applied sciences

MDPI

Article

Novel Designated Ownership Transfer with Grouping Proof

Kuo-Yu Tsai [1,†], Ming Hour Yang [2,*,†], Jia Ning Luo [3,†] and Wei-Tim Liew [2,†]

1 Applied Mathematics, Chinese Culture University, Hwa-Kang Rd., Yang-Ming-Shan, Taipei City 11114, Taiwan; cgy13@ulive.pccu.edu.tw
2 Information and Computer Engineering, Chung Yuan Christian University, Chung Pei Road, Chung Li Dist., Taoyuan City 32023, Taiwan; kelvinli3w@gmail.com
3 Information and Telecommunications Engineering, Ming Chuan University, De Ming Rd., Gui Shan Dist., Taoyuan City 33348, Taiwan; deer@mail.mcu.edu.tw
* Correspondence: mhyang@cycu.edu.tw; Tel.: +886-3-265-4061 (ext. 4733)
† These authors contributed equally to this work.

Received: 20 December 2018 ; Accepted: 13 February 2019 ; Published: 19 February 2019

Abstract: In the supply chain management literature, various mobile radio frequency identification (RFID) protocols have been proposed for minimizing cargo theft during transport while ensuring the integrity of the entire cargo load or transferring ownership of a tagged item to another owner. These protocols are generally called grouping proof protocols and ownership transfer protocols, respectively. However, no protocol has been proposed that can achieve both requirements. In this paper, we propose a novel designated ownership transfer with grouping proof protocol that simultaneously generates grouping proofs and authenticates the consistency between the receipt proof and pick proof while ensuring that ownership of the cargo is transferred to the new designated owner in one attempt. In addition, the proposed scheme is robust against attacks (such as replay, denial-of-service, and denial-of-proof attacks) and has security features, such as forward/backward secrecy and message integrity.

Keywords: mobile RFID; supply chain management; multilayered grouping proof; ownership transfer; denial of proof; message integrity

1. Introduction

In supply chain management (SCM), mobile radio frequency identification (RFID) has recently been widely and rapidly adopted for tracking and identifying objects. An RFID system consists of mobile readers, a back-end server that acts as a trusted third party (TTP), and tags that can be further classified into two types: active and passive. An active tag usually contains an internal power source (typically a battery) to continuously power it, thus giving it a long reading range, whereas a passive tag relies on the radio frequency energy transmitted from the reader as its power source, resulting in a shorter reading range. In SCM, passive tags are more widely used than active tags because of their lower implementation cost. In the early stages of SCM, tags were employed to store information about cargo and facilitate automated stocktaking. However, with this setup, in the later stages of SCM, increasing challenges led to a rise in cargo theft [1]. The key task under such circumstances is to identify the suspect among the supplier, transporter, and recipient based on evidence. To solve this problem, a protocol is needed to generate an undeniable proof of cargo. In 2004, Juels [2] introduced the yoking-proof protocol that proves the existence of two tags within the range of an RFID reader. However, Saito and Sakurai [3] proved that this protocol is insecure, and they extended the concept into the grouping proof. Saito and Sakurai's proposed protocol

allows a single RFID reader to simultaneously prove the existence of a group of tags. The generated proof is later sent to a TTP for further verification. Based on Elliptic Curve Cryptography (ECC), Batina et al. [4] proposed a privacy-preserving multiparty grouping-proof protocol in the setting of a narrow-strong attacker. Batina et al.'s proposed protocol allows generating a proof that is verifiable by a trusted verifier in an offline setting, even though readers or tags are potentially untrusted. However, Hermes and Peeters [5] demonstrated that an adversary can generate a valid grouping proof in Batina et al.'s protocol [4]. In addition, Hermes and Peeters introduced two formal models for yoking proofs. Based on the proposed models, they further proposed two protocols to generate sound yoking proofs. Shen et al. [6] proposed a lightweight RFID grouping authentication protocol, in which one object to be authenticated is attached with a group of RFID tags. If only some of the tags are successfully authenticated, the generated proof can be used to guarantee that the object is here and trace the identities of the disabled tags. In 2016, Abughazalah et al. [7] proposed an offline two-round grouping-proof protocol. Abughazalah et al.'s proposed protocol improves tag's memory and computing performance. Burmester and Munilla [8] presented an anonymous grouping-proof with untrusted readers, in which the generated proof can be checked by the trusted verifier and the untrusted reader can recover the identifiers for missing tags, but the untrusted reader cannot generate a proof if tags are missing. In the same year, Burmester and Munilla [9] extended earlier work on grouping proofs and group codes to capture resilient group scanning with untrusted readers. Rostampour et al. [10] adopted authenticated encryption to design a scalable grouping proof protocol with message authentication code, which provides both confidentiality and message integrity simultaneously. Each tag individually computes authenticated message, and the reader is responsible for gathering the response messages of all tags in the authentication phase. That is, Rostampour et al.'s protocol eliminates the dependency among the tags' responses. Based on parallel mode and dynamic host configuration protocol, Shi et al. [11] proposed a lightweight RFID grouping-proof protocol that adopts parallel communication mode between reader and tags. It achieves grouping-proof efficiency.

When cargo is delivered to a new party, the ownership of that cargo should be transferred simultaneously. Most ownership transfer protocols operate based on the following assumptions. After the ownership transfer is complete, the former owner can no longer access the RFID tags, and the new owner can prove ownership of cargo by means of mutual authentication with the RFID tags [12–17]. Yang and Xie [17] proposed a RFID protocol for group ownership transfer, in which a group of tags' ownership can be transferred in one attempt. Later, Li et al. [13] presented a physical-unclonable-function-based RFID ownership transfer protocol in an open environment. A physical unclonable function is used to to prevent that the tag is cloned, and a reader does not need to store the response values in Li et al.'s protocol. According to the EPC Class-1 Generation-2 Version 2 standard, Niu et al. [15] proposed an ultra-lightweight authentication and ownership management protocol, which reduces the storage and the computational costs of the tags. In 2015, Li et al. [14] presented lightweight authentication and delegation protocol for RFID tags, which also adopts physically unclonable function to enhance the security of the tags. Li et al.'s protocol can achieve the security requirement of privacy for the original owner and the new owner. Based on the XOR operation and a 128-bit pseudo-random number generator, Sundaresan et al. [16] proposed a ownership transfer protocol for a multi-tag multi-owner RFID environment, in which it protects privacy for individual-owner and prevents tracking attacks. However, Munilla et al. [18] mounted some attacks on Sundaresan et al.'s protocol [16]. In the demonstrated attacks, an adversary can trace a tag, and the previous owner can obtain the private information shared among the tag and the new owner. By using the ownership transfer protocol, both parties can trade tagged objects more easily. A few researchers have worked to reduce security threats in ownership transfer, including replay [3] and denial-of-service (DoS) attacks [19].

A ownership handover process between a supplier or retailer and a recipient requires two protocols, namely, a grouping proof protocol to ensure all cargo is in place and an ownership transfer protocol to transfer ownership of cargo to the designated party. This process is inefficient and

time consuming. Owing to an increasing number of security threats in SCM, there should be a complete RFID protocol that preserves message integrity and privacy. Consider two scenarios in which a malicious party might intercept messages transmitted between tags and mobile readers. In the first scenario, the malicious party might retransmit the messages to execute unauthorized operations, such as by generating bogus proofs or achieving fake authentication. In the second scenario, the malicious new owner might attempt to gain access to the previous transactions of the former owner or the former owner might try to access the future transactions of the new owner. To solve these problems, the grouping proof and ownership transfer protocols must be combined. A different scenario would involve certain products needing to be shipped in groups—for example, safety regulations that require medication to be shipped along with the corresponding information leaflets [20]. Ownership transfer guarantees that the medication and the leaflets originate from the respective owners within a drug manufacturer, and grouping proof ensures that each of the medications is imported along with the correct leaflets. Scenarios in which medication is delivered without the correct leaflets or the leaflets are delivered with counterfeit medication are unacceptable, because they might cause a patient's death due to medication-related errors [21]. To prevent such a scenario from occurring, grouping proof is required for a valid group ownership transfer. To that end, in 2010, Zuo [22] proposed a protocol based on ownership transfer with the capability to integrate any grouping proof protocols. Although Zuo's protocol can ensure the simultaneous presence of a group of tags during ownership transfer, it suffers from performance problems. For example, it requires mutual authentication to be performed twice.

In this paper, we propose a novel designated ownership transfer protocol with grouping proof that can ensure message integrity and privacy. Although the ownership transfer and grouping proof protocols are designed for different purposes, they have similarities, such as requiring authentication before information is exchanged and generating random numbers to derive fresh messages. By combining the similarities between the protocols, the proposed protocol not only retains the main security and privacy features of the grouping proof and the ownership transfer protocols but also reduces the number of message or responses needed, because ownership transfer and proof generation are performed in one attempt.

The remainder of this paper is organized as follows. The proposed protocol is described in Section 2, and a security analysis of the proposed protocol is given in Section 3. Finally, we provide security analysis and a conclusion in Sections 4 and 5, respectively.

2. Proposed Protocol

The proposed protocol is centered on the hierarchical-management-framework-based grouping proof protocol proposed by Yang et al. [23], which allows several readers to simultaneously scan a group of tags to generate pieces of proof that will later be combined into a final grouping proof by an authorized reader. The protocol consists of three phases: initialization, integrity verification, and ownership transfer. The notations used in the proposed protocol are listed in Table 1.

