ACSP: A Novel Security Protocol against Counting Attack for UHF RFID Systems

Ce Chen¹, Zhuzhong Qian¹, Ilsun You², Jue Hong¹, Sanglu Lu¹
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China¹
School of Information Science, Korean Bible University, South Korea²
chence@dislab.nju.edu.cn, qzz@nju.edu.cn, isyou@bible.ac.kr, jhong@dislab.nju.edu.cn, sanglu@nju.edu.cn

Abstract—Current researches on UHF RFID system security mainly focus on protecting communication safety and information privacy between a pair of specific tag and its corresponding interrogation reader. However, in many scenarios, instead of stealing detailed private information of tags, adversaries aim at estimating the cardinality of tags, which is called counting attack. Unfortunately, most existing protocols are vulnerable to counting attack. To defend against this attack, in this paper we propose ACSP, a novel Anti-Counting Security Protocol. ACSP employs session identifier and provides a corresponding authentication metric to verify the commands sent by the reader. To handle counting attack, ACSP periodically updates the session identifier, and securely identifies tags with encryption. We evaluate the performance of ACSP through theoretical analysis and qualitative comparison. Results show that ACSP can efficiently withstand counting attack while being able to defend against regular security threats as existing protocols.

I. INTRODUCTION

Ultra high frequency Radio Frequency Identification (UHF RFID) is an advanced technology for non-tangent automatic identification, which is widely used in retailing, logistics, etc. It provides advantages such as non-line-of-sight operation, higher inventory rates, and re-writable memory [1]. A complete RFID system is mainly composed of tags, readers and backend systems. Each UHF RFID tag, without battery applied, is assigned with a unique ID. And the RFID reader collects tag’s information through wireless communication.

Since RFID tags are usually attached to entities (objects or persons), adversaries can identify or trace their targets via illegitimately accessing the related tags. Four typical security threats towards UHF RFID system are summarized by Rieback in [2]: i) sniffing: adversaries obtain tag IDs by eavesdropping backscatter messages from tags; ii) tracking: adversaries track target’s movement through observing the messages attached tag responds; iii) spoofing: adversaries mimic legal RFID tags by writing proper data and ID information into a blank tag; iv) replay attack: adversaries play as men-in-the-middle in order to intercept and retransmit reader’s queries and camouflage themselves as authentic readers to communicate with tags.

Generally, the computing capability of passive tags is rather low due to the no built-in battery design. As a result, regular encryption techniques and security protocols are infeasible for RFID system, especially on the tag side. Therefore, challenge-response model is introduced to solve this problem. In this model, the reader first challenges the tag for identification. The tag then responds to the challenge and challenges the reader for authentication. Finally, the reader responds to finish the procedure.

Based on the challenge-response model, Molnar proposed a security protocol used in library RFID system [3] named PRF private authentication protocol which uses pseudo-random functions to encrypt communication messages. Henrici advanced hash-based ID variation protocol which implements ID update operation in [4], and improved this protocol in [5]. The new protocol is so called Hash Chain and it greatly simplifies the hash-based ID variation protocol. Moreover, Kulseng [6] elaborated how to realize security protocol via utilizing physical properties of tag’s circuits. On the other side, Molnar [7] and Dimitiou [8] discussed the method to improve the search efficiency when the reader attempts to find the exact ID value of the communicating tag in system applied with security protocols. They mainly used binary tree structure to manage tag’s IDs. Lu [9] proposed SPA protocol which implements tag ID update in this type of protocol. In [10], Lu further improved this protocol via utilizing spare tree structure to break the connection of IDs among each tag. These works mainly focus on how to protect the privacy and security of individual communication, which means providing a security tunnel between the reader and the tag.

Consider the following scenarios: in a warehouse, each piece of merchandise is attached with one tag which keeps a unique ID for automatic identification. A corporate spy may sneak into the warehouse and rapidly get stock information, by using a handheld reader to count the number of tag. We define counting attack as attacker’s behavior of counting the number of objects through estimating cardinality of attached tags. Although the attacker probably could not understand identification or authentication messages secured by security protocols, by observing communication messages between readers and tags or mimic as an authentic reader to probe tags, they still can conduct such attack.

