Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 5046284, 20 pages
http://dx.doi.org/10.1155/2016/5046284

Research Article

A Tokenization-Based Communication Architecture for HCE-Enabled NFC Services

Busra Ozdenizci, Kerem Ok, and Vedat Coskun

NFC Lab-Istanbul, Department of Information Technologies, ISIK University, 34980 Istanbul, Turkey

Correspondence should be addressed to Busra Ozdenizci; busra.ozdenizci@isikun.edu.tr

Received 28 April 2016; Revised 16 October 2016; Accepted 26 October 2016

Academic Editor: Laurence T. Yang

Following the announcement of Host Card Emulation (HCE) technology, card emulation mode based Near Field Communication (NFC) services have gained further appreciation as an enabler of the Cloud-based Secure Element (SE) concept. A comprehensive and complete architecture with a centralized and feasible business model for diverse HCE-based NFC services will be highly appreciated, particularly by Service Providers and users. To satisfy the need in this new emerging research area, a Tokenization-based communication architecture for HCE-based NFC services is presented in this paper. Our architecture proposes Two-Phased Tokenization to enable the identity management of both user and Service Provider. NFC Smartphone users can store, manage, and make use of their sensitive data on the Cloud for NFC services; Service Providers can also provide diverse card emulation NFC services easily through the proposed architecture. In this paper, we initially present the Two-Phased Tokenization model and then validate the proposed architecture by providing a case study on access control. We further evaluate the usability aspect in terms of an authentication scheme. We then discuss the ecosystem and business model comprised of the proposed architecture and emphasize the contributions to ecosystem actors. Finally, suggestions are provided for data protection in transit and at rest.

1. Introduction

Near Field Communication (NFC) is a promising short-range wireless communication technology notable for its significant contribution to the Internet of Things (IoT), Ubiquitous Computing, and Cloud Computing [1]. NFC is a short-range half duplex communication technology, which was initially developed in late 2002. NFC is compatible with RFID (Radio Frequency Identification) technology and ISO/IEC 14443 contactless smart cards. NFC communication occurs between two NFC compatible devices within a few centimeters using 13.56 MHz operating frequency [2].

NFC technology provides easy communication between various NFC devices on ISO/IEC 18000-3 air interface, at transfer rates ranging from 106 to 424 Kbits per second. The importance of NFC technology comes from its ease of use for initiating communication. In order to engage in an NFC communication, the user needs to touch her NFC Smartphone to either an NFC tag, another NFC Smartphone, or an NFC reader [3]. When the NFC Smartphone is touched to an NFC tag, her Smartphone reads/writes data from/to an NFC tag; when it is touched to another NFC Smartphone, they exchange data, and when it is touched to an NFC reader, the reader reads the data stored on the Smartphone. An operating mode name is given to each interaction: reader/writer mode to the tag interaction, peer-to-peer mode to the Smartphone interaction, and card emulation mode to the reader interaction [2, 3].

As the most promising NFC operating mode, card emulation mode enables a Smartphone to emulate a contactless smart card. Card emulation mode supports the realization of diverse applications like mobile payment, ticketing, coupon, loyalty, access control, identification, and so on. In this mode, Secure Element (SE) is focal area of the activity, which is defined as the area on NFC Smartphones for securely storing sensitive data (e.g., credit card, identity number, and loyalty data) needed to perform an NFC transaction. Several SE options have been proposed to date, including UICC-(Universal Integrated Circuit Cards-) based SE, embedded hardware-based SE, SD (Secure Digital) card-based SE, and Software-based SE. However, all the proposed and deployed SE options have created dependencies and disagreements

among stakeholders in the NFC ecosystem over time, which is explained in Section 2. Thus, reaching a fair solution among the stakeholders in this complex ecosystem is of the utmost importance.

The Cloud-based SE concept emerged with the introduction of Host Card Emulation (HCE) technology in Android 4.4 (KitKat) OS (Operating System). HCE technology separates the card emulation functionality from the hardware-based SE [4] and provides virtual representation of the sensitive data [5]. HCE uses a mobile OS to enable a virtual SE on the Cloud as a remote environment. The most important HCE deployment models are the Full Cloud-based Model and Tokenization-based Model, as described in Section 2.

HCE-based SE technologies are rather recent emerging concepts and only a few studies have been carried out to date. In [6], a remote server of SEs (i.e., SIM Server) termed the Cloud of Secure Elements (CoSE) is proposed. The study reports a Smartphone with HCE functionality remotely using an SE, hosted on a server, through the establishment of a secure TLS channel. The system is capable of resolving trust issues for Internet users, mobile applications, and virtual machine environments. Another related study [7] presents Tokenization-based architecture depending on the proposed CoSE for mobile payment platforms. In the study, Token-Generator applications are hosted in the CoSE, and Smartphones supporting HCE functionality remotely access these applications. In [8], an access control and management system for locking/unlocking doors is presented which benefits from NFC Smartphones with HCE functionality storing UIDs (i.e., unique key identification). The system is used for attendance control of students in educational institutions and for observing student location within the institution.

Due to the novelty of Cloud-based SE concept using HCE functionality, recent studies have focused on the Cloud-based SE concept for commercial implementations or research purposes for specific NFC service domains. In particular, recent commercial models and implementations mostly focus on HCE-based payment services and provide diverse business opportunities for actors in the domain of payment services. Existing models, standards, and specifications, which are applicable to payment services, cannot be used for non-payment services directly; they are not independent of the service type. Therefore, rigorous and comprehensive end-to-end communication and authentication architecture is still a promising avenue of research for promoting the development of diverse card emulation services using NFC.

The aim of this study is to fill the specified gap by presenting a promising and service independent communication architecture that can be used for diverse nonpayment HCE-based NFC services (e.g., access control, identification, loyalty, and membership). The presented communication architecture includes a unique Tokenization-based authentication mechanism, Two-Phased Tokenization which supports the identity management of both users and applications of Service Providers, and also provides a centralized, win-win business model. Using the presented communication architecture, Service Providers can easily deploy diverse HCE-based card emulation NFC services such as access control, identification, loyalty, and membership; and NFC Smartphone users can store, manage, and use their sensitive data on the remote environment of the Cloud and benefit from several NFC services easily. Also, with the presented architecture, NFC Smartphones can act as a significant object-based authenticator for users.

After a short introduction to the research topic, the remainder of this paper is organized as follows. In Section 2, a brief overview of SE and its alternatives, HCE technology, and HCE deployment methods, together with Tokenization systems and issues, are provided. In Section 3, our proposed novel communication architecture is introduced; system registration and system usage models are explained thereafter. We present an access control case study for meeting rooms, the implementation of the corresponding prototype, and the usability discussion of the proposed communication model in Section 4. In Section 5, we present the business and ecosystem implications of the proposed model and provide requirements for data protection. Section 6 concludes the paper.

2. Research Background

SE and HCE are important concepts for the card emulation mode of NFC services. In this section, SE and HCE technological issues and recent studies shaping our proposed architecture are examined and presented.

