12-2010

Protecting and Restraining the Third Party in RFID-Enabled 3PL Supply Chains

Shaoying CAI
Singapore Management University, shaoyingcai.2009@smu.edu.sg

Chunhua SU

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

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

DOI: https://doi.org/10.1007/978-3-642-17714-9_18

Protecting and Restraining the Third Party in RFID-Enabled 3PL Supply Chains

Shaoying Cai[1], Chunhua Su[1], Yingjiu Li[1], Robert Deng[1], and Tieyan Li[2]

[1] Singapore Management University, 80 Stamford Road, Singapore
[2] Institute for Infocomm Research (I2R), 1 Fusionopolis Way, Singapore

Abstract. "Symmetric secret"-based RFID systems are widely adopted in supply chains. In such RFID systems, a reader's ability to identify a RFID tag relies on the possession of the tag's secret which is usually only known by its owner. If a "symmetric secret"-based RFID system is deployed in third party logistics (3PL) supply chains, all the three parties (the sender of the goods, the receiver of the goods and the 3PL provider) should have a copy of those tags' secrets to access the tags. In case the three parties in 3PL supply chain are not all honest, sharing the secrets among the three parties will cause security and privacy problems. To solve these problems, we firstly formalize the security and privacy requirements of RFID system for 3PL supply considering the existence of the internal adversaries as well as the external adversaries. Then we propose two different protocols which satisfy the requirements, one is based on aggregate massage authentication codes, the other is based on aggregate signature scheme. Based on the comparisons of the two protocols on performance and usability, we get the conclusion that overall the aggregate MAC-based solution is more applicable in 3PL supply chains.

1 Introduction

Radio Frequency IDentification (RFID) technology is an automatic identification technology that uses radio waves (wireless) to transmit the messages. RFID systems consist of two main components: tags and readers. Tags are radio transponders attached to physical objects, while radio transceivers, or readers, query these tags for identifying information about the objects to which tags are attached. RFID technology, when combined with internet and networking technology, enables product information to be collected, integrated, shared, and queried at various levels (e.g., item, pallet, case and container) in real time in a supply chain. Third party logistics (3PL) is one of the most dominating kind of supply chains, it has been widely adopted by many companies. The companies outsources part or all of their supply chains to professional logistics service provider to get better management efficiency and at same time reduce the cost.

RFID-based system's high efficiency is due to the contactless identifying property; however, this property also benefits the potential adversaries. Radio transmits through open air, then an adversary can eavesdrop or interfere

the communications between reader and tag without the awareness of the tag's owner. Dozens of cryptographic protocols have been proposed to provide secure and private identification and authentication of the tag (Sometimes the reader authentication is also required), such as the "hash-lock" protocol of Weis et al. [15], OSK protocol [11] of Ohkubo, Suzuki and Kinoshita, and the tree-based protocol of Molnar et al. [10]. There are also many works [9,14] et al. deal with the secure and private ownership transfer between two parties. Most of these solutions for authentication and ownership transfer are "symmetric secret"-based that an authorized reader shares a secret with each tag.

The "symmetric secret"-based solutions are designed to protect the system against external adversaries who do not know the secrets. However, in 3PL supply chains that three parties (the sender of the goods, the receiver of the goods and the 3PL provider) are involved in the processing of the tags, internal adversaries should be considered. In 3PL supply chains, all of the three parties need to access the tags, hence all of them should have a copy of the secret when a "symmetric secret"-based solution is deployed. With a tag's secret, any party can fabricate the tag. In case inside adversaries exist, disputes on the goods' originality will be hard to solve since all the three parties have the ability to fabricate the tags.

Currently, there does not exist any solution that is suitable for 3PL supply chains considering the exitance of internal adversaries. It does not mean that putting effort on 3PL supply chains is not necessary. 3PL has large market size, a study[1] shows that in U.S. the 3PL market gross revenues reached $107.1 billion in 2009 and 8.3% growth is predicted for 2010. It is crucial to enhance the security and privacy level of RFID-enabled system for 3PL supply chains. We are the first ones to work on this new direction. Our contributions can be summarized as follows:

- We firstly formulate the security and privacy requirements of RFID system for 3PL supply chains with respect to both the internal and external adversaries.
- To execute the authentication of the tags in 3PL supply chains without revealing the secrets to the 3PL provider and the receiver of the goods, we provide two solutions that enable the tags' aggregate authentication on batch level. One solution is based on an aggregate Message Authentication Code(MAC), the other is based on an aggregate signature scheme.

Both the two solutions match the privacy and security requirements of 3PL supply chains. The comparisons on performance and usability between the two proposals show that the aggregate MAC-based solution is more applicable than the aggregate signature-based solution in 3PL supply chains.

