On a New Formal Proof Model for RFID Location Privacy

(Extended Version[*])

Ton van Deursen[**] and Saša Radomirović

University of Luxembourg
Faculty of Science, Technology and Communication,
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg,

Abstract. We discuss a recently proposed formal proof model for RFID location privacy. We show that protocols which intuitively and in several other models are considered *not* to be location private, are provably location private in this model. Conversely, we also show that protocols which obviously *are* location private, are not considered location private in this model.

Specifically, we prove a protocol in which every tag transmits the same constant message to not be location private in the proposed model. Then we prove a protocol in which a tag's identity is transmitted in clear text to be weakly location private in the model. Finally, we consider a protocol with known weaknesses with respect to location privacy and show it to be location private in the model.

Keywords: Location privacy, untraceability, RFID protocols, formal proof models.

1 Introduction

The ubiquity of radio frequency identification (RFID) systems has given rise to concerns about the privacy of RFID tag bearers. These privacy concerns are addressed by requiring that unauthorized entities must not be able to trace the movements of a tag or its bearer. This means that it should be impossible for any attacker to recognize any tag that he has previously observed or interacted with. This security property is typically referred to as *untraceability* [2, 3], *(strong) privacy* [4, 5], or *location privacy* [6].

To verify whether a communication protocol satisfies a security property, such as location privacy, one creates a model which specifies what powers an adversary is given, how the adversary interacts with his environment, and what the definition of the security property within the model is. Clearly, proving a

[*] An abridged version appears in [1].

[**] Ton van Deursen was supported by a grant from the Fonds National de la Recherche (Luxembourg).

protocol correct in such a model should guarantee that a real-world attacker with equal powers is not able to invalidate the modeled security property.

In this work, we give a brief overview of five existing formal proof models for location privacy. We compare the strengths of four of these models. We then take a closer look at the fifth and most recent of these models and identify significant deficiencies in it. We give reasons why the strongest of the four previous models does not suffer from these deficiencies.

2 Formal Proof Models for Location Privacy

The first formal proof model for location privacy in the RFID setting is due to Avoine [2]. In Avoine's model the adversary is a probabilistic polynomially-bounded Turing machine (PPTM), which interacts with RFID tags and readers through oracles. There is an oracle for communication with the reader, an oracle for communication with tags, and an oracle that gives the adversary access only to the messages sent from a reader to the tag. Finally, there is an oracle modeling the physical compromisation of a tag by giving the attacker access to the internal state of the tag. After it is queried, the adversary is not allowed to further query the other oracles. The strength of the adversary is modeled by selecting a subset of the available oracles. Untraceability is defined through an experiment in which the adversary is first given access to a target tag. Then the adversary is given access to two tags, of which one is the target tag. The adversary wins if he correctly guesses which of the two tags is the target tag. The protocol is said to be untraceable if the adversary has no non-negligible advantage of correctly guessing the target tag compared to random guesses. Avoine separates untraceability into *existential* and *universal* untraceability. An existentially untraceable protocol allows the adversary to trace a tag for a restricted period of time, while a universally untraceable protocol does not.

Juels and Weis [5] extend Avoine's model by providing a slightly stronger definition of untraceability. In their proposal, tags and readers are probabilistic interactive Turing machines modeled as ideal functionalities resembling the equally named interactive Turing machines in Canetti's universal composability paradigm [7]. The tag and reader functionalities each have several interfaces which can be addressed by sending a particular message to the functionality. The adversary has access to these interfaces and controls the channel between all the functionalities. Untraceability (called RFID privacy in [5]) is defined through a privacy experiment in which the adversary may interact with all tag functionalities and may compromise all but two tag functionalities. Two of the uncompromised tag functionalities are selected by the adversary. One of them, say T, is chosen at random and the adversary's advantage in guessing which functionality was chosen decides whether the protocol satisfies the privacy property. In order to make a guess about T, the adversary is permitted to interact with T. Additionally, the adversary is permitted to interact with and compromise all but the two selected tag functionalities in the system's environment. It is due to this last fact that the Juels–Weis adversary is stronger than Avoine's adversary.

