A Secure RFID Identity Reporting Protocol for Physical Attack Resistance

Zhaoyu Liu and Dichao Peng
Department of Software and Information Systems
University of North Carolina at Charlotte, Charlotte, USA
Email: {zhliu, dpeng}@uncc.edu

Abstract— **RFID tags will exist pervasively in our daily life in a near future. In RFID systems, privacy and security are of critical importance to avoid potential tracking abuse and privacy violations. Many protocols have been proposed to address privacy and security issues in RFID systems. Physical attacks receive few considerations from the current research. Through physically attacks, attackers can get the secret identification-related information stored on RFID tags, and can later use the obtained information to impersonate legitimate readers for illegal tracking. In this paper we describe a threat model of RFID systems and present a detailed analysis of physical attacks on RFID tags and other security threats. We propose a secure identity reporting protocol to address these threats. Our security and performance analysis show that the proposed protocol can defeat physical attacks, in addition to other security threats, and scale well to large RFID systems.**

Index Terms— **RFID, Privacy, Physical Attacks, Secure Identity Reporting**

I. INTRODUCTION

Radio Frequency Identification (RFID) technology, one of the forerunners of pervasive computing, is widely regarded as the successor of optical bar codes. Industries of manufacturing, supply chain management, and inventory control can benefit this technology to help reduce the costs wherever bar codes used to dominate. In a report released in May 2005, the US Government Accountability Office found that thirteen government agencies are using or plan to use Radio Frequency Identification tags [7].

RFID systems can be roughly classified into two categories, according to the capability of the tags: one is named active RFID system, with self-powered tags that can actively report its ID with support of complicated on-chip algorithms while the other is passive RFID system, in which the tags have no power supply and can only respond when it is queried. Passive RFID system is more promising because of its low cost and similarity to the current application model of bar code systems. For simplicity, we refer to passive RFID system throughout this paper, without specifying the passiveness.

The RFID system is composed of three parts: RFID tags, readers and a backend database (Figure 1). The tag (also called the transponder) consists of an integrated circuit and an antenna without any power supply. Only when it receives a reader's query, the tag becomes activated and makes use of the energy absorbed from the reader's query signal to respond with this limited power. The reader is a device that sends query signal and identifies the tag from its response according to their communication protocol. The reader is usually connected to a centralized backend database running on a secure server, where all tags' information is stored. For security reasons, the data that a tag sends directly to the reader should not contain any identity information in plaintext. Only the authentic readers connected to the backend database can extract the tag ID from its response.

The communication channels between readers and the backend database can be protected by common security technologies and we will assume these channels to be secure throughout this paper. The radio frequency communication channel between reader and tags is vulnerable to various attacks. Especially, because of the abundant power of the reader, the channel from reader to tag can be very easily overheard by adversaries.

Privacy is an important issue in RFID systems. If an attacker can distinguish a tag by using fake readers to query its identity, the tag's carrier can be tracked. Thus the victim's privacy (e.g. real-time location, medicine that she takes, shopping habit, etc.) could be seriously violated.

A secure identity reporting protocol is requisite to protect user privacy. Many approaches have been proposed, but most of them mainly focus on the threat from eavesdropping. In this paper we thoroughly discuss various attacks (including the most powerful one: physical attack) to RFID systems and propose a new protocol that can resist these attacks to protect user privacy.

Our main contributions are as follows. First, we present a threat model to address possible attacks to RFID systems. We provide thorough analysis and definition of physical attacks. We classify resistance to physical attacks into three levels and provide countermeasure directions. Our second contribution is to propose a secure identity reporting protocol for privacy protection against various attacks. We analyze that our protocol can resist physical attacks, and other possible attacks including

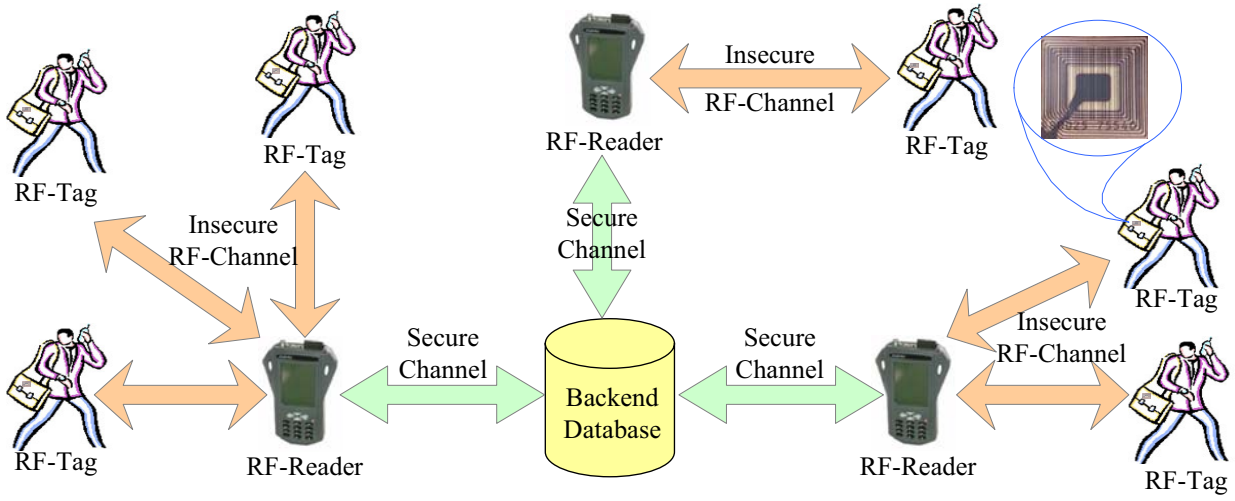

Figure 1. RFID System

eavesdropping and message hijacking. The performance analysis shows that our protocol can scale well to large RFID systems.

