ORIENTAL JOURNAL OF
COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
Published By: **Oriental Scientific Publishing Co., India.**
www.computerscijournal.org

ISSN: 0974-6471
March 2017,
Vol. 10, No. (1):
Pgs. 127-134

RFID Security Issues in IoT: A Comparative Study

DENVER BRAGANZA* and B. TULASI

Department of Computer Science, Christ University Bangalore, India.
*Corresponding author E-mail: denver.raphael@cs.christuniversity.in

http://dx.doi.org/10.13005/ojcst/10.01.17

(Received: February 28, 2017; Accepted: March 15, 2017)

ABSTRACT

The landscape of Internet of Things (IoT)has been evolving atan increasing rate over the recent years. With the ease of availability of mobile devices, there has been a tremendous leap in technology associated with it. Thus, the need for efficient intercommunication among these devices arises. To ensure that IoTis seamlessly integrated into the daily life of people using appropriate technology is essential. One of the important associated technologies with IoT is RFID. RFID proves to be a simpler and efficient technology to implement IoT at various levels. Since IoT is greatly imapcting the lives of people, one of the major concerns of IoT is the security. IoT will have millions of devices and users connected to each other. It is important to authenticate both users and devices to prevent any breach of information. With the limitations in RFID technology, various authentication protocols have been developed to provide optimal solutions.

Keywords: Internet of Things, RFID, Security issues, Privacy concerns, Anonymity, Authentication protocols.

INTRODUCTION

Basic IoT Concept and the Widespread of IoT

The Internet of Things (IoT) is a concept that came into being in 1999. It consists of a variety of objects that are interconnected and that can communicate with each other by sending and receiving relevant data. With the size of devices getting smaller and smaller, the Internet of Thing has been gaining more importance lately. Developers are focusing on connecting a large number of things and giving them numerous capabilities that make the Internet of Things a very exciting concept[4]. While the focus is on its development, security in the Internet of Things is being overlooked. The future of the Internet of Things looks very promising and will probably have every object in the world connected with each other[9]. Though it might not seem to be much of a concern now, privacy of connected objects should be maintained at early stages to prevent misuse of the Internet of Things at later stages[15].

Technologies used in IoT

In today's world, every device is preferred to be kept wireless. Since the Internet of Things connects every device to each other, wired technology would obviously not be a feasible option.

Though IoT still uses wired interfaces to connect objects,using wireless technologies to do the same has become a trend. There are many technologies used to enable the Internet of Things.

The enabling technologies used in IoT are RFID (Radio Frequency Identification), NFC (Near Field Communication), Li-Fi, Optical Tags and QR codes, Bluetooth Low Energy, Low energy wireless IP networks, ZigBee, Z-Wave, Thread, LTE Advanced, Wi-Fi Direct, HaLow, HomePlug, MoCA and Ethernet.

Issues in IoT Technology

The Internet of Things is a promising technology but a very complicated one on the backend[14] and therefore faces quite a few problems still waiting to be solved.

The problems faced by IoTare:

- User consent – users need to give permission to devices to allow them to collect data but they may not have the time or technical knowledge[3].
- Freedom of choice – the privacy protection and the current structure of IoT should provide freedom of choice to users[3].
- Anonymity – As of now IoT pays less information to user anonymity. Users should be provided with completely anonymity the way TOR browser provides anonymity[3].

The Internet of Things has to gain the trust of people so that it can be deployed faster[1]. To do this, the security of the IoT technologies being used should be tightened[13].

RFID

With low to no security in the Internet of Things, a tap into one of the devices can give a hacker access to all connected devices, thereby revealing all information that could be confidential and private to the user[8]. In this paper, one of the most commonly used technology in the Internet of Things is discussed. This is the RFID technology which consists of RFID tags and readers, and an optional database server. RFID stands for Radio Frequency Identification. RFID is mostly used in supply chain management to keep track of goods being transported and sold. But now it has found better use and is preferred over other technologies in IoT[5] since it is cheaper and uses much less energy.