Vaudenay [4] proposes a more flexible, hierarchical model for location privacy. His model captures eight classes of adversary capabilities ranging over four different types of tag corruption and two modes of observation. An adversary is a PPTM whose strength is defined by the set of oracles it is allowed to query. A *weak* adversary is never allowed to corrupt a tag, that is, he may never query the *corrupt* oracle. A *forward* private adversary may corrupt a tag at the end of the attack, a *destructive* adversary may corrupt a tag at any time, which leads to the destruction of the tag, that is, the adversary may no longer interact with the tag. A *strong* adversary may corrupt a tag at any time without destroying it. Corresponding to the two modes of observation, an adversary is called *wide* if he may observe whether the protocol ended successfully, and *narrow* else. Since the four types of corruption are orthogonal to the narrow/wide separation, eight different adversarial classes are considered. Privacy is defined by comparing the adversary to a special adversary which makes no use of protocol messages, as follows. An adversary is called *blinded* if he is not allowed to communicate with tags and reader. An adversary is *trivial* if there exists a blinded adversary which essentially performs equally well at guessing a tag's identity. A protocol is P*-private*, where P is one of the eight adversary classes, if all adversaries that belong to that class are trivial. Since it may corrupt tags, the Juels-Weis adversary is stronger than the wide-weak adversary of Vaudenay. The Juels-Weis adversary is weaker than the wide-strong adversary, since the wide-strong adversary may even corrupt the target tag.

Van Deursen et al. [3] define untraceability in a symbolic formal model. They consider a standard Dolev–Yao adversary [8] who may eavesdrop on any message exchanged between tag and reader, modify or block any message sent from tag to reader or vice versa, and may inject his own messages making them look like they were sent by tag or reader. Untraceability is defined as a property on all traces (behaviors) of the protocol. It requires that for any trace in which the tag role is executed twice by the same tag, there must be an indistinguishable trace in which the tag role is executed by two different tags. A Dolev–Yao adversary corresponds to a narrow adversary which cannot corrupt tags. Thus it is at most as strong as Vaudenay's narrow-weak adversary.

For future reference, we note that Vaudenay's wide-strong adversary is the strongest of all the adversaries considered above.

3 The Proof Model of Ha et al.

In this section we briefly outline the proof model of Ha, Moon, Zhou, and Ha. The reader is referred to the original paper [6] for full details.

The model defines two attack games: one for indistinguishability and one for forward secrecy. We will restrict our analysis to the authors' definition of *weak location privacy*, allowing us to only focus on the indistinguishability game.

However, similar results can be obtained when considering the authors' notion of strong location privacy.

The indistinguishability attack game consists of three phases. In the initialization phase, tags are created and the RFID system's database is populated. In the learning phase, the adversary may, depending on his capabilities, query a set of oracles allowing him to interact with tags and database. In the challenge phase, the adversary chooses a target tag T and may again query a set of oracles. Additionally, he may query the *reveal*-oracle that reveals the contents of the tag, for every tag except T. At the end of the challenge phase, the adversary calls the *test*-oracle. The test oracle tosses a fair coin b:

- If $b = 1$: the message that T would send after being queried is given to the adversary.
- If $b = 0$: a random value of the same bit length as T's messages is given to the adversary.

It is then the adversary's task to guess the value of b. The adversary wins the game if his guess is correct.

Note that the test oracle does not provide a full transcript of the protocol execution between the reader and the tag, but only a random message or the message the tag would send to the reader.

The protocol is defined to be *weakly location private* if the adversary does not have a non-negligible advantage of winning the indistinguishability game.

4 Analysis of the Proof Model

In the following, we will represent protocols graphically using message sequence charts, such as in Figure 1. Every message sequence chart shows a reader role R and a tag role T with 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, and assignments are shown in boxes. Messages to be sent and expected to be received are specified above arrows connecting the roles.

4.1 Constant-response protocol

Our first example is a protocol that is intuitively and by the notions of [2–5] untraceable (strongly private, or location private), but which can be proven not to be location private in the proposed model.

The protocol description is as follows. The reader R and tag T share a secret ID. The term c is a public, system-wide constant. The protocol starts by R querying T for a response. The tag T responds with the constant c, after which R sends c back to the tag. We emphasize that every tag responds with the same constant c. Figure 1 depicts the protocol. For simplicity, we omit the communication between reader and database, since it is assumed to be secure.

