BEST: A Bidirectional Efficiency-Privacy Transferable Authentication Protocol for RFID-Enabled Supply Chain

Saiyu Qi*†, Li Lu‡, Zhenjiang Li† and Mo Li†
*Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
†School of Computer Engineering, Nanyang Technological University, Singapore
‡School of Computer Science and Engineering, University of Electronic Science and Technology, China
syqi@ust.hk, luli2009@uestc.edu.cn, {lzjiang, limo}@ntu.edu.sg

***Abstract*—Radio Frequency Identification (RFID) technique is gaining increasing popularity in supply chain for the product management. By attaching a tag to each product, a reader can employ an authentication protocol to interrogate the tag's information for verification, which facilitates the automatic processing and monitoring of products in many applications. However, most current solutions cannot be directly used as they cannot balance the tradeoff between the privacy and efficiency for individual parties. In this paper, we design a bidirectional efficiency-privacy transferable (BEST) authentication protocol to address this issue. In a relatively secure domain, BEST works in an efficient manner to authenticate batches of tags with less privacy guarantee. Once the tags flow into open environment, BEST can migrate to provide stronger privacy protection to the tags with moderate efficiency degradation. The analytic result shows that BEST can well adapt to the RFID-enabled supply chain.**

***Keywords*-RFID; Supply chain; Authentication; Privacy; Communication efficiency**

I. Introduction

Due to the non-line-of-sight nature and cost-effectiveness, Radio Frequency Identification (RFID) technology [23] has been an important enabling solution for everyday applications, among which the most important one is supply chain management. A typical RFID-enabled supply chain usually involves the flow of tag batches among multiple parties. The tagged products are created by manufacturers and flow across various entities such as professional logistical companies, commercial points, customers, and so on. In the process of the supply chain, tags normally store the information related to products they attach to and are verified by different parties [4]. Specifically, when a party receives a batch of tags, it can use an RFID reader to remotely collect tags' information for verification such that fake tags can be identified and filtered out. Accordingly, products can be monitored and prevented from being stolen and replaced by malicious adversaries during the transformation.

The authentication operation has two fundamental requirements: *privacy protection* and *communication efficiency*. On one hand, a tag suffers from private threats due to its low cost design. An adversary can launch various attacks to steal the private information of tags. On the other hand, as the number of tags increases, communication efficiency becomes a bottleneck of processing time [17] for two reasons. First, since each tag needs to generate a reply for authentication, a large volume of data need to be retrieved. Second, to avoid collisions among tag replies, sophisticated scheduling mechanisms need to be used during the authentication process, which results in additional communication overhead. As shown in [17], the communication overhead to authenticate a tag batch with 10000 tags is 14 minutes in ideal transmission conditions, which is unpractical in large-scale RFID systems. In the RFID-enabled supply chain scenario, things become more complicated as the two requirements dynamically change when tags flow across different domains. At the beginning, batches of tagged products are processed through a series of manufacturers to complete the production. During the production process, a large volume of batches are transferred under a secure environment as each party has professional trucks and warehouses to transfer and store tags [19]. This motivates a party to concern more on the communication efficiency to facilitate the production process for economic reasons [4]. After the production, tagged products are distributed through logistical companies to individual customers. During the distribution process, large volume of batches are subdivided into small sets and distributed to multiple parties under an open environment [19]. In this scenario, communication efficiency is not a bottleneck and a party concerns more on the privacy as it is easier for an adversary to launch various attacks against tags [2].

Although there have been many approaches proposed to address the privacy and efficiency concerns in RFID systems, most of those approaches separately address the two concerns related to individual entities. They are not suitable for the supply chain scenario where we may have dynamic trade-off on privacy and efficiency. Recently, Cai *et al.* [27] propose a protocol set to balance the security and computation efficiency in supply chain scenario. Unfortunately, computation efficiency is not the bottleneck of the efficiency aspect as hash computations can be efficiently processed by powerful computers. In this paper, we jointly study the dynamic requirements of the privacy protection and

1521-9097/12 $26.00 © 2012 IEEE
DOI 10.1109/ICPADS.2012.65

Table I
NOTATIONS USED IN THIS PAPER

Notation	Description
f	a frame size
f_i	a slot in a frame
r_1, r_2, r_3	random numbers generated by the reader
t_j	a tag
id_j	an ID of a tag
(r_{j2}, k_j)	a pair of secret values shared between reader and a tag t_j
d_j	a record maintained by reader for a tag t_j
cur_j	a record used to save a new generated (r_{j2}, k_j) after a successful authentication
pre_j	a record used to save an old (r_{j2}, k_j)
new_j	during each legal authentication, a new secret pair is generated to update (r_{j2}, k_j), new_j is a record at the reader side to save this new secret pair
R	a message broadcasted by the reader in the first round communication
E	a response of a tag in the first round communication
U	a message broadcasted by the reader at the end of each slot f_i in the second round communication
$h()$	a hash function

the communication efficiency in RFID-enabled supply chain. We first identify and summarize the dynamic requirements of supply chains. We then design a lightweight bidirectional efficiency-privacy transferable authentication (BEST) protocol to satisfy the dynamic requirements. BEST works as a private preserving authentication protocol with dynamical balance between the privacy and communication efficiency.

