Security of RFID Protocols – A Case Study

Ton van Deursen[1] Saša Radomirović[2]

Université du Luxembourg
Faculté des Sciences, de la Technologie et de la Communication
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg

Abstract

In the context of Dolev-Yao style analysis of security protocols, we investigate the security claims of a proposed strong-security RFID authentication protocol. We exhibit a flaw which has gone unnoticed in RFID protocol literature and present the resulting attacks on authentication, untraceability, and desynchronization resistance. We analyze and discuss the authors' proofs of security. References to other vulnerable protocols are given.

Keywords: Verification, RFID, security protocols.

1 Introduction

Radio frequency identification (RFID) systems aim to identify tags to readers in an open environment where neither visual nor physical contact is needed for communication. Because of their low production costs [33] and small size, RFID tags are expected to replace traditional identification methods such as bar codes. Currently, RFID tags are deployed, for instance, in passports [15], access control cards for public transportation [35], and location tracking systems [20,25].

These examples show that RFID tags, as opposed to bar codes, are not only useful to identify objects, but also to authenticate objects or people. Such applications, however, raise privacy concerns, in particular the worry that people may become traceable by carrying RFID-equipped objects.

The computational limitations of RFID tags impose significant restrictions on the number and type of cryptographic primitives that can be implemented on them. It is then a delicate task to attain strong authentication and untraceability properties with simple cryptographic primitives. For example, in a communication with an RFID reader, a tag's messages need to convey sufficient information for the reader to be able to authenticate the tag, without revealing anything that would allow an

[1] Email: ton.vandeursen@uni.lu

[2] Email: sasa.radomirovic@uni.lu

adversary to identify the tag. Additionally, to allow for large-scale deployment, the RFID reader needs to be able to *efficiently* identify and authenticate the tag using the supplied information. Thus, RFID tags, and therefore the protocols they are running, have to satisfy several contradictory requirements.

Consequently, there is a wide variety of proposals aiming at achieving strongly secure, untraceable, and efficient protocols within today's resource limitations for RFID tags, e.g. [7,8,21,36,34]. The significant number of publications breaking such protocols, for instance [3,4,13,32,6], show that the design of resource-constrained RFID protocols is still not well-understood.

In this paper we report on the security of the RFID authentication protocol proposed in [12], which we will call HMNB after the last names of the authors. This protocol is interesting for several reasons. While there is a trend in recently proposed RFID protocols to use non-standard constructions, such as operators with algebraic properties [4,22,34], custom-made hash functions [7], and even resource-constrained public key cryptography [23], to achieve the stringent RFID protocol requirements, the HMNB protocol uses only standard techniques. Notwithstanding, a simple flaw in the protocol leads to the compromisation of three design goals, authentication, untraceability, and desynchronization resistance. Moreover, as discussed in the concluding section, the resulting attacks on this protocol are different from attacks on RFID protocols that have been described before.

Our paper is structured as follows. In Section 2 we explain our terminology and describe the HMNB protocol. In Section 3 we exhibit an authentication flaw in HMNB. In Section 4 we show an attack on untraceability. In Section 5 we show a desynchronization attack and a related untraceability attack, and in Section 6 we conclude with a comparison of our attacks to typical attacks on these properties and give an outlook on future work.

2 Protocol Description

We begin by explaining our terminology and adversary model and then proceed with the description of the HMNB protocol.

In the following, *reader* refers to the actual RFID reader as well as the database communicating with the reader, since this communication takes place over a secure channel. An *agent* can be a tag or a reader, while a *role* refers to the protocol steps a tag or reader is expected to carry out. A *run* is the execution of a role by an agent.

We assume a standard Dolev-Yao intruder model [9] in which the adversary controls the network. While there might be attacks in this model which are not feasible in a real-world RFID system, this model is by far the most mature intruder model. Furthermore, the feasibility of attacks does not only depend on the intruder model, but also on the circumstances under which a system is used. Therefore, for each of the attacks presented in the following sections, we explain a scenario in which it can be carried out. When describing an explicit attack scenario, we speak of one or more *attackers* carrying out the attack. Thus the single theoretical adversary embodies one or more real attackers.