To achieve counting attack, adversaries could apply either framed slotted ALOHA or tree-based scheme. In ALOHA scheme, the reader first broadcasts the initialization message to set the frame size (slot number), and then each tag picks a slot to reply based on a specific probability distribution. The reader can repeat this procedure several times and estimate the tag cardinality by analyzing reply information. For tree-based scheme, the reader identifies each tag like traversing a tree, in which root node represents the whole tag set and each leaf
node indicates a specific tag. Hence, although adversaries may not understand the encrypted messages, they still can conduct counting attack.

In order to defend against counting attack towards RFID system, we propose a novel security protocol ACSP (Anti-Counting Security Protocol). ACSP is designed to secure the whole system via protecting two main phases in RFID system: SELECTION and LOCATION. Compared with existing security protocols, ACSP provides overall security of system besides individual privacy. Our contributions in this paper are as follows:

- We define a new security threat toward UHF RFID system named counting attack, which might seriously threaten the system security.
- We propose a security protocol ACSP which can effectively defend against this attack while providing regular security features.

We evaluate the performance of ACSP through theoretical analysis and qualitative comparison. Results show that ACSP can efficiently withstand counting attack while being able to defend against regular security threats as existing protocols.

The rest of the paper is organized as follows. Section 2 describes fundamentals of RFID system, cardinality estimation approaches and counting attack. In Section 3, we propose a novel security protocol ACSP. Section 4 analyzes ACSP, and section 5 is the discussion of ACSP. We conclude this paper in Section 6.

II. FUNDAMENTALS

A. UHF RFID System Protocol

We take the prevalent EPCglobal Class 1 Generation 2 (C1G2) specification [11] as an example to demonstrate UHF RFID system model and its security weakness.

The C1G2 physical layer protocol defines a series of physical options for both up-link and down-link. The communication procedure is specified in the MAC layer section. In this paper, we mainly focus on MAC layer protocol of C1G2 specification. The communication model is called Reader Talk First, in which a Framed Slotted Aloha [12] like protocol is used. To identify a specific tag, the reader executes the following two phases: i) SELECTION, the reader sets the frame size and sends a Select command containing mask bits indicating which tag should reply; ii) LOCATION, the reader sends a Query / QueryRepeat / QueryAdjust command to start a new slot. If the selected tag decides to reply in this slot, it responds with a 16-bits random number upon receiving the slot start command. The reader echoes an ACK if no collision happens, and then the tag replies its ID information (EPC) for identification upon successfully receiving the ACK. Otherwise the reader starts another new slot or ends this frame. The details of the whole procedure are elaborated in [11].

Obviously the frame size determined in SELECTION phase is a critical parameter to RFID system performance. Schoute [13] showed that the optimum occurs when the number of slots is equal to the number of tags.

B. Cardinality Estimation Approaches

As mentioned in Section 2.A, frame size greatly affects system performance. ALOHA scheme is widely used to find a proper frame size. Kodialam and Nandagopal [14] provided a basic ALOHA method to do cardinality estimation. Let $t$ and $f$ denote the number of tag set and frame size respectively. Assume that each tag uniformly pick a random slot to reply, the number of reply the reader received in one slot follows binomial distribution $B(t, t/f)$. Thus the reader can count the number of no-reply / collision slots to do cardinality estimation and repeat the SELECTION phase several times to reduce the estimation variation. Qian [15] increased the accuracy of cardinality estimation in large-scale system via collecting the bitmaps of replies in each reader. Moreover, they improved algorithm performance by supposing the tags’ reply follows geometric distribution.

Tree-based scheme is a deterministic scheme aim at directly identifying the whole tag set rapidly in contrast. The reader probes tags like traversing a binary tree from the top down. The root represents the whole set and each leaf node stands for one specific tag according to tag’s ID. Pan and Wu [16] proposed a tree-based algorithm named Smart Trend-Traversals (STT). STT adopts query traversal path (QTP) to indicate the reader probing movement, and generate the QTP according to the type of the node (empty / singleton / collision) it confronts when traversing in the tree.