The rest of this paper is organized as follows. Section II discusses the related works. We analyze the properties of RFID-Enabled supply chain and discuss its issues in section III. We present the design of BEST in Section IV. In Sections V and VI, we analyze the security and efficiency issues on our proposed system. We conclude this work in section VIII.

II. Related work

Currently, many works have been proposed to solve the privacy and efficiency concerns in RFID systems. Private preserving authentication (PPA) protocols [1], [2], [6], [7], [8], [9], [12], [14], [21] are such a type of protocols that allow a legal reader to authenticate the validity of a tag with privacy protection. In these protocols, a reader interacts with a tag in a three-round interaction to authenticate it. During the authentication, the reader first probes the tag via a query message with a nonce. Instead of answering the query in plaintext, the tag replies with a ciphertext using a secret key shared with the reader. The reader searches all the keys that it holds in the back-end database. If the tag is valid, the reader can find a proper key to recover the authentication message from the ciphertext, and thereby identify the tag. Most of these protocols, however, are communication inefficient as each tag authentication requires a three-round of message exchanges.

Besides PPA, Many approaches [15], [16], [24], [3], [18], [22], [10], [11], [26] have been proposed to efficiently acquire various information of a large volume of tags. ID collection algorithms [15], [16], [24] are proposed to acquire the ID of each tag in a large tag batch. Different with the data collection in sensor networks [25], the challenge here is that each tag replies a one-hop message simultaneously, which causes serious collisions. ID collection algorithms aim to settle this challenge by providing elegant scheduling mechanisms so that each tag can returns its ID correctly. Cardinality estimation [3], [18], [22] aims to estimate the cardinality of a large tag batch. Different with ID collection algorithms, each tag replies a short string to claim its existence. These replies do not need to be recovered individually at the reader side and can be directly used to estimate the cardinality. Missing tag monitoring [10], [11] aims to check if some tags are stolen from a large tag batch. Similar with cardinality estimation, each tag replies a short string to claim its existence. As the missing tags do not reply anymore, the whole reply pattern of the tag batch will change which will be detected by the tag owner. The advantage of this type of protocols is the high communication efficiency since only a small amount of data needs to be retrieved for the analysis. However, this type of protocols neither provides the basic authentication mechanism nor the privacy protection for tags.

Based on Anti-collision algorithms, Yang $et\ al.$ [17] design a batch authentication protocol SEBA to detect fake tags in tag batches. Given a batch of tags, a reader first broadcasts a size of frame, which consists of a series of time slots, to the tags. Then, each tag uses a secret key shared with the reader to choose a slot within the size and reply an echo at that slot. As the reader has all the secret keys of the tags, it can pre-compute a correct reply pattern of the frame and then compare it with the one formed by the tags. As fake tags do not know the secret keys, they may arbitrarily choose a slot to reply. This will make the reply pattern of tags deviate from the correct one and thus, fake tags can be detected with certain probability. SEBA enjoys high communication efficiency as each tag only needs to reply a short echo. Unfortunately, it cannot provide privacy protection for tags.

III. System model and assumption

We first briefly discuss the communication model used in our RFID system. We then introduce our system model and analyze the requirements we aim to achieve. For simplicity, we term ”a batch of tagged products” as ”a tag batch” in the remainder of this paper.

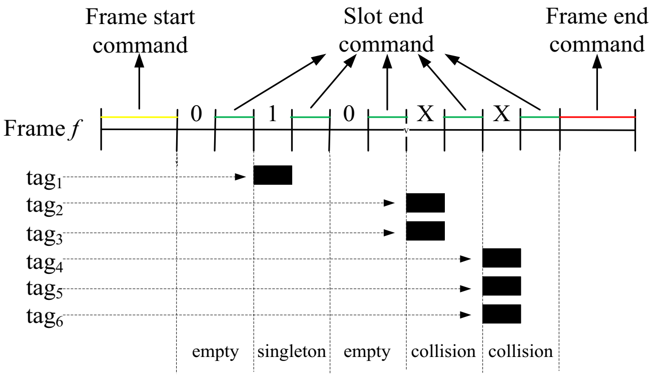

Figure 1. An example of the communication pattern.

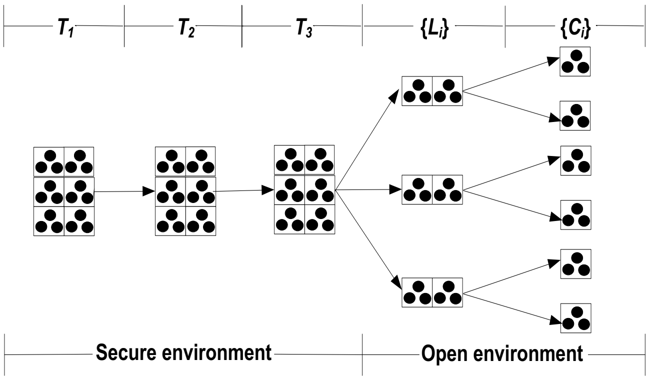

Figure 2. An illustration of RFID-enable supply chain.