2.1. Secure Element. As the leading standardization organization in its domain, GlobalPlatform [9] defines SE as a tamper-resistant platform, typically a one-chip secure microcontroller, that is capable of securely hosting applications and their confidential and cryptographic data (e.g., keys) in accordance with the security requirements set forth by a set of well-identified trusted authorities.

The introduction of NFC technology with SEs created a completely new ecosystem with huge potential for increasing the usage of Smartphones in totally new forms, especially in financial transactions involving credit cards, digital money, and digital wallets [1]. SE has led to the development of new business models and partnerships pertaining to SE ownership and management issues.

In time, NFC ecosystem actors (e.g., MNOs, mobile handset manufacturers, smart card manufacturers, financial institutions, and transport institutions) tried to impose an alternative to SE using a specific business model from which they could benefit most. To date, several SEs including UICC, embedded hardware, and SD-based SEs have been proposed as a means of enabling secure card emulation services, as depicted in Figure 1. All such SE alternatives are hardware-based solutions.

The UICC option is the ad hoc model for providing SE infrastructure on NFC Smartphones. UICC-based SE obviously creates great advantages and opportunities for MNOs (Mobile Network Operators), since SIM cards are issued and managed by them. However, other stakeholders in the ecosystem did not accept the ownership and management of SE by MNOs, and attempted to implement alternative SE and business models.

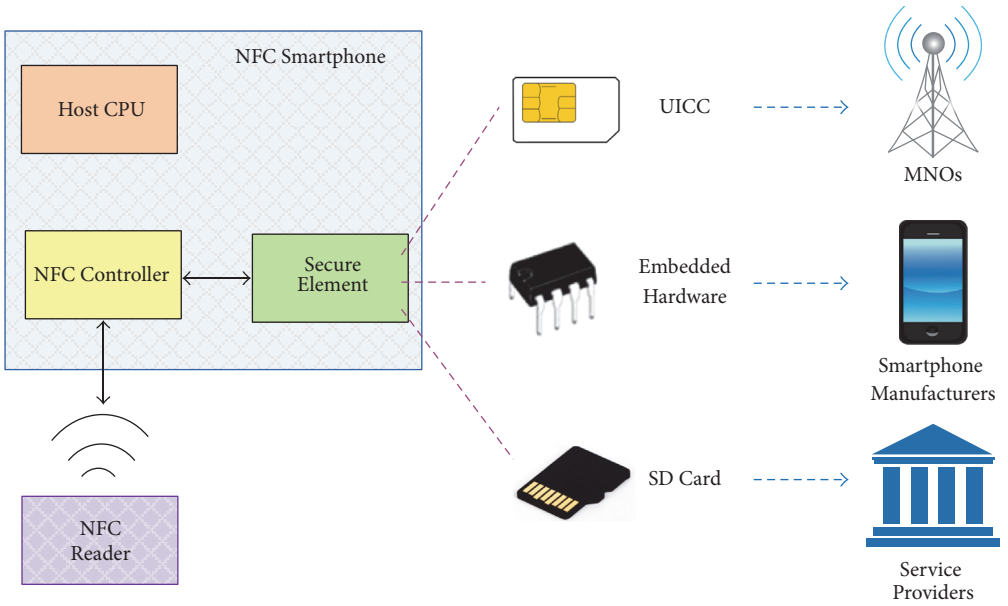


Figure 1: Hardware-based SE alternatives and SE ownership.

The next alternative was embedded hardware-based SE which is integrated into the Smartphone during the manufacturing process and can be personalized after the device is delivered to the end user [10]. This solution is obviously extremely advantageous for Smartphone manufacturers.

The latest hardware alternative was the Secure Digital (SD) card SEs, which mostly advantaged the Service Providers, since neither SIM card nor handset hardware is used as the SE [1]. This option was also unsuccessful and unpopular, since new hardware was required for each Smartphone.

Due to the disagreements and limitations of hardware-based SE alternatives, the NFC ecosystem actors and NFC standardization bodies tried to impose more independent solutions. Trusted Mobile Base (TMB) as Software-based SE was one of these. It was defined as a secure isolated section on the Core Processor Units of Smartphones [11]. In this alternative, the secure storage of the sensitive data on this section of Smartphones becomes again an important issue. Due to inadequate attention and support from stakeholders, this option also lost its popularity.

Each SE alternative enables different business model and meets different stakeholders' needs. It is obvious to say that dissonance among actors has limited the development of NFC technology and its card emulation based services; hence, more independent SE solutions alongside an acceptable business model are required.

In this context, storing valuable data on the Cloud instead of storing it on Smartphones and trying to achieve a more independent solution became an important effort for SE development which exposed Cloud-based SE concept. The recent technology on Smartphones, HCE (Host Card Emulation), is increasingly being considered as the initiator and enabler of a Cloud-based SE concept for card emulation based NFC services.

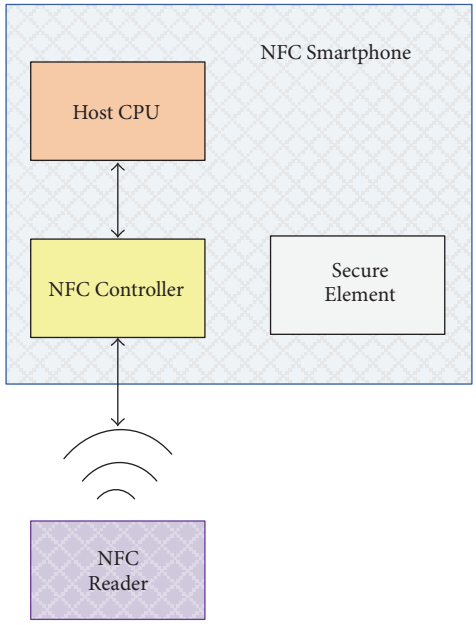


Figure 2: HCE Communication flow.

2.2. Host Card Emulation (HCE). HCE technology makes use of Cloud-based SE, where data are stored and managed on the Cloud instead of on the Smartphone. The Smartphone still performs card emulation functions but the private data are stored, secured, and accessed on the Cloud.

Currently, many Smartphone OSs support HCE, such as Android 4.4 (KitKat) and higher, Blackberry OS 7 and higher, and Microsoft Windows 10 [12]. HCE functionality is located in libraries and APIs of OS, which help developers to control the NFC interface and send commands to NFC devices [13]. As illustrated in Figure 2, an NFC controller routes the data

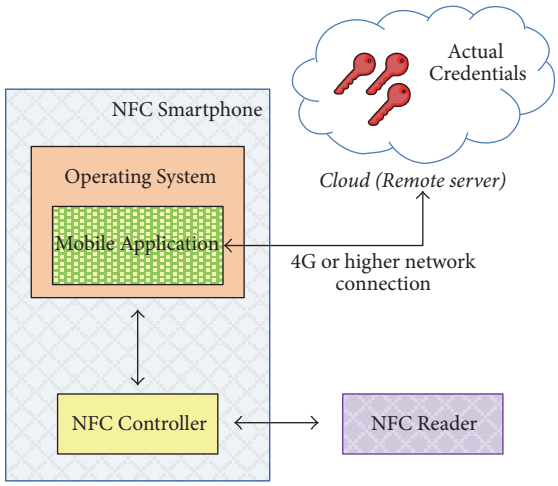


Figure 3: Full Cloud-based HCE solution.

