Achieving Privacy and Security in Radio Frequency Identification

Aaditeshwar Seth and Mirza Beg
School of Computer Science
University of Waterloo, Ontario
Canada, N2L 3G1
{a3seth, mbeg}@cs.uwaterloo.ca

Abstract— **Radio Frequency Identification RFID systems are gaining popularity in a wide variety of applications like asset tracking, personnel identification, and sensor networks. However, unique security and privacy issues arise in these systems because (a) low computation capabilities of RFID tags prevent the use of complicated cryptographic protocols, and (b) wide deployment of tags opens up room for illegal tracking of people and objects. In this paper, we first describe a basis-set of requirements that need to be necessarily satisfied to mitigate security and privacy problems in RFID systems. We then outline some recent proposals that try to solve these issues, and then explore in detail a research publication by Molnar, et al [1] that uses a pseudonym based tree walking security scheme, and claims to meet all the requirements. However, we identify some attacks that are still possible in this scheme in slightly different threat models, and then extend the scheme to mitigate these attacks. We also address the issue of secure establishment of session keys to exchange information between tags, readers, and centralized trusted centers, which had not been proposed earlier. Our extensions make the overall scheme complete, and provides a comprehensive solution to security and privacy issues in RFID systems that meets all the requirements.**

I. Introduction

Radio Frequency Identification (RFID) systems have gained immense popularity during recent years. The motivation behind the pervasive use of RFID systems is the need to fully automate remote tracking and identification of objects by embedding cheap and low power RFID tags in the objects. RFID tags are composed of an antenna, and a small microchip with some identification information encoded in it. The antenna powers the tag with radio waves emitted by a nearby reader. The data transmitted by the tag may contain identification or location information, or specifics about the product being tagged, such as price, color, date of purchase, etc. In addition to the capabilities of the passive tags described above, active tags may have an internal power source and some computational capability, which increases their 'read range' and allows for simple cryptographic computations.

Traditionally barcodes and magnetic strips served the same purpose but they are being rapidly replaced by RFID tags for keeping tabs on people, pets, products, and vehicles. Their use is being extended to even drivers licences, national identification cards, passports [6, 8], and even bank notes [16]. One reason for this is the read/write capability of an active RFID system which enables its use for low-cost interactive applications. Also, the tags can be read from a distance and through a variety of substances such as snow, fog, ice, or paint, where barcodes have proved useless.

However, the RFID technology is rife with problems related to security and privacy. There are concerns that information stored on RFID tags could be read by anyone with an RFID reader - data thieves, hackers, or forgerers. Some of these issues are explained in [7, 10, 17], and outlined below.

- **Surveillance of individuals and objects:** RFID tags are likely to be embedded into objects and documents with or without the knowledge of the individual. As radio waves travel easily and silently through fabric, plastic, and other materials, it is possible to read RFID tags sewn into clothing or affixed to objects contained in purses, shopping bags, or suitcases making it possible to track the location of items or the owner.
- **Massive data aggregation:** The Electronic Product Code (EPC) [15] potentially enables every object on earth to have its own unique ID. The use of unique ID numbers could lead to the creation of a global databases in which every physical object is identified and linked to its purchaser or owner at the point of sale or transfer. These records can be linked with personal identifying data and can be later used for different objectives such as identifying consumer habits without consent of the consumer.
- **Eavesdropping:** Tags can potentially be read from a distance, not restricted to line of sight, by readers that can be incorporated invisibly into nearly any environment where human beings or items congregate. RFID passports can be read from as far as 30 feet [9]. This information can potentially be used to forge identification documents such as passports and licences for various nefarious purposes.

A. Our contributions

In this paper, we describe a basis-set of requirements that should necessarily be met in order to deal with the attacks mentioned above. We then survey research on security and privacy issues in RFID systems and explore in detail the protocol by Molnar, et al [1]. We look at scenarios in which the protocols fail to meet the basis-set of requirements, and propose enhancements on how to improve the scheme to make it secure. More specifically, the protocol fails to handle a clone attack between two consecutive reads of the tag. It also fails to secure the tag from a DoS attack which can render the tag unusable. We present enhancements to the protocol to successfully prevent both the above mentioned attacks. In addition to that, we also present techniques to provide secure and authenticated communication in cases where the tags must transfer information to the reader when RFID tags are used in sensors. Also we propose a scheme to renew the key of a tag when it is about to expire. In all, our extensions provide a complete RFID scheme that meets all the security and privacy goals in the basis-set.

This paper is organized as follows. In Section II, we identify the basic set of requirements necessary to prevent the attacks in RFID systems enumerated above. We describe some related works and scenarios where they fail in Section III, and choose the security scheme proposed in [1] for further analysis. We describe the scheme and its security properties in Section IV, and identify two attacks possible on the original scheme. In Section V we then propose certain extensions to the scheme that are able to mitigate the attacks, and also enhance the scheme to enable secure transfer of information between various entities in the RFID network. In Section VI we describe a detailed state maintenance procedure to clarify implementation details. Finally we present our conclusions in Section VII.

II. Basis set of goals for security and privacy in RFID systems

We see that attacks can be made in RFID systems due to violations in one or more of the following basis set of requirements:

1. **Tag authentication:** This is required to prevent tag cloning because duplicated tags can lead to impersonation attacks. Cloning attacks on unprotected or weakly-protected tags can be conducted if any of the following are possible.

(a) An adversary is able to overhear transmission from valid tags, and replay the transmissions when it is queried by a tag reader.

(b) If security mechanisms are built such that direct replay attacks are not possible, an adversary is able to reproduce the response of a valid tag by collecting enough information from compromised tags to be able to break the security scheme.

2. **Privacy:** This is required to prevent movement tracking of RFID tagged items. Privacy can degrade if any of the following are possible.

(a) Adversary readers are able to pretend to be valid readers and query tags to obtain their IDs. Over time, colluded readers are able to track the movement of tags in physical space.

