Untraceable RFID protocols are not trivially composable

van Deursen, Ton; Radomirović, Saša

Published in:
Cryptology ePrint Archive

Publication date:
2009

Untraceable RFID protocols are not trivially composable: Attacks on the revision of EC-RAC

Ton van Deursen* and Saša Radomirović

University of Luxembourg

Abstract. It is well-known that protocols that satisfy a security property when executed in isolation do not necessarily satisfy the same security property when they are executed in an environment containing other protocols.
We demonstrate this fact on a family of recently proposed RFID protocols by Lee, Batina, and Verbauwhede. We invalidate the authentication and untraceability claims made for several of the family's protocols.
We also present man-in-the-middle attacks on untraceability in all of the protocols in the family. Similar attacks can be carried out on some other protocols in the literature, as well.
We briefly indicate how to repair the protocols.

1 Introduction

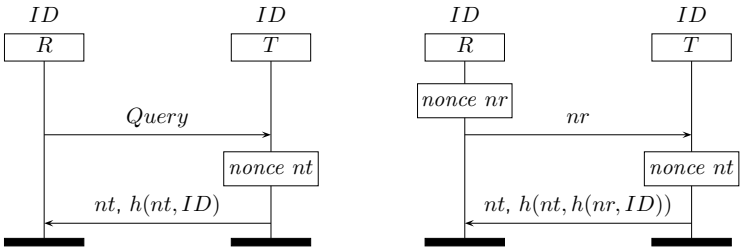
It is well-known [1–5] that protocols satisfying a security property when executed in isolation do not necessarily satisfy the same security property when they are executed in an environment containing other protocols.

In particular, it has been shown that composition of two secrecy-preserving protocols may introduce attacks [6]. Similar results have been obtained for the composition of authentication protocols [7]. It is easy to see that the same holds true for untraceability. Consider for instance the two protocols shown in Figure 1.

Each of the protocols can be shown to be untraceable in isolation. But if an RFID tag implements both protocols, it becomes traceable. An attacker selects protocol A to obtain $nt, h(nt, ID)$ from any tag he is interested in tracing. To test whether a random tag is a tag the attacker is interested in, the attacker selects protocol B and sends the challenge nt to the tag. The tag answers with $nt', h(nt', h(nt, ID'))$. The attacker can then obviously test whether $ID = ID'$.

While the protocols in Figure 1 can be considered as specially crafted protocols, we show that the protocols of Lee, Batina, and Verbauwhede [8] suffer from the same type of problem. These protocols are a revision of EC-RAC [9] which has been shown to be flawed with respect to authentication and untraceability [10–12].

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



(a) Protocol A (b) Protocol B

Fig. 1: Protocols untraceable in isolation, but not in common environment.

2 The proposed family of protocols

The protocols proposed by Lee, Batina, and Verbauwhede are constructed from four components. Proof sketches for the authentication property of the individual components have been given, then the components have been composed leading to the six protocols shown in Figures 2 and 3. We note that no proof of untraceability has been given.

