J Med Syst (2012) 36:431–435
DOI 10.1007/s10916-010-9487-y

ORIGINAL PAPER

Patient Safety Through RFID: Vulnerabilities in Recently Proposed Grouping Protocols

Anne-Katrin Wickboldt · Selwyn Piramuthu

Received: 16 February 2010 / Accepted: 30 March 2010 / Published online: 14 April 2010

Abstract As RFID-tagged systems become ubiquitous, acceptance of this technology by the general public necessitates addressing related security/privacy issues. The past eight years have seen an increasing number of publications in this direction, specifically using cryptographic approaches. Recently, the Journal of Medical Systems published two papers addressing security/privacy issues through cryptographic protocols. We consider the proposed protocols and identify some existing vulnerabilities.

Keywords RFID · Cryptography · Protocol

Introduction

Radio-Frequency Identification (RFID) tags are increasingly being used in the health care domain for improving the effectiveness of health care delivery (e.g., [3]). As RFID tags become ubiquitous, concerns with respect to their privacy/security characteristics are increasingly being discussed among relevant entities. In a world where identity-theft, counterfeit products, corporate espionage, tracking, etc. are rampant, the urgent need to address privacy/security issues are especially crucial for successful deployment of RFID technology. This is especially crucial in the medical environment where privacy and security issues are rather critical given the sensitive nature of data in this domain as well as possible ramifications associated with violation of privacy and security.

Although these issues themselves are not new in a general context, they are new and idiosyncratic to the RFID context primarily due to their item-level (as opposed to class-level in the case of bar codes) identification capabilities. Whereas RFID was introduced about seven decades ago, major thrust in research addressing its security/privacy vulnerabilities has existed only for about eight years. Researchers have approached several facets of security/privacy issues from a cryptographic perspective. RFID is a very active area (e.g., [5]), and several papers have already been published in the Journal of Medical Systems (e.g., [1, 2, 4, 6–8]). The Journal of Medical Systems recently published two papers addressing issues related to RFID cryptography. Both these papers consider *grouping protocols*, which are cryptographic authentication protocols that validate the *simultaneous* presence of a specific group of RFID tags in the field of the reader. We consider the protocols presented in these papers and identify some of their inherent vulnerabilities.

The remainder of this paper is organized as follows: we consider the three protocols that were proposed in these two papers and identify some vulnerabilities in these protocols in the next Section. We conclude the paper with a brief discussion in the last section.

A.-K. Wickboldt
Massachusetts Institute of Technology,
Cambridge, MA 02139, USA

S. Piramuthu (✉)
Information Systems and Operations Management,
University of Florida, Gainesville, FL 32611, USA
e-mail: selwyn@ufl.edu

Mutual Authentication Protocols

Notations used in this paper:

- r_i, r_p, r: random l-bit vectors
- ID, PIN, EPC, R: identifier
- k: l-bit shared secret key
- f_k: keyed (with key k) encryption function
- TS: time stamp
- CRC: cyclic redundancy check
- PRNG: pseudo-random number generator
- $\oplus$: Exclusive-OR (XOR)

Huang and Ku Protocol

The Huang and Ku grouping proof protocol [2] considers two types of tags—the pallet tag that is worn as a bracelet by the patient and the other (item) tags that are on items that interact with the patient. The protocol is initiated by the Reader which sends the current time stamp (TS) to both the pallet tag and the first item tag in the sequence for this grouping proof. The pallet tag waits till it hears again from the Reader. The first item tag computes its m value (Fig. 1) and sends it to the Reader, which then forwards it to the second tag, and so on. In Fig. 1, n refers to the total number of item tags and 'TS or m_{i-1}' results in TS for the very first tag and m_{i-1} for the remainder of the tags. Here, both PIN and EPC (Electronic Product Code) are used as tag identifiers. When the last tag (i.e., tag n) gets back to the Reader with its m value (i.e., m_n), the Reader forwards this to the pallet tag, which computes P that is then verified by the Reader.

