Security problems of RFID authentication protocols

VALDIS PORNIEKS, EGILS GINTERS
Sociotechnical Systems Engineering Institute
Vidzeme University College
4 Cesu Street, Valmiera LV-4200
LATVIA
http://www.va.lv

Abstract: - The RFID market in Europe is growing. It is expected that in 2007 the market exceeds 2.5 billions USD. Unfortunately, despite of benefits of the technology some problems remaining. One of them is security. Many different attacks have been described in theory and tested practically, but it is difficult to find the perfect secure solution for a wireless unit that has limited computing abilities because of cost constraints. On the way to find the suitable solution, this paper describes different attack types and analyses the ability of two proposed authentication protocols to resist these attacks.

Key-Words: - Radio Frequency Identification, RFID, Logistics, Data security, Cryptographic algorithms, Authentication

1 Introduction

Radio Frequency Identification (RFID) dos not a new rather than emerging technology. It was developed almost 30 years ago for the automated remotely identification of objects. For adding an identity to an object, a RFID tag (transponder, label etc.) must be attached to it [1, 2]. The tag is built from an antenna and a microchip with some limited processing abilities and storage capacity for a unique identification number (ID) and optionally some free space for user-specific needs. The RFID reader (scanner) sends a query to the tag using alternating electromagnetic field (see Fig.1).

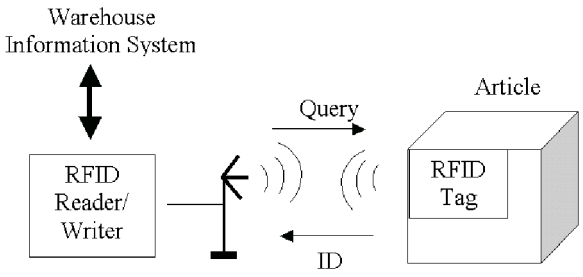

Fig.1. The physical structure of RFID solution

The field generates the induction current in the antenna contour of the tag. Otherwise, the induction current going through the coil of the antenna generates the response signal with appropriate frequency and ensures power supply for the microchip.

By response of the tag, the user receives the unique ID of the object and can lookup additional information about it in a database.

Because enhancing manufacturing technologies ensures lower size and power consumption of RFID tags then introduction capacity is growing rapidly, which reduces the prime costs and promotes introduction coverage more extensive.

RFID is going to be the replacement for barcode identification systems as it has some significantly better features. Most of RFID readers do not need line-of-sight to get the information from tags. Tags can be read much faster, almost simultaneously, so the time for scanning a batch of items can be reduced to seconds in comparison with barcodes where each item has to be scanned separately. Finally, the RFID tag can hold much more information than a barcode. All mentioned above promote RFID use for developing of the quality of the logistics services and needs of other businesses. Unfortunately, RFID still have the problems related with standardization and data security [3, 4].

2 Classification of attacks

Sometimes new technologies, especially wireless, give new space for unethical persons to attack the users of the technology, as is also the case with RFID. There are different kinds of attacks that can compromise the availability or integrity of data or the privacy of the users of RFID. Let us look at some known attack types [5]:

Unauthorized tag disabling - or Denial-of-Service (DoS) attacks make tags or the contained information temporarily or permanently inaccessible. Such attacks can affect accounting systems like inventory or logistic applications, where tags are used to determine presence of goods. If the tags become disabled, then the inventory must be done manually that costs time and money. In addition, shoplifters might want to disable tags so they can carry them out of the store without being noticed by the RFID readers.

Unauthorized tag altering - these attacks can have even worse implications as disabling, for example if the product codes for drugs are exchanged, human lives can become endangered.

Unauthorized tag cloning - tags can be accessed by distant readers and the information gained from them can be used to create a tag that sends identical information upon request. Tag cloning most likely would be used to overcome RFID enabled counterfeit protection. Thus, companies must pay close attention on how to secure their tags against cloning. Otherwise, the tag would be just another component to add for the perfect counterfeit.

Unauthorized tag tracking - these attacks have different goals than the previously mentioned. Instead of attacking the data integrity, tags are used to gather information about their users or owners. A person's movements [6] or even choices can be analyzed by associating a tags ID with this person and analyzing where and when the tag has been scanned or what other information has been associated with it.

Here the RFID technology enables adversaries' possible access to the tags information, as this can be done from distance and it is impossible with out any special devices for the owner of the tag to determine when and by whom his tag is read.

Man-in-the-middle attacks - an adversary attacks the data integrity by impersonating a legitimate reader to obtain a legitimate tag's response to a challenge, so he can later on pretend to be the legitimate tag. Such attacks are often used to gain access to secured areas. Again the problem is that people are not able to see of feel who and when is accessing their tags.