(b) If IDs are not transmitted as such but are encoded through some security mechanisms, eavesdroppers are able to disambiguate between different tags based on the uniqueness properties of communication arising from different tags. Since eavesdroppers are able to uniquely identify the tags, over time colluded eavesdroppers can then track the movement of tags.

Our goals in this paper are to design a security mechanism that can meet the requirements stated above without involving any high cost cryptographic procedures that cannot be implemented on RFID tags.

III. Related work

There is some amount of research available to design secure, light-weight cryptographic schemes for RFID systems to address the privacy concerns and to achieve the goals stated in Section II, but it has had limited success.

Juels et al [14] consider a formal security model called the *detection model*. The underlying assumption is that the adversary's goal is to clone a tag without being detected in the attempt. Although the scheme *detects* unauthentic attempts to read the tag, it is unable to prevent the adversary from cloning a tag.

Ohkubo et al [11] propose a method of changing RFID ID's on each read using pseudonyms. The drawback of this scheme is that the trusted center must be online on each tag read. Recovering tag identity requires work linear in terms of possible tags because of the key pre-distribution scheme. The proposed scheme does not mutually authenticate the reader and the tag.

Juels [13] proposes a security model for low-cost passive tags. The model assumes that the adversary comes into close proximity of the tag only on a periodic basis. The model assumes a cap on the number of times that the tag can be read before going into *private* mode in which it can only be read by an authentic reader. This model also fails to achieve the goal of authentication mentioned in Section II. Furthermore, the tag must be refreshed at frequent intervals.

Juels et al [12] propose a privacy-protecting scheme that is called *blocking*. Their scheme depends on the incorporation into tags a modifiable bit called the *privacy bit*. A '0' value for this bit marks it for unrestricted public scanning

and a '1' marks the tag to be in the *private zone*. A *blocker tag* is a special tag that prevents unwanted scanning of tags mapped into the privacy zone.

Molnar et al [1] propose a key pre-distribution scheme for the tags that claims to handle all the issues with the schemes enumerated above. We explain the scheme in detail in Section IV. However, we find that the scheme fails in certain attack models which we discuss in this paper, and we propose some extensions to the scheme in Section V to mitigate the attacks.

IV. Scalable, delegatable, pseudonym protocol [1]

The authors have defined an RFID *pseudonym* protocol in [1] where the tag emits a different pseudonym each time it is queried by a tag reader. The tag reader cannot decipher the identity of the tag from the pseudonym alone. It queries a TC (Trusted Center) which maps the pseudonym to the tag ID and returns the ID to the reader. Privacy is ensured because (a) only trusted readers are allowed to query the TC, and (b) an eavesdropper cannot disambiguate between any two pseudonyms to determine whether it was the same tag that emitted both the pseudonyms or not. Through an interesting tree-walk algorithm, the protocol is also able to provide ownership transfer primitives and time-limited delegation to offline readers.

A. Protocol

Fig. 1 shows the operations of the protocol. Each tag maintains a state variable *c* instead of its ID. The prefix of *c* is unique for each tag however, and the TC maintains a mapping between the unique prefix and the tag ID. Based on *c* and a pseudo-random variable *r*, the tag generates a unique pseudonym *p* in response to each HELLO message from a reader. The reader forwards *p* and *r* to the TC, which is able to recalculate *c* given *p* and *r*. The TC then finds the tag ID based on the unique prefix of *c*, and returns it to the reader. At the same time, the tag increments *c* upon each transaction.

When bootstrapping a new tag, the TC assigns a unique identifier *s* to each tag. This identifier may be the same as the tag ID, or mappings of (*s*, ID) can be maintained locally in a database. As shows in Fig. 1, *s* corresponds to an ordered traversal [1] in a binary tree of height d_1 from the root to a unique leaf node. This is done as follows. The integer *s* is represented in binary, with a 0 denoting the left branch and 1 denoting the right branch. Thus, each path in a tree from the root to a leaf node at level d_1 corresponds to a unique integer *s*. The protocol works by extending this tree by an additional d_2 levels, where each value of *c* corresponds to an ordered traversal up to a node in the tree between levels d_1 and d_2. The first d_1 bits prefixing *c* are equal to *s*. The value of *c* for each tag is incremented from $(s_0...s_{d_1}||\{0\}^{d_2})$ to $(s_0...s_{d_1}||\{1\}^{d2})$, corresponding to the bottom-left-most and bottom-right-most nodes respectively in the sub-tree for the given value of *s*. Instead of transmitting *c* as such, the tag encodes it in a pseudonym *p* containing $(d_1 + d_2)$ pseudo-random numbers, calculated on the basis of a pseudo-random variable *r* and the current value of *c*. Each pseudo-random number p_i in *p* is calculated on the first *i* bits of *c*. The TC decodes *p* by reconstructing *c* through a simple DFS algorithm that matches at each level *i* the received p_i with the p_i calculated on $(c_0...c_{i-1}||0)$ and $(c_0...c_{i-1}||1)$. The exact algorithms are shown in [1].

Three pseudo-random functions are needed on the tag: A hash function *h*: $\{0,1\}^n \rightarrow K$, a pseudo-random generator PRG, and a pseudo-random function *F*: $\{0,1\}^n \ X \ K \rightarrow \{0,1\}^n$. As explained in [1], all these functions can be implemented using the same implementation of *F* with fixed salt values. The authors suggest using AES for implementing *F* because AES implementations have been shown to be possible within 500 gates on low-cost tags [3]. Further optimizations are possible by just implementing the restriction of *F* on *s* (that is, $F|_s$) on the tags. Ownership transfer and time-limited offline delegation can be done by offloading appropriate restrictions of *F* on suitable subtrees to trusted readers so that these readers need not have to query the TC while the values of *c* are within the restrictions given to them.

B. Security analysis

[1] provides replay-only security against impersonation and privacy attacks against a radio-only adversary because tag disambiguation is guaranteed. Replay-only security is also provided against impersonation attacks even if an adversary can compromise tags because each tag has at least one secret not shared with any other tag. To perform a successful non-replayed impersonation, the adversary would need to predict the value of a pseudo-random function keyed with such a secret.

