32

Low Overhead RFID Security*

Shlomi Dolev, Marina Kopeetsky, Thomas Clouser, and Mikhail Nesterenko

CONTENTS

32.1 Introduction

A radio frequency identification device (RFID) technology is poised to revolutionize supply-chain management and retail industry [29]. An RFID system consists of a tag, a reader, and a database. An RFID *tag* is a miniature electronic circuit that is capable of elementary information storage, processing, and radio communication. An RFID *reader* is a device that is designed to communicate with the tag. A reader can extract the information from the tag that identifies the tagged item. The reader is connected to a *database* that contains additional information about the tag and the item. A tag can be self-powered or it can use the power of the reader to do its processing and communication through a mechanism called backscatter. Depending on the design, the range of this communication varies from a few centimeters to hundreds of meters.

* The preliminary version of some of the material of this chapter appeared in [6, 17].

As line-of-sight between the tag and the reader is not required for communication, the RFID systems reduce the time and cost of processing tagged items compared to optical bar-codes. For example, purchased goods can be processed right inside a shopping cart as the customer walks through an automated checkout gate. Potential applications of RFID systems range from inventory-control to smart credit cards, automated toll collection, and counterfeit protection. To be a viable alternative to optical bar-codes, the price of individual tag should be under ten cents [29].

Since the reader can potentially communicate with multiple tags, the problem of *singulation* arises: the reader should be able to identify multiple tags or be able to communicate with each tag individually. Juels et al. [15] propose *treewalking singulation*. The tag identifiers are arranged as leaves of a binary tree. The reader poses queries to individual bits of each tag and descends this binary tree depth-first to identify individual tags.

One of the main hurdles for the widespread adoption of RFID systems is privacy concerns. The concerns become particularly salient as the retail industry contemplates moving from pallet and crate tagging to individual item tagging [29]. RFID use substantially differs from that of other systems. The tag has a close association with the item it identifies. Moreover, the sensitive information usually does not pertain to the tag itself but to the item. This close association between the tag and the item that it identifies gives rise to novel threats such as *tracking* [22] that are not usually addressed in conventional security systems. For example, a conventional system is considered secure if the principal is capable of recognizing the intruder and aborting the communication session before the intruder is able to learn any sensitive information. However, even if the sessions are aborted, the intruder may be able to match the tag across several communication sessions. This gives the intruder the information about the location of the item or the person who carries it. The tag is often used in the environment where the intruder can easily approach it and either eavesdrop on the communication or interfere with it without the knowledge of the communicating parties [10]. For example, an RFID-enabled credit card can be read through the envelope as it is en route from the bank to its owner [11].

To be economically viable for most applications, the tag is not allowed to possess sophisticated data processing capabilities. Thus, the design of security protection for RFID systems is challenging. For example, extensive cryptosystems such as AES, DES, ECC [18], or high-quality random number generators may not be available on the tag. Hence, a substantial amount of recent research effort has been dedicated to design security techniques with sufficiently low overhead to be feasible on RFID systems.

The remainder of the chapter is organized as follows. We survey various approaches to RFID security in Section 32.2. We then present two examples of low overhead authentication algorithms. In Section 32.3, we present an algorithm for mutual tag and reader authentication—M2M that is specialized for the rentalagency setting. In Section 32.4, we describe reader-only authentication algorithm—PISP. In Section 5, we describe how low overhead algorithms such as M2M and PISP can be concurrently applied to multiple tags. We conclude the chapter in Section 32.6 with the discussion of the potential of low overhead cryptography in RFID applications.

32.2 Approaches to RFID Security

32.2.1 Using Traditional Cryptography

There is a large number of recent studies that consider RFID security. Avoine [1] maintains an extensive bibliography on the subject. Juels [13] provides a comprehensive survey of the general area. Most articles focus on RFID security using a variation of traditional symmetric or asymmetric cryptosystems [18].