The rest of this paper is constructed as follows. Section II discusses the general privacy and security issues in RFID systems and presents a threat model. In Section III we first provide an analysis of physical attacks and suggest countermeasure directions. Then we present our own protocol to address various threats, including physical attacks, in order to achieve secure identity reporting. In Section IV we analyze the security and performance of the proposed protocol in Section IV. In Section V we discuss related work as well as our future research directions. Finally we conclude the paper.

II. Privacy and Security Issues in RFID Systems

The threat to user privacy is a major hurdle to the expansion of RFID industry. The US State Department is even considering backing off its RFID passport program due to the potential threat of unauthorized data access.

In many adopted RFID systems, the tag responds the reader's query with its unique serial number, without verifying the reader's authenticity. This unique number can act as a clue for the adversaries to identify the tag carrier, thus threatening the user privacy. Due to this reason, many boycotts [4, 5, 16] are lunched against RFID technology. RF-Dump [18], a project in practice, has made the anxiety of privacy violation come true. With a normal RFID reader connected to a laptop via serial port, RF-Dump can collect tag ID without hacking anything.

Many approaches have been proposed to address the privacy issue. Most of them adopt similar schemes based on secret-sharing between tags and the database to achieve secure identity reporting. These approaches are designed to protect user privacy from various attacks such as eavesdropping. Most of them can solve the problem if there are no physical attacks performed by dedicated attackers, who can break into the tag memory by force and steal the shared secret information [15, 24]. With this derived information, the attacker can later differentiate this victim tag from thousands of tags' response so as to track the tag's carrier. Our research is motivated by the threat of physical attacks. We thoroughly analyze physical attacks to propose a secure identity reporting protocol to prevent various attacks. In the following section, we will present a threat model to identify these attacks.

A. Threat to Data Secrecy

The major threat to RFID system is the potential violation of data secrecy, which comes from the illegal reader's attempt to query tag identity. Technically the adversary's goal is to identify a victim tag from its peers to assist social analysis towards compromising the tag owner's privacy.

The attacks performed by the adversary can be generally divided into two phases. The first phase is the process of critical information collecting: the adversary chooses a victim tag to attack and try to collect any tag related information for later identification. During Phase II the adversary sends queries to unknown tags and collects their responses. She then analyzes the collected responses to distinguish the victim tag, with the help of information collected in Phase I. Phase I attack is usually carried out before Phase II, but not necessarily.

1) Phase I, Critical Information Collecting

During the attack preparation phase, the identity of the victim tag is known to the adversary, who is trying to break security obstacles to obtain critical information related to this specific tag. The attacker's goal in Phase I is to collect enough identifying information of the victim so as to recognize it from the wild later. According to the RFID systems architecture, we categorize the possible attacks into the following classes:

- Eavesdropping
- Message Hijacking
- Physical Attacks

Eavesdropping is the most cost-efficient and practical form of attack because the attacker only needs an RFID reader to passively listen on the communication channel between the authentic reader and a victim tag that she

wants to track. The overheard message will then be analyzed so as to assist later re-recognition of the victim tag. The analysis of this message could be cryptanalysis such as off-line guessing the encryption keys (if applicable), trying to find out the relations of different responses from the same tags and etc. Since this analysis is a ciphertext-only-attack, it presents very little threat to most modern cryptographic algorithms. However, when combined with other attacks, eavesdropping may become powerful.

Most current researches assume that only the simplex channel from reader to tag is easy to be eavesdropped due to the low transmission power of tags. We believe that a robust protocol should also resist eavesdropping on the reverse channel, since the attacker may manage to achieve enough proximity to the tag in some application scenarios. For example, an attacker may pretend to check out a book at the library gateway just behind the person who she wants to track. Thus the communication between the tag on the victim's book and the library's RFID reader can be easily overheard by the attacker's device.

Additionally, the attackers can hijack the communication packets between reader and tag. They can not only eavesdrop on the message sending from the reader, but block the tag from hearing it by adding noise and generate a fake message to spoof the tag. This hijacked message may contain reader authentication information for tag management, depending on the protocol design. A special form of message hijack is record-and-replay attack, in which the attacker first blocks the tag from hearing the reader and later replays it in order to impersonate the reader. In addition, message hijack attacks can also be performed in the reverse direction (tampering the tag to reader packets). However, this attack will less affect the potential exposure of the tag's identity since only the legitimate reader is spoofed. However, we do not consider impersonating the tag is an active issue. There are two reasons for this: first, comparable attacks such as physically cloning a tag is always possible and second, for the RFID's precursor - optical bar codes, impersonating is much easier but not many complains have been heard about this issue.

Finally, a more dedicated attacker can even achieve physical access to a tag and compromise its memory. By the means of shaped charges, laser etching, ion-probe etc [19, 23], attackers can derive critical information such as identity number, authentication keys etc. from the tag's memory. This attack was not given enough respect in the current researches. However, as we will discuss later, physical attack is a feasible option for dedicated attackers and the cost of attack is moderate.