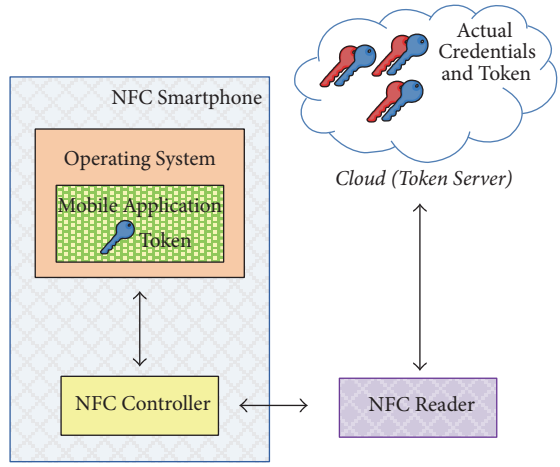


Figure 4: Tokenization-based HCE solution.

to host CPU instead of SE and, on host CPU, OS applications run which can process the related NFC communication.

The motivation behind HCE technology is its independence from hardware-based SE alternatives. In the case of hardware-based SEs, APDU commands received from an NFC reader are passed to the application on the SE of Smartphone with the help of an NFC controller, so that the SE processes the APDU commands and sends responses [13]. In case of HCE technology, the received APDU commands are passed to the active NFC application by the NFC controller as illustrated in Figure 2, and the mobile application processes the APDU commands received from NFC reader. HCE technology eliminates the need for a hardware-based SE, and the private data can be stored on a secure remote environment such as the Cloud.

With respect to computing capacity, storage capacity, deployment complexity, and cost, HCE-based NFC services are more advantageous when compared with hardware-based SEs [13]. Moreover, in terms of NFC ecosystem and business models, HCE-based solutions are independent of MNOs, Service Providers, and TSM (Trusted Service Manager); hence HCE technology can be considered as a game changer [14]. According to [15], HCE will make NFC more accessible and versatile to developers as well as more familiar to end users.

HCE technology is the preferred option for NFC-based mobile payment systems among other business case alternatives all over the world. Some important recent HCE implementations are Google Wallet in US, Tim Hortons in Canada, and BBVA in Spain [4].

There are two deployment models for HCE: Full Cloud-based model and Tokenization-based model [14].

In the Full Cloud-based HCE solution, the card emulation functionality is completely performed on the Cloud. The mobile application on the NFC Smartphone authenticates the user and enables secure connection to the remote server. A NFC Smartphone aiming to obtain the credentials from the Cloud needs to connect to the remote server repeatedly for each distinct transaction (Figure 3). As a matter of fact, in order to complete an NFC transaction more rapidly, the Smartphone's Internet connection should be as fast as possible (e.g., 4G, even 5G) [14]. In addition, another study [13] states that since the credentials can be exposed by malwares potentially resident on the device, the Full Cloud solution is not secure enough, and it increases risk especially in monetary applications.

To mitigate technical limitations and security risks, a second option based on Tokenization has emerged. Tokenization opens up the possibility of enabling more secure and efficient offline transactions. Tokenization replaces the actual data exchange by a token, which is a disguised representation of the original value [14, 16]. Threats via brute force attack to the tokens on mobile devices can be prevented by methods such as limiting the number of transactions or the validity time of each token. The details and requirements of a Tokenization system are also briefly described in next subsection.

First, a token is generated and saved on the mobile application. For each transaction, the mobile application sends the token to an NFC reader. The Service Provider of the NFC reader sends the token to the Token Service Provider (TSP) in order to obtain the actual credential, after which the Service Provider may authorize the transaction (Figure 4). The Smartphone does not need to access the Cloud, and transactions are entirely based on tokens. This scheme ultimately provides more secure communication [4, 14].

According to [4], the security of Tokenization-based HCE can be enhanced and supplemented through implementations such as white box cryptography, tamper proofing of software (i.e., tamper detection or temper resistance for software security), device fingerprinting, and other biometric techniques for authorization on mobile phones.

2.3. Tokenization. A token is a surrogate value which can be referred as a link to access to the actual data. When used, the actual data is stored in a remote environment and the token is used both as the necessary information to find the actual value on the remote environment and also as the proof for authentication permission to access the data.

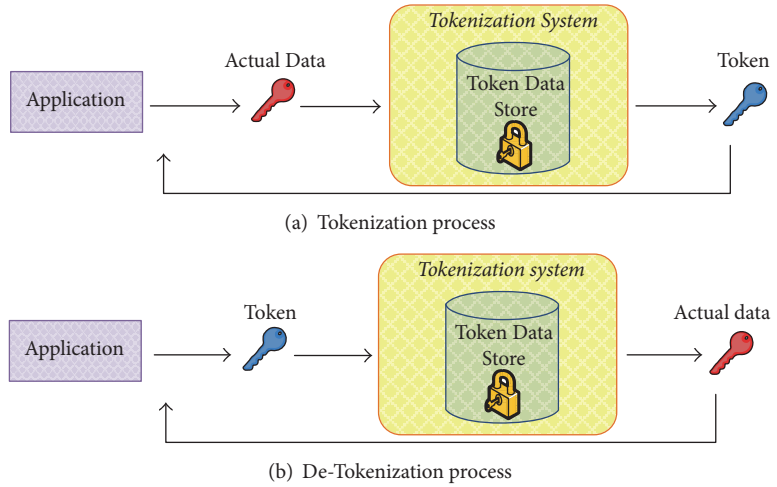


Figure 5: Tokenization system and processes.

In a Tokenization system, two processes are important: Tokenization and de-Tokenization. Tokenization is the process wherein the actual data is replaced with a surrogate value as token; and de-Tokenization is the reverse process of redeeming a token for its associated actual value [16, 17] as shown in Figure 5.

ASC X9 (Accredited Standards Committee X9), Visa, EMVCo (EuroPay, MasterCard, and Visa), and PCI DSS (Payment Card Industry Data Security Standard) have attempted to develop standards and specifications for the development of Tokenization systems especially in payment domain. Among others, PCI DSS provides the most widely-used standard for Tokenization systems used by payment industry stakeholders [16]. EMVCo has issued a Tokenization framework to describe the requirements for the creation and use of payment tokens (i.e., surrogate values that replace the Primary Account Number, PAN) in the context of digital transactions [17, 18]. The framework introduces a TSP that generates and resolves tokens.

According to existing standards and specifications, Tokenization systems have some common components and considerable issues: Token Generation, Token Mapping, Token Data Store (i.e., data vault), Encrypted Data Storage, and Cryptographic Key Management.