The HMNB protocol aims to mutually authenticate RFID tag and reader, keep

Table 1
Reader's verification and update procedure in the HMNB protocol

Tag response	Update
$h(ID), nt$	$ID' := ID$; $ID := h(ID, nr)$; $HID := h(ID)$
$h(ID, nt, nr), nt$	$ID' := ID$; $ID := h(ID, nr)$; $HID := h(ID)$
$h(ID', nt, nr), nt$	$ID := h(ID', nr)$; $HID := h(ID)$
other	reject tag

the tag untraceable, and resist a particular form of denial-of-service attacks, known as desynchronization attacks. Furthermore, it has been designed with limited computational requirements on tags in mind employing a hash function as the only cryptographic primitive. Reader efficiency is addressed as described in the following.

The protocol assumes that the reader R and tag T share a secret ID, which is updated at the end of a successful protocol execution. For efficiency reasons, the reader also stores the hash of the ID in HID and the value of ID before the last update in ID'. Additionally, the tag keeps track of whether its last run ended successfully or not. For this purpose, the variable S is used. Thus the protocol is stateful.

The protocol starts with the reader challenging the tag with a nonce nr. The tag then generates a nonce nt. The response of the tag depends on the value of S. In case the previous run ended successfully the value of S is 0 and the tag responds with $(h(ID), nt)$ allowing the reader to look up the tag in constant time. In case it did not end successfully, the value of S is 1 and the tag responds with $(h(ID, nt, nr), nt)$. This case should occur only rarely. In either case, the tag sets S to 1. The reader accepts the tag if the response, aside from the nonce nt, is equal to HID, $h(ID, nt, nr)$, or $h(ID', nt, nr)$ for any stored value of HID, ID or ID'. The reader then updates the information for the tag according to Table 1 and sends $h(ID', nt)$ to the tag. Finally, if the received message matches $h(ID, nt)$, the tag replaces its ID by $h(ID, nr)$ and sets S to 0. The protocol is depicted as a message sequence chart in Figure 1.

The message sequence chart shows the role names, framed, near the top of the chart. Above the role names, the role's secret terms are shown. Actions, such as nonce generation, computation, verification of terms, and assignments are shown in boxes. Messages to be sent and expected to be received are specified above arrows connecting the roles. It is assumed that an agent continues the execution of its run only if it receives a message conforming to the specification.

3 Authentication

When dealing with authentication properties, one needs to first make precise what particular form of authentication is considered. In terms of Lowe's authentication hierarchy [29], we consider *recent aliveness* to be the most appropriate authentication requirement for RFID protocols. Recent aliveness captures the fact that the