The Organization of this paper. The rest of this paper is organized as follows. In Section 2, we model the 3PL supply chain and analyze the security and privacy requirements, introduce the motivation of our work in details. In Section

[1] `http://www.prnewswire.com/news-releases/us-and-global-third-party-logistics-market-analysis-is-released-94771894.html`

3 and Section 4, we respectively show our aggregate MAC-based solution and aggregate signature-based solution as well as the analysis on security and privacy. In Section 5, we compare the performance and usability of the two schemes. We review the related work in Section 6 and finally conclude in Section 7.

2 Third Party Logistics Supply Chain

In this section, we provide a brief review of 3PL supply chain. Then, we provide the attacking scenario of adversaries in 3PL supply chain and formulate the security and privacy requirements for preventing these attacks.

2.1 3PL Supply Chain

We depict the model of 3PL supply chains in Figure 1. A 3PL supply chain contains three parties. We denote the sender (customer A) who entrusts the transportation of goods to a 3PL provider as Party A, the 3PL provider as Party C and the receiver (customer B) of the goods as Party B[2]. The procedures in 3PL supply chains contain three steps:

1. *Ownership transfers from Party A to Party C*: Party A transfers the goods to Party C after the three parties have reached an agreement of the transaction.
2. *Party C' transports the goods*: Party C takes over the goods, and guarantees the goods' security during the transportation.
3. *Ownership transfers from Party C to Party B*: Party B verifies the goods, accepts them if the goods are intact, or denies the goods if the goods are not satisfactory.

A successful transaction is finished after Party B accepts the goods. Traditionally, when a party transfers goods to another party, the originality and the quantity of the goods are checked manually. However, when RFID system is deployed to enhance the efficiency of the supply chain, automatic identification replaces the manually checking. In RFID-enabled supply chains, the existence of the tags indicates the existence of the original goods[3].

2.2 Attacking Scenario of Adversaries

Different with the general adversary model which only considers the external adversaries, our adversary model for 3PL supply chain also considers the internal adversaries as well as the external adversaries. We analyze the potential dishonest behaviors of the three parties and the disputes that may happen on the ownership transfer between Party C and Party B as below. Note that we

[2] Party A and Party B can be the same entity in some occasions, eg. a factory entrusts a 3PL provider to transmit a batch of goods to its branch plant.

[3] Suppose each tag is imbedded in or stick on one item and it is hard to separate the tag from the item.

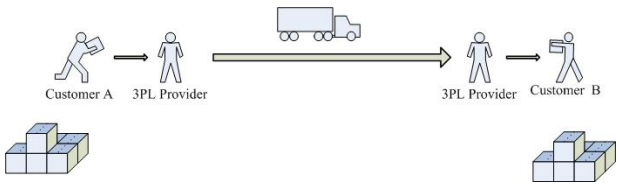

Fig. 1. 3PL Supply Chain Model

suppose ownership transfer from Party A to Party C is free of disputes. The reason is that a transaction will not begin unless Party A and Party C get into an agreement.

- *In case Party A is dishonest:* Party A sends a batch of low quality goods which do not satisfy Party B's requirements. When Party B refuses to accept the goods, Party A may claim that the goods are not original ones that it delivered, they have been replaced by Party C.
- *In case Party C is dishonest:* In case Party C loses or damages some goods during the transportation, to escape the compensation[4], Party C then fabricates the tags of the lost goods, and attaches them on fake goods. When Party B detects the replacements, Party C may claim that the faked goods came from Party A.
- *In case Party B is dishonest:* The dishonest Party B may intentionally refuse to accept the goods by claiming that goods do not satisfy the requirements.

2.3 Security and Privacy Requirements

Against internal adversaries. The major work for RFID system in 3PL supply chain is to facilitate Party C to transfer goods from Party A to Party B. The system should be able to detect Party C's malicious behavior. Inherently, we cannot prevent Party A (Party B) from cheating Party B (Party A), however, at least we should keep Party C away from Party A (Party B)'s malicious behavior. The requirements of the RFID system for 3PL supply chains against internal adversaries are listed as below:

- *Restrain dishonest Party C:* Party C should not be able to replace any goods without being detected. About privacy, in 3PL supply chain, Party A and Party B may not want to leak the goods' information to Party C. While the tags will be under Party C's control, the system should protect the tags' information leakage against Party C.
- *Protecting honest Party C:* If Party C honestly and successfully transfers the goods to Party B, Party B should accept the goods unconditionally, even if the goods do not meet the requirements, Party B should accept the goods (then Party B negotiates with Party A without involving Party C).

[4] Even worse, Party C replaces some goods and steals the original ones.

