RFID Security without Extensive Cryptography

Sindhu Karthikeyan and Mikhail Nesterenko*
Computer Science Department
Kent State University
Kent, OH, USA
{skarthik, mikhail}@cs.kent.edu

ABSTRACT

A Radio Frequency Identification Device (RFID) allows effective identification of a large number of tagged objects without physical or visual contact. RFID systems are a promising technology for supply chain management and inventory control. As individual item tagging becomes a reality, privacy concerns over RFID use come to the fore. The shared radio medium allow eavesdropping and unauthorized tag reading which poses threats to individual's privacy. Moreover, due to the mode of use of RFIDs, new threats emerge. For example, an intruder may be able to track the movement of an individual by repeatedly querying an RFID attached to the item that this individual carries. The limited size and cost considerations do not allow to implement conventional cryptographic systems on RFIDs. In this paper we propose an efficient RFID tag identification algorithm that incorporates reader-authentication. Our algorithm is secure against the anticipated threats to RFID systems. Our algorithm does not require computationally expensive cryptographic mechanisms, it relies on rather simple matrix multiplication. To further enhance the utility of our algorithm we propose a scheme that allows for the algorithm to carry out secure identification of multiple tags simultaneously.

Categories and Subject Descriptors

C.2.0 [**Network Architecture and Design**]: Security and protection (e.g., firewalls); C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Security, Algorithms

*This research is supported in part by DARPA contract OSU-RF#F33615-01-C-1901 and by NSF CAREER Award 0347485

SASN'05, November 7, 2005, Alexandria, Virginia, USA.

Keywords

RFID, privacy, security

1. INTRODUCTION

RFID systems are a novel technology with a large number of applications. An RFID system consists of a tag, a reader and a database. An RFID tag is a miniature electronic circuit (containing between 500 and 5000 gates [2]) that is capable of elementary information storage, processing and radio communication. An RFID reader is a device that is designed to identify the tag. The reader is connected to the database that contains additional information about the tag and the tagged item.

RFID systems reduce the time and cost of processing tagged items. These savings have led to the broad acceptance of such systems. Wal-Mart stores use RFID tags for tracking and maintaining their inventory. Boeing and Airbus plan to use RFID tags to simplify identifying and tracking the airplane parts.

Most current RFID applications use pallet/crate tagging. Individual item tagging enhances the utility of RFID systems. For example, libraries can use RFID tags to track books [9]; toll booths can automatically collect toll by inspecting a tag attached to the windshield of a car. However, privacy concerns hamper the widespread adoption of the technology. Radio is a shared medium: it is easy for an intruder to either eavesdrop on the communication between the tag and the reader or query the tag without authorization.

Scarce computational and storage capabilities of the tag make designing security systems for RFID challenging. For example, the use of extensive cryptography-based authentication or high-quality random numbers on the tag-side may not be possible. Extensive cryptographic operations can be shifted to the reader-side. However, this requires the tag to either store large keys or frequently communicate with the reader over a secure out-of-band channel to obtain authorization information. The former option is impractical due to limited tag-side storage; the latter one decreases the utility of an RFID system as a time and cost saving identification technology.

Scalability is an additional concern that an RFID security system designer has to address: the reader should be able to identify multiple tags that share the same radio channel.

RFID-specific security threats. Note that due to the way the RFID systems are used, intruder may obtain sensitive information about individuals even without learning

any of the encrypted data. Molnar and Wagner [9] identified that RFID use has the following potential types of security threats: tracking, hotlisting and profiling. The intruder can *track* the movement of the tag holder by periodically querying the tag or eavesdropping on the communications between the tag and the reader. Notice that the intruder can track a particular tag without identifying the tag itself: as long as the intruder is able to match the tag across multiple identification sessions, the intruder may reconstruct the itinerary of the tag. The intruder may *hotlist* a certain list of items of particular interest and then single out the individuals in possession of these items. Alternatively, the intruder may *profile* an individual by learning what items the individual has in his possession.

The question that we address in this paper is the design of a secure and scalable tag identification algorithm that tolerates tag-side scarcity of resources.