However, these mechanisms may be unavailable. The manufacturing costs limit the functionality of the tags that are expected to replace optical bar-codes in retail industry. Such tags contain between 500 and 5000 gates [8] most of which are dedicated to basic operations. Only on the order of a few hundred gates are available for cryptographic operations. For example, Feldhofer et al. propose to use AES [7]. Their solution requires 3595 equivalent gates for the security component of an RFID tag. Poschmann et al. [27] present a lightweight version of DES and apply it to RFID. Their implementation requires 1848 gates. Ohkubo et al. [23] present a solution that utilizes two cryptographic hashes. Their estimate is that their solution requires from 6000 to 13,000 gates. Several other keyed hash-based approaches to secure tag authentication are proposed [4,33]. Batina et al. [3] evaluate the feasibility of using elliptic curve cryptography for RFID. They design a processor dedicated to computing ECC on a tag. Their estimate is that such processor will require either over 12,000 equivalent gates or over 8000 gates of dedicated circuits. Skiyama et al. [30] present a specialized processor for ECC cryptography which, they claim, needs only 2171 gates. Molnar and Wagner [22] analyze the security threats of library RFID tags and propose a solution that requires an RFID tag to use random numbers. They do not provide an estimate of the implementation of their algorithm. However, it is known [18, Chapter 11] that obtaining high-quality pseudorandom numbers is equivalent to producing cryptographic hashes or digital signatures. Avoine and Oechslin [2] underline the risks of inadequate pseudorandom number generation in RFID tags. Juels [12] describes a *one-time random pad* security scheme where the communication between the tag and the reader is padded by a random sequence of bits. Both the tag and the reader store this sequence. The communication is secure as long as the sequence is used only once. Because of the limited tag storage, this method restricts the frequency of communication. Thus, an alternative to conventional cryptography may be required for RFID security.

32.2.2 Using No Cryptography

The problems with consumer privacy prompted the suggestions to disable the tags after its services are no longer required. Juels et al. [15] propose to erase all information from the tag after it has been read. Certainly, this diminishes the usefulness of RFID. However, the main difficulty in this approach is to reliably verify if the tag is indeed disabled. Karjoth and Moskowitz [16] propose to physically separate the antenna from the tag. This, however, requires physical contact with the device, which diminishes the applicability of this technology.

Several researchers [9,14,15,28] propose a *blocker tag* or a guardian—a device that monitors communication to protected tags and, if necessary, blocks unauthorized inquiries. For example, Juels et al. [15] propose a blocker tag that does not allow the reader to descend past a certain depth in the singulation tree without authorization. This device answers to the reader as if the identifiers are present in every leaf of the tree thus foiling identification. However, a blocker tag requires the user to inform it which authentication requests are legitimate and which tags need to be blocked. Thus, this approach may not be appropriate for some applications.

32.2.3 Using Low Overhead Cryptography

The advent of RFID technology and the need for lightweight security renewed interest in the solutions that do not require extensive resources from one of the principals: these solutions use limited memory and relatively simple operations such as *XOR*, addition, and possibly multiplication. Such mechanisms might work in RFID security because of the specifics of the application. For example, classic cryptographic protocols are designed to be secure against open or chosen plaintext attacks [18] where the intruder needs to learn the keys and either has access to the unencrypted text or can force one of the legitimate principals to encrypt the text of its choosing. These attacks do not not seem to be applicable to RFID systems.

A number of low overhead RFID security algorithms are proposed [24–26,34]. Peris-Lopez et al. present a series of simple RFID authentication algorithms: LMAP [24], M2AP [25], and EMAP [26]. The proposed algorithms use bitwise *XOR, AND*, and other simple operations and require from 100 to 500 gates. However, recent publications [19,20] demonstrate that all three protocols are vulnerable to *desynchronization attack*. To maintain the freshness of its keys, the tag relies on the reader to provide them in each communication session. If the intruder sends arbitrary information to the tag, the tag may accept it and refresh its keys. This leaves the tag and the legitimate reader unable to communicate. It is also shown that the intruder may determine the keys by observing a sufficient number of communication sessions. Vajda and Buttyán [34] propose another low overhead authentication protocol. To encrypt the communication, in each session the reader provides the tag with a one-time pad. However, Defend et al. [5] show that due to random pad reuse, the protocol can be compromised by a passive intruder within 70 sessions. Even quicker cryptanalysis is possible if active intruder is considered.

