Article

RFID Ownership Transfer with Positive Secrecy Capacity Channels

Jorge Munilla [1,*], Mike Burmester [2], Alberto Peinado [1], Guomin Yang [3] and Willy Susilo [3]

[1] Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad de Málaga, Málaga 29071, Spain; apeinado@ic.uma.es
[2] Department of Computer Science, Florida State University, Tallahassee, FL 32306, USA; burmester@cs.fsu.edu
[3] School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia; gyang@uow.edu.au (G.Y.); wsusilo@uow.edu.au (W.S.)
* Correspondence: munilla@ic.uma.es

Academic Editor: Leonhard M. Reindl
Received: 9 October 2016; Accepted: 23 December 2016; Published: 29 December 2016

Abstract: RFID ownership transfer protocols (OTPs) transfer tag ownership rights. Recently, there has been considerable interest in such protocols; however, guaranteeing privacy for symmetric-key settings without trusted third parties (TTPs) is a challenge still unresolved. In this paper, we address this issue and show that it can be solved by using channels with positive secrecy capacity. We implement these channels with noisy tags and provide practical values, thus proving that perfect secrecy is theoretically possible. We then define a communication model that captures spatiotemporal events and describe a first example of symmetric-key based OTP that: (i) is formally secure in the proposed communication model and (ii) achieves privacy with a noisy tag wiretap channel without TTPs.

Keywords: RFID; ownership transfer; trusted third party; RFID; EPCglobal Gen2

1. Introduction

Radio frequency identification (RFID) is a widely-deployed technology for supply-chain and inventory management, retail operations and more generally automatic identification. Most of these applications need to be secured.

Ownership transfer protocols (OTPs) allow the secure transfer of tag ownership from a current owner to a new owner. Three different entities are present in an OTP: the tag $\mathcal{T}$ whose rights are being transferred, the current 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 both the new and the previous owner of the tag. For RFID applications privacy addresses anonymity that protects the identity of tags and untraceability that prevents interrogations (partial or completed) of a tag being linked. Formal definitions for secure ownership and ownership transfer are provided by van Deursen et al. [1], while several theoretical models have been proposed in the literature that address the privacy of RFID systems [2–5].

Several OTPs that address security issues have been proposed. However, preventing a previous owner from accessing the key(s) of a tag whose ownership was transferred is still an unsolved problem when symmetric-key techniques are used [6,7]. The current approach for privacy is to either employ a trusted third party (TTP) to break the trust link between a tag and its owner (e.g., [8,9]), or an isolated environment (ISE) (e.g., [10,11]) without any adversarial interference. The first approach is centralized and not appropriate when tags belong to different authorities/companies. In fact, the TTP can be considered as the real holder of the tag's rights, while the different owners have simply delegated ownership. The second approach assumes a weak threat model and, as claimed in [7]: if such protection is adequate, then there is no need for security. Our main contributions in this paper are to:

(1) Define a communication model for ownership transfer that addresses spatiotemporal connectivity (Section 3). Many OTPs do not specify the communication setup and assume channels that are impractical for RFID settings.
(2) Provide a theoretical analysis of wiretaps with noisy tags (Section 4), show how these could be implemented and prove that perfect secrecy is achievable.
(3) Present an OTP that is provably secure in this communication model and that uses a wiretap channel with noisy tags to achieve privacy (Section 5). This is the first example of symmetric-key-based OTP that does not require TTPs or an ISE. GNYlogic and strand spaces [12–15] are used in the Appendix A for the security analysis.

2. Background

2.1. Definition and Security Requirements

Tag ownership can be defined as the ability to identify and/or access the tag, which in turn usually implies knowledge of private keys stored on the tag. Ownership transfer protocols enable the transfer of ownership rights of a tag $\mathcal{T}$ from the current owner Own_c, or seller, to a new owner Own_n or buyer. At the beginning of the OTP, the seller is the only entity that can identify and trace $\mathcal{T}$, while when the OTP is completed, $\mathcal{T}$ can only be identified and/or traced by the buyer. A TTP is usually deployed to manage this ownership transfer.

We next list some specific security requirements for OTPs:

Unlinkability or untraceability. An adversary that physically tracks tags can easily determine which executions are linked. This cannot be prevented. Unlinkability is related to the capability of linking interrogations after this physical tracking is temporarily interrupted. Different formal models can be found in the literature (e.g., [2–4]). Intuitively, a protocol guarantees unlinkability or privacy if no adversary can decide with advantage better than negligible whether two messages taken from different protocol executions belong to the same tag or not.

Privacy of Own_n (backward secrecy): The current owner Own_c cannot identify $\mathcal{T}$ once ownership rights are transferred to the new owner Own_n.

Privacy of Own_c (forward secrecy): Once ownership rights of $\mathcal{T}$ are transferred to the new owner Own_n, past communications between $\mathcal{T}$ and previous owners cannot be traced by an adversary (or subsequent owners), even if the current private information stored on $\mathcal{T}$ is revealed (e.g., by physical attacks).

OTPs are sometimes designed [10,16,17] to provide extended capabilities such as: tag assurance, undeniable ownership transfer, current ownership proof, ownership delegation and authorized recovery.

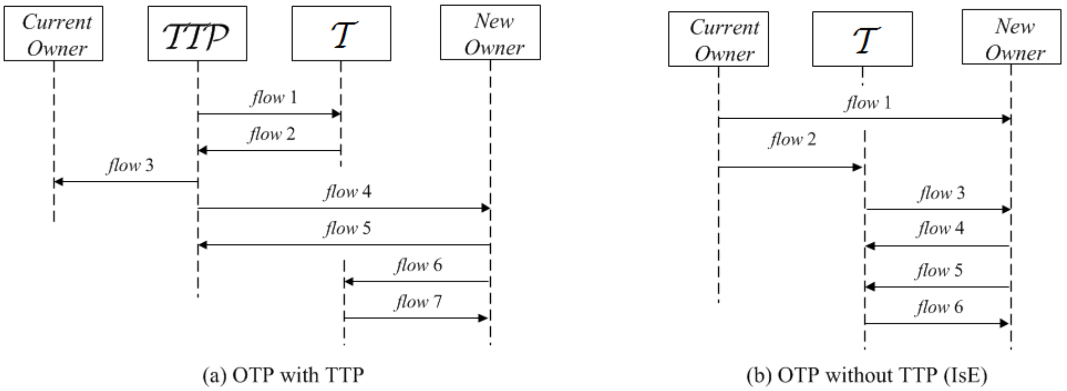
2.2. Related Work

We only review the most relevant symmetric-key-based OTPs for RFID. Saito et al. [18] and Molnar et al. [16] presented in 2005 the first OTPs for RFID applications. Saito et al. proposed two protocols: one with and one without TTP. The security of the latter is based on the short range of the backward channel and assumes that it is hard for adversaries to eavesdrop on this channel. Molnar et al. proposed a scheme with TTP to manage tag keys by using a tree structure. Some vulnerabilities of this scheme are discussed in [19]. Soppera and Burbridge [20] modified Molnar et al.'s scheme by replacing the TTP with distributed local devices called RFID acceptor tags. Osaka et al. [21] used a kind of TTP with hash values to protect messages and a keyed encryption function for ownership transfer. Chen et al. [22] and Japinnen and Hamalainen [23] modified Osaka et al.'s scheme to prevent DoS attacks. Yoon and Yoo [24] also modified Osaka et al.'s scheme, by assuming that owners are able to change the tag's key in an ISE. Their scheme had some vulnerabilities described in [25]. Dimitriou [26] proposed RFIDdot, an ownership transfer scheme based on random nonces and a keyed encryption function, making the assumption that key updates are performed in a private environment. More recently, Song and Mitchell [27,28] also assumed an ISE, but used keyed hash functions and one-time tag identifiers with hash chains. Kapoor and Piramuthu proposed two new schemes [7] based on a TTP and ISE respectively for the transfer of single tags, while a variant of these protocols for multiple tags has also been published [29]. Finally, several schemes have recently been proposed that comply with the EPCGen2 [30] standard for low-cost tags in the UHF band. These again assume TTPs or ISE and combine simple XOR operations, Cyclic Redundancy Codes (CRC16) and/or use the on-board PRNG as the security primitive (e.g., [9,31–33]). The security problems of some of these have been described recently [34].

Motivation: Comparison with Previous Works

As observed, the ownership transfer protocols proposed in the literature rely either on the use of TTPs or the assumption of an ISE. Typically, TTPs have a centralized management 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 is an assumption of a weak adversary model: if such an environment were available, then no other security protection would be needed [7]. This paper proposes a key exchange protocol that addresses the new owner's privacy concerns without resorting to either TTPs or an ISE.

The discussed protocols also use communication models that are sometimes impractical for real-life scenarios. To illustrate this, let us consider the two protocols proposed in [7]: one with TTP, the other without TTP (but with an ISE), whose flows are shown in Figure 1. In the first, Figure 1a, the TTP does not use a reader to communicate with tag $\mathcal{T}$, but communicates directly (Flows 1–2). This begs the question: if such a TTP were installed in the buyer's or seller's location, what trust issues would arise if the transferred goods belong to different authorities. In the second protocol, Figure 1b, $\mathcal{T}$ interacts first with the current owner (the seller, Flow 2) and then with the new owner (the buyer, Flows 3–6). However if something goes wrong (Flow 6 is not received correctly), then the process must be repeated from the beginning. This implies that the buyer and the seller must be available during the transaction, which restricts the possible transaction scenarios to one location (e.g., to a shop). In this paper, we define a communication model where tags can only communicate through readers. This leads to designs of protocols with, if deployed, centralized TTP infrastructures and, in contrast to the examples described above, that allow the seller and buyer to be in different physical locations.