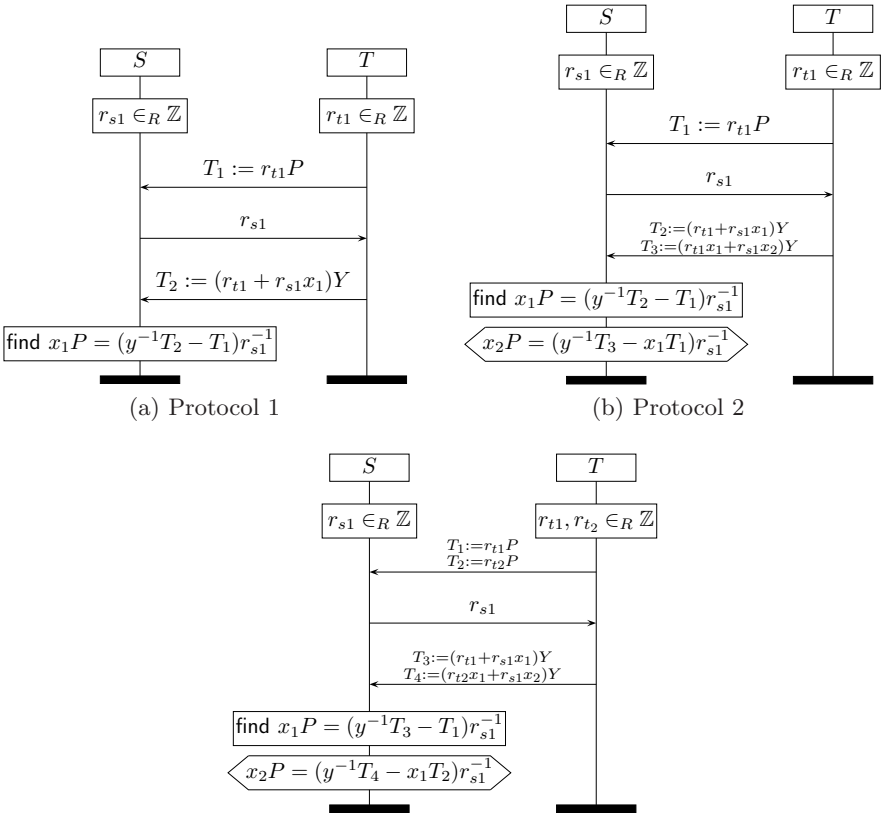
The protocols are based on a fixed, system-wide elliptic curve over a finite field. The points P and $Y = yP$ on the elliptic curve are publicly known, the scalar y and the points x_1P and x_2P are only known to the server, and the scalars x_1, x_2 are unique to each tag and only known to the tag. For scalability reasons, the server additionally knows x_1 in protocols 2, 3, 5, and 6. The elliptic curve is assumed to have been chosen such that the computational Diffie-Hellman problem is hard, that is, given only the points xP, yP, and P on the elliptic curve, it is hard to compute xyP.

All the protocols follow the same commitment-challenge-response structure. More precisely, in all protocols the tag sends a random point on the elliptic curve which serves as a commitment. The server challenges the tag with a random integer upon which the tag answers with a point depending on the commitment and the challenge. The idea of such schemes is that anybody able to produce the correct response can also compute a particular secret, thus successful completion of the protocol constitutes a proof of knowledge for the secret. A moment's thought shows that for the present protocols, the secrets in question are the points x_1Y and x_2Y.

Protocols 4 through 6 additionally include a challenge-response loop where the tag challenges the server and the server proves knowledge of its secret key y.

3 Attacks on the protocols

The main flaw in protocols 4 through 6 is that the challenge-response loop which is supposed to prove the server's authenticity can be abused as an oracle in order to impersonate a tag to a server. This flaw is not surprising, since the same



(a) Protocol 1 (b) Protocol 2

(c) Protocol 3

Fig. 2: Protocols 1 through 3

secret key y is being used for two different purposes. Both the tag-to-server authentication and the server-to-tag authentication depend on it. Thus the two components are not independent of each other (in the sense of [3]) and hence the security of the two components in isolation does not imply their security when they are composed.

To use the first two messages of protocols 4, 5, and 6 as an oracle, the adversary submits any non-zero point on the system's elliptic curve and receives the multiple of the point by the server's secret key y. In the following we will refer to this oracle as the function $P \to O(P) = yP$.