Authentication is not done because privacy is guaranteed even otherwise. However, as we will show next, certain attacks still remain possible unless the basic scheme is not extended suitably.

C. Replay attack

Consider a situation where RFID tags are used to control access to a building. An attacker can go into a bar where employees working in the building generally hang out, and scan a few tags. As shown in Fig. 2, the radio-only attacker can do this easily by sending a HELLO message to a tag and then store the (*p, r*) pair. The attacker can next retransmit the (*p, r*) pair to a trusted reader that controls access into

[1] An ordered traversal can be a preorder or postorder traversal.

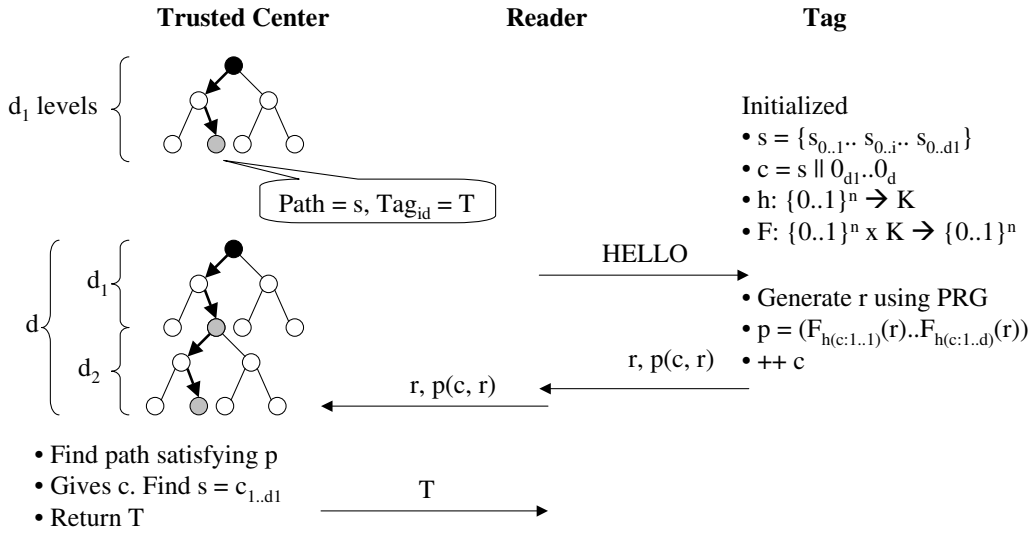

Fig. 1. Basic protocol

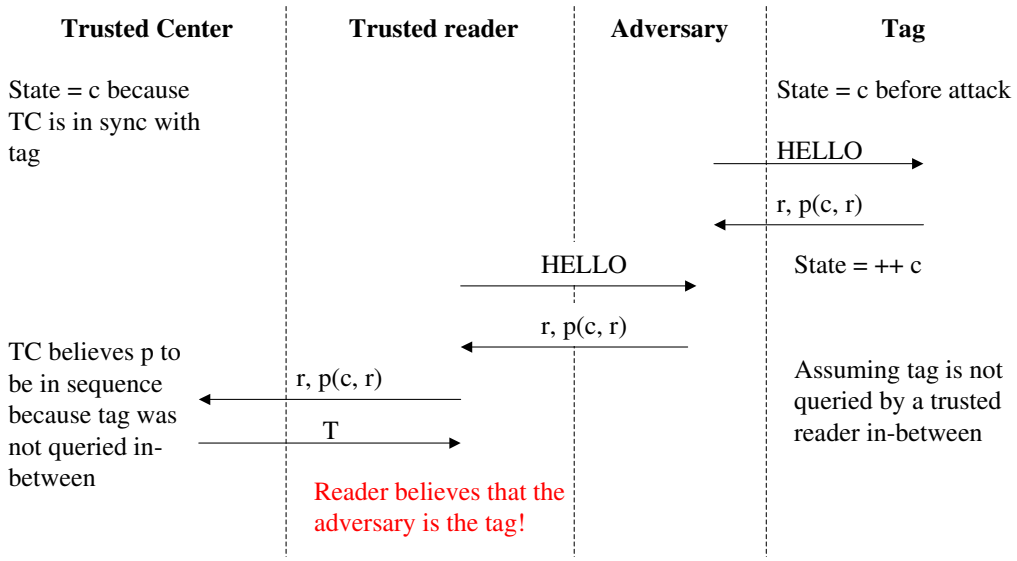

Fig. 2. Replay attack

the building. Thus, impersonation attacks can be launched even if the TC keeps track of the expired values of *c* for each tag, provided that the tag is not queried by a valid reader just before the attack is launched. Such attacks have not been considered as valid attacks in traditional security literature, but the notion of disconnected authentication in RFIDs can open up new attacks similar to this.

D. DoS attack

Since any radio-only adversary can query a tag even if the adversary is not able to decipher the tag's ID, repeated querying can eventually lead to a buffer overflow on the tag by successive increments of *c*. This is shown in Fig. 3 where a buffer overflow can render the tag useless. In addition, a reader can query the same tag repeatedly and collect enough information about the tag secret to be able to break the scheme.

