Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

PAP: A privacy and authentication protocol for passive RFID tags

Alex X. Liu *, LeRoy A. Bailey

Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824-1266, USA

ARTICLE INFO

Article history:
Received 12 October 2008
Received in revised form 6 January 2009
Accepted 20 March 2009
Available online 31 March 2009

Keywords:
RFID
Privacy
Authentication
Protocol design

ABSTRACT

Passive Radio Frequency Identification (RFID) tags, due to their ability to uniquely identify every individual item and low cost, are well suited for supply chain management and are expected to replace barcodes in the near future. However, unlike barcodes, these tags have a longer range in which they are allowed to be scanned, subjecting them to unauthorized scanning by malicious readers and to various other attacks, including cloning attacks. Therefore, a security protocol for RFID tags is necessary to ensure the privacy and authentication between each tag and their reader. In order to accomplish this, we propose PAP, a privacy and authentication protocol for passive RFID tags. This protocol requires little computation and achieves both privacy and authentication, making it sufficient enough for use in supply chain management; however, this protocol is also suitable for use in other RFID applications as well.

1. Introduction

Radio Frequency Identification (RFID) tags are small electronic components that are used to identify and track objects. They have applications in various fields such as inventory tracking, supply chain management, theft-prevention, and the like. An RFID system consists of an RFID tag (i.e. transponder), an RFID reader (i.e. transceiver), and a back-end database. An RFID reader consists of an RF transmitter and receiver, a control unit, and a memory unit. These instruments work together to transfer and receive information stored on radio waves between it and an antenna attached to an RFID tag. This information interacts with stored items upon a back-end database that some readers are able to connect to. Depending on the type of the tag, they too have the capability to perform different functions with the information transferred from a reader.

There are three broad categories of RFID tags: passive, semi-passive, and active. Passive tags are powered by the signal of an interrogating reader and can only work within short ranges (a few meters). Active tags maintain their internal state and power transmission using a battery. Semi-passive tags are battery assisted tags that use some battery power to maintain their internal volatile memory but may still rely on the reader's signal to power their transmission. They can initiate communication and operate over longer ranges (several meters), but are also more expensive and bulkier than passive tags. Passive tags, however, are also more popular and cheaper. In particular, passive tags are used more often in supply chain management. Therefore, during the rest of this paper, we will be dealing only with passive tags.

RFID tags are able to uniquely identify individual items of a product type, unlike barcodes, which only identify each product type. This is particularly useful when the transaction history of each item needs to be maintained or when individual items need to be tracked. Furthermore, RFID tags do not require line-of-sight reading like barcodes, increasing the scanning process of a tag significantly. Due to the these and other advantages that RFID tags have over barcodes, RFID is increasingly becoming more popular and is expected to replace the current barcode technology in the near future. However, there is also a growing concern among people about consumer privacy protection and other security loopholes that make RFID tags an easy target for malicious attacks. Passive RFID tags in their current form are vulnerable to various types of attacks and thus there is a pressing need to make this technology more secure before it is viable for mass deployment. Therefore, privacy and authentication are the two main security issues that need to be addressed for the RFID technology.

The two primary concerns of privacy with RFID tags are clandestine tracking and inventorying [1]. Clandestine tracking deals with issue of a nearby RFID reader being able to scan any RFID tag, since these tags respond to readers without discretion. Clandestine inventorying on the other hand is a method of gathering sensitive information from the tags, thus gaining knowledge about an organization's inventory. An organization called EPCglobal [2] manages the development of the Electronic Product Code (EPC), a code in RFID tags that is equivalent to the code used to store information in a barcode. EPC compliant RFID

* Corresponding author. Tel.: +1 517 353 5152; fax: +1 517 432 1061.
E-mail addresses: alexliu@cse.msu.edu (A.X. Liu), baile129@cse.msu.edu (L.A. Bailey).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.03.006

tags have fields to store the manufacturer code and the product code that makes it easy to follow the inventory patterns of a store [1].