(i) *Token Generation*. In study [16], Token Generation is defined as the process of creating a token by using any method such as mathematically reversible cryptographic function based on strong encryption algorithms and key management mechanisms, one-way nonreversible cryptographic functions (e.g., a hash function with strong, secret salt), or assignment through a randomly generated number. Exchanging tokens instead of actual values is a popular approach for enabling the protection of sensitive data like credit card numbers; there is no direct relationship between the original value and the token, so the original data cannot be determined from the token [16, 18, 19]. Random Number Generator (RNG) algorithms are generally the most recommended solution for creating token values [16, 17]. According to the Federal Information Processing Standards (FIPS140-2), RNGs used for cryptographic applications typically produce a sequence of zero and one bits that may be combined into subsequences or blocks of random numbers [20]. They are easy to adapt to any format constraints and offer high security since the value cannot be reverse engineered. Thus, using random token values is a desirable solution in Tokenization systems.

(ii) *Token Mapping*. Token Mapping is the second important common component that refers the assignment of the generated token value to its original value. A secure cross-reference table needs to be established to allow authorized look-up of the original value using the token as the index [16].

(iii) *Token Data Store*. Token Data Store is a central repository for the Token Mapping process and stores original values and corresponding token values following the Token Generation process [16]. The sensitive data and token values need to be securely stored in an encrypted format on data servers. In addition, these servers need to provide efficient authentication services, return sensitive data, or restrict transactions as necessary.

(iv) *Encrypted Data Storage*. It is customary to encrypt the sensitive data at rest. The cryptographic algorithms are mainly classified as symmetric or asymmetric. The advantage of symmetric algorithms is their speed; however, the key management issue needs to be handled more efficiently due to same key usage for encryption and decryption. The most popular current symmetric encryption methods are AES and Triple DES. In the case of database encryption, two options exist: cell/column level encryption and Transparent Data Encryption (TDE). The cell/column level encryption technique is applied to individual columns/rows/cells within a database. It allows a data

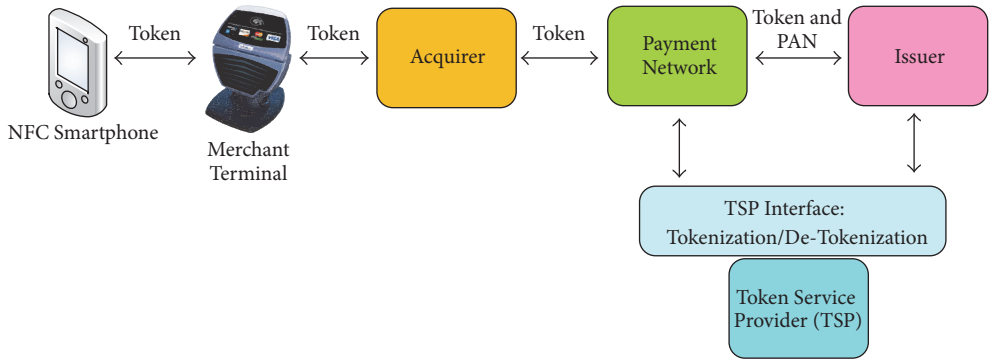


Figure 6: EMVCo specification: Tokenization-based payment for NFC transactions.

server to store data from different applications in the same database using different encryption keys [21–23]. TDE was first introduced in Microsoft SQL Server 2008 and is designed to provide protection to the entire database at rest, without affecting existing applications. It encrypts the entire database including the backups and log files using a single key that is called *database encryption key* and algorithms such as AES or Triple DES [21–23].

(v) *Cryptographic Key Management*. It is important to provide strong key management mechanisms for sensitive data encryption on Token Data Stores. The cryptographic keys should be created, managed, and protected; Token Servers need to have one or more unique keys to encrypt sensitive data. In this context, KMIP (Key Management Interoperability Protocol) is an important and popular standard for interoperable Cloud-based key management, which is provided by OASIS (Organization for the Advancement of the Structured Information Society). KMIP provides a comprehensive protocol for communication between enterprise key management servers and cryptographic clients [24]. It is essential to emphasize that data encryption and key management implementations complement the Tokenization method by protecting the original value.

2.4. Current Tokenization-Based Payment Systems. Introduced Tokenization specifications and standards are mainly for improving the security aspect of payment systems. EMVCo's Payment Tokenization Specification is an important specification for deployment of Tokenization solutions which benefits Acquirers, Merchants, Card Issuers, and Cardholders [17]. EMVCo defined several use cases in this document for payment transaction token flows; the Tokenization system architecture for NFC-based payment systems is illustrated in Figure 6. In accordance with EMVCo specification, current popular examples of Tokenization-based payment systems are Google Wallet [4, 25] and Apple Pay [26].

Google Wallet performs NFC transactions by using HCE technology. The token values are generated in the Cloud of Google and payment account information of customers is stored on the servers of Google; actual data are not shared with retailers [4, 25]. Google acts as an intermediary for NFC transactions. In case of Apple Pay, tokens are generated in Secure Element of the Smartphone and Apple does not store the actual data or token data in its own Cloud servers [26]. Merchant receives token and one-time-use security code from the customer and the token value is translated into credit card information (actual data) on the Payment Network who has the information about both the person and the transaction.

Each implementation deploys Tokenization with different business model and technical infrastructure. In this study, instead of using generic Tokenization system, a unique communication model (i.e., Two-Phased Tokenization) for HCE-enabled NFC services is provided that supports identity management of users and Service Provider's applications. The model centralizes system actors, diverse NFC services, access control, identification, loyalty, membership, and so on, on the same platform.

3. The Tokenization-Based Communication Architecture

The Cloud-based HCE concept using an efficient model promises great opportunities for promoting the development of the card emulation mode of NFC technology. As already mentioned in Section 1, no study provides a complete architecture and business model that can be applied to diverse NFC services using HCE functionality. The proposed architecture aims to provide secure communication architecture and also a centralized business model for HCE-enabled NFC services using Tokenization standards and specifications.

Our proposed business model consists of three actors: HCE-enabled Smartphone users, Service Providers supplying HCE-enabled NFC services, and the TSP (Token Service Provider). Figure 7 illustrates the relationship between actors in proposed business model. A Smartphone user may use diverse HCE-enabled NFC services provided by a Service Provider(s) and TSP centralizes all applications of a Service Provider(s) on the same platform. TSP is responsible for acting as a trusted entity by providing secure data storage service on the Cloud through a centralized data server (i.e., Token Data Store); it stores and manages the sensitive data

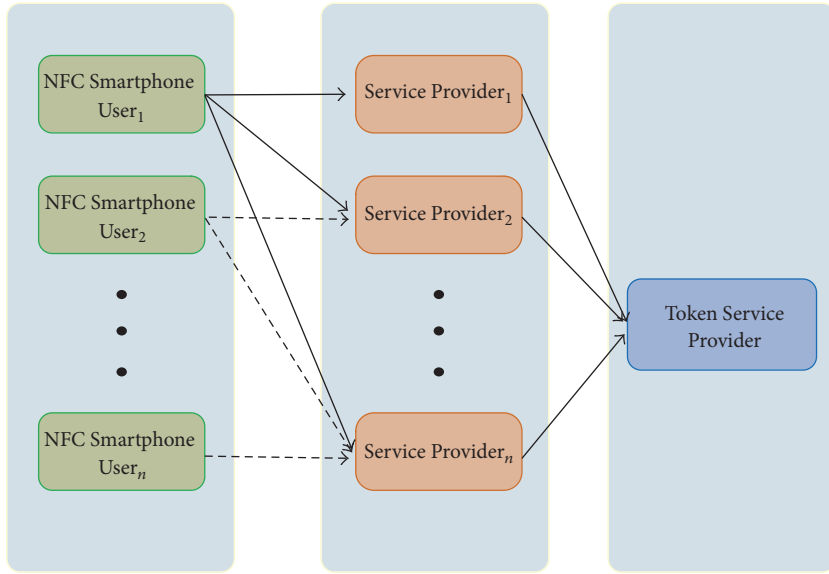


Figure 7: Proposed business model actors and relationship diagram.

of HCE-enabled Smartphone users with the corresponding token values securely and provides secure data storage and management and token mapping operations for applications of Service Providers.