3.1 Untraceability attacks on protocols 4, 5, and 6

Consider the messages $r_{t2}P$, r_{s1}, $(r_{t2} + r_{s1}x_1)Y$ an attacker learns from protocols 4, 5, and 6 by eavesdropping on a communication between a server and a tag. In order to trace the tag, the attacker needs to be able to decide whether

a tag presented to him is the same as the one he eavesdropped on earlier. By eavesdropping on another communication of a tag and server (or by querying a tag himself) the attacker learns $r'_{t2}P$, r'_{s1}, $(r'_{t2} + r'_{s1}x'_1)Y$. He then computes

$$r_{s1}\ r'_{t2}P - r'_{s1}\ r_{t2}P = (r_{s1}r'_{t2} - r'_{s1}r_{t2})P$$

and

$$r_{s1}(r'_{t2} + r'_{s1}x'_1)Y - r'_{s1}(r_{t2} + r_{s1}x_1)Y \tag{1}$$

For $r_{s1}, r'_{s1} \neq 0$, the term in (1) is equal to $(r_{s1}r'_{t2} - r'_{s1}r_{t2})Y$ if and only if $x_1 = x'_1$, that is, if the tag being queried by the attacker is the same tag as the one that was observed earlier. The attacker uses the oracle to decide whether this is the case or not: On submitting $(r_{s1}r'_{t2} - r'_{s1}r_{t2})P$ to the oracle, the attacker receives $(r_{s1}r'_{t2} - r'_{s1}r_{t2})Y$. This equals the term in (1) if and only if the tag has been observed before.

Thus none of the protocols 4, 5, and 6 are untraceable. In particular, protocols 4 and 6 do not satisfy the claimed forward and backward untraceability properties either.

3.2 Authentication attacks on protocols 4, 5, and 6

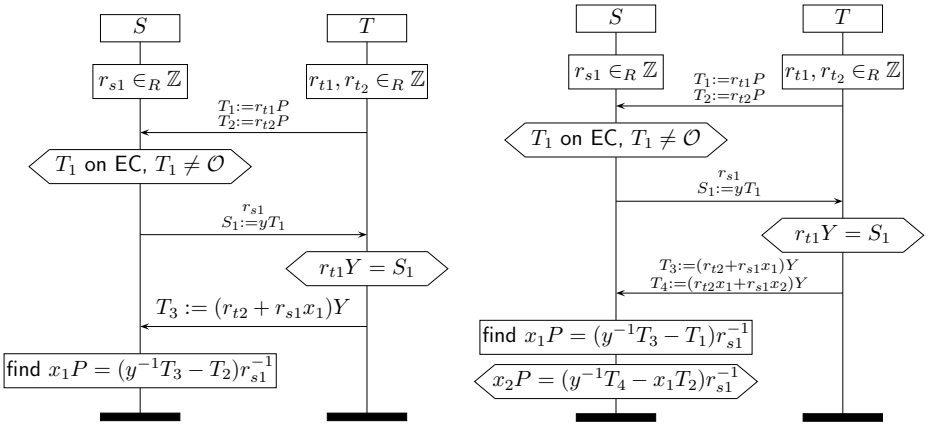
In order to break tag-to-server authentication in these protocols, an adversary needs to know the term x_1Y, and the term x_2Y (in protocols 5 and 6). The adversary learns these two terms from the tag's public keys x_1P, x_2P by computing $O(x_iP_1) = x_iY$, $(i = 1, 2)$. According to the attacker model specified for these protocols, an attacker is initially only allowed to know Y, P, and the order of the system's elliptic curve, but not the tags' public keys. Under this restriction, only a rogue server in the system is able to impersonate tags. Protocol 4, however, is even vulnerable if the adversary does not know the tag's public keys. In this case the adversary can learn x_1Y by eavesdropping on one protocol execution between a tag and a server and performing the following computation.

By eavesdropping on one communication between a tag and a server, an attacker obtains $r_{t2}P$, the challenge r_{s1}, and $(r_{t2} + r_{s1}x_1)Y$. He then computes $r_{s1}^{-1}r_{t2}P$ and $r_{s1}^{-1}(r_{t2} + r_{s1}x_1)Y = (r_{s1}^{-1}r_{t2} + x_1)Y$. Using the oracle, the attacker obtains $O(r_{s1}^{-1}r_{t2}P) = r_{s1}^{-1}r_{t2}Y$ and computes the difference $(r_{s1}^{-1}r_{t2} + x_1)Y - r_{s1}^{-1}r_{t2}Y = x_1Y$. After learning x_1Y and x_2Y by using the oracles as described above, an attacker can impersonate a tag as follows.