RFID privacy is already a concern in several areas of everyday life. Here are a few examples. Automated toll-payment transponders, small plaques positioned in windshield corners, are commonplace worldwide. In a recent judiciary, a court subpoenaed the data gathered from such a transponder for use in a divorce case, undercutting the alibi of the defendant [3]. Some libraries have even implemented RFID systems to facilitate book checkout and inventory control and to reduce repetitive stress injuries in librarians. Concerns about monitoring of book selections, stimulated in part by the USA Patriot Act, have fueled privacy concerns around RFID [4]. Lastly, an international organization known as the International Civil Aviation Organization (ICAO) has promulgated guidelines for RFID-enabled passports and other travel documents [5,6]. The United States has mandated the adoption of these standards by 27 "visa waiver" countries as a condition of entry for their citizens. The mandate has seen delays due to its technical challenges and changes in its technical parameters, partly in response to lobbying by privacy advocates. One may see how verification of the information stored upon the passport would also become an issue as well. This brings us to the other security threat in RFID, authentication.

Authentication is another major security issue for RFID tags. Privacy deals with authentic tags being tampered by attacking readers, while authentication deals with valid readers being misled by deceptive tags. One example where authentication would play a useful role is when scanning counterfeit tags. It has been shown that one can rewrite what a tag emits onto another tag, effectively making a clone [1]. Therefore, authentication is as much of a concern as privacy is.

The key challenge in providing security mechanisms to passive RFID tags is that such tags have extremely weak computational power because they are designed to be ubiquitous low cost (e.g., a few cents) devices [7]. Previous solutions have been developed to solve both security threats for RFID tags (such as [8–12] and [13]); however, these solutions are not suitable for passive RFID tags. For example, many protocols (such as [8,9] and [10]) for RFID authentication use heavy duty cryptography. Some previous protocols (such as [11,12], and [13]) address the privacy issues of RFID systems by requiring users to carry a large device on a daily basis, which seems to be impractical.

In order to deal with these issues, we propose PAP, a privacy and authentication protocol for passive RFID tags. Using our PAP protocol, each tag has a secret numeric value, for which a reader and a tag establish authentication. Upon verification of the reader by the tag, the tag sets itself to a state that upon query, only gives an authenticated reader enough information to change the tag to a prior state and release its EPC information. However, the information given in this state is also general enough to not allow an unauthenticated reader to gain access to the EPC code or know what the product is, thereby establishing privacy.

Our protocol is practical and useful for two reasons. First, it requires only an extremely small amount of computation; therefore, it has the capacity to be implemented within passive RFID tags, unlike the cryptography intense protocols in prior work. Second, our protocol deals with both privacy and authentication. This also decreases the overall cost production; but more importantly, it eliminates the need for any extra security devices.

The rest of the paper proceeds as follows. In Section 2 we review related work. In Section 3, we describe our system and threat model. Section 4 presents the details of our protocol. In Section 6, we analyze the security and cost of our protocol. We give concluding remarks in Section 7.

2. Related work

The first approach to dealing with consumer privacy was developed by the company that will oversee the barcode to RFID transfer, EPCGlobal Inc. Their approach is to just "kill" the tag [2]. In other words, the tag will be made inoperable, allowing it not to be scanned by malicious readers. This process is done by the reader sending a special "kill" command to the tag (including a short 8-bit password). For example, after you roll your supermarket cart through an automated checkout kiosk and pay the resulting total, all of the associated RFID tags will be killed on the spot. Though killing a tag may deal with consumer privacy, it eliminates all of the post-purchase benefits for the consumer. One example of these types of post-purchase benefits are items being able to interact with what are being called "smart" machines. For example, some refrigerators in the future will interact with the RFID tags on food items. This will allow the refrigerator to scan what items you normally buy, and once it notices that so many items have been removed over a period of time, it will inform of what items are missing so you may purchase some more. Another example of a "smart" machine would be a microwave. The microwave would scan the RFID tag from the purchased item and automatically set the timer to the correct amount of time needed. From these examples, you can see that killing a tag would not be an appropriate approach to deal with consumer privacy.

Another approach to dealing with consumer privacy involves shielding the tagged item, either by using radio wave blocking materials or scrambling any outgoing signals from the tag. The first approach is better known as a Faraday Cage [1], a container made of metal mesh or foil that is designed to block certain radio frequencies and is often used by criminals as a method of sheilding an item to surpass shoplifting detection systems. This approach however only partially succeeds in establishing privacy, as its contents are not designed to fit over uniquely shaped or larger items such as wrist-watches, computers and televisions. The latter approach is also known as the active jamming approach [1]. This approach will allow the consumer to carry a device that would block nearby RFID readers by transmitting or broadcasting its own signals. This approach is dangerous however, for if the broadcast signaling power of a jammer is too high, it might cause the jammer to interfere with surrounding legitimate RFID readers.

