# Hospital Automation using RFID Technology

Gustavo Florentino
Department of Electrical & Computer Engineering
University of Colorado at Colorado Springs
gflorent@uccs.edu

**Abstract**

This work presents a system developed for hospital analysis laboratories. This system employs smart cards to allow patients and doctors to authenticate themselves on the system without typing any user/password. Also the card stores relevant patient's information that can be displayed to doctors. This work focus on the security side of this application. One of the greatest challenges that this work tries to solve is providing security to smart cards that do not have processing capabilities.

## I. Introduction

On this work, it is proposed an information system which makes use of smart cards to speed up the procedures on a hospital clinical analysis laboratory. This work employs smart cards as a authentication device for hospital employees (physicians, biochemists, receptionists, and so on), and patients as well. Using a smart card, for instance, the physician can just approximate the card to an appropriate card reader and automatically the physician's modulo screen is opened. Thus, physicians and other employees can login on the system without spending time on a user/password authentication procedure. Furthermore, important patient's information is stored on the smart card, so the physician is able to rapidly access information like blood type and allergies.

Since every patient has one smart card, then the smart cards have a huge impact on the final cost of the system. Therefore it is employed on this application smart cards that do not have processors. Those smart cards are also known as memory cards. Again, due to cost issues, the smart cards used on this project are highly memory constrained. So this system faces a great challenge about providing security under such constraints. Foremost, most security schemes available on the literature for smart cards are based on challenge/response protocols. Furthermore, the high memory constraints deny the reservation of at least a reasonable amount of bytes for security purposes.

The smart card used on this project has many similarities with RFID tags, and those smart cards can be seen as RFID tags with card shape casing.

## II. Introduction on Smarts Cards

Smart card is a portable and resistant computer with programmable data storage capabilities. Smart cards have the exact dimensions of a credit card and they may be able to have data processing capabilities [1].

A smart card can be only a storage device or it can have a processor to execute some operations. Some smart cards allows the embed user's programs. According to Scheuermann [2], smart cards can be classified into memory cards and processing cards.

The section II-A introduces some basic concepts of memory cards. And the section II-B refers to contactless smart cards.

### *A. Memory Cards*

Memory cards are the simplest kind of smart cards. They do not have processors to execute programs, they have only a memory to store user data. The memory is accessed through a sequential logic [3].

Memory cards are smaller and easier to handle than paper documents and their data can be electronic read by a computer [2]. Another advantage is that memory cards are cheaper than cards with processor. For that reason, they are prevailing employed on large scale applications where the unit cost of the card have a huge impact on the final product cost.

### *B. Contactless Cards*

The contactless smart cards are a kind of smart card that can be read or written within some specific distance from the reader. So the contactless smart cards waives the need of placing the card in an determined position on the reader. The procedure of placing the smart card in a specific position on the reader (as required by contact readers) can consume a considerable amount of time in some applications. In some applications, the use of the cards that work on the proximity from the reader considerably enhances the operation efficiency [1].

The ISO 7810 specifies the format for all smart cards. The dimensions of the smart cards are defined as 85.46 mm x 53.92 mm x 0.76 mm. The 0.76 mm thickness is a big challenge for the smart card manufactures since this thickness constrains the size of the antenna and the chip.

The contactless cards can be manufactured using four PVC foils with 0.2 mm thickness each one. The figure 1 shows the physical structure of a contactless smart card.

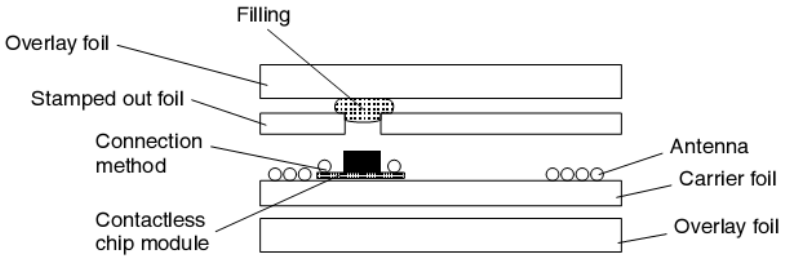


Fig. 1. Contactless smart card foil structure

| Block | Size : 8 bytes |
|---|---|
| 0 | Serial Number (64 bits) |
| 1 | Configuration block |
| 2 | e-purse |
| 3 | Debit Key |
| 4 | Credit Key |
| 5 | Application Issuer Area |
| 6 | Application Area 1 : protected by kd |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | ---------- |
| 12 | ***Modifiable limit*** |
| 13 | |
| 14 | Application Area 2 : protected by kc |
| - | |
| 31 (2K) | |

Block 6 to 12 are write lockable

Fig. 2. Memory structure of the smart card used on this project

Contactless smart cards have a less complex mechanics and a cheaper maintenance cost due to wear, corrosion and dirty [4] [3].

