A survey on RFID security and provably secure grouping-proof protocols

Dang Nguyen Duc*

Department of Computer Science,
KAIST 119 Munjiro, Yuseong-gu,
Daejeon 305-732, Republic of Korea
E-mail: nguyenduc@kaist.ac.kr
*Corresponding author

Divyan M. Konidala

Department of Information and Communications Engineering,
KAIST 119 Munjiro, Yuseong-gu,
Daejeon 305-732, Republic of Korea
E-mail: divyan@kaist.ac.kr

Hyunrok Lee and Kwangjo Kim

Department of Computer Science,
KAIST 119 Munjiro, Yuseong-gu,
Daejeon 305-732, Republic of Korea
E-mail: tanker@kaist.ac.kr
E-mail: kkj@kaist.ac.kr

Abstract: RFID security is a relatively new research area. Within less than a decade, a large number of research papers dealing with security issues of RFID technology have appeared. In the first part of this paper, we attempt to summarise current research in the field of RFID security and discuss some of their open issues. In the second part of this paper, we address some of the open problems we suggested in the first part. In particular, we deal with scalability problem of existing grouping-proof protocols for RFID tags. In addition, we also present the first security definition for a secure grouping-proof protocol for RFID tags. The definition is then used to analyse the security of our proposed grouping-proof protocol which employs a (n, n)-secret sharing scheme to solve the scalability problem of previous protocols.

Keywords: RFID security; tag cloning; tag privacy; authentication; grouping-proof; scalability; security model; security definition, provable security.

Reference to this paper should be made as follows: Duc, D.N., Konidala, D.M., Lee, H. and Kim, K. (2010) 'A survey on RFID security and provably secure grouping-proof protocols', *Int. J. Internet Technology and Secured Transactions*, Vol. 2, Nos. 3/4, pp.222–249.

Biographical notes: Dang Nguyen Duc received his PhD in Cryptography and Information Security in February 2010 from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea. He is currently a Post-Doctoral Researcher at the Department of Computer Science, KAIST. His research interests include design and analysis of cryptographic protocols and efficient implementation of cryptographic primitives.

Divyan M. Konidala is a PhD student at the Department of Information and Communications Engineering, KAIST, Daejeon, Republic of Korea. His research interests include RFID security, security in EPC framework and mobile payment system.

Hyunrok Lee is a PhD student at the Department of Computer Science, KAIST, Daejeon, Republic of Korea. His research interests include side-channel attacks and countermeasures.

Kwangjo Kim received his PhD from Yokohama National University, Japan in 1991. He is currently a Professor at the Department of Computer Science, KAIST, Daejeon, Republic of Korea. He has served as the IACR Director and a Program Committee Member of many security-related conferences. His research interests include the theory and practices of cryptology and information security.

1 Introduction

1.1 Overview of RFID

Radio frequency identification (RFID) is a mean to efficiently and quickly, auto-identify objects, and assets. With RFID technology, passive-RFID tags are attached to consumer items and these tags contain tiny, but durable computer chips with very small antennas. Passive-tags are powered-up from the interrogation radio-frequency (RF) signal of a RFID reader. The tiny computer chips contain an electronic product code (EPC) number that uniquely identifies the item to which it is attached to, and the antennas automatically transmit this EPC number without requiring line-of-sight scanning, to readers. All the information associated with that EPC number is stored on a network of databases, called the EPC-information services (EPC-IS) or the back-end server. Some typical applications of RFID technology are described below:

- *Automatic supply chain management:* An RFID tag gives an item an identity just like a barcode. However, RFID tags can be scanned in bulk without human intervention. As a result, one can implement an automatic supply chain management system by attaching RFID tags on all tracked items and placing RFID readers at different check points. The RFID readers can be connected to the back-end server to provide real-time tracking information.
- *Smart home appliances:* RFID readers can be integrated into home appliances to provide added benefits to end users. For example, a refrigerator equipped with an RFID reader can scan RFID-tagged items stored inside. Then, the refrigerator accesses to a publicly available or a home server to get various information on each item like the date of expiration, the origin of an item, etc.

- *Ubiquitous computing experience for end users:* The sheer ubiquitous availability of RFID-tagged items also brings many powerful applications to ubiquitous computing. An end user equipped with an RFID reader (possibly intergraded into his smartphone or PDA) can scan tagged items and collect information about them on-the-go. For instance, when a customer goes shopping in a supermarket, he/she can queries tagged items to get detailed information on the goods, compare the prices, etc. On the other hand, an RFID tag can also be embedded into the end user's smartphone. The RFID tag may store information about its owner including personal identity (possibly a pseudo one for privacy reason), banking account, etc. This information might help the user to do some micro payments like bus and subway tickets, movies, etc. The surrounding environment can also recognise the user via the embedded tag to provide entrance to buildings, recommendation of available computing services and the likes.
- *Anti-counterfeiting:* An RFID tag is a computing device and thus capable of storing more information and performing sophisticated scanning protocol than a barcode. An RFID tag can be embedded into bank notes, money papers and passports to prevent counterfeiting. When a tag is scanned, tag-to-reader authentication and vice versa can be performed so that counterfeited tags can be detected. In case of passport, an RFID tag can store not just identity but also biometric information of the owner to provide even stronger anti-counterfeiting. RFID tags also have been used in military in order to identify friend or foe in the battlefield.

1.2 Security threats to RFID

Despite being a fairly new technology and not yet widely deployed, RFID technology has already been the target of some real-world attacks (Hutter et al., 2007; Oren and Shamir, 2007; Carluccio et al., 2005; Koscher et al., 2009; Plos, 2008). Ironically, RFID suffers from a number of security threats because of the core functionality of an RFID tag that is to communicate the identification information stored in the tag. The security threats are as follows:

- *Tag cloning:* When queried by an RFID reader, an RFID tag emits a unique number called EPC. This EPC number serves as product identity which points to an entry in a database of the back-end server. Unfortunately, this is an inherent security risk since we depend on the EPC number to recognise a product as genuine or fake. An attacker equipped with a compatible reader can scan many RFID tags to harvest a large number of EPC numbers. He then can produce RFID tags which emit exactly the same EPCs he has collected. We call this kind of tags *cloned tags*. The cloned tags can be attached on counterfeited items which should be recognised as genuine ones.
- *Privacy invasion:* The core functionality of an RFID tag also raises privacy concern. As EPC number is unique, an attacker with a compatible reader can recognise, track and trace RFID tags which lead to privacy invasion of people carrying tagged items.
- *Denial/disruption of service:* RFID technology is really useful when it is deployed at a very large scale. The hope is to attach to each and every item of human interest an RFID tag. In this case, the infrastructure for an RFID system has to maintain and

process a large amount of data. If a large number of fake tags and even malicious readers are deployed, computational resources can be abused and disruption of service may happen. Another form of denial-of-service attack is to physically interfere (e.g., by using electromagnetic jamming technique) the communication channel between tags and readers. However, while protocol-level denial-of-service attacks are possible at a large scale, physical attacks might be possible at a much smaller scale, e.g., effective against a tag population of less than a hundred tags. Therefore, in this thesis, we only consider attacks and defences at protocol level.

- *Location-based attacks (mafia fraud/terrorist attacks):* Wireless communication is inherently subject to location-based attacks including so-called *mafia fraud attack* and *terrorist attack* (Desmedt, 1988). These type of attacks happen even if cryptographic protocols are used. Mafia fraud attack (sometimes referred to as distance fraud attack) is a man-in-the-middle relay attack. The attacker simply relays messages between two honest parties involved in a protocol and makes the two parties believe that they are in close proximity. The mafia fraud attack is specially effective against RFID because an RFID reader is supposed to scan only tags within its communication range. Terrorist attack is a more sophisticated variation of mafia fraud attack in a sense that an attacker can collaborate with a dishonest party involved in a protocol.
- *Side channel analysis:* Timing information, computation fault, power consumption, electromagnetic radiation, etc are typical side channel information that may expose secret information stored in RFID tags. Due to the characteristics of RFID tags, electromagnetic radiation based non-invasive analysis becomes more viable than invasive analysis.

1.3 Security requirements for a secure RFID system

In response to the security threats mentioned above, one should implement cryptographic protocols between RFID tag and reader such that it is infeasible for malicious parties to realise the security threats. The desirable security properties of a cryptographic protocol for RFID are described below.

- *Mutual authentication between tag and reader/back-end server:* In order to prevent EPC numbers from being harvested by malicious parties, reader-to-tag authentication must be provided. In addition, the server should not waste its computational resource on verifying and identifying fake tags. Therefore, RFID readers should also authenticate tags before forwarding legitimate tags to the server for identification.
- *Privacy protection:* In order to prevent a tag from being tracked by malicious parties, it is not sufficient to avoid communicating the tag's EPC number in clear text. Indeed, the information exchanged during different authentication sessions should not help a malicious party to trace a tag. We call such a property *unlinkability*. We refer to a protocol that provides both secure authentication and unlinkability as *a privacy-preserving authentication protocol*. A common approach to provide privacy-preserving authentication is to use pseudonym. More specifically, for each authentication session, a tag uses a different *temporary identity* called pseudonym to communicate with an RFID reader.