Against external adversaries. We assume that external adversaries only conduct the attacks during the transportation of goods.[5] The privacy and security requirements against external adversaries are listed as below.

- *Tag information privacy:* It means that external adversaries cannot get the information of the tags.
- *Tag location privacy:* If the responses of a tag are linkable to each other or distinguishable from those of other tags, then the location of a tag could be tracked by multiple collaborating tag readers. Tag location privacy means no one except the legitimate party can trace the tags.
- *Resistance of tag impersonation attack:* It means that the attacker impersonates a target tag without knowing the tag internal secrets and pass the authentication of the reader.
- *Resistance of replay attack:* It means that the attacker reuses communications from previous sessions to perform a successful authentication between a tag and a server.

2.4 Designing Principle

Adopting "symmetric secret"-based RFID systems in 3PL supply chains requires the three parties to share the secrets. Considering the existence of internal adversaries, sharing the secrets among three parties is problematic since having a tag's secret means having the ability to fabricate it. If all the three parties have the ability to fabricate the tags, disputes on the originality of the tags is difficult to solve.

Our method is to authorize each valid party with a credential instead of the secrets. The credential can be used by the authorized parties to check the status of the goods. At the same time the credential should not reveal any information about the tags. In a 3PL supply chain, Party A is the tags' owner. Only Party A possesses the tags' secrets. Party A grants a credential $credential_C$ to Party C so that Party C can check the tags' existing during the transportation. Party A grants a credential $credential_B$ to Party B, so that Party B can use it to verifies the goods. The construction and using is a subtle work.

Recalled that there are two requirements required against internal adversaries, namely, restraining Party C and protecting Party C. Restricting Party C requires that with $credential_C$, Party C cannot get any information of the tags. Protecting Party C requires that Party C should be able to confirm that the tags will pass the verification according to $credential_B$ before taking over the goods from Party A. Hence a systematic scheme should be designed for the three parties to make agreement on the credentials. And considering outside adversaries, the system should make sure that external adversaries cannot forge the tags that pass the verification according to the credentials.

[5] The two ownership transfer happens in relative secure environments that under two parties' surveillance.

The system will contain two parts: 1) designing a protocol that enable a authorized party to verify the tags with a credential; 2) designing a scheme that enables the three parties to make an agreement on the credentials. In this paper, we tackle the first part. We observe that in 3PL supply chains, normally, the goods are checked on batch level. In the following, we provide two group checking protocols to enable an authorized party verify the tags according to a credential on batch level based on two different credential designing schemes. While we focus on the originality checking of the tags, we suppose the number of the tags is stable, namely, if a malicious party take away one tag, he will replace it with a fake tag.

3 Solution Based on Aggregate MAC Scheme

Our first solution is based on an aggregate MAC scheme proposed in [7]. The intuition of this proposal is: each tag T_i (with k_i as its individual secret) is deployed with a MAC function. The authorized reader is granted with a credential that contains several couples of m_j and the aggregate MAC value $Agg(m_j)$ on m_j under each tag's key, where $1 \leq j \leq d$, d is the number of the pairs. The reader chooses an unused pair $(m, Agg(m))$, and uses m to query all the tags, each tag replies with the MAC value on m under its key. Upon receiving all tags' replies, the reader aggregates them and compares the aggregated value with $Agg(m)$, if they are the same, then the tags are intact, else, there are not all original ones.

3.1 Building Blocks of Our MAC-Based Solution

MAC: In cryptography, a message authentication code (MAC) is a short piece of information used to authenticate a message. A MAC algorithm, sometimes called a keyed (cryptographic) hash function $h_k(\cdot)$, it accepts the inputs as a secret key k and an arbitrary-length message m to be authenticated, and outputs a MAC tag $t = h_k(m)$ (sometimes known as a tag). The MAC value protects both the message's data integrity and its authenticity by allowing verifiers (who also possess the secret key) to detect any changes to the message content.

Aggregate MAC: In [7], Katz and Lindell proposed and investigated the notion of aggregate message authentication codes (MACs) which has the property that multiple MAC tags, computed by (possibly) different senders on multiple (possibly different) messages, can be aggregated into a shorter tag that can still be verified by a recipient who shares a distinct key with each sender. The aggregation is done by computing XOR of all the individual MAC tags. They proved that if the underlying MAC scheme is existentially unforgeable under an adaptive chosen-message attack and is deterministic, then the aggregate message authentication code generated by computing the XOR of every individual MAC values is secure.

3.2 Aggregate MAC-Based Solution

Requirements of the tags: The tags should be able to perform a MAC function $h_k(\cdot)$ under a key k stored in the tag. The tag should contain a random string generater.