This protocol uses PRNGs to encrypt messages passed among the entities (item tag, reader, pallet tag). A fundamental vulnerability is in this encryption since PRNG used in these tags are readily available to an adversary who can use this to compromise security of the RFID tagged items. Chien et al. [1] identify two vulnerabilities in this (Fig. 1) protocol of which only one is valid: Denial of Service (DoS) attack and replay attack. The former occurs when an adversary forces the tag to update its PIN by repeatedly sending random TS values. Since the tag does not verify the reader, this is easy to accomplish and the reader's version of the tag's PIN would be different from the tag's actual PIN and they no longer would be able to communicate. The latter attack purportedly occurs when an adversary sends future TS values to the tag to collect its response, which are then replayed to the reader at later points in time. However, this latter attack will not work since the tag would have changed its PIN value and the reader expects a different (updated) PIN value. Moreover, Chien et al.'s implicit assumption that the latter attack (even if it) works is false since TS is sent only to the first tag in the sequence—i.e., it does not apply to the remainder of the tags. Since the attack as proposed by Chien et al. does not work for any tag, this is moot.

There are a few other vulnerabilities in this protocol. Since the EPC of every tag (including the pallet tag) is sent in the open in plaintext, an adversary can use this identifier information to track the tagged (item as well as pallet) object [5]. Being able to be tracked violates privacy/security of the tagged object. Since these tags do not have an on-board clock, an adversary can accomplish this attack by sending any random TS to the item tag, which would respond with its EPC value. As for the pallet tag, the adversary can send random TS and m_n to the pallet tag which responds with its EPC since it does not validate TS or m_n.

An adversary can impersonate the first tag in the sequence once using the ease with which PRNG(x) can

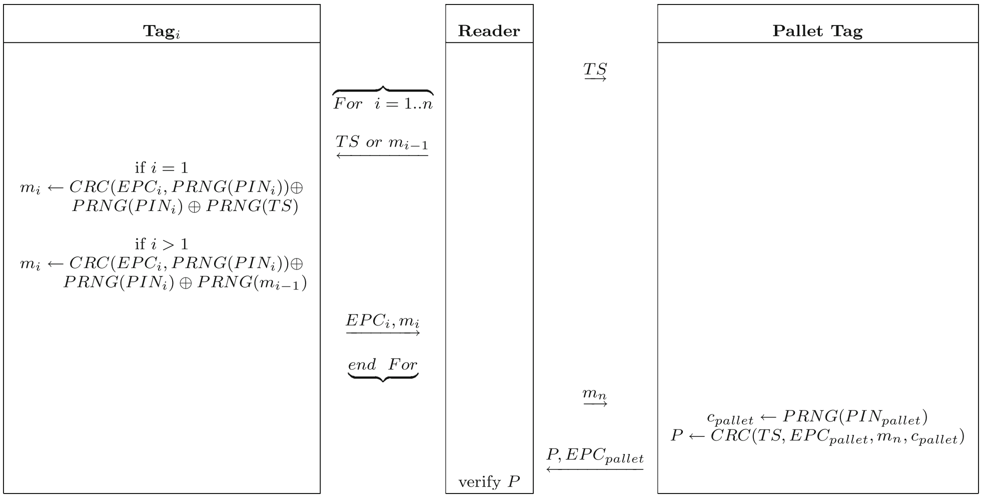
Tag_i
Reader
Pallet Tag
$\xrightarrow{TS}$
$\overbrace{For \quad i = 1..n}$
$\xleftarrow{TS \ or \ m_{i-1}}$
if $i = 1$
$m_i \leftarrow CRC(EPC_i, PRNG(PIN_i)) \oplus PRNG(PIN_i) \oplus PRNG(TS)$
if $i > 1$
$m_i \leftarrow CRC(EPC_i, PRNG(PIN_i)) \oplus PRNG(PIN_i) \oplus PRNG(m_{i-1})$
$\xrightarrow{EPC_i, m_i}$
$\underbrace{end \quad For}$
$\xrightarrow{m_n}$
$c_{pallet} \leftarrow PRNG(PIN_{pallet})$
$P \leftarrow CRC(TS, EPC_{pallet}, m_n, c_{pallet})$
$\xleftarrow{P, EPC_{pallet}}$
verify P

Fig. 1 Protocol of Huang and Ku [2]

be computed for any given x since PRNG is public information. The adversary can send a random TS_A value to the tag, which responds with (EPC_1, m_1). The adversary can capture this and wait till the reader initiates a Grouping proof round. When the reader sends a new TS_{new} to the first tag, the adversary impersonating this tag can capture it and compute $m_{\text{new}} \leftarrow m_1 \oplus \text{TS}_A \oplus \text{TS}_{\text{new}}$ and send $(\text{EPC}_1, m_{\text{new}})$ to the reader, which will accept this as a valid message.

