Traceable Privacy of Recent Provably-Secure RFID Protocols

Khaled Ouafi[1] and Raphael C.-W. Phan[2⋆]

[1] Laboratoire de sécurité et de cryptographie (LASEC)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
khaled.ouafi@epfl.ch
[2] Electronic & Electrical Engineering
Loughborough University
LE11 3TU, United Kingdom
r.phan@lboro.ac.uk

Abstract. One of the main challenges in RFIDs is the design of privacy-preserving authentication protocols. Indeed, such protocols should not only allow legitimate readers to authenticate tags but also protect these latter from privacy-violating attacks, ensuring their anonymity and untraceability: an adversary should not be able to get any information that would reveal the identity of a tag or would be used for tracing it. In this paper, we analyze some recently proposed RFID authentication protocols that came with provable security flavours. Our results are the first known privacy cryptanalysis of the protocols.

Key words: RFID, Privacy, Untraceability, Authentication protocols.

1 Introduction

Radio frequency identification (RFID) tags are being deployed in many consumer, financial and governmental applications, for instance respectively in supply chain [1, 6, 20, 21, 32], in contactless credit cards [13], and in e-passports [15, 5, 14, 17, 22].

In view of the pervasiveness and inconspicuous nature of these tiny RFIDs, privacy for RFID tag users is a major concern that could potentially impede the public's long-term adoption of RFID-enabled applications. To the best of our knowledge, formal treatments of privacy for RFID protocols include the work of Avoine [2], Juels and Weis [16], Le, Burmester and de Medeiros [18]; and Vaudenay [34, 35, 26]. The difference in these models lie basically in the power of the adversary's tag-corruption ability.

⋆ Work done while the author was with the Laboratoire de sécurité et de cryptographie (LASEC), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

We analyze in this paper the privacy (or security when relevant) issues of the following provably-secure RFID authentication protocols: the protocol by Lim and Kwon [19] at ICICS '06, and two protocols at AsiaCCS '07 by Le et al. [18]. The first protocol is a nice unconventional design in the sense that it achieves both forward and backward untraceability in the face of tag corruption, while typical protocols only provide backward untraceability. That paper also defined a provable security model for backward and forward untraceability. The latter two protocols are interesting since they come with provable security in the sense of universal composability [4] which has strong guarantees. In fact, except for subsections 4.2 and 4.3 corresponding to breaks on forward privacy/security and therefore the notion of tag corruption is inevitably assumed by definition, our attacks do not even need the strong requirement of corrupting tags [33, 16, 34, 19, 18, 35].

2 RFID Privacy Models

For completeness and for better clarity, we describe here the general untraceable privacy (UPriv) model [25] that will be the setting in which we use in later sections to demonstrate how to trace tags and thus show that the schemes do not achieve the notion of untraceable privacy.

In fact, the model defined herein can be seen as an alternative definition of the Juels-Weis model [16] in a style more in line with the Bellare et al. [3] models for authenticated key exchange (AKE) protocols, for which RFID protocols can be seen to have close relationship with. With this model as a reference, our emphasis throughout this paper is on the analysis of the privacy (or security) issues of recent RFID protocols.

A protocol party is a $\mathcal{T} \in Tags$ or $\mathcal{R} \in Readers$ interacting in protocol sessions as per the protocol specifications until the end of the session upon which each party outputs Accept if it feels the protocol has been normally executed with the correct parties. Adversary $\mathcal{A}$ controls the communications between all protocol parties (tag and reader) by interacting with them as defined by the protocol, formally captured by $\mathcal{A}$'s ability to issue queries of the following form:

Execute$(\mathcal{R}, \mathcal{T}, i)$ **query.** This models *passive* attacks, where adversary $\mathcal{A}$ gets access to an honest execution of the protocol session i between $\mathcal{R}$ and $\mathcal{T}$ by eavesdropping.

