Securing NFC application data with combination of CLOUD (as SaaS Service) and secure element

Sarabjeet Singh*

* Research and Development, Syscom Corporation Limited

Abstract- Near field Communication (NFC) technology provides many possibilities in consumer application domain other than just payment applications as it provides characteristics such as contactless communication and exchange of information between devices. But it has some limitations and restriction on data storage and security. To overcome these limitations a combination of CLOUD as SaaS services and secure element can be used. This paper describes how the combination of 'Software as a Service' feature of Cloud technology and Secure Elements (SIM cards) can be used to overcome these limitations.
This paper does not goes into implementation details on coding side but provides a basic flow of solution implementation.

Index Terms- NFC, Secure Element, Cloud, SPI, SaaS, ACE

I. INTRODUCTION

Near Field Communication technology is in the market since a while now. NFC application allows communication with many applications and devices therefore this application requires security measures to control access to the available data on the devices and also requires the security of data on the device itself. While not all NFC applications require security, but those who allows/supports financial transactions certainly do require some security measures. One of the security measures is use of "secure element". Secure element can exist within the phone itself or within a smart chip essentially a NFC chip. This secure element is used to securely store applications and/or credentials and provide for secure execution of applications. Here the secure element is a secure area of environment on which NFC application code and its related data is processed, stored and administered.

For NFC applications, data can also be stored on either a secure element or on the Mobile equipment NFC enables Card. Table1 lists some of the available NFC chips with their memory information. NFC chips can have different sizes of memory and similarly can have different memory configurations. This affects the amount of information that can be stored on certain chips but it also affects how the chip can be locked and other very important factors. The table contains two memory listings:

Memory Size: It is the total amount of memory within the chip. Some of this are one time programmable (OTP), some will be for locking features and so on. Most will be for user read/write functions.

User Memory: This is the memory that is important to a user because this is the available memory that a user can use to store the data.

Table 1 : NFC chips availed in Market

NFC Chip	Memory Size (bytes)	User Memory(bytes)
Ultralight	64	48
NTAG203	168	144
NTAG210	80	48
NTAG213	180	144
NTAG215	540	504
NTAG216	924	888
Ultralight C	192	148
Mifare 1k	1024	716
Desfire 4k	4k	4094
Topaz 512	512	454

There are three problems with these secure elements:

First with secure element used on the device itself has a risk of data loss with loss of device and issue of data compromise as some other application can access the data stored by the NFC application without the knowledge of the user.

Second problem is with the used NFC chips. Firstly, not all chips or smart cards are NFC enabled secondly as shown in Table 1; the user memory space is limited so it has limitation on how much data can be stored on the card.

Third problem is the access to the NFC enabled smart cards. As there are many methods and APIs such as OpenMobile API available that can be used in development of NFC applications, these applications can access Smart card and the applications on it easily. If not controlled, these applications can also store and access data that they should not store/access.

Research Elaboration for proposed solution- Basic thought behind this paper is to provide an approach and solution to secure the NFC application data and provide measure to access the data in a secure and controlled way.

For this purpose, a combination of CLOUD computing services and secure element can be used. Here we are taking secure element to be the smart card used for NFC transactions. Global platform has provided specification for secure elements access based upon which restrictions can be applied on android applications on how they should access secure elements. These restrictions can be used to store data processed by NFC applications so that they do not need to rely on any other security application of securing the data. For this we will use both secure element and Software as Service characteristics of Cloud Services.

II. Proposed Implementation

Following tasks are required to implement the proposed solution for securing the NFC application data:

1. Move the Mobile backend and storage of data to Cloud Storage instead of using local storage on device.
2. Implement a Security Middleware to control and authentication access to Cloud from the application
3. Access the smart card in controlled and secure manner to store sensitive information.

A. Moving Mobile backend and storage to Cloud

We all know that Android Applications are capable using cloud storage. Cloud provides three types of services, namely: Software as a Service, Platform as a Service and Infrastructure as a Service. These three services constitute the SPI model of cloud. For this solution, Software as a Service will be used. In this, access to cloud applications is provided over internet and services can be accessed by any device capable of connecting to internet and have a user interface. Following figure depicts an access model from android to Cloud as SaaS service:

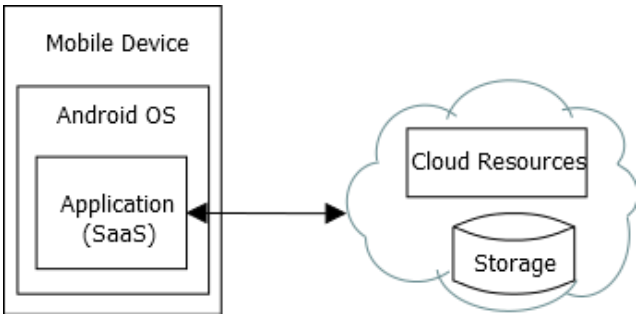

Figure 1: Cloud Access (SaaS)

Whenever an NFC application need to store data it should access cloud storage and store the data on it this proposal assumes cloud implementation on Google Cloud Platform. Google Cloud Platform provide Google App Engine on which mobile backend can be developed and custom code can be run and can be accessed by the client Applications running on mobile devices using Google Cloud Endpoints. Endpoints provide methods to use REST API from communication and consuming REST APIs from Android.