2) Phase II, Query

During the Phase I attacks, the adversary tries to obtain critical information of the victim tag. At that time the tag's identity is known to the attacker and she wants to later identify the victim after sending it back to the wild. While collecting victim information is the main task of Phase I, Phase II is dedicated to query thousands of tags in the wild and distinguish the victim.

To achieve this goal, the attacker may install one or several fake readers at different places in the outside world, which follow the normal query protocol as legitimate readers does and collect tag responses. Then the attacker applies the previously collected information obtained in Phase I to distinguish the victim tag among thousands of tag responses.

This fake reader's query is different from eavesdropping attacks though they may share similar devices. With eavesdropping attacks, the adversary only passively listens on the communication channel between the tag and legitimate readers, while during the query the fake reader actively send messages to ask the tag to respond. Moreover, during eavesdropping attack the victim's identity is known to the attacker and in the query process, the attack has no idea if the queried tag is just the victim that she wants to track.

Since in most proposed schemes the reader's query is a general plaintext message with no authenticity information for the tag to verify, to generate a fake query is very simple and requires no attack techniques. In addition, the tag can not distinguish queries from legitimate readers or fake ones. So the attacker's query can never be prevented. To protect the tag identity from being leaked, protocols based on cryptographic algorithms should be proposed to set barrier to the attacks in Phase I.

B. Threat to Data Integrity

In addition to the majority attacks of Phase I and II that attempt to violate user privacy, an additional minor threat is the potential compromise of data integrity and availability, as discussed below.

The threat to data integrity comes from the adversary's attempt to tamper the data, either by hijacking the message or physically tampering the tag's memory. As X. Zhang et al. discussed in [26], we should put two aspects of integrity need into consideration: first is how well the protocol resists unauthorized modification (including physical attacks) and the second is how to detect such tampering. Theoretically, the first criteria can not be met because we can neither stop attackers hijacking the message in the open air, nor prevent people from physically analyzing tag's logic component, reverse engineering the IC chips or modifying the data in the memory. But it is possible to satisfy the second criteria and detect unauthorized modification, which is the best achievable data integrity.

C. Threat to Data availability

Additionally attackers can also disable the tag to threaten the data availability. The simplest way to do that is to use a Blocker tag [11] to temporarily block the tags in a small area. A more destructive attacker can even use an electromagnetic weapon to physically damage a tag [10]. This Denial of Service attacks is a general problem to all cryptographic protocols and the discussion of preventing such attacks is beyond the scope of this paper.

III. A Secure Identity Reporting Protocol

In this section, we present a thorough analysis of physical attacks which motivates our research. Then we propose a secure identity reporting protocol to address this problem, together with other threats discussed in Section II.A.

A. Physical Attacks Analysis: Motivation

1) Attack Scenario

Hereby we use this RFID scenario as an example: Bob wants to know what his friend Alice is doing everyday. Bob knows that recently Alice has borrowed an RFID embedded handbook from the local library. So he installed a few RFID readers in places where Alice might show up, since he knows the handbook is always carried along with Alice. A couple of weeks later, Alice returns the book to the library and Bob, of course, becomes its next "reader". He uses special devices to access the tag memory and successfully digs out the secret key of this tag. With the help of data collected by his RFID readers and this secret key, Bob can filter the data that his readers have collected over these days and clearly see where Alice has been. When she gets up and left home, how often she goes out for shopping, when and how long she takes part in a club etc. Alice's private life is under surveillance by the RFID book she carries. If this attack is not severe enough, critical articles such as RFID tagged banknotes [12, 20, 25, 26] can be tracked in the same way to help criminals committing a robbery.

2) Attack Feasibility

According to the attack scenario, we define physical attacks in RFID systems as follows: attacks that use special devices to read tag memory and obtain/change identification-related information so as to compromise identity reporting protocols. Though attackers may also take advantages of these devices to physically access other item-related information stored in tag memory such as product price, book title, check out time, etc. that is not the physical attacks we are discussing about.

Generally speaking, all embedded systems without occasional human surveillance should give consideration to physical attacks. In addition, due to the price limit, RFID tags can afford no special provisions for secure storage areas where secrets are kept, which makes physical attacks even easier.

Physical attacks have been seen on various median to small sized IC chips such as USB tokens [8] and smartcards [1]. Up till now, no physical attacks on RFID chips have been published. However, since there is no significant difference between the IC design of RFID tags and smartcards at the chip level, similar attacks performed in [1] can be deployed on the RFID tags. For example, the attacker can also use fuming nitric acid to remove the resin coat of the tag, disconnect the microprocessor and attach one single micro-probing needle to the data bus. Then providing the address signal on the address bus, the secret EEPROM content can be accessed.

In many RFID applications, the tags can be temporarily possessed by the adversaries, e.g. in RFID library systems. Then the tags can be physically attacked and returned to the legitimate owner. This physical attack can be carefully performed so that the legitimate owner will not notice that it has been attacked. Even if the attacker has to damage a tag in order to access it memory, she can clone one since the data in tag memory is known.

Another simple but efficient way to achieve tracking is to attach a new tag to the victim other than "wasting time" to perform physical attacks to the legitimate tag. However, attaching a new tag to the item and returning it to the legitimate user plays a risk of being detected, especially when the item that carries tags are small and in "clear" form, e.g. bank notes. Correspondingly, physical attack can be used to track the victim without leaving evidence. So physical attack is still a safe option for the attackers, though the attacker has to pay higher cost.