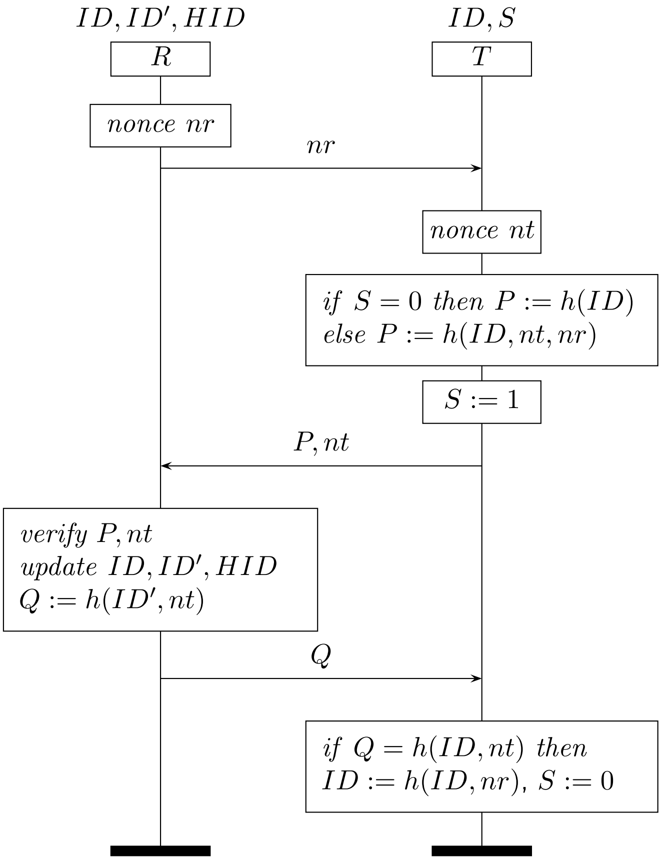

Fig. 1. The HMNB protocol

tag needs to have generated a message as a consequence of a reader's query. More formally, a protocol guarantees to an agent a in role A that any corresponding agent b in role B has been recently alive, if and only if, whenever a completes a run, there has been an event of b during that run.

Note that a weaker authentication property than recent aliveness would need to have a weaker recentness requirement. Thus, one would get a notion of aliveness which merely guarantees that a tag has ever been alive. In that case, a replay attack would not invalidate the security claim. Since for the protocol under consideration the replay attack is explicitly stated to be undesirable, we conclude that the authors intended an authentication notion which is at least as strong as recent aliveness.

3.1 The attack

It is well-known that in order to establish recent aliveness in a challenge-response protocol, the challenge and response must be related [1,11,18,19]. Observe, however, that if no messages are blocked or lost in the HMNB protocol, the tag always responds to a reader's challenge with $h(ID)$ which does not depend on the challenge. While this construction allows for an efficient lookup by the reader, if a tag is in state $S = 0$, the protocol does not provide tag authentication. An adversary can impersonate any tag which is in state $S = 0$ by sending a query to it and replaying the tag's response to a reader before the tag has been queried by an authorized reader. In a scenario where tags are used for access control, an adversary could use a rogue reader to query several tags. Out of a sufficiently large number of tags, it is very likely that there will be several tags which are in state $S = 0$. In fact, the protocol is designed with the assumption that most tags will be in state $S = 0$. If the adversary times the attack properly, for instance by querying tags when people are leaving a restricted area or returning items under access control, it is likely that the adversary will be able to replay the captured messages to the authorized reader before the victims' tags return into the reader's vicinity.

To physically mount this attack the attacker may install a reader in a place where he expects to be able to query a tag he wants to impersonate. The attacker's reader does not have to adhere to official regulations that limit the communication range of RFID systems. Since there need not be physical contact or even a clear line-of-sight between the reader and the tag, the communication is likely to go unnoticed by the holder of the tag. After receiving the tag's response, the attacker can build a tag that sends the captured response when queried by a legitimate reader.

We have found the same type of flaw in several other protocols. The protocols in [10,26,27,28] are also challenge-response-based protocols in which the response does not depend on the challenge. Therefore, these protocols contain a similar authentication flaw that has hitherto gone unnoticed.

3.2 The flaw in the security analysis

In the basic security analysis of HMNB, it is first established that the adversary cannot compute the tag's ID from observed messages. Then it is stated that $h(ID)$ cannot be computed without knowledge of ID. The authors then use the fact that ID is updated at the end of the protocol to claim that the adversary's knowledge of $h(ID)$ is useless for impersonating a tag. The idea behind this reasoning is that the adversary may observe $h(ID)$ during a communication between a tag and a reader, but still cannot use it to impersonate the tag, because tag and reader will have updated the value of ID at the end of their communication.

This reasoning is only valid as long as $h(ID)$ cannot be observed by the adversary in absence of a trusted reader. As the attack in the preceding section shows, the adversary can simply query a tag to obtain $h(ID)$. If no trusted reader is present, then there is no communication between tag and reader, thus the reader will not update ID. Therefore the adversary may use $h(ID)$ to impersonate the tag.