V. EXTENSIONS

A. Solution to the replay attack

As shown in Fig. 4, we mitigate the replay attack by introducing a mechanism to authenticate the tags. The readers now send a random nonce r_1 to the tags, and the tags

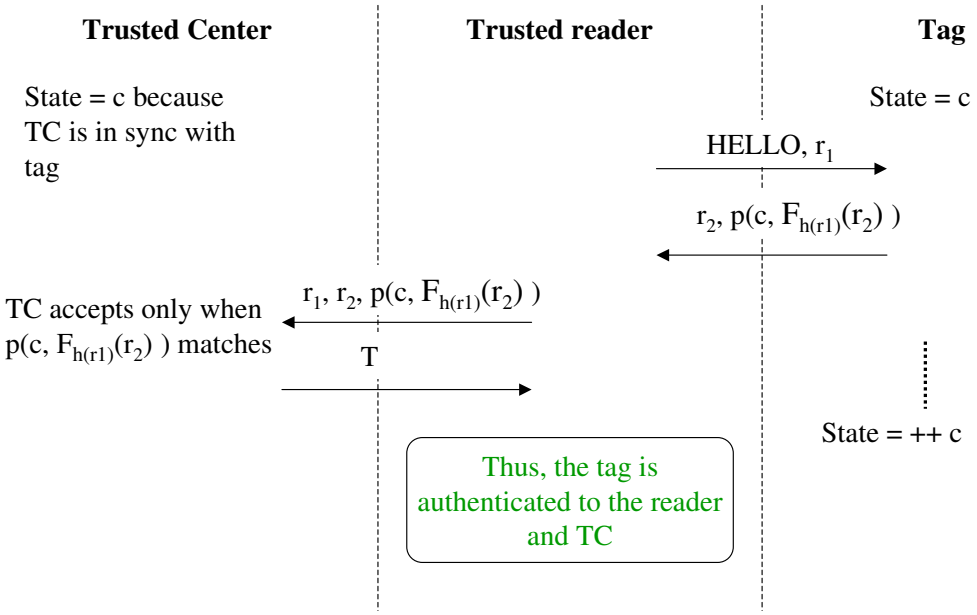

Fig. 4. Mitigation of replay attack

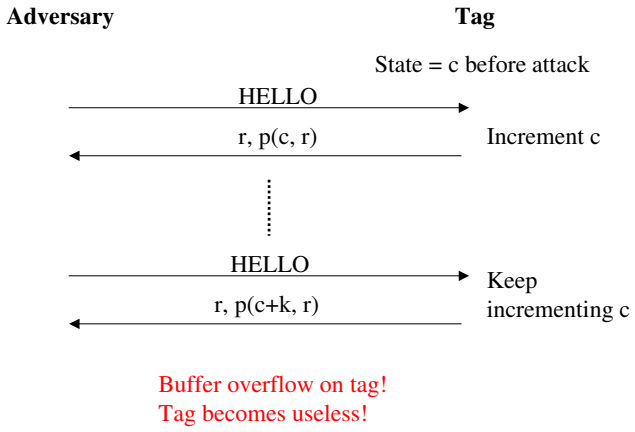

Fig. 3. DoS attack

use r_1 along with their own randomly generated nonce r_2 to calculate $r = F_{h(r_1)}(r_2)$. This *r* is then used as before to find *p*. The TC can redo the calculations in the same way as earlier by using $F_{h(r_1)}(r_2)$ in place of *r*. Here, we assume that r_1 expires quickly so that attackers cannot make use of delayed authentication to get challenges from the reader, followed by corresponding responses from the tags. This assumption is practical to make in the modified threat model of disconnected authentication that we outlined in Section IV-C. The timeout to control r_1 expiry is shown later in Section VI-B as the transition from state S_2 to S_0 of the tag reader.

The probability of attack in this modified scheme is equal to the probability of generating a valid pseudonym *p* given *r*, which is also in synchronization with *c*.

B. Solution to the DoS attack

DoS attacks on tags can be prevented through the extensions shown in Fig. 5, by authentication of valid readers to tags. The TC first verifies valid tags, and then sends back to valid readers a pseudonym calculated on a new pseudo-random number r_3. The reader forwards this pseudonym to the tag, and only upon verification does the tag increment *c*. The probability of attack in this modification is the same as in the previous modification.

Divulsion of too much information about the tag secret can be prevented by introduction of a sufficiently large wait-time on the tag so that the same reader can be prevented from rapidly sending HELLO messages with the same value of r_1. Thus, most attacks can be avoided in realistic scenarios. This wait-time is indicated in Section VI-A as the timeout for transition from state S_3 to S_0 of the tag.

C. Secure transfer of data

RFID tags are likely to find use as sensors because of their low power consumption characteristics. In such cases, there will be data that resides on the tags and not with the TC, as is assumed in [1]. Transmission of this data in a secure manner requires the establishment of session keys between the tags and readers, and between tags and the TC. Similarly, session key establishment is also needed in case the TC is required to send data to tags in a secure manner, for example, to renew tags with new secret

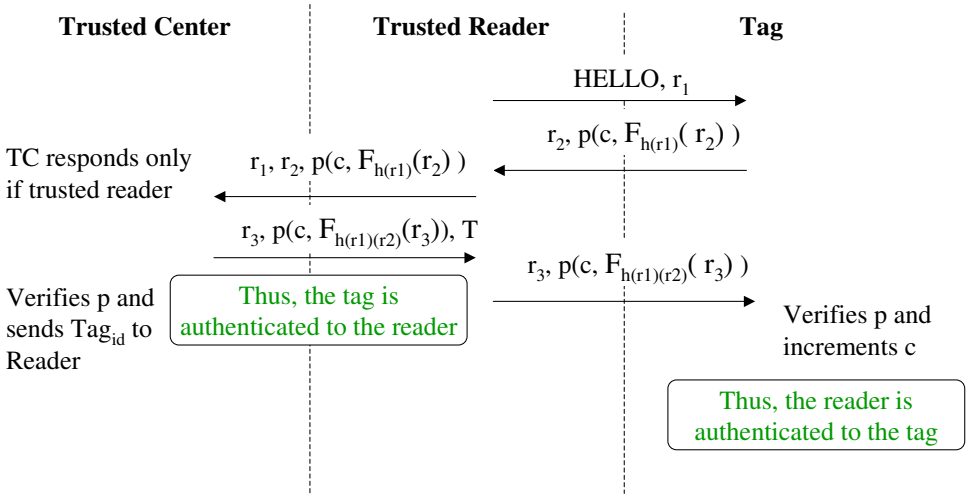

Fig. 5. Mitigation of DoS attack

keys when tag state counters are about to overflow. We explain below the procedures required to provide secure transfer of data.

