sensors

MDPI

Article

An RFID-Based Smart Structure for the Supply Chain: Resilient Scanning Proofs and Ownership Transfer with Positive Secrecy Capacity Channels †

Mike Burmester [1], Jorge Munilla [2,*], Andrés Ortiz [2] and Pino Caballero-Gil [3]

[1] Department of Computer Science, Florida State University, Tallahassee, FL 32304, USA; burmester@cs.fsu.edu

[2] Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad de Málaga, 29070 Málaga, Spain; aortiz@ic.uma.es

[3] Facultad de Ciencias, Departamento de Ingeniería de Informática y de Sistemas, Universidad de La Laguna, 38271 Tenerife, Spain; pcaballe@ull.es

* Correspondence: munilla@ic.uma.es; Tel.: +34-952-134-166

† This paper is an extended version of our paper published in Burmester, M.; Munilla, J. Resilient Grouping Proofs with Missing Tag Identification. In Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2016), Canary Islands, Spain, 29 November–2 December; Garca C., Caballero-Gil P., Burmester M., Quesada-Arencibia A., Eds; Ubiquitous Computing and Ambient Intelligence; Springer: Cham, Switzerland, 2016; Volume 10070, pp. 544–555.

Received: 9 May 2017; Accepted: 28 June 2017; Published: 4 July 2017

Abstract: The National Strategy for Global Supply Chain Security published in 2012 by the White House identifies two primary goals for strengthening global supply chains: first, to promote the efficient and secure movement of goods, and second to foster a resilient supply chain. The Internet of Things (IoT), and in particular Radio Frequency Identification (RFID) technology, can be used to realize these goals. For product identification, tracking and real-time awareness, RFID tags are attached to goods. As tagged goods move along the supply chain from the suppliers to the manufacturers, and then on to the retailers until eventually they reach the customers, two major security challenges can be identified: (I) to protect the shipment of goods that are controlled by potentially untrusted carriers; and (II) to secure the transfer of ownership at each stage of the chain. For the former, grouping proofs in which the tags of the scanned goods generate a proof of "simulatenous" presence can be employed, while for the latter, ownership transfer protocols (OTP) are used. This paper describes enhanced security solutions for both challenges. We first extend earlier work on grouping proofs and group codes to capture resilient group scanning with untrusted readers; then, we describe a modified version of a recently published OTP based on channels with positive secrecy capacity adapted to be implemented on common RFID systems in the supply chain. The proposed solutions take into account the limitations of low cost tags employed in the supply chain, which are only required to generate pseudorandom numbers and compute one-way hash functions.

Keywords: RFID; grouping proof; ownership transfer; supply chain; secrecy capacity

1. Introduction

The National Strategy for Global Supply Chain Security [1] identifies two goals for securing supply chains: (1) promote efficient and secure services, and (2) foster resilience. In particular, the infrastructure should be modernized with security mechanisms integrated into supply chain operations to mitigate vulnerabilities.

Radio Frequency Identification (RFID) is a widely deployed technology for supply chain management, inventory, retail operations and more generally automatic identification. A typical

RFID deployment has three main components: tags or transponders, which are electronic data storage devices attached to (or embedded in) objects to be identified; readers or interrogators, that manage tag population, read data from and write data to tags; and a back-end server (or verifier in security applications), which is a trusted entity that exchanges tag information with the readers and processes data according to specific task applications. When combined with the Internet (see Figure 1), RFID technology enables real-time product flow visibility, with information shared at any point in the distribution chain. For example, when a product runs low at the distribution center (detected, for instance, by "smart" shelves with RFID readers), the supplier is automatically alerted to ship more products. Real-time information also allows for accurate ordering. There is no need to keep products piled up in warehouses. Logistics software can be used to trace trucks with GPS (Global Positioning System) locators, while trucks monitor their content with RFID readers. Thus, RFID technology helps to address the three main concerns for efficient supply chain management: (a) inventory inaccuracy, (b) the bullwhip effect (increasing swings in inventory in response to shifts in customer demand along the supply chain), and (c) inventory replenishment rules/policies. The ultimate goal of supply chain management is to optimize the supply chain operation: to deliver goods to the end-customers on time at the lowest cost, while realizing the best profit for the involved agents. This also implies addressing security concerns—in particular, guaranteeing the privacy, integrity and availability of applications, bearing in mind the limited computational capabilities of the employed tags. For the supply chain, low-cost passive UHF (Ultra High Frequency) tags, which operate in the far field with backscatter communication [2], are commonly used. These are computationally constrained and cannot carry out complex cryptographic operations. Readers and verifiers/servers, by contrast, are able to perform complex cryptographic operations.

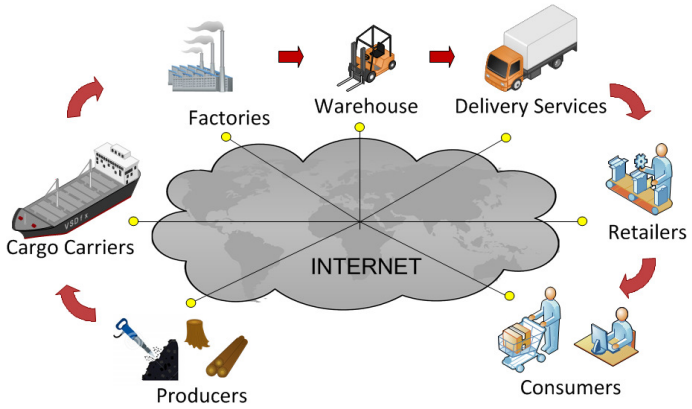


Figure 1. Real-time awareness in the supply chain: flow information is shared at any point of the distribution chain.

As goods move along the supply chain from the suppliers to the manufacturers, and then on to retailers, until eventually they reach the end-customers, two types of agents can be identified: the owners, who have complete ownership rights of the goods, and the carriers, who have delegated ownership rights. Thus, the supply chain can be seen as a series of multiple segments in which the current owner (or the seller) of goods ships the goods to a new owner (the buyer) via a carrier (see Figure 2). Then, when goods arrive, the seller must transfer ownership of the goods to the buyer. The security mechanisms (protocols) described in this paper target both phases of these segments: shipping and ownership transfer. We shall assume that owners are trusted, although possibly curious (knowledge of supply chain information can be used to gain competitive advantage), but this trust does not extend to other parties of the chain (e.g., the carriers).

RFID-tagged objects are typically shipped via (potentially untrusted) carriers in pallets. In such cases, it is important that the owner can periodically check the integrity of a shipment. Tags are

beyond the communication range of the owner so that the common sequential interrogation of the tags cannot be employed here and grouping proofs can be used instead. A grouping proof involves multiple tags generating a proof of simultaneous presence in the range of an RFID reader [3,4] controlled by a potentially untrusted party. While grouping proofs are designed to provide integrity evidence for complete groups that can be verified by the owner, they do not provide any information about incomplete groups. Group codes, which are stored in the tags and work as a forward error correction mechanism, are combined with grouping proofs in this paper to address this issue. Additionally, grouping proofs usually follow a tag-chaining structure, where each tag authenticates the message coming from the previous tag in the chain. This causes availability issues when "alien" tags, not belonging to the pre-defined group, are involved in the protocols, and privacy concerns, as the total number of tags and the reply order are usually leaked. The grouping proof proposed in this paper avoids the use of this chaining structure implementing a two-round protocol with missing tag identification, which prevents the aforementioned problems.

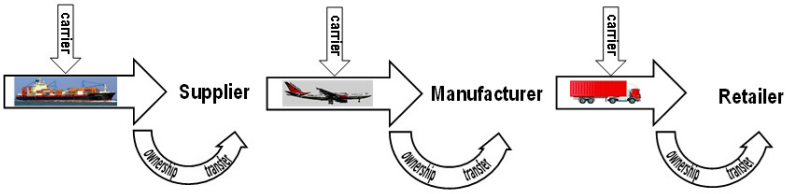


Figure 2. Segments of the supply chain: shipping and ownership transfer.

Ownership Transfer Protocols (OTPs) allow the secure transfer of tag ownership from a current owner to a new owner. Three entities are present in an OTP: the tag $\mathcal{T}$ whose rights are being transferred, the current (or previous, after the transaction) owner who has the initial control of $\mathcal{T}$, and the new owner who will take control of $\mathcal{T}$ when the protocol is completed. OTPs must incorporate security requirements that protect the privacy of the parties. More specifically, apart from providing anonymity and untraceability against external adversaries, ownership transfer protocols should (I) guarantee the privacy of the previous owners (*forward secrecy*) so that subsequent owners (or an adversary) cannot link a specific tagged object with previous communications, even if the current private information stored on the tag is revealed (e.g., by physical attacks), and (II) guarantee the privacy of the new and future owners (*backward secrecy*) so that, once ownership is transferred, previous owners cannot access, or trace, the tagged product. The latter, when using symmetric cryptography, is particularly challenging [5,6] and most proposed solutions only solve it partially either by:

1. Employing a Trusted Third Party (TTP) to break the trust link between the tag and its current owner (e.g., [7,8]), or by,
2. Assuming an Isolated Environment (IsE) (e.g., [9,10]), without any adversarial interference.

In the first approach, although the use of TTPs could be envisaged for this kind of infrastructure (e.g., acting as certificate authorities), most of the OTP protocols proposed in the bibliography for RFID assume symmetric-keys so that TTPs usually share the master key with the tags, becoming the real holders of the tag's rights, while the actual owners just share session keys generated by these TTPs. This leads to a centralized architecture that may not be appropriate when tags belong to different authorities/companies. The second approach assumes a weak threat model and, as claimed in [6]: if such protection is adequate, then there is no need for security. More recently, a novel approach [11] has shown that the privacy of the new owner can be guaranteed by using channels with positive secrecy capacity. Such channels can be implemented with noisy tags controlled by the new owner that obfuscate the communication channel for the previous owner.