2.1. Initialization Phase

When a cargo shipment with a tag collection T^q is delivered to the recipient P^q, the reader requests the verifier to establish a secure multicast connection to ensure that the generated grouping proofs are in accordance with those on the recipient's reader and that the message can be transmitted to δq tags in T^q. Accordingly, the verifier generates a k-ary group key with a subtree height difference of $\leq 1 (h_{max} = \lceil \log_k (\delta q / k) \rceil)$ by using the secret key Kt_i^q, where Kt_i^q is shared between the verifier and the tags TID_i^q. In summary, the group keys that can transmit messages to δq tags in T^q are defined as GK_0^q. Figure 1 shows an example of a 3-ary group key (GK_0^q is the starting node) generated for a set of 23 tags; the group key GK_1^q is employed to encrypt the multicast messages transmitted to the tags numbered from TID_1^q to TID_9^q, and tags TID_1^q, TID_2^q, and TID_3^q can decrypt the multicast messages

encrypted with the group key GK_4^q by using their own shared keys Kt_1^q, Kt_2^q, and Kt_3^q, respectively. The! details of group key generation can be found in Yang and Xie's proposed methods [17].

Table 1. Symbol notations.

Notation	Description
A	a transporter who delivers cargo
T^q	a cargo shipment with a tag collection
P^q	the qth recipient who receives the cargo
AID	an identification code for A
RID_0	an identification code of the reader used by A
PID^q	an identification code of P^q
RID_j^q	an identification code of the jth reader used by P^q
TID_i^q	an identification code of the ith tag for P^q
TH_i^q	a hash value for verifying TID_i^q
Kr_j^q	a secret key shared between RID_j^q and V
Kt_i^q	a secret key shared between TID_i^q and V
Ky_i^q	a secret key shared between PID^q and TID_i^q
GK_s^q	a secret key shared between G_s^q and V
SK_j^q	a session key shared among readers
PK^a/PR^a	a public/private key pair for a
PK^q/PR^q	a public/private key pair P^q
Nr_j^q	a random number generated by RID_j^q
Nt_i^q	a random number generated by TID_i^q
Na	a random number generated by a
Np^q	a random number generated by P^q
TS_v	a timestamp generated by V
$E(key, Msg)$	an encryption function with two inputs: the message (Msg) and the symmetric key (key)
$Sign(key, Msg)$	a signing function with two inputs Msg and key
$MAC(key, Msg)$	a key-hashing function for generating message authentication codes, where the inputs are Msg and Key
$H(Msg)$	a hashing function with an input Msg
OT	a ownership transfer protocol

Moreover, a verification code TA^q is generated for δq tags inT^q according to Equation (1).

$$TA^q = TH_1^q \parallel \ldots \parallel TH_{\delta q}^q, \text{where } \forall \text{i } \mathrm{TH_i^q} = \mathrm{H(TID_i^q \parallel Kt_i^q \parallel TS)}. \quad (1)$$

After the verifier has generated the verification codes TA^q, group key GK_0^0, and timestamp TS_v, they are then transmitted to the transporter's reader to be forwarded to the n recipients' readers.

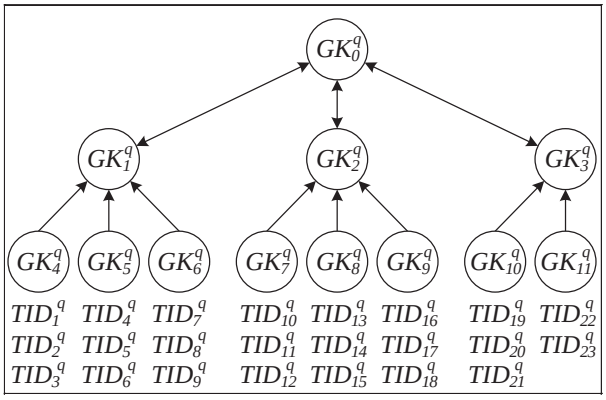

Figure 1. The 3-ary key tree for a group of tags.

2.2. Integrity Verification Phase

After the transporter delivers the cargo to the recipient P^q and simultaneously generates grouping proofs by using a reader with a maximum reading capacity of r, the group keys are distributed to several mobile readers from the transporters' reader RID_0 to securely multicast messages to δq tags via the recipients' readers RID_j^q, thus enabling each reader to receive the maximum number of tags by performing only one multicast. In other words, the grouping proof is generated using the minimum number of group keys.

In this phase, the reader RID_0 uses the distributed keys to encrypt the recipient's identification code PID^q, ownership transfer request OT, timestamp TS_v, group key set RG_j^q for the child node reader RID_j^q, and tag verification code set RT_j^q, and then transmits the ciphertext until all leaf nodes are reached. For example, as shown in Figure 2, the reader RID_0 first uses the key SK_1^q to encrypt PID^q, OT, TS_v, $RT_1^q = \{TH_{13}^q, TH_{14}^q, TH_{15}^q, TH_{16}^q\}$, and $RG_1^q = \{GK_8^q, GK_9^q\}$, and then transmits them to the reader RID_1^q, which will use the session key SK_1^q to decrypt the message, split RT_1^q and RG_1^q and, accordingly, encrypt PID^q, OT, TS_v, $RT_7^q = \{TH_{13}^q, TH_{14}^q\}$, $RG_7^q = \{GK_8^q\}$, PID^q, OT, TS_v, $RT_8^q = \{TH_{15}^q, TH_{16}^q\}$, and $RG_8^q = \{GK_9^q\}$ into separate messages by using the session keys SK_7^q and SK_8^q, and send them to the leaf node readers RID_7^q and RID_8^q.

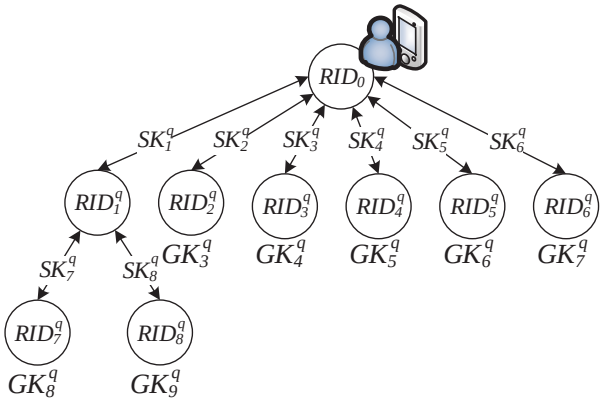

Figure 2. One group key.

Subsequently, the encrypted messages will be distributed to the corresponding tags. The leaf reader will then collect pieces of proof from the tags, which will be transmitted back to the upper levels and then to reader RID_0 to generate a grouping proof, as shown in Figure 3.

[Message 1] After receiving the encrypted message from RID_0, the reader RID_k^q proceeds to decrypt the message using the session key SK_j^q. Then, depending on the child node, RID_k^q splits RT_j^q, and RG_j^q. Furthermore, RID_k^q encrypts PID_k^q, OT, TS_v, TSC, RT_j^q, and RG_j^q by using the child node session key SK_j^q as message F_j then transmits to RID_j^q.

[Message 2] Upon receiving the encrypted message F_j from RID_k^q, the leaf node reader RID_j^q uses its session key SK_j^q to decrypt the message. The multicast message $MG_{j,s}^q$ is constructed, encrypted using the group key GK_s^q along with PID^q, OT, TS_v, and TSC, and then transmitted to the corresponding tags to generate pieces of proof.

[Message 3] When any tag TID_i^q receives a multicast message $MG_{j,s}^q$, the tag will proceed to decrypt the message by using the shared key Kt_i^q, and then verify whether the decrypted message contains the correct PID^q and ownership transfer request OT. When the verification is correct, the shared key Kt_i^q is then employed to compute the pieces of proof $M_{j,i}^q$ along with the tag TID_i^q, a randomly generated number Nt_i^q, and a timestamp (if offline then TSC else TS_v). Subsequently, a message verification code

$V_{j,i}^{q}$ is computed for the reader RID_j^q to verify the tag by using the hashing value $H(TID_i^q \| Kt_i^q \| TS_v)$, the shared key Kt_i^q, the timestamp TS_v, pieces of proof $M_{j,i}^q$, and a random number $Nt_{j,i}^q$.

[Message 4] To verify the message integrity, a leaf node reader RID_j^q receives a response messages from the tags, the obtained Nt_i^q, $M_{j,i}^q$, and tag verification value $TH_i^q = H(TID_i^q \| Kt_i^q \| TS_v)$ transmitted from RID_k^q previously, and further, the reader computes $V_{j,i}^q$. Through comparison with the message verification code $V_{j,i}^{q'}$ transmitted by the tags, the reader RID_j^q can block and prevent proof that is not associated with this delivery. Subsequently, the reader RID_j^q employs the XOR operation to combine all pieces of proof $M_{j,i}^q$ and the verification code $V_{j,i}^q$ into pieces of proof $M_{j,0}^q$ and message verification code $V_{j,0}^q$. The pieces of proof M_j^q generated by the reader are then computed using the shared key Kr_j^q along with the reader identification code RID_j^q, and randomly generated numbers Nr_j^q and $M_{j,0}^q$. Moreover, a message verification code V_j^q is also computed by hashing M_j^q, $V_{j,0}^q$ and Nr_j^q. The session key SK_j^q is used to encrypt PID^q, M_j^q, V_j^q, Nr_j^q, $M_{j,i}^q$, and Nt_i^q for all group member tags, and the encrypted messages are transmitted back to parent node reader RID_k^q. After the parent node reader RID_k^q receives the response message F_k transmitted by child node reader RID_j^q, the encrypted message is decrypted by using the session key SK_j^q to verify whether the message contains the same recipient PID^q. Once the recipient is authenticated, the reader RID_k^q uses the same method as the reader RID_j^q to generate the required message, then transmit the message $F_{\lfloor (k-1)/r \rfloor}$ to the reader at the upper level, and finally back to the reader RID_0.