32.2.4 Application of Low Overhead Cryptography through Domain Restriction

The apparent vulnerability of low overhead cryptography appears to limit its usability to RFID security. However, we believe that such approaches are still applicable if the domain of RFID use is clearly defined. For example, libraries or other rental agencies have a rather particular routine for tag use. As the tagged items are in storage, the tags may be read for inventory purposes but the security of tag reading is not of primary concern. The tags are read at checkout. Upon return, the tagged items may have to be inspected for damage or otherwise manually checked in. Thus, the tags are exposed to the intruder only between the check-out and check-in time. At the return, there is a possibility to use a back-channel to reinitialize the tags. In Section 32.3, we present a mutual authentication algorithm M2M that is specialized for such environment. Alternatively, there may be some applications where only the reader authentication is required. In Section 32.4, we describe PISP, which is such an algorithm.

32.3 Mutual Authentication with M2M*

32.3.1 Algorithm Description

Each tag stores two square $p \times p$ matrices: M_2 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.

* This algorithm was presented by Karthikeyan and Nesterenko [17].

$$A = (a_1, \ldots, a_{pr}),\ B = (b_1, \ldots, b_{pr}),\ M = (m_{p\times p})$$

$$(a_{p(i-1)+1}, \ldots, a_{pi}) = (m_{p\times p})\begin{pmatrix} b_{p(i-1)+1} \\ \vdots \\ b_{pi} \end{pmatrix},\ \text{where } 1 \le i \le r$$

FIGURE 32.1
Explanation of $A = MB$ notation. (From Karthikeyan, S., and Nesterenko, M., *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks*, November 2005. With permission)

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 (see Figure 32.1). That is, each p-element component A_i of vector A, where $1 \leq i \leq r$, is obtained by multiplying M and the following elements of B: $b_{p(i-1)+1}, \ldots, b_{pi}$. 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 $X = M_1K$ 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 authentication session.

The authentication session has two parts: the tag identification and reader authentication. A complete session of the algorithm is shown in Figure 32.2. 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 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 $K_{\text{new}} \leftarrow X_{\text{new}}M_1^{-1}$. 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.

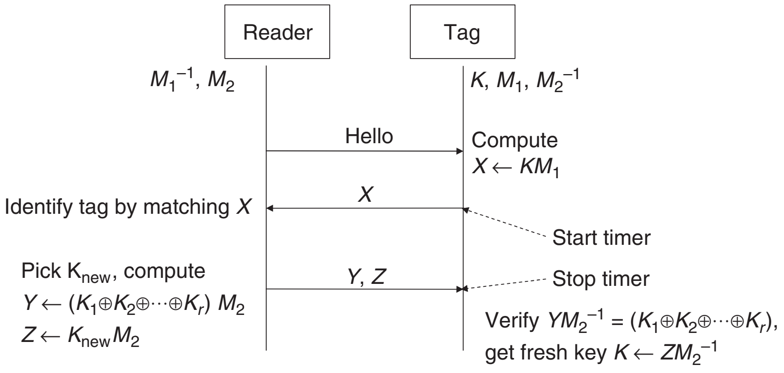
Reader
Tag
M_1^{-1}, M_2
K, M_1, M_2^{-1}
Hello
Compute
$X \leftarrow KM_1$
X
Identify tag by matching X
Start timer
Pick K_{new}, compute
$Y \leftarrow (K_1 \oplus K_2 \oplus \cdots \oplus K_r)\ M_2$
$Z \leftarrow K_{\text{new}} M_2$
Y, Z
Stop timer
Verify $YM_2^{-1} = (K_1 \oplus K_2 \oplus \cdots \oplus K_r)$,
get fresh key $K \leftarrow ZM_2^{-1}$

FIGURE 32.2
M2M authentication scenario. (From Karthikeyan, S., and Nesterenko, M., *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks*, November 2005. With permission)

32.3.2 Security Discussion

The security of our algorithm is based on the difficulty of recovering the multiplicand or multiplier from the product of matrix multiplication [31, Chapter 2]. 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 tracking. The tag does not participate in multiple concurrent 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.

Note we assume that the intruder is not capable of matching multiple authentication sessions of the same tags through nonradio means (e.g., by observing the tagged objects).

32.4 Reader Authentication with PISP*

32.4.1 Security Model

The proactive informational secure protocol (PISP) is based on the the limited intruder capabilities. The underlining assumption of this protocol is that the intruder in not eavesdropping in at least one of each n successive interactions between the tag and the reader. The underlying assumption of PISP is that each communication session is atomic. We mean that the intruder cannot modify part of the communication in a session. The intruder may either listen to the communication during a session, or try to communicate (on behalf of the RFID tag) during an entire session. The intruder may not impersonate the tag.