Related work. A number of publications discuss the security of RFID systems. Juels et al [8] consider erasing the information from the tag after it has been scanned. However, this does not allow repeated use of the tag on the same item and thus limits the utility of the technology. Alternatively, they propose to use a special blocker tag that selectively prevents certain tags in its vicinity from being read. Juels and Brainard [6] propose a similar soft blocker tag scheme. However, a blocker tag approach requires the user to carry and manipulate the blocker which may not be practical.

Juels and Pappu [7] describe a privacy protection scheme for RFID-tagged European banknotes. The tags are to carry cryptographically encrypted serial numbers of the banknotes. However, their scheme is vulnerable to RFID-specific attacks, such as tracking of an individual by the contents of the tags he carries. Golle et al [3] propose an elegant security infrastructure. Security agents deployed throughout the area of RFID use are to modify the data carried by the tag. These agents are able to re-encrypt this data without either learning its contents or jeopardizing its safe retrieval. However, the tag allows arbitrary reader to access the stored data. Thus, the proposed infrastructure does not completely eliminate the tracking threat.

A few RFID security schemes [1, 4, 9, 11, 12] employ tag-side cryptographic encryption, random numbers or cryptographic hash. Due to tag size limitations, such operations may not be available. Juels [5] describes a one-time-pad security scheme. After a fixed number of authentication sessions, the pad is to be either reused or replaced through an out-of-band secure communication channel. Due to limited memory resources on the tag, this scheme either requires repeated pad reuse, which undermines security guarantees; or frequent use of the out-of-band channel, which limits the practicality of the scheme.

Juels at al [8] address the issue of secure identification of multiple tags. They propose to consider the identifiers of the tags as leaves of a binary tree. The reader descends the tree depth-first to identify individual tags.

Our contribution. We propose a tag identification algorithm. It is based on matrix multiplication and does not involve either extensive cryptographic operations or random number generation. Our algorithm is secure against known-ciphertext attacks. It is also secure against the RFID-specific attacks. The tag-side storage and computation requirements for the algorithm are rather modest. The algorithm can be implemented on currently available RFIDs. We propose a multiple tag sequencing scheme that extends our algorithm so that the reader can handle simultaneous identification of multiple tags over the shared radio channel.

The rest of the paper is organized as follows. In Section 2 we describe our tag identification algorithm. We describe how multiple tag sequencing can be used to extend our algorithm in Section 3. In Section 4 we conclude the paper by discussing implementation aspects and further extensions of our algorithm.

2. SECURE TAG IDENTIFICATION

Problem specification. An RFID system contains two principals: a tag and a reader. The *tag* is a device attached to a certain item. The tag is resource constrained. It is, however, capable of storing a limited amount of data and performing elementary operations such as byte-size integer addition and multiplication. The tag is capable of running a timer. The *reader* has sizable computational facilities and access to a database for fast lookup and update of the information related to the tag and the tagged item. The *intruder* is an entity who tries to compromise the RFID system. The objective of the intruder may be to directly identify the tag. Alternatively, the intruder may attempt to track, blacklist or profile the tagged item.

The tag and the reader communicate over an insecure channel (radio). All the information exchanged over this channel is available to the intruder. The intruder, however, can gain access to neither the database records nor to the internal memories of the tag or the reader.

Algorithm description. Each tag stores two square $p \times p$ matrices: M_1 and M_2^{-1}. The reader maintains another two matrices: M_2 and M_1^{-1} of the same size. The matrices M_1^{-1} and M_2^{-1} are the inverses of M_1 and M_2 respectively. The tag and the reader also share a key K which is a vector of size q, where $q = rp$. Factor r is an integer. The matrices and the key are randomly chosen per each tag.

As a slight abuse of notation we denote $A = MB$, where M is a $p \times p$ matrix and B is a vector of size q, a component-wise multiplication of M and B. That is, each p-element component A_i of vector A, where $1 < i < r$, is obtained by multiplying M and the following elements of B: $b_{p(i-1)+1}, \cdots, b_{pi}$. Also, we assume that in our calculations the vector is always properly transposed so as to be compatible with the matrix.