A. Communication model

The communication pattern for our RFID system is based on the framed-slotted ALOHA model, which is used for the tag identification. An identification instance among the reader and a tag batch is composed of multiple frames. Each frame is further partitioned into slots. In an instance of identification, the reader first broadcasts a begin command containing the frame size f to inform the tags that a new frame starts. Each tag that probes this command will reply its *ID* at slot *h*(*ID*) *mod f* where *h*() is a hash function and *ID* is the identity of it. The reader then sequentially scans every slot in the frame. At the end of each slot, the reader perceives the slot as an empty slot, if no tag replies in this slot; a singleton slot, if a single tag replies in this slot, and a collision slot, if multiple tags reply in this slot simultaneously. After that, the reader broadcasts a command to end the slot. On the other hand, the tags that choose the current slot to reply can choose to keep active or silent based on this command in the subsequent frames. At the end of the frame, the reader sends an end command to terminate the whole frame. After that, the reader can decide to start a new frame or to end the identification instance. Fig. 1 shows an example with six tags. Suppose there are N tags and let f denotes the frame length, the probability of the i^{th} slot is an empty slot, a singleton slot, or a collision slot can be computed as follows:

$$P_0(N,f) = (1-\frac{1}{f})^N$$

$$P_1(N,f) = N\frac{1}{f}(1-\frac{1}{f})^{N-1}$$

$$P_X(N,f) = 1 - P_0(N,f) - P_1(N,f)$$

B. System model and requirements analysis

We consider three types of parties: a set of manufacturers $\{T_i\}$, a set of logistical companies $\{L_i\}$ and a set of customers $\{C_i\}$ in our supply chain model. The internal structure of the supply chain can be divided into two processes: production process and distribution process. In the production process, a large number of tag batches are processed through a series of manufacturers $\{T_i\}$ under a secure environment. The whole process can be formalized as $T_1 \rightarrow T_2 \cdots \rightarrow T_N$. After the production, the tag batches are further distributed through a set of logistical companies $\{L_i\}$ to a set of customers $\{C_i\}$ under an open environment. In each distribution, the batches are divided into multiple small sets and distributed to a set of parties. The whole process can be formalized as $T_N \rightarrow \{L_i\} \rightarrow \{C_i\}$. Fig. 2 shows an example. As shown in fig. 2, six tag batches are processed through three manufacturers T_1, T_2 and T_3 under a secure environment. After the production, they are further distributed through a set of logistical companies $\{L_i\}$ to a set of customers $\{C_i\}$ under an open environment. Specifically, T_3 first distributes the six tag batches to three logistical companies with each acquires two batches. Each logistical company further distributes its two batches to two customers with each acquires one batch.

Our goal is to design a bidirectional efficiency-privacy transformation protocol for tag batches to meet the dynamic requirements from different parties when tags flow through the supply chain. As we can see, each tag batch flows through a path from the initial manufacturer T_1 to a customer C_i. During the transformation, each party may be in a secure environment or an open environment, and thus has different requirements to the tag batch. Given a tag batch, we classify two kinds of tags in the batch: fake tags and valid tags. With this classification, we identify two kinds of requirements for the tag batch: *efficiency-oriented authentication* and *privacy-oriented authentication* as follows.

Efficiency-oriented authentication: Given a tag batch B_i, the validity of the tags needs to be authenticated with high communication efficiency. On the other hand, privacy protection of valid tags in the batch is not critical and can be ignored.

Privacy-oriented authentication: Given a tag batch B_i, the validity of the tags needs to be authenticated and the privacy of the valid tags needs to be protected. Specifically, we consider a wide range of privacy protections which have been commonly considered in PPA include: (1) *Unlinkable-privacy:* each time a valid tag is successfully updated, the adversary can not link the replies of the tag before the update with the ones after the update. (2) *Forward/backward-untreacability:* when a tag is transferred between two parties, each of them owns the sensitive information of the tag. A malicious party may use this information to explore the other's privacy. *Forward/backward untreacability* defenses against this attack. (3) *forward privacy:* due to its cost concern, a tag can be easily compromised by the adversary. The adversary can read the secret information stored in the tag to reveal the previous transaction of the tag. *Forward privacy* aims to prevent such an attack. On the other hand, relatively low communication efficiency should be tolerant.

When a tag batch flows through a series of parties, each party in the secure environment aims the first requirement and each party in the open environment aims the second requirement.

