# NFAuthenticator: the key is your phone

Daniele Ronzani
Department of Mathematics
University of Padua
Padua, Italy
daniele.ronzani@gmail.com

Michele Massaro
Department of Mathematics
University of Padua
Padua, Italy
michele.massaro@me.com

***Abstract*—NFC is a technology that allows communication over short distance. Nowadays it is used to exchange data between smartphone or for authentication system, for example many buildings use tags instead of keys to open doors. Actually, most of the current solutions are based on the identification of the user using the information contained inside the NFC tag. This approach implies some problems involved by the use of a passive key, such as the inability to lock the key to prevent unauthorized use. The purpose of this work is to create an authentication system based on a client-server architecture that allows to access, through a login, to some areas inside a building identified by NFC tags and read by an Android phone. Moreover, those tags can be used to uniquely identify the positions of the users during the authentication process, allowing the administrator to track the users activity while inside the building.**

***Keyword- NFC, TLS, Security, Android***

## I. Introduction

In recent years the NFC technology is becoming widely used, a protocol that provides a bidirectional short range wireless connectivity for information exchange. The NFC had a strong impact on many practical use, and that caused a big interest from many big companies (Nokia, Sony, Philips, Samsung, Motorola). Thanks to the ease of use and the lack of configuration of the protocol, it became used on a lot of fields:

- contactless payment;
- health care;
- smart touch;
- authentication.

One field where it became very popular is the authentication one: nowadays it is easy to find NFC tags used like electronic keys in hotels or big buildings. For example a lot of hotels replaced their keys with NFC/RFID cards. Anyway, the simple replacement maintains some negative feature of the old system, such as:

- cards/tags are passive like keys, so there is no way to protect them with password in case of loss;
- the user must carry another object with himself;
- the inability to automate the key distribution process (for example in the hotel scenario the user must ask for the card to the reception).

This work consists of the implementation of an authentication system that could be applied on buildings where different users can access only certain parts of it (for example in a hotel every user can access only his room). The system has to maintain the ease of use of tags, provide a better security and must solve all the problem explained before; to allow the user to not bring additional objects, we chose to use for the authentication a device that most people already own: the smartphone. The system follows three objective:

- provide an authentication system that allows only to authorized users to open doors;
- keep the user-friendliness of the classic NFC/RFID system;
- provide a better security than NFC tags.

For the first step it was required to implement a server with a database to store users accounts and resources (doors informations). In this way, a registered user can be authenticated through a login and can get (or not) the permission to open a certain door. The second step was reached making the system able to detect what resource is required by a user. Even if the position of every resource is known, the A-GPS cannot be used, because of the low granularity provided for indoor localization. Instead we used one NFC tag (with a 4-5 cm range) for every resource, that contain an ID that allows the system to understand where the user is, and what resource is required. The third step is partially reached by design, because any Android smartphone can be protected with a password lock screen. The wireless communication between the Android phone and the authentication server implies possible security issues, so has been used the TLS encryption to ensure a secure communication.

*Related Work:* A similar solution already exists [6], and it uses an innovative access control system. Through a NFC P2P connection, the smartphone communicates with an NFC reader inside an electronic door lock. In this case the key can be locally stored (inside the phone) or stored in the cloud. Even if this system is really advanced, every lock contains a lot of features, as the NFC communicator, that implies a substantial cost for each door, unlike our work.

## II. NFAuthenticator System

The system introduced here is proposed as a substitute for the current authentication system though NFC tags. The basic idea, which differs from the classical solutions, is the reverse approach of how the user is authenticated: while, in the

classical way, the tag represents the access key and the reader is the "lock", in our approach the NFC reader of smartphone become the key, and the tag represents the door. In this way, the user does not bring with him the tag, but the reader, that can identify the resource approached and request the access to a remote server.

In addition, the relationship between tag and its resource is inherently a system of localization, because the reading implies physical proximity between the smartphone and the place where the door resides, allowing the system administrator to have an almost real-time view of accesses (and relative positions) of the users during the authentication process.