Figure 2: Google cloud access by Cloud Endpoints

It also provides OAuth2-based authentication, so that the mobile backend code can know the identity of the caller service.

B. Security Middleware

Now that most part of the application is on cloud, access to the cloud service needs to be secured from cloud end. For this a seed based key will be used for user authentication to cloud service. This key will act similar to as a password but will be generated at both the cloud end and at the client end. A Seed key of 1 KB will be provided to user. This will be stored on the SIM card or secure element itself. A permanent user ID will be provided to each user. With the combination of the Seed and ID a key will be generated that will be used for authenticating the session.

After generation of the secret key, it is required to secure the key also so that spoofing attacks can be avoided. For this 3DES encryption will be used to encrypt the key. This encrypted key will be used by the application along with other information for authentication. The transfer will occur in JSON format. The transfer mechanism can be written in any format for example as a REST service. Following will be the format of the JSON. This format is not mandatory and can be modified according to requirement. It is indication of what can be sent in string and how can be sent:

```
[
 {
    "userId"  : "userId"
     "appId"  : "Application ID trying to authenticate with Cloud"
     "service" : "Required Service "
     "_id": "XXXXXXXXXXXXXX",
     "_ids": "553c73fa5e3d8b91473529b2",

 }
]
```

Following figure shows key generation encryption and authentication process:

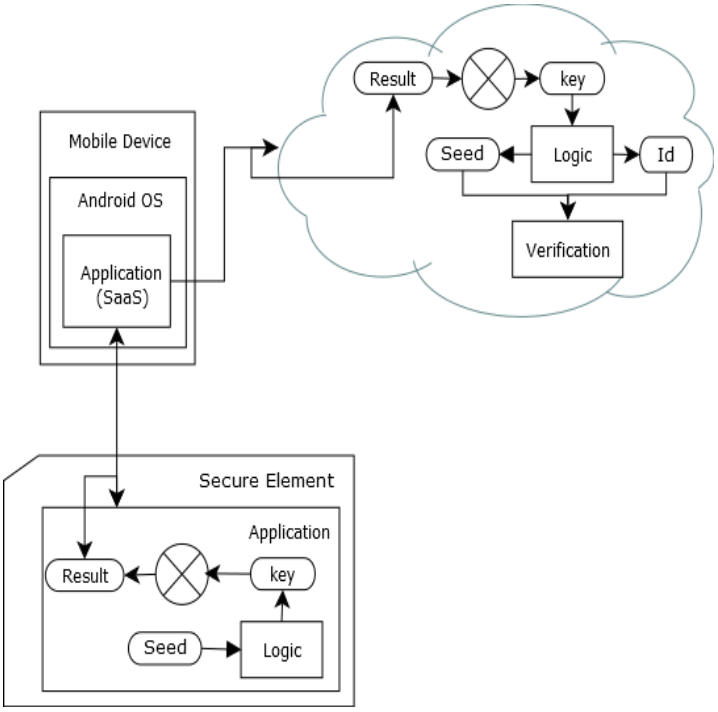

Figure 3: Authentication and Key Generation

C. Controlled Access to Secure Element by Android Application

Third part is to control access of the android application. Since new business opportunities allows the application developers to take advantage of android-secure element interactions. GlobalPlatform has defined a standard that enables several

application development companies to manage their stakes in a single Secure Element. APIs such as OpenMobile API has allowed the applications to exchange data with the applications running on secure element as in our case the key generation application runs on secure element. Now other than this applications, there are other applications that uses such APIs also and since do not provide an efficient mechanism to prevent unauthorized parties from abusing the API and potentially causing damage to the Secure Element itself, therefore we need to control this access. Global Platform "Secure Element Access Control" specification has provided generic mechanism for access control that is applicable for any kind of secure element: embedded SE, micro SD or NFC chip. There are two main parts of this secure access architecture: Access Rule Enforcer is the Device Itself and the Access Rule Application that should be used to access the application on the secure element. Simply stating following process happens in this access control:

1. Card issuer defines rules for controlling the access for secure element applications.
2. There is an access control master application which is supplied those rules.
3. When a device application has to access the SE application, the access control enforcer on the device should use the interface provided by the access control application to get the access rules from secure element.
4. Access is only permitted if the rules specify that access is possible to the application.