32.4.2 PISP Description

The pseudocode for PISP is shown in Figure 32.3. The tag and the reader share a square $n \times n$ matrix B. Each element of this matrix contains two parts: a_{ij} and b_{ij}. The first part a_{ij} is a key assigned to the tag and the reader at initialization. The second part b_{ij} is a random number obtained by the tag and the reader during their communication.

During initialization, both the tag and the reader get a square matrix $B = (a_{ij})$ such that dim $(B) = n$ (see Figure 32.3, Protocols for the tag and the reader, lines 1–5). To authenticate the reader, the tag initiates the session by sending the reader a message $s_1 = (X_1, b_1)$ (Figure 32.3, Protocol for RFID Tag, lines 6–9). The message contains the following two elements: XOR of the nth column $X_1 = a_{1n} \oplus a_{2n} \oplus \cdots \oplus a_{nn}$ and a randomly generated n-dimensional vector $b_1 = (b_{11}, b_{12}, \ldots, b_{1n})$. After message transmission both the tag and the reader shift B's rows down so that $b_1 = (b_{11}, b_{12}, \ldots, b_{1n})$ is treated as the first B's row and the last row is deleted (Figure 32.3, Protocol for RFID Tag, lines 10 and 11). This procedure repeats in the next authentication session. That is, the tag generates a new random n-dimensional vector $b_2 = (b_{21}, b_{22}, \ldots, b_{2n})$, calculates XOR of $(n-1)$ B's column

* This algorithm was presented by Dolev and Kopeetsky [6].

```
Protocol for RFID Tag

1:   Initialization:
2:   Define Data Structure E of
3:   int array [1..n], int XOR, E · next
4:   Create Linked List L of n
         elements of type E
5:   int i := 1;
     column = n − (i − 1)mod n

6:   Upon user request
7:       Call function
         XOR X of column i
8:       Create new random array b
9:       Send = (X,b) to Reader
10:      Call Updating procedure
11:  End user request

u1:  Updating procedure
u2:      Add b into the head of List
u3:      Remove last element of List
u4:      i := i + 1

c1:  Function
     XOR X of index column
c2:      X := 0
c3:      current := head
c4:      while current.next not equal to
         NULL do
c5:          X := X ⊕
             current.array[column]
c6:          current := current.next
c7:      end while
c8:      Return X
```

```
Protocol for RFID Reader

1:   Initialization:
2:   Define Data Structure E of
3:   int array [1..n], int XOR, E · next
4:   Create Linked List L of n
         elements of type E
5:   int i = 1;
     column = n − (i − 1)mod n

6:   Upon reception of key message
7:       Call function
         XOR Y of column i
8:       if X = Y
9:           Send "OPEN" to Tag and
             Call Updating procedure
10:      else Send "DoNotOpen"
         to Tag
11:  End of key message reception
```

FIGURE 32.3
PISP pseudocode.

elements $X_2 = b_{1n-1} \oplus a_{1n-1} \oplus \cdots \oplus a_{n-1n-1}$ and sends the newly generated message $s_2 = (X_2, b_2)$ to the reader. The reader generates the response message r_{i+1} as described above. That is, the *i*th authentication procedure consists of scanning the matrix columns and shift the B's rows down so that the last matrix' B row is deleted and the vector b_i occupies the first row of B. Note that b_i is randomly generated by the tag and sent to the reader in the message s_{j-1}. See Figure 32.3 for the matrix update and *XOR* calculation procedure (Protocol for RFID Tag, lines u1–u4 and c1–c8).

To verify the authentication the reader executes the following authentication procedure: after receiving message $s_i = (X_i, b_i)$, the reader verifies that X_i is the correct *XOR* of the appropriate $(n - (i - 1)(\mathrm{mod}(n)))$th column. If so, the reader confirms the authentication by sending message $r_i = \mathit{Open}$ to the tag. The reader also updates B (Figure 32.3, Protocol for RFID Reader, lines 6–11). Otherwise, the reader sends message $r_i = \mathit{DoNotOpen}$ to the tag and does not update B.