One of the more effective approaches to providing consumer privacy deals with an idea proposed by Juels consisting of a "privacy bit". The technique proposed in [14] uses a privacy bit in tags that can take a value of 0 or 1 and can be easily toggled by a reader after authenticating with a unique pin for that tag. While inside a store, a tag's bit value is 0, indicating public access to a tag's identification. While during checkout, this value is changed to 1, denoting the tag is about to enter a location with restricted access. In order to establish privacy while in this state, the tag must interact with another tag known as the "blocker tag" [14]. Depending on the amount of privacy a consumer may need, the blocker tag will manipulate the query result of a normal tag by scrambling the bits of all tags within range (known as the full blocker) or only certain tags determined by their privacy bit (known as soft blocking [15]). Either way, the tag is secure only in the presence of the blocker tag. Our method borrows the idea of the privacy bit but does not require the presence of any additional specialized tag to safeguard the original tag.

In dealing with authentication, there have been a few hash-based protocols developed due to the low processing power of a passive tag. One is the HashLock scheme [16]. In this scheme, each tag carries key K and its hashed value $h(K)$, better known as its *metaID* value. Upon query by the reader, a tag will respond with its *metaID*, which is forwarded to the reader's back-end database.

Assuming this is a valid tag, the database will recognize it from its *metaID* and will send back the corresponding secret key K of the tag to the reader, where it will continue to forward this value to tag. The tag will then proceed to validate the reader by hashing the received value and if it matches the stored *metaID*, it will unlock itself, allowing its EPC information to be received. This protocol has two major drawbacks. First, an attacker can eavesdrop $h(K)$, which is sent in the air, and make a fake tag that simply emits $h(K)$, which consequently can be authenticated to an authentic reader. Second, a tag can be tracked by its *metaID*, which violates consumer privacy. To prevent a tag being tracked, the same authors of [16] proposed a randomized version of the scheme where the response of a tag changes in every query. In particular, upon a query, the tag generates a random nonce r and computes the hash $h(ID, r)$. Then, the tag sends both r and $h(ID, r)$ to the reader for verification. When the reader receives r and $h(ID, r)$, the reader computes $h(ID_i, r)$ for every ID_i. The authentication is successful if and only if there exists ID_i such that $h(ID_i, r) = h(ID, r)$. This protocol has two major drawbacks. First, an attacker can still eavesdrop r and $h(ID, r)$, which is sent in the air, and make a fake tag that simply emits r and $h(ID, r)$, which consequently can be authenticated to an authentic reader. Second, this protocol can be extremely inefficient when the number of possible *ID*s that the reader needs to check is large.

The work that is closest to our is the recently proposed RFIDGuard protocol in [17]. The similarity between RFIDGuard and PAP protocols is that they both deal with the privacy and authentication of passive RFID tags and they both focus on inventory control applications. The difference between these two protocols is that the RFIDGuard protocol is designed for RFID tags with extremely weak computational capabilities (e.g., cannot compute secure hash functions) and therefore the RFIDGuard protocol provides weak security guarantees. In comparison, the PAP protocol is designed for RFIDs that are capable of computing secure hash functions and provides stronger security guarantees than the RFIDGuard protocol.

3. Modeling

3.1. System modeling

In this section, we specify the security properties that we want our PAP protocol to achieve. We begin by describing our assumptions regarding the readers and tags being used. We then discuss the assumptions and limitations of attacks upon our tags.

3.1.1. Readers and tags

The two principal parties involved in this protocol are *readers* and *tags*. We assume the existence of both authorized tags and malicious tags. There are three types of authorized readers in our protocol: *inventory readers, checkout readers, and return readers*. An inventory reader is the most basic reader of these three, only allowing the ability to query the tag. A checkout reader contains all the functions of an inventory reader as well as the ability of connecting to a back-end database. The information retrieved from the back-end database could be used by the checkout reader to authenticate itself to a tag. A return reader has the same functionality as a checkout reader.