Send(U_1, U_2, i, m) **query.** This query models *active* attacks by allowing the adversary $\mathcal{A}$ to impersonate some reader $U_1 \in Readers$ (resp. tag $U_1 \in Tags$) in some protocol session i and send a message m

of its choice to an instance of some tag $U_2 \in Tags$ (resp. reader $U_2 \in Readers$). This query subsumes the TagInit and ReaderInit queries as well as challenge and response messages in the Juels-Weis model.

$\mathsf{Corrupt}(\mathcal{T}, K)$ **query.** This query allows the adversary $\mathcal{A}$ to learn the stored secret K' of the tag $\mathcal{T} \in Tags$, and which further sets the stored secret to K. It captures the notion of *forward security* or *forward privacy* and the extent of the damage caused by the compromise of the tag's stored secret. This is the equivalent of the SetKey query of the Juels-Weis model.

$\mathsf{Test}_{\mathsf{UPriv}}(U, i)$ **query.** This query is the only query that does not correspond to any of $\mathcal{A}$'s abilities or any real-world event. This query allows to define the indistinguishability-based notion of *untraceable privacy* (UPriv). If the party has accepted and is being asked a Test query, then depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$. Informally, $\mathcal{A}$ succeeds if it can guess the bit b. In order for the notion to be meaningful, a Test session must be *fresh* in the sense of Definition 2.

Definition 1 (Partnership & Session Completion) *We say that a reader instance $\mathcal{R}_j$ and a tag instance $\mathcal{T}_i$ are partners if, and only if, both have output* Accept*$(\mathcal{T}_i)$ and* Accept*$(\mathcal{R}_j)$ respectively, signifying the completion of the protocol session.*

Definition 2 (Freshness) *A party instance is fresh at the end of execution if, and only if,*

1. *it has output* Accept *with or without a partner instance,*
2. *both the instance and its partner instance (if such a partner exists) have not been sent a* Corrupt *query.*

Definition 3 (Untraceable Privacy (UPriv)) UPriv *is defined using the game $\mathcal{G}$ played between a malicious adversary $\mathcal{A}$ and a collection of reader and tag instances. $\mathcal{A}$ runs the game $\mathcal{G}$ whose setting is as follows.*

Phase 1 (Learning): $\mathcal{A}$ is able to send any Execute, Send, and Corrupt queries at will.

Phase 2 (Challenge):

1. At some point during $\mathcal{G}$, $\mathcal{A}$ will choose a fresh session on which to be tested and send a Test query corresponding to the test session. Note that the test session chosen must be fresh in the sense of Definition 2. Depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given a tag $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$.

2. $\mathcal{A}$ continues making any Execute, Send, and Corrupt queries at will, subjected to the restrictions that the definition of freshness described in Definition 2 is not violated.

Phase 3 (Guess): Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b', which is its guess of the value of b.

The success of $\mathcal{A}$ in winning $\mathcal{G}$ and thus breaking the notion of UPriv is quantified in terms of $\mathcal{A}$'s advantage in distinguishing whether $\mathcal{A}$ receives $\mathcal{T}_0$ or $\mathcal{T}_1$, i.e. it correctly guessing b. This is denoted by $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{UPriv}}(k)$ where k is the security parameter. In relation to other models, note that the Le-Burmester-de Medeiros model [18] similarly allows the corruption of tags. For the purpose of our attack descriptions in later subsections 4.2 and 4.3, it suffices to consider their definition of corruption in their model. This will be treated later as required.

The Vaudenay model [34, 35] is stronger than both the Juels-Weis and Le-Burmester-de Medeiros models in terms of the adversary's corruption ability. In more detail, it is stronger than the Juels-Weis model in the sense that it allows corruption even of the two tags used in the challenge phase. It is stronger than the Le-Burmester-de Medeiros model in the sense that it considers all its privacy notions even for corrupted tags, in contrast to the Le-Burmester-de Medeiros model that only considers corruption for its forward privacy notion.

We chose to describe our tracing attacks in later sections with reference to a defined model in order for more uniformity between similar attacks on different RFID protocols, and for better clarity to illustrate how an adversary can circumvent the protocols using precise types of interactions that he exploits, as captured by his oracle queries. This will facilitate the task of a designer when an attempt is made to redesign a protocol which had been attacked.