This paper focuses on protecting the two phases of the supply chain segments, and as main contributions we:

(1) Extend the notion of a grouping proof of integrity to a broader class of applications where items may be missing. The primary concern of the owner of a shipped pallet is to establish its integrity; however, if some tagged items are missing, then the owner wants a list of the missing items and proof that nothing else is missing (resiliency). Thus, based on the work published in [12], we present a two-round anonymous RFID scanning proof that supports tag privacy such that: (a) the verifier (owner) can authorize an untrusted reader (carrier) to scan a group of tagged items and either generate a proof of integrity, or if some tagged items are missing, identify these and prove that nothing else is missing, (b) the authorization is for one only scanning, (c) tagged items are untraceable while the group is not scanned, and (d) only the verifier (owner) can check the proof: unauthorized inspections or forged proofs will not be accepted.

(2) Extend the implementation of positive secrecy capacity channels for provably secure OTP in [11] by using time-slot modulation, similar to the random-slotted medium access control protocol, to make it possible to implement them without requiring multi-level but binary detection.

These security mechanisms proposed for the segments of the supply chain take into account the particular characteristics of RFID systems such as the vulnerability of the radio channel, the constrained power of devices, the low-cost and limited functionality of tags and the request-response operation mode. In particular, tags are only required to generate pseudorandom numbers and compute one-way hash functions.

The rest of this paper is organized as follows. In Section 2, we provide the background for RFID grouping proofs, group codes and OTPs. Section 3 focuses on the shipment link, discussing grouping proof deployments and capabilities, erasure codes, the threat model and presents, along with our design criteria, a two-round anonymous grouping proof of integrity with missing tag identification. Section 4 addresses security concerns during ownership transfer. We describe a provably secure OTP that uses noisy tags to achieve privacy, and introduce a novel way to implement positive secrecy capacity channels adapted for RFID deployments in the supply chain. Section 5 analyzes this implementation, proving that perfect secrecy is achievable, and providing optimal values for real implementations. Finally, in Section 6, we conclude by summarizing the main results.

2. Background

2.1. Brief Review of Grouping-Proofs

In 2004, Ari Juels defined a new RFID application called a yoking-proof that generates evidence of simultaneous presence of two tags in the range of an RFID reader. This was extended to grouping proofs for multiple tags—see e.g., [13]. Burmester et al. presented in [14], a protocol that achieved anonymity by using randomized pseudonyms for the group identifer, and forward-security by updating the secret keys and the group keys after each session. Huang and Ku [15] presented a grouping-proof for passive low-cost tags that uses a pseudo-random number generator to authenticate flows and a cyclic redundancy code to randomize strings. The protocol has several weaknesses, some of which were addressed by Chien et al. [16] who, in turn, proposed a new grouping-proof. More recently, Liu et al. [3] proposed a grouping-proof for distributed RFID applications with trusted readers. This proof is vulnerable to de-synchronization and privacy leaks [17]. Peris-Lopez et al. [18] proposed guidelines for securing grouping proofs as well as a yoking-proof protocol (for two tags). Most of these protocols propose tag-chaining structures where each tag authenticates the message coming from the previous tag in the chain. This, however, as mentioned, causes privacy problems, as the total number of tags and reply order are leaked, and availability issues when tags that do not belong to the group participate.

For the shipment link of the supply chain, the typical deployment of an RFID grouping proof involves: a pallet P containing a collection of tagged goods, the owner *Own* of P who knows the

private information stored by the tags, and a reader that is controlled by the carrier whose services are contracted by the owner and has physical possession of P—see Figure 3 for an illustration. In this scenario, if the carrier is trusted, then the owner can entrust the carrier with sensitive information so that the carrier can act as the owner by proxy. The integrity can then be checked by authenticating the different tagged products sequentially. This option is also possible if the owner enjoys full connectivity with the carrier so that the owner can communicate with the tags in real time while the carrier just relays the messages, acting as a communication enabler. The problems arise when the carrier is not trusted and: (a) the owner is only willing to give the carrier information that is strictly needed to ensure the efficient monitoring of the transported goods, and (b) the owner and/or carrier do not enjoy full connectivity. In these cases, grouping proofs are employed. A grouping proof involves a collection of tagged objects generating a proof of simultaneous presence. With low-cost RFID tags, symmetric key cryptography is usually employed so that this proof must be checked by a party that shares private information with the scanned tags, namely *Own*.

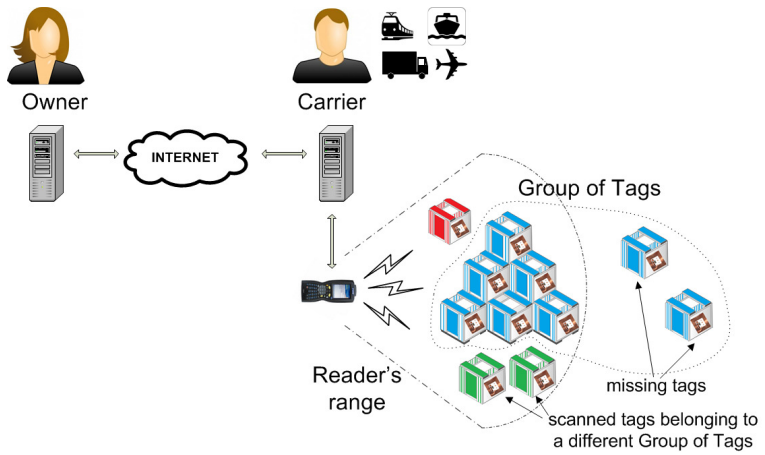


Figure 3. An untrusted carrier can compile a scanning proof of integrity for the tagged goods of a pallet that the owner can verify, and identify any missing tagged goods of the pallet (or, that are beyond the reader's range).

2.2. Brief Review of Group Codes

While grouping proofs provide integrity evidence for complete groups of tags, they do not address incomplete groups. In particular, they do not provide any information about missing tags. Sato et al. [19] proposed a group code that makes it possible to find the identifiers of missing tags without requiring a packaging list or an external database. More specifically, the missing tags are identified by storing redundant information w_i in the memory of each tag. Figure 4 illustrates the write-transmit-read process with forward error correction for supply chain applications. The possible loss of tags is modelled by using an erasure channel. An erasure channel is a memoryless channel that, on inputting a symbol x, outputs symbol x or no symbol at all. Note that the loss of a tag implies not only the loss of its identifying information id_i (when systematic codes are used) but also the loss of the redundancy information w_i.

Sato et al. [20] use Gallager low-density parity check (LDPC) codes for forward error correction [21]. However, the randomised nature of LDPC codes makes it difficult to get specific decoding guarantees. To address this, Su et al. [22] use strongly selective families (SSF). Su and Tonguz [23] propose a variant that uses the Chinese remainder theorem to construct non-regular generating matrices. Another modification proposed by Su [24] uses resolvable transversal designs to generate parity-check matrices and group splitting to improve performance. Mabrouk and Couderc [25] propose a group code that is based on Reed–Solomon (RS) codes. However, the size of the blocks and the partitioning of the redundancy is not optimal. Burmester and Munilla [26,27] analyze the

memory-erasure tradeoff of these group codes and consider optimized approximations for practical settings. They conclude that optimized RS codes are the most efficient from a memory point of view, but impose a higher computational burden on the verifier (reader), particularly when the total number of tagged goods is large. By contrast, LDPC codes are more efficient from a computational point of view, but require considerable more memory that makes them impractical for most RFID applications. Thus, in this paper, we shall assume Reed–Solomon codes to encode tag identifiers.

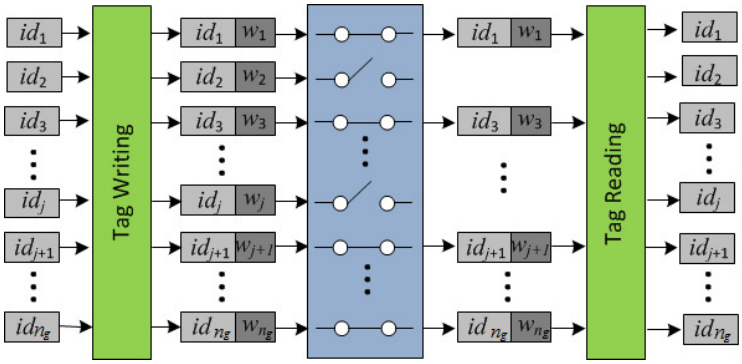


Figure 4. The write-transmit-read process with forward error correction in the supply chain. The loss of tags is modelled using an erasure channel.

2.3. Brief Review of Ownership Transfer Protocols

Ownership Transfer Protocols (OTP) are intended to transfer of ownership rights of a tag $\mathcal{T}$ from a seller or current owner Own_c to a buyer or new owner Own_n. The ownership of a tag usually implies the knowledge of keys that allow to identify and/or access the tag. Before the execution of the OTP, the current owner is the only one that can identify and trace the tag, while after its execution, to guarantee forward secrecy, the tag $\mathcal{T}$ can only be identified and traced by the new owner.

The first OTPs for RFID were presented in 2005 by Saito et al. [28] and Molnar et al. [29]. The former describes two proposals: one with TTP, and another without TTP whose security relies on the short range of the backward channel, assuming that it is difficult for adversaries to eavesdrop on this channel. The latter proposes an OTP that, by using a tree structure to manage tag keys, uses a distributed TTP. Ref. [30] analyzes some vulnerabilities of this scheme and a modification that replaces the TTP with distributed local devices is presented by Soppera and Burbridge [31]. Hash values to protect messages and a keyed encryption function combined with a sort of TTP were used by Osaka et al. [32]. This scheme was later modified by Chen et al. [33] and Japinnen and Hamalainen [34] to prevent Denial of Service (DoS) attacks, and by Yoon and Yoo [35] to assume an IsE where the owners can change the tag's key (some vulnerabilities are described in [36]). RFIDdot, proposed by Dimitriou [37], is an ownership transfer scheme based on random nonces and a keyed encryption function that assumes a private environment where key updates are carried out. An IsE is also assumed by Song and Mitchell [38,39], but they use keyed hash functions and one-time tag identifiers with hash-chains. Ref. [40] defines extended capabilities such as: Tag Assurance, Undeniable Ownership Transfer, Current Ownership Proof, Ownership Delegation, and Authorized Recovery. Ref. [6] proposes two new schemes based on a TTP and IsE, respectively, for ownership transfer of single tags. A version for multiple tags has also been published [41]. More recently, several OTPs that comply with the EPCGen2 standard [42] have been published. These also assume TTPs or IsEs and rely on XOR operations, Cyclic Redundancy Codes (CRC16) and/or Pseudo Random Number Generators as security primitives (e.g., [8,43–45]). Some security issues of such proposals are analyzed in [46].