1) Applications of RFID

RFID is a technology less heard of when compared to other technologies but is very useful in IoT. RFID tags are used in IoT and are either embedded or tagged onto devices. There are 2 types of RFID tags, active RFID tags and passive RFID tags. Active RFID tags are very costly and need to be powered by an externa source to operate. But they provide a much larger read range and a larger storage capacity than passive tags and can also broadcast their own signal. Passive RFID tags on the other hand are very cheap. They run on the energy transmitted by the RFID reader and work within a shorter range as compared to active tags. Since passive RFID tags are cheaper and consume less energy they are widely preferred in the IoT[19]. Passive RFID tags can be classified under 3 categories:

i) Low Frequency tags
Low Frequency tags work within a range of 30 cm or less.

ii) High Frequency tags
High Frequency tags work within a maximum range of 1.5 meters.

iii) Ultra-High Frequency tags
Ultra-High Frequency tags work within a range of 1 meter to 15 meters.

2) Drawbacks of RFID

As IoT advances, so should its security. A number of security issues have been identified and will be discussed in this paper. Various security protocols need to be implemented to maintain the privacy of data being shared across IoT devices. Although there are quite a few security protocols that have already been implemented, they have to be improved upon to provide maximum security. The problem with RFID tags is that they have very limited storage space. This makes it very difficult to implement full-fledged security protocols onto such RFID tags[11]. One solution is to increase the storage capacity of the RFID tags. This solution may seem easy but is not feasible since increasing storage capacity would also mean increasing the amount of power given to the tags. Another solution is to implement lightweight security protocols on such tags that will cater to the security needs of only that type of data that is being transferred to and from these tags.

Security Issues in RFID

Just like other wireless technology, RFID has quite a few security and privacy risks for both manufacturers and customers. An RFID system has to be as secure as possible considering all security risks[17]. Maintaining user privacy is a major concern when RFID is a part of IoT. But it should be clear that it is practically impossible to maintain a perfectly secure system. Once this is understood, it is possible to list down all the security and privacy concerns of a given RFID system. For the public to accept a RFID-based Internet of Things technology strong technical and operational along with strong security and privacy solutions should be in place.

A number of security issues have been identified. These security issues have been given as follows:

a) Jamming

Jamming is the process of paralyzing the air interface between an RFID reader and a tag thereby preventing the communication between the reader and the tag[16]. Since the communication medium is air, it is very simple to disrupt it. An attacker generates radio noise at the same frequency as that used by the RFID system and that in turn prevents communication within the system.

b) Eavesdropping

Eavesdropping is an attack where in the attacker uses a fake reader to get information being passed between the original RFID reader and tag. Most RFID readers and tags use simple non-encrypted text communication because of their limited storage and capabilities[16]. This makes it easier for the attacker to retrieve information easily using the fake reader. This information can easily be used to manipulate the data and also for replay attacks. The fake reader has to communicate at the same frequency as that of the original reader in order to get the information.

c)Replay Attack

Replay attacks are those attacks where an attacker eavesdrops on a particular RFID system, records the details being sent to and from the reader and the sender and then replicates the data being passed to act as either the original reader or the tag[12].

d) Deactivation

This is an attack in which the attacker makes the RFID system useless by sending either delete or kill commands to the tag[16]. This will either make the reader unable to identify the tag or even detect the presence of the tag even though it is in range.

e) Detaching the tag

In this type of attack the tagged items can be switched with another item making the reader think that it is the same original item[16].

f) Spoofing

In spoofing the attacker tries to understand the security protocol used in a particular RFID system. With this information, the attacker writes the received data with the same format to blank RFID tags[16]. This is called duplication of tags. This can be used to change information that isn't validated by the reader like price of an item.

g) Man-in-the-Middle attack

The Man-In-The-Middle(MITM) attack is an attack where in the attacker places a fake reader between the original reader and the tag. The fake reader receives the fake information from the original reader and the tag respectively manipulates the information and sends it back to the original tag and reader respectively[16]. All this is done during the transmission of data and is therefore a real-time attack.

h) Cloning

In cloning the information of an original tag is copied to a blank new chip to replicate it[16]. Cloning is generally categorized with spoofing but cloning and spoofing are not the same. Spoofing emulates the tag data being transmitted while cloning copies the data onto a new tag owned by the attacker.

Security Protocols

To counter and defend such attacks many protocols have been developed. The protocols that have been developed have been designed in such a way that they fit on the limited storage available on RFID tags as well as maintain the privacy of the users at the same time. Many such protocols like O-TRAP, A-TRAP, O-FRAP, O-FRAKE, YA-TRAP, etc. have been designed for specific purposes trying to maintain privacy to a great extent. But unfortunately, all of them have been found to have some security loop holes that have to be take care of. Some of these protocols will be discussed later on in this paper.