IV. Protocol design

A. Basic idea

BEST is constructed by combining a privacy-preserving protocol O-FRAP [8] and an efficient probabilistic technique [11], [17]. Specifically, BEST consists of two components: *initialization* and *batch-authentication*. When an initial manufacturer T_1 produces a tag batch B_i with cardinality N, it performs an initialization procedure to initialize the internal information of the tag batch and generates a record batch D_i. T_1 then injects both the tag batch B_i and the message (N, D_i) into the supply chain. When a party receives B_i and (N, D_i), it performs a batch-authentication procedure to authenticate the validity of B_i. The batch-authentication component provides two sub protocols: multi round-checker and three phase-detector and the party can flexibly choose one of the two sub protocols based on its requirement. The design of BEST relies on the following three key points.

Key point 1: To achieve the second requirement, each tag in the tag batch needs to be interrogated by the reader in an authentication process. The reason is that as the valid tags are mixed with the fake tags, we need to first interrogate a tag to decide its validity and then update its internal state if it is valid. To protect the privacy of all the valid tags, each tag needs to be interrogated. This motivates us to design a sub protocol called multi-round checker to check each tag in a tag batch. Multi-round checker employs a privacy-preserving protocol O-FRAP [8] and incurs a relatively high communication efficiency. Intuitively, each tag must reply in a single slot to be correctly interrogated. This means that the collision slots and empty slots in a frame are useless, and the party has to repeat multiple frames until each tag can choose at least one singleton slot to reply.

Key point 2: To achieve the first requirement, communication efficiency should be largely improved. To achieve this goal, we remove the per-tag interrogation manner as analyzed below. First, as privacy protection is not a critical point in the first requirement, we do not need to interrogate all the valid tags. Second, if we can show that there is no fake tag in the tag batch, all fake tags need not to be interrogated. This motivates us to design a sub protocol called three-phase detector to first detect fake tags for a tag batch. Three-phase detector uniquely combines O-FRAP [8] and an efficient probabilistic technique [11], [17], and does not need to interrogate each tag in the tag batch. Specifically, three-phase detector could use both collision slots and singleton slots to detect fake tags, which largely improves the communication efficiency. Obviously, if no fake tags are detected for a tag batch, the first requirement is automatically achieved against the tag batch.

Key point 3: In key point 2, we show that under certain condition, three-phase detector can be used to achieve the first requirement. We further observe that as the first requirement is raised in a secure environment, it is reasonable to assume that only a small fraction of tag batches contain fake tags. This means that a party in a secure environment can first use three-phase detector to scan the received tag batches to achieve the first requirement in most cases. This ensures that valid tag batches can be quickly transferred to downstream parties and only the tag batches that contain fake tags are retained for further checking (for example, interrogate each tag one by one to filter all the fake tags).

B. Initialization

Suppose a manufacturer T_1 generates a tag batch B_i with cardinality N. T_1 then creates an empty record batch D_i and initializes each tag $t_j \in B_i$ as follows. T_1 generates a production message m_j, which contains a unique ID of t_j, the production date and other production description about the product tagged by t_j. T_1 then generates a secret pair (r_{j2}, k_j) and loads the pair to t_j. The r_{j2} is a random number used in authentication and the k_j is t_j's secure key. Finally, T_1 creates a record d_j=(m_j, pre_j, cur_j) and adds it to D_i. pre_j and cur_j are two subrecords to save the secret pair (r_{j2}, k_j). Initially, pre_j = ($\perp$, $\perp$) and cur_j = (r_{j2}, k_j). After the initialization, T_1 injects the tag batch B_i and the message (N, D_i) into the supply chain.

We also introduce two algorithms of O-FRAP [8] which are used by tags and readers respectively. **Algorithm 1** is used by a tag to generate a response and **Algorithm 2** is used by a reader to verify a tag in a singleton slot. **Algorithm 2** uses an operation called update(). Suppose a reader runs **Algorithm 2** to verify a tag t_j. During the procedure, both the reader and t_j agree on a new secret pair (r'_{j2}, k'_j) and

Algorithm 1 Tag reply generation

```
Input:
    secret pair ($r_{j2}$, $k_j$), random number $r_1$
Procedure:
    compute a hash value $h(k_j||r_1||r_{j2})$;
    divide $h(k_j||r_j||r_{j2})$ into four equal length-parts: $p_1||p_2||p_3||p_4$;
    compute $r'_{j2}$ as: $r'_{j2} \leftarrow r_{j2}$;
    update $r_{j2}$ as: $r_{j2} \leftarrow p_1$;
    return the message $E = \langle p_2||r'_{j2} \rangle$
```

use it to update the shared (r_{j2}, k_j). The reader uses a subrecord new_j to save the newly generated pair (r'_{j2}, k'_j). Also, the reader will detect whether the t_j is using pre_j or cur_j currently. If t_j uses cur_j, the reader will replace pre_j with cur_j and cur_j with new_j. If t_j uses pre_j instead, then cur_j is replaced with new_j, while pre_j is preserved. We denote the above operation as update(new_j, d_j).

C. Batch-authentication

When a party receives the tag batch B_i and a message (N, D_i), the party performs a batch-authentication procedure against B_i. Batch-authentication consists of two sub protocols: three phase-detector and multi round-checker. If the party prefers the first requirement, it can choose three phase-detector. If it prefers the second requirement, it can choose multi round-checker.