## III. Memory structure of the smart card

The application of contactless memory smart cards is justified by two factors. One regards to the low cost of a memory card when compared to smart cards with processor. And the other factor is the low occurrence of maintenance problems.

The smart card used on this project has 2 Kbits EEPROM memory, arranged as shown on figure 2. There are smart cards with memory up to megabytes, but as it will be presented on this report, 2 Kbits are enough to store the most important information about the patient.

The memory is organized in 32 blocks of 8 bytes each. So, 256 bytes of space. The first six blocks are configuration blocks and therefore are not used to store used data. Thus we have only 26 blocks left, or 208 bytes, to work with. Among the configuration blocks, the zero block is of special interest because

| Block | 8 bytes |
|---|---|
| 6 | Version id |
| 7 | User id |
| 8 | User data |
| 9 | User data |
| 10 | User data |
| 11 | User data |
| ... | User data |
| 31 | User data |

Fig. 3. Memory mapping of the smart card used on this project

there is the position where the card serial number is stored. It is assured by the manufacture that every card has an unique serial number.

## IV. Memory Management

As seen on section III, there is a huge restriction on memory space to store user data in the smart card. On the other hand, as affirmed by Anderson [5], the health institutions work with a high volume of patient's information. So there is a clear conflict between storage space on the smart cards and the amount of data about the patients. Another factor that influences on the on the storage space is the security and data integrity model employed, what can result in a larger required space to store keys and verification codes. In general, more security schemes require larger codes and keys.

To overcome the space restrictions, it is necessary to store in the card only the most relevant patient's information. Besides that, the data formating must be the most possible concise.

## V. Memory Mapping

In order the application to be able to identify the meaning of the data present on each block, it is necessary to have a previous mapping of the smart card memory. The figure 3 shows the memory mapping proposed on this laboratory system.

The block 6 is reserved to store the memory manager version. This memory version identification offers the flexibility to change the data format without creating incompatibility with the other obsolete cards. This field is the first one read by the system when a new card is found. With this version identification, the system can load the correct memory manager. This way the data can be read or written on the correct format in the card.

The block 7 corresponds to the user id on the database. However just an integer number stored on the card would not provide any security. A simple malicious attack could be based on putting an integer number on the block 7 and so that would yield an improper authentication on the system. To avoid this flaw, the authentication process goes through a second verification making use of cryptography. The section VII deal with the security measures applied on the system.

The shadowed are on the figure 3 means that the blocks 8 through 31 are reserved for user data storage. The data stored on this area are secured in terms of integrity and confidentiality by means of cryptography techniques described on section VII. The data stored on this area follow an specific formating defined on section VI.

## VI. Data formating

At first, it was envisioned the employment of XML markup language to format the data stored on the card. However, essentially the XML language outputs documents quite larger than the ones produced by other specifications, carrying out a high cost in terms of storage [6]. Therefore XML became an impractical solution under the high limited memory constraints imposed by the smart cards.

In order to perform the data formating, it was created a simple markup language that requires few memory space. By convention on this language, each marking element is started by a backslash followed by 3 ASCII characters. The following listing shows the marking elements created for this language.

- **beg** - Beginning of the user data space
- **rnd** - Random number generated every read or write on the card. This value is used in the security part e its use is discussed on the section VII.
- **crc** - Cyclical redundancy code of 16 bits applied to the text between \beg and \en (including those marking elements).
- **usr** - Card's user name.
- **bty** - Blood type
- **dia** - Diabetic patient. The following value must be "T" for true or "F" for false.
- **hpt** - Hypertension carrying patient.
- **ale** - Listing of the drugs or substances in which the patient is allergic.
- **end** - End of the user data area.

The text below exemplifies the data on the card belonging to the patient Joseph Jr who has blood type

AB+, has diabetes, who has not hypertension and is allergic to Acetylsalicylic acid.

```
\beg\rnd37026\crcb80e\usrJoseph Jr\btyAB+\diaT\hptF\aleAcetylsalicylic
acid\end
```

The formated data of this previous example occupy 79 bytes on the smart card memory, so begin within the limit of 208 bytes. The useful data corresponds to 57% of the total space against 43% of the marking elements and security data. It can be verified that the security fields, \rnd and \crc, correspond to approximately 18% of the formated data.

It is worth-noting the the elements set can suffer changes as long as a new identification version is created for the memory manager. Besides that, the order of appearance of the fields can be disposed randomly.

## VII. Security

The smart cards used on this project can perform to functions: authenticate users and store user's informatation.

The user authentication in a relatively secure way using smart cards is a great challenge in this work. Since the smart cards used do not have processor to execute user programs, so many security schemes proposed on the literature could not be applied in this work. Most part of the solution are based on the send of challenges and replies between the system and the card [7], [8], [9], [10].

Concepts of digital signature were employed to assure that only the cards written by the system are considered authentic. The authentication protocol is described as follows.