- *Forward security:* As an RFID tag is not a tamper-proof device, it can be easily stolen and dissected to reveal secret information stored in the memory of the tag. Many authentication protocols for RFID including (Le et al., 2007) have taken this threat into account by providing a security property called *forward security*. In the case of a privacy-preserving authentication protocol, forward security guarantees that all of authentication sessions of a tag happened before the tag's secret is revealed remains unlinkable. In other words, the piracy of the tag is protected up to the point of the loss of the secret information. A well-known method to achieve forward security is to update the secret key frequently (say, after every authentication session). Once a secret key is revealed, the previous authentication sessions that are associated with old and unknown secret keys are unlinkable. Updating secret keys regularly might also have positive impact on providing privacy-preserving as a tag possesses different keys during different authentication sessions. Unfortunately, updating the secret key interactively between a reader and a tag is often subject to de-synchronisation of secret, i.e., the attacker can cause the reader and the tag to posses different keys which makes future communication impossible.
- *Secure key exchange:* Wireless communication is vulnerable to eavesdropping. To prevent sensitive information like EPC and secret key from being eavesdropped, the information exchanged between a tag and a reader or back-end server should be encrypted. That requires a fresh session key for each interrogation session.
- *Secure tag location:* To defeat location-based attacks like mafia fraud attack, it must be possible for two parties involved in a protocol to measure (at least approximately) the distance between them. A common method is to use round-trip time of messages exchanged between two parties to estimate the distance. Brands and Chaum presented such a protocol which they called distance-bounding protocol in Brands and Chaum (1994).
- *Availability and dependability:* Considering the huge amount of tags that would be live in a whole RFID ecosystem, a protocol designed for RFID should make it possible to filter out unwanted traffic as early as possible. The back-end server and the middle-ware layer should not be overwhelmed by the amount of illegitimate or unnecessary data. We also want any legitimate RFID tag to be securely authenticated and correctly identified at the back-end server. On the other hand, illegitimate tags should be rejected with an overwhelming probability.
- *Immunity against side channel analysis:* RFID tags are certainly an easy target to be captured for side channel analysis. Furthermore, the hardware is cheap and one must take this constraint into account when designing techniques to prevent side channel analysis.

1.4 Contribution of this paper

In this paper, we first review the most representative works in RFID security. These works include cryptographic protocols for RFID tags, security models for RFID and countermeasures against some attacks like location-based attacks and side channel analysis. We then point out some of the issues that need more work and consideration.

In the second part of this paper, we attempt to solve some of open issues related to a special kind of cryptographic protocols for RFID called grouping-proof protocols. A grouping-proof protocol enables multiple RFID tags to be scanned at once such that their co-existence is guaranteed. One typical application of a grouping-proof protocol is to scan tags that are supposed to stay *together*. For example, RFID tags attached on different parts of a car should be located near each other. We point out that all of the previous grouping-proof protocols in Juels (2004), Saitoh and Sakurai (2005), Piramuthu (2006), Lin et al. (2007) and Burmester et al. (2008) suffer from scalability problem. More specifically, a reader has to relay messages from one tag to another tag which makes it difficult to scan a large number of tags at the same time. Our proposed protocol aims to solve this problem by removing the requirement to relay messages among tags. An important part of designing a secure protocol is to define a security model in which the term secure correctly captures our intuition about real-world security of the protocol. We argue that this task has not been done adequately in previous works. In particular, no previous work addresses *mafia fraud attack* presented in Desmedt (1988). Mafia fraud attack is simply a relay attack where an attacker relays messages exchanged between a reader and tags. As noted in Duc and Kim (2010a), all of grouping-proof protocols for RFID are inherently insecure against this attack because the attacker can relay messages exchange between a reader and tags that reside out of the communication range of the reader. The result is an invalid proof that contains tags not in the communication range of the reader at the time of interrogation. Indeed, a security model that does not address this issue cannot be a proper security model for grouping-proof protocols because it would be impossible to prove the security. In practice, we can somehow mitigate this attack by using a distance bounding protocol (Brands and Chaum, 1994) so that a relay attacker does not have sufficient time to relay messages out of the communication range of the reader. Indeed, some of previous protocols (Saitoh and Sakurai, 2005; Lin et al., 2007) make use of timestamp which is actually used to defeat replay attack. However, this prevention method works well if an interrogation session lasts as short as possible. Since a reader has to relay messages among tags in previous protocols, a protocol session can be prolonged which makes mafia fraud attack more feasible. Therefore, it is also important to solve the scalability problem in order to defeat mafia fraud attack. Note that, the use of timestamp does not mean that we do not need to take mafia attack into account when defining a security notion for secure grouping-proof protocols. After all, mafia fraud attack is always feasible from a theoretical point of view. In fact, in Chandran et al. (2009), the authors showed that it is impossible to securely verify the geographic location of a device. Another issue when defining a security model for grouping-proof protocol is that the verifier has no knowledge of what or how many tags are actually in the communication range of a reader. Therefore, we cannot achieve security at all if a reader is allowed to behave maliciously in an arbitrary way. For example, a reader can deliberately avoid scanning some tags resulting in an invalid co-existence proof. In this paper, we present a secure model for secure grouping-proof protocols which takes the above issues into account. In particular, we put the following assumptions in our proposed security model:

- Relaying messages out of the communication range of a reader is not allowed. We address this assumption by restricting the adversary's access to the tag oracle during the last phase of an experiment in which the adversary interacts with a set of oracles,

receives a challenge and attempts to solve the challenge. We shall discuss this in more details in Section 4.

- The reader is trusted to execute the protocol fruitfully but it may report an invalid co-existence proof to the verifier. In particular, before reporting a valid proof to the verifier, a dishonest reader may try to remove a tag from the proof, replace a tag in the proof with another tag or add another tag to the proof. In practice, the protocol can be implemented in a tamper-proof hardware and a proof is assembled and sent to the verifier by the reader in software (and therefore is subject to malicious behaviours of a reader).

It is important to note that, none of previous protocol appears to be secure in a weaker assumption. We then propose a grouping-proof protocol for RFID by using a (n, n)-secret sharing scheme (also referred to as unanimous consent control in Menezes et al. (2010). The goal of using a (n, n)-secret sharing scheme in our protocol is to let n tags sign n different challenges. The n challenges are n shared secrets of a number which is randomly chosen by the verifier. The threshold property of a (n, n)-secret sharing scheme guarantees that n signed challenges are *tied together*. We then prove the security of our protocol.

2 Current research on RFID security

For the rest of this paper, we will use the notations summarised in Table 1.

Table 1 Notations

Notation	*Description*
Ber_η	A Bernoulli distribution with expected value η
D	Tag database at the back-end server
$G(.)$, $H(.)$	Cryptographic hash functions
$f(.)$	Pseudorandom function
K_i	Secret key of tag $\mathcal{T}_i$
$\text{MAC}_K[.]$	Message authentication code (MAC) with secret key K
P	A co-existence proof of multiple tags
$\mathcal{R}$	Reader
$\text{SK}_K[.]$	Symmetric encryption with secret key K
TS	Timestamp
$\mathcal{T}_i$	An RFID tag
$\mathcal{V}$	Verifier (back-end database)

2.1 Cryptographic protocols for secure RFID

We now summarise some of the most well-known cryptographic protocols designed for RFID. These protocols are called lightweight protocols as they use only cryptographic

primitives that require low computational resources. An example of such primitives is cryptographic hash function.

2.1.1 Security features in Gen-2 specification and other compliant protocols

The industry recognised the security threats to RFID very early. In this section, we will discuss security features in the most popular industrial standard for RFID tags at the moment, the Gen-2 specification by EPCglobal Inc (2009).

Reader-to-tag and tag-to-reader authentication

The Gen2 specification does not provide true tag-to-reader and reader-to-tag authentication. A Gen-2-compliant RFID tag simply backscatters its EPC number once being queried by a compatible reader so that the tag can be identified later at the back-end server. Unfortunately, this clearly leaves Gen-2 tags vulnerable to cloning threats since any compatible reader can harvest EPC numbers. A Gen-2-compliant reader is required to be authenticated only when it needs to read or write directly from or to a tag's memory. To do so, a reader and a tag share a common 32-bit secret key (called *access password*). The reader-to-tag authentication protocol is implemented as follows:

1 The tag must already be selected and identified (*acknowledged* state). A reader starts by sending a request to the tag (Req_RN command).

2 The tag responses with 16-bit random number $RN16$.

3 The reader takes the first 16 bits of the access password (MSB first, the second half of access password is used when the reader needs to access the tag again), denoted as $APwd16$, and computes its authentication token as $t = APwd16 \oplus \text{RN16}$.

4 Once receiving the reader's authentication token t, the tag computes $APwd16' = t \oplus RN16$. If $Pwd16'$ does not match with the tag's version of the access password, the tag rejects the reader. Otherwise, the reader is successfully authenticated (open or secured state).

The above protocol is clearly not secure against eavesdropping, i.e., passive adversaries. An eavesdropper can listen to the communication channel between the tag and the reader and collect t and $RN16$. A half of the access password is then revealed by computing $t \oplus RN16$. The Gen-2 specification recommends that reading and writing to a tag's memory should be done in a physically secure location (so that eavesdropping is not possible). However, the assumption that the protocol is carried out in a secure location is too strong. Indeed, if reading and writing can be done in a secure location, no authentication is required.

Privacy protection by disabling tags

The Gen-2 specification proposes a rather conservative method to provide privacy protection, that is to permanently disable a tag, e.g., at the point-of-sale in a supermarket. A Gen-2-compliant tag can be *killed* after receiving a kill command. A reader-to-tag authentication protocol similar to the one described above must be successfully

completed before the tag accepts the kill command. Indeed, the authentication protocol is carried out twice, each using one half of a 32-bit kill password.