Key K is selected such that product $X = M_1 K$ is unique for each tag in the system. The tag information stored in the reader's database is indexed by X. A fresh key is used for every identification session.

The identification session has two parts: the tag identification proper and reader authentication. A complete session of the algorithm is shown in Figure 1. At first, the tag is identified by the reader. The reader initiates the session by contacting the tag. The tag replies with $X = KM_1$. After replying, the tag starts a timer. Product X uniquely identifies the tag. Thus, when the reader receives X, the reader can obtain the rest of the information about the tag and the tagged item from its database.

In the second phase, the reader authenticates itself to the

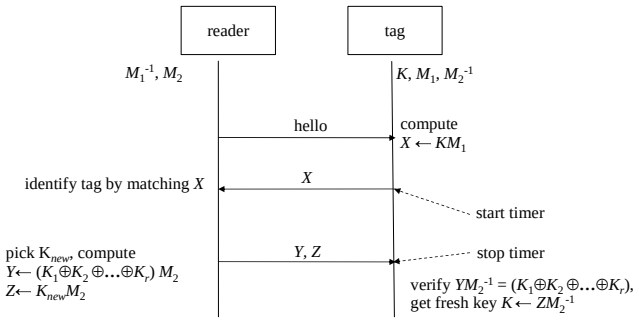

Figure 1: Secure tag identification algorithm.

tag and supplies it with a new key. For authentication, the reader proves to the tag that it is in possession of the key. To save tag resources, rather than sending the whole key back to the tag, the reader uses exclusive OR bitwise on the p-size components of K and multiplies the result by M_2. To calculate a fresh key, the reader selects unique X_{new} and obtains the key as $K_{new} \leftarrow X_{new} M_1^{-1}$. The reader sends both vectors to the tag. The tag verifies the reader's credentials and accepts the new key. In case the reader authentication fails or the reader fails to respond before the timeout expires, the tag stops further communication until reset. The tag is allowed to participate in only one authentication session at a time.

Security discussion. The security of our algorithm is based on the difficulty of recovering the multiplicand or multiplier from the product of matrix multiplication [10]. Hence, the intruder cannot discover the key or the matrix used by the tag and the reader. This prevents the intruder from identifying the tag. Observe that the algorithm is only secure against known-ciphertext attacks. However, we assume that such guarantee is sufficient for RFID systems.

Let us consider the security of our algorithm against the RFID-specific threats. Since the intruder cannot identify the tag, the intruder cannot mount either a hotlisting or profiling attack. The tracking threat is more sophisticated as the intruder does not have to identify the tag to succeed. Notice however, as the intruder cannot deduce either the key or the matrices, he cannot authenticate himself to the tag. Thus, any identification session with the intruder is aborted.

The tag does not participate in multiple authentication sessions, neither does it respond to identification requests after an unsuccessful session. Thus, there may be at most one aborted session per tag. Observe that during each session, including the single aborted session, the tag and the reader send data based on a fresh key. Since the intruder cannot decode the transmission, he cannot match the tag across multiple sessions. Hence, the intruder may not be able to track the tag.

Notice we assume that the intruder is not capable of matching multiple authentication sessions of the same tags through non-radio means (for example by observing the tagged objects). In conclusion we discuss how this assumption can be lifted.

3. MULTIPLE TAG SEQUENCING

Observe that the tag identification algorithm assumes that the reader and the tag use the radio channel exclusively. In practice, multiple tags may potentially share the channel. However, the tags do not have sophisticated channel arbitration capabilities.

In this section we discuss the scheme that augments our tag identification algorithm to enable the reader to communicate with multiple tags. The main change is in the identification phase of the algorithm. Recall that in this phase the reader obtains the key from the tag. In the multiple-tag version, the reader learns the keys of all the tags present. Moreover, each tag learns its key's position in the order (e.g., ascending) of the keys of the tags participating in the identification session. We call this scheme *multiple tag sequencing*. Once the tag knows its position, the second phase of the identification algorithm can proceed sequentially. The reader broadcasts the messages for the tags in the order of their keys. Each tag receives the message sent specifically to it and ignores the rest.