(a) OTP with TTP (b) OTP without TTP (IsE)

Figure 1. Example sketches of ownership transfer protocols (OTPs) with trusted third parties (TTPs) (**a**) and without TTPs (isolated environment) (**b**) [7].

3. A Communication Model for RFID Ownership Transfer

3.1. Entity Capabilities

High-level entities include RFID readers, servers and TTPs. In general, these are able to perform complex cryptographic operations, such as asymmetric encryption/decryption and digital signatures/verification.

RFID tags: In this paper, we are only concerned with UHF passive tags that operate in the far field [35], which are the most common for supply chain applications. These work at higher distances than tags with inductive coupling, but the delivered power is low; therefore, not too complex (lightweight) cryptographic tools should be used [36]. Low price is also a common requirement, and therefore, tamper-resistant shielding and on-board clocks cannot be usually assumed.

3.2. Communication Model

This is defined in terms of its channels with security features, such as privacy and integrity, and connectivity (availability).

3.2.1. Privacy/Integrity Channels

Between high-level entities (readers, servers or TTPs): These can be considered secure, since fully-fledged cryptographic techniques can be used.

Between readers and tags: By contrast, these are particularly vulnerable; they are wireless (the adversary can eavesdrop and block/modify/inject messages), and tags can only implement lightweight cryptographic mechanisms. Passive tags can only communicate with active entities that are physically close and provide them with energy: i.e., RFID readers.

3.2.2. Connectivity

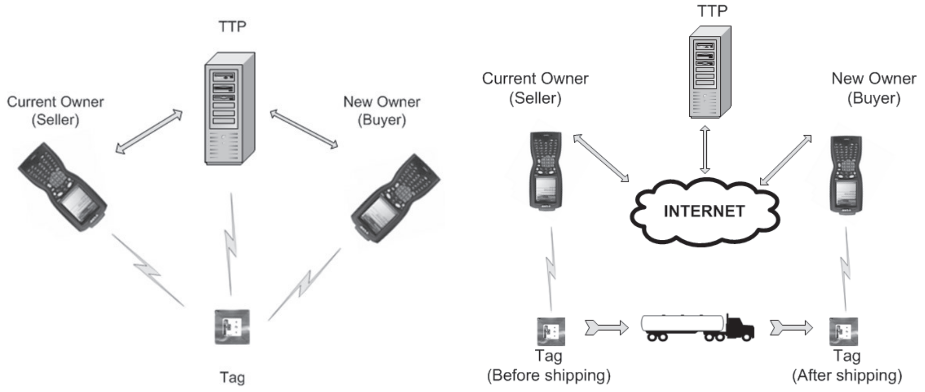
Connectivity is a function of space and time. As far as we know, OTPs proposed in the literature do not discuss spatiotemporal connectivity issues, though several (e.g., [7,9,17]) assume channels that allow high-level parties, including a TTP (e.g., [7]), to communicate with a tag $\mathcal{T}$ in real time during the execution of the OTP: for example, to restart the protocol if it fails. This implies that $\mathcal{T}$ must be physically close to the corresponding high-level parties during the execution of the protocol, which in many practical scenarios may not be the case. Suppose for example that a client purchases RFID-tagged items for tracking and counterfeit prevention via the Internet. The seller dispatches the items, and when these reach the destination, the client requests the transfer of ownership rights. In this case, ownership transfer takes place in a different location from the seller's location, and a different connectivity model is needed, where the seller cannot communicate with the tags at this stage (likewise, buyers cannot communicate with tags at the beginning of the transaction). We also need a spatiotemporal TTP network infrastructure in which TTPs may have to communicate in real time (as in [7]). Figure 2 illustrates the differences between the traditional and the extended communication model.

Let $\mathcal{R}1$, $\mathcal{R}2$, TTP be the readers of Own_c, Own_n, TTP, $\mathcal{T}$ a tag, a, b be OTP parties and $\exists\,(a \overset{t}{\leftrightarrow} b)$, $\exists\,(a \overset{t}{\Leftrightarrow} b)$ stand for "there exists a channel at time t between a, b", "there exists a secure channel at time t between a, b", respectively. When t is not indicated, continuous connectivity is assumed. We formally define the connectivity requirements of the OTP model by the relations:

1 $\exists\,(\mathcal{R}1 \Leftrightarrow \mathcal{R}2) \;\wedge\; \exists\,(\mathcal{R}1 \Leftrightarrow TTP) \;\wedge\; \exists\,(\mathcal{R}2 \Leftrightarrow TTP)$,

2 $\left.\begin{array}{l} \exists\,(\mathcal{R}1 \overset{t}{\leftrightarrow} \mathcal{T}) \;\text{ for } t_0 \le t < t_1 \\ \exists\,(\mathcal{R}2 \overset{t}{\leftrightarrow} \mathcal{T}) \;\text{ for } t_2 \le t < t_3 \end{array}\right\}$ with $\mathrm{t}_1 \le \mathrm{t}_2$.

Thus, a TTP, if deployed, can only communicate with tags $\mathcal{T}$ via readers $\mathcal{R}1$, $\mathcal{R}2$.



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

Figure 2. OTP communication models. (**a**) Basic model (static); (**b**) Dynamic model.

4. A Wiretap Channel with Positive Secrecy Capacity

To guarantee the privacy of a new owner Own_n of a tag $\mathcal{T}$ and prevent the previous owner Own_c from accessing $\mathcal{T}$, Own_n and $\mathcal{T}$ must agree on a fresh key in the presence of Own_c: that is, with Own_c a potential eavesdropper. Note that Own_c has full knowledge of the private keys of $\mathcal{T}$. We shall show that by using Wyner's wiretap channel [37] with noisy tags, we can achieve positive secrecy.

The fundamental property of the superposition of the wireless medium can be pitted against eavesdropping by using interference at the physical layer to degrade communication. Degrading is implemented via reader-controlled interferers called noisy tags. Noisy tags were first used by Juels et al. [38] to protect consumers from unwanted RFID scanning. Later, Castellucia and Avoine [39] used noisy tags for sharing secret keys, which however only addresses passive adversaries since authentication is not ensured. We shall assume that noisy tags do not present any special features, so any tag can become a noisy tag. If more sophisticated noisy tags are available, then implementations with better performance can obviously be achieved.

We use the following notation: X, Y, N are random variables taking values x, y, n in the alphabets $\mathcal{X}, \mathcal{Y}, \mathcal{N}$, respectively. Figure 3 depicts our model of a wiretap channel with input alphabets $\mathcal{X}, \mathcal{N}_1, \ldots, \mathcal{N}_{n_T}$, output alphabet $\mathcal{Y}$ and transition probabilities $p(y|x, n_1, \ldots, n_{n_T})$.

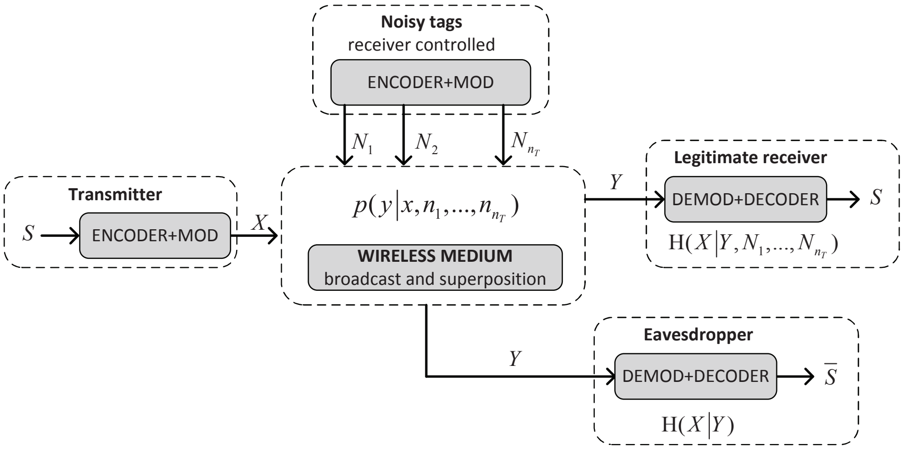


Figure 3. A model for the wiretap channel with noisy tags.

Tag $\mathcal{T}$ transmits the message S (coded as X) to the new owner Own_n (the intended receiver) with the help of n_T noisy tags, in the presence of the current owner Own_c, who acts as a passive eavesdropper. The wiretap channel can be seen as a stochastic encoder of X with output alphabet $\mathcal{Y}$. The variable 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 $N_1, \ldots, N_{n_T}$. Thus, if we assume the wireless medium is noiseless, then the estimate $S = s$ of Own_n is correct, while the estimate $\overline{S} = \overline{s}$ of Own_c is degraded by the stochastic encoder. This degradation can be quantified by the conditional entropy $H(X|Y)$.