C. Approaches to Counting Attack

In order to do counting attack, adversaries can just simply implement algorithms described in Section 2.B, either deterministic or probabilistic schemes. Actually we can abstract both two algorithms perform as three steps: i) the reader determines which tags to probe and sets parameters such as frame size, and then sends commands (e.g. Select and then Query) to start. ii) The reader adjusts pertinent parameters, e.g. the frame size and the mask bits, according to the responses’ number and distribution in last slot / frame and then queries the tags repeatedly. iii) The reader makes estimation based on these replies information at the end.

We notice that both two algorithms mainly work in SELECTION phase, and what they demand is that whether one specific slot is a collision slot, a singleton slot or an empty one, instead of the detailed information contained in back-scatter messages. The aforementioned challenge-response model (as shown in Fig. 1) only secures LOCATION phase to prevent the leak of ID information, and neglects the threat towards
III. ANTI-COUNTING SECURITY PROTOCOL

ACSP imports a session identifier (SID) to defend against RFID counting attack. Tags will only respond to query commands with correct SID and reader is required to periodically update SID to ensure freshness of SID. ACSP is composed of SID Update and Tag Identification procedures.

We first define some notations used in the protocol:

- \( R_x \) Pseudo-random number \( x \) generated by reader or tags
- \( CRC \) Cyclic redundancy check code of message
- \( H(x_1, x_2) \) One-way hash function with variables \( x_1, x_2 \)
- SID Session identifier
- TID Tag identifier
- \( MASKVAL \) Bit mask contained in Select command used for selecting target tag
- \( MSGHEADER \) Header of message which indicates the type of the message

A complete message should include a header and \( CRC \) sum besides message contents.

A. SID Update

\[
\text{SELECT}_{\text{R}_1, \text{R}_2,H(R_1, \text{SID}_{\text{cur}}), \text{CRC}}
\]

This procedure periodically updates SID, which contains two steps, as shown in Fig. 2: i) The reader generates \( R_1 \) and \( R_2 \), and then sends command \( \text{UPDSID}_{R_1, R_2, H(R_1, \text{SID}_{\text{cur}}), \text{CRC}} \) to notify tags to update SID stored in their local memory. Here \( \text{SID}_{\text{cur}} \) represents the identifier used in current communication session. Due to the property of one-way hash function, SID_{\text{cur}} is hard to crack even though \( R_1 \) and \( H(R_1, \text{SID}_{\text{cur}}) \) are known. Hence, those commands which cannot provide the correct value of \( R_1 \) and corresponding \( H(R_1, \text{SID}_{\text{cur}}) \) is considered invalid. ii) Each tag checks \( H(R_1, \text{SID}_{\text{cur}}) \) and \( CRC \) and update \( \text{SID}_{\text{cur}} \) as \( \text{SID}_{\text{new}} \) if all valid. \( \text{SID}_{\text{new}} \) will be used in following session. Note that \( \text{SID}_{\text{cur}} \) is no longer valid after updating.

B. Tag Identification

Tag identification procedure provides secured communication and information exchange between the reader and tags and is compatible with prevailing industrial specifications, e.g. EPCglobal C1G2 and ISO 18000-6C.

Fig. 3 illustrate the procedure. The reader first determines the target by sending \( \text{SELECT} \) command. Here, \( R_3 \) and corresponding \( H(R_3, \text{SID}) \) are regarded as certification information. Moreover the reader uses SID to conceal the real value of MASKVAL. Thus, adversaries are hard to learn the reader’s target via eavesdropping without correct SID. Then, upon receiving the Select command, the tag \( CRC, R_3 \) and corresponding \( H(R_3, \text{SID}) \) and the tag gets MASKVAL by XORing (MASKVAL XOR SID) with SID if all correct. The tag gets ready to respond if its ID matches MASKVAL, otherwise keeps silent until receiving the next Select command.

After sending \( \text{SELECT} \) command, the reader sends \( \text{QUERY} \) command. The tags in ready state examine the \( CRC \) and \( (R_4, H(R_4, \text{SID})) \) in the \( \text{QUERY} \) command after receiving it. If correct, the ready tags respond \( IDENT \) messages respectively. If the reader successfully receives the identification message, it will search for the tag’s exact TID in database according to \( R_4 \) and \( H(R_4, \text{TID}) \). Otherwise, the reader sends \( (\text{QUERY REP}/ \text{QUERY ADJUST}, R_p, H(R_p, \text{SID}), CRC) \) to end this slot and start a new one if the frame size is not exceeded.