While disabling a tag is obviously an effective countermeasure against illegal tracking, it is arguably over-killed. In many scenarios like tracking animal, smart home appliances, etc., a tag should not be permanently disabled. Furthermore, in case of supply chain management, a tag is still likely helpful in many ways after the item is purchased (e.g., for warranty purpose).

Many researchers have attempted to improve the security of Gen-2 tags by designing cryptographic protocols which employ primitives supported in the Gen-2 specification. These compliant protocols are usually referred to as *ultra-lightweight protocols*. The basic idea of these protocols is to provide RFID security using ultra-lightweight primitives like EXOR, ADD, MULTIPLY, AND, OR, NOT, ROTATE, random number generator, and CRC. In this regard, we have the following protocols: Karthikeyan and Nesterenko (2005), Chien and Chen (2007), Peris-Lopez et al. (2006a, 2006b, 2006c), Chien (2007) and Konidala et al. (2007). Unfortunately, these protocols generally offer weak security strength because the weak cryptographic primitives are being used.

2.1.2 Ohkubo-Suzuki-Kinishita protocol and other hash-based protocols

One of the most famous protocols for RFID is the Ohkubo-Suzuki-Kinishita protocol (Ohkubo et al., 2004) which has triggered a significant number of followed-up papers. The protocol assumes that a tag can compute two cryptographic hash functions, $G(.)$ and $H(.)$. A tag i is given an initial EPC number s_i^1 which is also stored in the database at the back-end server. After each interrogation session, the EPC number is updated in a hash chaining fashion, that is $s_i^{k+1} = H\left(s_i^k\right)$. The goals of updating the EPC numbers are two-fold:

- to provide privacy protection by using a different EPC number in each authentication session
- to provide forward-security as it is infeasible to compute previous EPC number from the current EPC number due to the pre-image resistance property of a cryptographic hash function.

During the kth authentication session, a tag computes its authentication token r_i^k as the hash of its current EPC number, i.e., $r_i^k = G\left(s_i^k\right)$. To verify a tag, the server starts from the initial EPC of all tags in the database and compute $G\left(s_i^1\right)$, $G\left(s_i^2\right)$,..., $G\left(s_i^k\right)$ for i =1, 2, ···, n until a match is found. The Ohkubo-Suzuki-Kinishita protocol provides privacy-preserving, forward security and tag-to-server authentication. However, it does not provide server-to-tag authentication. In addition, the server has to go through the whole tag database and compute the hash chains to identify a tag. This makes the server an attractive target for denial-of-service attacks.

Many other hash-based protocols are proposed to address the weakness of the Ohkubo-Suzuki-Kinishita protocol like Weis et al. (2004), Chien (2006) and Avoine and Oechslin (2005).

2.1.3 HB$^+$ protocol

HB$^+$ protocol represents a new approach in designing a secure lightweight protocol for RFID tags (Juels and Weis, 2005). Instead of using known primitives like hash function, HB$^+$ employs a new cryptographic primitive called binary-inner product. The *binary inner product* of two k-bit numbers a and x, denoted as $a \cdot x$ is computed as follows: $a \cdot x = (a_0 \wedge x_0) \oplus (a_1 \wedge x_1) \oplus \cdots \oplus (a_{k-1} \wedge x_{k-1})$.

A foundation for the security of a protocol is a computational hard problem. In the case of HB$^+$ protocol, the hard problem is *learning parity with noise* (LPN for short) problem. LPN problem can be described as follows: given m pairs, each of the form (a^i, z^i), where a^i is k-bit randomly chosen number and $z^i = (a^i \cdot x) \oplus \gamma^i$; the bit γ^i (called the noise bit) is generated from a Bernoulli distribution with expected value $\eta \in \left(0, \frac{1}{2}\right)$ (denoted as Ber_η); the problem is to find the unknown k-bit number x.

HB$^+$ protocol repeats a basic authentication protocol for r times where r is a security parameter. A reader $\mathcal{R}$ and a tag $\mathcal{T}$ share two k-bit secret keys x and y. Because of the noise bit γ, after r rounds of the basic authentication protocol, the reader should accept the tag if roughly $r\eta$ responses from the tag are invalid. We illustrate HB$^+$ protocol in Figure 1.

Figure 1 One round of HB$^+$ authentication protocol

$\mathcal{T}(x,y)$		$\mathcal{R}(x,y)$
$\gamma \leftarrow \text{Ber}_\eta$		
$b \in_R \{0,1\}^k$	$\xrightarrow{b}$	
	$\xleftarrow{a}$	$a \in_R \{0,1\}^k$
$z = (a \cdot x) \oplus (b \cdot y) \oplus \gamma$	$\xrightarrow{z}$	
		Verify $z = (a \cdot x) \oplus (b \cdot y)$

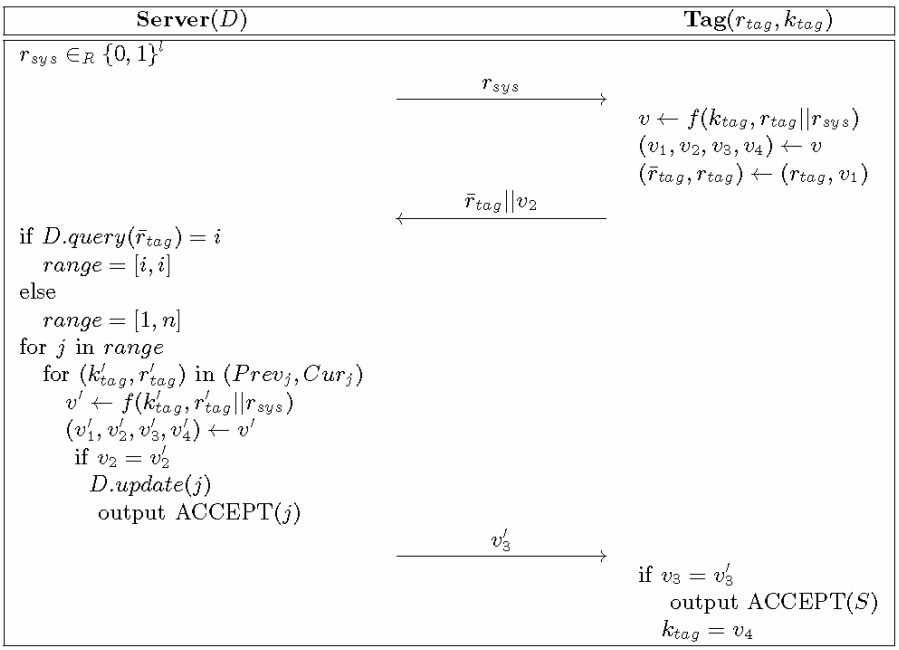
Unfortunately, HB$^+$ is not secure against man-in-the-middle attacks (Gilbert et al., 2004). Several attempts (Bringer et al., 2006; Munilla and Peinado, 2007; Gilbert et al., 2008) to secure HB$^+$ against man-in-the-middle attack have failed so far. In addition, HB$^+$ protocol does not consider other important security properties including anonymity and forward security.

2.1.4 O-FRAP and O-RAP protocols

O-FRAP which stands for optimistic forward secure RFID authentication protocol (Le et al., 2007) is the first full-fledged protocol for RFID. The protocol provides mutual authentication, privacy-preserving, forward security and a mechanism to prevent de-synchronisation of secret key attack. O-FRAP can also be extended to include key exchange. O-FRAP uses pseudonym approach to protect privacy of a tag. That is for each session, a random number r_{tag} is used as a temporary identity of the tag. Each tag also shares with the server a secret key k_{tag}. The secret key is also updated after each session. To deal with de-synchronisation of secret key attack, the server stores two versions of the shared secret key in the database, one previously-used key and one currently-used key. The tag pseudonym is also stored in the database in a similar fashion. In addition, the

database is indexed with tag pseudonym for quickly looking up a tag given the tag pseudonym. The O-FRAP protocol between the server $\mathcal{S}$ and a tag $\mathcal{T}$ is depicted in Figure 2.

Figure 2 The O-FRAP protocol



O-FRAP protocol is shown to be secure in a security model called *universal composability framework*. This security model guarantees the security of the protocol running in isolation as well as a component of a bigger system.

O-RAP which stands for *optimistic RFID authentication protocol* is a simplified version of O-FRAP which appeared in Burnmester et al. (2009). O-RAP is essentially O-FRAP but without a key updating procedure. As a result, O-RAP does not provide forward security and the back-end server does not need to store two versions of key for each tag.

Unfortunately, several weaknesses of O-FRAP and O-RAP were reported in Ouafi and Phan (2008) and Duc and Kim (2010b). These weaknesses exploit flaws in the server's tag look-up procedure and key updating method.

2.1.5 PUF-based protocols

A relatively new approach in designing lightweight authentication protocols for RFID is to exploit the physical properties of an RFID tag to produce a primitive called *physically unclonable function* (PUF) (Devadas et al., 2008). A PUF does not require any expensive implementation. Instead, it can be constructed from anomalies during the manufacturing of tags. Generally, these anomalies would result in a unique PUF for each tag. A PUF takes an arbitrary input and produces an unpredictable output, in such a way that the same

input always results in the same output. A PUF-based authentication protocol requires tags to be enrolled into a system before deployment. In the enrolment phase, a tag is queried a large number of times, each with a different challenge. These challenges are fed into a PUF module of the tag to produce the corresponding number of responses. These challenge-response pairs (CRPs) are securely stored and later used to authenticate the tag. When a reader authenticates a tag, it picks a challenge in the CRP table and queries the tag. If the tag's response is matched with the response in the CRP table, then the tag is authenticated. Note that, a CRP pair should be used only once to prevent replay attack.