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

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_n's). In the noiseless case, we have $C_{main} = H(X)$, and therefore, the secrecy capacity coincides with the conditional entropy of the eavesdropper $C_s = H(X|Y)$, while the analysis of secrecy reduces to the eavesdropper's channel. In general, the more degraded the wiretap channel, the higher the secrecy capacity. We assume for this analysis that the adversary cannot identify the source of each message via signal characteristics (fingerprints, level power, phase shifts, etc.). This implies that tags should be close and implement the same modulation alphabet; i.e., $\mathcal{N}_j = \mathcal{X}$, $1 \leq j \leq n_T$. Possible implementation imperfections, such as delays, signal levels, frequency deviations, etc., should not reveal their origin; i.e., be insignificant or have sufficient randomness. Note that this assumption is implicit in the RFID literature in protocols that address privacy issues: traceability cannot be prevented if tags are physically identified. In this particular case, to prevent an adversary from identifying the target tag, we should guarantee that the tag is close enough to the noisy tags and that it does not present distinguishable imperfections; i.e., insignificant or significant, but changing in every execution. In practice, fortunately, although it is true that no two tags have identical signals, the differences are typically insignificant, making it hard to disambiguate them. As a consequence of the superposition property of the wireless channel, from a theoretical point of view, any modulation can be used (with initial calibration if required), but in practice, some modulations have better features than others. Figure 4 shows a simplified example that uses PPM (pulse position modulation). A bit is encoded by transmitting a pulse in one of two possible time slots. Synchronization between tags is helped by the fact that they share the same reference (reader's) signal. Perfect synchronization is not necessary: tags may have different delays provided there is no pattern that can be exploited to identify a tag.

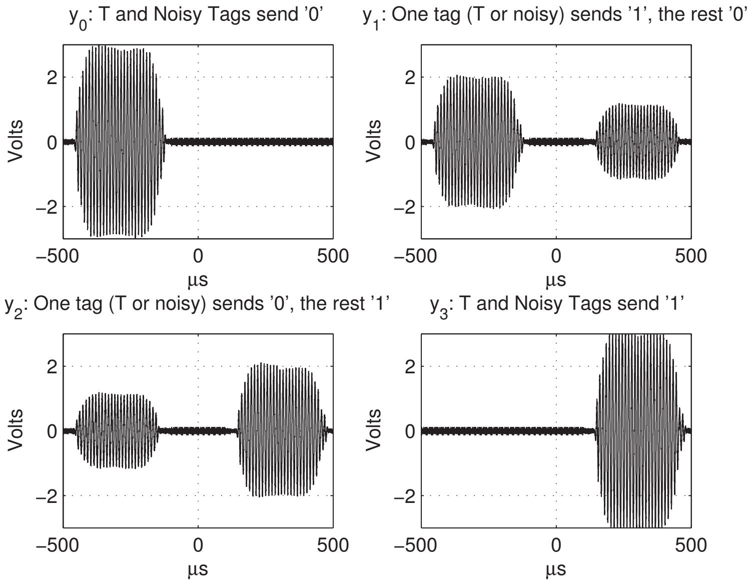


Figure 4. Alphabet $\mathcal{Y} = \{y_0, y_1, y_2, y_3\}$ for tag $\mathcal{T}$ and two noisy tags using pulse position modulation (PPM).

If noise and imperfection implementations are not considered, the security of the system relies exclusively on the stochastic encoder. For r-ary input alphabets $\mathcal{X} = \{x_0, x_1, ..., x_{r-1}\}$, with $p(x_i) = 1/r$, $0 \leq i \leq r-1$, the output alphabet is $\mathcal{Y} = \{y_i\}_{i=0}^{|\mathcal{Y}|-1}$, and the cardinality of $\mathcal{Y}$ (combinations with repetition of r elements taken $n_T + 1$ at a time) and the transition probabilities can be computed as follows:

$$|\mathcal{Y}| = \begin{pmatrix} n_T + r \\ r - 1 \end{pmatrix} = \begin{pmatrix} n_T + r \\ n_T + 1 \end{pmatrix}, \tag{2}$$

$$p(y_{m_0 m_1 ... m_{r-1}} | x_i) = \frac{1}{r^{n_T}} \begin{pmatrix} n_T \\ m_0 \; m_1 \, ... \, m_{r-1} \end{pmatrix} \tag{3}$$

where $y_{m_0 m_1 ... m_{r-1}}$ is the output symbol resulting from the combination of m_0 symbols x_0, m_1 symbols x_1, and so on, until m_{r-1} symbols x_{r-1}, with $m_0 + m_1 + ... + m_{r-1} = n_T$.

Particularizing for binary input alphabets ($r = 2$), $\mathcal{X} = \{x_0, x_1\}$, with $p(x_0) = p(x_1) = 0.5$ ($H(X) = 1$), the output alphabet is $\mathcal{Y} = \{y_i\}_{i=0}^{n_T+1}$, where y_i is the combination of i symbols x_0 and $(n_T + 1 - i)$ symbols x_1. The transition probabilities $p(y_i | x_j)$ are given by:

$$p(y_i | x_0) = p(y_{N+1-i} | x_1) = 2^{-n_T} \binom{n_T}{i}, i = 0, \ldots, n_T + 1\,. \tag{4}$$

Own$_c$'s detector receives y_i and applies the decoding specified by:

$$\begin{array}{l} n_T \text{ even}, \bar{s} = \begin{cases} g(x_0) \text{ if } i < \dfrac{n_T + 1}{2} \\ g(x_1) \text{ otherwise} \end{cases} \\ n_T \text{ odd}, \bar{s} = \begin{cases} g(x_0) \text{ if } i < \dfrac{n_T + 1}{2} \\ g(x_1) \text{ if } i > \dfrac{n_T + 1}{2} \\ \text{otherwise, choose at random } g(x_0) \text{ or } g(x_1) \end{cases} \end{array} . \tag{5}$$

with g the mapping function $g : X \rightarrow S$.

The error probability, defined as $p_e = Pr\left[\bar{s} \neq s\right]$, is computed as:

$$p_e = 2^{-n_T} \left(\sum_{i=0}^{\lfloor \frac{n_T}{2} - 1 \rfloor} \binom{n_T}{i} + \frac{1}{2} \binom{n_T}{\frac{n_T+1}{2}} \right), \tag{6}$$

where the last summand is zero when n_T is even. Figure 5 plots the secrecy capacity C_s of the wiretap channel, the error probability and Fano's bound, against the number of noisy tags. Secrecy increases sharply until $n_T \approx 5$; as $n_T \rightarrow \infty$, the equivocation of the eavesdropper approaches the unconditional source entropy, and we get perfect secrecy: $\lim_{n_T \rightarrow \infty} H(X | Y^{(n_T)}) = H(X) = 1$. For $n_T = 3$, the secrecy capacity $C_s = H(X|Y) = 0.78$ offers a good compromise between features and ease of implementation. The capacity of *Own*$_c$'s channel is just $C_{eav} = 0.22$ bits.

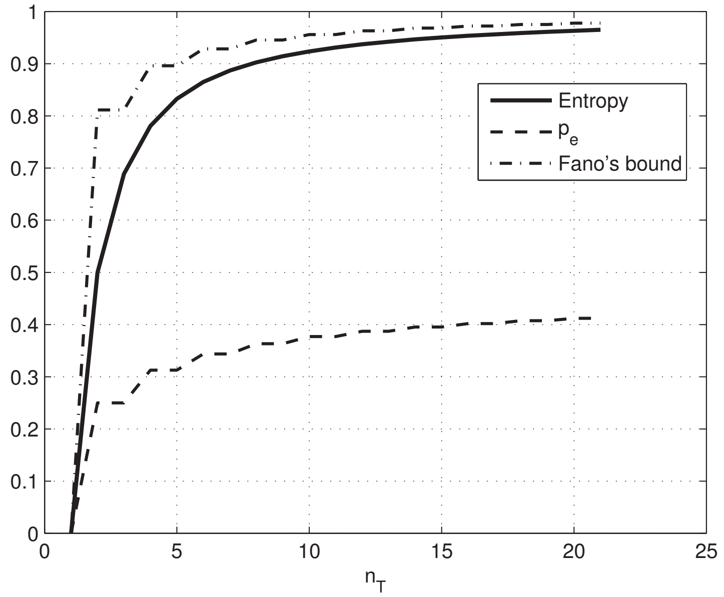


Figure 5. The conditional entropy, error and Fano's bounds of the wiretap channel.

5. An Ownership Transfer Protocol

We next present an example of an OTP that: (i) works according to the communication model defined in Section 3.2 and (ii) uses a channel with positive secrecy capacity, implemented with noisy tags, to guarantee the privacy of the new owner.

The protocol addresses practical design features, such as (secure) singulation of tags and the interrogator-talks-first requirement (communication must be initiated by the reader), and guarantees that the information stored on the tag coincides with that provided to the new owner (tag assurance [17]). Note also that it complies with the restrictions in Section 3.1 regarding entities' capabilities. That is, while RFID readers can implement fully-fledged cryptographic tools, RFID tags are restricted to a pseudorandom number generator (PRNG) and a cryptographic (one-way, collision-resistant) hash function $F : \{0,1\}^* \rightarrow \{0,1\}^n$. The number of inputs is, however, designed to be intentionally low so that it can be more easily adapted to other possible primitives. We assume that identifiers, random numbers and keys all have the same (bit) length n, which is the security parameter of the protocol. We introduce our notation.