C.1 Secure transfer between tags and readers

Fig. 6 shows the communication protocol for secure transfer between a reader and tag. The TC calculates a pseudonym based on a new pseudo-random number r_3 and sends this to the trusted reader in response to a tag query. The reader only forwards r_3 to the tag; the tag calculates the same pseudonym itself. Thus, the pseudonym can now be used as a secret key between the reader and tag. The data to be securely transferred can either be encrypted using the same AES implementation, or else a simple XOR of the data with the session key can also be used.

Note that the authentication mechanism explained earlier for reader authentication can be added to this extension as well. We did not show it in Fig. 6 to keep the explanation simple.

C.2 Secure transfer between tags and TC

The fundamentals of the protocol shown in Fig. 7 are almost the same as the previous protocol. This time, the TC calculates a new pseudonym but only sends the pseudo random number r_3 to the reader. The reader forwards this to the tag, which calculates the pseudonym on its own and uses it as the secret session key for secure communication with the TC. It is even possible to sign the encrypted message using the same principles. Note that this signature scheme can be included in the previous scheme for secure transfer between readers and tags.

VI. Implementation

The protocol can be defined in terms of state changes in the tag, the reader and the TC. These state description models include the authentication extensions, but not the secure transfer extensions.

A. Tag States

A comprehensive picture of the state diagram for a Tag in our system is shown in Fig. 8.

- S_0 : QUIET
  $Remain\ Idle$

In this state the tag remains dormant. It does not require any energy to remain in this state.

- S_1 : RECEIVE
  $Receive$ HELLO $and\ r_1\ from\ reader$
  $Generate\ a\ random\ value\ r_2$
  $Compute\ psuedorandom\ value\ p(c,\ F_{h(r_1)}(r_2))$

The tag goes into this state once it receives a HELLO message accompanied by an integer value r_1 from an RFID reader. The tag then generates a random value r_2 using its pseudorandom number generator. Then it computes the value

$$p_1 = p(c,\ F_{h(r_1)}(r_2))$$

using the values c and the random numbers r_1 and r_2 and functions h and F. Notice that unlike the basic protocol, the value of c is not incremented until Tag state S_6; thus preventing the attacks mentioned in Section IV.

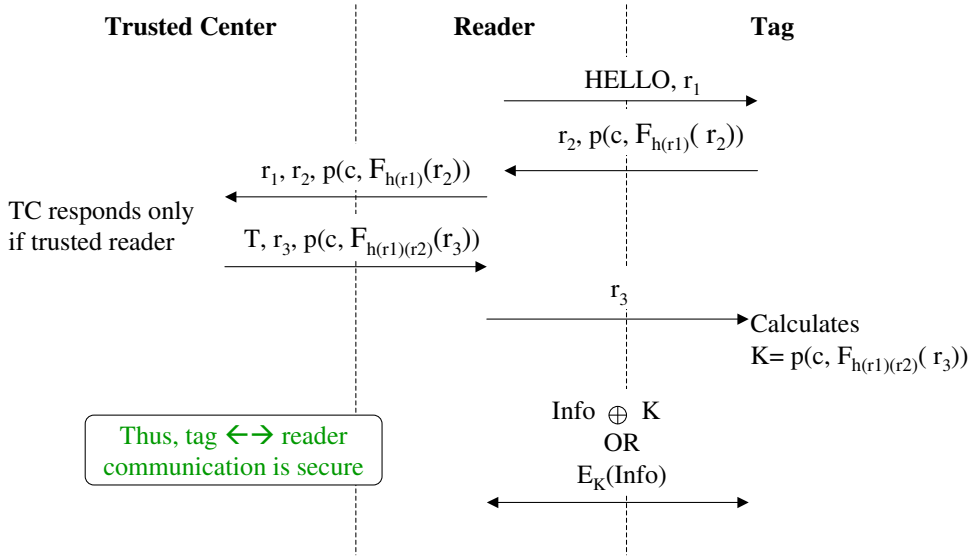

Fig. 6. Secure communication between reader and tag

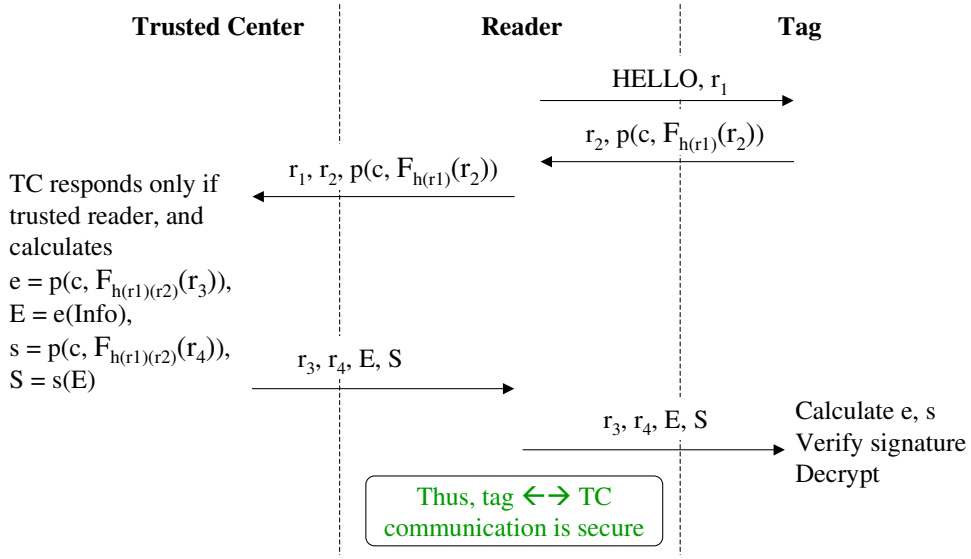

Fig. 7. Secure communication between reader and TC

- S_2 : SEND
  Send r_2 *and* $p(c,\ F_{h(r_1)}(r_2))$ *to reader*

Once p_1 is successfully calculated by the tag, it is transmitted to the reader. With high probability, each time the tag is read, the value of p_2 is unique and cannot be related to previous transmissions. Also it does not contain possibly sensitive data from the tag to ensure privacy.