4 Untraceability

The ubiquity and wireless communication capability of RFID tags facilitate and encourage their tracing through space and time. From a privacy perspective this is highly undesirable. Intuitively, a protocol provides *untraceability* if an adversary is not able to recognize a tag he has previously observed. For stateful protocols, such as HMNB, it follows that the adversary should not be able to observe in which state a certain tag is.

Formal definitions of the untraceability notion have been proposed in [3,17,6]. HMNB has been claimed to be untraceable based on the definition in [17], where untraceability is defined in terms of privacy experiments. Such an RFID privacy experiment consists of two phases. In the learning phase, the adversary $\mathcal{A}$ may initiate a communication with the reader $\mathcal{R}$ (ReaderInit) or tags $\mathcal{T}$ (TagInit), after which he may interact with them. The reader and tags respond according to their protocol specification. In the challenge phase, the adversary chooses two tag candidates $\mathcal{T}_i$ and $\mathcal{T}_j$. Then one of these tags is selected at random (referred to as $\mathcal{T}^*$) and $\mathcal{A}$ is given access to this tag. The adversary may again interact with the reader and the tags and must then decide whether the selected tag is $\mathcal{T}_i$ or $\mathcal{T}_j$. If the adversary has a non-negligible advantage in successfully guessing the selected tag, the protocol is not untraceable. To show untraceability using this definition, a standard indistinguishability proof is given.

The authors of HMNB claim that their protocol provides untraceability, because the tag never sends the same response twice. They provide a formal proof for untraceability using the strong privacy definition of [17] explained above. In the following, we first show that the protocol is not untraceable by providing an algorithm that gives the adversary a non-negligible advantage of guessing the selected tag. We then discuss the flaw in the security analysis of HMNB.

4.1 The attack

In the HMNB protocol, the tag's response to an RFID reader's challenge depends on the state the tag is in at the beginning of the protocol. Recall that the tag's state is represented by S. If $S = 0$ the tag responds with $h(ID), nt$ and otherwise the tag responds with $h(ID, nr, nt), nt$. Because the adversary does not know ID, he can not conclude from the tag's response in which state the tag was. The adversary may, however, take advantage of the reader's ability to distinguish between the two states. If the tag was in state $S = 0$ at the beginning of the protocol, the reader cannot verify whether the value of the nonce nt has changed during transmission. Thus, an accidental or malicious modification of nt does not result in a rejection of the tag's response by the reader. The reader completes its run by sending the third message of the protocol. If the tag was in state $S = 1$, the reader uses nt, its own nonce nr, and ID to compute a hash value and compare it with the received one. In this case, a modification of nt can be detected and leads to a rejection of the tag's response and an early termination of the protocol execution by the reader.

In terms of the privacy experiments, the strategy of the adversary is therefore as follows. In the learning phase, two tags $\mathcal{T}_i$ and $\mathcal{T}_j$ are selected and tag $\mathcal{T}_i$ is put into state $S = 1$. The adversary can do this by challenging $\mathcal{T}_i$ and terminating the protocol before sending the third message. The learning phase is formalized by Algorithm 1.

Algorithm 1 *Phase 1: The learning phase*

```
$\mathcal{A}$ chooses a pair of distinct tags $\mathcal{T}_i$ and $\mathcal{T}_j$ uniformly at random.
$\mathcal{A}$ initiates communication with $\mathcal{R}$ using ReaderInit and obtains challenge $nr$.
$\mathcal{A}$ initiates communication with $\mathcal{T}_i$ using TagInit.
$\mathcal{A}$ sends $nr$ to $\mathcal{T}_i$.
```