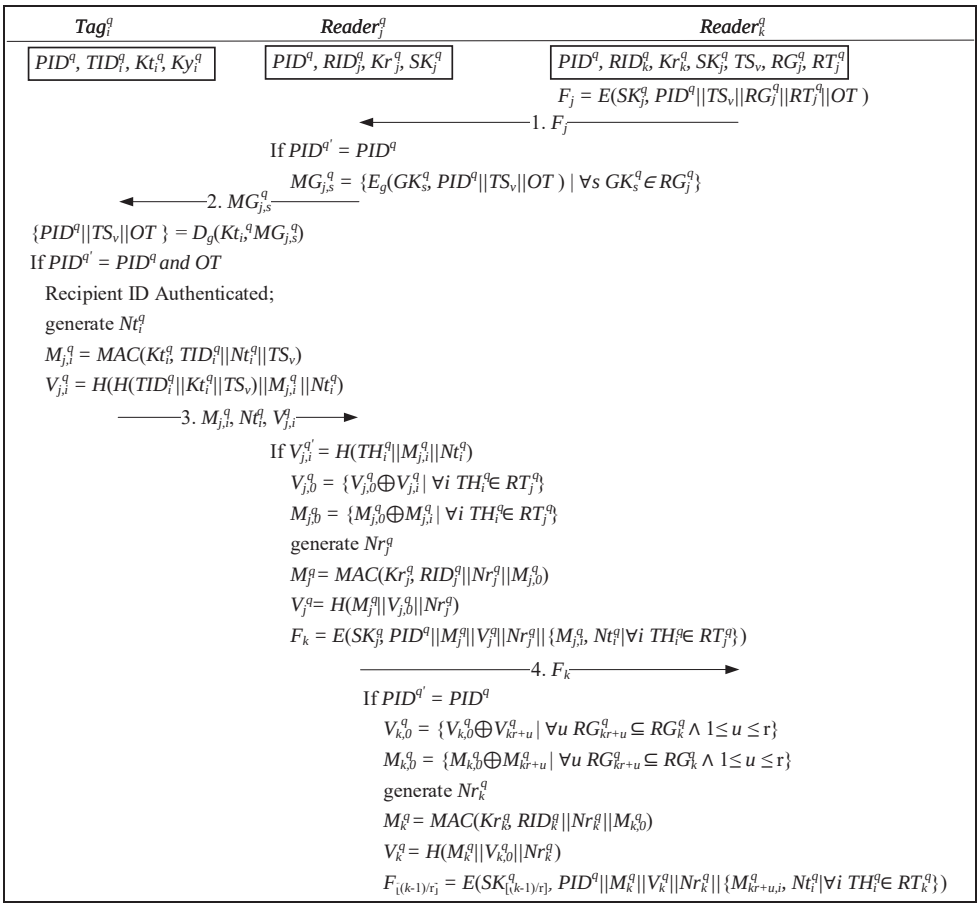

Figure 3. Generating grouping proofs by using a multilayered reader.

As shown in Figure 4, the reader RID_0 receives a response message from the recipient PID^q.

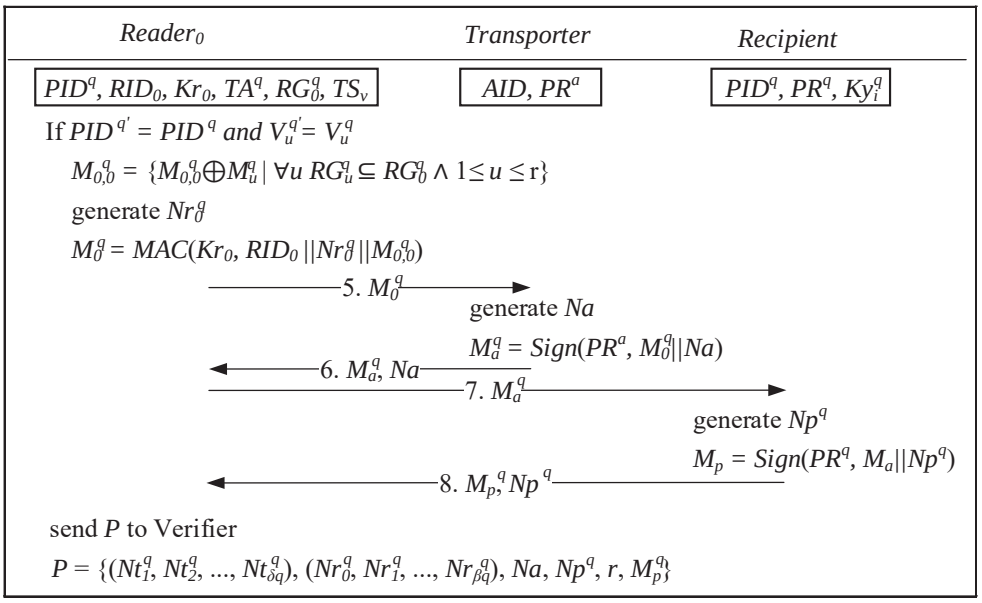

Figure 4. Affirming tags and proofs signed by both sides and verifying time constraint.

[Message 5] Once all messages transmitted by the child node reader are verified by matching the message verification code V_j^q to reconfirm message integrity, the reader RID_0 combines all pieces of proof received from RID_k^q into a combined proof $M_{0,0}^q$. The shared key Kr_0 is employed to generate the grouping proof M_0^q by using the identification code RID_0, a random number Nr_0^q generated by reader RID_0, and the combined proof $M_{0,0}^q$. The grouping proof M_0^q is then transmitted to the transporter's tags to be signed.

[Message 6] When the transporter's tags receive a request message from the reader RID_0 to sign the grouping proof M_0^q, a random number Na is generated to be used along with the transporter's private key PR^a to sign the grouping proof M_0^q and change it into a signed proof M_a^q. The signed proof M_a^q and the random number Na are then transmitted back to the reader RID_0.

[Message 7] After the reader RID_0 receives the signed proof M_a^q from the transporter's tag, the signed proof M_a^q is transmitted to the recipient's tag for signing. Using the random numbers Np^q generated by the recipient's tag and the private key PR^q, the recipient uses the signing function to sign M_a^q into the signed proof M_P^q. The signed proof M_p^q and a random number Np^q are then transmitted back to the reader RID_0.

[Message 8] After the reader RID_0 receives the signed proof from both the transporter's tag and the recipient's tag, the final grouping proof P is then transmitted to the verifier. When the verifier receives the final grouping proof P from the reader RID_0, the verifier first computes the time difference between the current system time and the timestamp TS_v to check whether it was completed within the time threshold. Subsequently, the proof M_p^q is decrypted using the recipient's public key PK^q and a random number Np^q to obtain the signed proof M_a^q, which is then decrypted using the transporter's public key and a random number Na to obtain the grouping proof $M_0^{q\prime}$. The verifier computes M_0^q to determine whether the received grouping proof M_0^q is identical, thus completing the grouping proof protocol.

2.3. Ownership Transfer Phase

Once the verifier confirms that there are no problems with the proof received from reader RID_0, it generates new ownership by shared key Ky_j^q for the recipient's tag, as shown in Figure 5.

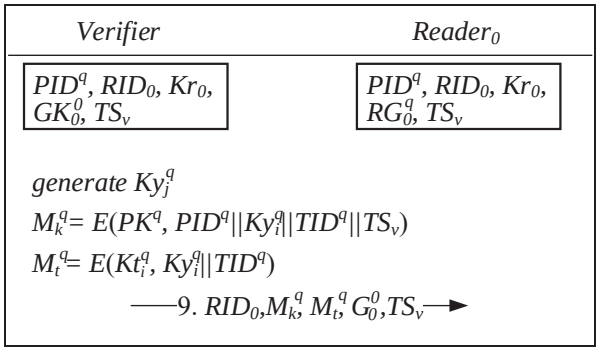

Figure 5. Acquiring the new ownership of shared keys from the verifier.

[Message 9] The verifier uses the tag's current shared key $Ky_j^{q'}$ and the random number Nt_i^q previously generated by the tag to compute the new ownership by shared key $Ky_i^q = E(Ky_i^{q'}, Nt_i^q)$. Subsequently, two encrypted messages are generated using the recipient's public key PK^q and the tag's shared key Kt_i^q. The encrypted message M_k^q consists of the recipient identification code PID^q, new ownership by shared key Ky_i^q, tag's identification code TID_i^q, and the new timestamp TS_v, whereas M_t^q consists of the new ownership by shared key Ky_i^q and the tag's identification code TID_i^q. Moreover, a new set of group keys G_0^0 is generated and transmitted to the reader RID_0 along with TS_v, M_k^q, and M_t^q.