- S_3 : WAIT
  Wait for reader response
  Check for time out if no response
  Go to state S_0 *upon timeout*

After transmitting its response the tag waits for a response from the reader. If there is no response within a specific time frame t then the tag goes into its initial state S_0.

- S_4 : RECEIVE
  Recieve reader response r_3 *and* p_2

Once the tag receives a response containing two values it moves into the TAG.VERIFY state to verify the values from

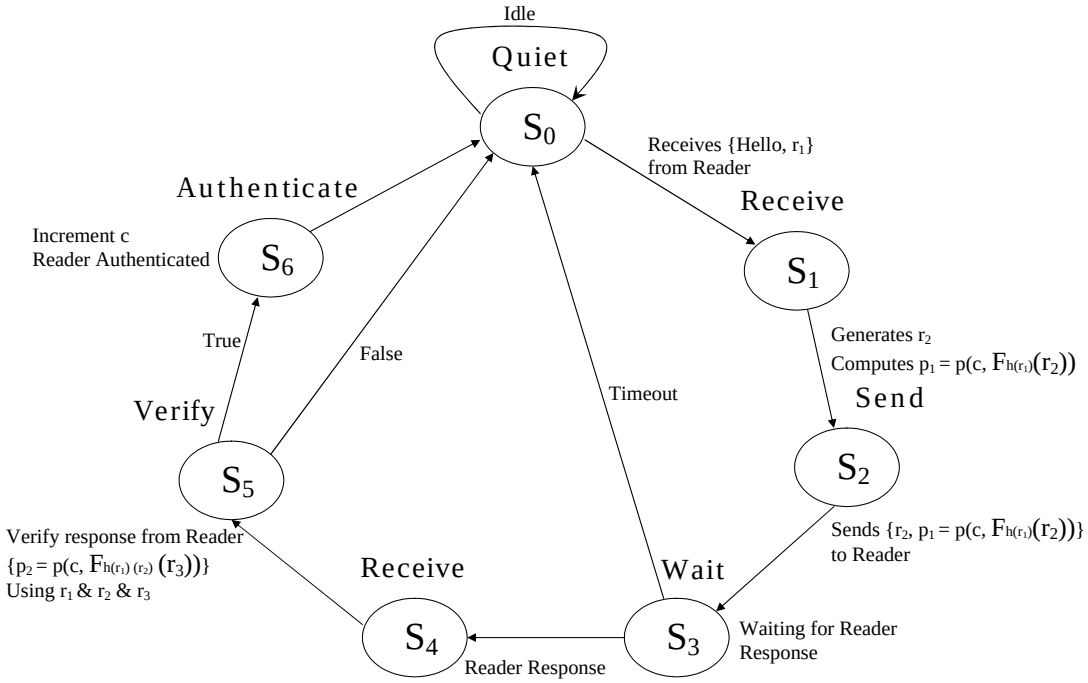

Fig. 8. State diagram for the RFID Tag

the reader.

- S_5 : VERIFY
  $Verify\ that\ p_2\ from\ previous\ state\ is$
  $p(c,\ F_{h(r_1)(r_2)}(r_3))\ by\ recomputing\ the\ functions$
  $F\ and\ h\ on\ r_1,\ r_2\ and\ r_3$
  $if\ reader\ response\ p_2\ is\ verified\ then\ the$
  $accept\ the\ reader$
  $else$
  $go\ back\ to\ state\ S_0$

The tag computes the value p_2 using the functions F and h and compares it with the value it received from the reader. If the values are equal then the reader is authenticated and if the values do not match the reader is recognized as a rogue reader. Thus, even if the reader is rogue, no sensitive information specific to the tag is revealed during the transmissions and the tag returns to its initial state S_0 without changing the internal state of its variables.

- S_6 : AUTHENTICATE
  $Reader\ is\ accepted$
  $Increment\ c$

Once the response from the reader is verified the tag accepts the reader as being valid and updates its internal state by incrementing the value of c. By making this increment it moves to the next available node in the tree of sequence and the earlier verification affirms that the Trusted Center is maintaining the same state for this tag. This is the last valid state of the tag and it once the reader is authenticated and a valid read confirmed, the tag goes into the TAG.QUIET state, ready to be read again.

B. Reader States

The state diagram for an RFID reader is given in Fig. 9.

- S_0 : QUIET
  $In\ Idle\ mode$

The reader remains in the idle state until prompted to read the tags in its range.

- S_1 : READ
  $Generate\ a\ random\ value\ r_1$
  $Transmit$ HELLO $and\ r_1\ to\ tag$

When the reader is prompted to read the tags in its range, it generates a random value r_1 using a pseudorandom gen-