In this context, TSP can be either an authorized data center provider or another authority that both performs Cloud Service Provider responsibilities and conforms to Tokenization system standards and requirements [16, 27, 28].

The proposed architecture ensures an efficient Tokenization-based authentication infrastructure for users and Service Providers through a centralized business model as shown in Figure 6.

(i) The user can securely store and use her sensitive data on the TSP server for the diverse card emulation based NFC services of Service Providers. In each NFC transaction, the user authenticates herself by sending the token value stored on her Smartphone to the Service Provider and receives authorization to use the Service Provider's corresponding NFC service.

(ii) Service Providers can offer several NFC services for users efficiently using TSP's authentication and communication infrastructure. In each NFC transaction, the Service Provider authenticates itself through a token value and requests authorization from TSP for the token value received from the user.

The originality of the proposed architecture is that in each NFC transaction both users and Service Provider authenticate themselves using token values. Depending on both of the token values, the TSP performs the de-Tokenization process and then sends an authorization response to the Service Provider upon an affirmative result. The described communication using Tokenization standards is entitled Two-Phased Tokenization that defines two major features of our architecture: User Identity Management and Application Identity Management, both of which are defined further as follows:

(i) User Identity Management aims to provide user authentication to the system. The NFC Smartphone stores a User Token (i.e., userToken as used in our model definition) that contains the user's identity data. When the user initially registers on the system, the User Token is first generated on the user's Smartphone using appropriate token generation algorithms described in Section 2, and it is subsequently transferred to the TSP's data server. Both the User Identity Data and the User Token are stored on the TSP's data server in encrypted form so that unauthorized parties cannot access them.

(ii) In Application Identity Management, identity management of distinct applications of each Service Provider is handled. Since the same Service Provider may have many HCE-enabled NFC services, a token value for each Application Token (i.e., appToken as used in our model definition) is used to identify and authenticate each application of the same Service Provider. The Application Token is generated by the TSP using token generation algorithms, described in Section 2, and shared with the Service Provider's backend system. The token is also stored on TSP's data server.

The proposed architecture is evaluated across two parameters: system registration and system usage. Both phases are described through generic models step by step below.

3.1. Registration Phase. Prior to using the application, the user needs to register on the system and receive a token from the TSP. The system registration phase is performed between the Smartphone and TSP. All communication between the

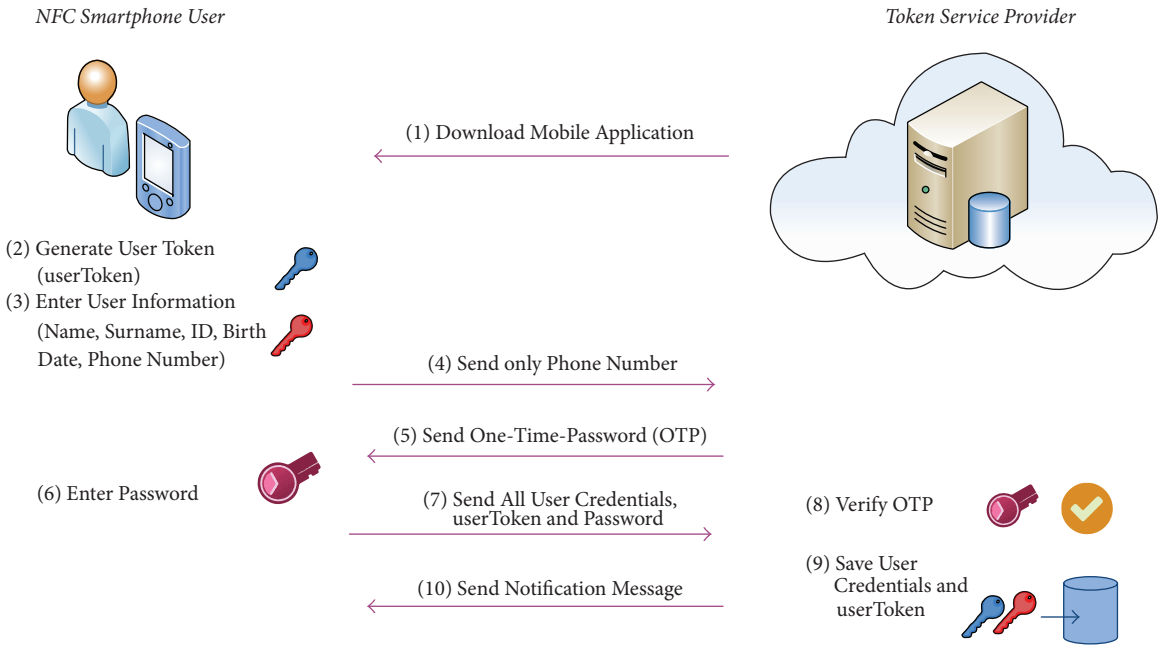


Figure 8: System registration generic model.

Smartphone and TSP is performed using a network connection. Only the communication in 5th step is performed using SMS (Short Messaging Service). In Figure 8, the system registration process is described step by step hereunder:

(1) The user downloads the TSP's mobile application onto her Smartphone.

(2) When the user runs the application, the application creates a token (userToken) by using an RNG algorithm as described in Section 2.

(3) The user enters her credentials including her ID, name, surname, birth date, and phone number (i.e., mobile subscriber number).

(4) The mobile application sends the Smartphone's phone number to the TSP.

(5) The TSP sends an OTP (One Time Password) to the Smartphone to validate the phone number.

(6) The user inputs the incoming password to the mobile application.

(7) The mobile application sends all user credentials (i.e., identity number, name, surname, birth date, and phone number), userToken, and password to the TSP.

(8) TSP checks and validates the phone number of user and password.

(9) If valid, then the TSP creates an SE area on the Cloud and saves the user credentials and the corresponding userToken.

(10) Finally, the TSP sends an approval message to the user and completes the registration process.

If the user is already registered and reinstalls the application, only steps (3) and (9) in Figure 8 differentiate for registration. The user only needs to enter her identity number and phone number information in Step (3); but on the other hand, the user does not need to re-enter other personal credentials since she already has a secure area on the Cloud. A new userToken is generated and is updated on the Cloud in Step (9) after verification of the user.

3.2. Usage Phase. The usage of the proposed architecture is illustrated in Figure 9 and explained below:

(1) An NFC Smartphone user touches a Service Provider's NFC reader (e.g., an access point, turnstile, or loyalty POS terminal). The NFC reader requests the user ID, after which the userToken value on the NFC Smartphone is sent to the reader.