Study of the Protocols

In this paper, someRFID protocols will be studied because they are very similar in their implementation and working. These protocols will be compared. Each of the protocols will be described below.

O-TRAP

O-TRAP is an abbreviation for Optimistic Trivial RFID Authentication Protocol. This protocol is said to be optimistic because its overhead is minimal when it is not under attack[10]. In this protocol, it is assumed that all RFID readers that are authenticated are connected to a server at the backed through a communication channel that is secure. Each RFID tag stores a private long term key, $k_{tag,}$ and a (constantly changing) value, r_{tag}, that is updated each time the tag is challenged. The long-term key is shared with the back-end server. The server has a database, D, in which for each tag it stores pairs of values. One of the value in the pair is the private long term key, k_{tag},and the other value is the constantly changing value, r_{tag}. Each pair of values is identified by $r_{tag,}$ from among the pair of values.

At regular intervals, the server generates a random string, r_{sys}, that it broadcasts to all tags that are in reach. When a tag is activated by an RFID reader,it computes 2 values by applying a fixed function to get a random value taking the long-term key, k_{tag}, as one parameter and r_{tag} or the r_{sys} as the second parameter. The first value, v_1, is used for the updation ofr_{tag}, while the second value, v_2, is required for the authentication tag. While the reader is passive, the long term private key, k_{tag}, can be retrieved from the database, D, by the server by using r_{tag}. It can then verify the correctness of the tags response and update r_{tag}, corresponding to the long term private key, k_{tag},stored in D. In this case the cost for both the server and the tag is the use of just one function to get a random value.

If a malicious reader has recently attempted to attack a tag, the values stored on the tag will not be synchronized. In such a case, all the long term private keys, k_{tag}, will have to be scanned by the serverto locate the correct long term private key, k_j, and update its corresponding value with the newly computed value, v_1 in the database D.

During attacks, extra computational costs are borne only by the server. Performing computation on the tags is avoided as much as possible. Note that, during the time of interrogation the randomly generated string remains the same to all the tags in the range of the RFID reader. During this time, the server maintains a record of all the tag replies and the reject replays. Tags that are authorized will give a different reply each time. To refrain from having a very long list of replies the server can control the interrogation period.

A-TRAP

A-TRAP stands for Absolutely Trivial RFID Authentication Protocol. It is a protocol used to secure against fly-by attacks. Fly-by attacks are those attacks in which the attacker attempts to attack the tag for a very short duration of time. To be specific a time duration less than $2^m t_0$ time units[18]. m is the number of times the tag is interrogated. A Pseudo Random Generator (PRG) and a Time Delay Scheduler (TDS) are required.

The delay between every authentication session is controlled by the TDS which is a hardware module. The time delay after each successful authentication is very little, about t_0. They end with the update of the tag's key. With each incomplete session, the time delay is multiplied by 2. The time delay will be $2^m t_0$ after m successive incomplete sessions. When an attacker attempts to modify, the values being passed between the server and the tag by triggering incomplete sessions, the TDS is used to prevent such attacks. The number of time-delay doublings is limited. Capacitors are used to gather just the right amount of energyin order to run the protocol and/or the counters. The tag is turned off at the time of these delays. Although there is a limited amount of power still sustained to run the counter. The clock rate is reduced just enough to run the counter. Because of this a delay can be extended to a great extent.

In A-TRAP, 3 values v_1, v_2 and v_3 are generated by a pseudo-random generator (g_{tag}) which are exchanged by the tag and the server[18]. If the value received, that is g_{tag} exists in the Database at some value $d_{i,j,}$ then the authentication of the tag is considered to be successful. F so the i^{th}row of the database D is updated by: (a) removing the first j entries, (b) moving the rest of the entries to the to the start, and then (c), passing the next j values $g_i(1)$, . . . , $g_i(j)$ extracted from the PRG$g_i$to the empty cells. If the received value ($g_{tag)}$ is not in the database, the authentication of the tag is considered unsuccessful[18]. A slightly different version of A-TRAP generates an extra value v4 using the PRG(g_{tag}) to achieve authenticated key.
A-TRAP protocols are not very efficient against desynchronization attacks.A tag that is attacked more than m successive times will be invalidated permanently. But against a fly-by attack, A-TRAP protocols prove to offer more secure authentication, availability, forward-anonymity, and key indistinguishability[18]. The A-TRAP protocols cannot prove helpful if a tag is captured during an attack, but can prevent attackers who secretly desynchronize the tag.