Protocol 4. The attacker chooses a random integer r_{t1}, submits $r_{t1}P$ to the server, and is challenged by r_{s1}. To answer this challenge, the attacker computes $r_{s1}x_1Y$, and $r_{t2}Y$ and sends back the sum of these two points.

Protocol 5. The attacker chooses random integers r_{t1}, r_{t2}, submits T_1, T_2 to the server, and is challenged by r_{s1}. To answer this challenge, the attacker computes T_3 from the sum of $r_{s1}x_1Y$, and $r_{t2}Y$. To compute T_4, the attacker multiplies x_1Y by r_{t2} and x_2Y by r_{s1} and computes the sum of these two points.

Protocol 6. The attacker chooses random integers r_{t1}, r_{t2}, r_{t3}, submits T_1, T_2, T_3 to the server, and is challenged by r_{s1}. To answer this challenge, the attacker computes T_4 from the sum of $r_{s1}x_1Y$, and $r_{t2}Y$. To compute T_5, the attacker multiplies x_1Y by r_{t3} and x_2Y by r_{s1} and computes the sum of these two points.



(a) Protocol 4

(b) Protocol 5

(c) Protocol 6

Fig. 3: Protocols 4 through 6

3.3 Untraceability attacks on all protocols

We demonstrate a man-in-the-middle attack on the ID-transfer component that allows a wide-strong adversary of (in the sense of Vaudenay [13]) to trace a tag in all of the six protocols.

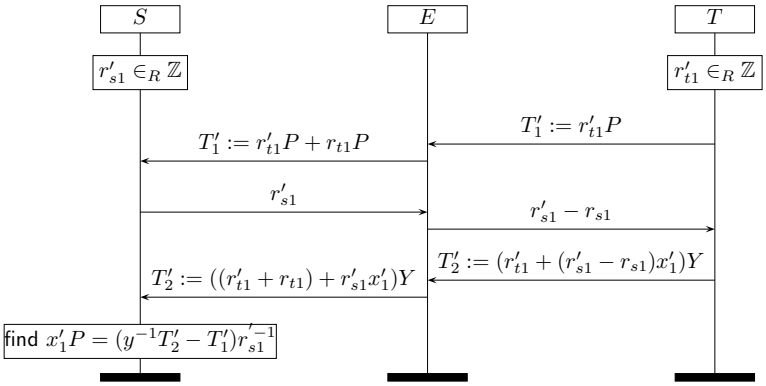


Fig. 4: Man-in-the-middle attack on Protocol 1

By eavesdropping on protocol 1, the adversary obtains the messages $r_{t1}P$, r_{s1}, and $(r_{t1}+r_{s1}x_1)Y$. He then mounts a man-in-the-middle attack on a second communication to test whether the same tag is present as shown in Figure 4. The adversary adds the previously observed $r_{t1}P$ to the commitment $r'_{t1}P$ and subtracts r_{s1} from the new commitment r'_{s1}. The new and old responses are added and sent to the server:

$$\begin{aligned} T'_2 &= (r_{t1} + r_{s1}x_1)Y + (r'_{t1} + (r'_{s1} - r_{s1})x'_1)Y \\ &= ((r_{t1} + r'_{t1}) + r_{s1}x_1 + (r'_{s1} - r_{s1})x'_1)Y \end{aligned}$$

This response is accepted by the server if and only if $x_1 = x'_1$, i.e. if the tag is the same as the one that was previously observed. Therefore, a wide-strong adversary can trace tags. The same attack is possible on protocols 2 through 6 since they are extensions of protocol 1.

Note that a wide-strong adversary can also trace tags in the protocol proposed by Bringer et al. [11] using the same method. This protocol has, however, only been claimed and proven secure against narrow-strong adversaries.

