analyze the BMM-protocol and find a subtle flaw, by which we can break the privacy property, namely untraceability. Exploiting this flaw, we design an easy-to-launch attack under a weak adversary model. Under our attack, an adversary can easily trace a tag in a supply chain party. Thus, one by one, we can trace such a tag in a whole supply chain if the BMM protocol is implemented.
2. To improve the protocol, we propose an anonymous RFID authentication protocol that can fulfill all privacy claims of [2], including defense against eavesdropping attack, spoofing attack, replay attack, de-synchronization attack, tracing attack and compromising attack.

The organization of this paper is as follows: In Section 2, we introduce the notation that will be used in this paper. In Section 3, we review the BMM protocol. In Section 4, we elaborate on our attack. In Section 5, an example on cracking the whole supply chain is presented. Further, in Section 6, we propose the improved protocol and analyze its security properties. In Section 7, we introduce the related works on RFID authentication. In Section 8, we conclude the paper.

2 Notation

If $A(\cdot,\cdot,...)$ is a randomized algorithm, then $y \leftarrow A(x_1, x_2, ...; cn)$ means that y is assigned the unique output of the algorithm A on inputs $x_1, x_2, ...$ and coins cn. Let g be a pseudorandom function (PRF) [7]. If S is a set, then $s \in_R S$ indicates that s is chosen uniformly at random from S. If $x_1, x_2, ...$ are strings, then $x_1\|x_2\|\cdots$ denotes the concatenation of them. If x is a string, then $|x|$ denotes its bit length in binary code. Let ε denote the empty string. If S is a set, then $|S|$ denotes its cardinality (i.e. the number of elements of S). If ctr is a counter which starts from n_1 and ends with n_ℓ, then $ctr(j)$ denotes its jth value, i.e. $ctr(j) = n_j$, where $1 \leqslant j \leqslant \ell$. Let IV be an initial vector for the PRF g.

3 The BMM Protocol

In this section, we review the BMM protocol, (which is shown in Figure 1).

In the RFID system constructed by BMM protocol, there is a set-up procedure which initializes the reader and every tag. Then, they will engage in a protocol to identify the tag. The whole RFID system is described as follows.

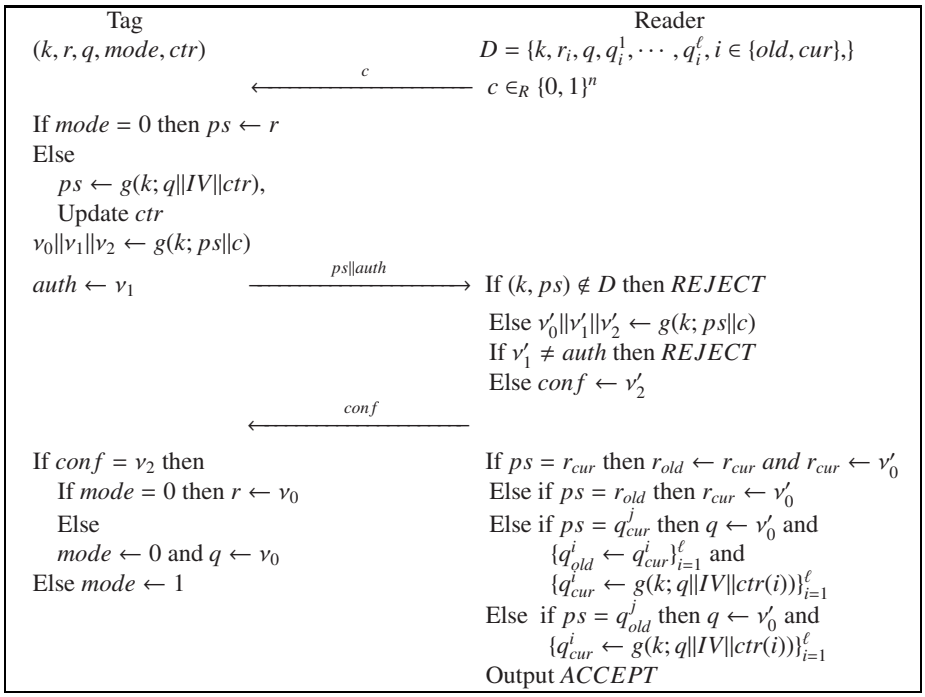

Fig. 1. BMM Protocol

Setup: When creating a new tag T, the system generates a secret key k, a pseudonym seed q, a one-time pseudonym r, a counter $ctr = 1$, and a flag $mode = 0$. Then it sets up the initial state information of the tag T as the tuple $(k, q, r, ctr, mode)$. The system also associates the tag T with its identity ID_T in the reader's database by initiating a tuple $(r_{old}, r_{cur}, q_{old}^1, \cdots q_{old}^\ell, q_{cur}^1, \cdots q_{cur}^\ell, k, q, ID_T)$, where $r_{old} = r_{cur} = r$ and $q_i^j = g(k; ||q||IV||ctr(j)$, for $i = \{old, cur\}$, and $j = 1, \cdots \ell$.