The protocol is intuitively location private since every tag responds with the same message. In fact, the tags in this protocol could even be identically built

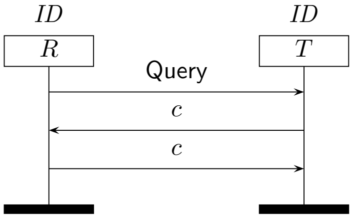

Fig. 1. Protocol 1.

and do not need to have an *ID*. Thus, regardless of his behavior, it is not possible for the adversary to recognize a tag he previously observed.

It is easy to give a proof for location privacy in any of the proof models [2–5]. We take the strongest adversary considered in Section 2, the wide-strong adversary of Vaudenay [4]. It suffices to see that any of the reader and tag oracles in Vaudenay's model can be simulated by letting them output the system-wide constant c. Therefore, no adversary can have a non-negligible advantage of winning the privacy game over a blinded adversary.

We now use the proposed model to prove that the protocol is not location private.

Lemma 1. *Protocol 1 does not satisfy indistinguishability for an active adversary.*

Proof. The adversary's strategy is as follows. He does not query any oracle during the learning phase. In the challenge phase, he selects one of the tags at random, and he only queries the *Test*-oracle, in order to obtain an answer x. The adversary guesses $b = 1$ if $x = c$, and $b = 0$ otherwise.

The adversary wins this game with probability $1 - 2^{-k-1}$, where k is the bit length of the constant. He thus has a non-negligible advantage to win the game. Therefore, Protocol 1 does not satisfy location privacy in the sense of [6].

This example shows a weakness in the indistinguishability game of [6]. At the end of the challenge phase, the adversary must be able to distinguish a tag's response from a random value. The set of possible tag answers is not considered in the game. Intuitively, what matters for location privacy is that the adversary must not be able to distinguish one tag's response from other tags' responses, but not necessarily that the adversary cannot distinguish the tag's response from any arbitrary value.

The flaw in the model extends to all protocols in which the tag responds with a cryptographic hash or encryption that is not indistinguishable from a random bit string of equal length. Since in general, pseudorandom output has not been an explicit requirement for hash functions, it is not safe to assume that the outputs *are* indistinguishable from random bitstrings. For instance, it is obvious that the output of hash functions whose range are points on an elliptic curve [9] can be distinguished from random bit strings. Furthermore, there are proposals for "light-weight" RFID protocols emulating public-key encryption, such as [10–12] (broken by [13–15], improved by [15]) where messages communicated between reader and tag are constructed from points on an elliptic curve. This class of protocols can also be expected to become more numerous in the near future. The proof model discussed in here, however, would not be adequate to prove location privacy for these protocols, since their messages can be distinguished from random bit strings.

4.2 Plaintext ID protocol

Our second example concerns a protocol that is intuitively and by the notions of [2–5] not location private, but can be proven location private with respect to the proposed model.

We assume that a legitimate reader R knows the IDs of all tags in the system. Aside from the reader, nobody except for a tag itself knows the tag's ID. Let $pk(R)$ be a reader's public key with corresponding private key $sk(R)$ and let $\{m\}_{pk(R)}$ denote an IND-CCA public-key encryption of m with the public key $pk(R)$. We further assume that the encryption scheme has the *ciphertext pseudo-randomness* property, such as the scheme proposed by Möller [16] which makes ciphertexts indistinguishable from pseudorandom strings of equal length. Figure 2 depicts the protocol.

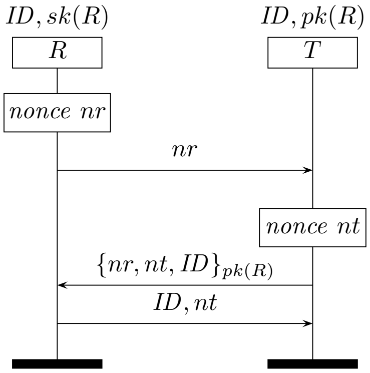

Fig. 2. Protocol 2.

It is easy to see that the protocol is not location private, since the identity ID of the tag is transmitted in the clear in every execution of the protocol. Thus even a passive adversary, namely one which merely observes messages, can trace tags. Since all the adversaries considered in [2–5] are at least as strong as a passive adversary, this protocol is not location private in any of the corresponding models.