We can see that PUF-based protocols should be very lightweight because no cryptography is involved. However, we need to store a large number of CRPs for each tag.

2.1.6 Grouping-proof protocols for RFID tags

Yoking-proof (Juels, 2004) is the first protocol to address the multiple tag scanning problem. The protocol enables a RFID reader ($\mathcal{R}$) to produce a co-existence proof of two RFID tags ($\mathcal{T}_1$ and $\mathcal{T}_2$). The main idea in the yoking-proof protocol is to let two tags sign each other's random number (in order to prove each other's presence). The signing algorithm is a MAC scheme. Yoking-proof proceeds as follows:

1 $\mathcal{R} \to \mathcal{T}_1$: request

2 $\mathcal{T}_1 \to \mathcal{R} : \mathcal{T}_1, r_1$ where r_1 is chosen at random

3 $\mathcal{R} \to \mathcal{T}_2 : r_1$

4 $\mathcal{T}_2 \to \mathcal{R} : \mathcal{T}_2, r_2, m_2 = \mathrm{MAC}_{K_2}[r_1]$, where r_1 is chosen at random

5 $\mathcal{R} \to \mathcal{T}_1 : r_2$

6 $\mathcal{T}_1 \to \mathcal{R} : m_1 = \mathrm{MAC}_{K_1}[r_2]$

7 $\mathcal{R} \to \mathcal{S} : \mathcal{P} = (\mathcal{T}_1, r_1, m_1, \mathcal{T}_2, r_2, m_2)$.

The co-existence proof P can be verified by checking the validity of two MAC values m_1 and m_2. Yoking-proof was later extended to support multiple scanning of more than two tags as well as to enhance the security.

Saitoh and Sakurai (2005) showed that yoking-proof is vulnerable to replay attack. The reason is that the two messages m_1 and m_2 are not guaranteed to be generated in the same session. As a result, an attacker can reuse m_2 in another session which results in a forged proof. To prevent this attack, Saitoh and Sakurai proposed a timestamp-based yoking-proof which requires an online verifier to issue a timestamp TS for each session. TS is included in each co-existence proof and needs to be signed by both $\mathcal{T}_1$ and $\mathcal{T}_2$. The online verifier accepts a proof only if it is received within the expected lifespan of one interrogation session.

In Saitoh and Sakurai (2005), the authors also proposed another protocol which allows the simultaneous scanning of more than two tags. The protocol is called grouping-proof and requires an additional entity called pallet tag. The pallet tag has more

computational resource than an RFID tag and acts as a representative of all RFID tags that are in the same package with the pallet tag.

1 $\mathcal{V} \to \mathcal{R}$: TS

2 $\mathcal{R} \to \mathcal{T}_1, \mathcal{T}_1, \cdots \mathcal{T}_n$: TS

3 $\mathcal{T}_i \to \mathcal{R} : m_i = \mathrm{MAC}_{K_i}[\mathrm{TS}]$, for $i = 1, 2, \cdots, n$

4 $\mathcal{R} \to$ pallet tag : TS, $m_1, m_2, \cdots, m_n$

5 pallet tag $\to \mathcal{R} : \mathrm{C}_P = \mathrm{SK}_K\left[\mathrm{TS}, m_1, m_2, \cdots, m_n\right]$

6 $\mathcal{R} \to \mathcal{V}$: $P = \left(\mathrm{TS}, \mathrm{C}_P, \mathcal{T}_1, \mathcal{T}_2, \cdots \mathcal{T}_n\right)$.

The proof P is subject to timestamp verification by the online verifier in order to prevent replay attacks. Then, the co-existence proof is verified by checking the validity of each m_i.

Piramuthu (2006) proposed another variant of yoking-proof which does not use timestamp to prevent replay attack (Piramuthu, 2006). The main idea is to let the tag $\mathcal{T}_1$ sign both its own random number r_1 and the tag $\mathcal{T}_2$'s MAC m^2. As a result, neither m_1 nor m_2 can be reused in another session.

Lin et al. pointed out that Piramuthu's protocol may suffer from interference problem when multiple readers are represent. More specifically, when a tag is queried by two readers at the same time, the tag might have problem determining what messages to sign if no proper session management is present. Lin et al. also showed that the timestamp-based yoking-proof presented in Saitoh and Sakurai (2005) is indeed not secure against replay attack. An attacker can query the tag T1 with many different timestamp values in the future. Then, the responses from $\mathcal{T}_1$ can be used to query the tag $\mathcal{T}_1$ which is in a different location and at different times. To counter the problem, Lin et al. proposed another variant of timestamp-based yoking proof in which the verifier encrypts a timestamp value before sending to a reader.

Lin et al. also proposed another grouping-proof protocol for any number of tags but without using a pallet tag. The protocol uses a method called timestamp-chaining. That is, to produce a co-existence proof of n tags, the first tag signs the hashed timestamp value TS1 from the reader and the ith tag signs the hash of timestamp TSi and the $(i-1)$th tag's MAC. The reader is in charge of forwarding the hashes and assigning proper values for each timestamp.

Burmester et al. (2008) proposed two grouping-proof protocols which employ a different approach. In particular, in the Burmester et al.'s protocols, a tag does not use MAC to sign its challenge. However, in order to produce a co-existence proof of two tags, Burmester et al.'s protocols assume that the two tags share a group id (denoted as gid) and a common secret key (denoted as K_g). Each tag also maintains a counter variable c such that c is increased by 1 after each successful protocol session. We describe below only one protocol in Burmester et al. (2008). The other protocol has the same design but provides privacy protection by updating the group id after each session.

1 $\mathcal{R} \to \mathcal{T}_1, \mathcal{T}_2 : r_{sys}$ chosen at random.

2 $\mathcal{T}_1, \mathcal{T}_2 \rightarrow \mathcal{R}$: gid.

3 $\mathcal{R} \rightarrow \mathcal{T}_1, \mathcal{T}_2$: $\mathcal{T}_1$ and $\mathcal{T}_2$ are linked.

4 $\mathcal{T}_1 \rightarrow \mathcal{R} : c, r_1$ where $r_1 \| s_1 = f\left(r_{sys}, c, K_g\right)$.

5 $\mathcal{R} \rightarrow \mathcal{T}_2 : r_1, c$.

6 $\mathcal{T}_2 \rightarrow \mathcal{R} : t_2, s_2$ if $r_1 = r_2$ where $r_2 \| s_2 = f\left(r_{sys}, c, K_g\right)$ and $t_2 = f(r_2, c, K_2)$. If $r_1 \neq r_2, \mathcal{T}_2$ terminates the protocol.

7 $\mathcal{R} \rightarrow \mathcal{T}_1 : s_2$.

8 $\mathcal{T}_1 \rightarrow \mathcal{R} : t_1$ if $s_1 = s_2$ where $t_1 = f(r_1, c, K_1)$. $\mathcal{T}_1$ also update its counter value $c = c + 1$. If $s_1 \neq s_2$, $\mathcal{T}_1$ terminates the protocol.

9 $\mathcal{R} \rightarrow \mathcal{V}$: $P = \left(r_{sys}, gid, c, r_1, t_1, r_2, t_2\right)$.

2.1.7 Distance-bounding protocols

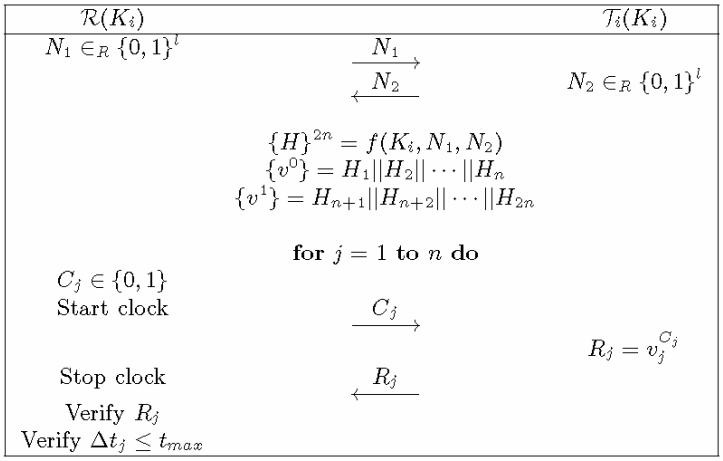
RFID protocols are inherently insecure against mafia fraud attack which was suggested by Desmedt (1988). It does not matter what type of cryptographic protocols is used, an attacker can always relay messages between a reader and a tag which is not in the communication range of the reader (and therefore is not supposed to be scanned).

Brands and Chaum (1994) were the first to propose a practical countermeasure against the mafia fraud attack. Since the mafia fraud attack is about faking the location, it is necessary to verify the location of each party involved in a protocol. However, there is no way to measure the distance between two autonomous parties. Therefore, Brands and Chaum suggested that round-trip time can be used to approximately measure the distance. While using round-trip time is not a new idea, Brands and Chaum pointed out that the round-trip time should be as short as possible and the measurement should be repeated multiple times to improve accuracy. Brands and Chaum called their countermeasure distance-bounding protocols.

Hancke-Kuhn's (2005) distance-bounding protocol is the first protocol which addresses mafia-fraud-attack against RFID protocols. The key idea is to repeat a simple authentication step multiple times so that each step can be complete in a very short time. Let t_{max} be the maximum time taken by one simple authentication step. A tag is accepted only if each simple authentication step completes successfully and within t_{max} amount of time, i.e., $\Delta t_j \leq t_{max}$. The protocol between a tag $\mathcal{T}_\rangle$ and a reader $\mathcal{R}$ is illustrated in Figure 3.