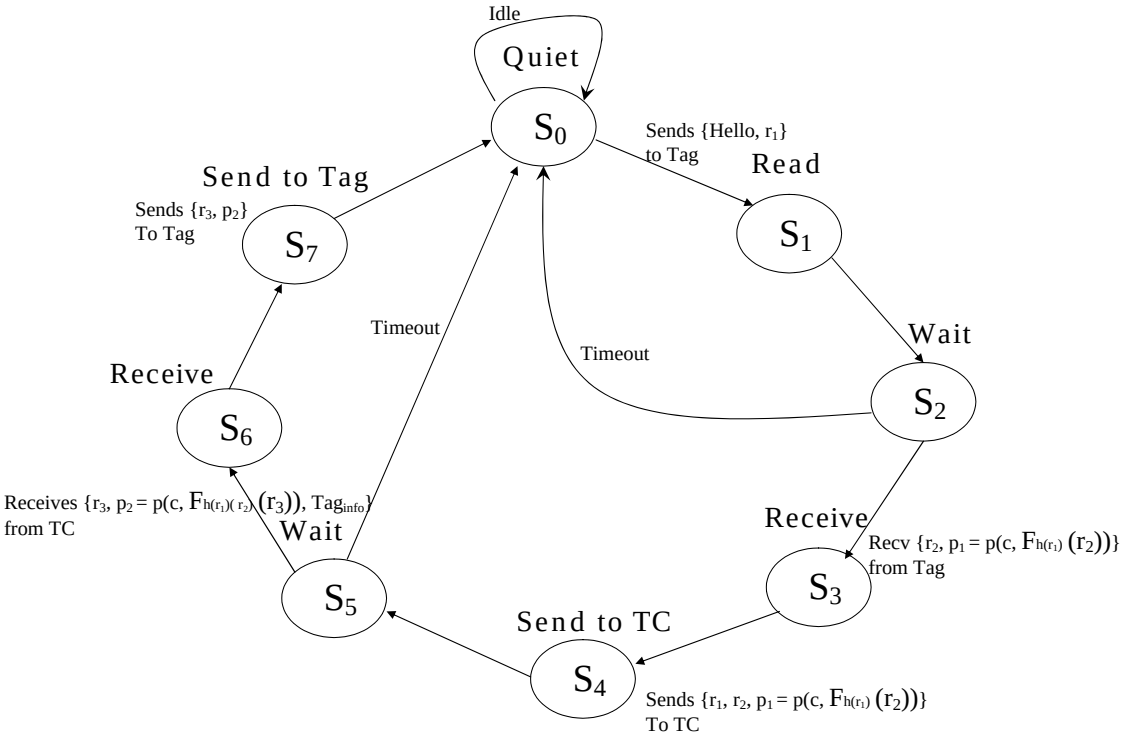

Fig. 9. State diagram for the RFID Reader

erator. It then transmits a HELLO message along with r_1 to a tag in close proximity.

- S_2 : WAIT
  $Wait\ for\ response\ from\ tag(s)$
  $if\ timeout\ reached,\ go\ to\ initial\ state$
  $else$
  $process\ responses\ form\ different\ tags$
  $in\ different\ threads$

After transmitting the HELLO message the reader goes into READER.WAIT state in which it waits for potential responses from nearby tags. The reader may handle responses from multiple tags and process them in different threads. If there is no response within the fixed time interval t, the reader goes into Reader.QUIET state. The description of the reader states that follow are for a single thread.

- S_3 : RECEIVE
  $Recieve\ response\ r_2\ and\ p_1\ from\ tag$

The tag computes a value over the random number sent by the reader and its own random number generated for this read and sends it to the reader. The reader receives two integer values r_2 and p_1.

- S_4 : SENDTOTC
  $Send\ r_1,\ r_2,\ p_1\ to\ TC$

The reader simply forwards the values received from the tag along with the random challenge that it generated for this read r_1 to the TC. The TC then uses these values to authenticate both the reader and the tag.

- S_5 : WAITONTC
  $Wait\ for\ response\ from\ the\ Trusted\ Center$
  $if\ the\ reader\ times\ out$
  $gotoinitialstateS_0$

The reader waits for the response from the TC and times out after a fixed time interval t. This is because if the TC is unable to authenticate the values sent to it by the reader it would not send any response and mark the reader as unauthentic. The TC only responds to authentic readers communicating with an authentic tag.

- S_6 : RECEIVEFROMTC
  $Recieve\ r_3\ and\ p_2\ from\ Trusted\ Center$
  $Also\ receive\ Tag\ information\ from\ TC$

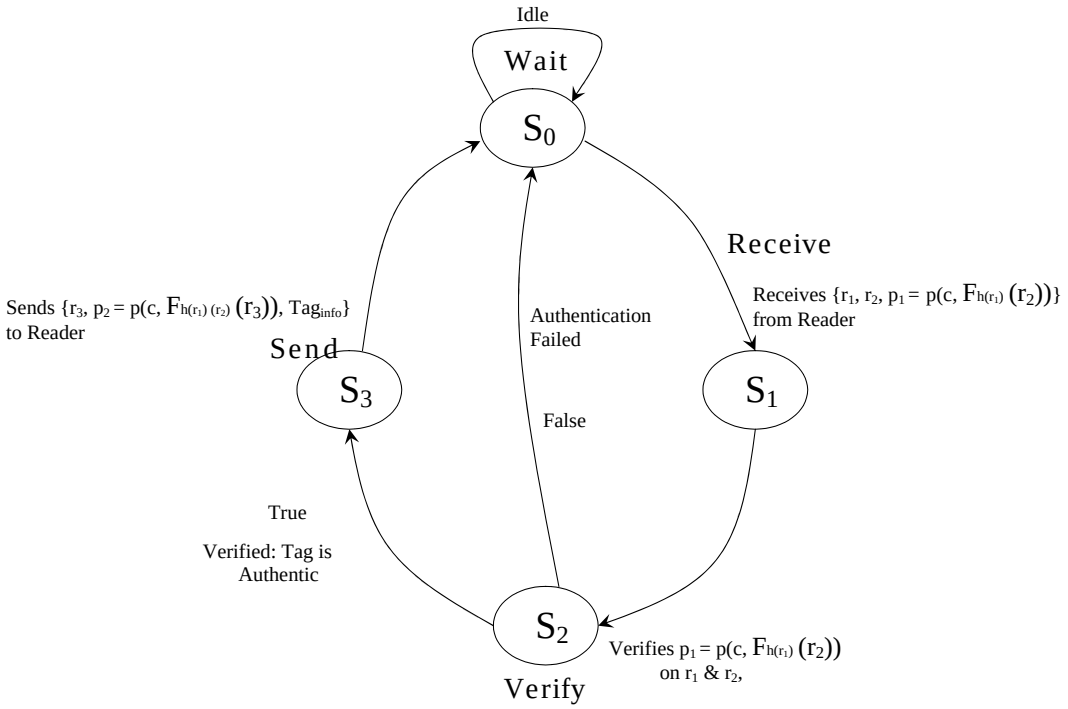

Fig. 10. State diagram for the Trusted Center (TC)

A response from the TC means that the tag is authentic. The reader receives two values r_3 and p_2 from the TC as well as the tag identification.

- S_7 : SENDTOTAG
  $Send\ r_3\ and\ p_2\ to\ tag$

The reader now has an authentic tag and forwards r_3 and p_2 to the tag. This allows the tag to authenticate the reader and update its state after an authentic read. The reader then goes to its initial state S_0.