Initialization: Party A initializes the tags. Suppose there are n tags in the system. Each tag T_i stores two secrets $(b,\ k_i)$, $1 \leq i \leq n$, b is a common group secret that is shared by all the tags, and k_i is the tag T_i's individual secret.

Authorization to a valid party: Party A keeps k_i secret, and grants to a valid party the group secret b and a credential. d denotes the estimated upper bound of the number of times that the party will check the goods. The credential contains d pairs of $(m_j, Agg(m_j))$, $Agg(m_j) = \bigoplus_{i=1}^{n} h_{k_i}(m_j)$, for $1 \leq j \leq d$ and $1 \leq i \leq n$.

Group checking protocol for an authorized party: For each checking, the authorized party chooses an unused pair $(m_j, Agg(m_j))$ from the credential, then uses m_j to query all the tags. The details are depicted as below. Figure 2 illustrates the protocol.

1. *Reader* $\rightarrow$ *Tag* T_i*:* The reader sends m_j to the tag T_i.
2. *Tag* T_i $\rightarrow$ *Reader:* On receiving m_j, T_i chooses $r_2 \in_R \{0,1\}^l$, l is the system parameter. T_i computes $M_1 = h_{k_i}(m_j) \oplus h(r_2)$, $M_2 = b \oplus r_2$, then sends (M_1, M_2) to the reader.
3. *Reader:* On receiving (M_1, M_2), the reader computes $r_2 = M_2 \oplus b$, and computes $t_i(m_j) = M_1 \oplus h(r_2)$. The reader stores the value of $t_i(m_j)$.

– After getting all the values $t_i(m_j)$, the reader checks whether $\bigoplus_{i=1}^{n} t_i(m_j) = Agg(m_j)$. If $\bigoplus_{i=1}^{n} t_i(m_j) = Agg(m_j)$, then the reader confirms the existing of the tags. If $\bigoplus_{i=1}^{n} t_i(m_j) \neq Agg(m_j)$, then the tags are not all the original ones.

Party A gives credentials $credential_B$ and $credential_C$ respectively to Party B and Party C. Party B uses $credential_B$ to verify the goods when taking over

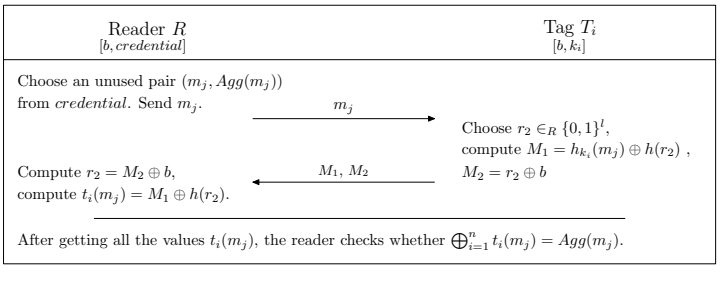

Fig. 2. Aggregate MAC-Based Solution

them from Party C. Party C uses $credential_C$ to verifying the tags during the transportation. In aggregate MAC-based scheme, Party B and Party C should keep its own credential in secret to each other, and $credential_B$ and $credential_C$ should not contain same $(m, Agg(m))$. Party C should make sure that the original tags will pass the verification using $credential_B$.

3.3 Analysis of the MAC-Based Solution

We first analyze the security and privacy properties of the MAC-based protocol against external adversaries.

- *Tag Information Privacy:* Without knowing b, the adversary cannot calculate the value of r_2. Without r_2, the adversary cannot get the value of $h_{k_i}(m_j)$. Then without $h_{k_i}(m_j)$, the adversary cannot get any information about k_i.
- *Tag location Privacy:* Without common secret b and individual secret of each tag, due to the cryptographic property of the MAC function $h_k(\cdot)$, the adversary cannot get any information of the tags through (M_1, M_2).
- *Resistance of tag impersonation attack:* Given (M_1, M_2), the adversary cannot retrieve any information of b and k_i. Without the knowledge of b and k_i, the adversary cannot retrieve any information about the tag. The probability that the adversary successfully impersonate a tag is equal to the probability that the adversary randomly chooses (M_1', M_2') and then (M_1', M_2') together with other valid tags' replies pass the verification.
- *Resistance replay attack:* The authenticated reader uses fresh pair of $(m_j, Agg(m_j))$ to verify the tags in each checking, so that the attacker cannot reuse the tags' replies from previous sessions.

Then we analyze the security and privacy properties of the MAC-based protocol against internal adversaries.