Once the reader finds the TID of the responding tag, updates tag’s TID as \( \text{TID}_{\text{new}} \). Then the reader responds \( \text{AUTHEN} \) message to inform the tag to update its TID immediately. The tag checks \( H(R_5, \text{TID}) \) and replace its ID with \( H(R_4 \text{ XOR } R_5, \text{TID}) \) if it is correct, otherwise it does nothing.

IV. EVALUATION

We evaluate the performance of ACSP in this section. Systems working in wireless environment are vulnerable to security threats [17], especially to RFID system whose computing capability is constrained. We first show that ACSP is able to handle with both passive and active attacks as well as counting attack. We then evaluate the overhead of ACSP.

Security Analysis. Here we assume that the communication between the reader and the database is secured.

Normally, passive attack in wireless environment refers to eavesdropping, which means attackers can know private information simply via eavesdropping communication messages. In RFID system, tag ID is the critical information. In ACSP, this attack is prevented by encrypted TID using one-way hash function and a random number. It is hard for attackers to find TID from \( R_x \) and \( H(TID, R_x) \) due to the preimage-resistance property of one-way hash function. Note that attackers may guess ID information by analyzing the MASKVAL in Select command and then observing the replies from tags. This potential danger is ignored in existing literatures. Hence, the reader is required to encrypt the MASKVAL by XORing with SID in ACSP. The original value of MASKVAL can only be revealed in case SID is known.

Active attacks towards RFID includes tracking, spoofing and replay attack. For tracking attack, ACSP utilizes TID
update and random number challenge to defend against illegitimate tracking. These operations ensure that the encrypted TID information $H(TID, R_4)$ in reply messages will vary according to the change of the random number. Besides, if successful authentication is made, the tag will update its TID consequently. Thus, attackers can hardly track the tag without seizing the precise value of TID. For spoofing attack, in ACSP, each tag is required to reply identification message containing $H(TID, R_4)$, where $R_4$ is given in Query / QueryRepeat command. This indicates the attacker could hardly generate correct $H(TID, R_4)$ to deceive the reader without knowing TID. For replay attack, ACSP ask the reader to respond according to the random number in identification message sent by the tag. The tag would not update its TID upon receiving an incorrect authentication message. Moreover, we can make the reader and the tag generate random number differs from the previous one. Each tag keeps a copies of the random numbers contained in last received valid command. If the tag receives a command with the same random number as the lastest one, this command is considered as an illegal one.

The most important concern of ACSP is how to prevent the system from counting attack. As mentioned above, attackers do not care about specific tag IDs and their corresponding objects. Nevertheless, common security protocols secure only the identification and authentication messages to protect the privacy of a specific tag and may fail to handle such attack. ACSP provides a simple but efficient way to eliminate the counting threat. If the reader could not give the correct $R_x$, $H(R_x, SID)$ pair, tags will simply ignore the command and keep silent until it receives the right one. In addition, MASKVAL is encrypted by XORing with SID to block the attacker from peceiving target tags to estimate the number of tags by observing the replies from tags. Adversaries cannot forge commands to make tags respond as his will without correct SID. This situation can hardly occur as the reader periodically updates SID to protect the attacker from guessing SID by brute-force trials.