After receiving the transmitted message from the verifier, the reader RID_0 proceeds to encrypt the message M_t^q by using the session key SK_j^q along with PID^q and the group key GK_0^0 as the encrypted message F_i. Both encrypted messages M_k^q and F_i are simultaneously transmitted to the recipient's tags and reader RID_k^q.

[Message 10] Once the recipient's tag receives the encrypted messages from reader RID_0, the recipient's tag decrypts the message to verify whether PID^q is correct. If the verification is successful, the recipient's tag updates the current ownership by shared key $Ky_i^{q'}$ with the new ownership by shared key Ky_i^q, as shown in Figure 6.

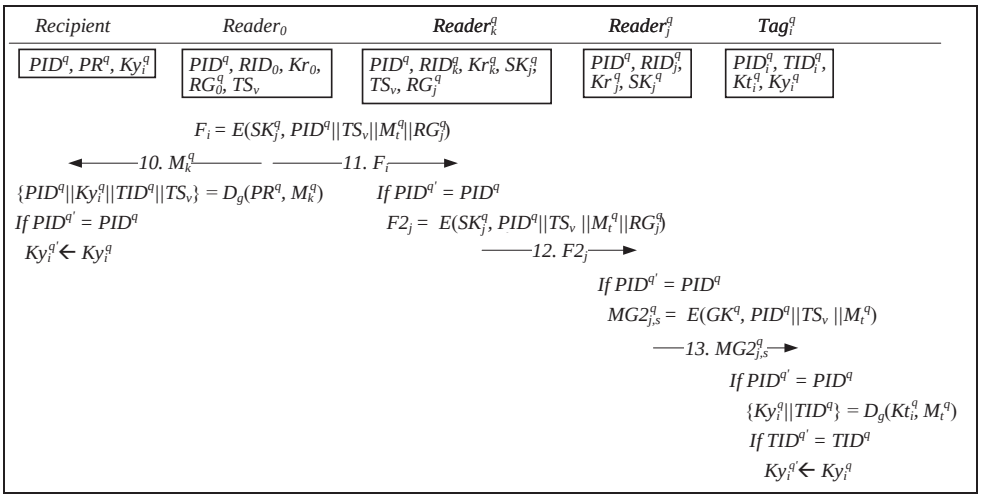

Figure 6. Updating ownership by shared keys.

[Message 11] After receiving the encrypted message from RID_0, the reader RID_k^q proceeds to decrypt the message by using the session key SK_j^q. Depending on the child node, RID_k^q splits RG_j^q (refer to

Figure 2) accordingly and then encrypts PID^q, TS_v, M_t^q, and RG_j^q by using the child node session key SK_j^q as message $F2_j$ and then transmits it to the reader RID_j^q.

[Message 12] Upon receiving the encrypted message $F2_j$ from RID_k^q, the leaf node reader RID_j^q uses its session key SK_j^q to decrypt the encrypted message. All group keys in RG_j^q are extracted, and the multicast message $MG2_{j,s}^q$ is encrypted by using the group keys GK_s^q along with PID^q, TS_v, and M_t^q and then transmitted to each tag.

[Message 13] When any tag TID_i^q receives a multicast message $MG2_{j,s}^q$, the tag will proceed to decrypt the message by using the shared key Kt_i^q and then determine whether the decrypted message contains the correct PID^q. If PID^q is correct, the message M_t^q is decrypted to retrieve the tag TID_i^q for further confirmation. When the message is successfully authenticated, the tag updates its current ownership by shared key $Ky_i^{q'}$ with the new ownership by shared key Ky_i^q, thus effectively completing the ownership transfer protocol.

3. Security Analysis

Table 2 presents a comparison of the security features. **O** denotes that a method listed in the comparison is capable of a feature; **X** denotes that a method fails to achieve the feature; and △ means that the method can achieve the feature when certain circumstances are satisfied. In addition, we use **OT** as an abbreviation for ownership transfer and **GP** for grouping proof to indicate which vulnerabilities are present in these two protocols. For example, the replay attack can take place in both protocols, whereas the denial of proof vulnerability is unique to grouping proof protocols.

Table 2. Comparison of security features of different protocols.

Protocol	Replay Attack * (OT/GP)	Denial of Proof (GP)	Concurrency Attack (GP)	Denial of Service (OT)	Forward Secrecy (OT)	Backward Secrecy (OT)
Zuo [22] + Hermes and Peeters [5]	*O*/*O*	X	X	X	O	O
Zuo [22] + Saito and Sakurai [3]	*O*/*X*	X	X	X	O	O
Zuo [22] + Sun et al. [24]	*O*/*O*	X	△	X	O	O
Zuo [22] + Yen et al. [25]	*O*/*O*	△	O	X	O	O
Our Protocol	*O*/*O*	O	O	O	O	O

* Denotes replayed transmission from either reader to tag or from tag to reader.

In the method proposed by Saito and Sakurai. [3], the messages are transmitted without random numbers or any counters to prevent old messages from being replayed.

In addition to the legitimacy of the generated proof, the previously proposed protocols [3,5,24] do not authenticate the responses received from the tags. Therefore, if the response messages are generated by tags that do not belong to the current tag group, the verifier will reject and discard the proof, leading to denial of proof. Concurrent attacks can occur when several readers simultaneously attempt to generate grouping proofs for the same tag, which prevents the proofs from being generated because the contents of the previous tags are overwritten by subsequent readers. The protocol proposed by Saito and Sakurai [3] requires the tags and readers to be written multiple times to generate the grouping proof, which can cause a problem in a scenario in which previously written information can be overwritten by other readers. In Hermes and Peeters's protocol [5], the reader must read the tags more than twice to generate a proof. Although Sun et al.'s protocol [24] does not overwrite proofs when the tags are read by different readers during inspection, the random numbers are not subject to the same security check and might therefore be overwritten.

Jannati and Falahati [26] proved that Zuo's protocol [22] is vulnerable to the desynchronization attack, which causes a tag to lose synchronization with the new owner, resulting in DoS. In Sections 3.1–3.7, we analyze our proposed protocol, which prevents all the aforementioned threats.

3.1. Replay Attack

Assume an attacker can intercept all previous generated grouping proof messages transmitted between all communicating parties and resend them later to bypass authentication or to generate a bogus proof.

However, because each piece of proof contains a different random number $Nt_{j,i}^{q}$ or a timestamp TS_v for every session, the reader can detect and ignore replayed messages by verifying the timestamps or the random numbers to check whether it has been used before.

3.2. Denial of Proof

Suppose a malicious attacker intercepts all previously generated grouping proof messages transmitted between all communicating parties and attempts to generate a bogus proof by using fake information, causing the verifier to reject and discard the proofs, resulting in denial of proof.

However, in our protocol, each piece of proof computes a verification code $V_{j,i}^{q}$. The reader can use the tag verification code TA^{q} provided by the verifier to check whether this piece of proof belongs to this delivery. To generate a valid proof, the hacker must obtain the secret Kt_i^q, which is shared only between the tags and the verifier. The secret Kt_i^q are not transmitted during protocol execution, and, therefore, the attacker will have no way to acquire them, avoiding the occurrence of denial of proof.

3.3. Denial of Service

When a malicious attacker interrupts the interaction between the reader and the tags by intercepting or blocking the shared key update message, the tag might lose synchronization with the verifier, resulting in DoS.

In our proposed protocol, the verifier stores the old ownership by shared key $Ky_i^{q'}$ and the new ownership by shared key Ky_i^q. In the case of a DoS attack that blocks these shared key update messages, the verifier can still authenticate the tags.

3.4. Forward Secrecy

Assume an attacker can intercept all previously transmitted messages between all communicating parties during the ownership transfer phase. However, without knowing the previous ownership by shared key $Ky_i^{q'}$, the new owner (recipient P^q) cannot decrypt the message transmitted between the tags and its former owner (supplier).

3.5. Backward Secrecy

Assume an attacker can intercept all forward messages transmitted between all communicating parties after ownership transfer.

However, in our protocol, new ownership by shared key Ky_i^q is computed by the verifier by encrypting the old ownership by shared key $Ky_i^{q'}$ and a random number Nt_i^q generated by the tags (Figure 3). Therefore, without the new ownership by the shared key, the former owner (supplier) cannot further track messages transmitted between the new owner and the tags.

3.6. Concurrent Attacks

When two readers simultaneously use the same tags, specific parameters may be overwritten. An adversary can use a reader to crisscross specific tags, thereby blocking generation of the grouping proof.

In our proposed protocol, there is no temporary parameter, and the reader needs to communicate with the tag only once to generate the pieces of proof $M_{j,i}^{q}$. Hence, no adversary can mount a concurrent attack.

3.7. Message Integrity

Assume an attacker can intercept all previously generated grouping proof messages transmitted between all communicating parties, and then attempts to modify the message to generate a bogus proof to fool the reader or verifier.

In our protocol, each legitimate tag generates a different random number Nt_i^q in each session to compute the pieces of proof $M_{j,i}^q$. Moreover, a verification code $V_{j,i}^q$ is included in the response message to ensure the integrity of that message. The attacker might attempt to retrieve the shared key Kt_i^q from the verification code $V_{j,i}^q$, but, owing to the assumption of OHF, no useful information will be obtained.