The tags that we deal with in this paper are Class 1 Generation 2 tags, where were standardized by EPCglobal [2] in 2004 for passive RFID tags. This global standardization has been adopted by US Department of Defense, Walmart, Metro AG, etc. [7]. Class 1 Generation 2 tags have four memory banks: Reserved Memory Bank (which as at least 32 bits for storing information such as the password for killing a tag), EPC Memory Bank (which as at least 496 bits for storing EPC information), TID Memory Bank (which as at least 32 bits for storing tag identifier), and User Memory Bank (for storing information related to the tag's application). Note that the upper limit of the user memory bank in a tag is not specified in the standard. In other words, the size of the user memory bank of a tag depends on the amount of memory that the manufacture puts on the tag. Our protocol requires a small amount of memory, which could be allocated from the user memory bank of a tag. Our protocol only requires a tag to perform four simple operations: comparing two numbers, execute a hash function, storing and retrieving a number in user memory bank, and flipping a bit. These operations could be easily implemented on Class 1 Generation 2 tags.

Note that we do not consider how the reader will distinguish between multiple tags because this is handled by singulation protocols [18] and it is out of the scope of this paper.

3.1.2. Security and privacy requirements

Our PAP protocol strives to achieve two requirements: authentication and privacy. In terms of authentication, a tag and a reader should be able to achieve mutual authentication, that is, a tag should be able to authenticate a reader and a reader should be able to authenticate a tag. In terms of privacy, a tag should only give out private information to authorized readers.

3.2. Threat modeling

Previous research has some assumptions on practical attacks on RFID systems, a small subset of which we entail into our protocol. First, due to the relatively short transmission range (i.e., several meters) of a tag, a malicious reader cannot eavesdrop the reply from a tag. Also, it is not easy for an attacker to hide himself between a legitimate reader and a tag in an active session due to the distance between a tag and a reader. Another security assumption suggests that it is not easy to intercept a message and modify the message over the air in real time. These three assumptions are made due to the fact that all authentication procedures will take place inside a retail store. Therefore, we assume that a retail store has some security mechanisms that prevent unauthorized readers from entering the store. This can be easily achieved by installing detection devices near the entrance of the store to detect unauthorized readers [18]. Lastly, we assume that an attacker has two major abilities: the ability to query a tag as a normal reader and the ability to clone a tag. We also assume that it is difficult to intercept a message and modify the message over the air in real time.

4. The PAP protocol

In the PAP protocol, each tag attached to a product stores (1) a secret key k shared by both the reader and the tag, (2) a generic name (i.e., the numeric representation of the product type), (3) an ID (i.e., the EPC code, which is the numeric representation of the individual item), and (4) a privacy bit, where value 0 indicates that the tag is in the non-privacy state (i.e., in store) and value 1 indicates that the tag is in the privacy state (i.e., out store). In order to achieve authentication between the tag and the reader, the tag first sends its ID (or generic name) and a random nonce to the reader upon query. The reader uses this information to determine the secret key k of the tag and applies a one-way hash function upon it, sending both the hashed result and another random nonce to the tag. The tag verifies the reader by performing the same hash function using its secret key k with the nonce sent to the reader. If this value matches the hashed result sent from the reader, the tag authenticates the reader. The tag will then perform another hash function using its secret key k with the nonce received from the reader and send this hashed value to the reader. The reader then performs the same hash function with its secret key k. If the result matches, the reader authenticates the tag. For the hash function, we can use the HMAC-MD5 [19] hash function with 128-bit keys.

In order to establish privacy, upon checkout, the privacy bit of a tag is changed from 0 to 1. At any point that the tag's privacy bit is 1 and a reader attempts to scan it, the tag will only return enough information for a trusted reader to perform the authentication procedure mentioned above, which only includes a number to represent its generic name. Since an unauthorized reader would not contain the secret key k, the tag will not give out its private information.

Next, we present the PAP protocol based on four different locations: inside a store, at a checkout counter, at a return counter, and outside a store.

4.1. In-store protocol

The in-store protocol concerns querying a tag located inside a store. We assume there is an established level of security that does not allow unauthorized RFID readers within a scanning range of these tags; therefore, the in-store protocol is designed to provide no authentication and privacy protection for efficiency purposes. Each tag when delivered to the store will have its privacy bit set to zero, denoting a location containing only authorized readers. Upon a reader querying a tag, the tag will send the reader its ID and a random nonce n_t. Fig. 1 illustrates this in-store protocol.