3 A Backward and Forward Untraceable Protocol

At ICICS '06, Lim and Kwon [19] proposed an RFID protocol that offers untraceable privacy (UPriv) both before and after corruption of a tag. This is indeed a major feat, since other RFID schemes in literature are only able to treat backward untraceability, i.e. a corrupted tag cannot be linked to any past completed sessions.

The initialization phase is as follows:

1. The reader chooses a random secret K_i for each tag $\mathcal{T}_i$, and evaluates $m-1$ evolutions of $K_i^0 = K_i$, i.e. $K_i^j = g(K_i^{j-1})$ for $1 \leq j \leq m-1$,

where g is a pseudorandom function. It then computes $t_i^j = ext_{l_2}(K_i^j)$ for $0 \leq j \leq m-1$, where l_2 is some appropriate bit length, $ext_l(x)$ is an extraction function returning l bits of x.

2. The reader also chooses a random u_i for each tag $\mathcal{T}_i$ and computes a key chain $\{w_i^j\}_{j=0}^{n-1}$ of length n, such that $w_i^n = u_i$ and $w_i^j = h(w_i^{j+1})$ for $0 \leq j \leq n-1$, where h is a pseudorandom function.
3. The tag stores $\langle w_{i,T}, K_i \rangle$ where $w_{i,T} = w_i^0$ and initializes a failure counter $c_i = 0$.
4. The reader creates two tables L_1, L_2 for $\mathcal{T}_i$ in its database, where L_2 is empty and L_1 has entries of the form $\langle s_i, \{t_i^j\}_{j=0}^{m-1}, u_i, n_i, w_{i,T}, w_{i,S} \rangle$ where $n_i = n$ and $w_{i,S} = w_i^1$ thus $w_{i,T} = h(w_{i,S})$.

After initialization, a normal protocol session is illustrated in Fig. 1, where f is a pseudorandom function. For further discussions on this protocol, the interested reader is referred to [19].

Reader $\mathcal{R}$		**Tag** $\mathcal{T}_i$
Database: $\{\ldots, (K_i, \text{tables } L_1, L_2), \ldots\}$		**Secret:** $w_{i,T}, c_i, K_i$
pick r_1	$\xrightarrow{r_1}$	
		calculate $t_i = ext_{l_2}(K_i)$
		pick r_2
	$\xleftarrow{t_i, r_2, \sigma_1}$	compute $\sigma_1 = ext_{l_1}(f(K_i, r_1 \|\| r_2))$.
check $\exists t_i^j : (t_i^j = t_i) \wedge (t_i^j \in \{t_i^k\}_{k=0}^{m-1}) \wedge$ $\langle K_i, \{t_i^k\}_{k=0}^{m-1}, u_i, n_i, w_{i,T}, w_{i,S} \rangle \in (L_1 \cup L_2)$		
calculate $K_i' = g(K_i)^j, \sigma_1' = ext_{l_2}(f(K_i', r_1 \|\| r_2))$ and check that $\sigma_1' = \sigma_1$		
calculate $\sigma_2 = f(K_i', r_2 \|\| r_1) \oplus w_{i,S}$	$\xrightarrow{\sigma_2}$	$w_{i,S} = f(K_i, r_2 \|\| r_1) \oplus \sigma_2$.
for $k = 0 \ldots m-j-1$ calculate: $\hat{t}_i^k = t_i^{j+k+1}$;		check $h(w_{i,S}) = w_{i,T}$.
for $k = m-j \ldots m-1$ calculate:		If yes:
$\hat{K}_i = g(K_i'), \hat{t}_i^k = ext_{l_2}(g(\hat{K}_i)^{k-m+j})$;		$c_i = 0; w_{i,T} = w_{i,S}$;
update $\hat{K}_i, \{t_i^k\}_{k=0}^{m-1}$ in L_2		$K_i = g(K_i \oplus (w_{i,T} \|\| r_1 \|\| r_2))$.
calculate $K_i = g(K_i \oplus (w_{i,S} \|\| r_1 \|\| r_2))$;		else
$t_i^j = ext_{l_2}(g(K_i)^j)$ for $j = 0 \ldots m-1$;		$c_i = c_i + 1$;
$n_i = n_i - 1, w_{i,T} = w_{i,S}, w_{i,S} = h(u_i)^{n_i}$		if $c_i < m$
update $\langle K_i, \{t_i^k\}_{k=0}^{m-1}, n_i, w_{i,T}, w_{i,S} \rangle$ in L_1		update $K_i = g(K_i)$.