- *Protect Party C:* For Party C, given a tag T_i, with the common secret b, it can challenge the tag with arbitrary message m and get the MAC tag $t_i(m) = h^{k_i}(m)$ on m with T_i's secret k_i, however it cannot compute the value of k_i if the underlying MAC scheme is secure. While without the knowledge of k_i, based on the security of the aggregate MAC scheme, it is computational impossible to forge the tags so that given another message m', the aggregation of the tags' replies $\bigoplus_{i=1}^{n} t_i(m')$ equals $Agg(m') = \bigoplus_{i=1}^{n} h_{k_i}(m')$. Hence, if the tags pass the verification using Party B's credential, then Party B cannot deny that the tags are original ones.
- *Restrain Party C:* Providing some pairs of $(m, Agg(m))$, together with group secret s, Party C can verify the tags on batch level. However, given a tag T_i, although Party C can get the value $h_{k_i}(m)$ on any message m, it cannot compute the value k_i. Hence Party C cannot obtain any extra information of the tag.

4 Solution Based on Aggregate Signature Scheme

Another solution is based on aggregate signature scheme proposed in [1]. Party A authorizes the valid party with a credential that contains a value V. Each tag T_i is considered as a signature function with key k_i. Upon receiving a query m, tag T_i replies with the signature $\sigma_i(m)$ on m under its key. Then the reader aggregates the individual signatures $\sigma_i(m)$ for $1 \leq i \leq n$, verifies the aggregate signature using the value V.

4.1 Building Blocks of Our Aggregated Signature-Based Scheme

Bilinear Map: A bilinear map is a map $e : G_1 \times G_2 \rightarrow G_T$, where: (a) G_1 and G_2 are two (multiplicative) cyclic groups of prime order q; (b) $|G_1| = |G_2| = |G_T|$[6]; (c) g_1 is a generator of G_1 and g_2 is a generator of G_2. The bilinear map $e : G_1 \times G_2 \rightarrow G_T$ satisfies the following properties: (a) Bilinear: for all $x \in G_1, y \in G_2$ and $a, b \in \mathbb{Z}$, $e(x^a, y^b) = e(x, y)^{ab}$; (b) Non-degenerate: $e(g_1, g_2) \neq 1$.

Short Signature Scheme: Boneh, Lynh, and Shacham proposed the short signature scheme in [2] using the bilinear map. The system contains two groups G_1 and G_2 with prime order q, a full-domain hash function $H(\cdot) : \{0,1\}^* \rightarrow G_1$, and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. g is a generator of G. Each singer has public key $X = g^x$, where $x \in \mathbb{Z}_q$ is the corresponding private key. Signing a message M involves computing the message hash $h = H(M)$ and then the signature $\sigma = h^x$. To verify a signature one computes $h = H(M)$ and checks that $e(\sigma, g) = e(h, X)$.

Aggregate Signature Scheme: Aggregate signature scheme aggregates n signatures on n distinct messages from n distinct users to one signature. Any one should be able to do the aggregation without knowing the users' keys. Boneh and Gentry proposed a scheme [1] to aggregate BLS signatures. Given n individual signatures, one computes the aggregate signature as follows: $\sigma_{1,n} = \prod_{i=1}^{n} \sigma_i$, for $1 \leq i \leq n$, where σ_i corresponds to the user $user_i$'s signature on message M_i. Verification of an aggregate BLS signature $\sigma_{1,n}$ includes computing the product of all message hashes and verifying the following match: $e(\sigma_{1,n}, g) \stackrel{?}{=} \Pi_{i=1}^{n} e(h_i, X_i)$ where X_i is the public key of the signer who generates σ_i on message M_i.

ElGamal encryption scheme: ElGamal encryption system [4] is an public key encryption scheme based on the Diffie-Hellman problem. The scheme firstly chooses a multiplicative cyclic group G of order q with generator g. Each user $user_i$ chooses $x_i \in_R \mathbb{Z}_q$, sets x_i as the private key, then computes his public key $X_i = g^{x_i}$. To encrypt a message m to $user_i$, the sender converts his secret message m into an element m' of G, then chooses $r \in_R \mathbb{Z}_q$, computes $c_1 = g^r$ and $c_2 = m' \cdot X_i^r$, and then sends (c_1, c_2) to $user_i$. To decrypt the ciphertext (c_1, c_2), $user_i$ calculates $m' = c_2 \cdot s^{-1}$ which she then converts back into the plaintext message m.

[6] G_1 and G_2 can be the same group.

4.2 Basic Aggregate Signature-Based Solution

Requirements of the tags: The tags should be able to perform multiplication and addition on a multiplicative cyclic groups G_1 of prime order q. Each tag stores a secret.