To be more precise, we construct a *non-trivial* weak-narrow adversary in Vaudenay's model [4] as follows. The adversary selects two challenge tags and observes one protocol execution of each. When being challenged with one of the tags the adversary again observes a protocol execution of the reader and the challenge tag. It can then guess with which tag it is challenged with probability 1. A blinded adversary has no access to protocol messages and can therefore only make a correct guess with probability $\frac{1}{2}$. Thus even this weak-narrow adversary of Vaudenay is able to trace a tag.

We now use the proposed model to prove that the protocol *is* weakly location private.

Lemma 2. *Protocol 2 satisfies indistinguishability for a passive adversary.*

Proof. In the learning phase, the adversary may query the *execute*-oracle to build a list of tuples $(nr, \{nr, nt, ID\}_{pk(R)}, ID, nt)$, corresponding to observed communications.

In the challenge phase, the adversary selects a challenge tag T and queries the *reveal*-oracle for all tags except T. He further extends his list of tuples $(nr, \{nr, nt, ID\}_{pk(R)}, ID, nt)$ by querying the *execute*-oracle.

Finally, the adversary queries the *Test*-oracle on T. The oracle tosses a fair coin b and

- for $b = 1$ it outputs the message $\{nr, nt, ID\}_{pk(R)}$,
- for $b = 0$ it outputs a random value of the same length as the second protocol message.

The adversary has to guess the bit b. If the adversary were able to guess this bit with a non-negligible advantage, then he could distinguish $\{nr, nt, ID\}_{pk(R)}$ from a random value with a non-negligible advantage. But this would contradict the ciphertext pseudo-randomness assumption on the encryption scheme.

This example shows a weakness in the challenge phase of the indistinguishability game. Before calling the *test* oracle, the adversary has full access to all messages sent by the tag and reader. But once he calls the *test* oracle, his capabilities are limited, in that the messages sent from the reader to the tag are not given to him and he must make a decision based on a single message of the tag. Thus he is not allowed to use information that in standard models would be available to a passive adversary.

Note that if the reader would, in the third message, additionally transmit sufficient information for the adversary to be able to *verify* whether the encryption in message two is indeed an encryption of ID, then the proof would still go through, while the location privacy property would seem even less plausible.

4.3 Published protocols

Our final example concerns a protocol proposed by Ha et al. [17] with known location privacy weaknesses as shown by Van Deursen and Radomirović [18]. The protocol can be shown to be location private in the proposed model.

The protocol aims to mutually authenticate RFID tag and reader, keep the tag untraceable, and resist a particular form of denial-of-service attacks, known as desynchronization attacks.

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 protocol run ended successfully or not. For this purpose, the variable S is used.

In case the tag's previous run ended successfully, the value of S is 0 and the tag responds to a reader's query nr 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.

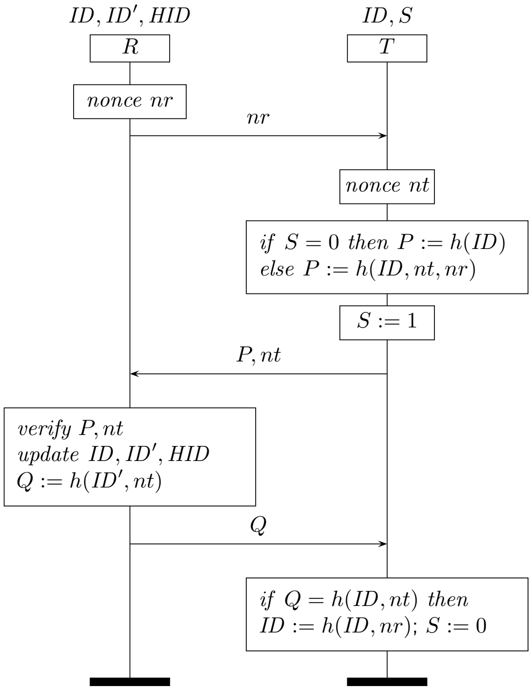

Fig. 3. Protocol 3.

Table 1. Reader's verification and update procedure in protocol 3.

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

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 3.