We assume that each tag is capable of broadcasting its key bit-by-bit. If multiple tags broadcast the same bit — 0 or 1 simultaneously, the reader is able to receive the bit successfully. If some tags broadcast 0 and others — 1, then all tags and the reader sense a message collision [8]. In case the tags are incapable of sensing the collision on their own, the reader has to notify the tags if the collision has occurred.

Reader-side sequencing. Our scheme is based on breadth-first descent of the binary tree of the key-space. See Figure 2 for the illustration. Notice that for the reader, learning the tag's key is equivalent to establishing the path from the root of the tree to the particular leaf. The reader discovers this path as it descends the tree. The part of the path already learned by the reader terminates in a *growth point*. The reader iterates through growth points in a sequence of *trials*. Observe that all paths share prefixes of various lengths. The objective of the trial is to let the reader know what the next bit on the path after the growth point is and whether the paths split.

In each trial the reader requests that every tag whose key contains the path from the root to the the particular growth point send its next bit. The reader appends the received bit to the growth point. If there is a collision, the path splits producing two growth points.

We illustrate the principle of multiple tag sequencing scheme with the example shown in Figure 2. Assume that the key length is three bits. The tags participating in the identification session have keys: (011), (100) and (101). The reader starts from the growth point a which is the root of the tree. The first trial results in collision. This produces two growth points — b and c. The reader examines b first. The trial produces the next bit without collision, the reader moves the growth point to d. Then the reader examines c and moves it to e. In the next two trials the complete keys of the tags are discovered.

Tag-side sequencing. The pseudocode for the algorithm executed by the tag is shown in Figure 3. The tag has to participate in trials as well as determine its position in the sequence of keys. To be able to do that, the tag maintains the number of growth points in front and behind the growth point that leads to its own key. The tag keeps track as to which growth point is being examined at the current trial. If there is a collision the appropriate number of growth points

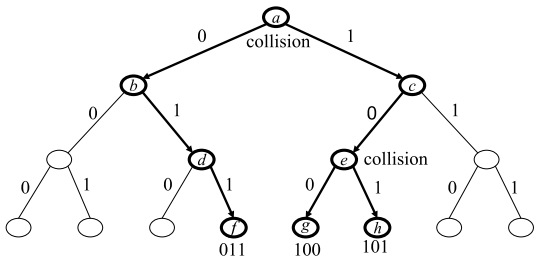

Figure 2: Example tag sequencing.

is incremented. After the entire tree is descended the growth points terminate in the concrete keys and the tag learns its position in the key sequence.

4. IMPLEMENTATION CONSIDERATIONS AND FUTURE WORK

Let us estimate the resource requirements placed on the tag by our algorithm. Key size of 8 bytes provides sufficient keyspace for most RFID applications. The matrices of 4×4 bytes provide adequate security [10]. A few byte-size integer counters are necessary to implement multiple tag sequencing. During the identification session, the reader and the tag exchange a hello-message, as well as two messages of 8 and 9 bytes respectively. Thus, the storage requirements of our algorithm are rather modest and most of the chip-space is to be occupied by the byte-multiplier unit. Overall, these requirements are within the current size limit of RFID tags.

An RFID system that employs our algorithm has to implement a secure mechanism of re-initializing blocked tags that aborted their identification sessions. To provide extra security assurance, this mechanism can also be used to periodically refresh the matrices installed on each tag.

There is a number of extensions of our algorithm that merit further study. Potentially the intruder may launch a denial of service attack. The intruder can block the tags from further identification by starting spurious authentication sessions with them. Blocked tags have to be re-initialized. Protection against this kind of attack would be an interesting extension of our algorithm.

We assume that the intruder is not capable of matching multiple authentication sessions of the same tag through non-radio means. If this is a possibility, the intruder may be able to deduce the product of $M_1 \times M_2^{-1}$ by observing subsequent authentication sessions of the same tag. To prevent this kind of attack, the reader and the tag have to share another key, whose length exceeds the capacity of the intruder to follow the authentication sessions of the same tag.