Initialization: Suppose there are n tags in a batch, each tag is denoted as T_i, where $1 \leq i \leq n$. Let G_1, G_2 be cyclic groups of the order q. Then Party A chooses a bilinear map: $e : G_1 \times G_1 \rightarrow G_2$. For each tag T_i , Party A chooses a value $k_i \in_R \mathbb{Z}_q$ as the tag's individual secret. k_i is T_i 's individual secret. Then Party A generates the credentials. A credential contains a value V. V is computed to satisfy the following equation:

$$V = g^{\Sigma_{i=1}^{n} k_i} \tag{1}$$

where g is a generator of G_1.

Authorization to valid party: Party A keeps the tags' secrets and grants a credential to a valid party as well as the system parameters.

Group checking protocol for authorized party: The details of the protocol are shown below. Figure 3 depicts this solution.

1. *Reader* $\rightarrow$ *Tag* T_i: The reader firstly chooses a random number $r_i \in_R \mathbb{Z}_q$, computes $m_i = g^{r_i}$, sends m_i to the tag.
2. *Tag* $T_i \rightarrow$ *Reader*: After receiving m_i, T_i computes $\sigma_i = m_i^{k_i}$, then sends σ_i to the reader. For each tag T_i, the reader records the reply σ_i.

– After getting all the tags's replies σ_i, the reader checks whether $e(\prod_{i=1}^{n} \sigma_i, g) = e(g^{\sum_{i=1}^{n} r_i}, V)$. If the equation holds, the reader confirms that the tags are the original ones; else, the tags are not all original.

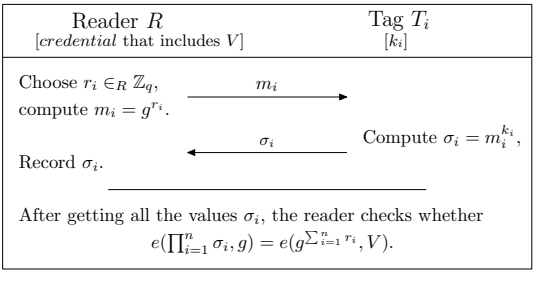

Fig. 3. Basic Aggregate Signature-Based Solution

4.3 Advanced Aggregate Signature-Based Solution

The basic aggregate signature-based scheme guarantees that the authorized party can check the tags without the secrets in batch level. However, it does

not provide location privacy since a tag sends same reply to the same challenge in different sessions. To get the anti-tracing property, we randomize the tag's reply by using the ElGamal encryption scheme.

Requirements of the tags: Additional to the requirements in basic scheme, teach tag T_i stores a copy of a public key S, $S = g^s$, where s is the corresponding private key in advance.

Initialization: The same as the basic scheme except that the credential contains another value s.

Authorization to valid party: The same as the basic scheme.

Group checking protocol for authorized party: The details of the advanced aggregate signature based scheme are shown below. Figure 4 illustrates the proposal.

1. *Reader* $\rightarrow$ *Tag* T_i: The reader firstly chooses a random number $r_i \in_R \mathbb{Z}_q$, computes $m_i = g^{r_i}$, sends m_i to query T_i.
2. *Tag* T_i $\rightarrow$ *Reader*: After receiving m_i, T_i computes $\sigma_i = m_i^{k_i}$. Then T_i generates a random number $r_2 \in_R \mathbb{Z}_q$, computes $M_1 = \sigma_i \cdot S^{r_2}$, $M_2 = g^{r_2}$, namely T_i encrypts σ using the ElGamal encryption scheme. T_i sends (M_1, M_2) to the reader finally.
3. *Reader*: Receiving (M_1, M_2), the reader decrypts M_1, M_2, gets $v_i = M_1/M_2^s$. The reader records the value of σ_i.

– After getting all the tags's replies σ_i, the reader checks whether $e(\prod_{i=1}^{n} \sigma_i, g) = e(g^{\sum_{i=1}^{n} r_i}, V)$. If the equation holds, the reader confirms that the tags are the original ones, else the tags have been replaced.

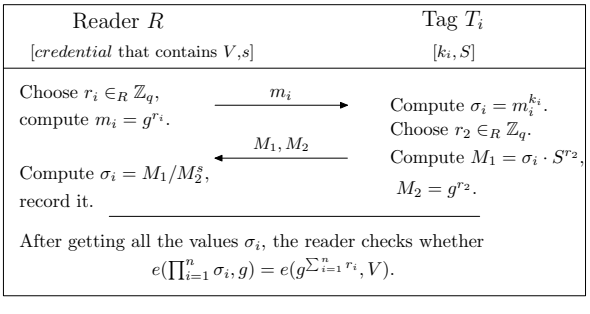

Fig. 4. Advanced Aggregate Signature-Based Solution

Note that different with the aggregate MAC-Based scheme that each checking consumes a pair of $(m, Agg(m))$, in aggregate signature-based scheme, the value V is reusable. Given V, one cannot forge the tags. Hence, credentials for different parties include the same value V.