Note that, recently Chandran et al. (2009) showed that it is impossible to securely measure the distance between two autonomous parties in any manner. Their result confirms that the use of distance-bounding protocols only provide practical defence against mafia fraud attack.

Figure 3 Hancke-Kuhn's distance-bounding protocol



2.2 Security Models for RFID

A security model is essential to analyse the security of a cryptosystem and it is no exception to a cryptographic protocol for RFID. The most important part of a security model is to rigorously define what we means by a *secure system*. In case of RFID, we need to define secure mutual authentication, privacy-preserving, forward security and secure key exchange so that the definition correctly captures our intuition about security properties of a secure protocol for RFID mentioned in Section 2. We briefly summarise two security models for RFID below.

2.2.1 Vaudenay's security model for RFID

Vaudenay presented an RFID-specific security model for RFID in Vaudenay (2007). The Vaudenay's model is a classical type of a security model in a sense that a number of oracles to provide information for the adversary are specified and the security is defined via a game between a challenger and an adversary. In the Vaudenay's model, there are one reader (i.e., the back-end server and the readers are seen as a single entity) and a tag population. An RFID authentication protocol is viewed as a collection of the following algorithms:

- SetupReader(.) is an efficient algorithm which takes a security parameter s as the input and returns a public key K_P and a secret key K_S for the reader.
- SetupTag(.) is an efficient algorithm which takes a security parameter *s*, the reader's public key K_P and an object identity ID as the input and returns a secret key and the initial internal state for a tag.
- An efficient two-party protocol Π between a tag and a reader such that at the end of the protocol (assuming that the reader has been set up properly and tag population has been created) the reader outputs either ⊥ or ID of the tag.

Definition 2.1: Correctness of an RFID authentication protocol an RFID authentication protocol is said to be correct if, for a negligible probability, the reader's output is $\perp$ and the tag is illegitimate, or ID and the tag ID is legitimate.

In order to provide a realistic security definition for RFID protocols, Vaudenay observed that in practice, it is impossible for the adversary to access to all tags available at once. Therefore, in the security model, the adversary should be allowed to access to some tags at once. Vaudenay took this observation into account by introducing two special oracles called DrawTag(.) and FreeTag(.). All oracles are defined below:

- CreateTag(b, ID): This oracle allows the adversary to create either a legitimate tag using the SetupTag(.) algorithm ($b = 1$) or an illegitimate one ($b = 0$). The resulting tag has an identity ID.
- DrawTag(D): This oracle draws n tags from the tag population according to a probability distribution D. The drawn tags can be either legitimate or illegitimate tags. Note that, in order to make sense in defining a security notion for privacy-preserving, even a tag is drawn twice, it should be given two different IDs.
- FreeTag(ID): This oracle releases a drawn tag with ID and renders it unreachable.
- Launch(.): This oracle runs a new instance π of the protocol Π. π is returned to the adversary.
- SendReader(m, π, m'): This oracle replaces a message m sent to the reader m' with in a protocol instance π.
- SendTag(m, π, m'): This oracle replaces a message m sent to a tag with m' in a protocol instance π.
- Result(π): This oracle returns 0 if at the end of an instance protocol π the reader outputs $\perp$ and 1 if otherwise.
- Corrupt(ID): This oracle returns the current state of the tag ID.

As usual, one can classify different types of attacks based on how the adversary interacts with the above oracles. Vaudenay distinguished the following types of attacks on RFID protocols:

- *weak attack:* in this type of attack, the adversary is not given access the Corrupt(.) oracle
- *narrow attack:* in this type of attack, the adversary is not given access to the Result(.) oracle
- *forward attack:* in this type of attack, the adversary can use the Corrupt(.) oracle only once
- *strong attack:* in this type of attack, the adversary can call all oracles in any fashion
- *destructive attack:* this one is similar to the strong attack except that the adversary is not allowed to interact with a tag after corrupting the tag.

The narrow attack can be combined with other types of attacks resulting in new attacks like narrow-strong, narrow-destructive, narrow-forward and narrow-weak attacks. The security definition for a secure RFID authentication protocol against strong attack is given as follows:

Definition 2.2: (Secure RFID authentication) An RFID scheme is said to provide secure authentication if for any polynomial-bounded adversary, the following probability is negligible: there exists one protocol instance π launched by the adversary in which the reader identified an uncorrupted legitimate tag ID but there was not matching conversation with this tag.

The above definition implies that if the reader authenticates and identifies a legitimate tag but never actually communicates with it, then the adversary must have impersonated the tag. However, it is still quite vague since there is no clear way to quantify the security. In addition, the definition only accommodates tag-to-reader authentication but not reader-to-tag authentication.

The security notion for privacy protection in RFID protocols is also given in Vaudenay (2007). We recall the definition below.

Definition 2.3: (Privacy-preserving RFID protocol) Consider an adversary working in two phases: the querying phase in which the adversary interacts with a set of oracles and the analysis phase without any oracle access. Before entering the second phase, the adversary receives his challenge as a set of tags drawn from the DrawTag(.) oracle. At the final step, the adversary should output either true or false. The adversary wins if his output is true. An RFID protocol provides privacy protection if all polynomial-bounded adversaries are trivial in a sense that the adversary can make no effective use of protocol messages.

The above security definition for privacy allowed Vaudenay to prove some interesting possibility and impossibility results as follows.

- A pseudorandom function is sufficient to construct a secure privacy-preserving authentication protocol for RFID under weak attack.
- An RFID protocol that achieves narrow-strong privacy can be used to construct a secure key agreement protocol. In other words, a secure key agreement protocol is a minimal requirement to build a narrow-strong private RFID protocol.
- A protocol that achieves strong privacy is impossible to realise.

2.2.2 Universal composability framework for forward secure RFID authentication

Universal composable framework (Canetti, 2007) is a security model whose goal is to ensure that a secure protocol should remain secure even when running in a complex system. Essentially, a security model of this kind should define a so-called *ideal functionality* in which a cryptographic task is implemented assuming that a trusted third party exists. In an ideal functionality, each party (including the adversary attacking the cryptographic task) who wishes to achieve his desired security goals only communicates with the trusted party. The security is defined as the indistinguishability between the ideal functionality and a real-world protocol. In Le et al. (2007), an ideal functionality for a

secure RFID protocol called $\mathcal{F}_{aauth}$ which defined mutual authentication, privacy-preserving and forward security was presented. Before describing $\mathcal{F}_{aauth}$, we summarise the following notations used here:

- $\mathcal{A}$: The adversary who communicates directly with the ideal functionality instead of intercepting with other parties in the system.
- P: An party which can be either a tag or the server. Like the Vaudenay's model, the authors of $\mathcal{F}_{aauth}$ also considered the back-end server and the reader as a single entity.
- *type*(P): The type of a party P which indicates whether P is a tag or the server.
- *sid*: Sub-session identifier. In an ideal functionality, the whole lifetime of a protocol is called a session and one instance of the protocol (in the view of one party) is called a sub-session. Each sub-session is uniquely identified with a *sid*.
- *active*(P): A collection of identifiers for preceding incomplete sub-sessions involving P.
- *state*(P): Internal state of a party P.

The ideal functionality $\mathcal{F}_{aauth}$ should now be defined. Essentially, it maintains a database and implements a number of interfaces for other parties to call. Interested readers are referred to Le et al. (2007) and Burnmester et al. (2009) for the detail description of these interfaces.

3 Open issues in RFID security

Even though many works have been done to counter security threats to RFID technology, many issues are still unsolved and some others need further investigation. Those issues include:

- *Lack of research on lightweight cryptographic primitives:* Even though there are on-going effort to put traditional cryptographic primitives like AES and elliptic curve operations on low-cost hardware, this approach might not be able to achieve the desirable performance for low-cost RFID tags. We argue that it is preferable to look for new cryptographic primitives which are lightweight and offer reasonable security strength. The binary inner product operation (with the corresponding hard problem, LPN problem) is a good example. Indeed, we need to work on many lightweight cryptographic primitives including pseudorandom number generator, symmetric encryption and MAC.
- *Study on possibility and impossibility of certain cryptographic tasks for RFID:* All of authentication protocols for RFID that provide forward security employ a so-called *interactive key-evolving protocol* to update the secret key at the tag and the server/reader. We suspect that it is impossible to realise an interactive key-evolving protocol that is robust against de-synchronisation of secret (which may imply the

loss of forward security). Another important task is to specify what kind of cryptographic primitive is required to realise secure authentication protocols for RFID with different security notions defined in Vaudenay's security model.

- *Study on new security models for RFID:* Multiple tag scanning is an useful technique. However, there are no security model for such a protocol so far. Because of this, the security analysis for multiple tag scanning protocols is very limited. In addition, known security models for RFID ignore the RFID reader as a party of a RFID system. In particular, the level of trust on the RFID reader should play an important role in analysing the security of a protocol. We argue that this might lead to broaden separation between theoretical security and real-world security. We need to develop a new security model or extend existing models that take the RFID reader as an indispensable entity of an RFID system (rather than combine it with the back-end server).
- *Consideration against location-based attacks:* There are surprisingly a few work on location-based attacks on RFID. We need to find alternative approach rather than Hancke-Kuhn distance-bounding protocol and a few of its variations. In addition, the fact that it is theoretically impossible to securely verify physical location of an object must be taken into account when devising security models and definitions for RFID.