Side-channel attacks – these attacks gained information from the physical implementation of cryptosystem, to calculate the secret values hidden in it [6]. To perform these attacks the adversary needs sophisticated knowledge and tools, but if the potential feasibility of such attacks is high enough, then it is quite likely that they will also be performed.

Most of the attacks can be practically prevented if cryptographically secure mutual authentication between the tag and the reader is provided. Security never comes free and adding cryptographically enabled mutual authentication features to a tag will require more calculation power than is available in the cheapest tags, this means once more a cost increase. To reduce the cost to a minimum a light-weighted algorithm must be selected that is secure enough so that it is not feasible for the attacker to spend his time and energy to crack it.

Many authentication protocols exist as LCAP, CRAP, OHLCAP, AES, HIDV, and other [5, 7]. Further will be compared only two of them.

3 OHLCAP authentication protocol

3.1. OHLCAP fundamentals

This is a protocol proposed by Eun Young Choi et al. [8]. The authors claim it to supply mutual authentication and anonymity against adversaries.

The structure of the protocol is as follows. The protocol consists of two phases – the set-up and mutual authentication phase. At the setup multiple values are written to the tag and the database:

ID – the tags ID is written to the tag (*l*-bit string);

S – the secret value of the separate tag (*l*-bit string);

GI_i – all tags are logically grouped in n groups, each tag receives a group index (*l*-bit string);

K – the secret value for all tags (*l*-bit string);

c – a random initial value for a counter set only to the tag.

At the authentication phase the following protocol is executed (see Fig.2), where + - addition operation; - exclusive-disjunction (*xor*); || - concatenation of two strings:

The Reader sends a *Query* with a random value r to the tag;

If r = 0 the communication is aborted, else the tag prepares the response:

$A^1 = K \; xor \; c$;

A^2 = ID + (GI_i *xor* r *xor* c) mod (2^l -1);

B = H (ID ‖ (S *xor* GI) ‖ (r *xor* c)) where H is a hash function;

$B = B_L \| B_R$, where B_L is the left half and the B_R right half of the hash string;

The values A^1, A^2, B_R are sent to the reader;

The counter c is increased by 1 and if it exceeds 2^l -1 it is set back to the initial value of c;

The reader forwards the values A^1, A^2, B_R, and its generated r to the Back-end database (DB).

Upon receiving A^1, A^2 and B_R and r from the reader:

The DB computes c'= A^1 *xor* K and $ID_j'=A^2$-(GI_j *xor* r *xor* c') mod (2^l -1) using all group indices GI_j (where j = [1, n]);

The DB checks if one of computed ID'_j (where j = [1, n]) is matching to one of the stored IDs in the DB. If this process succeeds, the DB check if the GI_j used to compute ID'_j is equal to the group index GI_i that contains the matching ID'_j :

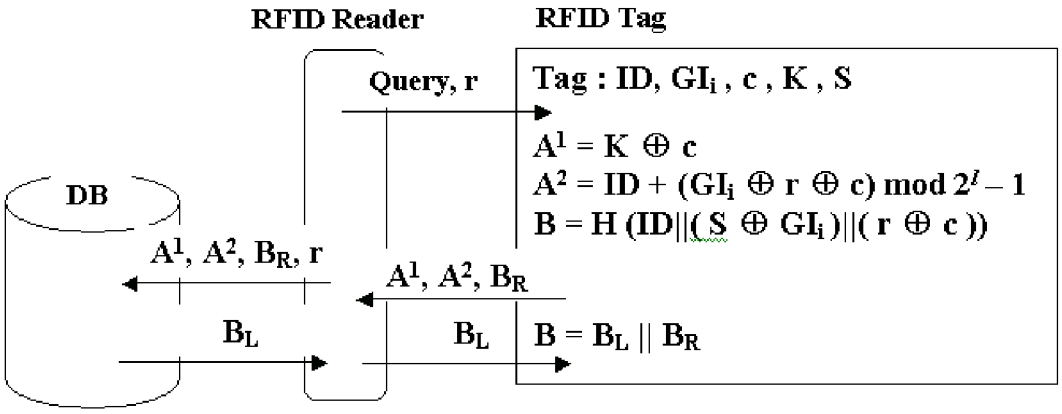

Fig.2. OHLCAP protocol

If this succeeds, the DB computes H (ID ‖ (S *xor* GI_i) ‖ (r *xor* c)) using c, r, GI_i, S and the matched ID. Otherwise, the DB halts this process.