O-FRAP

O-FRAP is an abbreviation for Optimistic Forward-secure RFID Authentication Protocol. A pseudorandom generator is used to generate two values r_{sys} by the server and r_{tag} by the tag (for optimally identifying of the tag), to make the session anonymous and to prevent replays. The server updates the tag's current key $k_{tag}{}^{a}$ after authentication of the tag, and the tags updates the key authentication of the server.
When the server activates a tag, four values ν_1, ν_2, ν_3, ν_4 are computed by applying the pseudorandom function F to ($k_{tag}{}^{a}$, $r_{tag}||r_{sys}$). The following convention is used: If the sender writes the value x to a channel, the receiver reads it as x'. The value x' may differ from x if the attacker corrupts it during transmission[18].

In O-FRAP, just like in O-TRAP, ν_1 is used to update the pseudo-random value r_{tag}; ν_2 is used for authentication of the tag; an extra value ν_3 is used for authentication of the server; and an extra value ν_4 is used to update $k_{tag}{}^{a18}$. In these protocols, the following convention is used: after using the pseudo-random function F with parameters (k_j^a, r_{tag} || r_{sys}), the four values the server computes are denoted by ν_1^*, ν_2^*, ν_3^*, ν_4^*. When the attacker is passive, these values correspond to v_1, v_2, v_3 and v_4. To be exact, $\nu_2^* = \nu_2^{'}$ and $\nu_3^{*'} = \nu_3$, and the server and tag accept the tag by giving out a ACCEPT flag.

After each server authentication, the tag key $k_{tag}{}^{a}$ is updated. This gives very distinguishable properties between each session. A tag that is successfully attacked by an attacker, cannot be used to link to the records of previous sessions. This proves there is forward-anonymity.

YA-TRAP

YA-TRAP is an abbreviation for Yet Another Trivial RFID Authentication Protocol. It has been described by Tsudik. He aimed this protocol to be run at environments where the information related to the tags is dealt with in batches as compared to applications like tagging of individual customer items or access control[2]. He has still not worked on the security issues that may be present in YA-TRAP formally, but he is currently working on it.

In the YA-TRAP protocol, the RFID reader shares a unique key x_i with a tag T_i. The tag $T_i$maintains an internal timestamp which keeps track of the time a reader tried to access its information. A reader sends the current time in order to interrogate a tag T_R. t_R is compared with t_i within the tag. If T_R is old when compared to T_i, i.e. T_R<= T_i, the tag then gives out a random response. If not, then the tag gives out $R = H_{xi}[t_R]$. H_{xi} is a HMAC (a keyed-hash message authentication code) computed using x_i. The tag also updates the timestamp from t_i to t_R. The reader checks if $R = H_{xj}[t_R]$ for any secret key x_j in its database to validate the response of the tag. If it is true the reader accepts the tag, otherwise it rejects it.

There are two attacks strategies that can be tried on YA-TRAP which prove to show that this protocol fails. The first one is denial of service. The attacker tries to "mark" a tag by querying it with a time t_{max}, where t_{max} is some time in the far future. The tag sets its internal timestamp t_i to the value of t_{max} and therefore outputs random values to all future queries. Thus, the RFID reader does not accept the tag in any of the future sessions. This lets the user distinguish this tag from the tags that are yet "unmarked".

The second attack strategy also attempts to "mark" a tag, but its goal is not to simply invalidate it. To do this the attacker selects a tag T_i and queries it with a future time u_i. This causes t_i's value to be set to u_i. u_i acts as a distinguished "mark" for tag T_i. At any time before ui, the attacker can check whether any tag T is the same as the "marked" tag T_i. This involves two steps:

Probing

The attacker selects two times u_i^{before} and u_i^{after} such that u_i^{before} is time slightly before u_i and u_i^{after} is a time slightly after u_i. The attacker then queries the tag T with u_i^{before} and u_i^{after}, obtaining 2 responses r^{before} and r^{after}.