4 Security model for a secure grouping-proof protocol

Before designing a protocol to secure certain cryptographic tasks, it is important that one should clearly define the meaning of the term *secure*. In this paper, we present a security model for a secure grouping-proof protocol for RFID tags which addresses mafia fraud attack and the level of trust on an RFID reader. We then define what a secure grouping-proof protocol for RFID tags is. Our security model is a conventional security model in a sense that the adversary is given access to a set of oracles and the term *secure* is defined via a game between a challenger and an adversary. In Burmester et al. (2008), the authors proposed another security model for secure grouping-proof protocol in the universal composable framework. The ideal functionality defined in Burmester et al. (2008) is called $\mathcal{F}_{group}$ which interacts with different involving parties via five interfaces: activate, initiate, link, complete and verify (whereas involving parties do not interact with each other directly). Interested readers are referred to Burmester et al. (2008) for the description of each of $\mathcal{F}_{group}$'s interface. The problem with $\mathcal{F}_{group}$ is that there is no condition for a tag to call $\mathcal{F}_{group}$'s initiate. Indeed, only tags within the communication range of a reader are qualified to make the initiate calls to $\mathcal{F}_{group}$. Unfortunately, the communication range of a reader is not modelled in $\mathcal{F}_{group}$. That is probably why the full security proofs for two protocols in Burmester et al. (2008) are not yet available. Note that, this does not mean security proofs for lightweight authentication protocols for RFID are invalid. In case of an authentication protocol, the goal of the adversary is to impersonate a tag. Simply relaying message between a legitimate tag and a reader does not imply impersonation. It is also worth mentioning that most of previous grouping-proof protocols employ timestamp which makes it difficult to rigorously

analyse their security. We believe that it is better to avoid using a physical object in the description of a protocol but to embed it into the security model or assumption.

We now describe our security model for a secure grouping-proof protocol. First of all, we realise that for a grouping-proof for RFID tags protocol, the primary goal of an adversary is to inject some tags (possibly genuine) into a valid coexistence proof while the tags are not actually in the communication range of the reader. In addition, the adversary might also want remove some tags from a valid co-existence proof. It is also assumed that the reader can behave maliciously but does execute the protocol correctly. When reporting a co-existence proof to the verifier, a malicious reader may try to replace some tags in the proof with different tags, add a tag to the proof or remove a tag from the proof. One can obtain a stronger security notion by allowing a malicious reader to deviate from the protocol in any fashion. However, it is impossible to achieve security because the verifier has no knowledge of what and how many tags are actually in the communication range of the reader. The malicious reader can violate the security by deliberately not scanning some tags. This issue also appears in all of the previous protocols in Juels (2004), Saitoh and Sakurai (2005), Piramuthu (2006), Lin et al. (2007) and Burmester et al. (2008). Indeed, the timestamp-chaining protocol by Lin et al. is vulnerable to malicious behaviours of a reader even if the reader is trusted to execute the protocol correctly. The reason is that before reporting a co-existence proof of n tags to the verifier, the malicious reader can remove some tags at the end of the timestamp chain from the proof without invalidating the proof. We now define a set of oracles that provide information to the adversary:

- The reader(.) oracle: This oracle simulates a reader during a protocol session. That it, it returns the reader's challenge to a tag.
- The corrupt-reader(.) oracle: This oracle corrupts a reader and returns the current state of the reader. The adversary is also allowed to control the reader after this oracle is called.
- The tag(.) oracle: This oracle simulates a tag during a protocol session. That is, it returns the tag's response given a challenge from a reader.
- The verify(.) oracle: This oracle takes a co-existence proof P as input and returns 1 if P is valid and 0 otherwise.

We now define the security notion for a secure grouping-proof protocol via the following game between a challenger and an adversary.

1 The challenger first sets the verifier and a reader and tags up to prepare for the game.

2 In the first phase of the game, the adversary collects information via four oracles: reader(.), tag(.), corrupt-reader(.) and verify(.). These oracles are simulated by the challenger.

3 In the second phase of the game, the challenger gives the adversary a valid proof P of n tags as a challenge. The adversary's goal is to either remove a tag from P or add a new tag to P or replace a tag in P with a different one. In this phase, the adversary is also given access to the corrupt-reader(.) oracle after the challenge proof P is

constructed. However, the tag(.) oracle is not provided to the adversary after the adversary has seen P. This is to reflect our assumption that relay attack is not possible. The adversary should output a new proof P' which satisfies one of its goals.

4 The adversary wins the game if verify (P') returns 1. That is, P' is a valid co-existence proof.

Definition 4.1: A grouping-proof protocol is said to be secure if the winning probability of the adversary in the above game is negligible. That is, for any polynomial-bounded adversary $\mathcal{A}$ and a sufficiently large security parameter k.

$$\text{Prob}[\mathcal{A}\ wins] < \frac{1}{poly(k)}$$

5 A scalable grouping-proof protocol for RFID tags

5.1 Scalability issue of previous grouping-proof protocols

The design of yoking-proof and timestamp-based yoking proof suffers from a serious scalability issue. The reason is that a reader needs to relay messages from one tag to another so that a tag can sign the random numbers that were generated by the other tags. As a result, if the reader wants to produce a co-existence proof of n tags, it is required to relay $n(n-1)$ messages among n tags. The number of relaying messages can be reduced to $(n-1)$ if a proof is constructed in a chaining fashion. That is, the first tag signs the second tag's random number. The second tag signs the third tag's random number and so on. However, this approach might be subject to replay attack if a protocol is not designed carefully. Let's assume that a tag $\mathcal{T}_i$ appears in two chaining proofs. Using $\mathcal{T}_i$ as a connector, an attacker might try to connect the first half of the first proof with the second half of the second proof to produce a forged proof. Nevertheless, this is a significant communication overhead compared to the traditional method of scanning one tag at a time which requires no message to be relayed by the reader. This same problem also appears in other variations of yoking-proof including the previous works (Piramuthu, 2006; Lin et al., 2007; Burmester et al., 2008).

The grouping-proof protocol by Saitoh and Sakurai does not use the same design of yoking-proof. However, it requires a pallet tag which is capable of performing symmetric encryption. This increases the cost of multiple scanning of tags and might not be flexible in practice. For example, in a retail store, items that are scanned at a point-of-sale usually do not have an accompanying pallet tag. In addition, the reader still needs to relay messages from all tags to the pallet tag. In order to scan n tags at once, the reader needs to relay n messages to the pallet tag.

As we pointed out earlier, the lifespan of one protocol session may affect the resilience of a grouping-proof protocol against mafia fraud attack. We believe that it is important to solve the scalability problem of previous grouping-proof protocols, for the sake of not only performance but also security.

5.2 Our proposed grouping-proof protocol

We now propose our grouping-proof protocol for multiple RFID tags which does not suffer from the scalability problem. We use a (n, n)-secret sharing scheme to stop messages being relayed among tags. A (n, n)-secret sharing scheme allows one to split one secret x into n of so called shared secrets such that x can only be reconstructed from the shared secrets if and only if all of n shared secrets are provided. This property is used in our proposed protocol so that each tag can sign its own random number to prove its existence. The random numbers are shared secrets generated by a secret sharing scheme. If the original secret generated by a verifier can be recovered from signed shared secrets that were backscattered by tags, then the proof of co-existence of tags is verified. A (n, n)-secret sharing scheme can be implemented as follows:

- Given a secret x, a dealer chooses $(n-1)$ random numbers $y_1, y_2, \cdots, y_{n-1}$ as the first $(n-1)$ shared secrets.
- The last shared secret y_n is computed by $y_n = x \oplus y_1 \oplus y_2 \oplus \cdots \oplus y_{n-1}$.

It is easy to see that the above protocol achieves perfect security since it is impossible to recover x without any of y1,y2, ••• or yn. In addition, for each randomly chosen x, a shared secret of x is also random. This property is important to prevent replay attack as a shared secret is used as a challenge in our proposed protocol. We now describe our grouping-proof protocol below.

1 $\mathcal{V} \to \mathcal{R}$: x chosen at random. The verifier also sets a time-to-live on x such that a co-existence proof associated with x must be received within the lifespan of x (which is approximately the time taken by one interrogation session of a reader).

2 $\mathcal{R} \to \mathcal{T}_i : x, y_i$ for $i = 1, 2, \cdots, n$ where $y_1, y_2, \cdots$, and y_n are n shared secrets of x.

3 $\mathcal{T}_i \to \mathcal{R} : \mathcal{T}_i, m_i = \mathrm{MAC}_{K_i}[y_i, x]$, for $i = 1, 2, \cdots, n$.

4 $\mathcal{R} \to \mathcal{V} : P = (\mathcal{T}_1, y_1, m_1, \mathcal{T}_2, y_2, m_2, \cdots, \mathcal{T}_n, y_n, m_n)$.

5 $\mathcal{V}$: The verifier verifies a proof P by checking if P is received within the lifespan of $x = y_1 \oplus y_2 \oplus \cdots \oplus y_n$ and each m_i is valid MAC of the tag $\mathcal{T}_i$ on (x, y_i).

Remark 5.1: Note that, it is important to stress that we do use timestamp in our protocol to prevent a malicious reader from abusing x (i.e., the malicious reader can take x and use shared secrets of x on different tags at different locations and times). However, the way which timestamp is used in our protocol is very different from in previous protocols. More specifically, we do not use timestamp as a challenge to a tag. Instead, only the verifier maintains timestamp for each interrogation session. This allows us to leave 'time-to-live of x' to the security model. Indeed, the fact that a co-existence proof must be received within the lifespan of x fits in the assumption that a reader always executes the protocol correctly until reporting a proof to the verifier. Therefore, we can ignore the use of timestamp in the security proof of our protocol.