4. Performance Analysis

This section analyzes the combination of Zuo's protocol [22] with grouping proof protocols that do not require a predetermined sequence [5,24,25] and compares the computation capacity loads against our proposed protocol. To ensure unbiased comparison, the analysis was conducted at scan rate of 3.55 M clock cycles per second according to Yang et al.'s effectiveness analysis [23]. Additionally, asymmetric encryption and error-correcting code were employed using the same security strength of 2^{80} bits. The computation time for XOR logic operation was minimal (compared to the crypto-algorithm) and was therefore neglected and not included in the comparison. The reader adopted for this comparison was assumed to have powerful arithmetic capability.

Table 3 lists the notations used for the computation comparison. Table 4 presents the computation costs of the compared protocols. Each of the readers in our proposed protocol can manage a maximum of r tags, and therefore sends only one multicast message to all the tags. The grouping proof protocols in [24,25] can also send multicast messages to all tags, but, when $m > r$, the reader must transmit the message multiple times, thus requiring a computation time of $\lceil m/r \rceil$ times. Our protocol adopts a multilayered grouping proof structure. Although a similar message is broadcast, readers are required to communicate with each other. Therefore, our computation time would require $\lceil log_r(m/r) \rceil$ times.

Table 3. Computational capacity symbol notations.

Symbol	Description
T_{SE}	computation time for conducting symmetric encryption and decryption
T_{RNG}	required time for generating a random number
T_H	computation time for executing a hash function
T_{EC}	required time for conducting elliptic curve encryption and decryption
T_{SIG}	required time for proof signing
T_M	required time to compute a message authentication code
T_G	required time for encrypting and decrypting a group key
m	total tags
r	maximum number of tags that a reader can scanned concurrently

Table 4. Computational capacity.

Schemes	Tag	Mobile Reader
Zuo [22] + Hermes and Peeters [5]	$9mT_{SE} + 4mT_H + (\lceil 2/r \rceil)mT_{EC} + (\lceil 2/r \rceil)mT_{RNG}$	$(4m+7)T_{SE} + 4mT_H + T_{SIG} + T_{RNG}$
Zuo [22] + Sun et al. [24]	$(9 + \lceil 2/r \rceil)mT_{SE} + 4mT_H + (\lceil 1/r \rceil)mT_{RNG}$	$(4m+9)T_{SE} + (4m+2)T_H$
Zuo [22] + Yen et al. [25]	$9mT_{SE} + 4mT_H + (\lceil 7/r \rceil)mT_{RNG}$	$(4m+7)T_{SE} + 4mT_H + (m+5)T_{RNG} + 2T_{SIG}$
Our Protocol	$2T_G + T_M + 2T_H + T_{RNG}$	$2T_G + 5T_{SE} + 2T_{SIG} + 4T_H + 3T_M + 3T_{RNG} + (\lceil log_r(m/r) \rceil)(3T_{SE} + 2T_H + T_{RNG})$

According to the method proposed by Yen et al. [25], the computational cost of the reader would increase depending on the number of m tags because the reader would need to verify the identification code of each tag. Hermes and Peeters [5] and Sun et al. [24] employed methods in which identical messages are broadcast to all tags, ensuring the constant computational capacity required by the reader to generate grouping proofs.

We use "OP" as an abbreviation for our proposed protocol; "ZY" represents the combination of Zuo [22] and Yen et al. [25]; "ZS" represents the combination of Zuo [22] and Sun et al. [24]; and "ZH" represents the combination of Zuo [22] and Hermes and Peeters [5]. Figures 8 and 9 show the readers with the maximum reading capacity of 200 tags employed to determine the computing times required by various methods. The number of tags starts from 100 and is doubled until it reaches 12,800.

According to Figure 7, OP involves fewer than 800 tags, and a longer computation time is required because the group key must be decrypted. As the number of tags increases, the computation time increases because computing time and tag number are linearly related. When the number of tags exceeds 800, OP is more efficient compared to the other methods. The computation times of ZY and ZS differ by only 1%.

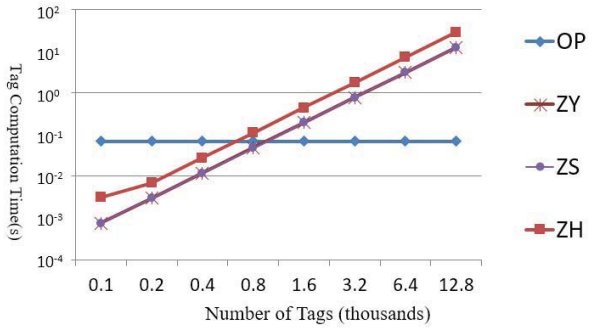

Figure 7. Comparisons of computation loads of tags.

In addition, Figure 8 shows that, when processing fewer than 3200 tags, OP needs more computing time because the group message must be encrypted and transmitted between the readers. Later, the proof requires signatures from both the transporter and the recipient. However, when the number of the tags exceeds 3200, OP is more efficient compared to the other methods. This advantage shows that OP is a favorable choice in an environment (e.g., SCM) where large numbers of tags must be scanned concurrently to generate proofs and transfer cargo ownership.

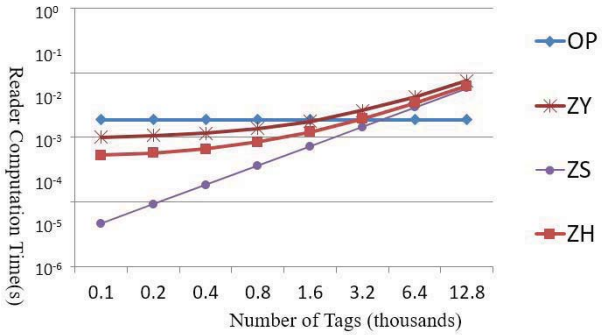

Figure 8. Comparisons of computation loads of readers.

According to EPC Class-1 Generation-2 standard, the highest transmission rate from a reader to a tag is 160 kbps and that from a tag to a reader is 640 kbps. We used the transmission rate and the message lengths stated in Table 5 to compute the required transmission time from a reader to their tags and from the tags to their reader.

Table 6 presents a comparison of the transmission time of OP with those of the other methods. In OP, we adopted a multilayered grouping proof structure, and a maximum of r tags were distributed to each reader. Thus, compared to other methods, an increase in the number of tags did not lead to an increase in transmission from the tags to the reader. Furthermore, the transmission time from the

reader to the tags increased by $\lceil log_r(m/r) \rceil$ times because a read-tree was employed. ZY, ZS, and ZH could not manage m tags simultaneously; therefore, the transmission was repeated $\lceil m/r \rceil$ times when those methods were used.

Table 5. Transmission capacity, symbol notation, and estimated length.

Symbol	Estimated Length	Deription
L_{ID}	64 bits	length of a tag identification code (based on ISO-18000-6)
L_{SE}	64 bits	message length after applying symmetric encryption
L_{RNG}	64 bits	message length for a random number
L_M	64 bits	message length for a message authentication code
L_H	64 bits	message length of a hash function
L_{EC}	192 bits	message length after applying elliptic curve encryption
L_G	1024 bits	represents the message length after performing group key encryption

Table 6. Transmission capacity.

Schemes	From Tag to Reader	From Reader to Tag (or Reader)
Zuo [22] + Hermes and Peeters [5]	$(4+2m)L_{SE}+2mL_{EC}+mL_{RNG}$	$11L_{SE}+4L_H+\lceil m/r \rceil(4L_{SE}+4L_H+L_{RNG})$
Zuo [22] + Sun et al. [24]	$(4+4m)L_{SE}+mL_{ID}$	$11L_{SE}+4L_H+\lceil m/r \rceil(4L_{SE}+7L_H)$
Zuo [22] + Yen et al. [25]	$(4+2m)L_{SE}+4mL_{RNG}$	$11L_{SE}+4L_H+\lceil m/r \rceil(4L_{SE}+3L_{RNG}+4L_H)$
Our Protocol	$r(L_{RNG}+L_H+L_M)+2L_{RNG}+2L_{SIG}$	$2L_G+3mL_{SE}+L_M+L_{RNG}+(\lceil log_r(m/r) \rceil)(3L_{SE})$

Figures 9 and 10 show the required transmission times for the tags and readers. According to Figure 9, OP requires slightly more transmission time when fewer than 100 tags are involved. However, when the number of tags is more than 100, OP is more efficient compared to the other protocols, because the other protocols would need to divide the tags and transmit them over several cycles, thereby increasing the transmission time. Figure 10 shows that OP has the shortest transmission time from readers to tags because OP requires only one multicast to broadcast the messages to all tags, as opposed to the other methods, which require the messages to be transmitted multiple times.

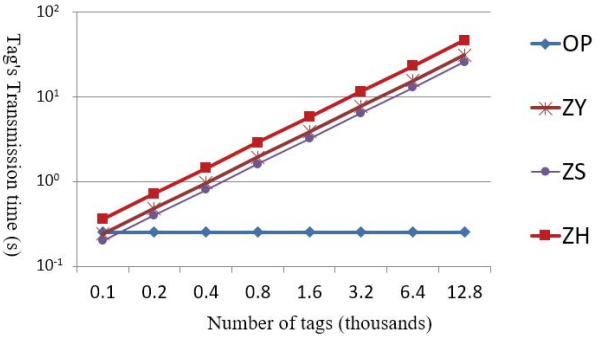

Figure 9. Comparisons of transmission load of tags.