From the discussion above, we can see that physical attack is a feasible attack with moderate cost. It has the potential to compromise user privacy in RFID systems. Although no physical attack incidents have been reported, as this technology's fast and widely deployment in, profitable gains will finally attract attackers' attention to put it into practice in the future.

3) Countermeasures against Physical Attacks

Generally there are two research directions towards preventions of physical attacks. One is hardware approach, applying tamper resistance property [22] to the system; the other is software approach, focusing on cryptographic ways to solve the problem.

Applying tamper resistance property to the tag is an effective way to prevent physical attacks. However, due to the current price ceiling of $0.10 per tag [24], tamper resistance is too luxurious to be deployed on RFID tags [15].

To resist physical attacks by the means of software, an important rule is that no authentication keys should be shared between tag and the backend database, because the secret key in tag memory is susceptible to physical attacks and can be utilized to compromise the protocol and track the victim. However, some necessary information needs to be shared for the backend database to identify the tag. We focus on minimizing the impact of physical attacks on the shared secret to prevent long term tracking.

4) Resistance Classification

To better evaluate the system protection over physical attacks, we classify RFID identity reporting protocol's resistance into three levels:

- Level I (insecure): Adversaries can perform physical attacks once and track the victim tag forever. If we set the time of physical attack as a time line, either the queried data before or after this time can be used to track the victim tag.
- Level II (backward secure): Through physical attacks, adversaries can not trace the data back through past events in which the tag was involved before physical attacks. But they can still use the secret information, which was compromised by physical attacks, to pass subsequent authentication and track the tags afterwards.

- Level III (secure): Adversaries can not compromise the identity reporting protocol through physical attacks.

Level III is the most secure, but Level II resistance is enough in some practical scenarios. For example, in our scenario, since Bob can not predict which book Alice will borrow from the library, he has to collect the tag's response first and later physically attack the tag when Alice has returned the book. In such application scenarios, backward security is acceptable.

Physical attacks are not a concern in many existing protocols that follows shared secret schemes, thus their resistant lies in Level I in our categorization. However, a simple but efficient improvement can level up their resistance. In most of these protocols some secret information (e.g. an encryption key) is shared between reader and each tag for authentication purposes but it is never changed. To prevent physical attackers from "hacking once, tracking forever", the tag can periodically ask the legitimate reader to refresh the secret information (e.g. update the key) after the reader is authenticated. This scheme is similar to some web system periodically asks its users to change password. In this way the resistance to physical attacks can be enhanced. Though this does not solve the physical attacks problem, such an information refreshing scheme is useful in scenarios where tags are frequently reused, e.g. in library RFID systems.

B. A Secure Identity Reporting Protocol

In this section we will propose a secure identity reporting protocol that can resist various attacks such as eavesdropping, message hijacking and physical attacks. We will propose the protocol and analyze its performance. The security analysis of the protocol will be presented in the next section.

As discussed above, to design a secure RFID identity report protocol, a few hardware limitations should be considered.

- Only cheap hardware implementations can be afforded by RFID tags. The price per tag limits the number of logic gates for security purpose to a few thousands, which can only support cheap functions such as one way hash. Though recent hardware development implements AES into low-cost RFID tags [6], public key cipher, e.g. RSA, is not an affordable option for a low-cost tag for the IC industry in the foreseeable future.
- The communication channel between backend database and the reader is secure from eavesdropping. The backend database is secured by authentication and authorization technologies so that only authentic readers can request the database to extract the tag identity.

Besides these hardware limitations, several designing principles should be followed to propose a secure identity reporting protocol. First of all, the tag's response to the reader should be ideally randomized. That is, no one should be able to tell whether two responses are from the same tag or not. Secondly, the protocol should scale with large number of tags in a system. For example, if the tag simply hashes its ID together with a random number and sends it back to the reader, the backend database has to exhaustively search every entry to find a match, which is not scalable to a large number of tags. Finally, after the reader successfully located the tag's data from its backend database, it should convince the tag of its authenticity to further manage the tag. A secure way for the reader to prove its authenticity to the tag should be developed. In this section we will propose a protocol concerning about these principles and the hardware limitations.

1) Initial Setup

Our proposed protocol relies upon a few tokens ($T_0...T_{k-1}$) pre-stored in tag memory during setup (Figure 2). Each token is composed of two parts. The first part is the tag *ID*, a random number R_i and the token serial number *s*, encrypted by the backend database with its master key *m* (i.e. $E_m(ID||R_i||s)$, where || is a concatenate operator). The second part is the one way hash of R_i. During setup, the backend database (the only party that knows *m*) generates *k* tokens $T_0...T_{k-1}$ for each tag.

Besides the *k* normal tokens, one extra token T_j is generated the same way and stored on the tag. This extra token is for second channel recovery only and will not be sent out when queried. Besides, each tag shares the value *pad, s* with the database, where *pad* is used as a one time pad to mask the token update and *s* is a token serial number. The backend database consists of entries of each tag's *ID* and the corresponding *pad, s*, as well as other information related to each tag carrier, such as product name, producing date, last check out time, etc., depending on the application requirement (optional). The shared value *pad* is used as a one time pad to mask the token update, while *s* is the token serial number used to synchronize the token version to prevent tampering and many other attacks. The function of *s* will be discussed in detail in Section IV.B.3.