Then, the DB authenticates the tag by checking if the right half of the computed value of the hash is equal to the received value B_R;

The DB sends B_L to the reader, where B_L is a left half of B, and the reader forwards B_L;

Finally, the tag authenticates the reader by checking if the received value B_L is equal to the left half of B calculated above.

3.2. Security analysis

This protocol uses simple operations, like XOR and addition, except for the undefined hash function that still must be selected. The protocol tries to hide the tags id by using a random value selected by the reader and the tags internal counter.

Unfortunately, the knowledge that c is increased after each started request gives enough information to figure out the values K, GI and ID.

As the adversary can supply its own r for the calculation of A^2 we have to create such an r so that (GI *xor* c *xor* r) = 0, in this case A^2 = ID. Let us forget about c for a moment, as GI never changes. Therefore, we could find out its value by testing changing each bit from 1 to 0 of r separately. Further we watching if the value of A^2 is becomes less or more, if the sum of ID + (GI *xor* r) becomes bigger as we set the single bit to 0, it means that the corresponding bit of GI has the value 1, because 1 *xor* 0 = 1 thus the value of the sum becomes bigger.

Now we still have the counter that changes some bits of (GI *xor* c *xor* r), but here comes the value of A^1 handy. As c is a counter, it will change last n consecutive bits depending on its current value. By watching A^1 we can determine the changes of the counters bit-string, because K never changes. If the 4 last bits flip at once we can know for sure that the first *l*-3 bits of c will not change for the next 7 reading attempts and we know how the last 3 will change. Therefore, with this knowledge we can forge our r for 7 readings to cancel out the changes of c and continue testing the ID bit by bit. In the best-case scenario, we need only *l*+1steps to find out the tags ID, in the worst case just a few more tries.

By this, we have compromised the tags anonymity and we gain the ability to track it.

However, this still cannot to authenticate to the database, because we do not have the access to the value of S that is used in the hash value sent for authentication. Here we could use a *man-in-the-middle attack* – when receiving a readers query with the random value r, we forward the query to a legitimate tag and get its calculated A^1, A^2 and B_R, and send them back to the reader as own authentication information.

4 Authentication with AES

4.1. Description of the protocol

Aiger and Feldhofer [9] proposed a protocol that uses symmetric encryption algorithm - AES. A symmetric algorithm has been selected, because these do not need as much costly arithmetic operations than asymmetric algorithms, which is very important, as the abilities of RFID tags are limited.

The protocol itself operates as follows:
First, the reader starts the communication with the tag sending authentication request, addressed with the ID of the tag. It also contains a nonce R_R generated by the reader;

The tag asks for the authentication of the reader by sending it a challenge, random value R_T;

The reader creates a challenge for the tags authentication – a random value R_R, appends it to the tags challenge RT, and then encrypts the concatenated values using the key. $E_K(R_T \| R_R)$, where E_K is the AES encryption with the key K;

The tag decrypts the message and if the value of the decrypted R_T is the same as the sent one then the reader has been successfully authenticated;

Now the tag takes the second part of the received message (R_R), appends the ID to it, encrypts with the common key $E_K(R_R \| ID)$ and sends the message to the reader;

The reader decrypts the message, if the first part is R_R then the tag has successfully authenticated itself and the reader now knows the ID of the tag.

As the execution of the AES encryption on the tag needs a time, the authors propose to run the protocol in an interleaved manner, to speed up the reading of multiple tags.

The communication is then divided in two phases – the authentication request (AR) and response request (RR). In the first phase, the reader authenticates to the tag and sends it a challenge, and then the communication is suspended so that the tag has time to compute the response and the reader can start an AR with another tag. Later on, the reader sends the RR to the tag and collects the response to the challenge.

4.2. Security of AES

The disadvantage of this protocol is that one secret key is shared between the database and all tags. If one tag is compromised, then the security of all tags becomes endangered. As the AES algorithm [10] is said to be secure it is not likely that an attacker might figure out the key by reading the challenges and responses.

Unfortunately, the attacks still could be done using side-channel [11, 6]. Once the key is found the attacker obtains the ability to track, clone and alter all tags that have this key set. It would be more secure if each tag would receive a different key, but in this case the reader would need to know which key belongs to which tag, so the tags would need to be uniquely identifiable.

This might be done as follows, where each tag becomes an alias that is changed after each valid authentication of the reader. The database holds a key and the alias (A) for each tag:

The reader starts a request;

The tag sends the random R_T and A to the reader;

The reader looks up the key associated with A and creates a challenge with a new alias A' for the tag $E_K^A(R_T \| R_R \| A')$;

If the decrypted R_T is the same as the sent one, then the tag sets its alias to be A' and sends $E_K^{A'}(R_R \| ID)$.