In the challenge phase, the adversary performs a man-in-the-middle attack. He obtains a challenge from the reader and sends it to the tag to obtain a response. He then replaces the nonce provided by the tag with a different value and submits the response to the reader. If the reader accepts the response, the tag was in state $S = 0$, hence the selected tag is $\mathcal{T}_j$. If the reader rejects the response, the tag was in state $S = 1$, hence the selected tag is $\mathcal{T}_i$. This phase is formalized by Algorithm 2.

```
Algorithm 2 Phase 2: The challenge phase
$\mathcal{A}$ submits $\mathcal{T}_i$ and $\mathcal{T}_j$ as its challenge candidates.
$\mathcal{A}$ initiates communication with $\mathcal{R}$ using ReaderInit and obtains a challenge $nr$.
$\mathcal{A}$ relays $\mathcal{R}$'s challenge $nr$ to the selected tag $\mathcal{T}^*$.
$\mathcal{A}$ modifies $\mathcal{T}^*$'s response $(P, nt)$ to $(P, nr)$ and sends it to $\mathcal{R}$.
If $\mathcal{R}$ accepts the response, $\mathcal{A}$ guesses $\mathcal{T}^* = \mathcal{T}_j$, and otherwise guesses $\mathcal{T}^* = \mathcal{T}_i$.
```

Because this is a man-in-the-middle attack, it is plausible in a scenario where we can assume that the attacker has simultaneous access to a legitimate reader and a tag which is not in the reader's vicinity. It works best if all or most tags are in state $S = 0$, unless modified by an attacker. This implies that either there is only one attacker modifying the tags' states or that all attackers collude. The attacker would put the tags to be traced in state $S = 1$ by querying each of them but not sending the third protocol message. During the reconnaissance stage, when a tag is near the attacker, the man-in-the-middle attack is performed by forwarding messages between reader and tag. If the modification of nt in the second protocol message leads to a rejection by the reader, the attacker recognizes the tag.

4.2 The flaw in the security analysis

To verify that the protocol satisfies the privacy definition stated in [17], all possible interactions with the reader and tag functionalities must be considered.

The authors provide a security proof that the adversary can not determine ID from the first and second message with a non-negligible advantage using the tag functionality. However, the authors' proof does not consider the reader functionality.

As shown above, using the reader functionality initiated with the ReaderInit call the adversary has a non-negligible advantage to guess the selected tag. In fact, the probability of correctly guessing the selected tag is 1.

A related flaw was found by Avoine [3] in the protocol in [14]. Avoine breaks untraceability of a tag by increasing the tag's internal counter to an abnormal level in order to recognize the tag later.

5 Desynchronization

The introduction of security notions such as *untraceability* and *forward untraceability* has led to many new stateful protocols, e.g. [21,24,30,31,34]. Protocols that aim to satisfy these requirements often update their secrets after a successful protocol execution. Obviously, both reader and tag have to carry out the key update in order to ensure that the reader will be able to authenticate the tag in future runs of the protocol.

In a *desynchronization attack* the adversary aims to disrupt the key update leaving the tag and reader in a desynchronized state and rendering future authentication impossible.

The authors of HMNB observe that if the last message from the reader is not received by the tag, the reader carries out an update of the key, but the tag does not. To prevent desynchronization they propose that the reader keeps track of the previous ID of every tag. Thus, during a run of the HMNB protocol, after receiving a tag's response to its challenge and in order to authenticate it, the reader searches through all tags' current ID values, as well as all tags' previous ID values. We show that this is insufficient to prevent desynchronization.

5.1 The attack

Any tag that is in state $S = 0$ can be desynchronized from a reader by a man-in-the-middle attack. In a communication between the reader and a tag, the adversary intercepts and modifies the reader's challenge nr to any value $nr' \neq nr$. The adversary then sends the modified value to the tag and forwards all other messages between reader and tag without modification. Since in the case $S = 0$ the reader does not verify that the tag received the correct value nr, the adversary's modification goes by unnoticed. Thus, at the end of the protocol execution, reader and tag update ID to different values. The reader stores $h(ID, nr)$, while the tag stores $h(ID, nr')$. Therefore, the reader and tag will be in a *desynchronized* state and future authentication of the tag becomes impossible. The attack is depicted in Figure 2. Note that HMNB's safety measures intended to counteract desynchronization do not allow for re-synchronization in this case, because the tag does not store the previous value of ID and the reader does not know the value nr' from which the tag's new ID is computed.

