RFID Protocols and Authentication

Ajit Kumar*[1]

1. Ajit Kumar, Research Scholar, Jagdishprasad Jhabarmal Tiberwala University, Vidhyanagari, Churu Road, Jhunjhunu, (Rajasthan) 333001.
e-mail : ajit.kaswan@gmail.com

Publication Info

Article history :

Received : 14th Oct. 2015

Accepted : 20th Nov. 2015

DOI : 10.18090/samriddhi.v8i1.11414

Keywords :

RFID Authentication, Technology, RFID Innovation

***Corresponding author :**

Ajit Kumar

e-mail : ajit.kaswan@gmail.com

Abstract

Last two decades encountered various technological innovations. RFID is one of these changes. RFID impacted and eased out the day to day life of merchandisers and customers as well. RFID replaced the paper bases system of keeping the records. RFID has proved its importance in various classifications of science society and technology. RFID is a technology that uses radio waves to exchange data between devices. The three main components of an RFID system are readers, tags, and a back-end. The initial objective of RFID was identication of objects. Early applications of RFID systems include tracking systems for farm animals, library items, and airport baggage. Over the years, RFID tags have become more powerful and are used in applications that require more than mere identification.

1. INTRODUCTION

The last few decades have seen a vast number of electronic systems penetrate our daily lives. We use our cell phones to stay in contact with friends and family. Books can be bought from online book stores, paid through an online portal of the bank, and delivered the next day. Paper-based patient records have been re-placed by electronic health records that provide medical sta with patient details whenever and wherever they are necessary. Even everyday items such as public transportation tickets have been replaced by electronic products. These technologies are meant to assist us in daily life, simplify tasks, and make our lives more enjoyable in general. At the same time, service providers embrace new technologies for their potential to increase eciency, to reduce costs, and to bring new business opportunities.

Unfortunately, we often hear about systems whose security is broken. In a typical week in spring 2011 the media reported on the following security incidents:

- Account details of 70 million online gamers were stolen. The details included e-mailaddresses, date-of-births, home addresses, and credit card numbers.
- A centralized database for ngerprints of all Dutch citizens was canceled. The main reason was that the ngerprinting system incorrectly matched one in ve ngerprints against the database.
- Three major mobile operating systems keep track of the physical location of the devices running their operating system. Periodically, these devices send the location data to the vendor without notifying the owner of the device.

These examples indicate that building secure systems is a very complex task. For-tunately, there are techniques to decide whether a system is secure before it is deployed. Formal variation is one of these techniques.

2. RFID SYSTEMS

RFID is a technology that uses radio waves to exchange data between devices. The three main

components of an RFID system are readers, tags, and a back-end. A tag is a small device attached to the object it is meant to identify. It consists of an integrated circuit with memory and processing capabilities and an antenna to receive and send signals. A tag is called active if it has its own power source and passive if it obtains power from the reader. Throughout this work, we concern ourselves with passive tags unless mentioned otherwise. A reader is a device that can detect the presence of RFID tags and communicate with them through radio waves. It has its own source of power and can communicate with the back-end. The back-end is a system that stores and processes information of tags and readers. The communication between back-end, RFID readers, and RFID tags is dened by RFID protocols.

Due to the miniaturization of electronic circuits, RFID tags can be incorporated in almost any other item. The smallest RFID tags are, in fact, smaller than sand particles. With prices starting at a few cents, RFID tags can be manufactured for a relatively low price. Due to their unique properties and the ability to commu-nicate wirelessly without a clear line-of-sight, RFID systems have the potential of becoming ubiquitous.

The initial objective of RFID was identication of objects. Early applications of RFID systems include tracking systems for farm animals, library items, and airport baggage. Over the years, RFID tags have become more powerful and are used in applications that require more than mere identication. At present, RFID systems are found, for instance, in systems for public transportation ticketing, electronic toll collection, and building access control. Most countries issue passports with embedded RFID tags. RFID tags have even been implanted in humans.

2. RFID PROTOCOLS AND UN-TRACEABILITY

2.1 Basic concepts

We start by explaining the main concepts used in our formal model after which we formalize each of these concepts.

A protocol consists of a number of roles. Each of these roles describes the steps that an agent is expected to carry out. There are dierent sorts of agents that can be involved in protocols, such as computers, humans, or RFID tags. One execution of a protocol role by an agent is called a run.