Though the reader would not need any more information beyond the tag's ID at this time, the random nonce generated by the tag is sent by default to lessen the cost of the tag. If it were to just send its ID, the tag would have to be programmed to know when to send additional information (e.g., the difference between a checkout reader and an inventory or price checking scanner), further increasing the cost of the tag.

4.2. Checkout protocol

The checkout protocol concerns querying a tag during a checkout procedure. To prevent the use of cloned tags, the checkout protocol allows the reader to authenticate the tag. To ensure that the proper type of reader is used during the checkout procedure, the checkout protocol also allows the tag to authenticate the reader as well. As previously mentioned in Section 3, different types of readers exist in the store; therefore, a tag always sends the random nonce n_t in the in-store protocol to save cost. Other readers beyond the checkout console should not have the ability to connect to the database that contains the secret key k associated with the product in order to fulfill the authentication requirements for this protocol. If a tag does not authenticate the reader, an employee with a hand-held reader could checkout any product and steal from the store.

The checkout protocol works as follows. The first two steps are the same as the two steps in the in-store protocol. In the third step, the reader retrieves the secret key k of the tag from its back-end inventory database using the EPC Code, *ID*, received in the second step. The reader will then perform a one-way hash function on this k and the random nonce, n_t, received from the tag. The reader then generates its own random nonce, n_r, and sends it along with the hash result, $h(n_t, k)$, to the tag. Because the tag knows key k, it can verify whether the hash result received from the reader is valid. Note that an unauthorized reader does not know the value of key k associate with the tag, and is not be able to compute $h(n_t, k)$. If the tag successfully authenticates the reader, the tag sets its privacy bit from zero to one, denoting the tag's traversal to a location that may contain unauthorized readers. The tag then computes $h(n_r, k)$ and sends the result back to the reader. The reader authenticates the tag by verifying the validity of the hash result $h(n_r, k)$ received from the tag. Fig. 2 illustrates this checkout protocol.

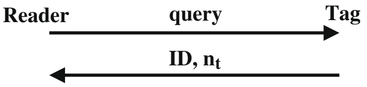

Fig. 1. The in-store protocol.

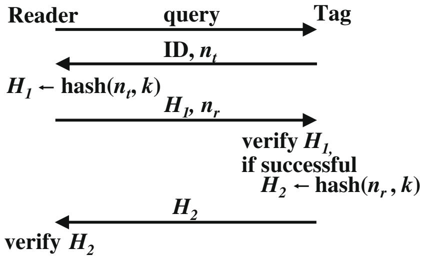

Fig. 2. The checkout protocol.

There are different levels of security obtained by using this checkout procedure. Only an authentic tag and an authorized reader would know the value of the secret k for a tag. Therefore, if a cloned tag does not contain the correct value for k, it would not send the correct hash result to the reader at the end of the procedure, causing the reader fails to authenticate that tag. If the reader does not verify a tag within a time limit, the system will timeout and not allow the customer to finish the checkout procedure. The second level of security deals with the random nonce sent by both the reader and the tag in this process. In order to reduce the chances of a replay attack, random numbers are hashed along with the value of the secret key k.

Before continuing to our next protocol, we need to discuss the *cover-coding* mechanism that has been standardized for Class 1 Generation 2 tags. As described in Section 3, the signal transferred by a Class 1 Generation 2 tag is only up to few meters; however, the signal from a reader could travel as far as one kilometer [7], allowing the information sent from a reader to a tag to be eavesdropped by an attacker who may be out of sight. In order to prevent this, each Class 1 Generation 2 tag incorporates the mechanism of *cover-coding*. In this procedure, when a reader queries a tag, the tag first generates a 16-bit random number and sends it to the reader. Note that this random number only travels a few meters. In the subsequent communication between the reader and the tag, all messages are XORed with the random number. Therefore, as long as attackers are not physically within a few meters, they cannot decode the messages sent out from the reader. Based on the cover-coding mechanism, in the last two steps of our check-out protocol, each message sent between the reader and the tag are not in plain text, rather, they are XORed with the random number that they established for that session. In essence, the cover-coding mechanism uses the widely known concept of one-time pad.