ID	identifying information of $\mathcal{T}$.
$Info_{ID}$	hash of the manufacturer information.
$\mathcal{R}1$, $\mathcal{R}2$	readers of Own_c and Own_n respectively.
IDR1, *IDR2*	identifiers for $\mathcal{R}1$ and $\mathcal{R}2$ respectively.
s_1	key that $\mathcal{T}$ shares with $\mathcal{R}1$.
s_2	key that $\mathcal{T}$ shares with $\mathcal{R}2$.
$\overline{s_2}$	key that $\mathcal{T}$ eventually shares with $\mathcal{R}2$.
$N_{\mathcal{T}}$, $N'_{\mathcal{T}}$	random numbers generated by $\mathcal{T}$.
$N_{\mathcal{R}1}$	random number generated by $\mathcal{R}1$.
$N_{\mathcal{R}2}$, $N'_{\mathcal{R}2}$	random numbers generated by $\mathcal{R}2$.
$\mathcal{T}_t^*$	the t noisy tag, with $1 \leq t \leq n_T$.
s_t^*	the key that the $\mathcal{T}_t^*$ shares with $\mathcal{R}2$.

5.1. The Ownership Transfer Protocol, Figure 6

Initialization

1. Initially, each owner knows for each tag *ID* its information and private key s_1. Likewise, each tag stores, along with its identifier *ID* and $Info_{ID}$, the identifier of its owner *IDR1* and the private key. $\mathcal{R}1, \mathcal{R}2$ agree to transfer ownership of tag $\mathcal{T}$ with identifier *ID*. $\mathcal{R}1$ sends (secure channel) $\mathcal{R}2$ manufacturer information about the tag ($Info_{ID}$ when hashed).

$$\mathcal{R}1 \Rightarrow \mathcal{R}2 : ID, \text{manufacturer information}$$

Setup for Ownership Transfer

2. $\mathcal{R}1$ regularly broadcasts *Query* messages to detect the presence of tags.

$$\mathcal{R}1 \rightarrow \text{tags} : Query$$

3. When $\mathcal{T}$ receives a *Query* (presumably because it is within the range of $\mathcal{R}1$), it selects a random nonce $N_\mathcal{T}$ and sends:

$$\mathcal{T} \rightarrow \mathcal{R}1 : F(N_\mathcal{T}, s_1), N_\mathcal{T}$$

4. $\mathcal{R}1$ searches for a pair (ID, s) in its database to get a match. If there is no match, then the process is repeated from Step 2. Otherwise, $\mathcal{T}$ is singulated: $\mathcal{R}1$ selects a random nonce $N_{\mathcal{R}1}$ and a request OTR and sends:

$$\mathcal{R}1 \rightarrow \mathcal{T} : OTR, IDR1, IDR2, F(s_1, N_\mathcal{T}), N_{\mathcal{R}1}$$

5. $\mathcal{T}$ checks $F(s_1, N_\mathcal{T})$ to authenticate $\mathcal{R}1$. $\mathcal{T}$ does not reply if there is no match. Otherwise, it computes $s' = F(N_\mathcal{T}, N_{\mathcal{R}1}, s_1)$, saves $[IDR2, s']$, until the protocol completes or a new command from $\mathcal{R}1$ is received and replies with:

$$\mathcal{T} \rightarrow \mathcal{R}1 : F(N_{\mathcal{R}1}, s_1)$$

6. If this message is not received correctly by $\mathcal{R}1$ after a period of time, the protocol is repeated from Step 2 ($\mathcal{T}$ will replace the stored values $IDR2, s'$). Otherwise, $\mathcal{R}1$ computes $s' = F(N_\mathcal{T}, N_{\mathcal{R}1}, s_1)$ and confirms (secure channel) to $\mathcal{R}2$ that $\mathcal{T}$ is ready to be transferred:

$$\mathcal{R}1 \Rightarrow \mathcal{R}2 : ID \text{ is ready}, s'$$

Ownership Transfer

7. If $\mathcal{R}2$ receives $\mathcal{R}1$'s confirmation, then it is ready to take ownership of $\mathcal{T}$. $\mathcal{R}2$ computes $s_2 = F(s', Info_{ID})$ and broadcasts regularly *Query* messages.

$$\mathcal{R}2 \rightarrow \text{tags} : Query$$

8. When $\mathcal{T}$ receives a *Query*, it selects a random nonce $N'_\mathcal{T}$ and sends:

$$\mathcal{T} \rightarrow \mathcal{R}2 : F(N'_\mathcal{T}, s_2), N'_\mathcal{T}$$

9. If $\mathcal{T}$ is singulated, then $\mathcal{R}2$ selects a fresh random number $N_{\mathcal{R}2}$ and sends:

$$\mathcal{R}2 \rightarrow \mathcal{T} : F(s_2, N'_\mathcal{T}), N_{\mathcal{R}2}$$

10. $\mathcal{T}$ checks this message for s_2, and if not correct, for s_1 (and waits for new commands). It does not reply if this is not correct. If $\mathcal{R}2$ is authenticated, $\mathcal{T}$ updates the stored values $(IDR1, s_1)$ to $(IDR2, s_2)$. These values determine tag ownership. $\mathcal{T}$ acknowledges this by sending:

$$\mathcal{T} \to \mathcal{R}2 : F(N_{\mathcal{R}2}, s_2)$$

11. If the received message is not correct, the protocol is repeated from Step 7. Otherwise, $\mathcal{R}2$ executes the key update protocol in Section 5.2 to prevent $\mathcal{R}1$ from accessing $\mathcal{T}$.

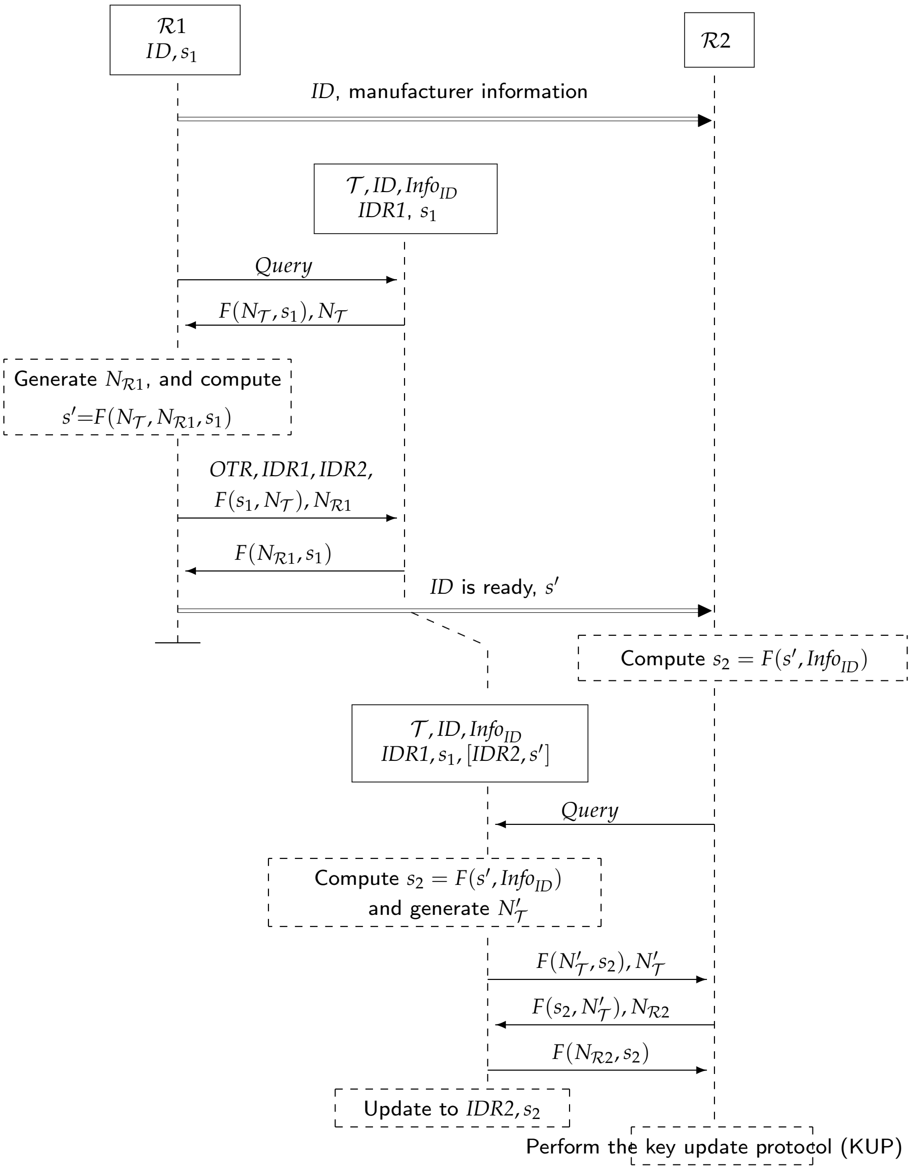

$\mathcal{R}1$
ID, s_1
$\mathcal{R}2$
ID, manufacturer information
$\mathcal{T}, ID, Info_{ID}$
$IDR1,\ s_1$
$Query$
$F(N_{\mathcal{T}}, s_1), N_{\mathcal{T}}$
Generate $N_{\mathcal{R}1}$, and compute
$s'{=}F(N_{\mathcal{T}}, N_{\mathcal{R}1}, s_1)$
$OTR, IDR1, IDR2,$
$F(s_1, N_{\mathcal{T}}), N_{\mathcal{R}1}$
$F(N_{\mathcal{R}1}, s_1)$
ID is ready, s'
Compute $s_2 = F(s', Info_{ID})$
$\mathcal{T}, ID, Info_{ID}$
$IDR1, s_1, [IDR2, s']$
$Query$
Compute $s_2 = F(s', Info_{ID})$
and generate $N'_{\mathcal{T}}$
$F(N'_{\mathcal{T}}, s_2), N'_{\mathcal{T}}$
$F(s_2, N'_{\mathcal{T}}), N_{\mathcal{R}2}$
$F(N_{\mathcal{R}2}, s_2)$
Update to $IDR2, s_2$
Perform the key update protocol (KUP)

Figure 6. The ownership transfer protocol.