We note that, to guarantee forward secrecy, most of the ownership transfer protocols proposed in the literature rely either on the use of TTPs, or on the assumption of an IsE. Symmetric-key based OTPs that use TTPs often have a centralized management structure that may not be compatible with the distributed management of RFID systems. For example, the RFID parties (the owners) with possibly

conflicting interests must trust the TTP that manages their tags. On the other hand, the assumption of IsEs where no adversary can interfere, presumes a weak threat model; Ref. [6] claims that if such an environment were available, then no other security protection would be needed. Moreover, most of the proposed protocols cannot be implemented when the seller and the buyer of shipped tagged goods are in different locations. Recently, a provable secure OTP that addresses these issues has been proposed [11]: this employs a channel with positive secrecy capacity to guarantee the privacy of the new owner, without requiring TTPs or IsEs, and a communication model in which the current owner and the new owner can be in different locations. This paper proposes a modification of this protocol adapted to the ordinary RFID readers used in the supply chain that is based on a binary level instead of multi level detection.

3. The Shipment Link

The combined functionality of a grouping proof of integrity with the automatic identification of missing items adds resilience by detecting shrinkages. In this section, we shall describe a pallet scanning proof, defined as a grouping proof with missing tag identification based on work in [12]. In particular, we present an enhanced two-round grouping proof for which the identifiers of the tags are extended to include redundancy, that makes it possible to identify missing tagged objects and prevent availability (when "alien" tags participate) and privacy issues (reply order leakage). Additionally:

1. The owner of the pallet P (e.g., the supplier, manufacturer, retailer, etc.) can authorize an untrusted carrier to inspect P for integrity and identify any missing goods.
2. The authorization is for a certain number of inspections (or limited time) defined by a counter T_s, and the contents of P are untraceable after the authorization expires. In particular, the carrier does not share any private keys with the tags and cannot access or even trace the tags beyond the lifetime of the counter T_s.
3. The carrier can generate a grouping proof of integrity for the pallet P that (only) the owner can verify if no goods are missing; if some goods are missing, then the carrier can (a) identify the missing goods without requiring a packing list (or an external database) and (b) generate a scanning proof of presence for the remaining goods.
4. The grouping proof is generated only if the tags of the group were scanned simultaneously (during the same session defined by the activation time of the tags) within a time window defined by T_s.

For the design of the scanning proofs, the following assumptions are made:

a *The tags of a pallet are not compromised*. This does not mean that tags cannot be compromised; but if they are, then the corroborating evidence generated for a scanning proof is compromised.
b *Simultaneity*. This is defined in terms of counters or timestamps provided by the owner.
c *Batch connectivity*. The owner does not enjoy permanent connectivity with the carrier and is restricted to: (a) broadcasting a challenge that is valid for a (short) time span and, (b) checking responses from tags that are compiled and sent from time to time by untrusted readers.
d *Balanced loading*. The tags of a pallet have similar hardware capabilities and the computation load per tag is balanced.
e *Messages must include destination information (possibly private) to allow for unicast/multicast communication.* This is sometimes neglected by designers, but it is particularly important for checking anonymity: each message must contain information that allows tags to decide if they are the intended recipient.

3.1. Extended Identifiers with Redundancy

We shall use a Reed–Solomon $RS(n,k)$ code over $\mathbb{F}_{2^m}$, $2 \leq m \leq 16$ (in compliance with RFC 6865 [47]) to encode the identifiers $(id_1, \ldots, id_{n_g})$ of a collection of n_g RFID tags, so as to recover

up to $s_t = (n-k)/(n/n_g)$ missing identifiers id_i. For this purpose, we rearrange the source data $x = id_1 \| \cdots \| id_{n_g}$, which is an $n_g\ell$-bit string, where ℓ is the binary length of the identifiers id_i, into k blocks of size m: $x = (x_i, \ldots, x_k)$ (so $x_i \in \mathbb{F}_{2^m}$), and then encode x to get an n block codeword $y = (y_1, \ldots, y_n)$ (depending on the implementation, some blocks x_i can be padded with zeros if necessary). The codeword y is then partitioned into n_g pieces of size n/n_g blocks, denoted ID_i, which are stored (written) to the memory of each tag_i. The ID_i contain the identifying information id_i as well as redundancy w_i (the systematic property of linear error-correcting codes allows the separation of source and redundancy information) needed to recover erased blocks. This code will recover up to $s = n - k$ blocks y_i, which corresponds to $s_t = floor((n-k)/(n/n_g))$ missing identifiers id_i. To identify the missing tagged products, the data collected (read) from the tags is used to generate a codeword y^* with erasures. To decode y^*, we need to order the scanned identifiers ID_i correctly so that y and y^* agree on all non-erased positions. For this purpose, control information is used: the information stored in each tag is extended to include a few bits that define its order i when it was encoded in the codeword y.

3.2. Scanning Proof Description

Let $P = \{tag_1, \ldots, tag_{n_g}\}$ be a collection of tags attached to the goods of pallet and h denote a cryptographic hash function. The owner of each collection P stores the tuple: $(T, k, \{(k_i, ID_i)\}_{i\in[1:n_g]})$, where T is a counter value, k is a key for the collection of tags, k_i is the private key of tag_i, and ID_i is the extended identifier of tag_i that includes its identifying code EPC_i and redundant information w_i used to recover missing tags. Each tag_i of P stores in non-volatile memory: (k, k_i, ID_i) and a counter T_i that is initialized to the same value T_0 for all tags of P. The carrier's reader R initially does not share any information with the tags of P, and the process starts when R receives from the owner a scanning request (T, T', k'), where: T is a fresh value of a counter, $T' = h(k, T)$ is a session authenticator, and $k' = h(k, T')$ is the session key. Then, a two rounds protocol takes place in which the reader and the tags generate a scanning proof with missing tag identification (see Figure 5):

Round 1. The reader R of the carrier broadcasts to all tags in its range: (T, T'), and sets a timer. Each tag_i in the range of R computes $k' = h(k, T')$ and checks the correctness of T' by verifying that $T' = h(k, T)$ and that the counter value $T > T_i$. If any of these fail, tag_i returns two random values. Otherwise, it updates its counter to T, draws a random number r_i and computes its authenticator $r_i' = h(k', r_i)$. Then, it sends (r_i, r_i') to R and sets a timer. The received nonces r_i are used by the reader R to identify (singulate) tags in this session (session pseudonyms). R checks the correctness of every r_i by verifying that $r_i' = h(k', r_i)$, and if this holds, R stores them in a list $L1$. On timeout, R computes the request $S = h(T, r_{j_1}, \ldots, r_{j_u})$, where $\{j_1, \ldots, j_u\} \subseteq \{1, \ldots, n_g\}$ are the indices of the tags of pallet P that were scanned, and its authenticator $S' = h(k', S)$. Thus, the first round incorporates the randomness provided by the owner's challenge T and the randomness r_i provided by the interrogated tags. This prevents replay attacks. The participation of "alien" tags does not affect the execution (availability is guaranteed) and information about the total number of tags or reply order is not leaked because tags do not follow any chaining structure. The scanning period is defined by the scanning request T of the reader, and simultaneity by the validity period of the nonces r_i that is set by the scanned tags.

Round 2. The reader R broadcasts the authenticated request (S, S') to all tags in its range. Each tag_i in the range of R that has not timed out, checks that $S' = h(k', S)$ and if so, it computes: $m_i = h(k', r_i, ID_i)$ and its session authenticator $m_i' = h(k', m_i)$, as well as a "proof of presence during the session" $p_i = h(k_i, r_i, S)$ (a message authentication code), and its authenticator $p_i = h(k', p_i)$. Then, it encrypts its identifier ID_i with the "one-time-pad" key m_i' to get $m_i' \oplus ID_i$, sends to R: $(m_i,\ m_i' \oplus ID_i,\ p_i,\ p_i')$, and timeouts. The reader R computes $m_i' = h(k', m_i)$ and retrieves the identifiers ID_i. Then, it checks (by exhaustive search)

that $m_i = h(k', r_i, ID_i)$ for some value r_i in the list $L1$, and that $p_i' = h(k', p_i)$. If these are correct, R stores the identifiers ID_i in a list $L2$. On timeout, R checks that $|L1| = |L2|$ (that all tags singulated in Round 1 responded in Round 2), and if so, compiles the proof $W = (T, ID_{j_1}, \ldots, ID_{j_u}, r_{j_1}, \ldots, r_{j_u}, h(p_{j_1}, \ldots, p_{j_u}))$ as evidence that the tags were scanned. Otherwise, R aborts the protocol. Then, using the control information, R checks that the cardinality of the group coincides with $|L2|$. If not, R finds the missing EPC_is by using the redundant information stored in the retrieved identifiers ID_j, provided that this is within the correction capabilities of the implemented forward error correction mechanism (i.e., the number of missing tags $(n_g - u)$ is no more than $s_t = (n-k)/(n/n_g)$). If there are no missing tags, then W becomes a grouping proof of integrity for pallet P that the reader R sends to the owner *Own*. Otherwise, R retrieves the list of identifiers EPC_i, $i \in \{1, \ldots, n_g\} \backslash \{j_1, \ldots, j_u\}$, of the missing goods, and sends *Own* the scanning proof $W^* = (T, ID_{j_1}, \ldots, ID_{j_u}, r_{j_1}, \ldots, r_{j_u}, h(p_{j_1}, \ldots, p_{j_u}))$ of presence for the remaining goods.