4.3. Out-store protocol

The out-store protocol resembles a tag's behavior once it leaves the store. At this point, various readers with different levels of security are assumed to be able to access the tag. Therefore, only enough information about the tag is given to allow authenticated readers access in order to flip the tag's privacy bit back to zero, which includes the tag's generic name and a random nonce. Since the tag's generic name is represented by a number, an unautho-

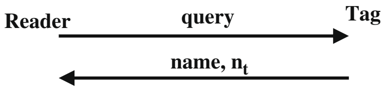

Fig. 3. The out-store protocol.

rized reader will not know what items are currently being read. The next section will explain how the information being given above will allow an authenticated reader to turn the privacy bit of a tag back to zero. Fig. 3 illustrates this checkout protocol.

Note that there are many reasons that an attacker would want to retrieve a tag's private information. For example, an attacker may want to know what people shop for in certain stores to develop spam or other similar shopping techniques. Also, an attacker may want access to the private information of a tag to gather secrets about the product's producer or the store in general. This type of attack is prevented in our out-store protocol.

4.4. Return protocol

The return protocol deals with the returning of an item to where it was sold. Many stores have returned items that they are still able to resell; therefore, these RFID tags need to be reset for resale. The return protocol requires mutual authentication between a tag and a reader as well. To prevent unauthorized readers from flipping the privacy bit of a tag from one to zero, the tag needs to authenticate the reader. Though this concept may appear clear, it may not be as easy to understand why the tag needs to be authenticated. If the tag were not authenticated, a person could create a counterfeit tag to indulge the price value of an item. This in turn would allow a customer to increase the price of an item, enabling them to receive a higher amount of money back or exchange the item for a higher valued one.

The return protocol works as follows. The first two steps are the same as the two steps in the out-store protocol. In the third step, the reader retrieves the secret key k of the tag from its back-end inventory database using the *name* received from the tag. The reader will then perform a one-way hash function on this k and the random nonce, n_t, received from the tag. The reader then generates its own random nonce, n_r, and sends it along with the hash result, $h(n_t, k)$, to the tag. Because the tag knows key k, it can verify whether the hash result received from the reader is valid. Note that an unauthorized reader does not know the value of key k associate with the tag, and is not be able to compute $h(n_t, k)$. If the tag successfully authenticates the reader, the tag sets its privacy bit from zero to one, denoting the tag's traversal to a location that may contain unauthorized readers. The tag then computes $h(n_r, k)$ and sends the result back to the reader. The reader authenticates the tag by verifying the validity of the hash result $h(n_r, k)$ received from the tag. Fig. 4 illustrates this checkout protocol.

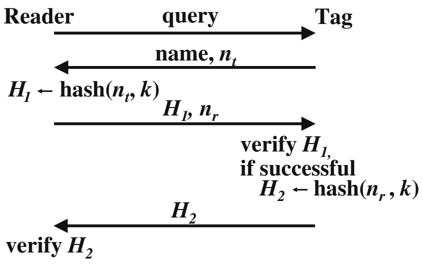

Fig. 4. The return protocol.

5. Example scenario

Let us consider a practical example of how our approach would work in a typical supermarket with items having RFID tags on them. Fig. 2 depicts the working of this scenario. When an item is inside the store, the privacy bit is Off(0). This enables the store owners to keep track of their inventory. During checkout, an RFID scanner scans the item and since the privacy bit is off, the tag emits the full information (ID, first pseudonym and current index) and increments its index value. If the product purchased is a TV, then the tag's ID has a unique number for that particular TV and the list of pseudonyms point to the product type (Television). The reader looks up the back-end databases using the pseudonym just received and gets the next pseudonym for TVs. The reader transmits this pseudonym, as a PIN to authenticate itself to the tag. The tag has logic to compare its current pseudonym with what is received. If there is a match, then the privacy bit is turned ON and the next pseudonym in the tag's list is returned. This is done to acknowledge the fact that the privacy bit is now set and also to inform the reader that the tag is genuine. (A fake tag cannot do this successfully.) The item has now entered a private zone and keeps the item protected. Once ON, the tag when interrogated, emits only the first pseudonym coupled with the index number of the pseudonym it is currently on. This makes it hard for any attacker to know what the current pseudonym is. Note that each read of the tag increments the index value in a cyclic order. This ensures privacy, although at a weak level, and also makes tracking difficult since the value returned each time is different. If the product needs to be returned to the store or if a rescanning is required at a different location of the store, then the privacy bit needs to be turned OFF which is done in the following way. A valid reader interrogates the tag and gets the base (first) pseudonym and the offset (index) in the table of pseudonyms for that product type. The database is looked-up to find the next pseudonym and this is sent to the tag as an authenticating PIN. The logic in the tag, as before, compares the incoming pseudonym with its current pseudonym and toggles the privacy bit to OFF, on a match. Once the privacy bit is OFF, the tag is open again and can be taken back into the store.