Three phase-detector: This protocol provides an efficient fake tag detection mechanism with no privacy protection. To start, the party first selects an optimal frame size f (the computation of f will be given in the next section) and a random number r_1. the party then creates an empty list L. For each d_j=(m_j, pre_j, cur_j) in D_i, the party adds cur_j to L. The party then constructs a response vector E_{RV} which is a state map with size f. Each element $E_{RV}[i]$($1 \leq i \leq f$) has three states, which we present as 0, 1 and 2. Initially, each $E_{RV}[i]$ is set to 0. For each secret pair (r_{j2}, k_j) in L, the party computes $h(k_j||r_1)$ *mod* f and checks the state of $E_{RV}[h(k_j||r_1)$ *mod* $f]$. If E_{RV} $[h(k_j||r_1)$ *mod* $f]$=0, sets it to 1. If $E_{RV}[h(k_j||r_1)$ *mod* $f]$=1, sets it to 2. Else do nothing.

After that, the party broadcasts the message $R = \langle r_1||f \rangle$ to B_i. When receiving R, each tag t_j computes a slot number $f_j = h(k_j||r_1)$ *mod* f, runs **algorithm 1** to generate a message $E = \langle p_2||r'_{j2} \rangle$ and replies E at slot f_i. On the other hand, the party overhears the communication channel to retrieve responses from the tags. At each slot f_i ($1 \leq i \leq f$), if f_i is empty, this means that no tag replies on this slot. The party directly moves to f_i+1. Otherwise, the party interrogates tags in the following two manners:

Precise detection: If f_i is a singleton slot, the party retrieves a single message E from this slot and runs **Algorithm 2** to authenticate the tag. If the tag is accepted, the party broadcasts the message $U = p'_3$ at the end of f_i and moves to next slot. Otherwise, the party stops and asserts the existence of fake tags.

Algorithm 2 Tag response verification

```
Input:
    the record batch $D_i$={$d_1$, $d_2$, ...$d_N$}, random number $r_1$, tag's reply $E$
Procedure:
 1: parse $E$ as $\langle p_2||r'_{j2} \rangle$
 2: for each record $d_j$=($m_j$, $pre_j$, $cur_j$) and each secure key $k'_j \in \{k'_{j1}, k'_{j2}\}$, where $k'_{j1}$ and $k'_{j2}$ are the secure keys contained in $cur_j$ and $pre_j$ respectively ($1 \leq j \leq N$) do
 3:     compute $h(k'_j||r_1||r'_{j2})$;
 4:     parse $h(k'_j||r_1||r'_{j2})$ as four equal length-parts: $p'_1||p'_2||p'_3||p'_4$;
 5:     decide if $p'_2$ = $p_2$;
 6:     if equal then
 7:         generate $new_j$ = ($p'_1$, $p'_4$);
 8:         execute the operation update($new_j$, $d_j$);
 9:         exit for
10:     else
11:         return to step 2;
12:     end if
13: end for
```

Probabilistic detection: If f_i is a collision slot, the party searches $E_{RS}[f_i]$. If $E_{RS}[f_i] < 2$, which means that at most one valid tag exists but more than two tags replies at this slot, the party asserts the existence of fake tags. Otherwise, the party generates a dummy string and broadcasts a message U =dummy string at the end of f_i and moves to next slot.

After checking each slot of the frame, B_i will be accepted as valid if no fake tags are detected. For each tag t_j which chooses f_i to reply, it will overhear the communication channel at the end of f_i to receive a message U from the reader. t_j then checks whether $U = p_3$? If yes, t_j updates its key k_j as $k_j \leftarrow p_4$; else it does nothing. After that, t_j keeps silent in the remainder of the authentication.

Multi round-checker: This protocol provides a low efficient fake tag checking mechanism with strong privacy protection. To start with, the party first selects an optimal frame size f=N (the expected number of singleton slots attains maximum when f equals the current active tags) and a random number r_1. The party then creates an empty list S_1 and broadcasts the message R = $\langle r_1||f \rangle$ to B_i. After receiving R, each t_j computes a slot number f_i = $h(k_j||r_1)$ *mod* f, runs **algorithm 1** to generate a message $E = \langle p_2||r'_{j2} \rangle$ and replies E at slot f_i.

On the other hand, the party overhears the communication channel at each slot f_i to retrieve responses from the tags. At each slot f_i ($1 \leq i \leq f$), if f_i is empty (f_i = 0) or collided (f_i = X), the party broadcasts a message U=dummy string to end this slot and moves to f_i+1. Otherwise, a single message E is received and the party runs **Algorithm 2** to authenticate the tag. If the tag is accepted, we say that the tag is correctly authenticated. The party then broadcasts the message U = p_3' to end this slot, adds d_j to S_1 and moves to next slot. Otherwise, the tag is identified as fake and the party directly moves to next slot.