### A. Architecture

The system is composed by three logical part:

- **Client** (NFC Locator)
- **Server** (NFC Authenticator)
- **Resource** (doors lock)

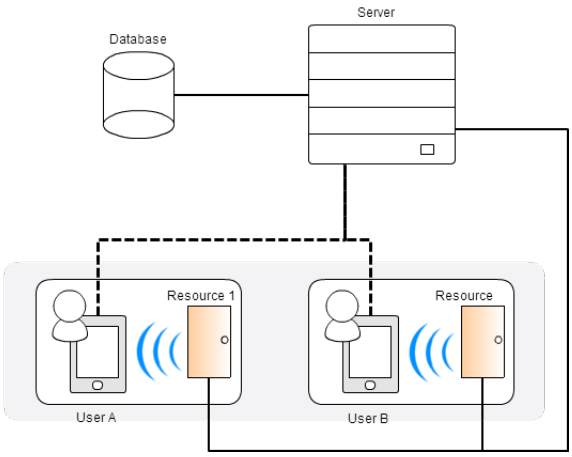


Fig. 1. System Architecture

The client consist of an Android application (NFC Locator) that allows the user to detect the resource. After the reading, the app will send the request to access the resource to the remote server. The server (NFC Authenticator) is an application that receive requests and contains a database that store usernames, passwords, and permission of everyone. There is also a graphical interface that allows the system administrator to manage the accounts. The resource consist of a component that can be remotely controlled by the server, in our case a prototype of a door lock. Through an ethernet connection it can communicate on the network with the server and receive the open command.

### B. Networks

Three kinds of connections are involved: wired network (between server and resource), wi-fi network (between server and client) and NFC (between client and resource).

The connection between the server and client can happen in two ways that differ regard to safety. The first is the connection via local wi-fi network, in which the smartphone is forced to connect to the local wireless network of the building to gain access to a gate, in fact only in this case can communicate with the control server. The second is the connection to the server using the UMTS mobile telephone technology: in this case the access request travels through the global network, and then get to the server. This mode can only be present if the server has a static external IP address, so that it can be accessed remotely. This second way is much less safer than the first one. For this reason, the connection between client and server is protected with a cryptographic protocol , the Transport Layer Security (TLS ), which allows secure communication in the transport layer , preventing eavesdropping or man in the middle attacks.

NFC communication allows data transfer between the client and the resource, in particular a unique identifier representing the door, which will be sent to the server during the authentication process.

### C. Client Side

As said before, the client consist of an Android application that allows to read NFC tags and send information to a remote server. The application is composed by two parts, the configuration and the reading.

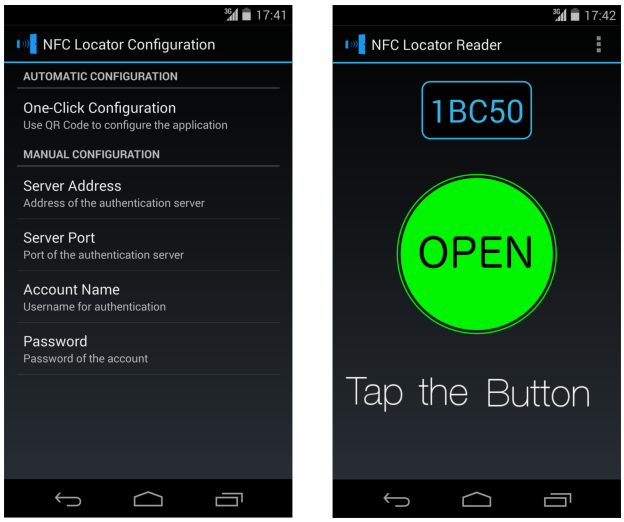


Fig. 2. **NFC Locator:** Left: Configuration window; Right: Reader window

*1) Configuration:* the configuration window allows the user to set all the network parameter and login information, like username and password. Even if the configuration must be done once, it seemed too difficult to let configure server address and port to an average user, and because the application was made to be user friendly and usable by most people have been added the One-Click Configuration option: after an user has been added into the server, can be automatically generated a QR code; the user just have to point the phone to the QR code to setup all network parameter. The protocol used by the QR reader needs a string designed with the following pattern:

*"address:<address> port:<port> user:<username>"*

The only information that have to be inserted manually is the password, for security reason.