(2) The NFC reader passes the userToken value to its backend system to authenticate the user and to get authorization.

(3) The Service Provider sends the corresponding application's appToken together with the userToken and requests authorization from the TSP.

(4) The TSP performs the de-Tokenization process, as described in Section 2, of userToken and appToken values on its data server and then authenticates the user and Service Provider's application.

(5) If the TSP authenticates the Service Provider's application, it sends an authorization response to Service Provider.

(6) The Service Provider transfers the authorization response to its own NFC reader.

(7) The NFC reader sends a verification and authorization message to the user's NFC Smartphone.

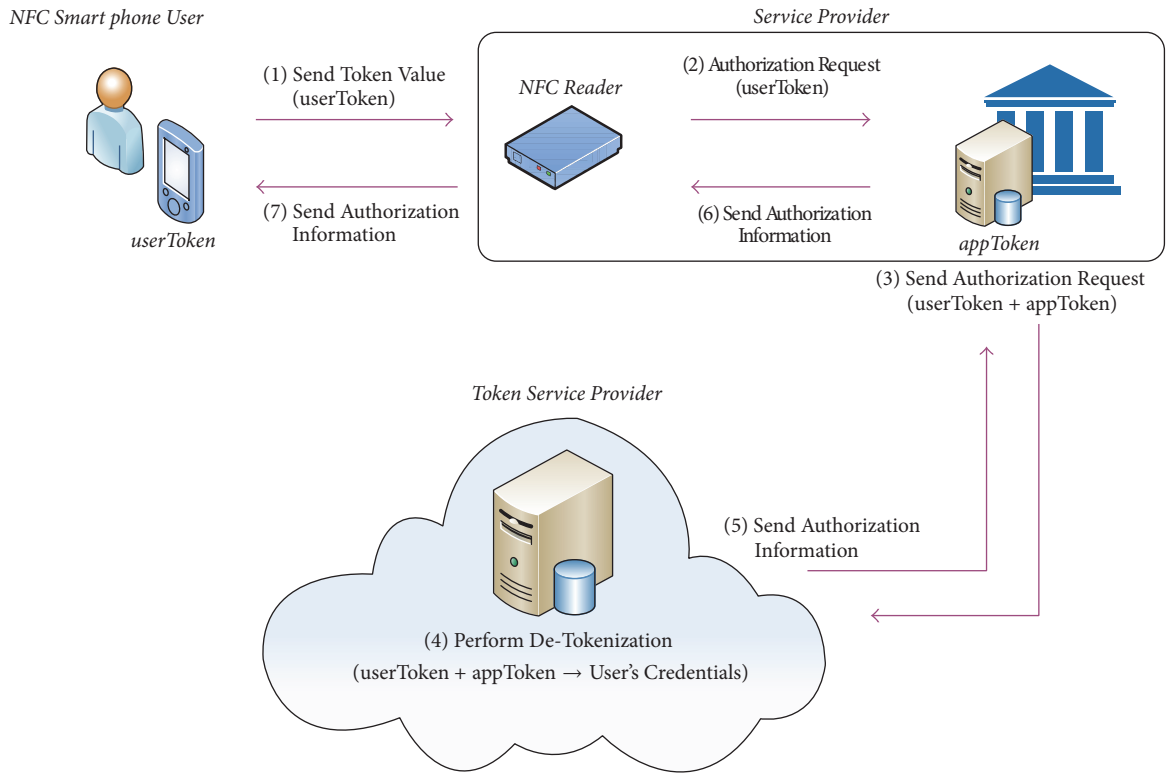


Figure 9: System usage generic model.

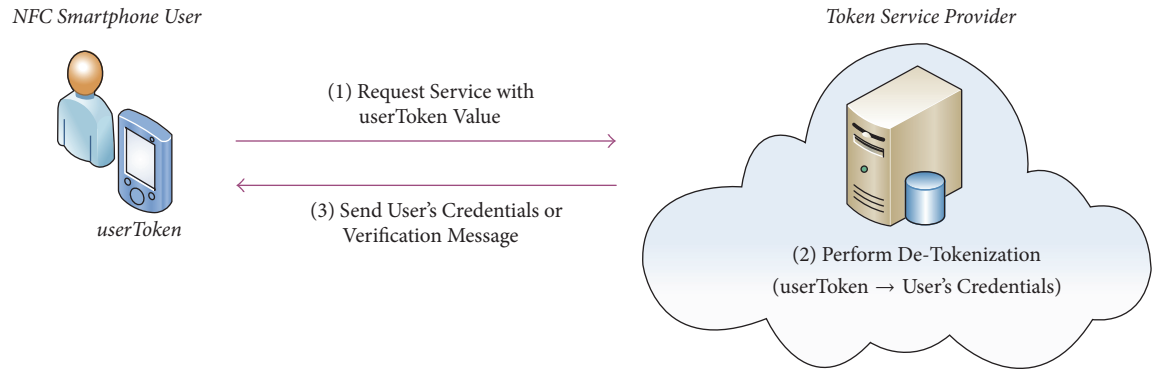


Figure 10: System usage through mobile application.

In the system usage context, users can also manage, control, and display their private data on the Cloud through the mobile application as illustrated in Figure 10:

(1) The NFC Smartphone user executes the mobile application. The mobile application sends the userToken and application service to the TSP.

(i) For access control applications, users may display their identity information and their company details; employees of a company may arrange meetings depending on their roles in that company, whether a manager or employee, and may check their past and upcoming meetings in their companies with the mobile application.

(ii) For loyalty and membership applications, users may display their membership status details, may check existing and upcoming campaign details, and may also view earned and used loyalty points.

(2) The TSP performs the de-Tokenization process of the userToken value on its data server, authenticates the user, and obtains the sensitive data of the user from the data server.

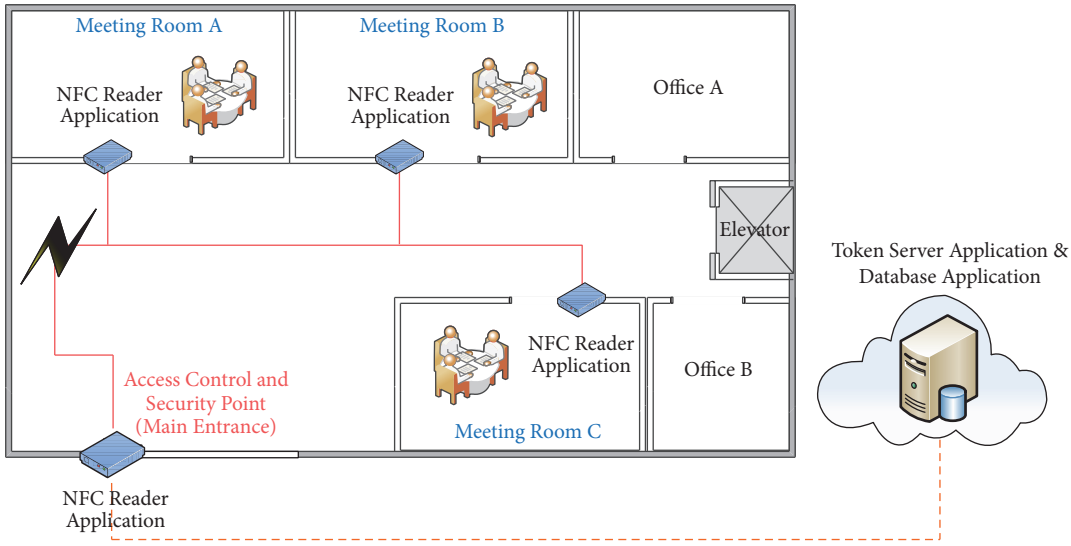


Figure 11: Initial setup of case study.