Fig. 1. The backward and forward untraceable RFID protocol

Tracing the Tag. For the purpose of understanding our attack, it suffices to review the gist of the Lim-Kwon protocol. The tag updates its stored secret K_i in two possible ways. If the reader is successfully authenticated, it would update as $K_i = g(K_i \oplus (w_{i,T} || r_1 || r_2))$. Else, the tag would

update as $K_i = g(K_i)$, up to m times of unsuccessful authentications, after which the tag stops updating its K_i. This eventual non-updating allows the reader to catch up.

Our attack works nevertheless, as follows, using the basic principle where we intentionally desynchronize the tag from the reader by sending the tag into the future [19].

1. **Learning:** An adversary sends m number of queries r_1^j for $1 \leq j \leq m$ to the tag $\mathcal{T}_0$, and records the tag's response t_j for $1 \leq j \leq m$. Since the adversary is impersonating the reader, thus each time it will not pass the check by the tag, and so each time the tag would update its stored secret as $K_i = g(K_i)$, from which t_i will be derived in the next session.
2. **Challenge:** Query r_1^m to the tag $\mathcal{T}_b \in \{\mathcal{T}_0, \mathcal{T}_1\}$, and obtain its response t^*.
3. **Guess:** Check if $t^* = t_m$. If so, then the adversary knows this was the tag it queried during the learning phase i.e. $\mathcal{T}_b = \mathcal{T}_0$. Else, it knows that $\mathcal{T}_b = \mathcal{T}_1$.

It was remarked in [19] that once a tag is successfully authenticated by a reader, then the tag's stored secret K_i would be freshly randomized so that tracing of any kind is prevented. Yet, our adversary can repeat the above step of the Learning phase by sending m arbitrary queries r_1^j for $1 \leq j \leq m$ to the tag again to desynchronize it and the same tracing attack applies.

In order to solve the DoS problem, the authors included into the design a feature that unfortunately allowed our attack causing the tag to be traceable even without corruption, although the goal for their protocol was much stronger i.e. backward and forward untraceability even with corruption.

Violating the Forward Untraceability. Another goal of the protocol is to achieve forward untraceability, i.e. even if a tag is corrupted thus leaking its stored secret K_i, it should be impossible for the adversary to trace the tag in future sessions. Nevertheless, an attack by the adversary proceeds as follows, using the example application in [19] of a tag embedded in a purchased item: Initially, the seller's reader $\mathcal{R}_1$ has legitimate access to the tag. At the point of purchase, ownership of this access should transfer to the buyer's reader $\mathcal{R}_2$. The attack can be mounted either by the seller's reader or by an outsider adversary having access to Corrupt queries.

1. An outsider adversary issues a Corrupt query to the tag $\mathcal{T}_b$, obtaining its stored secret K_i. Alternatively, the seller's reader $\mathcal{R}_1$ knows the stored secret K_i and $w_{i,T}$.
2. At the point of purchase, the buyer's reader $\mathcal{R}_2$ interacts with the tag in a protocol session, thus updating K_i. During this time, the adversary eavesdrops on the values r_1, r_2 communicated in the session.
3. Right after the interaction between the tag and the buyer's reader $\mathcal{R}_2$, the adversary initiates a protocol session with the tag. Since it knows the previous K_i, and also the latest values of r_1, r_2, it can recompute the latest $K_i = g(K_i \oplus (w_{i,T}||r_1||r_2))$ and thus pass the check by the tag without any problem. It can therefore trace the tag in all future sessions, and other readers including the buyer's can no longer successfully interact with the tag.