The attack described here is realistic if we assume that the attacker has simultaneous access to a reader and a tag which is not in the reader's vicinity. Alternatively, it suffices that the attacker is able to successfully corrupt the first message in the protocol, for instance by emitting noise near the reader. The attacker could achieve this by carrying a device that broadcasts interfering radio signals [16].

Desynchronizing a tag from the reader also compromises untraceability of the tag. The adversary can obtain a challenge from the reader and use this challenge to obtain a response from a tag. The adversary can then test the response against the reader, which will reject the response if and only if the response came from a desynchronized tag. Therefore, the adversary will always be able to recognize a desynchronized tag.

5.2 The flaw in the security analysis

The authors' analysis of the protocol with respect to desynchronization only considers attacks in which the adversary blocks certain messages. The analysis is split up into two cases, the case where the adversary blocks the second message of the protocol and the case where the adversary blocks the third message. While desynchronization attacks are traditionally performed by blocking messages, there are other possibilities to desynchronize a tag and a reader. In general, the successful

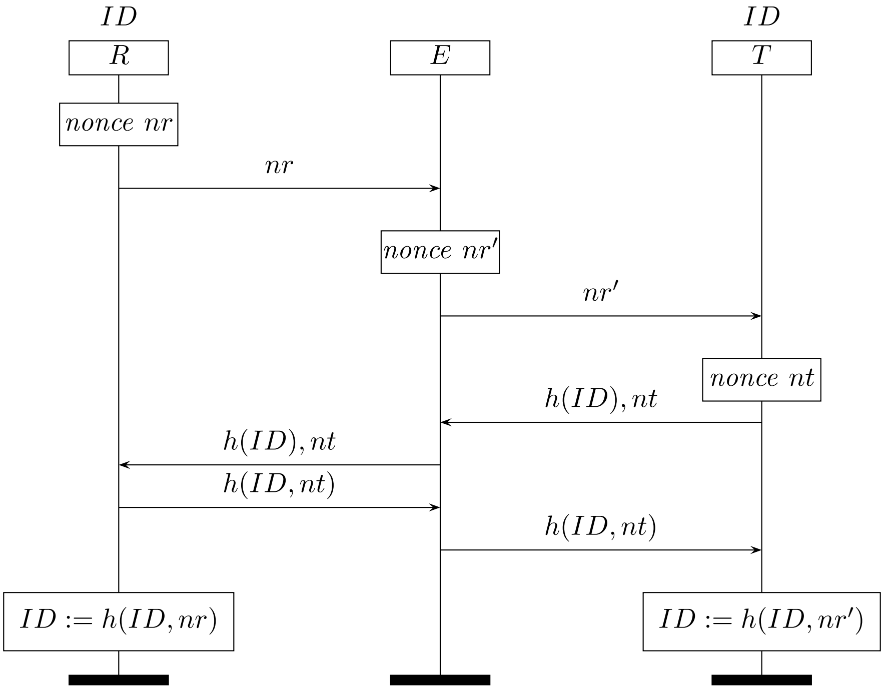

Fig. 2. Desynchronization attack on the HMNB protocol

impersonation of a reader to a tag by an adversary can lead to desynchronization, for instance because the tag updates its keys, ID, or other information to a value that the reader is not able to compute. Another possibility is the selective or random modification of transmitted messages, as has been demonstrated in the attack shown above.

Thus the flaw in the authors' security analysis consists of having considered only a specific attack on the desynchronization property, rather than giving a proof of correctness for the property.