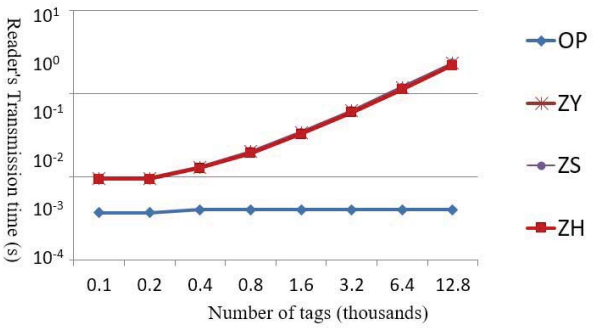

Figure 10. Comparisons of transmission load of the readers.

5. GNY Logic Proof

In this section, we apply GNY [27] logic to prove the security of our proposed protocol. Our analysis includes four parts:

1. Definition of GNY logic message (Table 6)
2. Initial assumptions (Table 7)
3. Goals of proposed protocol (Table 8)
4. Proving process (Table 9)

Table 7. Definition of GNY logic message.

Notation	Description
A	a transporter who delivers cargo
P^q	a recipient who receives the cargo
R_0	the transporter's reader
R_j^q	the recipient's reader
T_i^q	the cargo's tag
$\{X\}_K, \{X\}_k^{-1}$	message X is encrypted/decrypted with symmetric key k
$\{X\}_{+K}, \{X\}_{-K}$	message X is encrypted using a public key $+K$ or decrypted with a private key $-K$
$P \triangleleft X$	P is told message X
$P \triangleleft *X$	P is told message X that is not-originated-here
$P \ni X$	P possess message X
$P \mid\equiv Q \mid\sim X$	P believes Q once conveyed message X
$P \mid\equiv \sharp(X)$	P believes X is fresh
$P \mid\equiv \phi(X)$	P believes X is recognizable
$P \mid\equiv P \overset{s}{\longleftrightarrow} Q$	P believes s is a suitable secret for P and Q
$P \mid\equiv P \overset{+K}{\longrightarrow} Q$	P believes that $+K$ is a suitable public key for Q
$P \mid\equiv Q \mid\Rightarrow Q \mid\equiv *$	P believes Q has jurisdiction over all his beliefs

Table 8. Initial assumptions.

Transporter A	**Recipient P^q**
$A \ni AID, PR^a, Na$	$P^q \ni PID^q, PR^q, Np^q$
$A \mid\equiv \sharp(Na)$	$P^q \mid\equiv \sharp(Np^q)$
$A \mid\equiv R_0 \mid\Rightarrow R_0 \mid\equiv *$	$P^q \mid\equiv R_0 \mid\Rightarrow R_0 \mid\equiv *$
Reader R_0	Reader R_j^q
$R_0 \ni PID^q, RID_0, Kr_0, TH_i^q, GK_i^q, SK_j^q, TS_v, Nr_0^q, OT$	$R_j^q \ni PID^q, RID_j^q, Kr_j^q, SK_j^q, Nr_j^q, OT$
$R_0 \mid\equiv \sharp(Nr_0^q)$	$R_j^q \mid\equiv \sharp(Nr_j^q)$
$R_0 \mid\equiv \phi(PID^q)$	$R_j^q \mid\equiv \phi(PID^q)$
$R_0 \mid\equiv \overset{GK_i^q}{\longrightarrow} T_i^q$	$R_j^q \mid\equiv R_j^q \overset{Kr_j^q}{\longleftrightarrow} V$
$R_0 \mid\equiv R_0 \overset{SK_j^q}{\longleftrightarrow} R_j^q$	$R_j^q \mid\equiv R_j^q \overset{SK_j^q}{\longleftrightarrow} R_0$
$R_0 \mid\equiv R_0 \overset{Kr_0}{\longleftrightarrow} V$	$R_j^q \mid\equiv R_0 \mid\Rightarrow R_0 \mid\equiv *$
$R_0 \mid\equiv T_i^q \mid\Rightarrow T_i^q \mid\equiv *$	
Verifier V	Tag T_i^q
$V \ni PID^q, RID_0, TID_i^q, GK_i^q, Kt_i^q, Kr_j^q, TS_v, Ky_i^q, OT$	$T_i^q \ni PID^q, TID_i^q, Kt_i^q, Nt_i^q, OT$
$V \mid\equiv \sharp(TS_v)$	$T_i^q \mid\equiv \sharp(Nt_i^q)$
$V \mid\equiv \phi(TID_i^q)$	$T_i^q \mid\equiv \phi(PID^q)$
$V \mid\equiv \phi(RID_j^q)$	$T_i^q \mid\equiv T_i^q \overset{Kt_i^q}{\longleftrightarrow} V$
$V \mid\equiv \phi(RID_0)$	$T_i^q \mid\equiv T_i^q \overset{Ky_i^q}{\longleftrightarrow} P^q$
$V \mid\equiv V \overset{Kt_i^q}{\longleftrightarrow} T_i^q$	$T_i^q \mid\equiv R_0 \mid\Rightarrow R_0 \mid\equiv *$
$V \mid\equiv V \overset{Kr_j^q}{\longleftrightarrow} R_j^q$	
$V \mid\equiv V \overset{Kr_0}{\longleftrightarrow} R_0$	
$V \mid\equiv R_0 \mid\Rightarrow R_0 \mid\equiv *$	

Table 9. Goals of the proposed protocol.

First Goal	
$T_i^q \mid\equiv R_j^q \sim \sharp(\{PID^q, TS_v\})_{GK_i^q}$ $T_i^q \mid\equiv R_j^q \sim \phi(PID^q)$ $R_j^q \mid\equiv T_i^q \sim \sharp(M_{j,i}^q, Nt_i^q, V_{j,i}^q)$ $R_j^q \mid\equiv T_i^q \sim \phi(V_{j,i}^q)$ $R_0 \mid\equiv R_j^q \sim \sharp(\{PID^q, M_j^q, V_j^q, Nr_j^q, M_{j,i}^Q, Nt_i^q\}_{SK_i^q})$ $R_0 \mid\equiv R_j^q \sim \phi(PID^q)$ $R_j^q \mid\equiv T_i^q \sim \phi(V_j^q)$ $V \mid\equiv R_0 \sim \sharp(Nt_i^q, Nr_j^q, Na, Np^q, r, M_p^q)$ $V \mid\equiv R_0 \sim \phi(M_p^q)$	The recipient's reader R_j^q can authenticate all tags T_i^q, and the tags T_i^q can recognize the received message to generate pieces of proof $M_{j,i}^q$. The pieces of proof $M_{j,i}^q$ are later combined into a grouping proof M_p^q through the transporter's reader R_0 and then transmitted to the verifier V.
Second Goal	
$T_i^q \mid\equiv R_j^q \sim \sharp(\{PID^q, TS_v, M_t^q\}_{GK_i^q})$ $T_i^q \mid\equiv R_j^q \sim \phi(PID^q)$ $T_i^q \mid\equiv R_j^q \sim \phi(TID^q)$ $T_i^q \mid\equiv T_i^q \overset{Ky_i^q}{\longleftrightarrow} P^q$ $P^q \mid\equiv R_0 \sim \sharp(\{M_k^q\}_{+PK^q})$ $P^q \mid\equiv R_0 \sim \phi(PID^q)$ $P^q \mid\equiv P^q \overset{Ky_i^q}{\longleftrightarrow} T_i^q$	The recipient's reader R_j^q can authenticate all of the tags T_i^q and the tags T_i^q can recognize the received message, therefore it updates the shared key Ky_i^q. The recipient P^q recognized the received message from reader R_0 therefore updates the shared key Ky_i^q.

Please refer to the GNY reasoning studies [27] for the rules (e.g., P1, T1, and F1). The proof process of our proposed protocol is shown as in Table 10.

Table 10. Proof process.