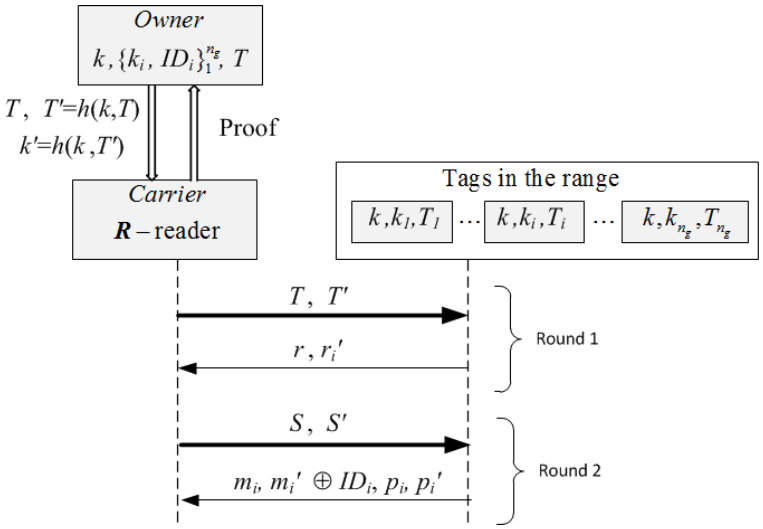


Figure 5. The two rounds of the anonymous scanning proof with missing tag identification.

To validate the scanning proof, *Own* first uses the value T to retrieve the tuple $(k, \{k_i, ID_i\}_{i \in [1:n_g]})$. Then, *Own* computes $S = h(T, r_{j_1}, \ldots, r_{j_u})$ using the values r_{j_i} given by the carrier and the corresponding $p_{j_i} = h(k_{j_i}, r_{j_i}, S)$. Finally, *Own* checks that the value of $h(p_{j_1}, \ldots, p_{j_u})$ is correct.

3.3. Security Discussion

1. *Traceability attacks (privacy).* Unlinkability is related to the capability of linking interrogations after physical tracking is temporarily interrupted. Different formal models can be found in the literature (e.g., [48–50]). Intuitively, a protocol guarantees unlinkability, if no adversary can decide with advantage better than negligible whether two transmitted messages from different protocol executions are linked to the same tag $\mathcal{T}$. In the scanning proof, tag_i is untraceable because, in every session, it updates its counter T_i and will draw a fresh (pseudo) random number r_i after responding to the reader's challenge T. Consequently, the responses of the same tag in different interrogations look random to an observer and cannot be linked. Tags do not follow a sequence to reply so that information about the order of a tag cannot be leaked.
2. *Replay Attacks.* The use of the counter T prevents replay attacks: if an adversarial reader re-uses T, the tags that received it earlier will have updated their counter and not respond.

3. *Impersonation attacks*. Impersonation attacks on tagged goods are prevented by using private keys k_i. Impersonation attacks on a reader will not yield a valid proof since the owner will only accept proofs from authorized readers that have been given (T, T', k') .
4. *Forged proofs.* The values $p_i = h(k_i, r_i, S)$ can only be generated by someone who knows k_i; i.e., tag_i and the owner. Values p_i from different sessions cannot be used to compile a proof since they involve the session nonces r_i of interrogated tags and the challenge of the reader R $(= h(T, r_{j_1}, \ldots, r_{j_u}))$ that depends on the counter T which specifies the validity time window. Note that all tags set timers in Round 1 of the protocol and will not respond after timeout.
5. *De-synchronization attacks (DoS attacks)*. The adversary cannot compute a valid pair of values (T, T') because this requires knowledge of the key k. On the other hand, if a protocol execution completes successfully, then all tags will share the same counter value. No tag will accept a previously used T. However, tags will accept future values of T, not necessarily the next value, so that even if they do not share the same counter value (e.g., because of an interrupted interrogation), there are no synchronization concerns.

Two possible bottlenecks for tag populations can be identified: (I) in time terms, the exhaustive search in Round 2 to check that $m_i = h(k', r_i, ID_i)$ for values r_i in list $L1$, and (II) in memory overhead, the extra bits stored in the tags as required by grouping codes. Only tags that know k' are included in $L1$ so that in normal conditions, for a low number of missing tags, these factors should not be a problem, even for large groups. However, when the rate of missing tags increases, the second factor could limit tag population [26]. In fact, for groups of about 100 tags, 12 and 144 extra bits are required for missing tag rates of 10% and 60%, respectively. Therefore, the tag population is expected to be limited not by the anonymization, but by the use of grouping codes and the expected missing tags rate. Note, however, that very large groups are not usually assumed for grouping proofs since, unlike the two-round protocol presented here, most of the previously published grouping proofs follow a sequential tag-chaining structure where each tag in the group authenticates a message coming from the previous tag in the chain.

4. Ownership Transfer Link

The provable secure OTP described in [11] captures spatiotemporal requirement so that it is appropriate for applications such as the supply chain where the seller and buyer are in different locations. The protocol has two phases. Initially, the tag $\mathcal{T}$ whose ownership is going to be transferred shares a private key k_0 with Own_c. In the first phase, a new fresh key k_1 is agreed between Own_c, $\mathcal{T}$ and Own_n. For this purpose, Own_c first exchanges privately the key k_1 with the tag. Then, Own_c sends privately to Own_n the key k_1. Finally, $\mathcal{T}$ and Own_n confirm mutually knowledge of k_1 and $\mathcal{T}$ updates its private key k_0 to k_1. This completes the first phase (see [11] for more details). Forward secrecy is guaranteed because k_1 does not provide Own_n with any information about k_0. However, the key k_1 shared by Own_n and $\mathcal{T}$ is also known to Own_c, who can use it to keep tracking the tag, violating backward privacy. The second phase of the protocol addresses this issue.

The second phase involves a Key Update Protocol (KUP) that exploits signal features at the physical layer to create a channel with positive secrecy capacity. In particular, "noisy tags" controlled by the new owner are used to obfuscate the previous owner's channel so that the new owner and the tag can exchange a new secret k_2 without the previous owner being able to access it [51]. Before going into technical details, to understand the idea behind this protocol, let us consider a scenario that involves a crowd of people all calling out "yes" or "no" (the noisy tags) to obfuscate the decision ("yes" or "no") of an oracle (the tag) from an eavesdropper (the previous owner). The eavesdropper will only know with absolute certainty the decision of the oracle if all calls (crowd + oracle) were "yes" or "no". In the other cases, the eavesdropper only knows the decision with varying degrees of certainty depending on the tally of the calls (crowd + oracle). However, anyone who knows the tally of the crowd (the new owner) can disambiguate the oracle's decision by subtracting the tallies. In this way, a person can send one bit ("yes" or "no") of information privately to a listener who knows the bit values

of the calls made by the crowd, while the eavesdropper only gets the bit with a certain probability. The difference between the information that the listener and eavesdropper get is called the *secrecy capacity*. The noisy tags create a communication channel with positive secrecy capacity that can be used by the new owner and the tag to exchange information privately. Thus, in the KUP proposed in [11], tag $\mathcal{T}$ and the noisy tags $\mathcal{T}^*$ are queried by the new owner with a random number r, and all respond at the same time with S and S_i^*, respectively. The new owner and the eavesdropper (the previous owner) receive the sum of the strings S and S_i^*, but only the new owner who knows S_i^* is able to extract S. The new owner and the tag then use this value to compute a new key k_2 (the full proof is in [11]).

The physical addition of the signals corresponding to S and S_i^* in the channel is the basis of the wiretap channel (see Figure 6). X and N_i^* are random input variables taking values s, s_i^* in the input alphabet $\mathcal{X}$. Y is an output random variable taking value y in the output alphabet $\mathcal{Y}$, and $p(y|s, s_1^*, \ldots, s_{n_t}^*)$ is the transition probability of the channel. Tag $\mathcal{T}$ transmits the message $X = x$ to the new owner (the intended receiver) with the help of n_t noisy tags, in the presence of the current (previous) owner Own_c, who acts as a passive eavesdropper. The wiretap channel is a stochastic encoder of X with output Y. Y is input to the maximum a posteriori probability (MAP) estimators of Own_n and Own_c, but while Own_c only knows the value of Y, Own_n also knows the values of the inputs $N_1^*, \ldots, N_{n_t}^*$. The wireless medium is assumed noiseless, so that the estimate $X = x$ of Own_n is correct while the estimate $X = \overline{x}$ of Own_c is degraded by the stochastic encoder. The degradation is quantified by the conditional entropy $H(X|Y)$:

$$H(X|Y) = \sum_{j=0}^{|\mathcal{X}|-1} \sum_{k=0}^{|\mathcal{Y}|-1} -p(x_j, y_k) \cdot \log_2 p(x_j|y_k) \ . \tag{1}$$

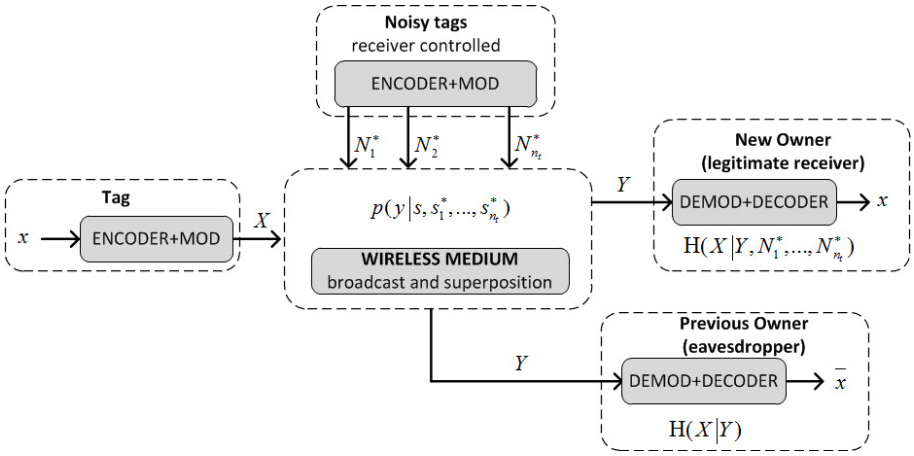


Figure 6. A model for the wiretap channel with noisy tags.