One flaw of the protocol is that an active attacker can find out whether a tag's state is $S = 0$ or $S = 1$. Combined with the facts that under normal circumstances tags tend to be in state $S = 0$ and that an active adversary can *flag* tags by setting them into state $S = 1$, and thus recognize previously flagged tags. More formally, using Juels and Weis' notion of strong privacy [5] it can be shown that the tags are not strongly private [18]. Important for this attack is that the adversary has access to the reader's third message. If a tag is in state $S = 0$, the reader does not (and cannot) verify the integrity of the nonce nt, while if the tag is in state $S = 1$, the verification of the nonce's integrity occurs implicitly.

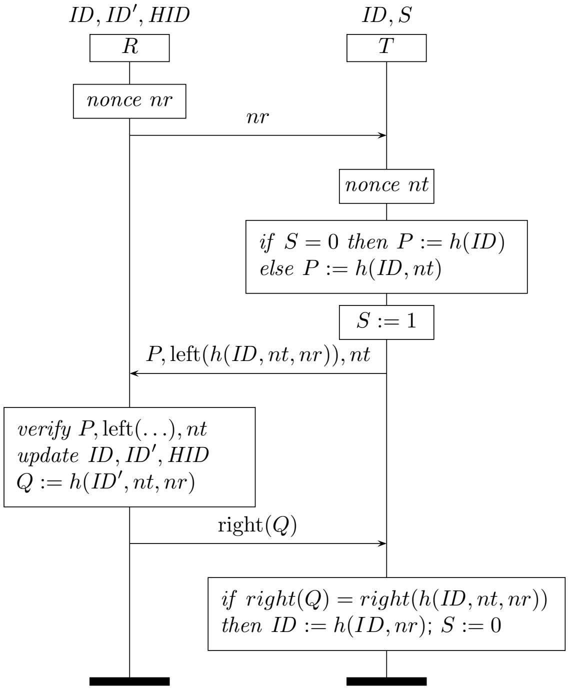
ID, ID', HID
R
ID, S
T
nonce nr
nr
nonce nt
if $S = 0$ *then* $P := h(ID)$
else $P := h(ID, nt)$
$S := 1$
$P, \text{left}(h(ID, nt, nr)), nt$
verify $P, \text{left}(\ldots), nt$
update ID, ID', HID
$Q := h(ID', nt, nr)$
$\text{right}(Q)$
if $right(Q) = right(h(ID, nt, nr))$
then $ID := h(ID, nr)$; $S := 0$

Fig. 4. LRMAP protocol.

A second flaw of the protocol is that the reader nonce nr is not included in the tag response if the tag is in state $S = 0$. Therefore, an active attacker can modify nr without the reader being able to notice. As a result, at the time tag and reader update the ID, they do not agree on the value of nr. This attack will result in desynchronization since nr is used in the update. The attacker can subsequently trace the tag [18].

We now argue why protocol 3 can be shown to be location private in the proposed model. As alluded to in Section 4.2, the adversary is not given access to the reader's third message to make a guess in the challenge phase. Thus neither flaw can be abused by the adversary to attack location privacy in the proposed proof model.

Furthermore, the proposed model is used to prove the so-called LRMAP protocol to be location private [6]. The LRMAP protocol is shown in Figure 4, its update procedure is shown in Table 2. This protocol does not suffer from the above-mentioned flaws, because the reader *always* checks the integrity of the nonces nr and nt, due to the inclusion of an extra term $\text{left}(h(ID, nt, nr))$ in the second message. Further differences between the LRMAP protocol and protocol 3 above are minor and irrelevant. Thus the proof of location privacy for LRMAP given in [6] can also be applied to the flawed protocol 3 to prove it location private.

Table 2. Reader's verification and update procedure in the LRMAP protocol.

Tag response	Update
$h(ID), \text{left}(h(ID, nt, nr)), nt$	$ID' := ID$; $ID := h(ID, nr)$; $HID := h(ID)$
$h(ID, nt), \text{left}(h(ID, nt, nr)), nt$	$ID' := ID$; $ID := h(ID, nr)$; $HID := h(ID)$
$h(ID', nt), \text{left}(h(ID', nt, nr)), nt$	$ID := h(ID', nr)$; $HID := h(ID)$
other	reject tag

The observations about protocol 3 above can be generalized to all protocols which fit the three-message pattern considered in the present model and are not location private only due to a desynchronization attack. To see why desynchronizing a tag from the reader compromises location privacy of the tag, consider the following attack. The adversary obtains a challenge from the reader and use this challenge to obtain a response from a tag. The adversary then tests the response against the reader, which will reject the response if and only if the response came from a desynchronized tag. To trace tags in such protocols the attacker needs to evaluate the reader's response. Since in the present model the adversary is not given access to the reader's third message, such attacks cannot be detected.