We now analyse the success probability of an adversary attacking our protocol. The probability is measured in terms of the success probabilities of adversaries attacking the underlying MAC and secret sharing schemes in the following theorem.

Theorem 5.2: Let α be success probability of an adversary attacking the underlying MAC scheme. Let ϵ be the success probability of an adversary that attacks our proposed grouping-proof protocol, we have:

$$\epsilon = O\left(\alpha + 2^{-\frac{1}{2}}\right)$$

where *l* is the bit length *x* and d is the number of tags in the tag database.

Proof: Let $\mathcal{A}$ be the adversary that attacks our proposed grouping-proof protocol. Given a challenge $P = \left(\mathcal{T}_1, y_1, m_1, \mathcal{T}_2, y_2, m_2, \cdots, \mathcal{T}_n, y_n, m_n\right)$ and let $x = y_1 \oplus y_2 \oplus \cdots \oplus y_n$, A wants to achieve one of the following goals:

- Construct a co-existence proof $P' = \left(\mathcal{T}_1^*, y_1^*, m_1^*, \mathcal{T}_2^*, y_2^*, m_2^*, \cdots, \mathcal{T}_n^*, y_n^*, m_n^*\right)$ such that $\left\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n\right\} \neq \left(\mathcal{T}_1^*, \mathcal{T}_2^*, \cdots, \mathcal{T}_n^*\right); y_1^* \oplus y_2^* \oplus \cdots \oplus y_n^* = x;$ and $m_1^*, m_2^*, \cdots$ and m_n^* are valid MACs of $\mathcal{T}_1^*, \mathcal{T}_2^*, \cdots,$ and $\mathcal{T}_n^*$ on $\left(y_1^*, x\right), \left(y_2^*, x\right), \cdots$ and $\left(y_n^*, x\right),$ respectively. In other words, $\mathcal{A}$ succeeds when it can replace at least one tag that is actually in the communication range of the reader by another tag. We call this type of adversary Type-I adversary.
- Construct a co-existence proof $\left(\mathcal{T}_1^*, y_1^*, m_1^*, \mathcal{T}_2^*, y_2^*, m_2^*, \cdots, \mathcal{T}_{n-1}^*, y_{n-1}^*, m_{n-1}^*\right)$ such that the cardinality of $\left\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n\right\} \setminus \left(\mathcal{T}_1^*, \mathcal{T}_2^*, \cdots, \mathcal{T}_{n-1}^*\right); y_1^* \oplus y_2^* \oplus \cdots \oplus y_{n-1}^* = x;$ and $m_1^*, m_2^*, \cdots$ and m_{n-1}^* are valid MACs of $\mathcal{T}_1^*, \mathcal{T}_2^*, \cdots$ and $\mathcal{T}_{n-1}^*$ on $\left(y_1^*, x\right), \left(y_2^*, x\right), \cdots$ and $\left(y_{n-1}^*, x\right),$ respectively. In other words, the adversary can remove a tag from *P*. We call this type of adversary Type-II adversary.
- Construct a co-existence proof

  $$P' = \left(\mathcal{T}_1^*, y_1^*, m_1^*, \mathcal{T}_2^*, y_2^*, m_2^*, \cdots, \mathcal{T}_n^*, y_n^*, m_n^*, \mathcal{T}_{n+1}^*, y_{n+1}^*, m_{n+1}^*\right)$$

  such $y_1^* \oplus y_2^* \oplus \cdots \oplus y_{n+1}^* = x;$ and $m_1^*, m_2^*, \cdots, m_n^*$ and m_{n+1}^* are valid MACs $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n$ and $\mathcal{T}_{n-1}^*$ on $\left(y_1^*, x\right), \left(y_2^*, x\right), \cdots, \left(y_n^*, x\right)$ and $y_{n+1}^*,$ respectively. In other words, the adversary can add the tag $\mathcal{T}_{n+1}^*$ to *P*. We call this type of adversary Type-III adversary.

In the first phase of the attack, $\mathcal{A}$ is given access to three oracles: the tag(.) oracle, the reader(.) oracle and the verify(.) oracle. The corrupt-reader(.) oracle is not required as $\mathcal{A}$ can eavesdrop *x* itself from challenges sent to tags (except that $\mathcal{A}$ can control the reader after seeing the challenge *P*, however this does not affect the analysis here). The tag(.) oracle is essentially a MAC oracle as it outputs MAC on an input value together with a tag ID. In the second phase of the attack, the adversary can only control the reader

after seeing the challenge P. No oracle access is given in this phase. As usual, we limit the number of calls to oracles and running time of the adversary to be polynomial in security parameters. We analyse the success probability of each type of the adversary below.

Type-I adversary: We distinguish two cases of Type-I adversary as follows:

- Case 1: None of $\left(y_i^*, x\right)$ for i =1, 2, ···, n has not been asked to the tag(.) oracle. In this case, $\mathcal{A}$ is essentially a MAC forger with m_i^* is a forged MAC. Indeed, if $\mathcal{A}$ can forge a MAC, then it is obvious to attack the proposed grouping-proof protocol by constructing a forged MAC on one of (y_i, x) for i =1, 2, ···, n such that the forged MAC is a valid MAC of a tag not in $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n\}$. Therefore, the success probability of A is bounded by the success probability of the MAC adversary.

- Case 2: At least one of $\left(y_i^*, x\right)$ for i =1, 2, ···, n has been asked to the tag oracle. We only consider the case that the adversary try to replace one tag in P with another tag. But it can be easily generalised to the case of replacing more than one tag. Since $\mathcal{A}$ is not supposed to forge a MAC (otherwise, it is easier to attack by executing the scenario of the adversary in the first case) and the tag(.) oracle is not provided in the second phase of the attack, $\mathcal{A}$ can only expect that its query to the tag oracle with $\left(y_i^*, x\right)$ results in $\left(\mathcal{T}_i^*, m_i^*\right)$ such that $\mathcal{T}_i^*$ is not among $\left(\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n\right)$ and y_i^* constitutes a valid shared secret. However, because the underlying (n, n)-secret sharing scheme is perfectly secure and x is randomly chosen for each session, y_i^* has to be one of $y_1, y_2, \cdots, y_n$. In other words, $\mathcal{A}$ succeeds only if one of the pairs (y_i, x) for i = 1, 2, ···, n has been queried to the tag(.) oracle in the querying phase such that the returned tuple $\left(\mathcal{T}_i^*, m_i^*\right)$ satisfies the adversary's goal. As shared secrets are randomly distributed and there are ($d - n$) candidate tags for $\mathcal{T}^*$, the success probability of $\mathcal{A}$ is $\frac{d-n}{d} 2^{-\frac{1}{2}}$.

Type-II adversary: Using the same analysis for Type-I adversary, we can see that the best option that adversary can succeed is to forge a MAC. For example, if the adversary wants to remove $\mathcal{T}_n$ from P, it can forge a MAC of $\mathcal{T}_{n-1}$ on ($y_{n-1} \oplus y_n$, x). The resulting proof P' is $\left(\mathcal{T}_1^*, y_1^*, m_1^*, \mathcal{T}_2^*, y_2^*, m_2^*, \cdots, \mathcal{T}_{n-1}^*, y_{n-1}^*, m_{n-1}^*\right)$ where $y_{n-1}^* = y_{n-1} \oplus y_n$ and m_{n-1}^* is the forged MAC. Otherwise, the adversary would have to hope that $\left(y_{n-1} \oplus y_n, x\right)$ was queried to the tag(.) oracle during the querying phase. To conclude, the success probability of Type-II adversary is bounded by $\alpha + \frac{n}{d} 2^{-\frac{1}{2}}$.

Type-III adversary: The success probability of Type-III adversary can also be analysed similarly. In particular, if the adversary can forge a MAC, he can add a tag $\mathcal{T}_{n+1}^*$ to P by

forging two MACs of $\mathcal{T}_n$ and $\mathcal{T}_{n+1}^*$ and $\left(y_n^*, x\right)$ and $\left(y_{n+1}^*, x\right)$, respectively, such that $y_n^* \oplus y_{n+1}^* = y_n$. The forged proof P is

$$\left(\mathcal{T}_1, y_1, m_1, \mathcal{T}_2, y_2, m_2, \cdots, \mathcal{T}_n, y_n^*, m_n^*, \mathcal{T}_{n+1}^*, y_{n+1}^*, m_{n+1}^*\right)$$

which should be correctly verified by the verifier. Therefore, we can obtain the success probability of Type-III adversary as $\frac{1}{2}\alpha + \frac{d-n}{d}2^{-\frac{1}{2}}$.

Combining the success probabilities of three types of the adversary, we complete the proof. □

Theorem 5.2 suggests that if the underlying MAC scheme is secure, i.e., α is negligible, and l is long enough, then the success probability of an adversary attacking our proposed grouping-proof for RFID tags protocol is negligible. We conclude that our scheme is provably secure.

Assuming that we want to produce a co-existence proof of n tags, we compare our proposed scheme with previous protocols in terms of performance and security in Table 2.