The BMM Protocol: It runs in three rounds:

Round 1. First, the reader starts the protocol by sending a challenge c to the tag. Upon receiving c, the tag first checks its $mode$ state: if $mode = 0$, it sets the pseudonym $ps = r$; otherwise, it computes $ps = g(k; q||IV||ctr)$ and updates the counter $ctr = ctr + 1$. Then, the tag calculates $v_0||v_1||v_2 = g(k; ps||c)$. Here, v_0 is used to replace the pseudonym r; $auth = v_1$ is used to authenticate itself to the reader, and v_2 is used to authenticate the reader.

Round 2. The tag sends the message $ps||auth$ to the reader. Upon receiving $ps||auth$, the reader requests to its back-end database to look up the tuple $(r_{old}, r_{cur}, q^1_{old}, \cdots q^\ell_{old}, q^1_{cur}, \cdots q^\ell_{cur}, k, q_0, ID_T)$ such that $r_i = ps$ or $q^j_i = ps$, where $i = \{old, cur\}$ and $j = 1, \cdots \ell$, through using ps as an index. If the tag is de-synchronized within ℓ times, we can find the tuple in constant time by $2\ell + 2$ indexes. If the tuple is found, the reader calculates $\nu'_0||\nu'_1||\nu'_2 \leftarrow g(k; ps||c)$ and accepts the tag if $auth = \nu'_1$. Otherwise, the tag is rejected. If a tag is accepted, the reader prepares a confirmation message $conf \leftarrow \nu'_2$.

Round 3. The reader sends the confirmation message $conf$ to the tag. The tag authenticates the reader by checking whether $conf = \nu_2$. If the reader is successfully authenticated, the tag then updates its pseudonym: if $mode = 0$, it updates the pseudonym $r = \nu_0$; if $mode = 1$, it updates pseudonym seed $q = \nu_0$ and keep the pseudonym r unchanged. If the reader is not authenticated, the tag sets $mode = 1$ and does nothing else. On the reader side, it updates the tuple $(r_{old}, r_{cur}, q^1_{old}, \cdots, q^\ell_{old}, q^1_{cur}, \cdots q^\ell_{cur}, k, q_0, ID_T)$ associated with the tag as follows. If $ps = r_{cur}$, it updates $r_{old} = r_{cur}$ and $r_{cur} = \nu'_0$. If $ps = r_{old}$, it only updates $r_{cur} = \nu'_0$. If $ps = q^j_{old}$ for some j between 1 and ℓ, it updates $q = \nu'_0$ and $q^j_{cur} = g(k; ||q||IV||ctr(j)$ for $j = 1, \cdots \ell$. If $ps = q^j_{cur}$ for some j between 1 and ℓ, it updates $q = \nu'_0$, $q^j_{old} = q^j_{cur}$ and $q^j_{cur} = g(k; ||q||IV||ctr(j)$ for $j = 1, \cdots \ell$.

Burmester, Medeiros and Motta claimed that it can *"support anonymity with constant key-lookup cost; however, it suffers from entrapment attacks"* [2]. To preserve the privacy of a queried tag, an adversary that eavesdrops over the protocol should not be able to figure out the identifier of the tag with higher likelihood than a pure random guess. The same should also apply to an unauthorized reader that attempts to query the tag. In other words, the protocol should ensure "tag anonymity", in terms of session unlinkability: an adversary should not be able to link together two or more protocol sessions involving the same tag (regardless whether the identity of the tag is known or not) to track the activities of the tag. To achieve this, any two protocol exchanges involving the same tag must appear reasonably random such that the adversary cannot differentiate it with non-negligible probability from two protocol exchanges involving two different tags.

Unfortunately, there exist some flaws in the updating procedures in the design of BMM protocol. The flaws can be subsequently exploited to launch a simple attack to trace a tag in a series of protocol runs.

4 Attacking the BMM Protocol

In this section, we describe a three-run interleave attack and show how to use it to track a tag. Our attack is easy to launch as it requires a weak adversary model as depicted below.

4.1 The Adversary Model

In typical RFID security scenarios, adversaries with different levels of power are modeled to analyze different RFID authentication protocols [10]. We consider adversaries with three levels of power as follows:

– **Level-1 (Passive attack):**
  Able to perform passive eavesdropping and intercept messages over legitimate protocol sessions.
– **Level-2 (Active attack with protocol participation & protocol disruption):**
  Able to communicate with a legitimate tag or reader by following the steps specified under the protocol and to replay, corrupt, block or inject (replace)messages.
– **Level-3 (Active attack with secret compromise):**
  Able to capture a legitimate tag and extract its secrets through physical layer attack and side channel attacks.

It is reasonable to assume that a higher level adversary also possesses the abilities of all levels preceding it, i.e. a level-3 adversary has the abilities of level-1 and level-2 adversaries, as well as the set of additional abilities of physical layer attacks and side channel attacks. As we will be showing in next subsection, our attack requires a relatively weak adversary model (*w.r.t.*, a level-2 adversary), where an adversary has limited ability to communicate with a legitimate tag following protocol steps.