(3) The TSP sends the user her credentials and verification.

4. Case Study and Prototype Implementation: Access Control for Meetings

Access control systems are promising use cases of NFC technology. Instead of carrying identity cards, contactless smart cards, or other similar cards and devices, using the NFC capability of a Smartphone seems an attractive alternative; access permission occurs after touching the Smartphone to an NFC reader and authentication of the user.

Our proposed model also provides a practical solution for access control applications. In this context, we developed a meeting scheduling and access control prototype. Each staff member in the company is categorized as either a manager or an employee. Managers can schedule a meeting and assign employees to that meeting through the mobile application. Afterwards, employees can easily authenticate themselves and attend a meeting by simply touching their Smartphones to the NFC reader that is positioned near the entrance door of the meeting room. Along with providing authentication of employees and access control of meeting rooms, the case study also enables managers to track the meeting attendance of designated employees.

This research study was financially and technically supported by KocSistem Information and Communication Services Inc., Turkey's best-established IT company as well as its biggest domestic data center provider. Accordingly, the mentioned meeting room access control case study was inspired by the needs of KocSistem and designed as illustrated in Figures 11 and 12, which is explained in Section 4.1. The developed mobile application is named *KocSEC Mobile Application*, which is illustrated during our description of the case scenario in Section 4.2. Below, we initially explain the system design and the case scenario is explained in detail thereafter.

4.1. System Description. The developed system architecture has four main applications: Token Server Application, Database Application, NFC Reader Application, and KocSEC Mobile Application. The details on system and network architecture are presented in Section 4.4.

Figure 11 illustrates the design of the case scenario. An NFC reader including the specific appToken value of the organization was placed on the main entrance of the building/headquarters. The appToken value indicates the access control NFC service of KocSistem. Each meeting room in the building has a specific NFC reader that includes a unique Meeting Room ID (similar to appToken value). All NFC readers were connected to the TSP's data server via backend systems. On users side; users downloaded and installed the KocSEC Mobile Application and registered themselves to the system. The important design aspect of the mobile application is that users with different roles (employee, manager, and so on) were able to deploy different functions of our unique mobile application, as mentioned above.

4.2. Case Scenario. The case study is illustrated in Figure 12. The user with manager role arranges a meeting through KocSEC Mobile Application and designates attendees, so that the specified employees can access the meeting afterwards.

Let us examine the activity flow illustrated in Figure 12 step by step.

(1) The manager runs her KocSEC Mobile Application as shown in Figure 13(a). The manager selects KocSistem Access Control option from the list and then schedules a new meeting using the application. After the manager provides meeting details as shown in Figure 13(b), the list of company employees is displayed, from which the manager can select the attendees. When she submits it, the meeting is saved to the Cloud.

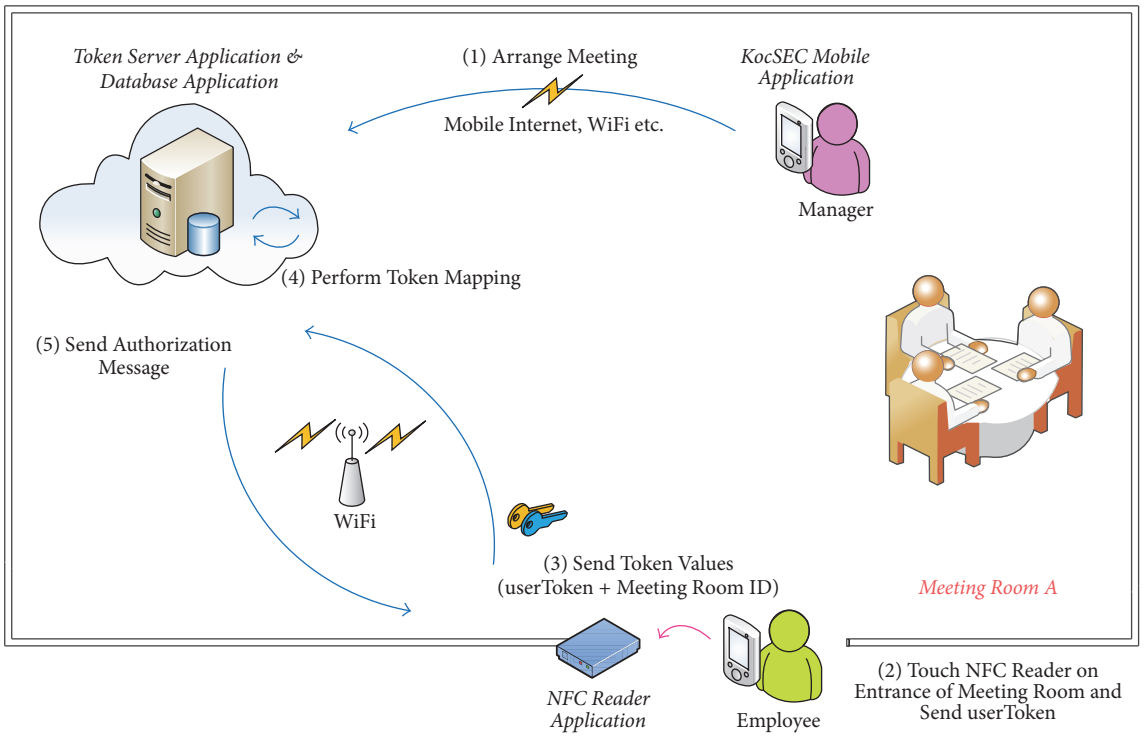


Figure 12: Case scenario usage flow.

(2) An attendee touches her Smartphone to the NFC reader of the meeting room.

(3) The NFC reader receives the userToken from the Smartphone and transfers it together with the Meeting Room ID to the Token Server Application for token mapping.

(4) The Token Server Application performs the token mapping process and authenticates the employee. The Token Server Application checks whether there is a meeting at that moment or not, as well as whether the employee is assigned to that meeting. Figure 14 shows a simple control activity flow performed on the Token Server Application.