Chien et al. Protocol [On-line Verifier]

After identifying vulnerabilities in [2], Chien et al. [1] propose modified protocols that address the identified vulnerabilities. They propose two protocols—one with an on-line verifier, and another with an off-line verifier. Both these proposed protocols, however, are still vulnerable to attacks from an adversary. We consider the on-line verifier protocol first followed by the off-line verifier protocol.

The idea behind the on-line verifier protocol is similar to the protocol presented in [2]. The primary difference is in the independent way in which each of the tags are authenticated to be simultaneously present in the field of the reader. Unlike [2], this protocol does not use a TS but uses a freshly generated random nonce (r) for each authentication round.

As in [2], the protocols proposed in [1] are vulnerable to tracking/tracing attack since the EPC values of item tags as well as the pallet tag are broadcast in plaintext (Fig. 2). Since the EPC value uniquely identifies a tag, and therefore the tagged item or pallet, this is a serious vulnerability in a medical setting.

This (on-line verifier) protocol exposes every tag in the sequence to vulnerabilities because of the independent way in which the tags are authenticated—i.e., this is unlike the protocol presented in [2] where the input to each tag (except the first tag in the sequence where it is TS) is the output from the previous tag in the sequence, where it is difficult to determine the input for any randomly chosen tag in the sequence. In response to the reader's r, every tag (including those on items as well as pallet) responds with $(m_i \leftarrow \text{PRNG}(\text{EPC}_i \oplus \text{PRNG}(\text{PIN}_i) \oplus \text{PRNG}(r) \oplus \text{PRNG}(r_i)),\ \text{EPC}_i,\ r_i)$. Since PRNG is known to the adversary and since $\text{PRNG}(\text{EPC}_i \oplus \text{PRNG}(\text{PIN}_i))$ is constant because EPC_i and PIN_i are not updated, the adversary can observe an authentication round and replay message from a previous round to the reader and can easily impersonate any tag (including the pallet tag) to the reader. This can be accomplished by generating an $r_{i(\text{new})}$ in response to r_{new} from reader such that $(\text{PRNG}(r_{i(\text{new})}) \oplus \text{PRNG}(r_{\text{new}}) = \text{PRNG}(r_i) \oplus \text{PRNG}(r))$ where r and r_i are from any previous authentication round. This attack can be prevented by a reader that keeps track of every message received from every tag and checking for repetitions. However, this involves a considerable amount of overhead that increases linearly with the number of authentication rounds.

The protocol checks for round-trip time taken by messages between reader and tag to ensure that they

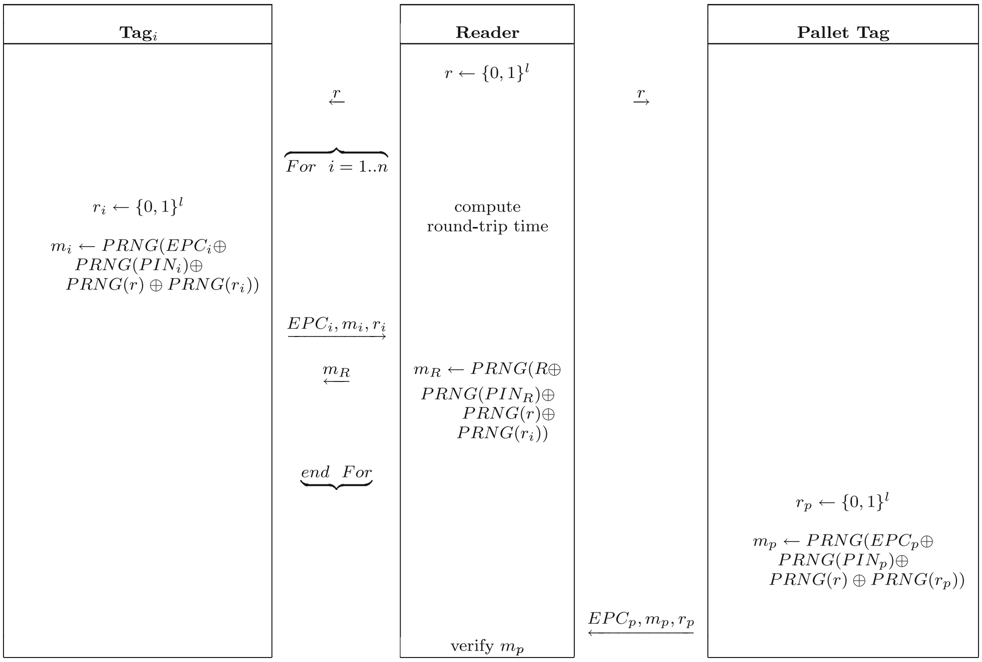
Tag$_i$
Reader
Pallet Tag
$r \leftarrow \{0,1\}^l$
$\xleftarrow{r}$
$\xrightarrow{r}$
$\overbrace{For \quad i = 1..n}$
$r_i \leftarrow \{0,1\}^l$
$m_i \leftarrow PRNG(EPC_i \oplus PRNG(PIN_i) \oplus PRNG(r) \oplus PRNG(r_i))$
compute round-trip time
$\xrightarrow{EPC_i, m_i, r_i}$
$\xleftarrow{m_R}$
$m_R \leftarrow PRNG(R \oplus PRNG(PIN_R) \oplus PRNG(r) \oplus PRNG(r_i))$
$\underbrace{end \quad For}$
$r_p \leftarrow \{0,1\}^l$
$m_p \leftarrow PRNG(EPC_p \oplus PRNG(PIN_p) \oplus PRNG(r) \oplus PRNG(r_p))$
$\xleftarrow{EPC_p, m_p, r_p}$
verify m_p