Step 1

$$\left(b_{11} \cdots\cdots\cdots b_{1n}\right)$$

$$\left(\begin{array}{cc|c} a_{11} & \cdots\cdots\cdots & a_{1n} \\ \vdots & \cdots \vdots \cdots & \vdots \\ \vdots & \cdots \vdots \cdots & \vdots \\ a_{n-11} & \cdots \vdots \cdots & a_{n-1,n} \\ a_{n,1} & \cdots\cdots\cdots & a_{nn} \end{array}\right)$$

Step 2

$$\left(b_{21} \cdots\cdots\cdots b_{2n}\right)$$

$$\left(\begin{array}{cc|c|c} b_{11} & \cdots\cdots & b_{1n-1} & b_{1n} \\ \vdots & \cdots \vdots & \cdots & \vdots \\ \vdots & \cdots \vdots & \cdots & \vdots \\ a_{n-21} & \cdots \vdots & a_{n-2n-1} & a_{n-2n} \\ a_{n-11} & \cdots\cdots & a_{n-1n-1} & a_{n-1n} \end{array}\right)$$

FIGURE 32.4
Operation of PISP.

Suppose that for any sequence of authentication messages of length n, at least one message is not received by the intruder. In order to compromise the authentication algorithm the intruder has to perform authentication procedure similar to the tag. To do so the intruder has to forge the key message s_{j_i} in any authentication session S_j^{th}. For that the intruder has to correctly guess the *XOR* of the corresponding $(n-(j_i-1)\ (\text{mod}(n)))$th column elements of the matrix B.

Recall that dim $(B)=n$. Let the intruder be unfamiliar with the authentication message $s_1=(X_1, b_1)$ sent by the tag to the reader during the first authentication session (Figure 32.4, Step 1). That is, the intruder does not know the nth column of B—$(a_{1n}, a_{2n}, \ldots, a_{nn})$ and the appropriate row vector is $b_1=(b_{11}, b_{12}, \ldots, b_{1n})$.

After the tag transmits the authentication message $s_1=(X_1, b_1)$, $X_1=(a_{1n} \oplus a_{2n} \oplus \cdots \oplus a_{nn})$, $b_1=(b_{11}, b_{12}, \ldots, b_{1n})$, both the tag and the reader shift the rows of B down as described earlier. Note that in the next session the tag will send to the reader the *XOR* of the updated $(n-1)$th $B's$ column $X_2=(b_{1n-1} \oplus a_{1n-1} \oplus \cdots \oplus a_{n-1n-1})$ and a new randomly generated vector $b_2=(b_{2n}, b_{2n-1}, \ldots, b_{21})$ (Figure 32.4, Step 2). Now matrix B differs from the previous one by the newly inserted first row and the appropriate deletion of the last row.

The PISP is secure from the information-theoretic standpoint, or unconditionally secure as was defined in [32, Chapter 2]. That is, the probability that the intruder will forge the key message and successfully perform session on behalf of the reader, is negligible for long enough l, where l is the number of bits of the element of B. Note also that the matrix size n factors in the security of PISP. If n is large the probability that the intruder does not overhear at least one authentication session is greater. However, large n and l require greater storage resources on the tag.

Note that if the intruder is allowed to eavesdrop n consecutive sessions, it can desynchronize the tag and the reader by forcing the reader to replace a row in B which is unknown to the tag.

32.4.3 Proactive Computationally Secure Protocol

We now describe the proactive computationally secure protocol (PCSP). It is the extension of PISP that allows us to lift the assumption on limited intruder eavesdropping. The pseudocode for the protocol is shown in Figure 32.5. As in PISP, the reader and the tag

Protocol for RFID Tag

```
1:   Initialization:
2:   Define Data Structure E of
3:   int array [1..n], int XOR, E · next
4:   Create Linked List L of n
         elements of type E
5:   int j := 1,   seed = 0
         column = n − (j − 1)mod n
6:   int keyword[k]

7:   Upon user request
8:       Call function XOR X [column]
9:       Create new random array b
10:      Create pseudo-random sequence
         (c[column])  of length m
11:      from seed = X[column] ⊕ seed
12:      Y = (b||keyword[k]) ⊕ (c)
13:      Send s = (Y) to Receiver
14:      Call Updating procedure
15:  End user request

u1:      Updating procedure
u2:      Add b into the head of List
u3:      Remove last element of List
u4:      j := j + 1

c1:  Function XOR X [column]
         of index column
c2:      X [column] := 0
c3:      current := head
c4:      while current.next not equal to
         NULL do
c5:          X [column] := X [column]⊕
             current.array [column]
c6:          current := current.next
c7:      end while
c8:      Return X
```