4.4 Analysis of the Aggregated Signature Based Solution

We first analyze the security and privacy properties of the advanced aggregate signature-based protocol against external adversaries.

- *Tag Information Privacy:* The security of ElGamal encryption scheme guarantees that only the authorized reader with s can decrypt the message (M_1, M_2), then get σ_i. For the authorized reader, with m_i and $\sigma_i(m_i)$, it is computational impossible for one to compute the value of k_i based on the hardness of Discrete Logarithm Problem. Hence our system guarantees tag information privacy.
- *Tag location Privacy:* Our system provides location privacy to external adversaries. Without knowing the secret s, the external adversaries cannot distinguish the two tags because ElGamal encryption introduces randomization.
- *Resistance of tag impersonation attack:* Based on the secure of the aggregate signature scheme, without the knowledge of the tags' keys, it is computational impossible for an adversary to forge the tags that pass the verification using the credential, even with the authorized parties' public key S.
- *Resistance of replay attack:* The authenticated reader uses fresh message m_i to query each tag. Hence, one cannot reuse the reply (M_1, M_2) that contains m_i's information as the response to another query m_i'.

Then we analyze the security and privacy properties of the aggregate signature-based protocol against internal adversaries.

- *Protect Party C:* The security of the aggregate signature scheme guarantees that without the knowledge of tags' secrets, Party C cannot forge the tags T_i' for $1 \leq i \leq n$ that satisfies $e(\prod_{i'=1}^{n} \sigma_i', g) = e(g^{\sum_{i=1}^{n} r_i}, V)$, where σ_i' corresponds to tag T_i''s signature on message m_i'. Hence if the tags pass the verification using V, no one can claim that Party C has replaced the tags.
- *Restrain Party C:* Since our system achieves the tag information private property and resist tag impersonation attack against the adversaries that do not know the tags' secret, Party C cannot gather any extra information of the tags and replace any of the tags.

5 Discussions

We provide two solutions to implement the group checking for the 3PL supply chains. One is aggregate MAC (AMAC)-based, the other is aggregate signature(AS)-based. As analyzed in Section 3 and Section 4, both the two proposals meet the requirements listed in Section 2, they achieve the same security and privacy level. In this section, we compare the two schemes' performances in Table 1 and their usability in Table 2. We realize the aggregate signature-based scheme on Elliptic Curves, more details on implementing Elliptic Curve Cryptography(ECC) on RFID chips can be found in [5]. Hence in aggregate

signature-based solutions, operation $\cdot$ denotes point addition , operation / denotes point subtraction, exponential operations denotes point multiplication.

In the following tables, n denotes the number of tags in a batch. d denotes the number of $(m, Agg(m))$ pairs in a credential. We ignore the cheap operations $\oplus$, addition, subtraction and comparison of two values on calculating the computation consumptions on the reader side.

Table 1. Comparisons of the Two Schemes on Computation Performance

	AMAC-based solution	AS-based solution
Generation of a credential	$n \cdot d$ hash operations	1 point multiplication
Computations required on tag (running the protocol)	2 hash operations	3 point multiplications 1 point addition
Computations required on reader (running the protocol)	1 hash operation	2 point multiplications 1 point subtraction
Computations required on reader (Aggregation and verification)	none	2 paring operations 1 point multiplication

Table 2. Comparisons of the Two Schemes on Usability

	AMAC-based solution	AS-based solution
Computation capability (tag)	hash function, random number generator	operations on elliptic curve, random number generator
Storage requirements (tag)	2 values	2 values
Length of the credential	$O(d)$	$O(1)$
Restrictions on query	only allow to use a same pre-fixed value to query a batch of tags	allow to use arbitrary value to query each tag
Systematical support required	a scheme enables Party C to verify the validity of $credential_B$ without knowing the contents of $credential_B$	none

From above comparisons, we can find that the aggregate signature-based scheme is better compared to the aggregate MAC-based scheme. The reader can use arbitrary challenges to query the tags, while in aggregate MAC-based scheme, the reader should use a same pre-fixed value to query the whole batch of tags. The length of a credential is constant in the aggregate signature-based scheme while in aggregated MAC-based scheme, the length of the credential relates to the number of checking granted to a party. In aggregate signature-based scheme, $credential_B$ and $credential_C$ share the same value V, while in aggregate MAC-based scheme, $credential_B$ and $credential_C$ should not contain same $(m, Agg(m))$ pairs and additional scheme is required to convince Party C the validity of $credential_B$ without knowing the contents of $credential_B$.