Refreshing tag-side information over the out-of-band channel may be time consuming, especially if the inventory is large or is not easily accessible. An algorithm that minimizes or eliminates secure channel communication would be desirable for these kinds of applications.

Notice that the combined key and the matrix size in our algorithm is 24 bytes. Having overhead a few identification sessions, the intruder may attempt to mount a brute-force matrix or key guessing attack. Increasing the matrix and

```
const
    q: integer {key size}
    k[1..q]: integer {key}
var
    collide : boolean {trial outcome}
    cfront, pfront: integer, initially 0
                    {currently and previously
                     number of
                     growth points in front}
    cback, pback: integer, initially 0
                    {currently and previously
                     number of
                     growth points behind}
for i ← 1 to q do

    for j ← 1 to pfront do
        collide ← trial()
        cfront ← cfront + 1
        if collide = true then
            cfront ← cfront + 1

    collide ← trial()
    if collide = true then
        if key[i] = 0 then
            cback ← cback + 1
        else
            cfront ← cfront + 1

    for j ← 1 to pback do
        collide ← trial()
        cback ← cback + 1
        if collide = true then
            cback ← cback + 1

    pback ← cback
    cback ← 0
    pfront ← cfront
    cfront ← 0
```

Figure 3: Tag-side algorithm for multiple tag sequencing.

key size may place our algorithm beyond the capabilities of RFID tags. A search for an algorithm of greater security against brute-force attack is an interesting avenue for further investigation.

5. REFERENCES

[1] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In Marc Joye and Jean-Jacques Quisquater, editors, *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370, Boston, Massachusetts, USA, August 2004. IACR, Springer-Verlag.

[2] Klaus Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons, 2nd edition, MAY 2003. Includes bibliographical references and index.

[3] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In Tatsuaki Okamoto, editor, *The Cryptographers' Track at the RSA Conference – CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, San Francisco, California, USA, February 2004. Springer-Verlag.

[4] Dirk Henrici and Paul Müller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In Ravi Sandhu and Roshan Thomas, editors, *Workshop on Pervasive Computing and Communications Security – PerSec 2004*, pages 149–153, Orlando, Florida, USA, March 2004. IEEE, IEEE Computer Society.

[5] Ari Juels. Minimalist cryptography for low-cost RFID tags. In Carlo Blundo and Stelvio Cimato, editors, *The Fourth International Conference on Security in Communication Networks – SCN 2004*, volume 3352 of *Lecture Notes in Computer Science*, pages 149–164, Amalfi, Italia, September 2004. Springer-Verlag.

[6] Ari Juels and John Brainard. Soft blocking: Flexible blocker tags on the cheap. In Sabrina De Capitani di Vimercati and Paul Syverson, editors, *Workshop on Privacy in the Electronic Society – WPES*, pages 1–7, Washington, DC, USA, October 2004. ACM, ACM Press.

[7] Ari Juels and Ravikanth Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In *FC: International Conference on Financial Cryptography*. LNCS, Springer-Verlag, 2003.

[8] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The blocker tag: selective blocking of RFID tags for consumer privacy. In Vijay Atluri and Peng Liu, editors, *Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS-03)*, pages 103–111, New York, October 27–30 2003. ACM Press.

[9] David Molnar and David Wagner. Privacy and security in library RFID: issues, practices, and architectures. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 210–219, New York, NY, USA, 2004. ACM Press.

[10] William Stallings. *Cryptography and network security: principles and practice*. Prentice-Hall, Englewood Cliffs, NJ 07632, USA, second edition, 1999.

[11] Istv An Vajda and Levente Butty An. Lightweight authentication protocols for low-cost RFID tags, August 06 2003.

[12] Stephen Weis, Sanjay Sarma, Ronald Rivest, and Daniel Engels. Security and privacy aspects of low-cost radio frequency identification systems. In Dieter Hutter, Günter Müller, Werner Stephan, and Markus Ullmann, editors, *International Conference on Security in Pervasive Computing – SPC 2003*, volume 2802 of *Lecture Notes in Computer Science*, Boppard, Germany, March 2003. Springer-Verlag.