*2) Reading:* to maintain the application easy to use, the user dont have to launch any application to detect a resource. It is enough to approach the phone to the resource to make appear the window on the image 4. The only part that can be touched in the interface is the big green button in the middle, that given its large dimension is easily usable even for visually impaired people. When the button is pressed, a request containing the login and the resource ID is sent to the server, while the app shows a progress indicator. The message sent must contains a string in the following pattern:

"<*username*>:<*password*>:<*resource ID*>"

After a moment the positive or negative response appears. At the end of the operation the window will automatically close, preventing the reiteration of the process.

### D. Server Side

The server program is an application that consists of two concurrent entities: one that handles the actual server and one that manages the GUI.

*1) Listener:* the server has a listener that waits for messages from clients requesting access to a resource. For this function there is a thread that listens on a port and waits for a TCP/IP communication from the client. The full flow can be seen in Figure 3. Once the channel is open, the server waits to receive a message from the client containing the user's account information, and the data of tag detected. First, a check is performed on login data received, and only if these are correct is checked that the user has the necessary permission to use the resource. If both controls have a positive outcome, the server open a connection to the requested resource, sending the command to open and possible additional data, such as the username of the requesting user. Only at the reception of the ack by the resource, the server considers the operation completed and sends the positive response to the user.

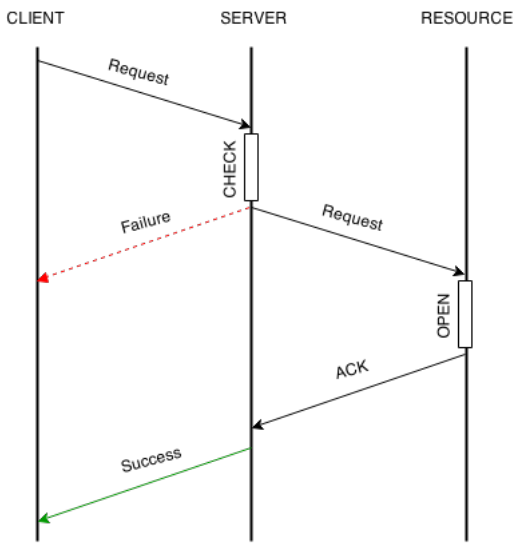


Fig. 3. Communication flow

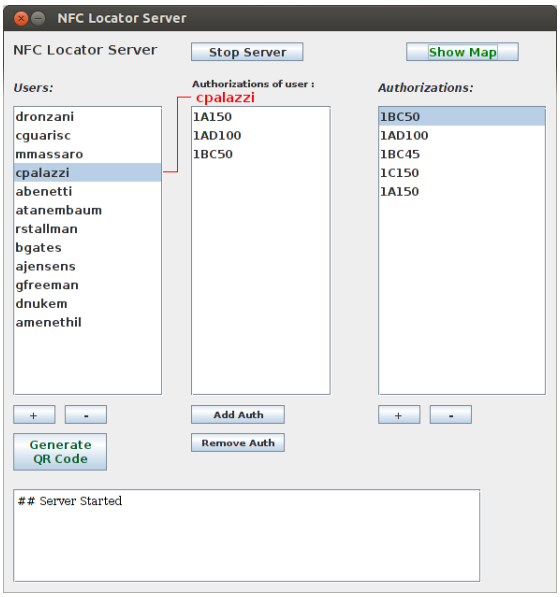


Fig. 4. Server GUI

*2) Graphics:* the graphics part is characterized by a window with three main areas:

- members list
- authorizations list
- list permissions owned by the user

An element can be added or removed to the user list through the buttons at the bottom. The same applies to the other two lists. Through the "Generate QR Code" button can be created a QR code to automatically configure the client of the selected user.