Table 2 Comparison of each protocol

Protocol	*Number of relaying messages*	*Cost of generating proof*
Yoking-proof	$n(n-1)$	$2n(n-1)$ MACs
Saitoh and Sakurai (2005)	n	n MACs 1 encryption
Piramuthu (2006)	$n(n-1)$	$2n(n-1)$ MACs
Lin et al. (2007)	$n(n-1)$	$2n(n-1)$ MACs 1 encryption
Burmester et al. (2008)	$n(n-1)$	$4n(n-1)$ $f(.)$ evaluations
Proposed protocol	0	n MACs

6 Conclusions

In this paper, we have attempted to give a glimpse of current research in RFID security. We did so by reviewing some representative works including standards, protocols and security models. We pointed out some of issues that we think deserve more attention and effort. We tackled ourselves one of the said issues. In particular, we presented a grouping-proof protocol that does not suffer from scalability issue of previous protocols. We also presented the first sound security model for a secure grouping-proof protocol. Our security model properly addressed the impact of mafia fraud attack and the level of trust on RFID readers when defining a security notion for a secure grouping-proof protocol.

In our future work, we will extend further our work on grouping-proof protocols by rigorously analysing security of previous grouping-proof protocols. In addition, we hope to tackle other problems raised in this paper.

Acknowledgements

The authors would like to thank the guest editor of this journal, Prof. Chan Yeob Yeun. This research was supported by the ICT Standardization program of the Ministry of Knowledge Economy (MKE).

References

Avoine, G. and Oechslin, P. (2005) 'A scalable and provably secure hash-based RFID protocol', in *The Proceedings of Workshop on Pervasive Computing and Communications Security – PerSec'05*, pp.110–114.

Brands, S. and Chaum, D. (1994) 'Distance-bounding protocols', in *Advances in Cryptology EUROCRYPT'93*, Springer-Verlag, LNCS 765, pp.344–359.

Bringer, J., Chabanne, H. and Dottax, E. (2006) 'HB++: a lightweight authentication protocol secure against some attacks', in *The Proceedings of IEEE Conference on Pervasive Services, Workshop Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pp.28–33.

Burmester, M., de Medeiros, B. and Motta, R. (2008) 'Provably secure grouping-proofs for RFID tags', in *The 8th Proceedings of International Conference on Smart Card Research and Advanced Applications* (CARDIS 2008), Springer-Verlag, LNCS 5189, pp.176–190.

Burnmester, M., Van Le, T., De Medeiros, B. and Tsudik, G. (2009) 'Universally composable RFID identification and authentication protocols', in *The ACM Transactions on Information and Systems Security*, Vol. 12, No. 4, Article 21.

Canetti, R. (2007) *Obtaining Universally Composable Security: Towards the Bare Bones of Trust*, available at http://eprint.iacr.org/2007/475.

Carluccio, D., Lemke, K. and Paar, C. (2005) 'Electromagnetic side channel analysis of a contactless smart card: first results', in *The Proceedings of Workshop on RFID and Lightweight Crypto (RFIDSec05)*.

Chandran, N., Goyal, V., Moriarty, R. and Ostrovsky, R. (2009) 'Position based cryptography', in *The Proceedings of CRYPTO'09*, Springer-Verlag, LNCS 5677, pp.391–407.

Chien, H-Y. (2006) 'Secure access control schemes for RFID systems with anonymity', in *The Proceedings of 2006 International Workshop on Future Mobile and Ubiquitous Information Technologies (FMUIT'06)*, p.96.

Chien, H-Y. (2007) 'SASI: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity', *IEEE Transactions on Dependable & Secure Computing*, Vol. 4, No. 4, pp.337–340.

Chien, H-Y. and Chen, C-H. (2007) 'Mutual authentication protocol for RFID conforming to EPC class-1 generation-2 standards', *Elsevier's Journal of Computer Standards & Interfaces*, Vol. 29, pp.254–259.

Desmedt, Y. (1988) 'Major security problems with the Unforgeableable (Feige)-Fiat-Shamir proofs of identity and how to overcome them', in *SecureCom'88*, pp.15–17.

Devadas, S., Suh, G.E., Paral, S., Sowell, R., Ziola, T. and Khandelwal, V. (2008) 'Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications', in *The Proceedings of IEEE International Conference on RFID 2008*, pp.58–64.

Duc, D.N. and Kim, K. (2010a) 'On the security of RFID group scanning protocols', *IEICE Transactions on Information and Communication Systems*, Vol. E93-D, No. 3.

Duc, D.N. and Kim, K. (2010b) 'Defending RFID authentication protocols against DoS attacks', to appear in the *Elsevier's Journal of Computer Communications*, available at http://dx.doi.org/10.1016/j.comcom.2010.06.014.

EPCglobal Inc. (2009) *Class 1 Generation 2 UHF Air Interface Protocol Standard*, Version 1.2.0, available at http://www.epcglobalinc.org/standards/uhfc1g2.

Gilbert, H., Robshaw, M. and Silbert, H. (2004) *An Active Attack Against HB+ – A Provably Secure Lightweight Authentication Protocol*, available at http://eprint.iacr.org/2005/237.

Gilbert, H., Robshaw, M. and Seurin, Y. (2008) *HB#: Increasing the Security and Efficiency of HB+*, available at http://eprint.iacr.org/2008/028.

Hancke, G.P. and Kuhn, M.G. (2005) 'An RFID distance bounding protocol', in *The Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM05)*, IEEE Computer Society, pp.67–73.

Hutter, M., Mangard, S. and Feldhofer, M. (2007) 'Power and EM attacks on passive 13.56 MHz RFID devices', in *The Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems* (CHES 2007), Springer-Verlag, LNCS 4727, pp.320–333.

Juels, A. (2004) 'Yoking-proofs for RFID tags', in *The Proceedings of the First International Workshop on Pervasive Computing and Communication Security*, IEEE Press, pp.138–143.

Juels, A. and Weis, S.A. (2005) 'Authenticating pervasive devices with human protocols', in Victor Shoup (Eds.): *The Proceedings of CRYPTO'05*, Springer-Verlag, LNCS 3261, pp.293–308.

Karthikeyan, S. and Nesterenko, M. (2005) 'RFID security without extensive cryptography', in *The Proceedings of SASN'05*, ACM Press, pp.63–67.

Konidala, D.M., Kim, Z. and Kim, K. (2007) 'A simple and cost-effective RFID tag-reader mutual authentication scheme', in *The Proceedings of RFIDSec'07*, pp.141–152.

Koscher, K., Juels, A., Brajkovic, V. and Kohno, T. (2009) 'EPC RFID tag security weaknesses and defenses: passport cards, enhanced drivers licenses, and beyond', in *The Proceedings of ACM CCS'09*, ISBN: 978-1-60558-894-0, pp.33–42.

Le, T.V., Burnmester, M. and de Medeiros, B. (2007) 'Universally composable and forward secure RFID authentication and authenticated key exchange', in *The Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, pp.242–252.

Lin, C-C., Lai, Y-C., J. Tygar, D., Yang, C-K. and Chiang, C-L. (2007) 'Coexistence proof using chain of timestamps for multiple RFID tags', in *The Proceedings of APWeb/WAIM International Workshop*, Springer-Verlag, LNCS 5189, pp.634–643.

Menezes, A., van Oorschot, P.C. and Vanstone, S.A. (2010) *Handbook of Applied Cryptography*, available at http://www.cacr.math.uwaterloo.ca/hac/.

Munilla, J. and Peinado, A. (2007) 'HB-MP: a further step in the HB-family of lightweight authentication protocols', *Elsevier's Journal of Computer Networks*, doi:10.1016/j.comnet.2007.01.011, Volume 51, No. 9, pp.2262–2267.

Ohkubo, M., Suzuki, K. and Kinoshita, S. (2004) 'Efficient hash-chain based RFID privacy protection scheme', in *the Proceedings of International Conference on Ubiquitous Computing, Workshop Privacy*.

Oren, Y. and Shamir, A. (2007) 'Remote password extraction from RFID tags', *IEEE Transactions on Computers*, Vol. 56, No. 9, pp.1292–1296.

Ouafi, K. and Phan, R.C-W. (2008) 'Traceable privacy of recent provably-secure RFID protocols', in *The Proceedings of ACNS 2008*, Springer-Verlag, LNCS 5037, pp.479–489.

Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M. and Ribagorda, A. (2006a) 'LMAP: a real lightweight mutual authentication protocol for low-cost RFID tags', in *The Proceedings of RFIDSec'06*.

Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M. and Ribagorda, A. (2006c) 'M2AP: a minimalist mutual-authentication protocol for low-cost RFID tags', in *The Proceedings of UIC'06*, Springer-Verlag, LNCS 4159, pp.912–923.

Peris-Lopez, P., Hernandez-Castro, J.C., M. Estevez-Tapiador, J.M. and Ribagorda, A. (2006b) 'EMAP: an efficient mutual authentication protocol for low-cost RFID tags', in *The Proceedings of IS'06*, Springer-Verlag, LNCS 4277, pp.352–361.

Piramuthu, S. (2006) 'On existence proofs for multiple RFID tags', in *The Proceedings of ACS/IEEE International Conference on Pervasive Services*, IEEE Computer Society, pp.317–320.

Plos, T. (2008) 'Susceptibility of UHF RFID tags to electromagnetic analysis', in *The Proceedings of CT-RSA 2008*, Springer-Verlag, LNCS 4964, pp.288–300.

Saitoh, J. and Sakurai, K. (2005) 'Grouping-proofs for RFID tags', in *The Proceedings of AINA International Conference*, IEEE Computer Society, pp.621–624.

Vaudenay, S. (2007) 'On privacy models for RFID', in *The Proceedings of ASIACRYPT'2007*, Springer-Verlag, LNCS 4833, pp.68–87.

Weis, S.A., Sarma, S.E., Rivest, R.L. and Engels, D.W. (2004) 'Security and privacy aspects of low-cost radio frequency identification systems', in *the Proceedings of the 1st Security in Pervasive Computing*, Springer-Verlag, LNCS 2802, pp.201–212.