Protocol for RFID Reader

```
1:   Initialization:
2:   Define Data Structure E of
3:   int array [1..n], int XOR, E·next
4:   Create Linked List L of n
     elements of type E
5:   int j := 1, seed = 0
     column = n − ( j − 1)mod n
6:   int keyword [k]

7:   Upon key message reception
8:       Call function XOR X [column]
9:       Create pseudo-random sequence
             (c[column]) of length m

10:      from seed = X[column] ⊕ seed
11:          Z = Y ⊕ c[column]

12:      if Z [(n+1)..m] = keyword[k]
13:          send "OPEN" to Tag and
             call Updating procedure
14:     else
15:          send "DoNotOpen" to Tag
16:  End of key message reception
```

FIGURE 32.5
Pseudocode for PCSP.

share a square $n \times n$ matrix B (Figure 32.5, protocols for the tag and the reader, lines 1–6). In addition, the tag and the reader also share a string keyword [k].

During the first authentication session the tag acts as follows. As in PISP, the tag calculates the *XOR* of the nth column of B and $X_1 = a_{1n} \oplus a_{2n} \oplus \cdots \oplus a_{nn}$. Similar to PISP, a new row $b_1 = (b_{11}, \ldots, b_{1n})$ is also created as in the proactive information secure protocol case. Tag uses X_1 to initialize its pseudorandom number generator [21, Chapter 12].

The tag creates a new vector row Y_1 that it sends to the reader in in the first authentication message. Y_1 is the *XOR* of the previously generated pseudorandom sequence

$(c_{11}, \ldots, c_{1m})$ and b_1 concatenated with the keyword: $Y_1 = (c_{11}, \ldots, c_{1m}) \oplus (b_1 \| keyword[k])$ (Figure 32.5). Eventually, the secure information encapsulation is provided. The first key message sent from the tag to the reader during the first communication session is $s_1 = (Y_1)$ (Figure 32.5, protocol for RFID Tag, lines 7–13).

Upon receiving this message $s_1 = Y_1$, the reader decrypts it by calculating $Y_1 \oplus (c_{11}, \ldots, c_{1m})$. If the decrypted suffix of the string is equal to the predefined string *keyword*[*k*], then the reader authenticates the tag and returns the message $r_1 = Open$ to the tag. The matrix *B* updating is provided by the prefix of the decrypted string as in the basic information secure protocol. Otherwise, the reader sends message $r_1 = DoNotOpen$ (Figure 32.5, Protocol for RFID Reader, lines 7–16). The update procedure and calculation of *XOR* for the corresponding column elements of *B* is described in Figure 32.5 (Protocol for RFID Tag, lines u1–u4 and c1–c8 respectively).

During any *j*th authentication session $S_j, j = 1, 2, \ldots,$ the tag sends authentication message $s_j = Y_j = (c_{j1}, \ldots, c_{jm}) \oplus (b_j \| keyword[k])$, where $c_j = (c_{j1}, \ldots, c_{jm})$ is the pseudorandom sequence generated by the seed $= X_j \oplus seed$, where the initial value of *seed* is zero (Figure 32.5, Protocols for RFID Tag and for RFID Reader, line 5).]] X_j is the *XOR* of $(n-(j-1)(\mathrm{mod}(n)))$th column elements, and b_j is a newly generated random vector that updates matrix *B*.

Note that the keyword and the pseudorandom number generation function can be known to the intruder. The random seed ensures the security of PCSP. The recursive reuse of the seed used in the previous communication session enhances the security of PCSP.

32.5 Tag Singulation*

Observe that the tag identification algorithms assume 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 algorithms to enable the reader to communicate with multiple tags. Notice that the singulation proceeds concurrently with authentication. Thus, multiple tags can be authenticated concurrently.

The main change in the algorithms is in the identification phase. 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. 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 [15]. 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 singulation. Our scheme is based on breadth-first descent of the binary tree of the key-space. See Figure 32.6 for the illustration. Note that for the reader, learning the

* This algorithm was presented by Karthikeyan and Nestereuko [17].