For each tag t_j which chooses f_i to reply, it will overhear the communication channel at the end of f_i to receive a message U from the reader. After that, t_j checks whether U = p_3? If yes, t_j updates its key k_j as $k_j \leftarrow p_4$ and keeps silent in the remainder of the authentication; else does nothing and keeps active in the next frame. At the end of the frame, the party decides whether $|S_1|<N$? If yes, the party chooses a new random number r_1 and computes a frame size f=N-$|S_1|$ and restarts a new frame with (r_1, f). The above process is repeated until no tag replies.

V. Security analysis

In this study, we consider two types of attacks launched by an adversary: (1) fake tag injection attack: the adversary may replace some valid tags from a tag batch with fake ones. (2) various attacks to explore the privacy of valid tags. We then analyze the resistance of BEST against these attacks.

A. Security of three phase-detector

Fake tag detection: The analysis is similar with [17]. Let the cardinality of the tag batch is N and assuming there are $\epsilon N (1 \geq \epsilon \geq 0)$ fake tags in the batch. Three phase-detector provides two manners: precise detection and probabilistic detection to detect fake tags in a tag batch. Precise detection checks all the singleton slots in the frame. Based on the security of [8], if the tag replied in a singleton slot is a fake one, it is computationally hard for the fake tag to pass the precise authentication. As a result, the detection probability $P_{f_i}^{PP}$ of this case is:

$$P_{f_i}^{PP} = P_0(N(1-\epsilon), f) P_1(N\epsilon, f) \tag{1}$$

Probabilistic detection checks all the collision slots in the frame. For a collision slot f_i, the party checks the state of $E_{RV}[f_i]$. If $E_{RV}[f_i] < 2$, fake tags can be detected. Specifically, consider the following two sub cases. If $E_{RV}[f_i]$ = 0, there should be no any valid tag selecting this slot. But the result shows that more than one tag chooses this slot. Fake tags can thus be detected. If $E_{RV}[f_i]$ = 1, there should be at most one valid tag choosing this slot. But the result shows that more than one tag chooses the slot. Fake tags can thus be detected. The detection probabilities Pr($E_{RV}[f_i]$=0) and Pr($E_{RV}[f_i]$=1) of the above two sub cases are:

$$Pr(E_{RV}[f_i] = 0) = P_0(N, f) P_X(N\epsilon, f) \tag{2}$$

$$\begin{aligned} Pr(E_{RV}[f_i] = 1) &= P_1(N, f) P_X(N\epsilon, f) \\ &+ P_1(N - N\epsilon, f) P_0(N\epsilon, f) P_1(N\epsilon, f) \end{aligned} \tag{3}$$

Let $P_{f_i}^{FP}$ denotes $Pr(E_{RV}[f_i] = 0)$+$Pr(E_{RV}[f_i] = 1)$, the total detection probability of three phase-detector over the whole frame is:

$$P = 1 - (1 - (P_{f_i}^{FP} + P_{f_i}^{PP}))^f \tag{4}$$

Note that P is a function with three parameters N, f and ϵ. So we denote P as $P^{(N,f,\epsilon)}$. With the fixed N and ϵ, the party can choose f to adjust $P^{(N,f,\epsilon)}$. In fact, we can provide a (1 - δ, ϵ) guarantee such that fake tags can be detected with probability 1 - δ by computing f to satisfy the following equation:

$$P^{(N,f,\epsilon)} = 1 - \delta \tag{5}$$

In real applications, it may hard for the party to determine ϵ as it has no idea how many fake tags the adversary will inject. Fortunately, we can compute a lower bound of the detection probability $P^{(N,f,\epsilon)}$ which is proportional to ϵ. As a result, the party can set ϵ as a small enough value, fix 1 - δ as a large enough value based on its requirement and compute f based on (6) (using the lower bound instead of $P^{(N,f,\epsilon)}$). If the real ϵ is larger than the fixed one, the real detection probability $P^{(N,f,\epsilon)}$ is also larger than the fixed 1 - δ.

Privacy protection: Three phase-detector cannot provide strong privacy protection as it cannot update the internal state of each valid tag in an execution instance. For example, three phase-detector suffers tracking attack and thus, unlinkable privacy cannot be guaranteed. Each time the reader broadcasts a message R=$\langle r_1 || f \rangle$, a valid tag will choose a slot f_i=($k_j || r_1$) *mod* f to reply. Obviously, unless its secret key k_j is updated, the adversary can broadcast the same R to trace it as it will always choose the same slot to reply. Similarly, the forward privacy and forward and backward untracability also cannot be achieved for the same reason.

B. Security of multi round-checker

The secure properties of multi round-checker is inherited from [8] as we use the protocol O-FRAP proposed in [8] as a sub protocol. As analyzed in [8], O-FRAP has the following secure properties under UC framework:

- *Mutual authenticity:* when the reader correctly authenticates a tag, it can be guaranteed that the tag is a valid one. If a valid tag accepts a reader, it can be guaranteed that the reader is a valid one.
- *Untraceable privacy:* the responses of a tag is uncorrelated so that the adversary cannot trace it.
- *Forward privacy:* Tags are easily compromised due to its low cost design. When an adversary compromises a tag, it can immediately get the secret information stored

in the tag. O-FRAP guarantees that the adversary cannot use the secret information to explore the previous transactions of the tag.