Upon the detection of a new card, the a factory class verifies the block 6 on the card (the memory manager version to be used) and loads an appropriate class to deal with the data on the card. Then the system reads the user id which is on the block 7, according to figure 2. The cryptography key sk associated with this user is retrieved from the database and the data contained on the blocks 8 through 31 are read. So the key sk is used to decipher the data read and obtain the original text. This text has a formated as described on section VI. So the field \rnd is read and compared to the random value associated with the user stored on the database. If the values match, then the user is authenticated on the system. Every read or write on the card, a new random number is generated and store both on the card and on the database. The cryptography key can also be changes on every read or write of the card.

The protocol previously presented works in a user/password fashion scheme. But the password is enciphered among the data user data stored on the card. So it becomes nearly impossible to find out the random number \rnd without the knowledge of the secret key sk. One question that may be raised is whether the formating markup tags are really necessary since there is a great concern about memory space. A solution that can be proposed is pre-allocate the position of each information and so we would not need to place tags to identify the meaning of the data. But the solution proposed, using tags, has the versatility of placing the tags on random order. So, the tags can be placed in an random order, hardening even more the identification of the random value among the formated user data.

Another question relevant to this work is the verification of the data integrity. It is fundamental to verify the consistency of the data contained in the card and that malicious attacks cannot change the data on the card such that they seem to be authentic. To reach such goal, a cyclic redundancy code is applied to the formated user data. Upon the data reading, it is performed the CRC checking to verify the data integrity. If the calculated CRC does not match the one stored on the card, then an error is raised and the user has the option to overwrite the data on the card with the data stored on the database. In a similar way that the \rnd field cannot be changed without prior knowledge of the key sk, the \crc code cannot be changed as well. If a malicious user tries to change the data on the card, he will end up changing the CRC and the data stored on the card will not be recognized as valid.

During this work, the MD5 hashing algorithm was considered to be employed instead of the CRC algorithm. But the MD5 algorithm generates 16 bytes hashes, a quite large length for this project. Therefore, the choice was made in favor of the CRC 16 bits algorithm, which occupies only 2 bytes on the smart card. Even having a small CRC length, the probability of collision occurrence is low since the smart card has just about one hundred bytes stored on it.

The cipher algorithm chosen for the system is the Advanced Encryption Standard, basically thanks to two reasons. The first is that the AES is a symmetric-key algorithm. And symmetric-key algorithms in general are hundreds to thousands times faster than asymmetric algorithms [11]. And the second reason is that the system does not need to change key with other parties. Only the designed hospital system is supposed to read or write on the card, therefore the employment of asymmetric key algorithms is not justifiable.

On this system, it is used one secret key for each card. This scheme was adopted instead of using one

shared key for all cards because if one card key gets compromised, only one card will be affected instead of the whole system [12].

## VIII. Conclusion

The application of smart cards indeed speed up the process on clinical analysis laboratories and avoid many errors from happening. We verify that a fast user authentication perfectly adapts to the dynamism of hospital activities.

Although the high constraints imposed by the memory and processing capabilities limitations present on the smart cards, this work showed that it is possible to implement a secure system even under such constraints. The digital signature approach was demonstrated to be suitable to solve this problem and not compromising too many bits for security issues.

The AES algorithm was demonstrated to be also suitable as solution since we need fast reads and writes on the card.

## References


[1] T. M. Jurgensen and S. B. Guthery, *Smart Cards: The developer's toolkit*. Prentice Hall, 2002.

[2] D. Scheuermann, "The smartcard as a mobile security device," *Electronics & Communication Engineering Journal*, Outubro 2002.

[3] K. Finkenzeller, *RFID Handbook: Fundamentals and Apllications in Contactless Smart Cards and Identification*, 2nd ed. Munich: John Wiley & Sons, 2003.

[4] M. David and K. Sakurai, "Enhancing airport control security," *Australian Information Warfare Conference*, Novembro 2002.

[5] J. G. Anderson, "Clearing the way for physicians' use of clinical information systems," *Communications of ACM*, Agosto 1997.

[6] W. Ng, W.-Y. Lam, and J. Cheng, "Comparative analysis of xml compression technologies," *World Wide Web*, vol. 9, no. 1, pp. 5–33, 2006.

[7] Z. Kfir and A. Wool, "Picking virtual pockets using relay attacks on contactless smartcard systems," *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005.

[8] A. Aziz and W. Diffie, "Privacy and authentication for wireless local area networks," *IEEE Personal Communications*, 1994.

[9] T. Phillips, T. Karygiannis, and R. Kuhn, "Security standards for the rfid market," *IEEE Security & Privacy*, Novembro/Dezembro 2005.

[10] G. Roussos, "Enabling rfid in retail," Maro 2006.

[11] J. Talbot and D. Welsh, *Complexity and Cryptography - An Introduction*. Cambridge University Press, 2006.

[12] D. Molnar and D. Wagner, "Privacy and security in library rfid: Issues, practices, and architectures," *Communications of the ACM*, Outubro 2004.