How To Ensure Forward and Backward Untraceability of RFID Identification Schemes By Using A Robust PRBG

J. Wu* and D.R. Stinson†
David R. Cheriton School of Computer Science
University of Waterloo, Waterloo, ON, Canada
{j32wu,dstinson}@uwaterloo.ca

Abstract

In this paper, we analyze an RFID identification scheme which is designed to provide forward untraceability and backward untraceability. We show that if a standard cryptographic pseudorandom bit generator (PRBG) is used in the scheme, then the scheme may fail to provide forward untraceability and backward untraceability. To achieve the desired untraceability features, the scheme can use a robust PRBG which provides forward security and backward security. We also note that the backward security is stronger than necessary for the backward untraceability of the scheme.

1 Introduction

1.1 RFID Identification Scheme

Radio Frequency Identification (RFID) is an automated object identification technology. An RFID system consists of RFID tags, RFID readers and a back-end server. A tag is a tiny microchip containing identification information. A reader queries a tag, receives responses from the tag via radio signal, and queries the back-end server which maintains a database of tags. The server retrieves and returns to the reader the detailed information of the tag.

RFID technology raises significant privacy issues. For example, since a tag automatically responds to queries via radio signal, sensitive data may be leaked to unauthorized readers. Even when the data is encrypted, the location of the tag may still be traced. There has been considerable research on these privacy issues and numerous privacy-reserving RFID identification schemes have been proposed. For a survey, see [3].

Two privacy features, termed *backward untraceability* and *forward untraceability*, were proposed in [6] and [4] (here we use the terminology from [4]). Backward untraceability means that, if the adversary reveals the internal state of a tag at time τ, the adversary is not able to tell whether a transaction before time τ involves the tag. Forward untraceability means that, if the adversary reveals the internal state of a tag at time τ, the adversary is not be able to tell whether a transaction after time $\tau + \delta$ (for some $\delta > 0$) involves the tag, provided that the adversary does not eavesdrop on the tag continuously after time τ. In [4], Lim and Kwon proposed an RFID identification scheme

*research supported by an NSERC post-graduate scholarship.
†research supported by NSERC discovery grant 203114-06.

to provide forward and backward untraceability (as well as other privacy features). In [7], Song and Mitchell proposed an RFID identification scheme to provide the same features as the Lim-Kwon scheme, but with improved performance in memory space, computation time and communication overhead. Their scheme is designed to withstand this attack. In both the L-K scheme and the S-M scheme, a tag needs to generate random numbers.

1.2 Pseudorandom Bit Generator

Random bit generation is very important in cryptography. However, true random bit generators rely on physically random processes, and hence they are inefficient or expensive in most practical environments. It is therefore more common to use *pseudorandom bit generators* (PRBG) in practice.

A standard (cryptographic) PRBG is a deterministic algorithm which, when given a truly random binary input of length n, outputs a binary sequence of length polynomial in n, say $p(n)$, which "appears" to be random. The input to the PRBG is called the seed, while the output of the PRBG is called a pseudorandom bit sequence. A standard PRBG is *secure*, if when given the first $l < p(n)$ bits of the output of the PRBG, it is infeasible in polynomial time (in n) to predict the next bit of the output ([5], [8]).

A *robust* PRBG provides additional security beyond a standard PRBG. In [2], Barak et al propose a formal model and an architecture for robust PRBG, which satisfy the following properties of *forward security* and *backward security*[1]: (1) backward security: past output of the PRBG looks random to an adversary, even if the adversary learns the internal state at a later time. (2) forward security: future output of the PRBG looks random to an adversary with knowledge of the current state, provided that the PRBG is later refreshed with data of sufficient entropy. Similar properties and constructions can be found in [1] from NIST.

One example of standard PRBG is a block cipher with a secret key working in counter mode, i.e., let $E_k()$ be a block encryption scheme with secret key k, the output is $s_i = E_k(i), i = 1, 2, \cdots$. Such a PRBG is a standard PRBG if the block cipher is secure, but it is not a robust PRBG.

Another example of standard PRBG is a keyed hash function with a secret key in output-feedback mode. Let $h_k()$ be a keyed hash function with secrete key k and an initial input s_0, the output is $s_{i+1} = h_k(s_i), i = 0, 1, \cdots$. Such a PRBG is not a robust PRBG, either.