6. Security and computational analysis

In this section, we analyze the security that our PAP protocol provides and the cost for providing such security.

6.1. Security analysis

As stated in Section 3, we assume that a retail store has a measure of preventing unauthorized readers from entering the store. When this assumption is perfectly held, our PAP protocol perfectly prevents attackers from cloning a tag and a reader. For cases in which this assumption may not hold, our protocol would still prevent the cloning of both a tag and a reader because the value of the secret key k is never sent over the air. The implication of not holding this assumption is that an unauthorized reader can eavesdrop the ID sent from the tag to the reader in a store. However, because of the short range that a tag can transmit information, holding an unauthorized reader within a short distance (e.g., a few meters) from an authentic reader could not last a long period of time without being noticed by the store.

Another security measure in the PAP protocol deals with the operation of the privacy bit. As explained in Section 4, this bit is toggled between numbers 0 and 1 to establish privacy for the tag by only releasing a limited amount of information in non-secure areas. To have more flexibility, we introduce another method, which we call *privacy guard*. In this method, we replace the privacy bit by two bits. Three of the four values would each contain a different level of privacy that the consumer can decide upon purchasing a product. The tag would be initially set to (00), denoting a secure place since its initial state is within the store, as the (0) state in the privacy bit. The second state of the tag is (11), denoting the tag maybe in a non-secure location, allowing it to only release enough information to return the item, as the (1) state in the privacy bit. There is a special state however, (01), which releases some general information about the tag, though the tag may still be in a non-secure location. A consumer may want to use this option if they have a "smart" machine at home, as described in Section 2. Though this option may be ideal for the more technological savy consumers, it still raises the security concern in which a third party may obtain some information about the products one is holding.

The security of the PAP protocol also relies on the one-wayness and collision resistance properties of the hash function used in the protocol. The one-wayness property implies that given $hash(x,k)$ it is computationally infeasible to compute x and k. The collision resistance property implies that it is computationally infeasible to find two distinct numbers x and y such that $hash(x,k) = hash(y,k)$ for any key k. Examples of such hash function are HMAC-MD5 and HMAC-SHA1 [19–21]. Using such secure hash functions, an attacker cannot compute k from $hash(n_t,k)$ and cannot find another $k'(\neq k)$ such that $hash(n_t,k') = hash(n_t,k)$.

6.2. Computational analysis

The PAP protocol deals with passive RFID tags, which have very limited computational power. There are only three requirements that the PAP protocol requires for each tag. This includes the ability to execute a secure hash function, to compare two numbers, and to generate a random number. Most passive RFID tags have the ability to execute secure hash functions [16]. Comparing two numbers are easy for passive RFID tags. Note that our protocol does not require an additional random number generator but rather uses one previously built within the tag. As explained in Section 4, each tag uses the cover-coding method to disguise the communication between the tag and reader by XORing each message with a random 16-bit number. The random number generator for cover-coding can be used for our protocol as well.

The computational and space complexity of the PAP protocol suite for both a tag and a reader is $O(1)$. Recall that a tag stores only a secret key, a generic name, an ID, and a privacy bit. A reader does not permanently store any information. The back-end database needs to store the information for every item. Let n be the total number of items. The space complexity of the back-end database is $O(n)$ and the computational complexity of finding the secret key of a particular item depends on the database query engine, which is typically $O(\log n)$.