The capacity of the eavesdropper channel (Own_c's) is defined as $C_{eav} = H(X) - H(X|Y)$. The secrecy capacity for the wiretap model is $C_s = C_{main} - C_{eav}$, where C_{main} is the capacity of the main channel (Own_ns). In the noiseless case, we have $C_{main} = H(X)$, and therefore the secrecy capacity coincides with the conditional entropy of the eavesdropper $C_s = H(X|Y)$. In particular, if the source is binary and equiprobable ($H(X) = 1$) and the length of S and S_i^* is n/C_s bits, then Own_c knows $C_{eav} \cdot n/C_s = (1 - C_s) \cdot n/C_s$ bits of S, while the remaining n bits are unknown. These bits are used by $\mathcal{T}$ and Own_n to compute the new key k_2 so that, once the KUP is completed, Own_c has no control over the tag $\mathcal{T}$ and cannot trace it.

In [11], the performance for different values of n_t is analyzed, proving that $\lim_{n_t \to \infty} H(X|Y^{n_t}) = H(X)$ (perfect secrecy), and showing that $n_t = 3$, with $C_s = 0.78$, is a good compromise for ease of implementation and performances. Figure 7 shows the outputs of $\mathcal{Y}$ for the KUP described in [11] with $n_t = 3$. For this implementation, the tags are assumed to respond

quasi-simultaneously (or without distinguishable delays) and the readers to demodulate output symbols with multiple amplitude levels. This is equivalent, in the example above, to the listener being able to identify the specific number of people that responded simultaneously "yes" and "no". Such implementations, however, may not be practical for the supply chain where binary RFID readers are preferred. That is, the listener only needs to detect if someone is saying "yes" or "no", but not the number of people that simultaneously are saying it. In the next section, we shall describe a modified version of this KUP that is adapted for the common binary RFID systems employed in the supply chain. More specifically, this employs a channel with positive secrecy capacity that circumvents the need for tags to respond quasi-simultaneously, and for readers to distinguish amplitude levels.

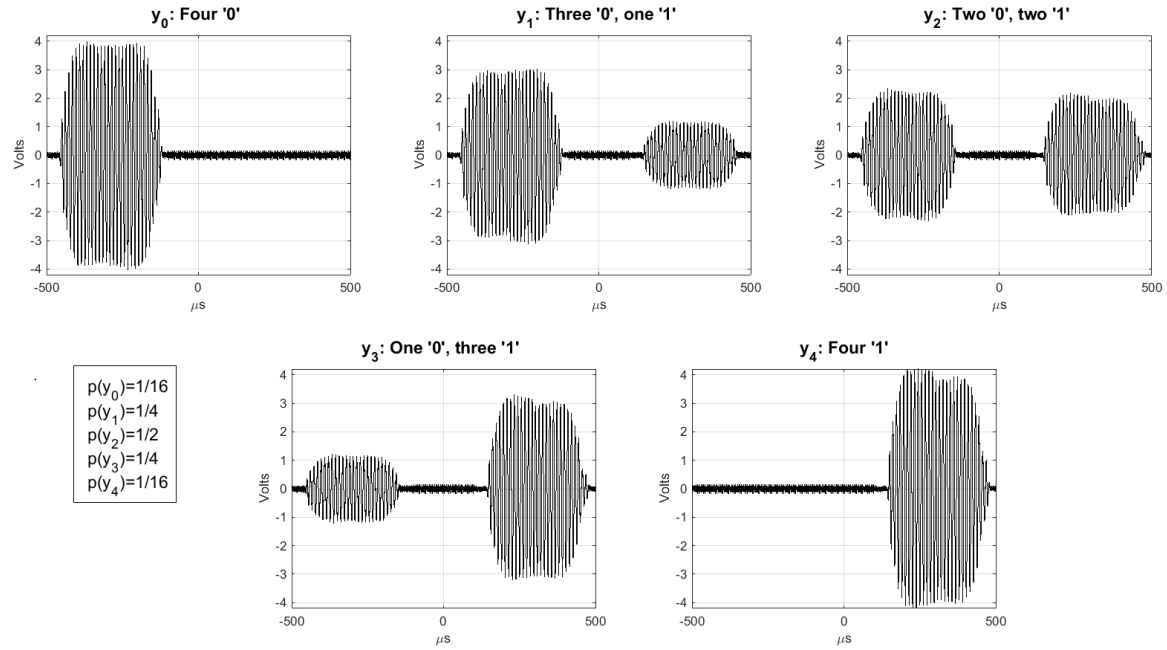


Figure 7. Examples of the output alphabet $\mathcal{Y} = \{y_0, y_1, y_2, y_3, y_4\}$ of the modulation described in [11] with $n_t = 3$.

5. A KUP That Uses a Positive Secrecy Capacity Channel Adapted for the Supply Chain

5.1. A Positive Secrecy Capacity Channel Based on Modified Random-Slotted Modulation

The EPCGen2 standard [42] specifies the Random-Slotted Collision Arbitration algorithm, which is a modified version of the Frame Slotted Aloha protocol, for random access in RFID systems. In this protocol, the reader sends to all tags in its range a query Q and tags pick a random value in the range $(0, \ldots, 2^Q - 1)$ to respond in that slot. As Miller coding is used, when two or more tags chose the same slot to transmit different values, a collision is detected. Otherwise, the reader detects the transmitted information. Based on this scheme, we shall now describe how to implement a channel with positive secrecy capacity that does not require multilevel detection.

In the proposed system, bits are transmitted in frames of f slots. To transmit a bit, a tag picks one of these slots and uses it to transmit the bit. Figure 8 illustrates an example with three frames when $f = 4$, $n_t = 2$ and Manchester coding is used. For the sake of simplicity, we have preferred here to use Manchester coding rather than Miller coding as this depends on the previous bit, but both coding allow detecting collisions. In the example, $\mathcal{T}$ transmits the bit string '101' using the slots 2, 3 and 4, respectively. Likewise, the noisy tag $\mathcal{N}_1$ transmits the bit string '010' using the slots 1, 3 and 2, and the noisy tag $\mathcal{N}_2$ transmits the '101' using the slots 4, 2 and 4. By knowing the values transmitted by the noisy tags, the reader can disambiguate the values transmitted by $\mathcal{T}$ in the first and the second frames, but it cannot determine it in the third frame because it cannot determine if $\mathcal{T}$ sent 0 in the slot 2 or 1 in

the slot 4. Thus, we have, in the first frame, three singletons $s_s = 3$, in the second frame one singleton $s_s = 1$ and a reconcilable collision $s_c = 1$, and in the third frame a singleton and an irreconcilable collision, but the readers wrongly detects them as two singletons $s_s = 2$. It is true that if all tags send the same value (e.g., $\mathcal{N}_1$ had also sent another 1 in the slot 3), then the reader can determine the value sent by $\mathcal{T}$, but this does not contribute to the secrecy capacity of the channel as the eavesdropper would also know with certainty the value sent by $\mathcal{T}$. As a result, frames with $s_s + 2s_c < n_t + 1$ are discarded. When these output symbols are removed, the original output alphabet $\mathcal{Y}$ is reduced to an alphabet $\mathcal{Y}'$ of size:

$$|\mathcal{Y}'| = \begin{pmatrix} 2f \\ n_t + 1 \end{pmatrix}, \text{with } 2f \geq n_t + 1. \tag{2}$$

We next analyze the performance of this channel for two coding schemes. In the first, as in the example above, the order of the slot is not used in the code so that $|\mathcal{X}| = 2$; $H(X) = 1$. In the second, by contrast, the order of the slot is used to code the input (Pulse Position Modulation) so that $|\mathcal{X}| = 2f$; $H(X) = \log_2 2f$. The secrecy capacity for the first coding can be computed as follows (see the Appendix for more details):

$$C_{s1} = 2 \begin{pmatrix} 2f \\ 2f - n_t - 1 \end{pmatrix}^{-1} \sum_{k=1}^{|\mathcal{Y}'|} \frac{W_k^0}{n_t + 1} \log_2 \frac{n_t + 1}{W_k^0}, \tag{3}$$

where W_k^0 is the number of 0's in the output symbol $y'_k \in \mathcal{Y}'$. The secrecy capacity for the second coding is:

$$C_{s2} = \log_2(n_t + 1). \tag{4}$$

Figure 9 compares, for different values of n_t and f, C_{s1}, C_{s2}. Secrecy capacity increases with the number of noisy tags but not with the number of slots. Perfect secrecy, $C_{s1} = H(X)$ and $C_{s2} = H(X)$, is achieved when n_t reaches its limit $n_t = 2f - 1$. This may mislead us to infer that reducing f improves the efficiency of the system, which is not true because the probability of frame rejection, or frame retransmission, for having irreconcilable collisions does increase when reducing f. Thus, perfect secrecy capacity when $n_t = 2f - 1$ involves very high probabilities of retransmission. The probability of retransmission p_r is plotted in Figure 10 and can be computed as:

$$p_r = 1 - \frac{(n_t + 1)!}{(2f)^{n_t+1}} \begin{pmatrix} 2f \\ 2f - n_t - 1 \end{pmatrix}. \tag{5}$$

Frame retransmissions not only increase the communication cost but also the computational cost, since a new value has to be generated (otherwise, the adversary can track tags by checking the repeated values in irreconcilable collisions). Thus, the time to generate and transmit a symbol in a frame slot is:

$$T = f \cdot t_f + t_c \cdot log_2 |\mathcal{X}|, \tag{6}$$

where t_f is the duration of a frame slot and t_c the processing time to generate a new bit. This time T is then multiplied by a factor $(1 - p_r)^{-1}$ to take into account the probability of retransmission:

$$T_t = T (1 - p_r) \sum_{i=1}^{\infty} i \cdot p_r^{i-1} = \frac{T}{(1 - p_r)}. \tag{7}$$

To understand how these different parameters conjugate, the number of secret bits transmitted per time unit t_f can be computed as:

$$C_f = \frac{C_s(1 - p_r)}{100\, log_2 |\mathcal{X}| + f + 1}, \tag{8}$$

where $t_c \approx 100 t_f$ has been assumed. Accurate approximations for the latter assumption can be made in each particular case taking into account the particular characteristics of the tag, the quality of the communication link, and the selected operation mode.