Different kinds of attacks can achieve variable goals. Eavesdropping attacks can track a tag successfully if the tag's responses keep same. Attackers can communicate with trusted readers and trusted tags through spoofing and replay attack. De-synchronization attacks can interrupt regular communications between trusted readers and tags through blocking, modifying and injecting messages. Denial of Service (DoS) attacks mean that a legitimate reader is flooded with useless messages so that it cannot communicate with legitimate tags normally.

4.2 Three-Run Interleave Attack

We first give the intuition behind our attack. We observe that the state information (index) 'r' in the tag always keeps unchanged in the protocol executions when $mode = 1$ and $conf = \nu_2$ (see Figure 1). It means that the tag will reply with the same response in the next interrogation. Our attack follows this observation and uses a 'three-run interleave' technique to push the tag into the state of $mode = 1$ and $conf = \nu_2$.

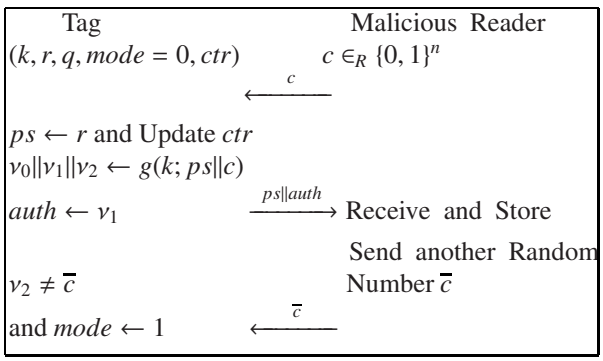

Fig. 2. First Run of The Attack

As mentioned in Section 4.1, we assume a level-2 adversary as the malicious reader, denoted by $\mathcal{R}^M$. We denote a legitimate tag by $\mathcal{T}$ and a trusted reader by $\mathcal{R}^T$. The attack consists of three runs, during which $\mathcal{T}$ is interrogated by $\mathcal{R}^M$ twice and by $\mathcal{R}^T$ once. We present the attack in detail as follows.

1. **First Run: $\mathcal{R}^M$ interrogates $\mathcal{T}$**
   This first run of our attack is illustrated in Figure 2. During the first protocol run, $\mathcal{R}^M$ interrogates $\mathcal{T}$ with an incomplete protocol execution. We assume that $\mathcal{R}^M$ can launch attacks after several legitimate communications between $\mathcal{R}^T$s and $\mathcal{T}$, so we can consider the initial status of $\mathcal{T}$ as $mode = 0$. After sending a challenge c, $\mathcal{R}^M$ receives the reply message $ps||auth = r||\nu_1$ from $\mathcal{T}$. As $\mathcal{R}^M$ does **not** share any secret with $\mathcal{T}$, it cannot compose the correct confirmation message for $\mathcal{T}$. Instead, $\mathcal{R}^M$ sends a random value $\overline{c}$ to $\mathcal{T}$. At the tag's side, $\overline{c}$ cannot be verified against $conf$, so $\mathcal{T}$ changes its status into an attacked state with $mode = 1$. To this end, $\mathcal{R}^M$ stores the reply '$r||\nu_1$' and continues to the next step.

   Note that if $\mathcal{R}^M$ sends queries to a tag continuously, he/she can only obtain the unlinkable information $ps||auth$. Therefore, to get useful information, which can link the same tag by comparing 'r', the adversary intentionally involves a trusted reader $\mathcal{R}^T$ in the second run.
2. **Second Run: $\mathcal{R}^T$ interrogates $\mathcal{T}$**
   The second run of our attack is shown in Figure 3. During the second protocol run, $\mathcal{T}$ is put forward and interrogated by a trusted reader $\mathcal{R}^T$ with a complete protocol execution, while $\mathcal{R}^M$ does nothing. Note that in the first run of our attack, $\mathcal{T}$ toggles its $mode$ in $\mathcal{T}$ to '1'; therefore, after $\mathcal{T}$ receives the confirmation message from the legitimate reader, its $mode$ is changed into '0'. As now, $\mathcal{T}$ only updates q into ν_0 but keeps r unchanged.
3. **Third Run: $\mathcal{R}^M$ interrogates $\mathcal{T}$**
   During the third protocol run, $\mathcal{R}^M$ interrogates with $\mathcal{T}$ again as in the first run for tracing the same tag $\mathcal{T}$ that has been interrogated in the first run. To achieve this, $\mathcal{R}^M$ sends the same challenge c to the tag and expects a repeated reply by $\mathcal{T}$. Recall that in the second run, a successful protocol run between $\mathcal{R}^T$ and $\mathcal{T}$ toggles $\mathcal{T}$ to a secure status $mode = 0$. Following the protocol, $\mathcal{T}$ shall reply with $ps||auth = r||\nu_1$, which is the same authentication information as that in the first run. It is thus easy for the attacker to trace the tag $\mathcal{T}$ by comparing the $ps||auth$ values.