This result counters the protocol's claim that its ownership transfer is perfect. In [19], it was argued that the protocol achieves forward untraceability under the assumption that the adversary cannot eavesdrop on all future legitimate interactions involving the tag and the reader; the above attack works without violating that assumption. [19] furthermore gives a provable security model for forward untraceability in its Appendix, yet their protocol was not rigourously proven under that model, but instead its security was supported with brief arguments.

4 O-FRAP and O-FRAKE

At AsiaCCS '07, Le et al. [18] presented a universally composable [4] privacy model for RFID protocols, and proposed O-FRAP and O-FRAKE. These two protocols are shown in figures 2 and 3 respectively, on which F denotes a pseudorandom function.

4.1 Tracing O-FRAP

O-FRAP is formally proven to be a secure untraceable RFID protocol in the Le-Burmester-deMedeiros model where corruption of tags is allowed, in the sense that the only information revealed to an adversary is if a party is a tag or a reader. Yet we show here how its untraceable privacy can be violated by presenting a tracing attack that is valid even in a weaker privacy model were corruption possibilty is not granted to the adversary.

The attack works as follows:

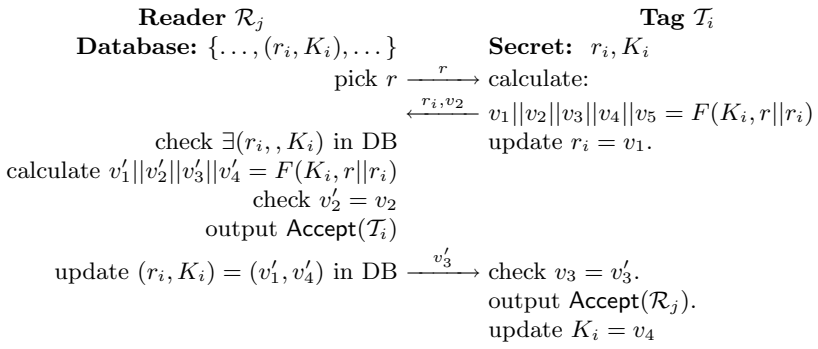

Fig. 2. The O-FRAP protocol

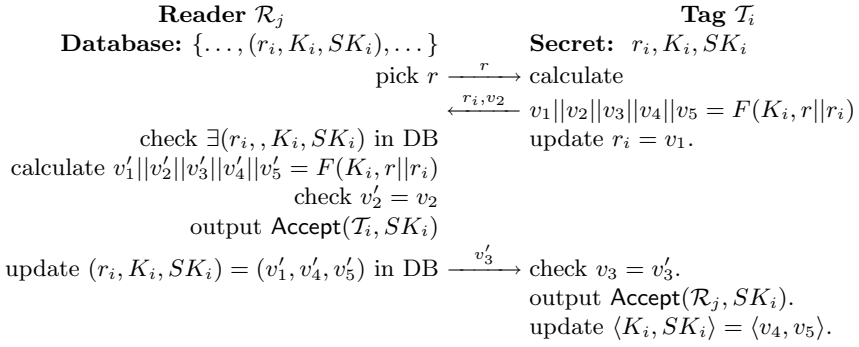

Fig. 3. The O-FRAKE protocol

1. **Learning:** The adversary sends an arbitrary r value to the tag $\mathcal{T}_0$, but does not complete the protocol. This causes the tag to update its r_i, while its K_i remains unchanged, thus marking the tag for future tracing.
2. **Challenge:** To trace the tag in future, the adversary observes the interaction between the reader and the tag $\mathcal{T}_b$.
3. **Guess:** If the reader does not output Accept, then the adversary knows that this tag was indeed the tag that it marked in step (1), i.e. $\mathcal{T}_b = \mathcal{T}_0$. Otherwise, he deduces that $\mathcal{T}_b = \mathcal{T}_1$.