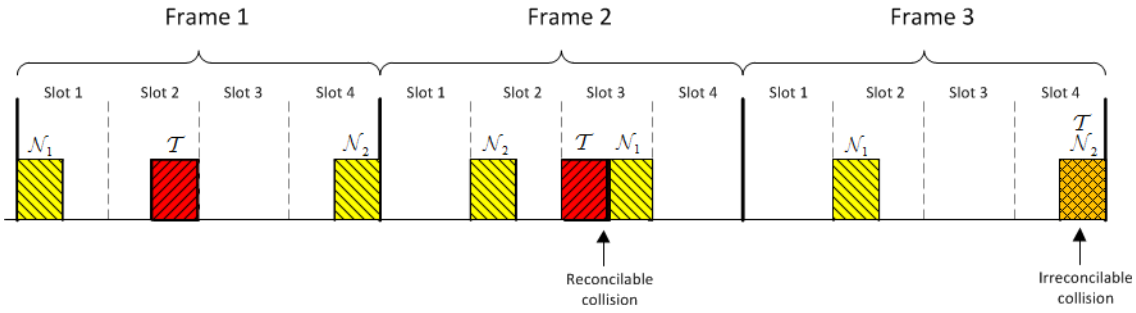


Figure 8. Examples of output symbols with the modified random-slotted modulation mechanism.

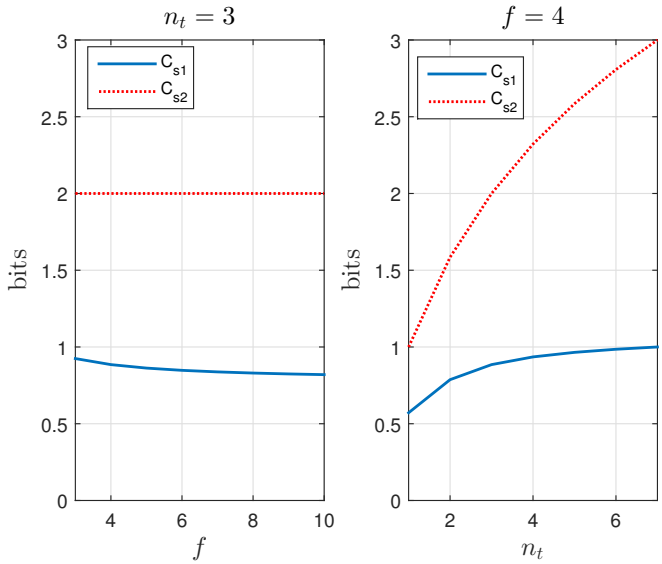


Figure 9. Comparison of the secrecy capacities C_{s1} ($|\mathcal{X}| = 2$) and C_{s2} ($|\mathcal{X}| = 2f$).

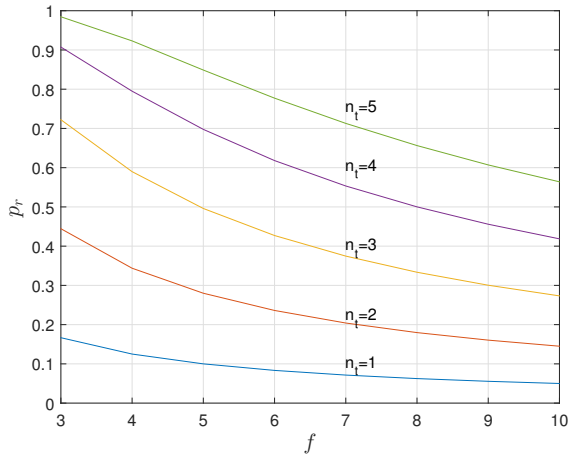
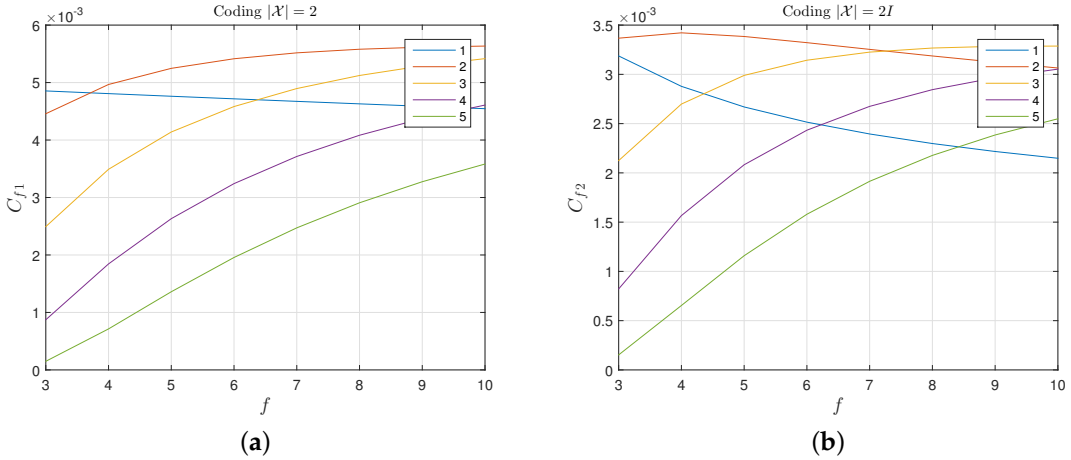


Figure 10. The probability of retransmission p_r increases with n_t and decreases with f.

Figure 11 shows the number of secret bits transmitted per unit time t_f when different values of n_t and f are employed. As a result, the coding with $|\mathcal{X}| = 2$ is more efficient than the coding with $|\mathcal{X}| = 2f$. This happens because, although, with the latter, more bits are transmitted per frame, the rate $C_s/H(X)$ is lower, and therefore a smaller fraction of the generated bits becomes part of the transmitted secret information. For the code with $|\mathcal{X}| = 2$, we identify the set of parameters: $n_t = 2$, $f = 8$ as offering an optimal compromise between performance and easiness of implementation. For these parameters: $C_s = 0.73$, $p_r = 0.18$.



(**a**) (**b**)

Figure 11. The number of secret bits transmitted per time unit C_f for a coding: (**a**) with $|\mathcal{X}| = 2$, and (**b**) with $|\mathcal{X}| = 2f$.

5.2. A KUP Based on a Positive Secrecy Capacity Channel with Modified Random-Slotted Modulation

The parties are: the reader $\mathcal{R}$ of the new owner, a tag $\mathcal{T}$, and n_t noisy tags $\mathcal{T}_i^*$, $i = 1, \ldots, n_t$. $\mathcal{R}$ shares with $\mathcal{T}$ a private key k_1 and with each $\mathcal{T}_t^*$ a private key k^*. The goal in this protocol is for $\mathcal{T}$ to update privately the key k_1 with a fresh key k_2 of length n bits while the previous owner, who also knows k_1, eavesdrops on the communication. The description of the protocol is as follows (see Figure 12):

1. $\mathcal{R}$ broadcasts r, r', where r is a nonce and $r' = h(k_1, r)$: $\mathcal{R} \rightarrow \mathcal{T}$, $\{\mathcal{T}_i^*\}_{i=1}^{n_t} : r, r' = h(k_1, r)$.
2. Upon receiving this, $\mathcal{T}$ and $\mathcal{T}_i^*$ check that r, r' are correct, and if so, generate a random bit string S and the bit strings $S_i^* = h(k^*, r, id_i^*)$, of length $L = \lceil nF_g/(C_s(1-p_r)) \rceil$, where $F_g \geq 1$ is a guard factor (e.g., $F_g = 1.1$). Then $\mathcal{T}$ and $\mathcal{T}_i^*$ broadcast these bit strings using a frame for each bit and picking random slots within such frames as described previously (Section 5.1): $\mathcal{T}$ and $\{\mathcal{T}_i^*\}_{i=1}^{n_t} \rightarrow \mathcal{R} : S$ and $\{S_i^*\}_{t=i}^{n_t}$.
3. $\mathcal{R}$ receives the added signals of S and $\{S_i^*\}_{i=1}^{n_t}$. First, $\mathcal{R}$ identifies the frames with irreconcilable collisions (by checking that $s_s + 2s_c < n_t + 1$) and stores their indices in a list U. Let $\overline{U} = \{1, 2, ..., L\} \setminus U$ be the set of frames without irreconcilable collisions. $\mathcal{R}$ generates a bit string S_s of length $|\overline{U}|$ with the values of S for the frames with indices in $\overline{U}$, and a bit string M of length L, whose i-th bit is 0 if $i \in U$ and 1 if $i \in \overline{U}$. Note that the expected value of $|S_s|$: $E[|\overline{U}|] = L \cdot (1 - p_r) = nF_g/C_s$, is greater than n/C_s. However if $|S_s| < n/C_s$, then $\mathcal{R}$ generates another random number r and repeats the first step, extracting a new S_s, and concatenating it to the previous one until $|S_s| \geq n/C_s$. Then, $\mathcal{R}$ computes $k_2 = h(k_1, r, S_s)$, and sends $M, M' = h(k_2, M)$: $\mathcal{R} \rightarrow \mathcal{T} : M, M' = h(k_2, M)$.
4. $\mathcal{T}$ generates S_s by taking the bits of S where M is equal to 1, computes $k_2 = h(k_1, r, S_s)$ and checks the correctness of the received M'. If this is not correct, then $\mathcal{T}$ aborts the protocol; otherwise,

it computes $h(k_2, M')$ and sends this to $\mathcal{R}$ to confirm that the updating was correct: $\mathcal{T} \rightarrow \mathcal{R} : h(k_2, M')$.

5. $\mathcal{R}$ checks the received message. If correct, the protocol is completed, and the current owner informs the previous one that the process has been completed. Otherwise, $\mathcal{R}$ resends the values M, M' in Step 3 to checks if $\mathcal{T}$ has updated its key. If not, the KUP is repeated.