Proof	
Message 1: $R_j^q \triangleleft *\{PID^q, TS_v, GK_i^q, TH_i^q, OT\}_{SK_j^q}$ $R_j^q \triangleleft \{PID^q, TS_v, GK_i^q, TH_i^q, OT\}_{SK_j^q}$ /*T1*/ $R_j^q \ni \{PID^q, TS_v, GK_i^q, TH_i^q, OT\}_{SK_j^q}$ /*P1*/ $R_j^q \mid\equiv R_0 \sim \phi(PID^q)$ /*IA, I1, R2*/ $R_j^q \mid\equiv R_0 \sim \sharp(TS_v, GK_i^q, TH_i^q, OT)$ /*IA, I1, F2*/ $R_j^q \mid\equiv \sharp(\{PID^q, TS_v, OT\}_{GK_i^q})$ /*IA, F3*/	Since the session key SK_j^q is generated using the shared key between verifier V, R_j^q believes that the messages come from R_0.
Message 2: $T_i^q \triangleleft *\{PID^q, TS_v, OT\}_{GK_i^q}$ $T_i^q \triangleleft \{PID^q, TS_v, OT\}_{GK_i^q}$ /*T1*/ $T_i^q \ni \{PID^q, TS_v, OT\}_{GK_i^q}$ /*P1*/ $T_i^q \mid\equiv R_j^q \sim \phi(PID^q)$ /*IA, I2, R2*/ $T_i^q \mid\equiv R_j^q \sim \phi(OT)$ /*R5*/ $T_i^q \mid\equiv R_j^q \sim \sharp(TS_v)$ /*IA, I2, F2*/ $T_i^q \mid\equiv \sharp(Nt_i^q)$ /*IA*/ $T_i^q \mid\equiv \sharp(M_{(j,i)^q}, Nt_i^q, H(H(TID_i^q, Kt_i^q, TS_v), M_{(j,i)^q}, Nt_i^q))$ /*IA, F10*/	Since the group key GK_i^q is generated by the verifier V, T_i^q believes that the messages come from R_j^q. Once the PID^q is identified, a fresh random number Nt_i^q will be generated to compute the proof $M_{j,i}^q$ and piece of the message verification code $V_{j,i}^q$ to ensure message is fresh, not replayed and has not been tampered with.
Message 3: $R_j^q \triangleleft *M_{j,i}^q, *Nt_i^q, *H(H(TID_i^q, Kt_i^q, TS_v), M_{j,i}^q, Nt_i^q)$ $R_j^q \triangleleft M_{j,i}^q, Nt_i^q, H(H(TID_i^q, Kt_i^q, TS_v), M_{j,i}^q, Nt_i^q)$ /*T1*/ $R_j^q \ni M_{j,i}^q, Nt_i^q, H(H(TID_i^q, Kt_i^q, TS_v), M_{j,i}^q, Nt_i^q)$ /*P1*/ $R_j^q \mid\equiv T_i^q \sim \phi(H(H(TID_i^q, Kt_i^q, TS_v), M_{j,i}^q, Nt_i^q))$ /*IA, I6, R5*/ $R_j^q \mid\equiv T_i^q \sim \sharp(M_{(j,i)^q}, Nt_i^q)$ /*I6, F1*/ $R_j^q \mid\equiv \sharp(Nr_j^q)$ /*IA*/ $R_j^q \mid\equiv \sharp(\{PID^q, M_j^q, V_j^q, Nr_j^q, M_{j,i}^q, Nt_i^q\}_{SK_j^q})$ /*IA, F2*/	Message verification code $V_{j,i}^q$ is verified to ensure the message from tag T_i^q has not beeen tampered with.

Table 10. *Cont.*

Proof	
Message 4:	
$R_0 \triangleleft *\{PID^q, M_j^q, V_j^q, Nr_j^q, M_{j,i}^q, Nt_i^q\}_{SK_j^q}$	Message verification code $V_(j,i)^q$ is verified to ensure the message has not been tampered with and PID^q is identified to ensure that the message comes from R_j^q.
$R_0 \triangleleft \{PID^q, M_j^q, V_j^q, Nr_j^q, M_{j,i}^q, Nt_i^q\}_{SK_j^q}$ /*T1*/	
$R_0 \ni \{PID^q, M_j^q, V_j^q, Nr_j^q, M_{j,i}^q, Nt_i^q\}_{SK_j^q}$ /*P1*/	
$R_0 \mid\equiv R_j^q \sim \phi(PID^q)$ /* IA, I1, R2*/	
$R_0 \mid\equiv R_j^q \sim \phi(V_j^q)$ /*IA, I1, R2*/	
$R_0 \mid\equiv R_j^q \sim \sharp(M_j^q, Nr_j^q, M_{j,i}^q, Nt_i^q)$ /*IA, I2, F2*/	
$R_0 \mid\equiv \sharp(Nr_0^q)$ /*IA*/	
$R_0 \mid\equiv \sharp(M_0^q)$ /*IA, F10*/	
Message 5:	
$A \triangleleft *M_0^q$	The transporter A generates a random number Na to ensure the message is fresh and uses private key PR^a to sign the grouping proof M_0^q as a proof of participation.
$A \triangleleft M_0^q$ /*T1*/	
$A \ni M_0^q$ /*P1*/	
$A \mid\equiv \sharp(Na)$ /*IA */	
$A \mid\equiv \sharp(\{M_0^q, Na\}_{-PR^a})$ /*IA, F4*/	
Message 6:	
$R_0 \triangleleft *\{M_0^q, Na\}_{-PR^a}, *Na$	Reader R_0 receives the signed grouping proof. M_a^q
$R_0 \triangleleft \{M_0^q, Na\}_{-PR^a}, Na$ /*T1*/	
$R_0 \ni \{M_0^q, Na\}_{-PR^a}, Na$ /*P1*/	
Message 7:	
$P^q \triangleleft *M_a^q$	The recipient P^q generates a random number Np^q to ensure the message is fresh and uses private key PR^q to sign the grouping proof M_a^q as a proof of participation.
$P^q \triangleleft M_a^q$ /*T1*/	
$P^q \ni M_a^q$ /*P1*/	
$P^q \mid\equiv \sharp(Np^q)$ /*IA */	
$P^q \mid\equiv \sharp(M_a^q, Np^q{}_{-PR^p})$ /*IA, F4*/	
Message 8:	
$R_0 \triangleleft *\{M_p^q, Np^q\}_{-PR^q}, *Np^q$	Reader R_0 receives the final grouping proof M_p^q.
$R_0 \triangleleft \{M_p^q, Np^q\}_{-PR^p}, Np^q$ /*T1*/	
$R_0 \ni \{M_p^q, Np^q\}_{-PR^p}, Np^q$ /*P1*/	
$V \triangleleft *Nt_i^q, *Nr_0^q, *Nr_j^q, *Na, *Np^q, *r, *M_p^q$	Verifier V will verify the correctness of the final proof M_p^q and identify whether the proof is generated under the time threshold. If there is no issue with the proof, the verifier V will proceed to the ownership transfer phase.
$V \triangleleft Nt_i^q, Nr_0^q, Nr_j^q, Na, Np^q, r, M_p^q$ /*T1*/	
$V \ni Nt_i^q, Nr_0^q, Nr_j^q, Na, Np^q, r, M_p^q$ /*P1*/	
$V \mid\equiv R_0 \sim \sharp(M_a^q)$ /*IA, I4, F4*/	
$V \mid\equiv R_0 \sim \sharp(M_p^q)$ /*IA, I4, F4*/	
$V \mid\equiv R_0 \sim \phi(M_p^q)$ /*IA, I1, R5*/	
Message 9:	
$R_0 \triangleleft *\{RID_0, \{PID^q, Ky_i^q, TID^q, TS_v\}_{+PK^q}, \{Ky_i^q, TID^q\}_{Kt_i^q}, G_0^0, TS_v\}_{SK_j^q}$	Verifier V generates TS_v to ensure the message is fresh, i.e. not replayed.
$R_0 \triangleleft \{RID_0, \{PID^q, Ky_i^q, TID^q, TS_v\}_{+PK^q}, \{Ky_i^q, TID^q\}_{Kt_i^q}, G_0^0, TS_v\}_{SK_j^q}$ /*T1*/	
$R_0 \ni \{RID_0, \{PID^q, Ky_i^q, TID^q, TS_v\}_{+PK^q}, \{Ky_i^q, TID^q\}_{Kt_i^q}, G_0^0, TS_v\}_{SK_j^q}$ /*P1*/	
$R_0 \mid\equiv V \sim \phi(RID_0)$ /*IA, I1, R2*/	
$R_0 \mid\equiv V \sim \sharp(TS_v, G_0^0)$ /*IA, I1, F2*/	
$R_0 \mid\equiv \sharp(\{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q})$ /*IA, F3*/	
Message 10:	
$P^q \triangleleft *\{PID^q, Ky_i^q, TID^q, TS_v\}_{+PK^q}$	PID^q is identified to ensure that message comes from R_0.
$P^q \triangleleft \{PID^q, Ky_i^q, TID^q, TS_v\}_{+PK^q}$ /*T1*/	
$P^q \ni \{PID^q, Ky_i^q, TID^q, TS_v\}_{+PK^q}$ /*P1*/	
$P^q \mid\equiv \sharp(TS_v)$ /*IA */	
$P^q \mid\equiv R_0 \sim \phi(PID^q)$ /*I2, R2*/	
$P^q \mid\equiv P^q \overset{Ky_i^q}{\longleftrightarrow} T_i^q$ /*J1*/	
Message 11:	
$R_k^q \triangleleft *_{\blacksquare}\{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q}$	Since the session key SK_j^q is generated using the shared key between verifier V, R_k^q that the messages come from R_0.
$R_k^q \triangleleft \{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q}$ /*T1*/	
$R_k^q \ni \{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q}$ /*P1*/	
$R_k^q \mid\equiv R_0 \sim \phi(PID^q)$ /*IA, I1, R2*/	
$R_k^q \mid\equiv R_0 \sim \sharp(TS_v, GK_i^q)$ /*IA, I1, F2*/	
$R_k^q \mid\equiv \sharp(\{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q})$ /*IA, F3*/	