The role speci es the messages that need to be sent and received. These messages are, for instance, encryptions or cryptographic hashes of simpler terms. Among the basic terms, we nd agent names, system-wide constants (such as the natural numbers), and cryptographic keys. An important type of basic term is the nonce, short for 'number-used-once'. Nonces are fresh, unpredictable terms that can be used to ensure that the messages in which they are used are not predictable.

2.2 *Formalization of RFID Protocols and Untraceability*

Protocols embody several constructs to de ne the control ow in an execution. The events describe the actions performed by an agent executing a role. An agent can execute read and send events, in order to read messages from or send messages to the network. There are several ways in which these events can be composed. For instance, one can specify that events have to be executed in sequence, or that one of two events must be executed. The agents in the system have two kinds of memory. The temporary memory contains variables whose values are only accessible to a single run of that agent. The persistent memory contains variables whose values are shared across all runs of that

S-JPSET : Vol. 8, Issue 1, ISSN : 2229-7111 (Print) & ISSN : 2454-5767 (Online)

agent. We call a protocol stateless if it does not update any persistent variable during protocol execution, or stateful if it does. We assume that RFID tags can run only one protocol execution at a time, but RFID readers can run dierent executions concurrently.

We consider an asynchronous communication model where messages are not in-stantly received after they are sent. Following Dolev and Yao [DY83], we assume that the adversary (sometimes called intruder) controls the messages that are being exchanged. This means that he can modify messages, block messages, eavesdrop on messages, and inject messages. When describing an explicit attack on a protocol, we refer to one or more attackers that carry out the attack. The adversary is thus an idealization of the capabilities that a real-world attacker (or set of attackers) might have. Any agent that is not malicious is called honest and it runs the protocol exactly as specied. If an agent does not receive the message he expects to receive according to his role speciation, he simply does not continue the execution.

A security requirement formalizes a security or privacy goal of the protocol. It de-scribes a property that the protocol must enforce when executed by honest agents. An example of a security requirement is secrecy, stating that the adversary cannot deduce a certain message. Another security requirement is untraceability, requiring that an adversary cannot recognize an agent he has previously observed. We call a protocol secure with respect to a security requirement if the adversary has no means to invalidate it. If the adversary can invalidate a claim, we call the protocol awed or vulnerable.

Central to our model is the concept of adversary knowledge. It contains the messages that the adversary has received in the past and all the public knowledge. By combining messages, the adversary can derive new terms. For instance, if the adversary knows a cipher text and the corresponding decryption key, he can de-rive the plaintext. We adopt the perfect cryptography assumption, stating that a cipher text leaks no information about the plaintext if the adversary does not have the decryption key. Furthermore, cryptographic hash functions are assumed to be perfect. That is, the cryptographic hash of a message does not leak any information about the message.

3. RFID PROTOCOL

We use the protocol by Ha, Moon, Nieto, and Boyd [HMNB07] as a running example. We call the protocol HMNB after the last names of the authors. The HMNB protocol is an RFID protocol that aims to mutually authenticate RFID tag and reader, keep the tag untraceable, and resist a particular form of denial-of-service attacks, known as desynchronization attacks. We give formal dentitions of untraceability in Section 3.5, of authentication in Chapter 6, and ofresynchronization resistance in Chapter 7. Furthermore, the HMNB protocol has been designed with limited computational requirements on tags in mind employing a hash function as the only cryptographic primitive.

The protocol assumes that all tags T have an identier ID. This identier is only known to the reader R and tag T. It is updated at the end of a successful protocol execution. Thus the protocol is stateful. The reader also stores the hash of the ID in HID and the value of ID before the last update in ID^0. Therefore, for a system with n tags, the reader stores n tuples (ID; ID^0; HID). The tag keeps track of whether its last run ended successfully or not. For this purpose, the tags use a variable S. If the last run ended successfully, the value of S is 0, otherwise 1.

 S-JPSET : Vol. 8, Issue 1, ISSN : 2229-7111 (Print) & ISSN : 2454-5767 (Online)

We assume that before the reader starts its protocol execution, it does not know the identity of the tag it is about to communicate with. By matching the rst message received from a tag against the list of tuples, the reader can identify the tag. If this procedure is successful, the reader continues the protocol execution and we say that the reader \accepts" the tag. Otherwise, the reader halts the protocol execution and we say that the reader \rejects" the tag.