2) Protocol Execution

The same as most existing protocols, our protocol starts with the reader's plaintext query to tag. On receiving the query signal, the tag sends its first token T_0 back to the reader, which forwards the token to the backend database to request a token decryption. The database verifies the authenticity of reader with normal authentication techniques and decrypts the token. It also generates *k* new tokens (i.e. new $T_k...T_{2k-1}$) using its master key *m* and a new *pad'* for future token update. The database then expand the current *pad* to the length of *token||pad'* and XOR it with them. This padded token update will be returned to the reader together with the extracted *ID* and R_i. The reader removes *ID* in the message and forwards the rest of message to the tag.

On receiving the token update, the tag will first check the validity of this message by hashing the authenticator R_i. If the hash of returned R_i equals to the corresponding value in token T_0, the tag will unpad the token updates and place these new tokens as well as new *pad'* into its memory. Otherwise this message is discarded. To acknowledge the database that the *pad* is successfully

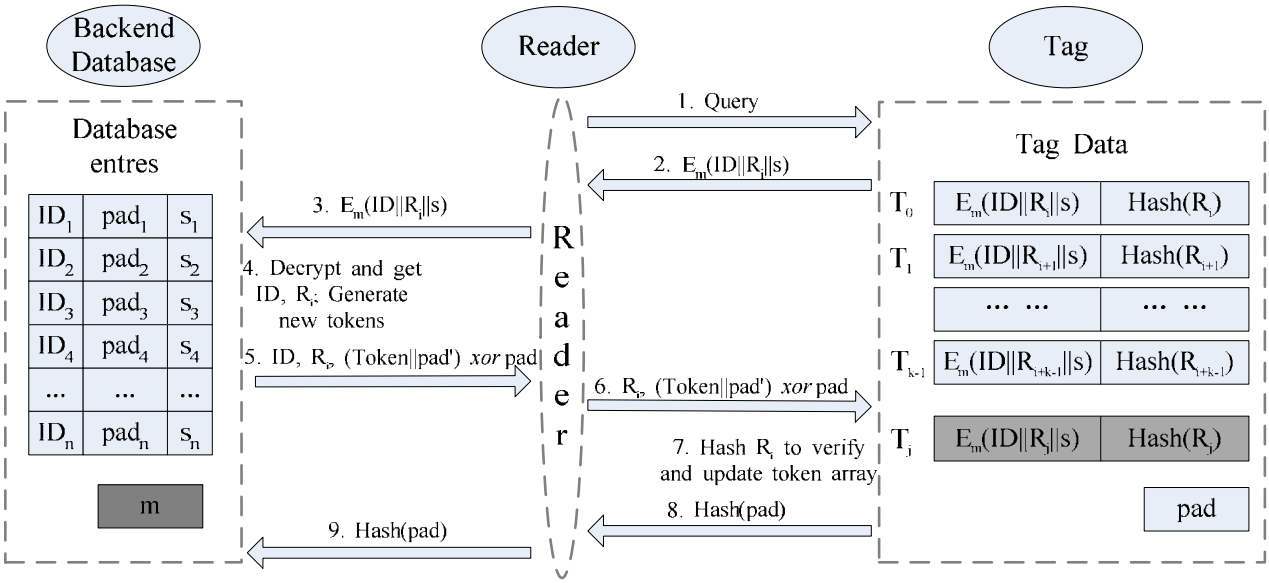

Figure 2. A Secure Identity Report Protocol

updated, the tag sends the hash of *pad'* back to complete the protocol.

3) Exception Recovery

If all the tokens T_0... T_{k-1} are consumed by unsuccessful queries with no valid token updates received, the tag will respond a message "main channel halt" to further queries and the second channel becomes active. The second channel could be a physical contact interface [21] or a radio frequency channel with a greatly attenuated power that can only be heard within a few centimeters.

It is reasonable to assume that the attacker can not track the tag within such proximity that she can almost contact the tag. So the second channel can safely respond every query with the same token T_j (See the shadow part in Figure 2) without being afraid to be tracked by this token. The tag then waits until a reader send back the right authenticator R_j for the token T_j and recharge the tag with plenty of new tokens, following the standard identity reporting protocol in Figure 2. Through this second channel replenishment, the tag gets recharged again.

IV. Performance and Security Analysis

A. Performance Study

The performance study in this section will show that our proposed protocol has light-weighted hardware complexity and good scalability, but with lower system reliability.

The first criterion to evaluate the protocol performance is the hardware complexity of the tag, which can seriously affect the price per tag and thus its practicability. In our proposed protocol, only two computational functions: one way hash and exclusive-OR, are implemented in the tag. The entire encryption/decryption burden is left for the backend database to provide the secrecy of tag ID. The tag follows very simple logic to perform transmitting/receiving message, hashing and updating memory. Though some redundant bits are used to store multiple tokens, the hardware complexity is moderate and the price per tag still remains at a low level.

Secondly for the criterion of protocol complexity, our approach adds three more communication rounds (message 6, 8 and 9) to most existing schemes. More communication round may suggest lower system reliability, or specifically, lower successful rates of identity reporting. Further implementation research is requisite to evaluate the degree of system reliability degradation.