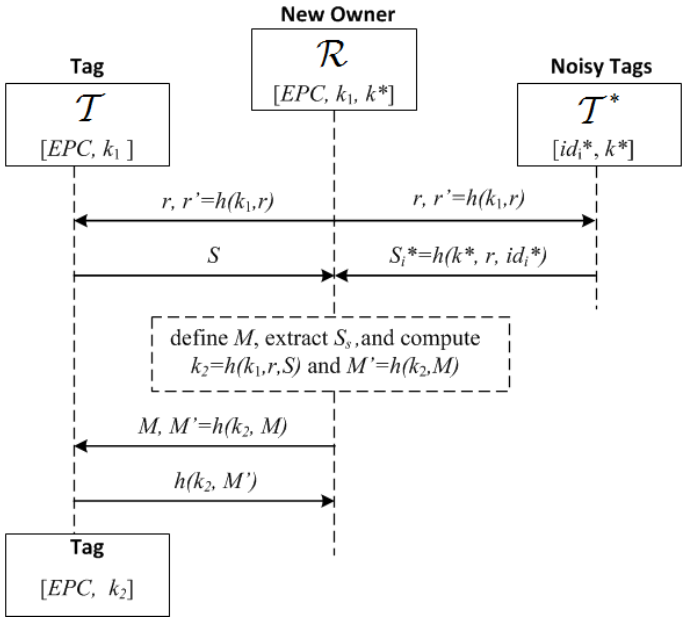


Figure 12. A Key Update Protocol based on a positive secrecy capacity channel with modified random-slotted modulation.

Security against an eavesdropper comes from the fact that even if the key k_1 is known, the key k^* of the noisy tags is not known, and therefore, the adversary cannot filter out S_i^* to get S_s and compute k_2. In particular, on average, the eavesdropper knows $C_{eav} \cdot |\overline{U}| = (1 - C_s) \cdot |\overline{U}|$ bits of S_s, which means that on average the remaining $C_s|\overline{U}| = nF_g$ bits of S_s are unknown. As an example, assume a security parameter $n = 128$ bits, a guard factor $F_g = 1.1$, and the parameters suggested in the previous section: $n_t = 2$ and $f = 8$, with $C_s = 0.73$, $p_r = 0.18$. Then, $L = 236$ bits, $\lceil n/C_s \rceil = 176$ and the probability that the first step is repeated (i.e., $|\overline{U}| < 176$) is lower than 0.2%:

$$\sum_{i=0}^{175} \binom{236}{i} (1 - 0.18)^i 0.18^{236-i} = 0.0017. \tag{9}$$

6. Conclusions

There are two major challenges for protecting smart supply chains when RFID systems are used for situational awareness: to protect the shipment of goods (in particular, reduce inventory shrinkage and prevent unauthorized tracking), and to secure ownership transfer. In this paper, we have addressed both challenges. We have presented an anonymous scanning proof with missing tag identification that can be used to authorize untrusted carriers to track pallets of tagged goods, check their integrity, and identify any missing items without requiring a packing list. The authorization is for only one scanning, after which the goods are untraceable. For ownership transfer, backward privacy has been addressed using a novel approach based on positive secrecy channels with modified random-slotted modulation. This approach does not require TTPs or an IsE. An analysis of this channel is carried out, leading to optimal implementations with two noisy tags and eight frame slots.

Acknowledgments: This material is based in part upon work supported by: (a) the National Science Foundation under Grant Nos. CNS 1347113, DGE 1538850, 1565215 and DUE 1241525, and (b) the Spanish MINECO and FEDER under project TEC2014-54110-R. Funds for covering the costs to publish in open access come from these grants.

Author Contributions: No significant distinction can be made; the five co-authors have worked together and contributed equally to the reported research and writing of the paper.

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

Appendix A

The value of the secrecy capacity (Equation (1)) can be obtained from the value of the following terms: $|\mathcal{X}|, |\mathcal{Y}|, p(x_j, y_k)$ and $p(x_j|y_k)$. To compute it, we shall consider three cases: (I) the general case when irreconcilable collisions are not removed, (II) the case when irreconcilable collisions are removed, and (III) the case when input data is also coded using the frame slot order. We shall assume that $2f \geq n_t + 1$.

Appendix A.1. General Case: $|\mathcal{X}| = 2$, with Irreconcilable Collisions

In this case, $p(x_j) = \frac{1}{2}$, $j = 0, 1$. Then,

$$|\mathcal{Y}| = \left(\begin{array}{c} 2f + n_t \\ n_t + 1 \end{array} \right), \tag{A1}$$

and, $p(y_k)$, $k = 0, \ldots, |\mathcal{Y}| - 1$, is computed using a multinomial coefficient:

$$p(y_k) = \left(\frac{1}{2f}\right)^{n_t+1} \left(\begin{array}{c} n_t + 1 \\ Z_1^0, Z_2^0, \ldots, Z_f^0, Z_2^1, Z_2^1, \ldots, Z_f^1 \end{array} \right), \quad k = 0, \ldots, |\mathcal{Y}| - 1, \tag{A2}$$

where Z_i^0 and Z_i^1 are the numbers of tags that used slot i to respond with 0 and 1, respectively, so that $n_t = \sum_{i=1}^{f} (Z_i^0 + Z_i^1)$.

Let $W_k^0 = \sum_{i=1}^{f} (Z_i^0 > 0)$ and $W_k^1 = \sum_{i=1}^{f} (Z_i^1 > 0)$. The joint probability $p(x_j, y_k)$ can be computed as: $p(x_j, y_k) = p(x_j|y_k) \cdot p(y_k)$, where

$$p(x_j|y_k) = \frac{W_k^j}{W_k^0 + W_k^1}, \quad j = 0, 1, \; k = 0, \ldots, |\mathcal{Y}'| - 1. \tag{A3}$$

Appendix A.2. $|\mathcal{X}| = 2$, without Irreconcilable Collisions

Let S_u be the set of symbols of $\mathcal{Y}$ for which there are irreconcilable collisions. The new alphabet $\mathcal{Y}'$ is obtained by removing S_u from $\mathcal{Y}$: $\mathcal{Y}' = \mathcal{Y} \backslash S_u$. The size of $\mathcal{Y}'$ can be computed as:

$$|\mathcal{Y}'| = \left(\begin{array}{c} 2f \\ n_t + 1 \end{array} \right). \tag{A4}$$

The probability $p(y'_k)$ of a new output symbol is: $p(y'_k) = p(y_k) \cdot p_s^{-1}$, where p_s is the probability of an output symbol not having an irreconcilable collision. The probability p_s is the complement of the sum of the probabilities of the symbols in S_u:

$$p_s = 1 - \sum_{y_k \in S_u} y_k = \frac{2f!}{(2f)^{n_t+1} (2f - n_t - 1)!} = \frac{(n_t + 1)!}{(2f)^{n_t+1}} \left(\begin{array}{c} 2f \\ 2f - n_t - 1 \end{array} \right). \tag{A5}$$

In this case, when applying the binomial coefficient (Equation (A2)), Z_i^0 and Z_i^1 are always either 0 or 1, so that the y'_k are equiprobable:

$$p(y'_k) = \left(\frac{1}{2f}\right)^{n_t+1} \frac{(n_t+1)!}{p_s} = \left(\begin{array}{c} 2f \\ 2f-n_t-1 \end{array} \right)^{-1}, \quad k = 0, \ldots, |\mathcal{Y}'| - 1. \tag{A6}$$

Finally, the conditional probability $p(x_j|y'_k)$ is:

$$p(x_j|y'_k) = \frac{W_k^j}{n_t+1}, \quad j = 0, 1, \; k = 0, \ldots, |\mathcal{Y}'| - 1. \tag{A7}$$

Appendix A.3. $|\mathcal{X}| = 2f$*, without Irreconcilable Collisions*

Input data are coded using the slot order and bit value (0 or 1), so that x_j, $j = 0, \ldots, f-1$, correspond with 0 in the slot $j+1$, while x_j, $j = f, \ldots, 2f-1$, is coded responding with 1 in the slot $j+1-f$. Thus, $|\mathcal{X}| = 2f$, $H(X) = \log_2(2f)$ and $p(x_j) = \frac{1}{2f}$, $j = 0, \ldots, 2f-1$. The output alphabet is still $\mathcal{Y}'$, and the value of $p(y'_k)$ is the same (Equation (A6)). The value $p(x_j|y'_k)$ changes: $p(x_j|y'_k)$, $j = 0, \ldots, f-1$, is now 0 for those y'_k with $Z_{j+1}^0 = 0$, and $1/(n_t+1)$ otherwise. Likewise, $p(x_j|y'_k)$, $j = f, \ldots, 2f-1$, is 0 for the y'_k with $Z_{j+1-f}^1 = 0$ and $1/(n_t+1)$ otherwise. Thus:

$$\sum_{k=0}^{|\mathcal{Y}'|-1} p(x_j|y'_k) \log_2 p(x_j|y'_k) = \left(\begin{array}{c} 2f-1 \\ n_t \end{array} \right) \frac{1}{n_t+1} \log_2 \frac{1}{n_t+1}, \tag{A8}$$

and a closed expression for the secrecy capacity can be obtained:

$$H(X|Y) = -|\mathcal{X}| \sum_{k=0}^{|\mathcal{Y}'|-1} p(x_j|y'_k) \log_2 p(x_j|y'_k) p(y'_k) = \log_2(n_t+1). \tag{A9}$$

References

1. The White House, Office of the Press Secretary. Available online: https://obamawhitehouse.archives.gov/sites/default/files/national_strategy_for_global_supply_chain_security.pdf (accessed on 3 July 2017).
2. Paret, D. *RFID and Contactless Smart Card Applications*; John Wiley & Sons Ltd.: Chichester, UK, 2005.
3. Liu, H.; Ning, H.; Zhang, Y.; He, D.; Xiong, Q.; Yang, L.T. Grouping-Proofs-Based Authentication Protocol for Distributed RFID Systems. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 1321–1330.
4. Burmester, M.; Munilla, J. *Chapter RFID Grouping-Proofs in Security and Trends in Wireless Identification and Sensing Platform Tags: Advancements in RFID*; Information Science Reference-IGI Gobal: Hershey, PA, USA, 2013; pp. 89–119.
5. Vullers P. Secure Ownership and Ownership Transfer in RFID Systems. Master's Thesis, Eindhoven University, Eindhoven, The Netherlands, 2009.
6. Kapoor, G.; Piramuthu, S. Single RFID Tag Ownership Transfer Protocols. *IEEE Trans. Syst. Man Cybern. Part C* **2012**, *42*, 164–173.
7. Osaka, K.; Takagi, T.; Yamazaki, K.; Takahashi, O. An Efficient and Secure RFID Security Method with Ownership Transfer. In *Computational Intelligence and Security*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4456, pp. 778–787.
8. Sundaresan, S.; Doss, R.; Zhou, W.; Piramuthu, S. Secure ownership transfer for multi-tag multi-owner passive RFID environment with individual-owner privacy. *Comput. Commun.* **2015**, *55*, 112–124.
9. Song, B. RFID Tag Ownership Transfer. In Proceedings of the Workshop on RFID Security—RFIDSec'08, Budapest, Hungary, 9–11 July 2008.
10. Lei, H.; Cao, T. RFID Protocol Enabling Ownership Transfer to Protect against Traceability and DoS Attacks. In Proceedings of the First International Symposium on Data, Privacy, and E-Commerce, Washington, DC, USA, 1–3 November 2007; pp. 508–510.