5.1.1. Analysis

In the Appendix A, we shall use GNY logic [12], which extends the Burrows–Abadi–Needham (BAN) logic (overcoming some of its problems [13,14]), to show the consistency of the assumptions with respect to the source message, as well as the beliefs of the sender and receiver of messages. Principals can only advance their beliefs and increase their possessions based on the physical content of the messages they receive. We use strand spaces [15] to show correctness by excluding vulnerabilities based on the structure of the protocol. Strand spaces use free encryption algebra to detect faults that exploit relations in this algebra. Below, we discuss the most important security properties informally.

1 Untraceable singulation: Replies to *Query*'s (Step 2, Step 7) have the same format and include a nonce selected by the tag. This prevents tag tracing, since messages look random to anyone who does not know the secret key.
2 The privacy of Own_c is guaranteed because the key s_1 remains unknown to the new owner Own_n. Indeed, if Own_n can compute s_1 given the values: s', $N_{\mathcal{T}}$ and $N_{\mathcal{R}1}$, then Own_n can also find the F-preimage of s', which contradicts the assumption that F is one-way.
3 Forward secrecy: Suppose the adversary succeeds in getting the new key s_2 of a tag. The privacy of the prior communications is guaranteed, as in the previous case, because to get s_1 from s_2, one has to invert F.
4 The privacy of Own_n is achieved by using the key update protocol in Section 5.2.
5 Tag assurance: $Info_{ID}$ is the hash of manufacturer information about the tag. The collision resistance of hash functions prevents the adversary from finding another message (pre-image) $Info'_{ID}$ with the same hash to forge the information given by the manufacturer. The use of $Info_{ID}$ to compute s_2 guarantees that the information provided by Own_c to Own_n matches with the information stored by $\mathcal{T}$. Note, however, that cloned tags and corruptible memories are beyond this security feature (cf. [17]).

5.2. A Key Update Protocol, Figure 7

The parties are: the reader $\mathcal{R}2$, tag $\mathcal{T}$ and n_T noisy tags $\mathcal{T}_t^*$, $1 \leq t \leq n_T$. $\mathcal{R}2$ shares with $\mathcal{T}$ a private key s_2 and with each $\mathcal{T}_t^*$ a private key s_t^*. In this protocol, $\mathcal{T}$ updates privately the key s_2 with a fresh key $\bar{s}_2$.

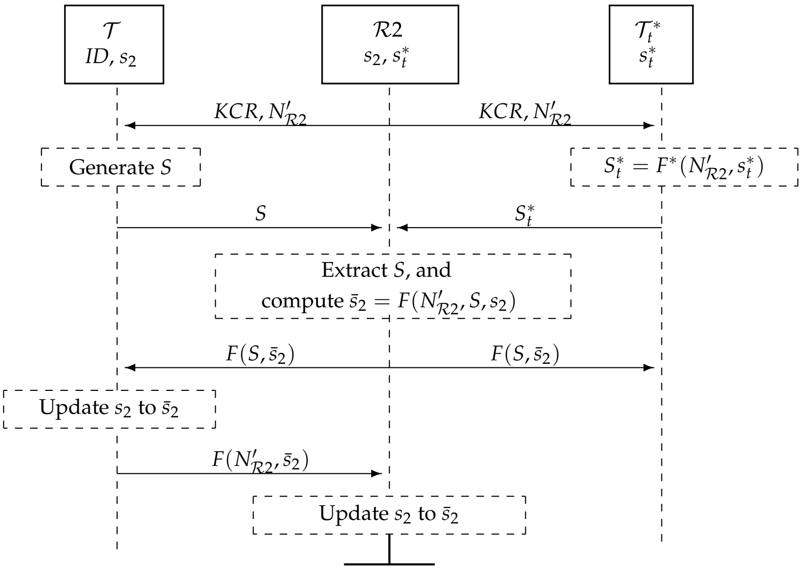


Figure 7. Key update protocol (KUP) with noisy tags $\mathcal{T}_t^*$, $1 \leq t \leq n_T$.

1 $\mathcal{R}2$ broadcasts a key change request (KCR) with a random nonce $N'_{\mathcal{R}2}$.

$$\mathcal{R}2 \rightarrow \mathcal{T}, \{\mathcal{T}_t^*\}_{t=1}^{n_T} : KCR, N'_{\mathcal{R}2}$$

2 Upon receiving this, $\mathcal{T}$ and $\mathcal{T}_t^*$ generate bitstrings S and S_t^* of length n/C_s and broadcast these simultaneously (as specified in Section 4): S is a random number, and $S_t^* = F^*(N'_{\mathcal{R}2}, s_t^*)$, where F^*

is a cryptographic hash function of length n/C_s. Note that F^* could be built from F; for example, for $C_s = 0.5$, $F^*(A, B) = F(A, B)||F(A+1, B)$, where $||$ denotes concatenation.

$$\mathcal{T},\ \{\mathcal{T}_t^*\}_{t=1}^{n_T} \rightarrow \mathcal{R}2:\ S \text{ and } \{S_t^*\}_{t=1}^{n_T}$$

3 $\mathcal{R}2$ receives the added signals of S and $\{S_t^*\}_{t=1}^{n_T}$, extracts S, computes $\bar{s}_2 = F(N'_{\mathcal{R}2}, S, s_2)$ and broadcasts $F(S, \bar{s}_2)$.

$$\mathcal{R}2 \rightarrow \mathcal{T},\ \{\mathcal{T}_t^*\}_{t=1}^{n_T} :\ F(S, \bar{s}_2)$$

4 $\mathcal{T}$ computes $\bar{s}_2 = F(N'_{\mathcal{R}2}, S, s_2)$ and checks that the message from $\mathcal{R}2$ is correct. If so, $\mathcal{T}$ updates its private key s_2 to $\bar{s}_2$.

$$\mathcal{T} \rightarrow \mathcal{R}2 : F(N'_{\mathcal{R}2}, \bar{s}_2)$$

5 $\mathcal{R}2$ checks the received message. If correct, the key update protocol (KUP) is completed, and $\mathcal{R}2$ informs $\mathcal{R}1$. Otherwise, $\mathcal{R}2$ sends a new *Query* and checks if $\mathcal{T}$ has updated its key. If not, the KUP is repeated.

$$\mathcal{R}2 \Rightarrow \mathcal{R}1 : \text{Ownership is transferred.}$$

5.3. Analysis

Attacks by external adversaries on the KUP can target privacy (traceability) or availability (de-synchronization). These are prevented by the wiretap channel with positive secrecy and a cryptographic hash function that authenticates messages. More specifically:

Traceability: $\mathcal{T}$ remains untraceable because the exchanged messages look random to anyone who does not know s_2.

De-synchronization: The adversary cannot compute $F(N'_{\mathcal{R}2}, \bar{s}_s)$ or $F(S, \bar{s}_2)$, that are required by parties to update their keys, without knowing s_2.

The protection extends to threats from past and future owners of $\mathcal{T}$. For example, even if $\mathcal{R}1$ knows s_1 and can get s_2, $\mathcal{R}1$ does not know the keys s_t^* of the noisy tags and, therefore, cannot filter out S_t^* to get S and compute $\bar{s}_2$. In particular, $\mathcal{R}1$ knows $C_{eav} \cdot n/C_s = (1 - C_s) \cdot n/C_s$ bits of S, but the remaining n bits remain unknown. Thus, once the KUP is completed, $\mathcal{R}1$ has no control over the tag $\mathcal{T}$ and cannot trace it.

6. Conclusions

Cryptographic protection is usually handled at the application layer and cannot exploit signal features at the physical layer, which restricts its scope. We have shown in this paper that backward privacy of an OTP can be guaranteed with the use of channels with positive secrecy capacity. The implementation of such channels with noisy tags has been analyzed and the value $n_T = 3$, for which the capacity of the eavesdropper's channel is only $C_{eav} = 0.22$ bits, provides a good compromise between performances and the ease of implementation. We also defined a communication model for RFID ownership transfer that captures spatiotemporal requirements. Protocols defined in this model can be applied to a wider range of practical scenarios. Finally, we have presented the first example of a symmetric-key OTP that does not require a TTP or ISE and formally proved that it is correct and secure in this model.