The access rule application is any SE application which can be selected by a GlobalPlatform-defined AID. GlobalPlatform Device Technology Secure Element Access Control (Version 0.10.0) provides complete description of how the access rules are to be implemented in the SE application and what is to be done with combination of the rules. Following depicts a particular access scenario:

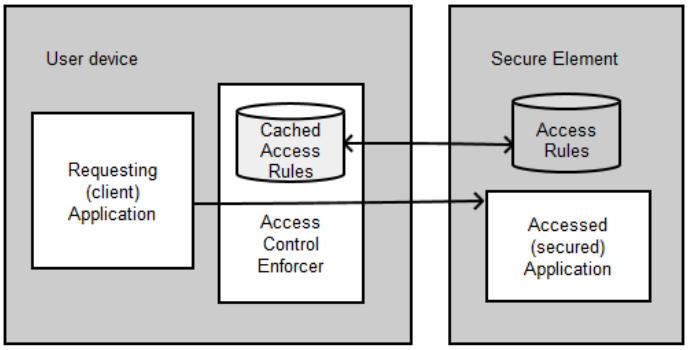

Figure 4: Enfrocing access control

1. A client application signed with a unique key tries to access a specific application through its AID on a Secure Element.
2. ACE reads the AR for the specific AID and the applications certificate hash.
3. Grant access to the client application according to the access rule or deny access if no rule is found.
4. Client application can communicate with the SE applet if the command APDUs match the filter list (if given) checked by the ACE.

After complete implementation following will be the assumed complete application schema:

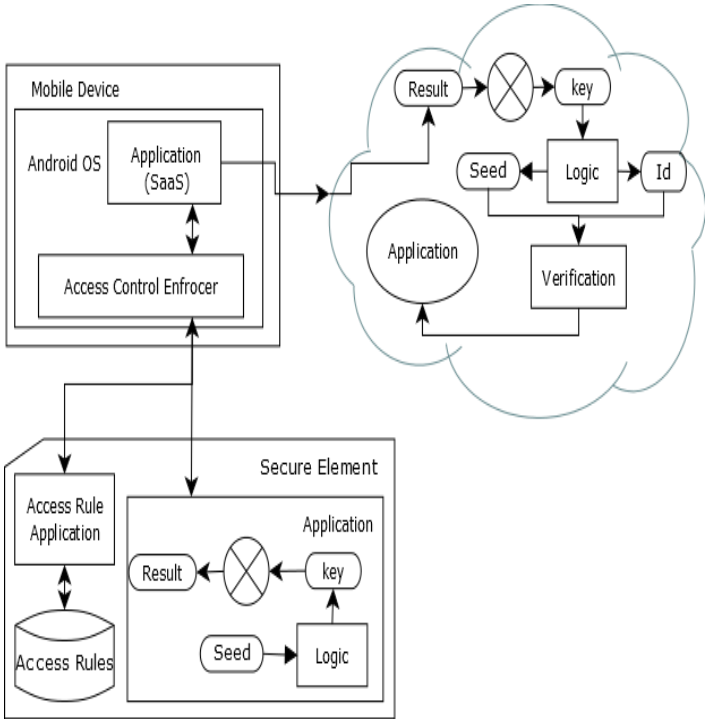

Figure 5: Complete Schema for implementation

III. Conclusion

NFC applications are full of potentials but constraints such as storage space and security issues are not allowing this technology to be used to its full extent to created enriched user experience applications. Cloud computing technology provides us many benefits such as centralized storage space, scalability and manageability, but most of all it provides device independence. With the help of cloud computing, device interface and back-end can be separated for added security. If Cloud is used with NFC, then potential of NFC application increases exponentially. Plus secure elements provide security at the client side for important information such as user credentials. Using Global Platform Access Control mechanism, secure access to NFC applications can be allowed to secure elements. Added benefit, the secure element in such implementation need not to be an NFC chip for such operations.

References

[1] GlobalPlatform System Messaging Specification for Management of Mobile-NFC Services Version 1.1.2
[2] GlobalPlatform Card Specification v 2.2.1, January 2011.
[3] GlobalPlatform Card, Confidential Card Content Management, Card Specification v2.2 – Amendment A, v1.0.1, January 2011.
[4] Global Platform Secure Element Access Control v0.10.

Authors

First Author – Sarabjeet Singh, Masters of Computer Applications, sarabjeet.singh2610@gmail.com.