Actually, multi round-checker consists of multiple rounds of authentications. The authentication in each round can be treated as a concurrent composition of multiple O-FRAP executions. As the secure properties of O-FRAP are proved under UC framework, such a concurrent composition preserves these properties. Therefore, multi round-checker inherits the secure properties of O-FRAP.

Fake tag checking: In multi round-checker, each tag in a tag batch must be correctly authenticated at least once. Based on the *Mutual authenticity*, a fake tag cannot pass the authentication and thus, will be identified.

Privacy protection: Multi round-checker can provide the following privacy protections:

- *Unlinkable privacy:* This property is directly inherited from O-FRAP. Note that each tag also needs to choose a random slot $h(k_j||r_1)\ mod\ f$ in a frame to reply. As a valid tag updates its internal key k_j in each successful authentication, the adversary cannot use this information to trace the tag.
- *Forward privacy:* This property is directly inherited from O-FRAP.
- *Forward and backward untracability:* This property is based on *forward privacy*. When a sender wants to achieve forward untracability, it can runs an instance of multi round-checker before transferring the tag batch. On the other hand, if the receiver wants to achieve backward untracability, it can also runs an instance of multi round-checker when receiving the tag batch.

VI. Performance analysis

In this section, we compare the communication efficiency of three phase-detector, multi round-checker and two types of privacy-preserving protocols: LAST [21] and ACTION [1] by simulation. In the current literature, also various privacy-preserving protocols have been proposed, many of them are shown to be insecure or cannot provide enough security properties in the subsequent works. As a result, we choose LAST and ACTION, both of which are proved to be secure, for comparison.

We set the parameters of the four schemes large enough to guarantee security as follows:

- The random numbers used in all the four schemes are 64 bits.
- The output length of hash function used in ACTION is 128 bits.
- The output length of hash function used in LAST is 64 bits.
- The length of index information used in LAST is 64 bits.
- The output length of hash function used in multi round-checker and three phase-detector is 256 bits.

Table II
COMPARISON OF COMMUNICATION EFFICIENCY.

tag size	ACTION	LAST	multi round	three phase
1000	113.6s	6.4s	4.8s	1.92s
2000	227.2s	12.8s	9.6s	3.84s
3000	340.8s	19.2s	14.4s	5.76s
4000	454.4s	25.6s	19.2s	7.68s
5000	568s	32s	24s	9.6s
6000	681.6s	38.4s	28.8s	11.52s
7000	795.2s	44.8s	33.6s	13.44s
8000	908.8s	51.2s	38.4s	15.36s
9000	1022.4s	57.6s	43.2s	17.28s
10000	1136s	64s	48s	19.2s

Since every bit consumes the same transmission time of 25μs on average [20], we measure the communication efficiency by computing the time cost of data transfer (in terms of bits). Consider a tag batch with N tags, we change N from 1000 to 10000 and set δ=0.01 and ϵ=0.05. We show the result in Table II.

We observe that the communication overhead of multi round-checker, LAST and ACTION are 2.5, 3.3 and 59 times higher than three phase-detector's respectively. This is because the three former schemes require each tag to exchange data with reader once and thus, the time consumed for delivering the data is linear to the batch size. On the contrary, three phase-detector uses both singleton and collision slots in a frame to detect fake tags. Furthermore, we do not consider the collision problem when evaluating the communication overhead of multi round-checker, LAST and ACTION. In reality, a tag does not know when to start a data exchange instance with reader. If more than two tags start data exchange instances simultaneously, Collision happens. To avoid the collision, we need scheduling mechanisms to assign slots for a batch of tags. For example, if we use Framed Slotted ALOHA for scheduling, each tag needs to exchange data with reader more than two times to avoid collision. This means that the communication overhead of multi round-checker, LAST and ACTION are at least 5, 6.6 and 118 times higher than three phase-detector's respectively. This indicates that three phase-detector enjoys a mach higher communication efficiency than the remainder protocols in large-scale scenarios.

In summary, LAST, ACTION and multi round-checker aim to interrogate each tag in a tag batch to authenticate them and protect the privacy of the valid ones. Such a design incurs high communication overhead and thus, is not convenient to deploy in the secure environment of supply chain.

VII. Conclusion

In this paper, we propose a bidirectional efficiency-privacy transferable (BEST) authentication protocol to satisfy the dynamic requirements of RFID-enabled supply chain. The most significant advantage of BEST is that it can balance the privacy and communication efficiency dynamically. We believe that our techniques will facilitate the deployment of RFID techniques in supply chain.

VIII. Acknowledgment

We would like to thank the anonymous reviewers for their helpful and valuable comments. This work is supported in part by Nanyang Technological University NAP Grant M4080738.020 and National Natural Science Foundation of China (NSFC) (No. 60903155 and No. 61173171).