4.3 Discussions

We stress that our attack is practical. There could be a number of ways to launch such an attack.

Recall that in the first protocol run of our attack, a malicious reader interrogates with a legitimate tag. We can further reduce this requirement if the adversary has minimum eavesdropping and blocking capabilities: in the first run, the adversary eavesdrops the first two protocol messages and blocks the third messages to make the protocol incomplete. Thereafter, the tag is triggered into an insecure state and the reader updates the status for the record of this tag. The attack continues with a successful second run and an incomplete third run (same as that of the first run). By comparing the eavesdropped

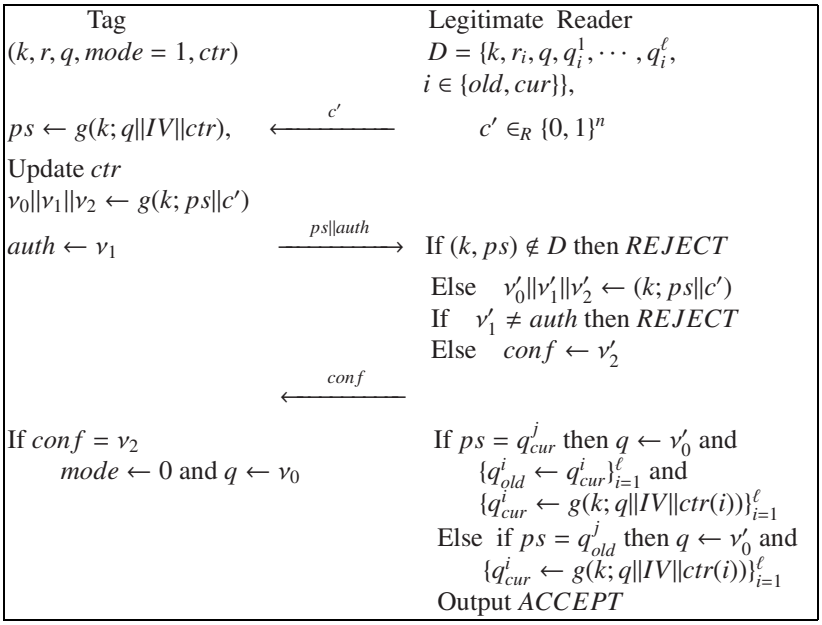

Fig. 3. Step Two of The Attack

messages in the first run and the third run, the adversary can trace the tag. Such an adversary is more stealthy as no active interrogation between a malicious reader and a legitimate tag is needed[1].

In summary, the attack can be extended, but not limited to the following forms:

- $\diamond$ --> $\mathcal{R}^M$ --> $\mathcal{R}^T$ --> $\mathcal{R}^M$ -->
- $\diamond$ --> $\mathcal{R}^T_{\mathcal{A}}$ --> $\mathcal{R}^T$ --> $\mathcal{R}^T_{\mathcal{A}}$ -->
- $\diamond$ --> $\mathcal{R}^M$ --> $\mathcal{R}^T$ --> $\mathcal{R}^T_{\mathcal{A}}$ -->
- $\diamond$ --> $\mathcal{R}^T_{\mathcal{A}}$ --> $\mathcal{R}^T$ --> $\mathcal{R}^M$ -->

Where $\mathcal{R}^T_{\mathcal{A}}$ denotes an adversary's presence in an interrogation between a trusted reader and a legitimate tag.

5 Cracking a Whole Supply Chain by Using the Basic Attack

Based on the basic three-run interleave attack, more advanced attacking strategies are designed to crack an RFID-enabled supply chain that implements the BMM protocol.

[1] Note that in the third run, a different challenge c'' could be used by a trusted reader to challenge the tag. As long as the r value is not updated in the second run, the ps value is still the same as the one in the first run.

5.1 Assumptions

We need to make several reasonable assumptions about an RFID-enabled supply chain before we elaborate on our attacking strategies.

1. ***Trusted Zone:***
   We consider a geographically distributed supply chain, in which each party in the supply chain may receive tagged articles, process these articles, and ship them out. For simplicity, we consider the area as a trusted zone inside a supply chain party, and public zone outside. An adversary is not able to interact with a legitimate tag in a trusted zone, but can interrogate with a tag in the public zone.
2. ***One-time Authentication:***
   While tagged articles are being processed by a supply chain party, the authentication is performed only once (*e.g.*, typically at the entry point of the trusted zone). This is reasonable as authentication procedure is much more expensive and time-consuming than identifier scanning procedure. As the area inside a supply chain party is considered as a trusted domain, indeed no additional authentication is necessary. While multiple scanning for identifying the tags is still allowed to facilitate other operations (which are not security related). This is to guarantee that only one successful session of authentication protocol is conducted in a trusted zone so that once the articles are shipped out to the public zone, the adversary can launch the tracing attack.
3. ***Sticky Adversary:***
   We assume that an adversary may possess multiple readers at multiple locations or equivalently possess one reader at multiple instant locations. In other words, we assume an ubiquitous adversary who is able to stick on the targeted articles in the public zone along a supply chain.