For the scalability criterion, in our approach the computational complexity on the database side is O(1), as our database always does only one decryption per authentication to derive the value of ID. Compared with the O(log(n)) complexity in [14] and O(n) complexity in [15, 17, 24] (where n is the number of tags within a system), our protocol is scalable with the gigantic size of RFID system in practice.

Another benefit of our approach can make its performance more attractive. Though not economical, some RFID systems require item-related information (e.g. product name, last check-out time) stored on tag other than in the centralized database. It is not a good practice to store such information in plaintext because the attacker can physically attack the tag and access this data. In our approach, this item-related information can be stored in cipher text within each token, (i.e. E_m *(ID* || R_i || *s* || *ItemInfo)*) which can provide confidentiality of the data.

B. Security Analysis

In this section we will evaluate the resistance of possible attacks of the proposed protocol according to our threat model. As we discussed, the potential attacks that may threaten user privacy are eavesdropping, message hijacking and physical attacks. In addition the attacker can also compromise the data integrity and availability by other means such as denial of service attacks.

1) Resistance against Physical Attacks

The main advantage of our protocol is its resistance to physical attacks. In our scheme, the only data stored in tag memory is the array of tokens and the *pad*. We

assume Carol, the intruder, can compromise the tokens and use them to analyze the response from tag.

For the first part of token: E_m *(ID* || R_i || *s)*, provided the encryption algorithm *E* is secure (e.g. AES-128), it is infeasible for Carol to get either *ID* or R_i without knowing the master key *m*. Thus she is not able to track the victim by extracting *ID*. However, Carol can memorize all the *k* tokens and send query to see if these tokens returned back so as to track it for a maximum of *k* times. If the tag meets legitimate readers before been tracked *k* times, all tokens will be refreshed and Carol will lost the track of victim tag. We call this fruitless attack "physical read tracking", which is totally different from regular physical attacks that compromise the authentication keys once and track the tag forever.

The above discussion is based on the premise that Carol can only read tag memory to compromise user privacy. If Carol can tamper the token content, she may generate fake tokens to replace the existing ones in order to gain completely control over the victim tag. As we discussed in Section II.B, this tampering attack could not be prevented under the assumption that physical attack is possible. The best resistance to tampering attack is to detect it. In our protocol, the backend database can not derive a valid *ID* by decrypting the fake token generated by the tempering attackers so that the attack can be detected when the tag meets the authentic reader.

2) Eavesdropping Attacks

In our proposed protocol, message 6 is a meaningful reader to tag message, which can be eavesdropped by the attackers. This message is composed of two parts, an authenticator R_i and the padded token update. Because the token update is encrypted by a one time *pad*, we assume the eavesdropper can not remove the pad to derive the token update as long as the *pad* is secret.

However, according to our security assumption, the *pad* on tag memory can be compromised by physical attacks, with a higher cost. Thus physical attacks combined with eavesdropping can successfully track a victim because subsequent *pad*s can be decrypted by the current *pad* thus forms a vicious circle. Though this attack the adversary can track the victim for *k-1*times per eavesdropping (she has to leave at least one token available for the legitimate reader to query). However, this combined attack has a very high premise that the attacker has to eavesdrop on every single session when the tag communicates with any legitimate reader. Once the attacker missed one communication session, she will lose the trail of *pad* thus lose the track of the tag, which can make the physical attack very fruitless. Moreover, another active way to defeat this combined attack is that the database can track the number of remaining tokens on each tag. If one tag shows abnormally small number of remaining tokens, the database will send a warning to the reader and ask it to refresh the tokens in a more eavesdropping-resistant way, such as using the second channel.

3) Message Hijacking

As discussed in our threat model, a more dedicated eavesdropper can hijack message 6 by replacing the token updates with a fake one. Since the authenticator R_i in the hijacked message can pass the validity check, the tag will update the fake tokens according to the hijacked message. This message hijacking attack can successfully compromise the data integrity, but it brings very little threat to user privacy. Without knowing the current *pad*, the attacker can not determine the token value after the hijacked message is padded with *pad*, thus she is not able to track the tag.

A more powerful attacker who is able to obtain the *pad* by physical attacks can strengthen the power of message hijacking. She can hijack the token update message and replace the new tokens with exactly the same existing tokens in the tag (i.e. prevent the tag from updating tokens). Later when her fake readers find a tag sending T_0 to respond query, the attacker knows it is from her victim. She will again prevent the tag from updating tokens in the same way since the authenticator R_i for T_0 is already known to her. Thus the attacker can circularly track the victim.

However, this combined attack also has a high premise that the attacker has to hijack the token update message in every single session when the tag communicates with any legitimate readers. Otherwise she will lose the control of the victim tag, until another physical attack is performed again to obtain the *pad*. This can make the combined attack fruitless.

As we discussed in Section II.B, the best resistance to tampering attacks is to detect it because we can neither prevent attackers physically tampering the tag memory nor stop them from hijacking a message in the open air. The discussed combined attack to our protocol can be detected by tracking the token serial number *s* in each token. The token serial number *s* is a growing version of the token and the *s* of last successful authentication is stored together with each tag's ID in the database. When the backend database detects a tag returns one token with an incorrect serial number *s*, it can inform the reader that the tag has been attacked. In that case the reader can take actions to prevent further tracking by communicating with the tag in a hijacking-resistant way (e.g. using the second channel).