Although aggregate signature-based scheme is more elegant, the aggregate MAC-based scheme overall takes more advantage since it requires much cheaper tag and performs more efficiently in running the protocol. Although the

additional required systematical support will be counted on the reader side, since the efficiency bottleneck of the system is on the tag side, hence the aggregate MAC-based solution is more suitable for supply chains application.

6 Related Works

There is a concept called "grouping proof" proposed by Juels [6] which is similar with our "group checking". The pharmaceutical distribution example is used to illustrate how grouping proof protocols work. Yoking-proof would provide an evidence that each container of the medication was dispensed with a leaflet in case that a tag is embedded in the container and another tag is embedded in an accompanying leaflet. Yoking proof only enables two tags to prove their co-existence and is vulnerable to replay attack. Later works on grouping proof [13,12,3,8] support multiple tags and putting their efforts on improving the security and efficiency.

Both in the "grouping proof" scenario and "group checking" scenario, the readers are not trusted, they do not hold the secrets of the tags. In "grouping proof" scenario, the reader aims to prove to a verifier that the tags are processed together, in case the reader have the secrets, he can forge a proof using the keys. In "group checking" scenario, the reader should prove the tags' originality to another party, in case he has the secrets, he can forge the tags. Besides reading the tags in batch level without knowing the secrets. The two kinds of schemes work differently. In "group proof" scenario, the reader does not need to authenticate the tags. The reader only acts as a transfer stop in the grouping proof protocols in transmitting the messages among the tags. The whole tags generate a proof. While in "group checking" even without getting the secrets of the tags, the reader should have the ability to check the integrity and originality of a batch of tags. The reader interacts with the tags and verifies the tags' originality.

7 Conclusions

With the considerations of the internal adversaries as well as external adversaries, we analyzed the security and privacy requirements of RFID system for 3PL supply chains. We provided two "group checking" protocols to enable a reader to check the tags' existences and originality in batch level without knowing the secrets of the tags. One protocol is based on aggregate MAC scheme and the other is based on aggregate signature scheme. Both of the two protocols achieve the goals of protecting Party C and restraining Party C, and provide security and privacy guarantees. We compare the usability and performance of the two schemes, we can see that the aggregate MAC-based protocol outperforms the aggregate signature-based protocol. In the future, we will design a protocol that enables Party C to verify the validity of $credential_B$ without knowing the contents of $credential_B$. Then the system will achieve clear ownership transfers among the three parties.

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

References

1. Boneh, D., Gentry, C.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003)
2. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. J. Cryptology 17(4), 297–319 (2004)
3. Burmester, M., de Medeiros, B., Motta, R.: Provably secure grouping-proofs for rfid tags. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 176–190. Springer, Heidelberg (2008)
4. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
5. Hein, D., Wolkerstorfer, J., Felber, N.: ECC is Ready for RFID A Proof in Silicon. In: RFIDSec 2008, Budapest, Hungary (July 2008)
6. Juels, A.: "Yoking-Proofs" for RFID Tags. In: Sandhu, R., Thomas, R. (eds.) PerSec 2004, Orlando, Florida, USA, pp. 138–143. IEEE Computer Society, Los Alamitos (March 2004)
7. Katz, J., Lindell, A.Y.: Aggregate message authentication codes. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 155–169. Springer, Heidelberg (2008)
8. Lin, C.C., Lai, Y.C., Tygar, J.D., Yang, C.K., Chiang, C.L.: Coexistence proof using chain of timestamps for multiple RFID tags. In: Chang, K.C.-C., Wang, W., Chen, L., Ellis, C.A., Hsu, C.-H., Tsoi, A.C., Wang, H. (eds.) APWeb/WAIM 2007. LNCS, vol. 4537, pp. 634–643. Springer, Heidelberg (2007)
9. 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)
10. Molnar, D., Wagner, D.: Privacy and Security in Library RFID: Issues, Practices, and Architectures. In: Pfitzmann, B., Liu, P. (eds.) CCS 2004, Washington, DC, USA, pp. 210–219. ACM Press, New York (October 2004)
11. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to "Privacy-Friendly" Tags. In: RFID Privacy Workshop. MIT, Massachusetts (November 2003)
12. Piramuthu: On existence proofs for multiple rfid tags. In: PERSER 2006, Washington, DC, USA, pp. 317–320. IEEE Computer Society, Los Alamitos (2006)
13. Saito, J., Sakurai, K.: Grouping proof for rfid tags. In: AINA 2005, Washington, DC, USA, pp. 621–624. IEEE Computer Society, Los Alamitos (2005)
14. Song, B.: RFID Tag Ownership Transfer. In: RFIDsec 2008, Budaperst, Hungary (July 2008)
15. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: 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. 201–212. Springer, Heidelberg (2004)

Author Index