4.2 Violating the Forward Privacy of O-FRAP

In the Le-Burmester-deMedeiros model, corruption is not allowed before a protocol session is initiated, and it is assumed that upon corruption of a party (tag or reader) then the corrupted party's current incomplete session offers no privacy. It is claimed that privacy is maintained for all previously completed sessions involving the heretheto corrupted party.

To motivate our case, we consider the definition of subsession completion in the Le-Burmester-deMedeiros model. A subsession is a party's view of its current protocol session, e.g. during an O-FRAP protocol session, both the reader and the tag have their own separate views of that session, so-called their subsession. To quote from [18], "Upon successful completion of a subsession, each party accepts its corresponding partner as authenticated." Thus, at the point where a party outputs Accept, its subsession is already considered completed.

Referring to the O-FRAP description in Fig. 2, the reader's subsession is completed at the point when it outputs Accept, i.e. before it updates its entry in L and before it sends v_3' to the tag. Meanwhile, the tag's subsession is completed at the point that it outputs Accept, i.e. before it updates its K_i. In the context of the Le-Burmester-deMedeiros model, corruption of a party at this point should not violate the privacy of the party corresponding to its completed subsession. This is the problem with the O-FRAP proof that we are exploiting. Indeed, we show how this can be circumvented.

1. The adversary first eavesdrops on an O-FRAP session and records $\langle r, r_i, v_2 \rangle$.
2. It then corrupts a tag $\mathcal{T}_i'$ at the point after the tag outputs Accept. It thus obtains K_i' corresponding to a previoulsy completed subsession, and not the updated $K_i' = v_4$.

3. The adversary calculates $v_1^*||v_2^*||v_3^*||v_4^* = F(K_i', r||r_i)$. It can then check the computed v_2^* with its recorded v_2 for a match, thereby associating the tag $\mathcal{T}_i'$ to the particular completed subsession corresponding to its recorded $\langle r, r_i, v_2 \rangle$.

Our attack here requires a stronger adversary than the other attacks we have presented in earlier sections of this paper, yet it fits into the Le-Burmester-deMedeiros model for which O-FRAP's privacy was proven, and shows that O-FRAP does not achieve its goal of forward untraceable privacy.

It appears that O-FRAP can be made to resist this attack by having the tag output Accept as the very last step of the protocol, i.e. after K_i has been updated.

4.3 Breaking the Forward Secrecy of O-FRAKE

The above attack can be extended to break the forward secrecy of the O-FRAKE protocol, which is an extension of O-FRAP that furthermore establishes a shared secret session key between the tag and reader.

1. The adversary first eavesdrops on an O-FRAKE session and records $\langle r, r_i, v_2 \rangle$.
2. It then corrupts a tag $\mathcal{T}_i'$ at the point after the tag outputs Accept. It thus obtains $\langle K_i', SK_i' \rangle$ corresponding to a previously completed subsession, and not the updated $\langle K_i', SK_i' \rangle = \langle v_4, v_5 \rangle$.
3. The adversary calculates $v_1^*||v_2^*||v_3^*||v_4^*||v_5^* = F(K_i', r||r_i)$. It can then check the computed v_2^* with its recorded v_2 for a match, thereby associating the tag $\mathcal{T}_i'$ to the particular completed subsession corresponding to its recorded $\langle r, r_i, v_2 \rangle$; and further it also knows that the established session key for that associated session is SK_i'.

Similarly, it appears that O-FRAKE can be made to resist this attack by having the tag output Accept as the very last step of the protocol, i.e. after $\langle K_i, SK_i \rangle$ have been updated.

5 Concluding Remarks

We described in an alternative manner the privacy models that capture the notion of untraceable privacy (UPriv) and briefly discussed its relation to existing models. The aim was to use this notion to show how some recent provably secure RFID protocols (with proofs of security in strong adversarial models) do not achieve this privacy notion even under the

weak adversarial model that does not require corruption of tags. In some sense, these results support the case [7–12, 27, 28] that while provable security is the right approach to design and analysis of protocols, more careful analysis and interpretation of provable security models and proofs are needed to ensure the right definitions [30] are put in place.