Testing the results

The attacker interacts with R at time u_i^{before} and u_i^{after} and it replays responses. If it accepts r^{before} and rejects r^{after}, then it is almost definite that T is the same as T_i.

In the setting for which YA-TRAP was developed, wherein a reader collects tag information in batches to be processed by a backend server that does not reveal information about tag identification to attackers, YA-TRAP does not work for the attacks described above.

Burmester and Munilla Protocol

The protocol proposed by Burmester and Munilla is a lightweight mutual authentication RFID protocol. It supports session unlinkability and forward and backward security[6]. Unlinkability ensure that the data being sent cannot be linked to the target tag. Forward security ensures that even if the long-term key is compromised the future session keys cannot be compromised. Forward security ensures that even if the long-term key is compromised the past session keys cannot be compromised.

A synchronized PRNG is shared with the server by each tag. Tag and server authenticate each other by exchanging three or five consecutive numbers from the PRNG. The PRNGs state can be reset if it is suspected that it has been compromised.

The original EPC-C1G2 protocol has four passes for identification[7], which involve the exchange of the following messages: a query, a random number RN 16 (16 bit), an acknowledgement ACK (RN 16) and the EPC data. These values have been replaced by three random numbers (RN 1, RN 2 and RN 3) in the so-called optimistic case. If RN 1 was previously used (a flag called alarm is ON), after which two more nonces (RN 4 and RN 5) have to be exchanged. Nonces are numbers, bits or strings that are used only once.

Table 1: Comparison of RFID Security Protocols

Protocol	O-TRAP	A-TRAP	O-FRAP	YA-TRAP	Burmester and Munilla Protocol
Tag Anonymity	Y	Y	Y	Y	Y
Replay Attack Resistance	N	Y	Y	Y	Y
De synchronization resistance	Y	N	Y	N	Y
Confidentiality and integrity	Y	Y	Y	Y	Y
Forward secrecy	Y	Y	Y	N	Y
MITM Attack resistance	Y	N	Y	N	N
DoS Attack resistance	N	N	Y	N	N

Comparisonof the Protocols

Now that the protocols have been discussed, it is easy to identify what they do to maintain security in RFID technology and also how they do it.

From the protocols studied above, it can be said that the protocol O-FRAP proves to be very effective in terms of security of RFID technology for most cases. It can also be seen that the other protocols have do not provide when it comes to some types of attacks. Most of these protocols are designed for a specific purpose. While focusing on this purpose for which they were intended these protocols may overlook some of the security issues.

If one had to choose a particular protocol, it would be beneficial to choose one based on the purpose for which it was intended. In certain cases, some of these security loopholes may be overlooked because such loopholes might not be relevant for the purpose at which it is intended.

Jun-Ya Lee, Wei-Cheng Lin, Yu-Hung Huang, O-TRAP is a general RFID authentication protocol that proves to be very useful. Although it may not be able to prevent replay attacks, it can correct the damage caused by the replay attacks by replacing unexpected values with the right ones. It also does not play any role in protecting against Denial of Service attacks. O-TRAP is considered to be a lightweight protocol as long as there are no attacks on it.

A-TRAP is used for attacks that last a very short time. This protocol doesn't expect the attacker to be around the tag for a long time. It protects against attacks known as fly-by attacks. Against such type of attacks, A-TRAP provides tag anonymity, resistance to replay attacks, maintains confidentiality and also forward anonymity. It fails against de-synchronization attacks, man-in-the-middle attacks and Denial of Service attacks.

O-FRAP is by far the best from all the five protocols compared. Though not excessively tested it proves to prevent all of the mentioned attacks. It is also not aimed at a specific purpose but at a general purpose. This makes it the most suitable protocol for RFID security among the rest specified.

YA-TRAP is aimed at RFID technology that processes information in batches. It is known to maintain tag anonymity and confidentiality It also can prevent replay attacks. But it fails badly against the rest. It is not recommended for RFID technology that involves processing of single tags at a time.

The Burmester and Munilla Protocol is also a very good protocol that maintains tag anonymity, confidentiality and forward anonymity and also prevents replay attacks and desynchronization attacks.

It cannot prevent man in the middle attacks and Denial of Service attacks. It still proves to be a very useful protocol.

Table I lists all the protocols mentioned above and clarifies which kind of attacks they can handle and which they cannot.