Fig. 2 Protocol of Chien et al. [on-line verifier] [1]

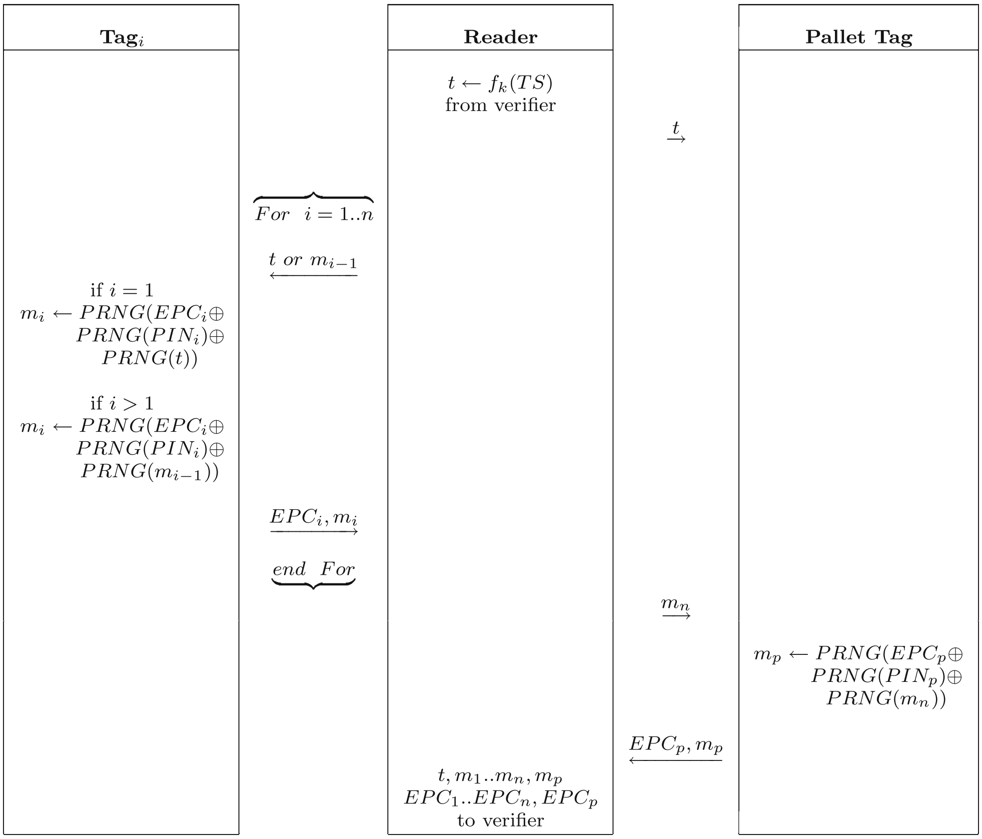
Tag$_i$
Reader
Pallet Tag
$t \leftarrow f_k(TS)$ from verifier
$\xrightarrow{t}$
$\overbrace{For \quad i = 1..n}$
$\xleftarrow{t \ or \ m_{i-1}}$
if $i = 1$
$m_i \leftarrow PRNG(EPC_i \oplus PRNG(PIN_i) \oplus PRNG(t))$
if $i > 1$
$m_i \leftarrow PRNG(EPC_i \oplus PRNG(PIN_i) \oplus PRNG(m_{i-1}))$
$\xrightarrow{EPC_i, m_i}$
$\underbrace{end \quad For}$
$\xrightarrow{m_n}$
$m_p \leftarrow PRNG(EPC_p \oplus PRNG(PIN_p) \oplus PRNG(m_n))$
$\xleftarrow{EPC_p, m_p}$
$t, m_1..m_n, m_p$
$EPC_1..EPC_n, EPC_p$
to verifier