For the next request, the tag will answer with the new alias, which has not been sent in plain to the tag, so that simple eavesdropping will not enable the adversary track the tag further.

In case the authentication of the tag fails, the reader might not be able to determine if the tag has or has not updated its alias. For this the database should keep the last valid known, with which the tag was successfully authenticated and the current alias that has been sent to the tag. To make it more difficult for the adversary also a new key might be sent together with the A'.

This way the information on the tags would be better protected, as the key for each tag separately would have to be cracked. An adversary would only be able to track the tag between two legitimate

reads, except for the case that he is able to identify the tag by other means than eavesdropping and associate it again with the new alias. The negative aspect of this method is that the system should be able to perform automated legitimate reads of the tag from time to time, to avoid more advanced tracking attacks.

The interleaved version of the protocol would speed up the reading of multiple tags, but this might give an adversary the necessary space for a man-in-the-middle attack.

If the adversary would be able to send a response request right after the tag has finished calculating the response to the challenge and the before the legitimate reader sends his RR, then the adversary might be able to impersonate the legitimate tag. This could be avoided if the tag would only respond to the RR after a fixed period. In such a case, the adversary would not have the ability to get the response before the legitimate reader.

5 Conclusion

Two lightweight encryption protocols have been analyzed. OHLCAP [12] has shown not to be well suited to protect the privacy and uses more or less expensive methods to authenticate the tag testing multiple keys for decryption. The method is resources consuming and would not be convenient for low-cost RFID tags.

Nevertheless, AES based protocol shows much better security performance, because one key is used for all tags. The method improves the security of authentication, but looses some abilities to resist advanced tracking attacks. The proposed protocol is better suited for cases where attacks on the data integrity are more feasible than attacks on privacy.

References:

[1] B. Glover and H. Bhatt, *RFID Essentials*, O'Reilly Media Inc., ISBN 0-596-00944-5, 2006.

[2] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd Edition, Wiley, ISBN 978-0470844021, 2003.

[3] F. Thornthona and C. Lathem, *RFID Security*, Syngress, ISBN 978-1597490474, 2006.

[4] S. Garfinkel and B. Rosenberg, RFID Applications, Security, and Privacy, Addison-Wesley Professional, ISBN 0321290968, July 6, 2005, http://www.rfidbuzz.com/news/2005/book_review_rfid_applications_security_and_privacy.html, 15.09.2007.

[5] M.Burmester et al., *RFID Security: Attacks, Countermeasures and Challenges*, Computer Science Department, Florida State University, 2007, http://www.cs.fsu.edu/~burmeste/133.pdf, 11/09/2007.

[6] S.A.Weis, *Security and Privacy in Radio-Frequency Identification Devices*, MIT, Department of Electrical Engineering and Computer Science, 2003, http://groups.csail.mit.edu/cis/theses/weis-masters.pdf, 11/09/2007.

[7] R. Chandramouli et. al., Security Standards for the RFID Market, in *Journal IEEE SECURITY & PRIVACY*, 2005, http://csrc.nist.gov/staff/kuhn/phillips-karygiannis-kuhn05.pdf, 15.09.2007.

[8] Eun Young Choi et al., *Efficient RFID Authentication protocol for Ubiquitous Computing Environment*, Center for Information Security Technologies (CIST), Korea University, 2005, http://protocol.korea.ac.kr/~dhlee/papers/Efficient%20RFID%20Authentication%20protocol.pdf, 11/09/2007.

[9] M.Aigner and M.Feldhofer, *Secure Symmetric Authentication for RFID Tags*, Graz University of Technology, 2005, http://tcmc.tugraz.at/tcmc2005/PDF/20050228-IAIK-SecureAuthentication.pdf, 11/09/2007.

[10] B. Toiruul and KyungOh Lee, An Advanced Mutual-Authentication Algorithm Using AES for RFID Systems, *IJCSNS International Journal of Computer Science and Network Security*, VOL.6 No.9B, September 2006, http://paper.ijcsns.org/07_book/200609/200609C02.pdf, 15.09.2007.

[11] Yossi Oren et al., *Power Analysis of RFID Tags*, 2006, http://web.archive.org/web/20070907101300/http://www.wisdom.weizmann.ac.il/~yossio/rfid, 07/09/2007.

[12] T. Enokido, Lu Yan, Bin Xiao et. al., *Embedded and Ubiquitous Computing - EUC 2005 Workshops*, EUC 2005 Workshops: UISW, NCUS, SecUbiq, USN, and TAUES, Nagasaki, Japan, December 8-9, 2005, Springer, ISBN 978-3540308034, 2006.