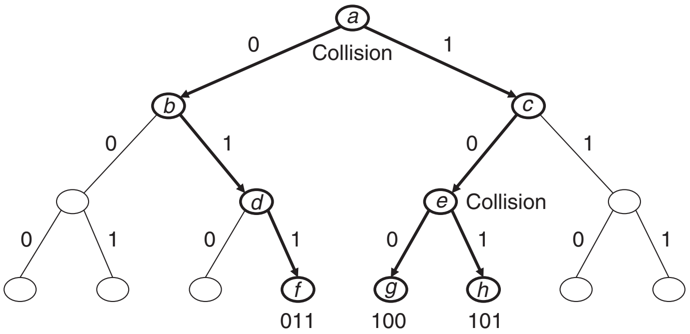

FIGURE 32.6
Example tag singulation.

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 singulation scheme with the example shown in Figure 32.6. 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 singulation. The pseudocode for the algorithm executed by the tag is shown in Figure 32.7. 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 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.

32.6 Conclusion

The need to adequately address RFID security and privacy is important for the technology to fully realize its potential. Certainly, this need will be addressed in part by the conventional cryptography algorithms that are adapted to use for RFID. However, for the systems

```
Initialization
constants
q: integer {key size}
k[1..q]: integer {key}

variables
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}

Operation

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 32.7
Tag-side singulation algorithm.

such as retail item-tagging where such heavyweight solutions are prohibitively expensive, low overhead cryptography may provide a convenient alternative.

Acknowledgments

This work was partially supported by Rita Altura Trust Chair in Computer Sciences, Lynne and William Frankel Center for Computer Sciences and Deutsche Telekom, DARPA contract OSU-RF #F33615-01-C-1901, NSF CAREER Award 0347485, and a grant from the Wright Center for Sensor System Engineering.

References

1. G. Avoine. Bibliography on security and privacy in RFID systems. Available Online, 2006.
2. G. Avoine and P. Oechslin. RFID traceability: A multilayer problem. In *Financial Cryptography (FC)*, Roseau, The Commonwealth Of Dominica, February–March 2005. *Lecture Notes in Computer Science*, vol. 3570, pp. 125–140. IFCA, Springer-Verlag, 2005.
3. L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. An elliptic curve processor suitable for RFID-tags. Cryptology ePrint Archive, Report 2006/227, 2006.
4. M. Burmester, T. van Le, and B. de Medeiros. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, Baltimore, Maryland, USA, August–September 2006. IEEE, 2006.
5. B. Defend, K. Fu, and A. Juels. Cryptanalysis of two lightweight RFID authentication schemes. In *International Workshop on Pervasive Computing and Communication Security (PerSec) 2007*, New York, USA, March 2007. IEEE Computer Society Press.
6. S. Dolev and M. Kopeetsky. Secure communication for RFIDs proactive information security within computational security. In *Stabilization, Safety, and Security of Distributed Systems (SSS), Lecture Notes in Computer Science*, vol. 4280 pp. 290–303. Springer, 2006.
7. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, Boston, Massachusetts, USA, August 2004. *Lecture Notes in Computer Science*, vol. 3156 pp. 357–370. IACR, Springer-Verlag, 2004.
8. K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley, New York, NY, USA, 2003.
9. C. Floerkemeier, R. Schneider, and M. Langheinrich. Scanning with a purpose—supporting the fair information principles in RFID protocols. In *International Symposium on Ubiquitous Computing Systems (UCS)*, Tokyo, Japan, November 2004. *Lecture Notes in Computer Science*, vol. 3598 pp. 214–231. Springer-Verlag, 2004.
10. G. Hancke. Practical attacks on proximity identification systems (short paper). In *IEEE Symposium on Security and Privacy*, Oakland, California, USA, May 2006 pp. 328–333. IEEE Computer Society Press, 2006.
11. T.S. Heydt-Benjamin, D.V. Bailey, K. Fu, A. Juels, and T. O'Hare. Vulnerabilities in first-generation RFID-enabled credit cards. Manuscript, October 2006.
12. A. Juels. Minimalist cryptography for low-cost RFID tags. In *International Conference on Security in Communication Networks (SCN)*, Amalfi, Italia, September 2004. *Lecture Notes in Computer Science*, vol. 3352 p. 149–164. Springer-Verlag, 2004.
13. A. Juels. RFID security and privacy: A research survey. Manuscript, September 2005.