Acknowledgments: This material is based in part upon work supported by: (a) the National Science Foundation under Grant Numbers 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: All authors contributed equally to this work.

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

Appendix A. Protocol Analysis

Because of space limitations, we only show here the consistency and correctness of the ownership transfer (OT) subprotocol in Section 5.1 (Flow 7–Flow 10). The analysis for the first part is similar.

Appendix A.1. GNY Logic

In Figure A1, we present the notation we shall use: $P, Q, \ldots$ are protocol parties (principals); $X, Y, \ldots$ are formulae; and s is a key. The conjunction (X, Y) is also a formula.

Initial assumptions: At the beginning of each run of the OT subprotocol, we assume that parties $\mathcal{T}$ and $\mathcal{R}2$: (i) believe (trust) each other: $\mathcal{T} \mid\equiv \mathcal{R}2 \mid\equiv \mathcal{T}$; (ii) believe that the secret s_2 to be shared between them is suitable: $\mathcal{T} \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$ and $\mathcal{R}2 \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$; and (iii) believe in the jurisdiction of $\mathcal{R}1$ over the secret s_2: $\mathcal{T} \mid\equiv \mathcal{R}1 \mid\Rightarrow (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$ and $\mathcal{R}2 \mid\equiv \mathcal{R}1 \mid\Rightarrow (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$. In addition, each party possesses the secret key s_2 and a fresh nonce: $\mathcal{T} \ni s_2$, $\mathcal{T} \ni N'_{\mathcal{T}}$, $\mathcal{T} \mid\equiv \sharp N'_{\mathcal{T}}$, $\mathcal{R}2 \ni s_2$ and $\mathcal{R}2 \ni N_{\mathcal{R}2}$, $\mathcal{R}2 \mid\equiv \sharp N_{\mathcal{R}2}$. Finally, $\mathcal{T}$ believes that $(s_2, N'_{\mathcal{T}})$ is recognizable, and $\mathcal{R}2$ believes that $(N'_{\mathcal{T}}, s_2)$ and $(N_{\mathcal{R}2}, s_2)$ are recognizable: $\mathcal{T} \mid\equiv \phi(s_2, N'_{\mathcal{T}})$, $\mathcal{R}2 \mid\equiv \phi(N'_{\mathcal{T}}, s_2)$ and $\mathcal{R}2 \mid\equiv \phi(N_{\mathcal{R}2}, s_2)$.

$P \ni X$:	P possesses X	$P \triangleleft X$:	P is told (or receives) X
$P \mid\equiv X$:	P believes X	$Q \mid\Rightarrow C$:	Q has jurisdiction over C
$\sharp X$:	X is fresh	$\sharp(X, Y)$:	either X or Y is fresh
$\phi(X)$:	X is recognizable	$P \mid\sim X$:	P once conveyed X
$*X$:	X is "not originated here" formula		
$P \overset{s}{\leftrightarrow} Q$:	s is a suitable secret for P, Q		

Figure A1. GNY reasoning notation.

The goal of the OT subprotocol is for $\mathcal{R}2$ and $\mathcal{T}$ to exchange the key s_2. The GNY logic parses the description of protocols for formal reasoning. A formalized description of the OT subprotocol is presented in Figure A2.

7. $\mathcal{T} \triangleleft *Query$
8. $\mathcal{R}2 \triangleleft *F(*N'_{\mathcal{T}}, *s_2), \mathcal{R}2 \triangleleft *N'_{\mathcal{T}} \leadsto \mathcal{T} \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2), \mathcal{T} \ni N'_{\mathcal{T}}$
9. $\mathcal{T} \triangleleft *F(s_2, N'_{\mathcal{T}}), \mathcal{T} \triangleleft *N_{\mathcal{R}2} \leadsto \mathcal{R}2 \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2), \mathcal{R}2 \ni N_{\mathcal{R}2}$
10. $\mathcal{R}2 \triangleleft *F(N_{\mathcal{R}}, s_2) \leadsto \mathcal{T} \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$

Figure A2. The parsed ownership transfer (OT) subprotocol.

In this, Flows 8, 9 and 10 include message extensions $(\cdots \leadsto X)$ that are assumed assumptions. To prove consistency, we must show that on completion of the subprotocol, the following formulae can be deduced: $\mathcal{T} \ni s_2$, $\mathcal{T} \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$, $\mathcal{T} \mid\equiv \mathcal{R}2 \ni s_2$, $\mathcal{R}2 \ni s_2$, $\mathcal{R}2 \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$, $\mathcal{R}2 \mid\equiv \mathcal{T} \ni s_2$.

Four of these are initial assumptions. Therefore, we only need to show the formulae:

$$\mathcal{T} \mid\equiv \mathcal{R}2 \ni s_2, \text{ and} \tag{A1}$$

$$\mathcal{R}2 \mid\equiv \mathcal{T} \ni s_2. \tag{A2}$$

For this purpose, we use the deduction rules of GNY logic. A deduction rule consists of a set of premises $P_1, \ldots, P_n$ and a conclusion C, written: $\frac{P_1, \ldots, P_n}{C}$. In Figure A3, we list the rules that we shall use to deduce formulae: $f(X)$ and $h(X)$ are computationally feasible functions of X, with $h(X)$ a one-way function.

$$\text{T1}\quad \frac{P \triangleleft *X}{P \triangleleft X}\,; \qquad \text{P1}\quad \frac{P \triangleleft X}{P \ni X}\,; \qquad \text{P3}\quad \frac{P \ni (X,Y)}{P \ni h(X)}$$

$$\text{J1}\quad \frac{P \mid\equiv \mid\Rightarrow C, P \mid\equiv Q \mid\equiv C}{P \mid\equiv \sharp(X,Y), P \mid\equiv f(X)}\,; \qquad \text{R5}\quad \frac{P \mid\equiv \phi(X),\ P \ni X}{P \mid\equiv \phi(h(X))}$$

$$\text{F1}\quad \frac{P \mid\equiv \sharp X}{P \mid\equiv \sharp(X,Y), P \mid\equiv f(X)}\,; \qquad \text{I6}\quad \frac{P \mid\equiv Q \mid\sim X,\ P \mid\equiv \sharp X}{P \mid\equiv Q \ni X}$$

$$\text{I3}\quad \frac{P \triangleleft *h(X,s),\ P \ni (X,s),\ P \mid\equiv P \overset{s}{\leftrightarrow} Q,\ P \mid\equiv \sharp(X,s)}{P \mid\equiv Q \mid\sim (X,s),\ P \mid\equiv Q \mid\sim h(X,s)}$$

Figure A3. GNY logic postulates.

To show that Formulas (A1) and (A2) can be deduced from protocol assumptions and transmitted messages, we analyze below the parsed OT subprotocol in Figure A2.

7. No belief or possession can be derived from this message.
8. Apply the being-told rule T1 and the possession rule P1 to $\mathcal{R}2 \triangleleft *N'_{\mathcal{T}}$ to get $\mathcal{R}2 \ni N'_{\mathcal{T}}$. Apply the recognizability rule R5 to the initial assumptions $\mathcal{R}2 \mid\equiv \phi(N'_{\mathcal{T}}, s_2)$ to get that $\mathcal{R}2$ recognizes $\mathcal{T}$. No postulate enables us to further derive new beliefs or possessions from this message. In particular, we cannot derive the freshness of the message.
9. Apply rules T1 and P1 to $\mathcal{T} \triangleleft *N_{\mathcal{R}2}$ to get $\mathcal{T} \ni N_{\mathcal{R}2}$. Apply the freshness rule F1 to the initial assumptions $\mathcal{T} \mid\equiv \sharp N'_{\mathcal{T}}, \phi(s_2, N'_{\mathcal{T}})$ to get $\mathcal{T} \mid\equiv \sharp(s_2, N'_{\mathcal{T}})$. Apply the interpretation rule I3 to: the previous result, $\mathcal{T} \triangleleft * F(s_2, N'_{\mathcal{T}})$ and the initial assumptions $\mathcal{T} \ni (s_2, N'_{\mathcal{T}})$ and $\mathcal{T} \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$, to get $\mathcal{T} \mid\equiv \mathcal{R}2 \mid\sim s_2$. Now, apply rule I6 to get Formula (A1): $\mathcal{T} \mid\equiv \mathcal{R}2 \ni s_2$.
10. Apply the freshness rule F1 to the initial assumptions $\mathcal{R}2 \mid\equiv \sharp N'_{\mathcal{R}2}, \phi(N'_{\mathcal{R}2}, s_2)$ to get $\mathcal{R}2 \mid\equiv \sharp(N'_{\mathcal{R}2}, s_2)$. Apply rule I3 to: the previous result, $\mathcal{R}2 \triangleleft * F(N'_{\mathcal{R}2}, s_2)$ and the initial assumptions $\mathcal{R}2 \ni (N_{\mathcal{R}}, s_2)$ and $\mathcal{R}2 \mid\equiv (\mathcal{T} \overset{s_2}{\longleftrightarrow} \mathcal{R}2)$, to get $\mathcal{R}2 \mid\equiv \mathcal{T} \mid\sim s_2$. Now, apply rule I6 to get Formula (A2): $\mathcal{R}2 \mid\equiv \mathcal{T} \ni s_2$.