6 Conclusion

In this paper, we have investigated the security claims of an RFID protocol [12], designed to achieve mutual authentication, strong privacy, and desynchronization resistance, while limiting the computational cost for the reader and tags. We have presented a flaw in the protocol which led to attacks on tag authentication, tag untraceability, and desynchronization resistance. We have referred to other RFID protocols suffering from the same authentication flaw, but which to our knowledge have not been noticed to be flawed before.

The attacks presented in this paper are different from attacks on RFID protocols previously described in the literature. For authentication vulnerabilities, attacks typically focus on determining a tag's or reader's secret, or replaying messages observed from previous communications between a tag and the reader. In the present case, we have shown that it is sufficient to simply query a tag in order to impersonate it and thus break authentication. The untraceability vulnerability is non-standard, because state information is used which was leaked by the tag, while traditionally an attacker attempts to fool the tag into reproducing a previously seen message. Finally, while desynchronization attacks are commonly achieved by blocking messages, here it was possible to desynchronize tag and reader through a man-in-the-middle attack and forcing the tag and reader to carry out different updates.

We note that the authentication flaw could have been found using automated verification tools, e.g. [2,5]. Although automated verification of untraceability is still an open problem, the flaws in the untraceability proof of HMNB show that there is a need for automated verification of this notion. Moreover, the introduction of stateful protocols leads to the need of verification of desynchronization resistance. As a first step, desynchronization resistance should be formally defined.

References

[1] Abadi, M. and R. Needham, *Prudent engineering practice for cryptographic protocols*, IEEE Trans. Softw. Eng. **22** (1996), pp. 6–15.

[2] Armando, A., D. A. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò and L. Vigneron, *The avispa tool for the automated validation of internet security protocols and applications*, in: *CAV*, 2005, pp. 281–285.

[3] Avoine, G., *Adversary model for radio frequency identification*, Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland (2005).

[4] Chien, H.-Y. and C.-W. Huang, *A lightweight RFID protocol using substring*, in: *EUC*, 2007, pp. 422–431.

[5] Cremers, C., "Scyther - Semantics and Verification of Security Protocols," Ph.D. dissertation, Eindhoven University of Technology (2006).

[6] Deursen, T. v., S. Mauw and S. Radomirović, *Untraceability of RFID protocols*, in: *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, Lecture Notes in Computer Science **5019** (2008), pp. 1–15.

[7] Di Pietro, R. and R. Molva, *Information confinement, privacy, and security in RFID systems*, in: *ESORICS*, 2007, pp. 187–202.

[8] Dimitriou, T., *A secure and efficient RFID protocol that could make big brother (partially) obsolete*, in: *PerCom*, 2006, pp. 269–275.

[9] Dolev, D. and A. Yao, *On the security of public key protocols*, IEEE Transactions on Information Theory **IT-29** (1983), pp. 198–208.

[10] Duc, D. N., J. Park, H. Lee and K. Kim, *Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning*, in: *Proc. of SCIS 2006*, 2006.

[11] Gong, L., *A variation on the themes of message freshness and replay or, the difficulty in devising formal methods to analyze cryptographic protocols*, in: *CSFW*, 1993, pp. 131–136.

[12] Ha, J., S.-J. Moon, J. M. G. Nieto and C. Boyd, *Low-cost and strong-security RFID authentication protocol*, in: *EUC Workshops*, 2007, pp. 795–807.

[13] Ha, J., S.-J. Moon, J. M. G. Nieto and C. Boyd, *Security analysis and enhancement of one-way hash based low-cost authentication protocol (OHLCAP)*, in: *PAKDD Workshops*, 2007, pp. 574–583.

[14] Henrici, D. and P. Müller, *Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers*, in: *PerCom Workshops*, 2004, pp. 149–153.

[15] Hoepman, J.-H., E. Hubbers, B. Jacobs, M. Oostdijk and R. Wichers Schreur, *Crossing borders: Security and privacy issues of the european e-passport*, in: H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama and S.-i. Kawamura, editors, *Advances in Information and Computer Security, First International Workshop on Security – IWSEC*, Lecture Notes in Computer Science **4266** (2006), pp. 152–167.