4. PROTOCOLS

4.1 System model

If an agent starts an execution of a protocol role, a run is created. A run is identied by a run identier chosen from the set of natural numbers. A run contains the agent name of the agent executing the run. It also contains the list of events to be executed by the agent. During execution, events are removed from this list. Finally, a temporary variable assignment is maintained to keep track of the values that are assigned to the temporary variables. A run is thus dened by:

$$\text{Run} = \text{N Agent Ev } (\text{Var}_T \,!\, \text{Term})$$

4.2 Semantics: Protocol Execution

In this section we describe how, through instantiation of variables, an abstract role speciation can be transformed into an execution by an agent, called a run. Furthermore, we de ne how the interleaved execution of a collection of runs de nes the behavior of a system.[1]

For example, we are presenting two sample protocols models.(Fig-1 & Fig -2). Figure 1 presents a sample protocol for for inventory access with selected tags. It is an ideal model for inventory handling.[2]

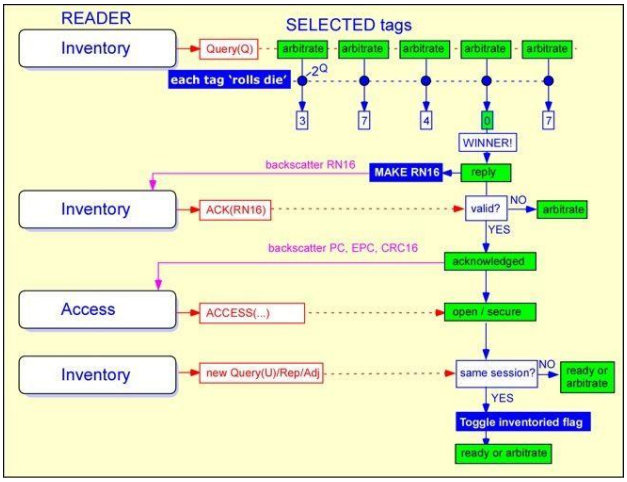

Fig.2: Sample RFID Protocol - 1

Figure 2 presents a sample RFID protocol for Bank. It has been configured for multiple tags and users at same time.

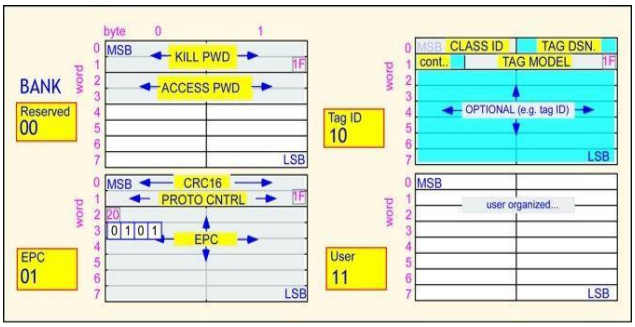

Fig.2: Sample RFID Protocol - 2

5. COMPOSITION RULES

The composition rules model the conditions that are induced by the event composition of the protocol. In the following, let e; e_1; e_2 2 Ev be events, a be an atomic (i.e. send or read) event, and x; y 2 Term be terms. We introduce the event X to denote successful execution of an event. The rules in Figure 3.3 describe the semantics for composing events. The sequential composition rule (seq) species that an atomic event a followed by any event e can always be executed. As specied by the exec rule, an atomic event a can always be executed. If two events e_1 and e_2 are composed using alternative composition, $e_1 + e_2$, then either of the branches can be executed (choice_1 and

S-JPSET : Vol. 8, Issue 1, ISSN : 2229-7111 (Print) & ISSN : 2454-5767 (Online)

choice$_2$). Finally, given two events e_1 and e_2 and two terms x and y, the conditional branching statement e_1 / x = y . e_2 can execute either of the branches.

5.1 Agent Rules

The agent rules describe the eect of the execution of an agent on his run and on the state. We de ne rules for creating a run, terminating a run, sending a message, and reading message.

Creating a run. The create rule models the start of a protocol execution byan agent. The environment in which RFID protocols are run imposes two restrictions on run creation. First, there are two clearly separated sets of agents.

Unless stated otherwise, we assume that tag agents can only execute the tag role of an RFID protocol and reader agents can only execute the reader role. Second, an RFID tag can only run one simultaneous protocol execution. Before a tag can start a run, its previous execution must have nished or been terminated. Readers can run multiple concurrent protocol executions. In order to faithfully model RFID communication, the semantics of the create rule must allow both restrictions to be enforced. Otherwise, the semantics would allow one to derive attacks that cannot be executed in a deployed RFID system.