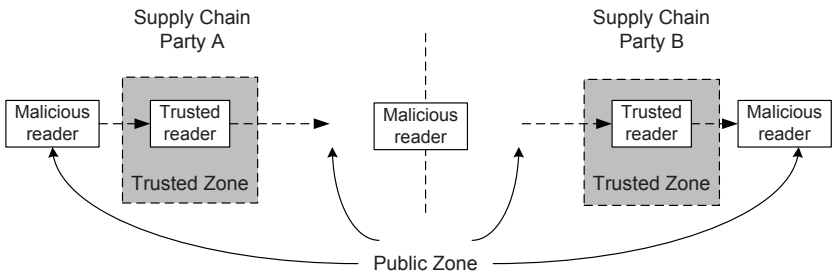

Fig. 4. An Example for Cracking Supply Chain System

With these assumptions, we illustrate how to crack a supply chain system as in Figure 4, where two supply chain parties are involved. In an attack, the adversary can setup malicious readers in the public zones near each supply chain party. Furthermore, two attacking strategies are given below.

5.2 Attacking Strategies

Case 1: Tracing a Single Tag along Supply Chain
Suppose an adversary targets on a particular article with an RFID tag $\mathcal{T}$. Before it arrives at supply chain party A, a malicious reader can launch its attack by interrogating with $\mathcal{T}$ and obtaining a ps value ($ps = r$) specific to this tag. Inside the domain of party A, $\mathcal{T}$ is authenticated once and processed in some other ways. At last, the article attached with $\mathcal{T}$ is shipped out. Once again, a malicious reader scans all outbound articles and find this particular tag with the pseudonym ps. Following on, the adversary repeats the attacks at various transportation locations visited by this article. Eventually, a list of visited sites of the article, $[\dashrightarrow A \Rightarrow B \Rightarrow C \Rightarrow D \Rightarrow E \dashrightarrow]$, are recorded, which enables the total visibility of this article (in the supply chain, which is serious breach of its privacy). The tracing attack is illustrated in Figure 5.

Case 2: Tracing Multiple Tags and Constructing Supply Chain Map
Suppose an adversary, for the purpose of obtaining commercial secret, targets on a manufacture who supplies its goods to various distributors, retailers, *etc.*, via complex supply chain paths. To construct such a map, he/she needs to trace all the goods attached with tags along their supply chains. As such, the adversary first builds a database for all the tags scanned immediately after the goods are shipped out. Suppose 100 tags are being scanned and recorded in the database, as shown in Figure 6. For each record of the database, $\surd$ (or $\times$) represents whether the tag is scanned at certain locations or not. 'ps' denote the pseudonyms of a tag, for simplicity, $|ps| = 32$. As long as the adversary

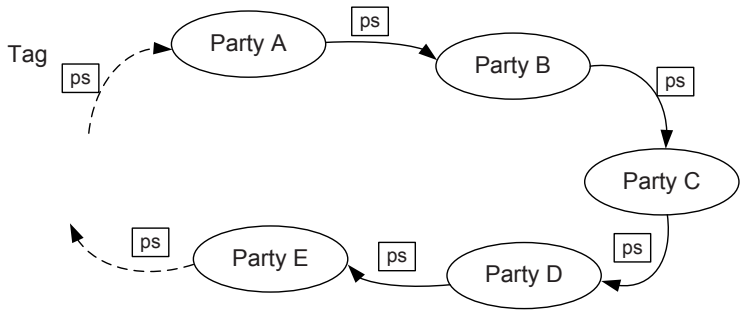

Fig. 5. Tracing A Single Tag along its Supply Chain

Tag	*ps*	Location 1	Location 2	Location 3	Location 4	Location 5	$\cdots$
Tag 1	09310A78	$\surd$	$\surd$	$\times$	$\times$	$\surd$	$\cdots$
Tag 2	38901D43	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\cdots$
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Tag 100	9A7B2811	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\cdots$

Fig. 6. The Adversary's Database

has enough resources to monitor all potential locations via a number of supply chains, it will finally draw a complete map for all delivery paths.

We assume that there are L possible locations for each tag, and the number of total tags is N. An attacker only needs to set up a database with size of $O(L \times N)$. He/she can efficiently query the information of a tag in polynomial time.

6 Improving the BMM Protocol

We observe that the main reason that the BMM protocol is vulnerable to our three-run interleave attack is that the pseudonym 'r' shared between the legitimate tag and the trusted reader is not properly updated. Intuitively, we solve the problem by updating the pseudonym r at both side after the third protocol message is sent even if the *mode* is 1 for the tag.