Fig. 3 Protocol of Chien et al. [off-line verifier] [1]

are within valid limits. The purpose of this step is unclear since this protocol is clearly not immune to relay attacks (e.g., [9]). It is also unclear as to why the protocol uses $\text{EPC}_i \oplus \text{PRNG}(\text{PIN}_i)$ instead of either EPC_i or $\text{PRNG}(\text{PIN}_i)$ since EPC_i and $\text{PRNG}(\text{PIN}_i)$ are constant for a given tag. Similarly, for the reader, it is not clear why $\text{R} \oplus \text{PRNG}(\text{PIN}_R)$ is used instead of either R or $\text{PRNG}(\text{PIN}_R)$ since both these are constant.

Chien et al. Protocol [Off-line Verifier]

The Chien et al. protocol with an off-line verifier is given in Fig. 3. In Fig. 3, n refers to the total number of item tags and 't or m_{i-1}' results in t for the very first item tag and m_{i-1} for the remainder of the item tags. This protocol is a modified version of the protocol presented in [2]. Again, instead of using time stamp in plaintext (as in [2]), encrypted time stamp is used apparently to avoid replay attacks. The Reader sends the encrypted time stamp to both the pallet tag and the first item tag. The pallet tag waits for another message from the Reader. The first item tag generates its m-value (m_1) and sends it to the Reader, which then forwards it to the second item tag, and so on. The Reader sends the m-value from the last item tag in the sequence (i.e., m_n) to the pallet tag which generates its m_p value. This m_p value is then validated by the Reader.

Again, the EPC values are sent in plaintext and this leads to security/privacy violations. An adversary can replay either t or m_{i-1} from an earlier round to the tag and can receive its EPC, which is an unique identifier of the tag.

Discussion

RFID authentication protocols, despite being a very active research area, have not received much attention in medical journals. There have only been two published research papers in the Journal of Medical Systems that directly address RFID authentication issues. However, the protocols presented in both these papers are vulnerable to attacks from an adversary and are therefore not secure. We identified some of these vulnerabilities. More research is needed to develop protocols that are resistant to attacks from both passive and active adversaries to gain trust among consumers of RFID technology.

Acknowledgements The authors would like to thank the reviewers for their suggestions to improve the content and presentation of this paper.

References

1. Chien, H.-Y., Yang, C.-C., Wu, T.-C., and Lee, C.-F., Two RFID-based solutions to enhance inpatient medication safety. *J. Med. Syst.*, 2009. doi:10.1007/s10916-009-9373-7.
2. Huang, H.-H., and Ku, C.-Y., A RFID grouping proof protocol for medication safety of inpatient. *J. Med. Syst.* 33:467–474, 2009.
3. Meiller, Y., Bureau, S., Zhou, W., and Piramuthu, S., Simulation of a health care knowledge-based system with RFID-generated information. In: *Proceedings of the Asian Simulation Technology Conference (ASTEC2010)*, pp. 110–114. Shanghai, 2010.
4. Min, D., and Yih, Y., Fuzzy logic-based approach to detecting a passive RFID tag in an outpatient clinic. *J. Med. Syst.*, 2009. doi:10.1007/s10916-009-9377-3.
5. Piramuthu, S., Protocols for RFID tag/reader authentication. *Decis. Support Syst.* 43(3):897–914, 2007.
6. Shim, H., Uh, Y., Lee, S. H., and Yoon, Y. R., A new specimen management system using RFID technology. *J. Med. Syst.*, 2010. doi:10.1007/s10916-009-9417-7z.
7. Stahl, J. E., Holt, J. K., and Gagliano, N. J., Understanding performance and behavior of tightly coupled outpatient systems using RFID: initial experience. *J. Med. Syst.*, 2009: doi:10.1007/s10916-009-9365-7.
8. Sun, P. R., Wang, B. H., and Wu, F., A new method to guard inpatient medication safety by the implementation of RFID. *J. Med. Syst.* 32:327–332, 2008.
9. Tu, Y.-J., and Piramuthu, S., RFID distance bounding protocols. In: *Proceedings of the First International EURASIP Workshop on RFID Technology (RFID2007)*, pp. 67–68. Vienna, 2007.