C. States of the Trusted Center (TC)

The following description of the states of the Trusted Center gives an overview of how it interacts with the reader to accomplish mutual authentication in the RFID system. The state diagram for the Trusted Center is given in Fig. 10.

- S_0 : WAIT
  $In\ Idle\ mode$

The Trusted Center (TC) remains in *idle mode* until it receives a transmission from the reader.

- S_1 : RECEIVE
  $Receive\ r_1,\ r_2,\ p_1\ from\ the\ reader$

It receives three values from a valid reader. r_1 and r_2 are random values generated by the reader and the tag respectively. p_1 is the value computed by the tag using its secret id from the tree of secrets and the values r_1 and r_2.

- S_2 : VERIFY
  $Verify\ p_1\ from\ the\ reader$
  $by\ recomputing\ p(c,\ F_{h(r_1)}(r_2))$
  $on\ values\ r_1,\ r_2\ and\ a\ walk\ on$
  $the\ tree\ of\ secrets$
  $if\ Verify\ fails$
  $Authentication\ failed,\ TC\ moves\ to\ initial$
  $else$
  $Autheticate\ the\ reader,\ and\ the\ tag$

The Trusted Center uses the values received from the reader to authenticate the reader and the tag. It computes the value

$$p_1 = p(c,\ F_{h(r_1)}(r_2))$$

using the random numbers r_1 and r_2 and functions h and F and walking the tree of secrets, thus figuring out the c

value of the tag. If the value is authentic and is within the range of valid *time-delegation* the Trusted Center authenticates the reader and the tag. If the authentication fails the Trusted Center returns to TC.WAIT state without making any further transmissions to the reader.

- S_3 : SEND
  $Generate\ a\ random\ value\ r_3$
  $Compute\ psuedorandom\ value$

$$p(c,\ F_{h(r_1)(r_2)}(r_3))$$

  $Send\ r_3\ and\ \ p(c,\ F_{h(r_1)(r_2)}(r_3))\ to\ reader$
  $Send\ Tag_{info}\ to\ the\ reader$

Once the reader and the tag are both authenticated at the Trusted Center (TC), the TC generates a new random value r_3 and computes

$$p_2 = p(c,\ F_{h(r_1)(r_2)}(r_3))$$

over it and sends r_3, p_2, Tag_{info} to the reader. Tag_{info} if the tag information for the tag stored in the central database which can only be accessed by the Trusted Center. Then the Trusted Center goes into its initial state S_0.

VII. Conclusions

In this paper, we have identified a basis-set of requirements necessary for security and privacy in RFID systems, and then surveyed available research works to observe the extent to which they fulfil the basis-set of requirements. We have then selected the scheme proposed by Molnar, et al [1], and outlined attacks that are possible on the scheme in different realistic scenarios. We have then successfully extended the scheme to mitigate these attacks and meet all the requirements. We have also proposed mechanisms to establish session keys on tags, readers, and trusted centers, to allow secure transfer of data between the entities. These extensions make the overall scheme complete and solves the security and privacy challenges that arise in RFID systems.

References

[1] D. Molnar, A. Soppera, D. Wagner, "A Scalable, Delegatable, Pseudonym Protocol Enabling Ownership Transfer of RFID Tags," Selected Areas in Cryptography, Aug 2005.

[2] L. Bolotnyy and G. Robins, "Randomized Pseudo-Random Function Tree Walking Algorithm for Secure Radio Frequency Identification," Proceedings of the Fourth IEEE Workshop on AutomaticIdentification Advanced Technologies (AutoID), 2005.

[3] M. Feldhofer, S. Dominikus, J. Wolkerstorfer, "Strong Authentication for RFID systems using the AES algorithm," In Proc. CHES, 2004.

[4] D. Molnar and D. Wagner "Privacy and security in library RFID: issues, practices, and architectures," In Proc. Of 11th ACM conference on Computer and Communications Security, 2004.

[5] X. Gao, Z. Xiang, H. Wang, J. Shen, J. Huang, S. Song "An Approach to Security and Privacy of RFID System for Supply Chain" In Proc. Of E-Commerce Technology for Dynamic E-Business, IEEE International Conference, 2004.

[6] International Civil Avaiation Organization ICAO "Document 9303, machine readable travel documents (MRTD)," part 1: Machine Readable Passports, 2005

[7] A. Juels, S. Garfinkel, R. Pappu "RFID Privacy: An overview of problems and proposed solutions" IEEE Security and Privacy, 3(3):34-43, May/June 2005

[8] A. Juels, D. Molnar, D. Wagner "Security and Privacy issues in e-passports" In IEEE CreateNet SecureComm, IEEE 2005

[9] K. Zetter "Feds rethinking RFID passports" Wired News, 26 April 2005.

[10] M. Ohkubo, K. Suzuki, S. Kinoshita "RFID privacy issues and technical challenges," In Communications of ACM, 2005.

[11] M. Ohkubo, K. Suzuki, S. Kinoshita "Cryptographic approach to a privacy friendly tag," In RFID Privacy Workshop at MIT, 2003.

[12] A. Juels, R. L. Rivest, M. Szydlo "The blocker tag: selective blocking of RFID tags for consumer privacy," In Proc. Of 10th ACM Conference on Computer and Communications Security, 2003.

[13] A. Juels "Minimalist cryptography for low-cost RFID tags," In 4th International conference on Security in Communication Networks, 2004.

[14] A. Juels, S. Weis "Authenticating Pervasive Devices With Human Protocols," Advances in Cryptology, CRYPTO 2005.

[15] EPCglobal Inc. "Standards" http://www.EPCglobalinc.org, 2005.

[16] A. Juels, R. Pappu, "Squealing Euros: Privacy protection in RFID enabled banknotes" In Financial Cryptography 2003.

[17] G. Avoine, P. Oechslin, "RFID Tracability: A Multilayer Problem." In Financial Cryptography, 2005.