4) Denial of Service Attacks

One potential threat to our token based scheme is the denial of service attacks: an adversary can send a mass of queries to quickly consume the tokens. Although data secrecy is not compromised, this process can disable tags from responding, thus threatening the data availability.

This denial of service attack is very similar to the DoS attacks discussed by Ari Juels in [10]. In [10] Ari argues that DoS is not considered as an active issue because there exists other simple but effective physical ways to achieve comparable DoS attacks. For example, an adversary with an electromagnetic weapon need not resort to breaking down the protocol in order to disable tags, especially if the tags are protected by some countermeasures of unintentional query, as will be explained in the next section.

Though DoS attack is not an active issue, RF-tags can be *unintentionally* scanned by readers that are not

associated with their designated reader. (E.g. a reader of Company A may inadvertently read tags of Company B.) This unintentional query, though not belonging to any forms of attacks, has the same effect as DoS attacks. In our proposed approach, we resort to other schemes to minimize the impact of unintentional query, as discussed below.

One possible solution to unintentional query is to use different query message for different association. For example, in airport luggage flow control systems, different airline companies can query tags with their own query message. In this way a tag will never respond to unintentional query from unknown readers of other companies.

This scheme of association-specific query fulfills the privacy requirement in some application scenarios, but may have potential threat in some other scenarios. For instance, a person carrying 3 books of library A, 2 books of library B and 4 books from library C may suffer from probabilistic tracking: the attacker can send query of library A, B and C to see if another person also return 3, 2, 4 responses of the three libraries respectively. If such responds are found, the attacker can conclude with high confidence that this person is the victim that she wants to track. More detailed discussion of this "set of tags characterizing a person" has been highlighted in [3].

If the unintentional query is inevitable, we should provide means to recover the tag. As we discussed Section III.B.3, the second channel is used to recharge a tag that died from unintentional query. When the tag is running out of token, it will respond "main channel halt" in the main channel and opens the second channel as an interface to recharge the tag with new tokens.

V. Related Work and Discussion

Many cryptographic protocols [2, 6, 9, 10, 14, 15, 17, 24] have been proposed to address the privacy issue. They deploy relatively cheap implementation such as exclusive-OR and one way hash function on tags to achieve secure identity report. Most of the proposed protocols merely address the threat of eavesdropping and physical attacks receive very few considerations. Detailed security comparison with the related works is summarized in Table I.

TABLE I.
Security Comparison with related works

Research Approaches	Resistance of Eavesdropping	Resistance of Message Hijacking	Resistance of Physical Attacks
Hash lock approach [14]	YES	YES	NO
Minimalist cryptography approach [10]	NO	NO	NO
Hash chain approach [15]	YES	YES	Level II Resistance
Our approach	YES	YES	Level III Resistance

A secret key based protocol named hash lock is proposed in [14], which mainly deals with the threat of eavesdropping. In this protocol, every single tag pre-shares a secret with the reader system. When the tag receives a query, it will first generate a pseudo random mask with this secret key, apply the mask to its real ID and send it back to the reader. The reader will use this masked ID to locate a corresponding entry in its backend database and authenticate itself to the tag if necessary. This protocol (and many other following approaches [6, 9, 17, 24]) perfectly resists eavesdropping and message hijacking by using a shared secret to a pseudo random function to generate a response that can not be reversed without knowing the secret. Though some of these approaches have realized the potential threat of physical attack, none of them can resist it.

A token based approach named minimalist cryptography is proposed in [10], which is similar to our scheme. This approach follows a token reuse scheme that a token is not discarded after unsuccessful authentication but only recycled to reuse again. Token reuse scheme does not suffer from token exhaustion due to unexpected queries. The main flaw of this scheme is that an attacker can send a few queries to obtain all the tokens on a tag and use these tokens to track this tag until the tag meets a legitimate reader again. In our proposed scheme, every token is used only once and we propose several methods, as discussed in Section IV.B.4, to handle unintentional normal queries. Due to the token reuse scheme, this minimalist cryptography approach can resist none of the three attacks.

The hash-chain protocol [15] is proposed to achieve Level II resistance to physical attacks. When the tag receives a query, it hashes a shared secret with two different hash functions and replaces the secret with its hashed result. The tag returns the current hashed value on its hash chain back to the reader as an authenticator. The backend database performs the same hash functions on its entries for each tag to find a match and recognize the tag. Though this protocol has scalability problems, a subsequent approach [2] proposed a scheme using time-memory trade-off to improve the performance. The hash-chain scheme can resist eavesdropping, message hijacking and physical attacks at resistant Level II only.

Our research was motivated by the potential threat of physical attacks. We propose an approach to address this issue while resisting eavesdropping and message hijacking, which was achieved by most current schemes. We adopt a token based scheme to provide a means for the database to recognize the tag while minimizing the impact of tracking a tag by this necessarily shared information. Our proposed protocol resists physical attacks and other forms of attacks such as eavesdropping.

In our future work, we will improve the robustness of our protocol. As we discussed previously in Section IV.B.1, "physical read tracking" can compromise k tokens on a tag and then be used to track the victim for k times (This attack is much less serious than other physical attacks that can compromise authentication keys and track tag forever). If the total number of tokens, i.e. k is

small, this attack can be so fruitless that the attacker would be reluctant to perform at the cost of physical attacks. On the other hand, to avoid unintentional queries, k should be relatively large. Otherwise the tokens could be easily used up and frequent second channel retrieval can become a big burden. We maintain k at a relative high value, for example, fifty, to balance the impact of unintentional query and the threat from physical read tracking. Using a true random number generator (TRNG) in a tag could be another way to resist physical attacks, including "physical read tracking" [13]. We will research further on the topic of "physical read tracking". As we discussed in Section IV.A, we also plan to implement our protocol and evaluate its performance.