References

[1] L. Lu, J. Han, R. Xiao and Y. Liu. ACTION: Breaking the Privacy Barrier for RFID Systems. In Proceedings of IEEE INFOCOM, Pages 1953-1961, 2009.

[2] L. Lu, Y. Liu, L. Hu, J. Han and L. M. Ni. A Dynamic Key-Updating Private Authentication Protocol for RFID Systems. In Proceedings of IEEE PerCom, Pages 13-22, 2007.

[3] B. Sheng, Q. Li and W. Mao. Efficient Continuous Scanning in RFID Systems. In Proceedings of IEEE INFOCOM, Pages 1-9, 2010.

[4] Y. Li, X. Ding. Protecting RFID Communications in Supply Chains. In Proceedings of ACM ASIACCS, Pages 234-241, 2007.

[5] S. Canard, I. Coisel. Data Synchronization in Privacy-Preserving RFID Authentication Schemes. In Proceedings of RFIDSec, 2008.

[6] T. Li, R. Deng. Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol. In Proceedings of ARES, Pages 238-245, 2007.

[7] M. Lehtonen, T. Staake, F. Michahelles and E. Fleisch. From Identification to AuthenticationCA Review of RFID Product Authentication Techniques. In Proceedings of RFIDSec, 2006.

[8] T. Van Le, M. Burmester and B. de Medeiros. Universally Composable and Forward-secure RFID Authentication and Authenticated Key exchange. In Proceedings of ACM ASIACCS, Pages 242 - 252, 2007.

[9] M. Burmester, B. De Medeiros and R. Motta. Robust, anonymous RFID authentication with constant key-lookup. In Proceedings of ACM ASIACCS, Pages 283-291, 2008.

[10] T. Li, S. Chen and Y. Ling. Identifying the Missing Tags in a Large RFID System. In Proceedings of ACM MobiHoc, Pages 1-10, 2010.

[11] C. Tan, B. Sheng and Q. Li. How to Monitor for Missing RFID Tags. In Proceedings of IEEE ICDCS, Pages 295-302, 2008.

[12] A. Juels, S. Weis. Defining strong privacy for RFID. In Proceedings of IEEE PerCom, Workshop PerTec, Pages 342-347, 2007.

[13] S. Fouladgar, H. Afifi. An Efficient Delegation and Transfer of Ownership Protocol for RFID tags. In Proceedings of EURASIP Workshop on RFID Technology, 2007.

[14] T. Dimitriou. A Secure and Efficient RFID Protocol that could make Big Brother (partially) Obsolete. In Proceedings of IEEE PerCom, 2006.

[15] EPCglobal Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960MHz version 1.0.9 2005.C.

[16] S. R. Lee, S. D. Joo and C. W. Lee. An Enhanced Dynamic Framed Slotted ALOHA Algorithm For RFIDTag Identification. In Proceedings of MobiQuitous, Pages 166-172, 2005.

[17] L. Yang, J. Han, Y. Qi and Y. Liu. Identification-free batch authentication for RFID tags. In Proceedings of IEEE ICNP, Pages 154-163, 2010.

[18] M. Kodialam, T. Nandagopal. Fast and Reliable Estimation Schemes in RFID Systems. In Proceedings of ACM MobiCom, Pages 322-333, 2006.

[19] A. Juels, R. Pappu and B. Parno. Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. In Proceedings of USENIX Security, Pages 75-90, 2008.

[20] S. Weis, S. Sarma, R. Rivest and D. Engels. Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems. Lecture notes in Computer Science, Vol(2802), Pages 50-59, 2004.

[21] L. Lu, Y. Liu and X. Li. Refresh: Weak Privacy Model for RFID Systems. In Proceedings of IEEE INFOCOM, Pages 1-9, 2010.

[22] Y. Zheng, M. Li. PET: Probabilistic Estimating Tree for Large-Scale RFID Estimation. IEEE Transactions on Mobile Computing, Vol. 11(Issue 11): Pages 1763-1774, November, 2012.

[23] Y. Liu, Y. Zhao, L. Chen, J. Pei and J. Han. Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays. IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 11, Nov, 2012.

[24] S. Tang, J. Yuan, X. Li, G. Chen, Y. Liu and J. Zhao. RASPberry: A Stable Reader Activation Scheduling Protocol in Multi-Reader RFID Systems. In Proceedings of IEEE ICNP, Pages 304 -313, 2009.

[25] Z. Li, M. Li, J. Wang and Z. Cao. Ubiquitous data collection for mobile users in wireless sensor networks. In Proceedings of IEEE INFOCOM, Pages 2246-2254, 2011.

[26] Y. Zheng, M. Li. Fast Tag Searching Protocol for Large-Scale RFID Systems. In Proceedings of IEEE ICNP, Pages 363-372, 2011.

[27] S. Cai, Y. Li, T. Li and R. Deng. Achieving High Security and Efficiency in RFID-Tagged Supply Chains. International Journal of Applied Cryptography, Vol. 2, No. 1, Jul, 2010.