[16] Juels, A., R. L. Rivest and M. Szydlo, *The blocker tag: selective blocking of RFID tags for consumer privacy*, in: *ACM Conference on Computer and Communications Security*, 2003, pp. 103–111.

[17] Juels, A. and S. A. Weis, *Defining strong privacy for RFID*, in: *PerCom Workshops*, 2007, pp. 342–347.

[18] Kao, I.-L. and R. Chow, *An efficient and secure authentication protocol using uncertified keys*, Operating Systems Review **29** (1995), pp. 14–21.

[19] Keung, S. and K.-Y. Siu, *Efficient protocols secure against guessing and replay attacks*, in: *ICCCN*, 1995, p. 105.

[20] Kulyukin, V., A. Kutiyanawala, E. LoPresti, J. Matthews and R. Simpson, *iWalker: Toward a rollator-mounted wayfinding system for the elderly*, in: *Proceedings of the 2008 IEEE International Conference on RFID*, 2008, pp. 303–311.

[21] Le, T. v., M. Burmester and B. d. Medeiros, *Forward-secure RFID authentication and key exchange*, Cryptology ePrint Archive, Report 2007/051 (2007).

[22] Lee, S., T. Asano and K. Kim, *RFID mutual authentication scheme based on synchronized secret information*, in: *Symposium on Cryptography and Information Security*, Hiroshima, Japan, 2006.

[23] Lee, Y. K., L. Batina and I. Verbauwhede, *EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol*, in: *Proceedings of the 2008 IEEE International Conference on RFID*, 2008, pp. 97–104.

[24] Li, Y. and X. Ding, *Protecting RFID communications in supply chains*, in: *ASIACCS*, 2007, pp. 234–241.

[25] Li, Z., C.-H. Chu and W. Yao, *SIP-RLTS: An RFID location tracking system based on SIP*, in: *Proceedings of the 2008 IEEE International Conference on RFID*, 2008, pp. 173–182.

[26] Lo, N. W. and K.-H. Yeh, *An efficient mutual authentication scheme for EPCglobal class-1 generation-2 RFID system*, in: *EUC Workshops*, 2007, pp. 43–56.

[27] Lo, N. W. and K.-H. Yeh, *Hash-based mutual authentication protocol for mobile RFID systems with robust reader-side privacy protection, to appear*, 2007.

[28] Lo, N. W. and K.-H. Yeh, *Novel RFID authentication schemes for security enhancement and system efficiency*, in: *Secure Data Management*, 2007, pp. 203–212.

[29] Lowe, G., *A hierarchy of authentication specifications*, in: *CSFW*, 1997, pp. 31–44.

[30] Osaka, K., T. Takagi, K. Yamazaki and O. Takahashi, *An efficient and secure RFID security method with ownership transfer*, in: *CIS*, 2006, pp. 778–787.

[31] Peris-Lopez, P., J. C. H. Castro, J. M. Estévez-Tapiador and A. Ribagorda, *An efficient authentication protocol for RFID systems resistant to active attacks*, in: *EUC Workshops*, 2007, pp. 781–794.

[32] Peris-Lopez, P., J. C. Hernandez-Castro, J. Estevez-Tapiador and A. Ribagorda, *Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard.* (2007).

[33] Sarma, S. E., D. Brock and D. W. Engels, *Radio frequency identification and the electronic product code*, IEEE Micro **21** (2001), pp. 50–54.

[34] Song, B. and C. J. Mitchell, *RFID authentication protocol for low-cost tags*, in: *WISEC*, 2008, pp. 140–147.

[35] Transport for London, *Oyster card*, http://www.oystercard.co.uk (last accessed: May 19, 2008).

[36] Tsudik, G., *A family of dunces: Trivial RFID identification and authentication protocols*, in: *Privacy Enhancing Technologies*, 2007, pp. 45–61.