CONCLUSION

RFID is being associated with the IoT in various sectors. RFID can be seen as an integral part of the IoT. There are quite a few concerns with RFID, authentication protocols being one among them. The need to strengthen these authentication protocols and identifying optimal one for usage is immediate requirement.

Different security protocols are required to be embedded into RFID technology based on the purpose they are being used for. The five protocols discussedin the paper can be classified based on their purposes. The protocols being targeted at specific purposes like A-TRAP and YA-TRAP have security issues and need to be fixed before they are deployed. While, the other general purpose RFID protocols like O-TRAP, O-FRAP and the Burmester and Munilla protocols need to be well tested before they are deployed.

Storage space is a concern in RFID. For this reason, lightweight security protocols need to be developed for it. With safer and secure RFID technology, the Internet of Things will be more trustworthy and its spread will be much faster.

REFERENCES

1. Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, ImrichChlamtac, 2012, Internet of things: Vision, applications and research challenges, *Ad Hoc Networks*, **10**(7), pp. 1497–1516.
2. Md. Mahmud Hossain, MaziarFotouhi, RagibHasan, 2015, Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things, pp. 21 – 28.
3. Rolf H. Weber, 2010, Internet of Things – New security and privacy challenges, **ScienceDirect**, **26**(1), pp. 23 – 30.
4. Rodrigo Romana, JianyingZhoua, Javier Lopezb, 2013, On the features and challenges of security and privacy in distributed internet of things, *ScienceDirect*, Volume **57**(10).
5. Charith Perera, Rajiv Ranjan, Lizhe Wang, Samee U. Khan, Albert Y. Zomaya, 2014, Big Data Privacy in the Internet of Things Era, *Research Gate*, **17**(03), pp: 32 – 39.
6. Arbia Riahi, YacineChallal, Enrico Natalizio, ZiedChtourou, AbdelmadjidBouabdallah, 2013, A Systemic Approach for IoT Security, IEEE, pp.351-355.
7. Qi Jing, Athanasios V. Vasilakos, Jiafu Wan,Jingwei Lu, DechaoQiu, 2014, Security of the Internet of Things: perspectives and challenges, *Wireless Networks*, **20**(8), pp 2481-2501.
8. Jung Tae Kim. 2014, Privacy and Security Issues for Healthcare System with Embedded RFID System on Internet of Things, *Advanced Science and Technology Letters*, **72**, pp.109-112.
9. Evan Welbourne, Leilani Battle, Garret Cole, Kayla Gould, Kyle Rector, Samuel Raymer, Magdalena Balazinska, and Gaetano Borriello, 2009, Building the Internet of Things Using RFID, *IEEE Internet Computing,* **13**(3), pp. 48 - 55.
10. Tuhin Borgohain, Uday Kumar, SugataSanyal, 2015, Survey of Security and Privacy Issues of Internet of Things
11. Nidhi Desai, Manik Lal Das, 2015, On the Security of RFID Authentication Protocols, Electronics, Computing and Communication Technologies (CONECCT).
12. Tieyan Li, Guilin Wang, 2007, Security analysis of two ultra-lightweight RFID authentication protocols, *IFIP International Federation for Information Processing*, **232**, pp. 109 – 120.
13. Sangita Mohite, GurudattKulkarni, Ramesh Sutar, 2013, RFID Security Issues, *International Journal of Engineering Research & Technology (IJERT),* **2**, (9), pp. 746 – 748.
14. Prajnamaya Dass, Hari Om, 2016, A secure authentication scheme for RFID systems, *Science Direct*, **78**, (1), pp. 100 – 106.
15. Mike Burmester, Breno de Medeiros, 2007, RFID Security: Attacks, Countermeasures and Challenges, The RFID Journal Conference.
16. Tri Van Le, Mike Burmester and Breno de Medeiros, 2007, Forward-secure RFID Authentication and Key Exchange, IACR ePrint.
17. Ari Juels, Stephen A. Weis, 2009, Defining strong privacy for RFID, ACM Transactions on I*nformation and System Security* (TISSEC), **13**(1).
18. Honorio Martín, Enrique San Millan, Pedro Peris-Lopez, Juan E. Tapiador, 2013, Efficient ASIC Implementation and Analysis of Two EPC-C1G2 RFID Authentication 2014, A Lightweight Authentication Protocol for Internet of Things, Next-Generation Electronics (ISNE).