7. Conclusions

In this paper, we present PAP, a privacy and authentication protocol for passive RFID tags. Our approach uses a passive RFID tag that stores a numeric value in which tags and readers are authenticated through the verification of hash function results and a privacy bit that can be toggled to move the tag to a secure zone. The information given out by a tag when queried by an RFID reader depends on the value of the privacy bit. Besides showing the details of our PAP protocol, we illustrate some common attack scenarios like clandestine scanning, inventorying and counterfeiting and how our protocol would provide security under those circumstances. We give an example application scenario in which our protocol takes place. We also present detailed security and computational analysis of our protocol. Our protocol is both secure and efficient. Although we presented our protocols in the context of supply chain management, adaptation of our protocol can be used for other applications as well.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions on improving the presentation of this work. The work of Alex X. Liu is supported in part by the National Science Foundation under Grant No. CNS-0716407.

References

[1] A. Juels, RFID security and privacy: a research survey, IEEE Journals on Selected Areas in Communications 24 (2) (2006) 381–394.
[2] EPCglobal, EPCglobal website, Available from: <http://www.EPCglobalinc.org/>.
[3] S. Stern, Security trumps privacy, Christian Science Monitor (2001).
[4] D. Molnar, D. Wagner, Privacy and security in library RFID: issues, practices, and architectures, in: B. Pfitzmann, P. McDaniel (Eds.), Proceedings of the ACM Conference on Computer and Communications Security, 2004, pp. 210–219.
[5] International Civil Aviation Organization ICAO, Document 9303, machine readable travel documents (MRTD), part i, Machine readable passports.
[6] A. Juels, D. Molnar, D. Wagner, Security and privacy issues in e-passports, in: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm), 2005, pp. 74–88.
[7] G. Barber, E. Tsibertzopoulos, B. Hamilton, An analysis of using EPCglobal class-1 generation-2 RFID technology for wireless asset management, in: Military Communications Conference, vol. 1, 2005, pp. 245–251.
[8] M. Ohkubo, K. Suzuki, S. Kinoshita, Cryptographic approach to "privacy-friendly" tags, in: RFID Privacy Workshop, MIT, MA, USA, 2003.
[9] S. Bono, M. Green, A. Stubblefield, A. Juels, A. Rubin, M. Szydlo, Security analysis of a cryptographically-enabled RFID device, in: USENIX Security Symposium, USENIX, Baltimore, Maryland, USA, 2005, pp. 1–16.
[10] J. Wolkerstorfer, Is elliptic-curve cryptography suitable to secure RFID tags? Handout of the Ecrypt Workshop on RFID and Lightweight Crypto (July 2005).
[11] C. Floerkemeier, R. Schneider, M. Langheinrich, Scanning with a purpose: supporting the fair information principles in RFID protocols, in: Proceedings of the Second International Symposium on Ubiquitous Computing Systems, 2004.
[12] M. Rieback, B. Crispo, A. Tanenbaum, RFID guardian: a battery-powered mobile device for RFID privacy management, in: Proceedings of the Australasian Conference on Information Security and Privacy, 2005, pp. 184–194.
[13] A. Juels, P. Syverson, D. Bailey, High-power proxies for enhancing RFID privacy and utility, in: Workshop on Privacy Enhancing Technologies – PET 2005, Dubrovnik, Croatia, 2005.
[14] A. Juels, R.L. Rivest, M. Szydlo, The blocker tag: selective blocking of RFID tags for consumer privacy, in: Proceedings of the 10th ACM Conference on Computer and Communication Security, 2003, pp. 103–111.
[15] A. Juels, J. Brainard, Soft blocking: flexible blocker tags on the cheap, in: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, 2004, pp. 1–7.
[16] S.A. Weis, S.E. Sarma, R.L. Rivest, D.W. Engels, Security and privacy aspects of low-cost radio frequency identification systems, in: Proceedings of the International Conference on Security in Pervasive Computing, 2003, pp. 454–469.
[17] A.X. Liu, L.A. Bailey, A.H. Krishnamurthy, Rfidguard: a lightweight privacy and authentication protocol for passive rfid tags, Journal of Security and Communication Networks, accepted for publication.
[18] T. Li, R. Deng, Vulnerability analysis of emap-an efficient RFID mutual authentication protocol, in: Proceedings of the International Conference on Availability, Reliability and Security, 2007.
[19] H. Krawczyk, M. Bellare, R. Canetti, Hmac: keyed-hashing for message authentication, RFC 2104.
[20] R. Rivest, The md5 message-digest algorithm, RFC 1321.
[21] D. Eastlake, P. Jones, Us secure hash algorithm 1 (sha1), RFC 3174.