**Overhead.** To implement ACSP, both reader and tags have to consume an extra memory space to store SID. Besides, a pseudo-random number generator and hash function logic circuits are required to equip. Fortunately, the state-of-the-art electronics industry provides feasible solutions for RFID system, especially at tag side, to realize logic functions, e.g. Linear Feedback Shift Register (LFSR) as random number generator [6], AES-like algorithm to achieve the effect of one-way hash function [18] etc. Assume each time either the reader or the tag executes XOR / hashing / random number generates operation cost $C_{XOR}$, $C_H$, $C_R$ units of resource respectively. Let $n$ stands for the total number of tag ID stored in backend database. The overhead of a successful SID update and tag identification procedure in ACSP is illustrated in TABLE I. Note that computation cost of ACSP includes two procedures as well as Select command, which is rarely mentioned in previous related works. It can be observed from TABLE I that our ACSP has the ability to handle counting attack with acceptable additional system cost compared with several representative existing security protocols.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Eavesdropping</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tracking</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Spoofing</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Replay Attack</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Low search overhead in identifying the tag. No update operation</td>
<td>Low search overhead with update operation</td>
</tr>
<tr>
<td>Counting Attack</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Features (Reader)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TID information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search Overhead</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(\log_2 n)$</td>
<td>$O(\log_2 n)$</td>
</tr>
<tr>
<td>Features (Tag)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overhead</td>
<td>$3C_{XOR} + 7C_H + 1C_R$</td>
<td>$2C_{XOR} + 2C_H + 1C_R$</td>
<td>$2C_H$</td>
<td>$1C_{XOR} + \log_2 nC_H + 1C_R$</td>
<td>$(\log_2 n + 1)C_H + 1C_R$</td>
</tr>
<tr>
<td>Computation Overhead (Tag)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computation Overhead (Reader)</td>
<td>$3C_{XOR} + 6C_H + 4C_R$</td>
<td>$1C_{XOR} + 1C_H + 1C_R$</td>
<td>$3C_H$</td>
<td>$1C_{XOR} + 1C_R$</td>
<td>$1C_H + 1C_R$</td>
</tr>
<tr>
<td>Search Overhead</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(\log_2 n)$</td>
<td>$O(\log_2 n)$</td>
</tr>
<tr>
<td>Eavesdropping</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tracking</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Spoofing</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Replay Attack</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Counting Attack</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Features</td>
<td>Easy to deploy. No update operation</td>
<td>Only require three hash function to implement the protocol</td>
<td>Low search overhead in identifying the tag. No update operation</td>
<td>Low search overhead with update operation</td>
<td></td>
</tr>
</tbody>
</table>

aUpdate overhead is not included in the table. Actually the computation consumption of one complete ID update procedure is $\log_2 n \times C_H$ at both sides.

bSearch computation overhead is shown in Row “Search Overhead”.

**TABLE I**

**PERFORMANCE COMPARISON**

In real system, signal turbulence and attenuation occurs frequently. In ACSP, message lost may result in desynchronization between the reader and tags. If SID / TID update command is lost, only the reader updates SID / TID while tags do not. Thus, the legal reader cannot communicate with tags in following communication. To solve this problem, we slightly modify our ACSP as follows.

In modified SID update procedure (as shown in Fig. 4), the reader is required to set a retransmit limit and repeatedly send
update command, as described in section 3.2, until no tag responds or the retransmit times reaches the limit. Each tag should respond an UPDACK message upon receiving the valid update command and updates its stored SID. Once the SID is updated, the tag will no longer reply upon receiving the old update command. Here the retransmit limit is mandatory due to the concern of Denial of Service Attack (DoS), if adversaries continuously send ACK messages.

To solve the message lost problem occurs in tag identification procedure, the reader preserve a copy of TID used in last successful identification for each tag. The reader can search for the last successful identification TID if the tag does not update its own TID. This technique has been already implemented in several related work such as [4] [5].

However, we do not consider physical attack during designing ACSP, because we believe that attackers may also lack of equipments which can directly read tag’s memory. Furthermore, we can apply access password to RFID tags so that if illegal access happens, the tag will automatically self-destroy. We will discuss protocol design concerned of physical attack in future work.

VI. CONCLUSION

Current research works on RFID system mainly focus on the methods to protect communication security, especially the privacy of ID information, between one single tag and one specific reader. Unfortunately, most existing protocols are vulnerable to counting attack, which happens frequently in lots of applications. In this paper, we first formally defined counting attack. Then, we proposed ACSP, which is compatible with industrial system specifications and is able to handle counting attack efficiently as well as other common security threats. We evaluated the performance of ACSP through theoretical analysis and qualitative comparison. Results show that ACSP can efficiently withstand counting attack while being able to defend against regular security threats as existing protocols.

ACKNOWLEDGMENT

This paper is partially supported by the National Basic Research Program of China (973) under Grant No. 2009CB320705, the National Natural Science Foundation of China under Grant No.90718031, 60721002 and Jiangsu Natural Science Foundation of China under Grant No. BK2008264.

REFERENCES