4 Repairing the flaws

The protocol compositions can be improved by assuring that one component in the composition cannot be used as an oracle for another. For protocol 4, this can be achieved without compromising efficiency of the scheme. We equip the server

with a second secret y_2, generated randomly and independently of y, and store the point y_2P in every tag. In the second message, the server sends y_2T_1 instead of yT_1, to prove server authenticity to the tag. A similar approach improves protocols 5 and 6.

To defend against the man-in-the-middle attacks, a stronger form of authentication seems to be unavoidable. In its current form, protocol 1 provides recent aliveness [14]: the server is guaranteed that the tag has recently produced a message. However, agreement [14] is clearly not satisfied as shown by the attack in Figure 4. At the end of the run, the server believes the following messages were exchanged

$$(r'_{t1} + r_{t1})P \quad r'_{s1} \quad ((r'_{t1} + r_{t1}) + r'_{s1}x'_1)Y,$$

while for the tag the transcript reads

$$r'_{t1}P \quad r'_{s1} - r_{s1} \quad (r'_{t1} + (r'_{s1} - r_{s1})x'_1)Y.$$

As shown above, the adversary abuses this discrepancy and the reader's reaction to it to trace tags. To foil the attack, we need to make sure that reader and tag agree on the contents of all messages.

The simplest solution is to use message authentication codes based on a shared secret. The last message would then include a hash of the previous messages (including the payload of the current message) and a secret known only to server and tag. A suitable candidate would be $h(r_{t1}P, r_{s1}, (r_{t1} + r_{s1}x_1)Y, xyP)$. Since $(r_{t1} + r_{s1}x_1)Y$ is uniquely determined by the other three components of the hash it does not have to be included, reducing the message authentication code to $h(r_{t1}P, r_{s1}, xyP)$.

Although this solution prevents these particular man-in-the-middle attacks, it is more resource intensive since it additionally requires a cryptographic hash function to be implemented (and computed) on the tag. We conjecture that protocol 5 can also be made resistant to man-in-the-middle attacks by modifying the computation of T_4 in the third message. Replacing T_4 by $(r_{t2}x_1 - r_{s1}x_2)Y$ prevents the presented man-in-the-middle attack and obvious derivatives thereof. It is unclear, however, whether this introduces new vulnerabilities.

References

1. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS. (2001) 136–145
2. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067 (2000) http://eprint.iacr.org/.
3. Andova, S., Cremers, C., Gjøsteen, K., Mauw, S., Mjølsnes, S., Radomirović, S.: A framework for compositional verification of security protocols. Information and Computation **206** (February-April 2008) 425–459
4. Heintze, N., Tygar, J.D.: A model for secure protocols and their compositions. IEEE Trans. Software Eng. **22**(1) (1996) 16–30
5. Kelsey, J., Schneier, B., Wagner, D.: Protocol interactions and the chosen protocol attack. In: Security Protocols Workshop. (1997) 91–104

6. Cremers, C.: Feasibility of multi-protocol attacks. In: Proc. of The First International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, IEEE Computer Society (April 2006) 287–294
7. Tzeng, W.G., Hu, C.M.: Inter-protocol interleaving attacks on some authentication and key distribution protocols. Inf. Process. Lett. **69**(6) (1999) 297–302
8. 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)
9. 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
10. van Deursen, T., Radomirović, S.: Attacks on RFID protocols (version 1.0). Cryptology ePrint Archive, Report 2008/310 (July 2008) http://eprint.iacr.org/2008/310.
11. Bringer, J., Chabanne, H., Icart, T.: Cryptanalysis of EC-RAC, a RFID identification protocol. In: CANS. (2008) 149–161
12. van Deursen, T., Radomirović, S.: Algebraic attacks on RFID protocols. In: Information Security Theory and Practices. Smart Devices, Pervasive Systems, and Ubiquitous Networks (to appear). Lecture Notes in Computer Science, Brussels, Belgium, Springer (2009)
13. 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
14. Lowe, G.: A hierarchy of authentication specifications. In: CSFW. (1997) 31–44