11. Munilla, J.; Burmester, M.; Peinado, A.; Yang, G.; Susilo, W. RFID Ownership Transfer with Positive Secrecy Capacity Channels. *Sensors* **2017**, *17*, 53.
12. Burmester, M.; Munilla, J. Resilient Grouping Proofs with Missing Tag Identification. In Proceedings of the 10th International Conference on Ubiquitous Computing and Ambient Intelligence, UCAmI 2016, San Bartolomé de Tirajana, Gran Canaria, Spain, 29 November–2 December 2016; pp. 544–555.
13. Piramuthu, S. On Existence Proofs for Multiple RFID Tags. In Proceedings of the IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing—SecPerU, Sydney, Australia, 14–18 March 2006.
14. Burmester, M.; de Medeiros, B.; Motta, R. *Provably Secure Grouping-Proofs for RFID Tags*; Grimaud, G., Standaert, F.X., Eds. (CARDIS 2008); Lecture Notes in Computer Science; Springer: London, UK, 2008; Volume 5189, pp. 176–190.
15. Huang, H.H.; Ku, C.Y. A RFID Grouping Proof Protocol for Medication Safety of Inpatient. *J. Med. Syst.* **2009**, *33*, 467–474.
16. Chien, H.Y.; Yang, C.C.; Wu, T.C.; Lee, C.F. Two RFID-based Solutions to Enhance Inpatient Medication Safety. *J. Med. Syst.* **2011**, *35*, 369–375.
17. Burmester, M.; Munilla, J. *Distributed Group Authentication for RFID Supply Management*; Technical Report E-Print; International Association for Cryptological Research. Available online: http://eprint.iacr.org/2013/779 (accessed on 3 July 2017).
18. Peris-Lopez, P.; Orfila, A.; Hernandez-Castro, J.C.; van der Lubbe, J.C.A. Flaws on RFID grouping-proofs. Guidelines for future sound protocols. *J. Netw. Comput. Appl.* **2011**, *34*, 833–845.
19. Sato, Y.; Igarashi, Y.; Mitsugi, J.; Nakamura, O.; Murai, J. Identification of missing objects with group coding of RF tags. In Proceedings of the IEEE International Conference on RFID, Orlando, FL, USA, 3–5 April 2012; pp. 95–101.
20. Sato, Y.; Mitsugi, J.; Nakamura, O.; Murai, J. Theory and performance evaluation of group coding of RFID tags. *IEEE Trans. Autom. Sci. Eng.* **2012**, *9*, 458–466.
21. Gallager, R.G. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *IT-8*, 21–28.
22. Su, Y.S.; Lin, J.R.; Tonguz, O.K. Grouping of RFID Tags via Strongly Selective Families. *IEEE Commun. Lett.* **2013**, *17*, 1120–1123.
23. Su, Y.S.; Tonguz, O.K. Using the Chinese Remainder Theorem for the Grouping of RFID Tags. *IEEE Trans. Commun.* **2013**, *61*, 4741–4753.
24. Su, Y.S. Extended Grouping of RFID Tags Based on Resolvable Transversal Designs. *IEEE Signal Process. Lett.* **2014**, *21*, 488–492.
25. Ben Mabrouk, N.; Couderc, P. EraRFID: Reliable RFID systems using erasure coding. In Proceedings of the 2015 IEEE International Conference on RFID, Tokyo, Japan, 16–18 September 2015; pp. 121–128.
26. Burmester, M.; Munilla, J. Tag Memory-Erasure Tradeoff of RFID Grouping Codes. *IEEE Commun. Lett.* **2016**, *20*, 1144–1147.
27. Burmester, M.; Munilla, J. Performance Analysis of LDPC-Based RFID Group Coding. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 398–402.
28. Saito, J.; Imamoto, K.; Sakurai, K. Reassignment Scheme of an RFID Tag's Key for Owner Transfer. In *EUC Workshops*; Enokido, T., Yan, L., Xiao, B., Kim, D., Dai, Y.S., Yang, L.T., Eds.; Lecture Notes in Computer Science; Springer: Nagasaki, Japan, 2005; Volume 3823, pp. 1303–1312.
29. Molnar, D.; Soppera, A.; Wagner, D. A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In Proceedings of the Workshop on Selected Areas in Cryptography (SAC 2005), Kingston, ON, Canada, 11–12 August 2005; Volume 3897.
30. Avoine, G.; Dysli, E.; Oechslin, P. Reducing Time Complexity in RFID Systems. In Proceedings of the 12th International Conference on Selected Areas in Cryptography, Kingston, ON, Canada, 11–12 August 2005; pp. 291–306.
31. Soppera, A.; Burbridge, T. Secure by default: The RFID acceptor tag (RAT). In *Workshop on RFID Security—RFIDSec'06*; Ecrypt: Graz, Austria, 2006.
32. Osaka, K.; Takagi, T.; Yamazaki, K.; Takahashi, O. An Efficient and Secure RFID Security Method with Ownership Transfer. In Proceedings of the 2006 International Conference on Computational Intelligence and Security, Guangzhou, China, 3–6 November 2006; Volume 2, pp. 1090–1095.

33. Chen, H.B.; Lee, W.B.; Zhao, Y.H.; Chen, Y.L. Enhancement of the RFID Security Method with Ownership Transfer. In Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, 15–16 January 2009; pp. 251–254.
34. Jappinen, P.; Hamalainen, H. Enhanced RFID Security Method with Ownership Transfer. In Proceedings of the International Conference on Computational Intelligence and Security, Suzhou, China, 13–17 December 2008; Volume 2, pp. 382–385.
35. Yoon, E.J.; Yoo, K.Y. Two Security Problems of RFID Security Method with Ownership Transfer. In Proceedings of the IFIP International Conference on Network and Parallel Computing, Shanghai, China, 18–21 October 2008; pp. 68–73.
36. Kapoor, G.; Piramuthu, S. Vulnerabilities in some recently proposed RFID ownership transfer protocols. *IEEE Commun. Lett.* **2010**, *14*, 260–262.
37. Dimitriou, T. rfidDOT: RFID delegation and ownership transfer made simple. In Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, Istanbul, Turkey, 22–25 September 2008; pp. 1–8.
38. Elkhiyaoui, K.; Blass, E.O.; Molva, R. ROTIV: RFID Ownership Transfer with Issuer Verification. In Proceedings of the 7th International Conference on RFID Security and Privacy, Amherst, MA, USA, 26–28 June 2011; pp. 163–182.
39. Song, B.; Mitchell, C.J. Scalable RFID security protocols supporting tag ownership transfer. *Comput. Commun.* **2011**, *34*, 556–566.
40. Ng, C.Y.; Susilo, W.; Mu, Y.; Safavi-Naini, R. Practical RFID Ownership Transfer Scheme. *J. Comput. Secur.* **2011**, *19*, 319–341.
41. Kapoor, G.; Zhou, W.; Piramuthu, S. Multi-tag and Multi-owner RFID Ownership Transfer in Supply Chains. *Decis. Support Syst.* **2011**, *52*, 258–270.
42. EPC Global UHF Air Interface Protocol Standard Generation2/Version2. Available online: https://www.gs1.org/epcrfid/epc-rfid-uhf-air-interface-protocol/2-0-1 (accessed on 3/July/2017).
43. Chen, C.L.; Lai, Y.L.; Chen, C.C.; Deng, Y.Y.; Hwang, Y.C. RFID Ownership Transfer Authorization Systems Conforming EPCglobal Class-1 Generation-2 Standards. *Int. J. Netw. Secur.* **2011**, *13*, 41–48.
44. Sabaragamu Koralalage, K.H.S.; Reza, S.M.; Miura, J.; Goto, Y.; Cheng, J. POP Method: An Approach to Enhance the Security and Privacy of RFID Systems Used in Product Lifecycle with an Anonymous Ownership Transferring Mechanism. In Proceedings of the 2007 ACM Symposium on Applied Computing, Seoul, Korea, 11–15 March 2007; pp. 270–275.
45. Chen, C.L.; Huang, Y.C.; Jiang, J.R. A secure ownership transfer protocol using EPCglobal Gen-2 RFID. *Telecommun. Syst.* **2013**, *53*, 387–399.
46. 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.
47. Roca, V.; Cunche, M.; Lacan, J.; Bouabdallah, A.; Matsuzono, K. Simple Reed-Solomon Forward Error Correction (FEC) Scheme for FECFRAME. Available online: http://www.rfc-editor.org/info/rfc6865 (accessed on 4 July 2017).
48. Avoine, G. *Adversarial Model for Radio Frequency Identification*; Technical Report; Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC): Lausanne, Switzerland, 2005.
49. Juels, A.; Weis, S.A. Defining Strong Privacy for RFID. *ACM Trans. Inf. Syst. Secur.* **2009**, *13*, doi:10.1145/1609956.1609963.
50. Vaudenay, S. On Privacy Models for RFID. In Proceedings of the Advances in Cryptology – ASIACRYPT 2007, Kuching, Malaysia, 2–6 December 2007; Volume 4833, pp. 68–87.
51. Castelluccia, C.; Avoine, G. Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags. In Proceedings of the International Conference on Smart Card Research and Advanced Applications—CARDIS, Tarragona, Spain, 19–21 April 2006; Domingo-Ferrer, J., Posegga, J., Schreckling, D., Eds.; Lecture Notes in Computer Science; Springer: Tarragona, Spain, 2006; Volume 3928, pp. 289–299.