1.3 Our Work

When a PRBG is used in a security protocol, it is important to know weather a standard cryptography PRBG suffices for the protocol or a robust PRBG is necessary. In some cases, a protocol designer may overlook the fact that a standard PRBG may not provide forward and backward security, which is necessary for certain properties.

In [7], the definition for a PRBG refers to the standard PRBG. It is worthwhile to point out that, in order to achieve forward untraceability and backward untraceability, the S-M scheme in [7] needs to use a PRBG stronger than the standard PRBG. In Section 2, we show that if a standard PRBG is used in the scheme, then the scheme may fail to provide forward untraceability and backward untraceability. We also construct an example of a robust PRBG for the S-M scheme to ensure its desired untraceability features.

[1] In [2], forward security means that past output is secure, while backward security means that future output is secure. Here we switch the two names to be consistent with the terminology we use in this paper.

2 Analysis Of The S-M Scheme

2.1 Scheme Description

We briefly recall the S-M scheme which is described in [7]. $f_k()$ is a keyed hash function with key k. $h()$ is a hash function. $x \gg y$ denotes the operation that right rotate-shifts x by y bits and $x \ll y$ denotes the operation that left rotate-shifts x by y bits. l is the length of the parameters in the scheme.

Initially, for each tag T_i, the server stores its identifier (u_i, t_i). u_i is a unique secret for T_i and $t_i = h(u_i)$. The value t_i is stored in the tag. An identification session takes place as follows.

1. The reader sends a random challenge r_1 to the tag.
2. The tag generates a random r_2, computes $M_1 = t_i \oplus r_2, M_2 = f_{t_i}(r_1 \oplus r_2)$, and sends (M_1, M_2) to the reader.
3. The reader forwards (r_1, M_1, M_2) to the server.
4. The server searches for a tag T_i such that t_i satisfies $M_2 = f_{t_i}(r_1 \oplus r_2)$ where $r_2 = M_1 \oplus t_i$, computes $M_3 = u_i \oplus (r_2 \gg l/2)$, and sends M_3 to the reader. The server also updates u_i and t_i as follows: $u_{i(new)} = (u_i \ll l/4) \oplus (t_i \gg l/4) \oplus r_1 \oplus r_2$, $t_{i(new)} = h(u_{i(new)})$.
5. The reader forwards M_3 to the tag.
6. The tag computes $u_i = M_3 \oplus (r_2 \gg l/2)$. If $h(u_i) = t_i$, then the tag updates t_i as follows: $t_i = h((u_i \ll l/4) \oplus (t_i \gg l/4) \oplus r_1 \oplus r_2)$.

To perform the above protocol, a tag needs to generate random numbers. [7] cites a definition for PRBG which is just a standard PRBG.

2.1.1 Forward Untraceability

We review forward untraceability as defined in [4] and [7]: at time τ, the adversary reveals the internal state of the tag T_i. At time $\tau' > \tau$, the tag performs a transaction with the server and the adversary does not eavesdrop on this transaction. Forward untraceability means that the adversary cannot tell if a transaction at time $\tau'' > \tau'$ involves the tag T_i.

We show that, if the PRBG used by the tag is not a robust PRBG, then the scheme does not achieve forward untraceability. To be concrete, we assume the block cipher based PRBG described in Section 1.2. Suppose at time τ, the adversary reveals the internal state of a tag, including its value t_i and the internal state of its PRBG. At some time $\tau' > \tau$, the tag has a transaction without being eavesdropped by the adversary. After time τ', the adversary observes multiple transaction messages.

Given the internal state (k and i) of the PRBG that the adversary obtained at time τ, the adversary can compute a sequence of n future outputs of the PRBG, where n is an upper bound on the maximum number of times that the tag could have invoked its PRBG since time τ. If any value r in this sequence and any observed message (r_1, M_1, M_2, M_3) after τ' satisfies $f_{M_1 \oplus r}(r_1 \oplus r) = M_2$, then the adversary can deduce that the message involves the tag T_i. In addition to identifying the tag, the adversary can further compute the t_i used in the observed transaction as well as the updated t_i after this transaction: $u_i = M_3 \oplus (r \gg l/2)$, current $t_i = h(u_i)$, and updated

$t_i = h((u_i \ll l/4) \oplus (t_i \gg l/4) \oplus r_1 \oplus r)$. Therefore, the adversary can launch more attacks, e.g., to impersonate the tag or clone the tag.