As a commonly accepted model addressing privacy and security in RFID has to be established and many RFID protocols are proposed without providing any formal security proof, these results strengthen the need for such a model to facilitate better design of RFID protocols that offer both privacy and security.

"Big Brother is watching you".

George Orwell, *Nineteen Eighty-Four*.

Acknowledgements

We thank the anonymous referees for constructive comments.

References

1. "Albertsons Announces Mandate," RFID Journal, 5 March, 2004. Available online at `http://www.rfidjournal.com/article/articleview/819/1/1/`.
2. G. Avoine, "Adversarial Model for Radio Frequency Identification," Cryptology ePrint Archive, report 2005/049, 20 February, 2005. Available at IACR ePrint Archive, `http://eprint.iacr.org/2005/049`.
3. M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated Key Exchange Secure against Dictionary Attacks," *Advances in Cryptology - EUROCRYPT '00*, LNCS 1807, pp. 139–155, 2000.
4. R. Canetti, "Universally Composable Security: A New Paradigm for Cryptographic Protocols," *Proc. IEEE FOCS '01*, pp. 136–145, 2001. Full version available at IACR ePrint Archive, `http://eprint.iacr.org/2000/067`, last revised 13 December 2005.
5. D. Carluccio, K. Lemke and C. Paar, "E-Passport: The Global Traceability or How to Feel Like a UPS Package," *Proceedings of WISA '06*, LNCS 4298, pp. 391–404, 2007.
6. CASPIAN, "Boycott Benetton," accessed 19 September 2007. Available online at `http://www.boycottbenetton.com`.
7. K.-K.R. Choo, "Refuting Security Proofs for Tripartite Key Exchange with Model Checker in Planning Problem Setting," *Proceedings of IEEE CSFW '06*, pp. 297–308, 2006.
8. K.-K.R. Choo and Y. Hitchcock, "Security Requirements for Key Establishment Proof Models: Revisiting Bellare-Rogaway and Jeong-Katz-Lee Protocols," *Proceedings of ACISP '05*, LNCS 3574, pp. 429–442, 2005.