6.1 Improved Protocol

Our improved protocol is shown Figure 7. In the first round, our protocol is the same as the BMM protocol except that we separate the result $g(k; ps||c)$ into four parts ν_0, ν_1, ν_2 and ν_3. The new part ν_3 is used to update r when the tag's $mode = 1$, and other parts are

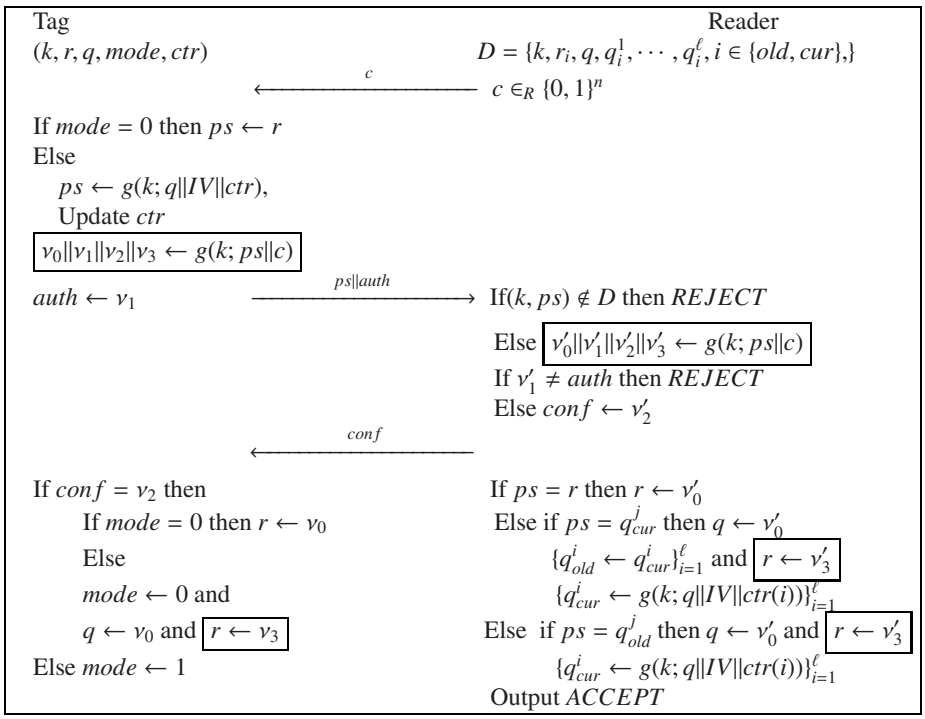

Fig. 7. Improved Protocol

kept the same as those of the original BMM protocol. In the second round, the reader also needs to divide the result of $g(k; ps||c)$ into four parts ν'_0, ν'_1, ν'_2 and ν'_3. Here, ν'_3 is used to update the reader when the received $ps = q_i^j, i \in \{old, cur\}, j = 1, 2 \cdots \ell$, and the reader keeps other operations the same as BMM protocol. In the third round, after receiving the confirmation message in the protocol, we update the status of r at the tag's side with $r \leftarrow \nu_3$ when '$mode = 1$' holds in the tag. In this round, we also update the status as described in the boxed parts at the reader in Figure 7. Since the pseudonym 'r' is updated whenever the *mode* is 0 *or* 1, the response of the tag behaves randomly at every interrogation. Therefore, our three-run interleave attack is no longer feasible.

6.2 Security Analysis

We analyze the improved protocol regarding some important security properties. The essential objective of the protocol is to achieve mutual authentication between a reader and a tag without disclosing the tag's identity to a third party, and it is based on a classic challenge-response mechanism. Without the shared secret, no polynomial probabilistic time (PPT) adversary can generate the authentication messages transferred between the two parties.

Our improved protocol's main purpose is to protect the tags' privacy, which means to keep tags' anonymity and untraceablity. Our improved protocol prevents tags from *tracing attack*. The meaning of untraceability contains two aspects: 1) The outputs of a tag in any two sessions are unlinkable, and 2) The outputs of readers are independent from those of tags. First of all, we analyze the outputs of any two sessions of a tag. For any two session i *and* j, $i \neq j$ of a tag, let $ps(i)||auth(i)$ and $ps(j)||auth(j)$ denote the output of the session i and j, respectively.

$$ps = \left\{ \begin{array}{ll} r, & mode = 0 \\ g(k; q||IV||ctr), & mode = 1 \end{array} \right\}$$

If $mode = 0$, then $ps = r$, and r is updated by a PRF $g(\cdot)$ in the tag after every successful protocol; otherwise, $ps = g(k; q||IV||ctr)$, the output of PRF $g(\cdot)$. Therefore, whether $ps = r$ or $ps = g(k; q||IV||ctr)$, $ps(i)$ and $ps(j)$ are independent as the output of a PRF are pairwise independent. The latter part of the tags' output is $auth = \nu_1$ which is a part of $g(k; ps||c)$ ($g(\cdot)$ is a PRF). Therefore, $auth(i)$ and $auth(j)$ are independent and unlinkable. As a result, $ps(i)||auth(i)$ is independent from $ps(j)||auth(j)$.