The "Start Server" button starts the listener server to accept requests. From that moment on, every client can connect to the server.

*3) Map Area:* Another feature is the presence of a window that alerts the administrator of authorized or unauthorized access in a specific location. This window is automatically shown, and display a map of the area (Figure 5a), highlighting the zone in which the user has accessed (Figure 5b), or which has not been authorized (Figure 5c).

### E. Database

Users and permissions are stored in a relational database implemented with Sqlite. There are three tables. A table contains the information of users: username and password. Another table contains data regarding the resource: name of door and IP:PORT address. Finally, a third table identifies each pair user-resource to store authorized accesses of each user. Database has been designed with ER diagram shown in Figure 6.

### F. Resource

The resource was built to be much similar as possible to a lock. To realize it was used an Arduino Uno r3, programmed to

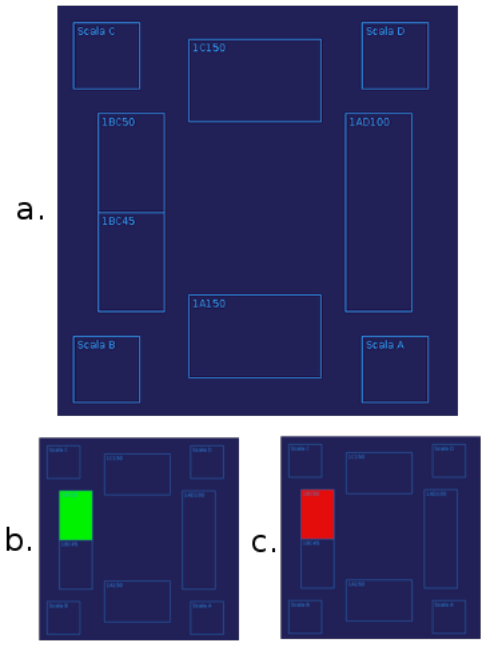


Fig. 5. Server map window

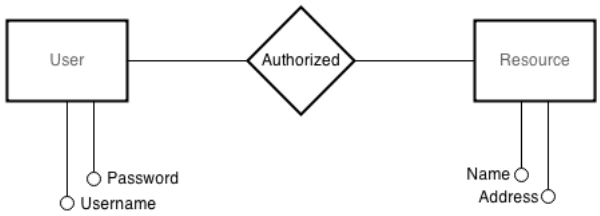


Fig. 6. Database ER diagram

reflect the behaviour of a remotely controlled lock, so there is a small motor that move the closing cylinder and a display that show the state when necessary. The connection to the server is provided by an ethernet interface, that allow the exchange of packets between them. Inside the lock there is an NFC tag, that contains the identification of the resource, needed to be recognised by the server. The tag used is a "NTAG203", that contains up to 137 byte of payload. To preserve the compatibility with cards that are often used now, have been added an RFID reader, also usable in case of emergency or if there are network issues.

### *G. Security*

This system was proposed to replace NFC tags, so will be done a comparison between the security offered by both methods. Tags are passive objects, that can be used even by not authorized users in case of loss. The advantage on the use of a smartphone as user interface is given by the possibility to use a password to unlock it, so it add an obstacle to a possible thief. A second problem of the tags is that they can be read by every reader, even if the tag is not used. For that reason most tag contains encrypted information, or use more sophisticated ways to avoid man in the middle attack [1]. In the proposed system, the smartphone is the reader, and the only passive