It follows that the OT subprotocol is consistent. In particular,

(a) Possession consistency: transmitted messages only include formulae that the sender possesses;
(b) Belief consistency: message extensions include only beliefs held by the sender at the time he/she sends the message.

Strand spaces: We next show the correctness of the OT subprotocol using strand spaces [12,15]. To simplify the analysis, we remove Flow 7, which does not provide any cryptographic information.

A strand space Σ is a collection of strands and a graph generated by a causality relation. A strand s is a sequence of events that represent either a protocol execution by a legitimate party (principal) or a sequence of actions by a penetrator. We refer to the messages that can be exchanged between the principals as terms of the strand. In a protocol, principals can either send or receive terms, and this is represented with a positive or a negative sign, respectively. We write $a \sqsubset b$ if a is a subterm of b. The trace $\text{tr}(s)$ of a strand is the sequence of its signed terms. A node of Σ is a pair $n = \langle s, i \rangle$, with $s \in \Sigma$, $1 \leq i \leq \text{length}(\text{tr}(s))$. The set of nodes is denoted by $\mathcal{N}$. We say that node $n = \langle s, i \rangle$ belongs to strand s. $\text{term}(n)$ is the i-th signed term $\text{tr}(s)_i$ of s.

We write $n_1 \prec n_2$ to indicate that n_1 precedes n_2 in a strand (not necessarily immediately). An unsigned term t occurs in n iff $t \sqsubset \text{term}(n)$;$n$ is an entry point for a set of terms $I \subset T$ iff (if and only if) term $(n) = +t$ for some $t \in I$, and whenever $n' \prec n$, then term $(n') \notin I$. An unsigned term t originates on n iff n is an entry point for $I = \{t' : t \sqsubset t'\}$. t is uniquely originating iff t originates at a unique $n \in \mathcal{N}$. A bundle is a portion of a strand space that consists of strands of a protocol session that are hooked together, where one strand sends a message and the other receives the same message. For a protocol to be correct, each such bundle must contain one strand for each one of the legitimate principals participating in a session, with all parties agreeing on nonces and session keys. The penetrator (adversary) has a set of keys $\text{K}_{\mathcal{P}}$ (shared with accomplices or "lost") and a set of penetrator traces $\mathcal{P}$ that model her/his capabilities. Penetration traces typically require hooking several atomic traces. In Figure A4, we list the atomic penetrator traces we shall consider [12]. A protocol attack is captured by combining penetrator traces with protocol strands.

M. Text message: $\langle +t \rangle$	C. Concatenation: $\langle -g, -h, +gh \rangle$
F. Fushing: $\langle -g \rangle$	S. Separation of components: $\langle -gh, +g, +h \rangle$
T. Tee: $\langle -g, +g, +g \rangle$	E. Encryption: $\langle -K, -h, +\{h\}_K \rangle$
K. Key: $\langle +K \rangle$	D. Decryption: $\langle -K^{-1}, -\{h\}_K, +h \rangle$

Figure A4. Atomic penetrator traces.

Definition A1. *$(\Sigma, \mathcal{P})$ is an infiltrated strand space if Σ is a strand space and $\mathcal{P} \subset \Sigma$ is such that $tr(p)$ is a penetrator trace for all $p \in \mathcal{P}$.*

Definition A2. *An infiltrated strand space $(\Sigma, \mathcal{P})$ is an OTP space if Σ has three kinds of strands:*

Step 1. *Penetrator strands $s \in \mathcal{P}$*

Step 2. *Initiator strands $s \in Init[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N'_{\mathcal{R}2}]$ defined by:*

$$\langle +(F(N'_{\mathcal{T}}, s_2), N'_{\mathcal{T}}), -(F(s_2, N'_{\mathcal{T}}), N'_{\mathcal{R}2}), +F(N'_{\mathcal{R}2}, s_2) \rangle,$$

with $s_2 \in \mathsf{K}$, $N'_{\mathcal{T}}, N'_{\mathcal{R}2} \notin \mathsf{K}$. $\mathcal{T}$ is the principal associated with this strand.

Step 3. *Responder strands $s \in Resp[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N'_{\mathcal{R}2}]$, defined by:*

$$\langle -(F(N'_{\mathcal{T}}, s_2), N'_{\mathcal{T}}), +(F(s_2, N'_{\mathcal{T}}), N'_{\mathcal{R}2}), -F(N'_{\mathcal{R}2}, s_2) \rangle,$$

with $s_2 \in \mathsf{K}$, $N'_{\mathcal{T}}, N'_{\mathcal{R}2} \notin \mathsf{K}$. $\mathcal{R}2$ is the principal associated with this strand.

(A) AGREEMENT: the responder's guarantee:

Proposition A1. *Suppose that: $(\Sigma, \mathcal{P})$ is an OTP space, $\mathcal{C}$ a bundle of Σ, $s \in Resp[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N'_{\mathcal{R}2}]$, $s_2 \notin \mathsf{K}_{\mathcal{P}}$ and $N'_{\mathcal{T}} \neq N_{\mathcal{R}2}$ with $N_{\mathcal{R}2}$ uniquely originating in Σ. Then, $\mathcal{C}$ contains an initiator strand $t \in Init[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N_{\mathcal{R}2}]$.*

Proof. We prove this using four lemmas. Let n_0 be the node $\langle s, 2 \rangle$ (the second node of the reader) that outputs the term $v_0 = (F(s_2, N'_{\mathcal{T}}), N_{\mathcal{R}2})$ and n_3 the node $\langle s, 3 \rangle$ that receives the term $v_3 = F(N_{\mathcal{R}2}, s_2)$. Two additional nodes n_1, n_2 such that $n_0 \prec n_1 \prec n_2 \prec n_3$ will be identified.

Lemma A1. *$N_{\mathcal{R}2}$ originates at node n_0.*

Proof. We know that $N_{\mathcal{R}2} \sqsubset v_0$, and the sign of n_0 is positive. We just need to show that $N_{\mathcal{R}2} \not\sqsubset \langle s, 1 \rangle$. Since term $(\langle s, 1 \rangle) = (F(N'_{\mathcal{T}}, s_2), N'_{\mathcal{T}})$, we only need to check that $N'_{\mathcal{T}} \neq N_{\mathcal{R}2}$, which is a hypothesis, and that $s_2 \neq N_{\mathcal{R}2}$, which follows from the stipulation $N_{\mathcal{R}2} \notin K$. □

The next lemma establishes that the crucial step is taken by a regular strand and not a penetrator strand.

Lemma A2. *The set $S = \{n \in \mathcal{C}\colon v_3 \sqsubset term(n) \wedge v_0 \not\sqsubset term(n)\}$ has a $\preceq$−minimal node n_2, which is regular and has a positive sign.*

Proof. S is non-empty because $n_3 \in \mathcal{C}$; and n_3 contains v_3, but not v_0. Since S is a partially-ordered set (because $\mathcal{C}$ is), it has at least one $\preceq$−*minimal* node n_2, and its sign must be positive. Therefore, we just need to check that n_2 does not lie on a penetrator strand p. For this purpose, we shall examine all of the atomic penetrator traces tr(p) listed in Figure A4.

M. tr$(p) = \langle +t \rangle$: Then, $N_{\mathcal{R}2} \sqsubset t$ and $N_{\mathcal{R}2}$ originates on t, which is not possible because $N_{\mathcal{R}2}$ originates on the regular node n_0 (Lemma A1).

F. $\mathrm{tr}(p) = \langle -g \rangle$: This has no positive nodes.

T,C $\mathrm{tr}(p) = \langle -g, +g, +g \rangle$ or $\langle -g, -h, +gh \rangle$: then, the positive nodes are not minimal occurrences.

K. $\mathrm{tr}(p) = \langle +K_0 \rangle$ with $K_0 \in K_P$: Since $v_3 \not\sqsubset K_0$, this case does not apply.

E. $\mathrm{tr}(p) = \langle -K_0, -h, +\{h\}_{K_0} \rangle$: Suppose $v_3 \sqsubset \{h\}_{K_0}$. Then, $h = N_{\mathcal{R}2}$, $K_0 = s_2$. Thus, there is a node m (the first of this strand) with term $(m) = s_2$. However, $s_2 \notin \mathsf{K}_{\mathcal{P}}$, so that this node is regular, but no regular node originates s_2. This contradicts the initial assumption.

D. $\mathrm{tr}(p) = \langle -K_0^{-1}, -\{h\}_{K_0}, +h \rangle$: If the positive node is minimal in S, then $v_0 \not\sqsubset h$ and $v_0 \sqsubset \{h\}_{K_0}$. However, because $v_0 \neq \{h\}_{K_0}$, if $v_0 \sqsubset \{h\}_{K_0}$, then $v_0 \sqsubset h$, which is a contradiction.

S. $\mathrm{tr}(p) = \langle -gh, +g, +h \rangle$: Assume term $(n_2) = h$ (there is a symmetric case with term $(n_2) = g$). By the minimality of n_2, $v_0 \sqsubset gh$. Hence, $g = F(N'_{\mathcal{T}}, s_2)$ and $h = N_{\mathcal{R}2}$. However, then $v_3 \not\sqsubset h$ and $n_2 \notin S$, contradicting the initial assumption.