14. A. Juels and J. Brainard. Soft blocking: Flexible blocker tags on the cheap. In *Workshop on Privacy in the Electronic Society (WPES)*, Washington, DC, USA, October 2004, p. 1–7. ACM Press, 2004.
15. A. Juels, R. Rivest, and M. Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In *Conference on Computer and Communications Security (CCS)*, Washington, DC, USA, October 2003, p. 103–111. ACM Press, 2003.
16. G. Karjoth and R. Moskowitz. Disabling RFID tags with visible confirmation: Clipped tags are silenced. In *Workshop on Privacy in the Electronic Society (WPES)*, Alexandria, Virginia, USA, November 2005. ACM Press, 2005.
17. S. Karthikeyan and M. Nesterenko. RFID security without extensive cryptography. In *Proceedings of the 3rd ACM workshop on security of ad hoc and sensor networks (SASN)*, New York, NY, USA, 2005, pp. 63–67. ACM Press, 2005.
18. C. Kaufman, R. Perlman, and M. Speciner. *Network Security: Private Communication in a Public World*. Prentice- Hall, 1995.
19. T. Li and R.H. Deng. Vulnerability analysis of EMAP—an efficient RFID mutual authentication protocol. In *Second International Conference on Availability, Reliability and Security (AReS)*, Vienna, Austria, April 2007.

20. T. Li and G. Wang. Security analysis of two ultra-lightweight RFID authentication protocols. In *International Information Security Conference (IFIP SEC)*, Sandton, Gauteng, South Africa, May 2007. IFIP, 2007.
21. A.J. Menezes, S.A. Vanstone, and P.C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, USA, 1996.
22. D. Molnar and D. Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *ACM Conference on Computer and Communications Security (CCS)*, Washington, DC, USA, October 2004, pp. 210–219, ACM Press, 2004.
23. M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic approach to ''privacy-friendly'' tags. In *RFID Privacy Workshop*, MIT, MA, USA, November 2003.
24. P. Peris-Lopez, J.C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Workshop on RFID Security (RFIDSec)*, Graz, Austria, July 2006. Ecrypt.
25. P. Peris-Lopez, J.C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. M2AP: A minimalist mutual- authentication protocol for low-cost RFID tags. In *International Conference on Ubiquitous Intelligence and Computing (UIC), Lecture Notes in Computer Science*, vol. 4159 pp. 912–923. Springer-Verlag, 2006.
26. P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, and A. Ribagorda. EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In *OTM Federated Conferences and Workshop: IS Workshop, Lecture Notes in Computer Science*, vol. 4277 pp. 352–361. Springer-Verlag, 2006.
27. A. Poschmann, G. Leander, K. Schramm, and C. Paar. A family of light-weight block ciphers based on DES suited for RFID applications, July 2006.
28. M. Rieback, G. Gaydadjiev, B. Crispo, R. Hofman, and A. Tanenbaum. A platform for RFID security and privacy administration. In *USENIX/SAGE Large Installation System Administration conference (LISA)*, Washington DC, USA, December 2006.
29. G. Roussos. Enabling RFID in retail. *IEEE Computer*, 39(3):25–30, 2006.
30. K. Sakiyama, L. Batina, N. Mentens, B. Preneel, and I. Verbauwhede. Smallfootprint ALU for public-key processors for pervasive security. In *Workshop on RFID Security (RFIDSec)*, Graz, Austria, July 2006. Ecrypt.
31. W. Stallings. *Cryptography and Network Security: Principles and Practices*, 2nd end. Prentice-Hall, Upper Saddle River, NJ, USA, 1999.
32. D. Stinson. *Cryptography: Theory and Practice*, 3rd end CRC/C&H, 2006.
33. G. Tsudik. YA-TRAP: Yet another trivial RFID authentication protocol. In *International Conference on Pervasive Computing and Communications (PerCom)*, Pisa, Italy, March 2006. IEEE Computer Society Press, 2006.
34. I. Vajda and L. Buttyán. Lightweight authentication protocols for low-cost RFID tags. In *Second Workshop on Security in Ubiquitous Computing (Ubicomp)*, Seattle, WA, USA, October 2003.