VI. CONCLUSIONS

The potential violation of user privacy is a big obstacle to the deployment of RFID systems. Many approaches have been proposed to address this problem. Most of current research and proposed security protocols mainly focus on eavesdropping and pay little attention to physical attacks.

In this paper we present our threat model of RFID systems and thorough analysis of physical attacks to RFID tags. Based on the threat model we propose a secure identity reporting protocol to address the possible attacks. In our proposed protocol, the tag responds to readers with pre-stored one-time tokens. The tokens contain the tag's encrypted ID that can only be decrypted by a legitimate reader. The reader sends back dynamically created new tokens to the tag. The new tokens are encrypted by a one-time pad, which is also dynamically updated by the reader. We analyze that our scheme can resist physical attacks, in addition to other security attacks. The performance analysis shows that our protocol is scalable for large RFID systems with huge volume of tags.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for the useful comments. This work was supported in part by the NSF grant 0406325.

REFERENCES

[1] R. Anderson and M. Kuhn, "Tamper Resistance- a Cautionary Note," *Proceedings of the Second Usenix Workshop on Electronic Commerce*, 1996.

[2] G. Avoine and P. Oechslin, "A Scalable and Provably Secure Hash-Based RFID Protocol," *IEEE International Workshop on Pervasive Computing and Communication Security*, 2005.

[3] G. Avoine and P. Oechslin, "RFID Traceability: A Multilayer Problem," *The 9th International Conference on Financial Cryptography*, 2005.

[4] Boycott against clothes with RFID tags, *http://www.boycottbenetton.com*

[5] Boycott RFID enabled products, *http://www.boycotttesco.com*

[6] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," *Workshop on Cryptographic Hardware and Embedded Systems*, 2004.

[7] Government report: "Radio Frequency Identification Technology in the Federal Government," Available at: *http://www.gao.gov/new.items/d05551.pdf*

[8] J. Grand (Kingpin), "Attacks on and Countermeasures for USB Hardware Token Devices," *Proceedings of the Fifth Nordic Workshop on Secure IT Systems*, 2000.

[9] D. Henrici and P. Muller, "Hash-based enhancement of location privacy for radiofrequency identification devices using varying identifiers," *Workshop on Pervasive Computing and Communications Security*, 2004.

[10] A. Juels, "Minimalist Cryptography for RFID Tags," *Security of Communication Networks* (SCN), 2004.

[11] A. Juels, RL Rivest, M. Szydlo, "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy," In *Proceedings of ACM Conference on Computer and Communications Security*, 2003.

[12] A. Juels and R. Pappu, "Squealing Euros: Privacy Protection in RFID-Enabled Banknotes," *Financial Cryptography*- FC03, 2003.

[13] Z. Liu and D. Peng, "True Random Number Generator in RFID Systems against Traceability," *IEEE Consumer Communications and Networking Conference (CCNC'06)*, 2006.

[14] D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures," In *Conference on Computer and Communications Security CCS*, ACM Press, 2004.

[15] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic Approach to a Privacy Friendly Tag," *RFID Privacy Workshop*, 2003.

[16] Protest against RFID, *http://www.spychips.com*

[17] K. Rhee, J. Kwak, S. Kim, and D. Won, "Challenge-Response based RFID Authentication Protocol for Distributed Database Environment," In *Proceedings of International Conference on Security in Pervasive Computing*, 2005.

[18] RF-Dump, *http://www.rf-dump.org*

[19] D. Samyde, S. Skorobogatov, R. Anderson, and J. Quisquater "On a New Way to Read Data from Memory," *first International IEEE Security in Storage Workshop*, 2002.

[20] "Security technology: Where's the smart money?", *The Economist*, pages 69-70. 9 February 2002.

[21] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks," In *International Workshop on Security Protocols*, 1999.

[22] TAMPER Lab, *http://www.cl.cam.ac.uk/Research/Security/tamper*

[23] S. H. Weigart, "Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses," *Workshop on Cryptographic Hardware and Embedded Systems*, 2000.

[24] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," In *Security in Pervasive Computing*, 2003.

[25] J. Yoshida, "Euro bank notes to embed RFID chips by 2005," *EE Times*, 19 December 2001.

[26] X. Zhang and B. King, "Integrity Improvements to an RFID Privacy Protection Protocol for Anti-Counterfeiting," *the 8th Information Security Conference* (ISC'05), 2005.

Zhaoyu Liu is an assistant professor in the Department of Software and Information Systems at the University of North Carolina at Charlotte and director of the Secure Infrastructure and Networking Group (SING). His research interest is in pervasive computing and system security. Liu received a PhD in computer science from the University of Illinois at Urbana-Champaign.

Dichao Peng is a PhD student in the Department of Software and Information Systems at the University of North Carolina at Charlotte and a member of the Secure Infrastructure and Networking Group (SING). He earned a BS in computer science from the University of Electronic Science and Technology of China in 2004. His research interests include pervasive computing, networking and computer security.