Therefore, n_2 does not lie on a penetrator strand. □

Lemma A3. *Node n_2 follows n_1 on the same regular strand t, and term $(n_1) = (F(s_2, N'_{\mathcal{T}}), N_{\mathcal{R}2})$.*

Proof. From Lemma A1, we know that $N_{\mathcal{R}2}$ originates at n_0, and by assumption, it is unique in Σ. Furthermore, $n_2 \neq n_0$ since $v_0 \sqsubset$ term (n_0) and $v_0 \sqsubset / -1.5$ mm term (n_2). Therefore, $N_{\mathcal{R}2}$ does not originate at n_2, and there is a node n_1 preceding n_2 on the same strand, such that $N_{\mathcal{R}2} \sqsubset$ term (n_1). By the minimal property of n_2, $v_0 \sqsubset$ term (n_1). However, as no regular node contains a combination as a proper subterm, $\mathrm{term}(n_1) = (F(s_2, N'_{\mathcal{T}}), N_{\mathcal{R}2})$. □

Lemma A4. *The regular strand t containing n_1 and n_2 is an initiator strand contained in $\mathcal{C}$.*

Proof. n_1 precedes n_2 in the same strand. Node n_2 is a positive regular node and comes after a node with the form $(F(s_2, N'_{\mathcal{T}}), N_{\mathcal{R}2})$. Hence, t is an initiator strand, since a responder strand would only contain a negative node after one of that form. Thus, n_1 and n_2 are the second and the third nodes of t, respectively. □

Lemmas A3 and A4 complete the proof of Proposition A1. □

Proposition A2. *If $(\Sigma, \mathcal{P})$ is an OTP space and $N'_{\mathcal{T}}$ is uniquely originating in Σ, then there is at most one strand $t \in Init[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N'_{\mathcal{R}2}]$ for any $\mathcal{T}$, $\mathcal{R}2$ and $N_{\mathcal{R}2}$.*

Proof. Let $t \in \mathrm{Init}[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N_{\mathcal{R}2}]$ for $\mathcal{T}$, $\mathcal{R}2$ and $N_{\mathcal{R}2}$. Then, $\langle t, 1 \rangle$ is positive, $N'_{\mathcal{T}} \sqsubset \mathrm{term}\langle t, 1 \rangle$, and $N'_{\mathcal{T}}$ cannot possibly occur earlier on t. Therefore, $N'_{\mathcal{T}}$ originates at node $\langle t, 1 \rangle$. Since $N'_{\mathcal{T}}$ originates uniquely in Σ, there can be at most one such t. □

(B) AGREEMENT: the initiator's guarantee:

Proposition A3. *Suppose that: $(\Sigma, \mathcal{P})$ is an OTP space, $\mathcal{C}$ is a bundle of Σ, $s \in Init[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N_{\mathcal{R}2}]$, $s_2 \notin \mathsf{K}_{\mathcal{P}}$ and $N'_{\mathcal{T}}$ is uniquely originating in Σ. Then, there exists a responder strand $t \in Resp[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N_{\mathcal{R}2}]$.*

Proof. Consider the set $\{m \in \mathcal{C} : F(s_2, N'_{\mathcal{T}}) \sqsubset \mathrm{term}(m)\}$. This is not empty, because it contains $\langle s, 2 \rangle$, and so, it contains a minimal node m_0. If m_0 lies on a regular strand t, then we can show that $t \in \mathrm{Resp}[\mathcal{T}, \mathcal{R}2, N'_{\mathcal{T}}, N_{\mathcal{R}2}]$. If instead, m_0 lies on a penetrator strand p, then p should be an E-strand with trace: $\langle -s_2, -N'_{\mathcal{T}}, +F(s_2, N'_{\mathcal{T}}) \rangle$, but this contradicts the assumption $s_2 \notin \mathsf{K}_{\mathcal{P}}$. □

References

1. Van Deursen, T.; Mauw, S.; Radomirovic, S.; Vullers, P. *Secure Ownership and Ownership Transfer in RFID Systems*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5789, pp. 637–654.

2. Avoine, G. *Adversarial Model for Radio Frequency Identification*; Technical Report; Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC): Lausanne, Switzerland, 2005.
3. Juels, A.; Weis, S.A. Defining strong privacy for RFID. *ACM Trans. Inf. Syst. Secur.* **2009**, *13*, 7:1–7:23.
4. Vaudenay, S. On privacy models for RFID. In *ASIACRYPT*; Kurosawa, K., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4833, pp. 68–87.
5. Ng, C.Y.; Susilo, W.; Mu, Y.; Safavi-Naini, R. RFID privacy models revisited. In *ESORICS*; Jajodia, S., Lpez, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5283, pp. 251–266.
6. Vullers, P. Secure Ownership and Ownership Transfer in RFID Systems. Master's Thesis, Eindhoven University, Eindhoven, The Netherlands, 2009.
7. Kapoor, G.; Piramuthu, S. Single RFID Tag Ownership Transfer Protocols. *IEEE Trans. Syst. Man Cybern. Part C* **2012**, *42*, 164–173.
8. 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.
9. 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.
10. Song, B. RFID Tag Ownership Transfer. In Proceedings of the Workshop on RFID Security—RFIDSec'08, Budapest, Hungary, 9–11 July 2008.
11. 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, ISDPE '07, Chengdu, China, 1–3 November 2007*; IEEE Computer Society Press: Washington, DC, USA, 2007; pp. 508–510.
12. Gong, L.; Needham, R.; Yahalom, R. Reasoning about belief in cryptographic protocols. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA, 7–9 May 1990*; IEEE Computer Society Press: Washington, DC, USA, 1990; pp. 234–248.
13. Boyd, C.; Mao, W. On a limitation of BAN logic. In *Advances in Cryptology EUROCRYPT 93*; Helleseth, T., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1994; Volume 765, pp. 240–247.
14. Nessett, D. A critique of the Burrows, Abadi, and Needham logic. *Oper. Syst. Rev.* **1990**, *24*, 35–38.
15. Thayer, F.; Herzog, J.; Guttman, J. Strand Spaces: Proving Security Protocols Correct. *J. Comput. Secur.* **1999**, *7*, 191–230.
16. 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.
17. Ng, C.Y.; Susilo, W.; Mu, Y.; Safavi-Naini, R. Practical RFID Ownership Transfer Scheme. *J. Comput. Secur.* **2011**, *19*, 319–341.
18. 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: Berlin/Heidelberg, Germany, 2005; Volume 3823, pp. 1303–1312.
19. Avoine, G.; Dysli, E.; Oechslin, P. Reducing time complexity in RFID systems. In Proceedings of the 12th International Conference on Selected Areas in Cryptography (SAC 2005), Kingston, ON, Canada, 11–12 August 2005.
20. Soppera, A.; Burbridge, T. Secure by default: The RFID acceptor tag (RAT). In Proceedings of the Workshop on RFID Security—RFIDSec'06, Graz, Austria, 12–14 July 2006.
21. 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; pp. 1090–1095.
22. 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, ICUIMC '09, Suwon, Korea, 15–16 January 2009.
23. Jappinen, P.; Hamalainen, H. Enhanced RFID security method with ownership transfer. In Proceedings of the 2008 International Conference on Computational Intelligence and Security, CIS '08, Suzhou, China, 13–17 December 2008; pp. 382–385.

24. Yoon, E.-J.; Yoo, K.-Y. Two security problems of RFID security method with ownership transfer. In Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing, NPC 2008, Shanghai, China, 18–21 October 2008; pp. 68–73.
25. Kapoor, G.; Piramuthu, S. Vulnerabilities in some recently proposed RFID ownership transfer protocols. *IEEE Commun. Lett.* **2010**, *14*, 260–262.
26. 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.
27. 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, RFIDSec'11, Amherst, MA, USA, 26–28 June 2011.
28. Song, B.; Mitchell, C.J. Scalable {RFID} security protocols supporting tag ownership transfer. *Comput. Commun.* **2011**, *34*, 556–566.
29. Kapoor, G.; Zhou, W.; Piramuthu, S. Multi-tag and Multi-owner RFID Ownership Transfer in Supply Chains. *Decis. Support Syst.* **2011**, *52*, 258–270.
30. EPC Global. EPC Tag Data Standards, vs. 1.3. Available online: http://www.epcglobalinc.org/standards/EPCglobal_Tag_Data_Standard_TDS_Version_1.3.pdf (accessed on 27 December 2016).
31. 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.
32. Koralalage, K.H.S.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, SAC '07, Seoul, Korea, 11–15 March 2007.
33. 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.
34. 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.
35. Paret, D. *RFID and Contactless Smart Card Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
36. International Organization for Standardization. *ISO/IEC 29192-1: Information Technology—Security Techniques—Lightweight Cryptography—Part 1: General*; ISO: Geneva, Switzerland, 2012.
37. Wyner, A. The Wire-Tap Channel. *Bell Syst. Tech. J.* **1975**, *54*, 1355–1387.
38. Juels, A.; Rivest, R.; Szydlo, M. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In *Proceedings of the Conference on Computer and Communications Security, Washington, DC, USA, 27–30 October 2003*; Atluri, V., Ed.; ACM Press: New York, NY, USA, 2003; pp. 103–111.
39. 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.