5.2 Semantics: Protocol Execution

A received term m is readable with respect to a pattern p if there is a substitution that makes them syntactically equivalent. Furthermore, every sub term of the received message must be inferable from the agent's knowledge or from the received message itself. We rst give the formal dentition of the sub term operator. It is used to decompose a term into the terms from which it was constructed.[3]

Figure 3 & 4 presents a semantics execution of Protocol for B2B. These model introduce an approach towards the binding private and public processes to be implemented as protocols for B2B and workflow types. These presented models also presents the inter – enterprise collaboration management approach.. [4,5]

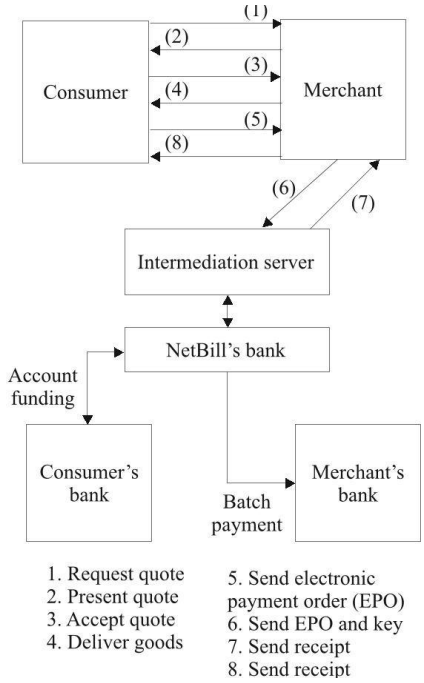

Fig.3: Sementic Execution of Protocols - 1

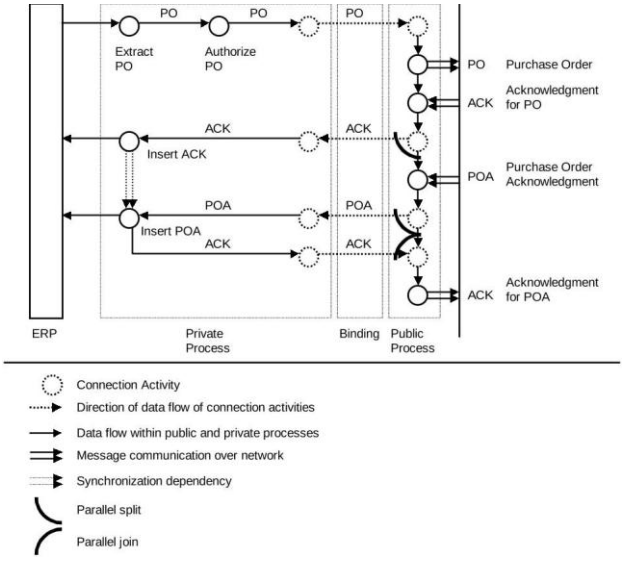

Fig.4: Sementic Execution of Protocols - 2

6. CONCLUSION

- Throughout this thesis we have applied formal verication to analyze the security of RFID systems. As a rst step, we have created a model for analyzing RFID protocols (see Chapter 3). The model includes a language for describing RFID protocols and allows one to systematically derive the execution traces of the protocol. As such, it provides a rigorous way of describing attacks and proofs.[4]

- Within the model we have expressed the security requirement of untraceability. Our de nition very closely resembles the intuitive de nition of the property, i.e. a protocol is untraceable if an attacker cannot recognize a previously observed tag. We have shown how to apply our model to analyze the untraceability of existing RFID protocols.
- We have analyzed a large number of existing RFID protocols with untraceability claims. Many of these protocols do not satisfy untraceability. It turns out that techniques to attack one protocol can often be used to attack another protocol. We have, thus, been able to classify attacks on untraceability into a number of cat-egories (see Chapter 4). This classi cation can be used as a reference for common attacks, so that new RFID protocol proposals do not su er from the same aws.

REFERENCES

[1] "The role of B2B Protocols in Inter – Enterprise Process Execution", Christoph Bussler, Oracle Corporation, Redwood Shores. 16-29

[2] Contract – Driven creation and operation of virtual enterprise, Hofner Y et al, Semantic Scholar, https://www.semanticscholar.org/paper/

[3] Nalluri, S. K., & Parasaram, V. K. B. (2015). Automating Software Builds with Jenkins: Design Patterns and Failure Handling. International *Journal of Technology, Management and Humanities*, 1(01), 16-33.

[4] "Operational and Semantic Protocols" Cas Cramer et al. Springer Book, 23-34.

[5] https://www.utdallas.edu/~zxl111930/fileAutoFormat_NDSS08.pdf