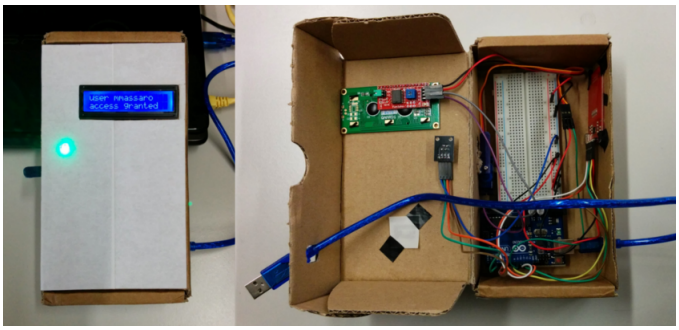

Fig. 7. Image of the prototype used in the demo

entity is the resource, that contains not sensible information (the ID is meaningless outside the system). Moreover, every Android phone turn off the NFC antenna when the screen is locked, preventing possible security issues. An essential point of the system is represented by the connection between the client and the server, where are exchanged sensible information. For this reason, the connection have been encrypted using the TLS protocol; the client contains a certificate that allow to authenticate the server identity, preventing "man in the middle" attacks. In addition, it is useful to emphasize that in the system there is no authentication data exchanged between the client and the resource. For that reason, even if the NFC communication is eavesdropped, no sensible data can be collected.

## III. Performance

The system is made to be used very often, so it does not just need to be easy to use, but it must be fast to not annoy users. For this reason have been done tests on the reactivity of the system, and the result are shown of the graph on the Figure 7.

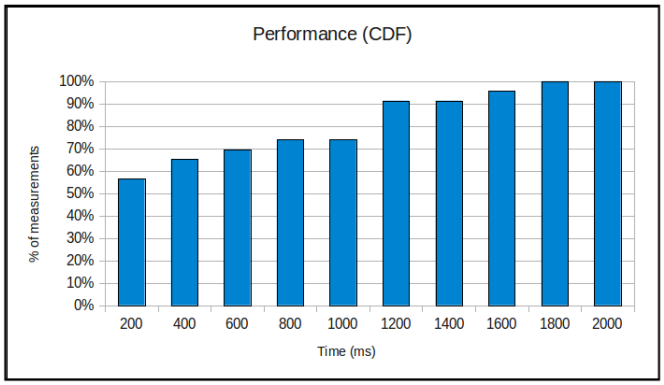


Fig. 8. Performance graph

In more than 60% of the measurements the client completed the communication process in less than 400 ms as can be seen in the cumulative graphic in Figure 8. Sometimes the time can be greater because of network congestions, but the procedure never spends more than a couple of seconds. Overall the mean of access time should be lower than the classical approach, because the user doesnt need to search the NFC card inside the wallet, but have to use his smartphone, that usually is in handy.

## IV. Conclusion

This work has been presented as an alternative to actual NFC authentication systems; have been proposed a new system, that allows to reach a higher security level than some classical RFID card, and that makes the key of a real door completely virtual. The virtualization does not just give us the chance to use and Android phone for the authentication, but could also be used to automate all those operation that are now manually handled, like the keys/cards distribution without requiring an higher operating cost. In fact the cost is lower than other solutions [6], because all the components can be retrieved for less than 30 euros. Moreover, all the elements can be easily assembled without the need of an industrial process, as showed in the demo.

## References


[1] Yun-Seok Lee, Eun Kim, and Min-Soo Jung, *A NFC based Authentication method for defense of the Man in the Middle Attack*, Bali, Indonesia: ICCSIT'2013.

[2] Pr Pascal Urien, *NFC Technologies for the Internet Of Things*, 2013.

[3] Ernst Haselsteiner and Klemens Breitfu, *Security in Near Field Communication (NFC), Strengths and Weaknesses* .

[4] Gauthier Van Damme and Karel Wouters, *Practical Experiences with NFC Security on mobile Phones*, 2009.

[5] W. Chen, G.P. Hancke and K.E. Mayes, Y. Lien, J-H. Chiu, *NFC Mobile Transactions and Authentication based on GSM Network*, 2010.

[6] Pascal Urien, *A Secure Cloud of Electronic Keys for NFC Locks Securely Controlled by NFC Smartphones*, Las Vegas, IEEE CCNC 2014.