Table 10. *Cont.*

Proof	
Message 12:	
$R_j^q \lhd *\{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q}$	Since the session key SK_j^q is generated using the
$R_j^q \lhd \{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q}$	shared key between verifier V, R_j^q
/*T1*/	believes that the messages come from R_k.
$R_j^q \ni \{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}, GK_i^q\}_{SK_j^q}$	
/*P1*/	
$R_j^q \mid\equiv R_0 \sim \phi(PID^q)$ /*IA, I1, R2*/	
$R_j^q \mid\equiv R_0 \sim \sharp(TS_v, GK_i^q)$ /*IA, I1, F2*/	
$R_j^q \mid\equiv \sharp(\{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}\}_{GK_i^q})$ /*IA, F3*/	
Message 13:	
$T_i^q \lhd *\{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}\}_{GK_i^q}$	Since the group key GK_i^q is generated by the
$T_i^q \lhd \{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}\}_{GK_i^q}$ /*T1*/	verifier V, T_i^q believes that the messages
$T_i^q \ni \{PID^q, TS_v, \{Ky_i^q, TID^q\}_{Kt_i^q}\}_{GK_i^q}$ /*P1*/	come from R_j^q.
$T_i^q \mid\equiv R_j^q \sim \phi(PID^q)$ /*IA, I2, R2*/	
$T_i^q \mid\equiv R_j^q \sim \phi(TID^q)$ /*IA, I2, R2*/	
$T_i^q \mid\equiv R_j^q \sim \sharp(TS_v)$ /*IA, I2, F2*/	
$T_i^q \mid\equiv T_i^q \overset{Ky_i^q}{\longleftrightarrow} P^q$ /*J1*/	

6. Conclusions

The emerging development of RFID technology has created the potential of massive deployment using the low cost and highly convenient RFID Tags. In a multi-party environment such as SCM, global trading is no longer just about delivering cargo quickly and efficiently, it is also about moving goods securely to the designated recipient [28]. This paper proposes an interesting approach in which a grouping proof protocol (to prove the existence of a group of tags) and ownership transfer protocol (to transfer the ownership of the tags) can be employed simultaneously without hindering mechanism of the original protocol [23]. In addition, once the verifier has confirmed the validity of generated proof provided by the transporter, the ownership transfer will be executed immediately, thus preventing anyone from tampering with the cargo goods (swapping legitimate goods with the counterfeit items, etc.). Furthermore, in terms of security and privacy, we found that the proposed protocol can prevent most known attacks such as replay attack, denial of service, etc. that aim to exploit the message being transmitted between the readers and the tags.

Author Contributions: The four authors have made equal substantial contributions to this work.

Funding: This research was funded by Ministry of Science and Technology (MOST) grant numbers MOST 107-2218-E-011-012, MOST 107-2221-E-033-010, MOST 107-2221-E-130-001, and MOST 106-2221-E-034-002 in Taiwan.

Acknowledgments: The authors gratefully acknowledge the support from Taiwan Information Security Center (TWISC) and Ministry of Science and Technology (MOST) under the grants MOST 107-2218-E-011-012, MOST 107-2221-E-033-010, MOST 107-2221-E-130-001, and MOST 106-2221-E-034-002 in Taiwan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Leong, C.E. A Research on Supply Chain Security in Malaysia. *Int. J. Supply Chain Manag.* **2014**, *3*, 85–93.
2. Juels, A. Yoking Proof for RFID Tags. In Proceedings of the 2nd Annual International Conference on Pervasive Computing and Communications, Orlando, FL, USA, 14–17 March 2004; pp. 138–143. [CrossRef]
3. Saito, J.; Sakurai, K. Grouping Proof for RFID Tags. In Proceedings of the International Conference on Advanced Information Networking and Applications, Taipei County, Taiwan, 25–30 March 2005; pp. 621–624.
4. Batina, L.; Lee, Y.K.; Seys, S.; Singelée, D.; Verbauwhede, E. Extending ECC-based RFID Authentication Protocols to Privacy-preserving Multi-party Grouping Proofs. *Pers. Ubiquitous Comput.* **2012**, *16*, 323–335. [CrossRef]

5. Hermes, J.; Peeters, R. Private Yoking Proofs: Attacks, Models and New Provable Constructions. In Proceedings of the 8th International Conference on RFIDSec, Nijmegen, The Netherlands, 2–3 July 2012; pp. 96–108.
6. Shen, J.; Tan, H.W.; Chang, S.H.; Ren, Y.J.; Liu, Q. A Lightweight and Practical RFID Grouping Authentication Protocol in Multiple-Tag Arrangements. In Proceedings of the International Conference on Advanced Communication Technology, Ho Chi Minh, Vietnam, 1–3 July 2015; pp. 681–686.
7. Abughazalah, S.; Markantonakis, K.; Mayes, K. Two Rounds RFID Grouping Proof Protocol. In Proceedings of the IEEE International Conference on RFID, Orlando, FL, USA, 3–5 May 2016; pp. 1–14.
8. Burmester, M.; Munilla, J. An Anonymous RFID Grouping Proof with Missing Tag Identification. In Proceedings of the IEEE International Conference on RFID, Orlando, FL, USA, 3–5 May 2016; pp. 1–7. [CrossRef]
9. Burmester, M.; Munilla, J. Resilient Grouping Proof with Missing Tag Identification. In Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence, Las Palmas de Gran Canaria, Spain, 29 November–2 December 2016; pp. 544–555.
10. Rostampour, S.; Bagheri, N.; Hosseinzadeh, M.; Khademzadeh, A. An Authenticated Encryption Based Grouping Proof Protocol for RFID Systems. *J. Secur. Commun. Netw.* **2017**, *9*, 5581–5590. [CrossRef]
11. Shi, Z.; Zhang, X.; Wang, Y. A Lightweight RFID Grouping-Proof Protocol Based on Parallel Mode and DHCP Mechanism. *Information* **2017**, *8*, 85.
12. Huang, H.H.; Yeh, L.Y.; Tsaur, W.J. Ultra-Lightweight Mutual Authentication and Ownership Transfer Protocol with PUF for Gen2v2 RFID Systems. In Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, China, 16–18 March 2016; Volume 2, pp. 655–658.
13. Li, Q.S.; Xu, X.L.; Chen, Z. PUF-based RFID Ownership Transfer Protocol in an Open Environment. In Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies, Hong Kong, China, 9–11 December 2014; pp. 131–137.
14. Li, G.C.; Xu, X.L.; Li, Q.S. LADP: A Lightweight Authentication and Delegation Protocol for RFID Tags. In Proceedings of the International Conference on Ubiquitous and Future Networks, Sapporo, Japan, 7–10 July 2015; pp. 860–865.
15. Niu, H.F.; Jagannathan, S.; Taqieddin, E.S. A Gen2v2 Compliant RFID Authentication and Ownership Management Protocol. In Proceedings of the IEEE Conference on Local Computer Networks, Edmonton, AB, Canada, 8–11 September 2014; pp. 331–336.
16. Sundaresan, S.; Doss, R.; Zhou, W.L.; Piramuthu, S. Secure Ownership Transfer for Multi-tag Multi-owner Passive RFID Environment with Individual-owner-privacy. *Int. J. Comput. Telecommun. Ind.* **2015**, *55*, 112–124. [CrossRef]
17. Yang, M.H.; Xie, K.P. TTP-Based Group Ownership Transfer in A Mobile RFID Environment. *Int. J. Digit. Content Technol. Its Appl.* **2013**, *7*, 51–69.
18. Munilla, J.; Burmester, M.; Peinado, A. Attacks on Ownership Transfer Scheme for Multi-tag Multi-owner Passive RFID Environments. *Comput. Commun.* **2016**, *88*, 84–88. [CrossRef]
19. Kapoor, G.; Piramuthu, S. Vulnerabilities in Some Recently Proposed RFID Ownership Transfer Protocols. *IEEE Commun. Lett.* **2010**, *14*, 260–262. [CrossRef]
20. Burmester, M.; Medeiros, B.; Motta, R. Provably Secure Grouping-proofs for RFID Tags. In Proceedings of the 8th Smart Card Research and Advanced Application Conference, London, UK, 8–11 September 2008; doi:10.1007/978-3-540-85893-5_13. [CrossRef]
21. Yu, Y.C.; Hou, T.W.;Chiang, T.C. Low Cost RFID Real Lightweight Binding Proof Protocol for Medication Errors and Patient Safety. *J. Med. Syst.* **2012**, *36*, 823–828. [CrossRef] [PubMed]
22. Zuo, Y.J. Changing Hands Together: A Secure Group Ownership Transfer Protocol for RFID Tags. In Proceedings of the Hawaii International Conference on System Sciences, Washington, DC, USA, 5–8 January 2010; pp. 1–10.
23. Yang, M.H.; Luo, J.N.; Lu, S.Y. A Novel Multilayered RFID Tagged Cargo Integrity Assurance Scheme. *J. Sens.* **2015**, *15*, 27087–27115. [CrossRef] [PubMed]
24. Sun, H.M.; Ting, W.C.; Chang, S.Y. Offlined Simultaneous Grouping Proof Protocol for RFID EPC C1G2 Tags. In Proceedings of the 2nd International Conference on Computer Science and its Applications, Jeju, Korea, 7–9 October 2013; pp. 1–6.

25. Yen, Y.C.; Lo, N.W.; Wu, T.C. Two RFID-Based Solutions for Secure Inpatient Medeciation Administration. *J. Med. Syst.* **2012**, *36*, 2769–2778. [CrossRef] [PubMed]
26. Jannati H.; Falahati, A. Cryptanaylsis and Enhancement of a Secure Group Ownership Transfer Protocol for RFID Tags. In Proceedings of the International Conference on Global Security, Safety and Sustainability, Thessaloniki, Greece, 24–26 August 2012; pp. 186–193.
27. Gong, L.; Needham, R.;Yahalom, R. Reasoning about belief in cryptographic protocols. In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, FL, USA, 14–16 April 1990; pp. 234–248.
28. Russell, D.M. ; Saldanha, J.P. Five Tenents of Security-Aware Logistics and Supply Chain Operation. *Transp. J.* **2003**, *44*, 44–54.