9. K.-K.R. Choo, C. Boyd and Y. Hitchcock, "On Session Key Construction in Provably-Secure Key Establishment Protocols," *Progress in Cryptology - Mycrypt '05*, LNCS 3715, pp. 116–131, 2005.
10. K.-K.R. Choo, C. Boyd and Y. Hitchcock, "Examining Indistinguishability-based Proof Models for Key Establishment Protocols," *Advances in Cryptology - Asiacrypt '05*, LNCS 3788, pp. 585–604, 2005.
11. K.-K.R. Choo, C. Boyd and Y. Hitchcock, "Errors in Computational Complexity Proofs for Protocols," *Advances in Cryptology - Asiacrypt '05*, LNCS 3788, pp. 624–643, 2005.
12. K.-K.R. Choo, C. Boyd, Y. Hitchcock and G. Maitland, "On Session Identifiers in Provably Secure Protocols," *Proceedings of SCN '04*, LNCS 3352, pp. 351–366, 2005.
13. T.S. Heydt-Benjamin, D.V. Bailey, K. Fu, A. Juels and T. O'Hare, "Vulnerabilities in First-Generation RFID-enabled Credit Cards," *Proceedings of Financial Cryptography '07*, LNCS 4886, pp. 2-14, 2008.
14. J.-H. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk and R.W. Schreur, "Crossing Borders: Security and Privacy Issues of the European e-Passport," *Proceedings of IWSEC '06*, LNCS 4266, pp. 152–167, 2006.
15. A. Juels, D. Molnar and D. Wagner, "Security and Privacy Issues in E-Passports," *Proceedings of SecureComm '05*, pp. 74–88, 2005. Full version available at IACR ePrint Archive, `http://eprint.iacr.org/2005/095`, last revised 18 September 2007.
16. A. Juels and S.A. Weis, "Defining Strong Privacy for RFID," *Proceedings of PerCom '07*, pp. 342–347, 2007. Full version available at IACR ePrint Archive, `http://eprint.iacr.org/2006/137`, 7 April 2006.
17. E. Kosta, M. Meints, M. Hensen and M. Gasson, "An Analysis of Security and Privacy Issues Relating to RFID Enabled ePassports," *Proceedings of IFIP SEC '07*, IFIP 232, pp. 467-472, 2007.
18. T.V. Le, M. Burmester and B. de Medeiros, "Universally Composable and Forward-Secure RFID Authentication and Authenticated Key Exchange," *Proceedings of ASIACCS '07*, pp. 242–252, 2007. Full version titled "Forward-Secure RFID Authentication and Key Exchange" available at IACR ePrint Archive, `http://eprint.iacr.org/2007/051`, 14 February 2007.
19. C.H. Lim and T. Kwon, "Strong and Robust RFID Authentication Enabling Perfect Ownership Transfer," *Proceedings of ICICS '06*, LNCS 4307, pp. 1–20, 2006.
20. "Michelin Embeds RFID Tags in Tires," RFID Journal, 17 January, 2003. Available online at `http://www.rfidjournal.com/article/articleview/269/1/1/`.
21. "Mitsubishi Electric Asia Switches on RFID," RFID Journal, 11 September, 2006. Available online at `http://www.rfidjournal.com/article/articleview/2644/`.
22. J. Monnerat, S. Vaudenay and M. Vuagnoux, "About Machine-Readable Travel Documents: Privacy Enhancement using (Weakly) Non-Transferable Data Authentication," *Proceedings of RFIDSec '07*, pp. 15–28, 2007.
23. M. Naor and M. Yung, "Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks," *Proceedings of STOC '90*, pp. 427–437, 1990.
24. M. Ohkubo, K. Suzuki and S. Kinoshita, "RFID Privacy Issues and Technical Challenges," *Communications of the ACM*, Vol. 48, No. 9, pp. 66–71, 2005.
25. K. Ouafi and R.C.-W. Phan, "Privacy of Recent RFID Authentication Protocols," *Proceedings of ISPEC '08*, LNCS 4991, pp. 263–277, 2008.
26. R.I. Paise and S. Vaudenay, "Mutual Authentication in RFID," *Proceedings of AsiaCCS '08*, to appear.

27. R.C.-W. Phan and B.-M. Goi, "Cryptanalysis of the N-Party Encrypted Diffie-Hellman Key Exchange using Different Passwords," *Proceedings of ACNS '06*, LNCS 3989, pp. 226–238, 2006.
28. R.C.-W. Phan and B.-M. Goi, "Cryptanalysis of Two Provably Secure Cross-Realm C2C-PAKE Protocols," *Proceedings of Indocrypt '06*, LNCS 4329, pp. 104–117, 2006.
29. C. Rackoff and D.R. Simon, "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack," *Advances in Cryptology - CRYPTO '91*, LNCS 576, pp. 434–444, 1991.
30. P. Rogaway, "On the Role Definitions in and Beyond Cryptography," *Proceedings of ASIAN '04*, LNCS 3321, pp. 13–32, 2004.
31. C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology.* , Vol. 4. N. 3. Pp. 161-174. January, 1991.
32. "Target, Wal-Mart Share EPC Data," RFID Journal, 17 October, 2005. Available online at `http://www.rfidjournal.com/article/articleview/642/1/1/`.
33. G. Tsudik, "YA-TRAP: Yet Another Trivial RFID Authentication Protocol," *Proceedings of PerCom '06*, pp. 640–643, 2006.
34. S. Vaudenay, "RFID Privacy based on Public-Key Cryptography," *Proceedings of ICISC '06*, LNCS 4296, pp. 1–6, 2006.
35. S. Vaudenay, "On Privacy Models for RFID," *Advances in Cryptology - Asiacrypt '07*, LNCS 4833, pp. 68-87, 2007.