Second, we illustrate the output of the reader is independent from the output of a tag. We consider the output of tag is $ps||auth$ and the output of the reader is $conf$. ps is the input of the PRF $g(k; ps||c)$, and $conf$ is the output of PRF $g(\cdot)$. As the input and output of a PRF are independent, ps is independent from $conf$. The $auth = \nu_1$ is the second part of the output $g(k; ps||c)$, and $conf = \nu'_2$ is the third part of the output of the PRF $g(k; ps||c)$. Therefore, $auth$ is independent from $conf$. In all, the output of tag $ps||auth$ is independent from the output of the reader $conf$. Thus, the independence of outputs between different sessions of a tag and the independence of outputs between a reader and a tag guarantee the privacy of tags, and attackers cannot trace a tag by eavesdropping or active interrogations.

Based on challenge-and-respond technique, mutual authentication, PRF in both tag and reader, and update processes, Level-2 attacks cannot be applied here, for instance,

de-synchronization attack. Because the trusted reader keeps not only the newly updated values, but also the old values corresponding to a former corrupted protocol run, if a tag is pushed de-synchronized with the legitimate reader by a malicious adversary, it can still be recognized by referring to the older record $q^i_{old}, i = 1, 2, \cdots, N$ in the database. By successful mutual authentication, the reader and tag can be re-synchronized again. As we argue in section 4.1, our improved protocol can prevent level-2 attack, so it can possess the ability of counteracting weaker attacks. To counteract Level-1 attacks, for example, *eavesdropping attack*, an adversary can only obtain the challenge c and pseudonyms $ps||auth$ and v'_2, which are generated by PRF, but nothing else. Level-1 adversaries cannot link the information together to trace a tag, either. To prevent Level-2 attackers, the challenge-and-respond technique protects the reader from Denial-of-Service (Dos) attack. In addition, since fresh random numbers are generated by both the reader and the tag for mutual authentication and both the tag and the reader update their states after a successful protocol run, simple *spoofing* and *replay attacks* have negligible success rate. In addition, unlike some tree-based RFID protocol [11], if some tags are compromised unfortunately, released information will not affect other tags' secrecy due to that tags do not share secrets in our protocol.

Nevertheless, the improved protocol does not incur any additional cost with respect to storage and computation. Therefore, the lightweightness of the BMM protocol is maintained. As stated in [2], the database stores limited numbers of q^j_i, when these numbers are used up, the BMM protocol suffers from an "entrapment attack". The "entrapment attack" means *"the tag is prevented from communicating with authorized readers and can only be interrogated by the adversary"* [2]. In conclusion, as mentioned in Section 3, the security analysis we conducted is limited to level-1 to level-2 adversaries, while level-3 adversary is more powerful and may bring more harmful attacks to the existing protocol.

7 Related Work

Numerous of papers addressing RFID security and privacy have been published recently (please refer to [8] for a detailed literature survey). Our concern in this paper is on RFID reader/tag (mutual) authentication, which has also been rigorously studied in the literature [3] [5], [6], [9], [11], [13], [17].

A number of RFID authentication protocols based on secure one-way hash functions have been proposed [18]. In one of the previous works, Ohkubo, Suzuki and Kinoshita (OSK) proposed using of hash chain to update the internal states [13]. The scheme needs to compute two different hash function values, one to update the tag's secret and the other one to compute the response that is transmitted to the reader during tag identification. This method incurs a large overhead at the reader's side due to the exhaustive search in the back-end database to identify the tag. To mitigate the high search cost, Avoine and Oechslin proposed an optimization of the scheme using a time-memory trade-off for the computation of OSK hash chains [1]. However, in the later works [5] and [6], the authors pointed out that the optimized scheme is still vulnerable to tag impersonation attack and suffers from low scalability in the presence of attacks. Dimitriou in [5] proposed a challenge-response protocol for tag-reader authentication. However, it is still possible for an adversary to de-synchronize tags , leading to a denial of service.

Pseudonym Random Function (PRF) has been used in the design of RFID protocols. In [17], Tsudik proposed YATRAP protocol for RFID authentication. It only needs a single key and a single pseudorandom function (PRF) in a tag, but it is vulnerable to de-synchronization and denial of service (DoS) attacks as the timestamps can be manipulated in this protocol. Then, Chatmon, van Le and Burmester's YATRAP+ and OTRAP [3] were proposed to address the problem of YATRAP. Their schemes were essentially designed mainly for privacy-preserving identification of tags without providing reader authentication.

To reduce protocol overhead, people used tree-structure in RFID protocols. Dimitriou proposed a tree-based privacy-preserving RFID identification scheme [6]. In [11], Molnar, Soppera, and Wagner proposed a tree based scheme with a high scalability of identifying tags. Under these schemes, each tag stores a group of secret keys that lie along the path of a key tree from root to leaf layer maintained by the back-end database. During RFID identification, a tag responds a group of values computed using the group of secret keys over a random challenge and the reader will use the group of responses to identify a tag. However, it is difficult to implement key updating because some keys are shared by different tags. Even worse, if one tag's secret is compromised, it may affect others and leak their secrets.