5 Conclusion

We have shown that the formal proof model for location privacy proposed by Ha et al. [6] does not coincide with the intuitive notion of location privacy. We have highlighted the flaws in the model and shown that they are not present in other models.

Specifically, we have used Ha et al.'s model to prove lack of location privacy of a protocol which should satisfy every notion of location privacy. This was possible due to the following flaw in the model. The adversary can violate location privacy if he can distinguish a tag's response from a random value. For privacy, however, it is important that the adversary cannot distinguish one tag from another tag.

Furthermore, we have used the model to prove location privacy of a protocol which transmits a tag's ID in plain text in each execution. This was possible because in this model the adversary is not given access to the information contained in the last message of a protocol.

Finally, we note that this unintuitive notion of location privacy does not only affect specially crafted protocols. In our third example we have shown a published protocol that can be proven to be location private in this model, but has also been shown to be susceptible to location privacy attacks.

We therefore do not consider the model of Ha et al. to be a useful model for proving location privacy of RFID protocols.

References

1. van Deursen, T., Radomirović, S.: On a new formal proof model for RFID location privacy. Information Processing Letters **110**(2) (2009) 57–61
2. 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 (September 2005)
3. van Deursen, T., Mauw, S., Radomirović, S.: Untraceability of RFID protocols. In: Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks. Volume 5019 of Lecture Notes in Computer Science., Seville, Spain, Springer (2008) 1–15
4. Vaudenay, S.: On privacy models for RFID. In: Advances in Cryptology - ASIACRYPT 2007. Volume 4833 of Lecture Notes in Computer Science., Kuching, Malaysia, Springer-Verlag (December 2007) 68–87
5. Juels, A., Weis, S.: Defining strong privacy for RFID. In: International Conference on Pervasive Computing and Communications – PerCom 2007, New York, USA, IEEE, IEEE Computer Society Press (March 2007) 342–347
6. Ha, J., Moon, S., Zhou, J., Ha, J.: A new formal proof model for RFID location privacy. In: ESORICS. (2008) 267–281
7. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS. (2001) 136–145
8. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory **IT-29**(2) (March 1983) 198–208
9. Icart, T.: How to hash into elliptic curves. In: CRYPTO 2009. Lecture Notes in Computer Science, Springer (2009) 303–316
10. Lee, Y.K., Batina, L., Verbauwhede, I.: Provably secure RFID authentication protocol EC-RAC (ECDLP based randomized access control). (2007)
11. Lee, Y.K., Batina, L., Verbauwhede, I.: EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol. In: Proceedings of the 2008 IEEE International Conference on RFID. (2008) 97–104

12. Lee, Y., Batina, L., Verbauwhede, I.: Untraceable RFID authentication protocols: Revision of EC-RAC. In: IEEE International Conference on RFID – RFID 2009, Orlando, Florida, USA (April 2009) 178–185
13. van Deursen, T., Radomirović, S.: Attacks on RFID protocols (version 1.1). Cryptology ePrint Archive, Report 2008/310 (August 2009) `http://eprint.iacr.org/2008/310`.
14. van Deursen, T., Radomirović, S.: Untraceable RFID protocols are not trivially composable: Attacks on the revision of EC-RAC. Cryptology ePrint Archive, Report 2009/332 (July 2009) `http://eprint.iacr.org/2009/332`.
15. Bringer, J., Chabanne, H., Icart, T.: Cryptanalysis of EC-RAC, a RFID identification protocol. In: CANS. (2008) 149–161
16. Möller, B.: A public-key encryption scheme with pseudo-random ciphertexts. In: ESORICS. (2004) 335–351
17. Ha, J., Moon, S.J., Nieto, J.M.G., Boyd, C.: Low-cost and strong-security RFID authentication protocol. In: Embedded and Ubiquitous Computing (EUC) Workshops. (2007) 795–807
18. van Deursen, T., Radomirović, S.: Security of RFID protocols – A case study. In: Proc. 4th International Workshop on Security and Trust Management (STM'08). Volume 244 of ENTCS., Elsevier (August 2009) 41–52