2.1.2 Backward Untraceability

We review backward untraceability as defined in [4] and [7]: at time τ, the adversary reveals the internal state of the tag T_i. Backward untraceability means that the adversary cannot tell if a transaction at time $\tau' < \tau$ involves the tag T_i.

Given the internal state (k and i) of the PRBG that the adversary obtained at time τ, the adversary can compute a sequence of i previous outputs of the PRBG. The adversary has also observed multiple transactions before time τ. If any r value in the computed output sequence and any observed message $(r_1, (M_1, M_2), M_3)$ satisfies $f_{M_1 \oplus r}(r_1 \oplus r) = M_2$, then the adversary can deduce that the message involves the tag T_i.

2.2 Using Robust PRBG In The S-M Scheme

It is clear that, for the S-M scheme to achieve forward untraceability and backward untraceability, the PRBG used in the tags needs to be forward secure and backward secure. We can follow the construction in [2] or [1] to build a robust PRBG for the tags. An example of a robust PRBG is based on a block cipher working in counter mode. Let k be the secret key. Each time the PRBG is invoked, $s = E_k(i)$ is outputted as the random bits, the key is updated as $k = E_k(i+1)$, and the counter i increases. The PRBG also refreshes its key as $k = k \oplus r_1$ each time a random challenge r_1 is received from the reader.

Now we briefly analyze the case when k and i are revealed at time τ. First we consider backward traceability. If E is a secure block cipher, then it is infeasible to find the previous keys it used; without the previous keys, it is infeasible to distinguish its previous output from a random number. Therefore backward untraceability of the schemes is preserved.

Next we consider forward traceability. Suppose in a certain transaction after τ, the adversary does not observe the messages, including r_1. Then the adversary does not know the updated key and thus he cannot predict the future outputs of E. Therefore, forward untraceability of the schemes is preserved.

Remark. We note that the backward security of the robust PRBG is sufficient but not necessary for the backward untraceability of the S-M scheme. Suppose the PRBG based on keyed hash function given in Section 1.2 is used in the S-M scheme. Given s_i and k at time τ, the adversary can tell if a given value x was generated by the PRBG by computing $x_1 = h_k(x), x_2 = h_k(x_2), \cdots$ and checking if s_i appears in the sequence. Therefore, this PRBG does not provide backward security. But given s_i and k at time τ, the adversary cannot compute any previous output of the PRBG. The attack described above for block cipher based PRBG does not work for the keyed hash based PRBG. However, we note that that the L-K scheme in [4] does need a robust PRBG to ensure its backward untraceability because the output of the PRBG is sent in plaintext.

3 Conclusion

In this paper, we analyzed an RFID identification scheme which is designed to provide forward untraceability and backward untraceability. We showed that, if a standard pseudorandom bit

generator (PRBG) is used in the scheme, then the scheme may fail to provide forward untraceability and backward untraceability. To achieve these untraceability features, the scheme can use a robust PRBG which provides forward security and backward security.

Acknowledgement

The authors would like to thank Ian Goldberg for refereing us to the paper [2].

References

[1] NIST special publication 800-90. Recommendation for random number generation using deterministic random bit generators. 2007.

[2] B. Barak and S. Halevi. A model and architecture for pseudo-random generation with applications to /dev/random. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 203–212, New York, NY, USA, 2005. ACM.

[3] Ari Juels. RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, 2006.

[4] Chae Hoon Lim and Taekyoung Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *ICICS*, volume 4307 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2006.

[5] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1996.

[6] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to "privacy-friendly" tags. In *RFID Privacy Workshop*, 2003.

[7] Boyeon Song and Chris J Mitchell. RFID authentication protocol for low-cost tags. In *WiSec '08: Proceedings of the first ACM Conference on Wireless Network Security*, pages 140–147. ACM Press, 2008.

[8] D.R. Stinson. *Cryptography: Theory and Practice, third edition*. Chapman & Hall/CRC, Boca Raton, 2006.