Next, we analyze the overhead of typical RFID protocols. Assume there are N tags in an RFID system. The hash-lock protocol in [18] requires an exhaustive search in the reader's database to identify a tag, so the overhead of this protocol is $O(N)$. In the OSK protocol [13], the reader has to calculate hash values with $O(N)$ complexity. Molnar and Wagner's method manages the keys of tags in [12] with a cost of $O(log(N))$. Although the cost is already much better than the exhaustive search in other protocols, it is still non-ignorable when the number of tags increases to unimaginable amount. At this time, the scalability is a headache of the database's administrator.

Therefore, we can see even if with the help of hash function, PRF, and tree structure, it is still a difficult problem to balance the security and scalability. Our improved protocol not only guarantees nearly all the security properties such that it protects tags from eavesdropping attacks, spoofing attacks, replay attacks, de-synchronization attacks, tracing attacks and compromising attacks, but also possesses constant key-lookup time in terms of exact match of an index in a database.

8 Conclusion

In this paper, we investigate the security and scalability of a newly proposed RFID authentication protocol by Burmester, Medeiros and Motta [2]. We found a subtle flaw in this protocol. Under a weak adversary model, an attacker can launch a three-run interleave attack to trace and identify a tag. Further on, complex attacking strategies can be constructed on cracking the whole supply chain using such an authentication protocol. We improve this protocol by eliminating the flaw in BMM protocol. We provide a security analysis on the improved protocol and claim that it meets its security requirements and that it is as efficient as the original protocol in each invocation.

Acknowledgment. This work is partly supported by A*Star SERC Grant No. 082 101 0022 in Singapore.

References

1. Avoine, G., Oechslin, P.: A scalable and provably secure hash-based RFID protocol. In: Third IEEE International Conference on Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops, pp. 110–114 (2005)
2. Burmester, M., de Medeiros, B., Motta, R.: Robust, anonymous RFID authentication with constant key-lookup. In: ASIACCS 2008: Proceedings of the 2008 ACM symposium on Information, computer and communications security, pp. 283–291. ACM, New York (2008)
3. Chatmon, C., van Le, T., Burmester, M.: Secure anonymous RFID authentication protocols. Technical Report TR-060112 (2006)
4. Czeskis, A., Koscher, K.: RFIDs and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In: Conference on Computer and Communications Security – ACM CCS, October 2008. ACM Press, New York (2008)
5. Dimitriou, T.: A Lightweight RFID Protocol to protect against Traceability and Cloning attacks. In: Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm, Athens, Greece, September 2005. IEEE, Los Alamitos (2005)
6. Dimitriou, T.: A secure and efficient RFID protocol that could make big brother (partially) obsolete. In: IEEE International Conference on Pervasive Computing and Communications, pp. 269–275 (2006)
7. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM 33(4), 792–807 (1986)
8. Juels, A.: RFID Security and Privacy: A Research Survey. IEEE Journal on Selected Areas in Communications 24(2), 381–394 (2006)
9. Juels, A., Pappu, R., Parno, B.: Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. In: 17th USENIX Security Symposium, San Jose, CA, USA, July 2008, pp. 75–90. USENIX (2008)
10. Lim, T.-L., Li, T., Gu, T.: Secure rfid identification and authentication with triggered hash chain variants. In: ICPADS 2008: Proceedings of the 2008 14th IEEE International Conference on Parallel and Distributed Systems, Washington, DC, USA, pp. 583–590. IEEE Computer Society, Los Alamitos (2008)
11. Molnar, D., Soppera, A., Wagner, D.: A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 276–290. Springer, Heidelberg (2006)
12. Molnar, D., Wagner, D.: Privacy and Security in Library RFID: Issues, Practices, and Architectures. In: Pfitzmann, B., Liu, P. (eds.) Conference on Computer and Communications Security – ACM CCS, Washington, DC, USA, October 2004, pp. 210–219. ACM Press, New York (2004)
13. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to "Privacy-Friendly" Tags. In: RFID Privacy Workshop, November 2003. MIT, MA (2003)
14. Rieback, M., Crispo, B., Tanenbaum, A.: RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 184–194. Springer, Heidelberg (2005)

15. Rieback, M., Crispo, B., Tanenbaum, A.: The Evolution of RFID Security. IEEE Pervasive Computing 5(1), 62–69 (2006)
16. Rotter, P.: A Framework for Assessing RFID System Security and Privacy Risks. IEEE Pervasive Computing 7(2), 70–77 (2008)
17. Tsudik, G.: YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In: International Conference on Pervasive Computing and Communications – PerCom 2006, Pisa, Italy, March 2006. IEEE Computer Society Press, Los Alamitos (2006)
18. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 454–469. Springer, Heidelberg (2004)