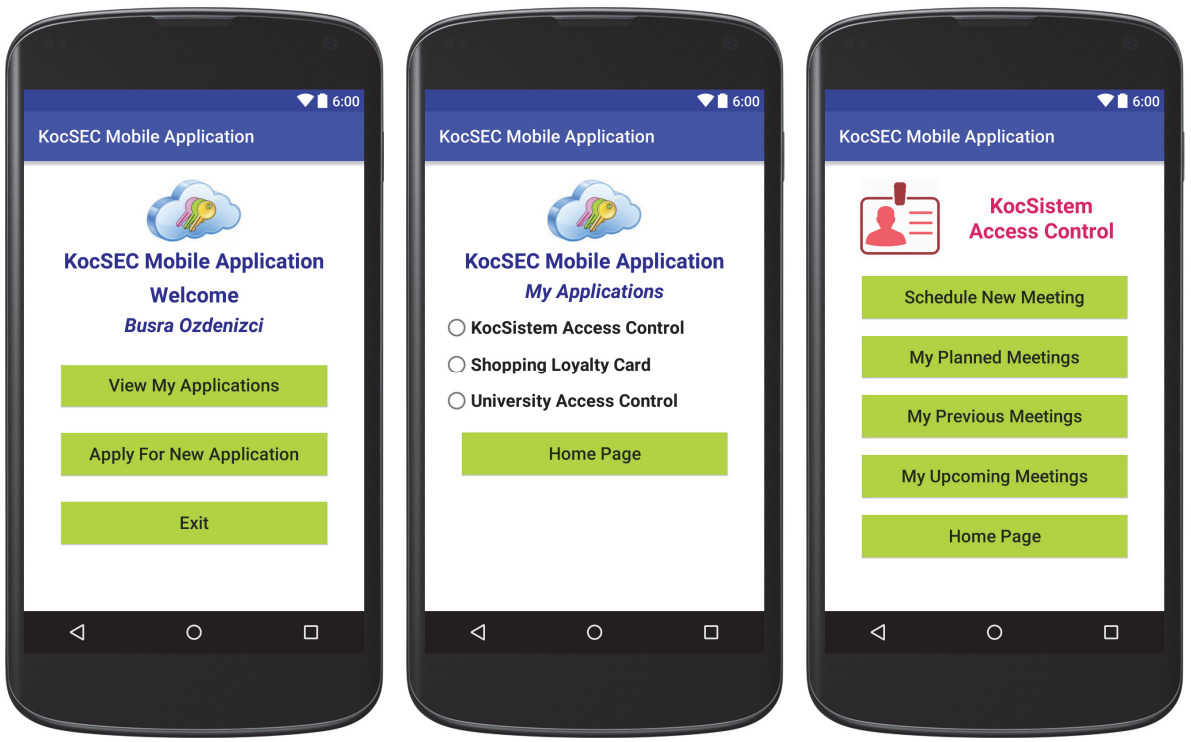
(5) Following the token mapping process, the Token Server Application sends an authorization message to the backend system of the NFC reader; the reader then sends the authorization message to the Smartphone. If the employee is not invited to that meeting, a notification message appears on the Smartphone. If the employee is assigned, the check-in or check-out time of the employee is displayed on the user's Smartphone, as seen in Figure 15.

4.3. Usability Evaluation of the Prototype. The proposed architecture provides an efficient authentication and centralized communication mechanism through the Tokenization method for HCE-enabled NFC services. In this section of the paper, a brief usability evaluation based on our study of the prototype implementation is provided, and the contributions to existing token authentication models are discussed for signifying the strength of the proposed scheme.

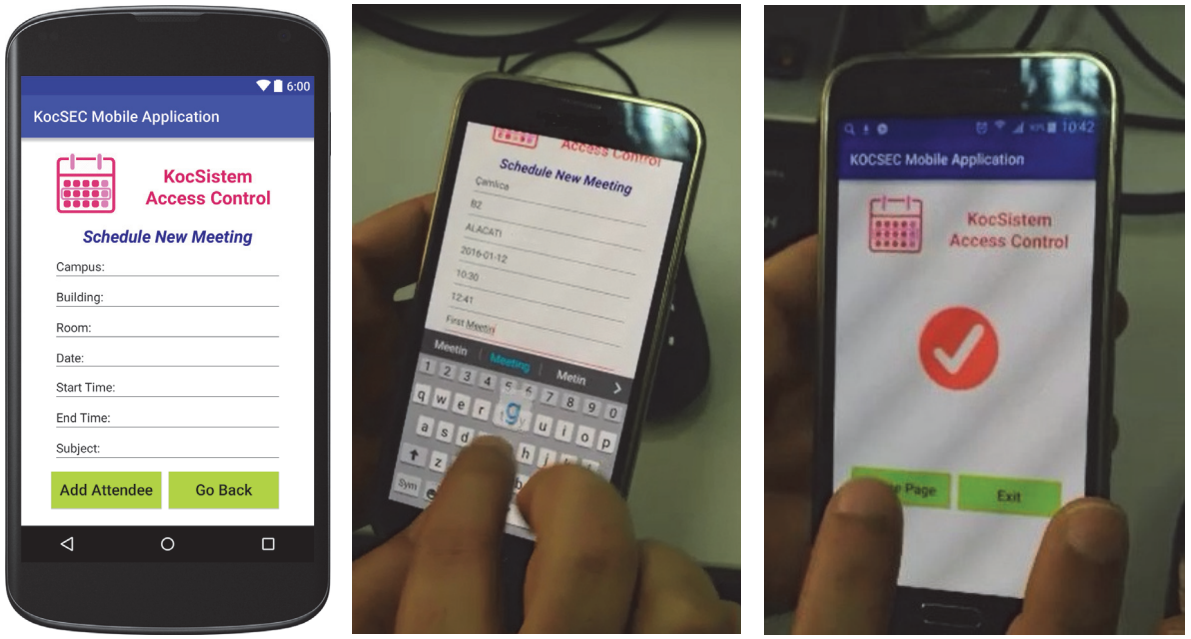
The Cambridge Technical Report study [29] provides a structured framework for evaluating user authentication schemes, specifically from unsupervised end user client devices (e.g., personal computers and mobile phones) to remote verifiers. The structured framework is inspired by inspection methods including feature inspections and Nielsen's heuristic analysis based on usability principles. The study [29] surveyed several specific authentication schemes like password management software, graphical passwords, hardware tokens, fingerprint recognition as biometrics, and others depending on the proposed framework.

In accordance with this study [29], we discuss the proposed architecture's user authentication performance in terms of the usability aspect, which is an important parameter for an acceptability of a system by users. In the same study, the usability of an authentication system is examined based on eight criteria.

(1) *Memory Wise Effortless*. This parameter measures the amount of information that the user needs to remember when using a system [29]. In our proposed architecture, all the user's private and sensitive data are stored on a trusted entity (the TSP) which is matched with a single token value. The Smartphone application generates this token value during the registration process and stores it for later system usage phases. The user does not need to remember all her sensitive data as well as her token value. The unique token value stored on her Smartphone helps



(a)



(b)

Figure 13: KocSEC Mobile Application Interfaces: schedule new meeting.

her to authenticate and use several NFC services: access control, security, membership, loyalty and couponing, and so on. Simply entering a password can be used to securitize the Smartphone and initiate the HCE functionality [4].

(2) *Scalability for Users*. This parameter measures the scalability of authentication schemes from the user's perspective, the burden placed on the users by each service [29]. In our model, the user can be authenticated for all NFC services with her single, unique

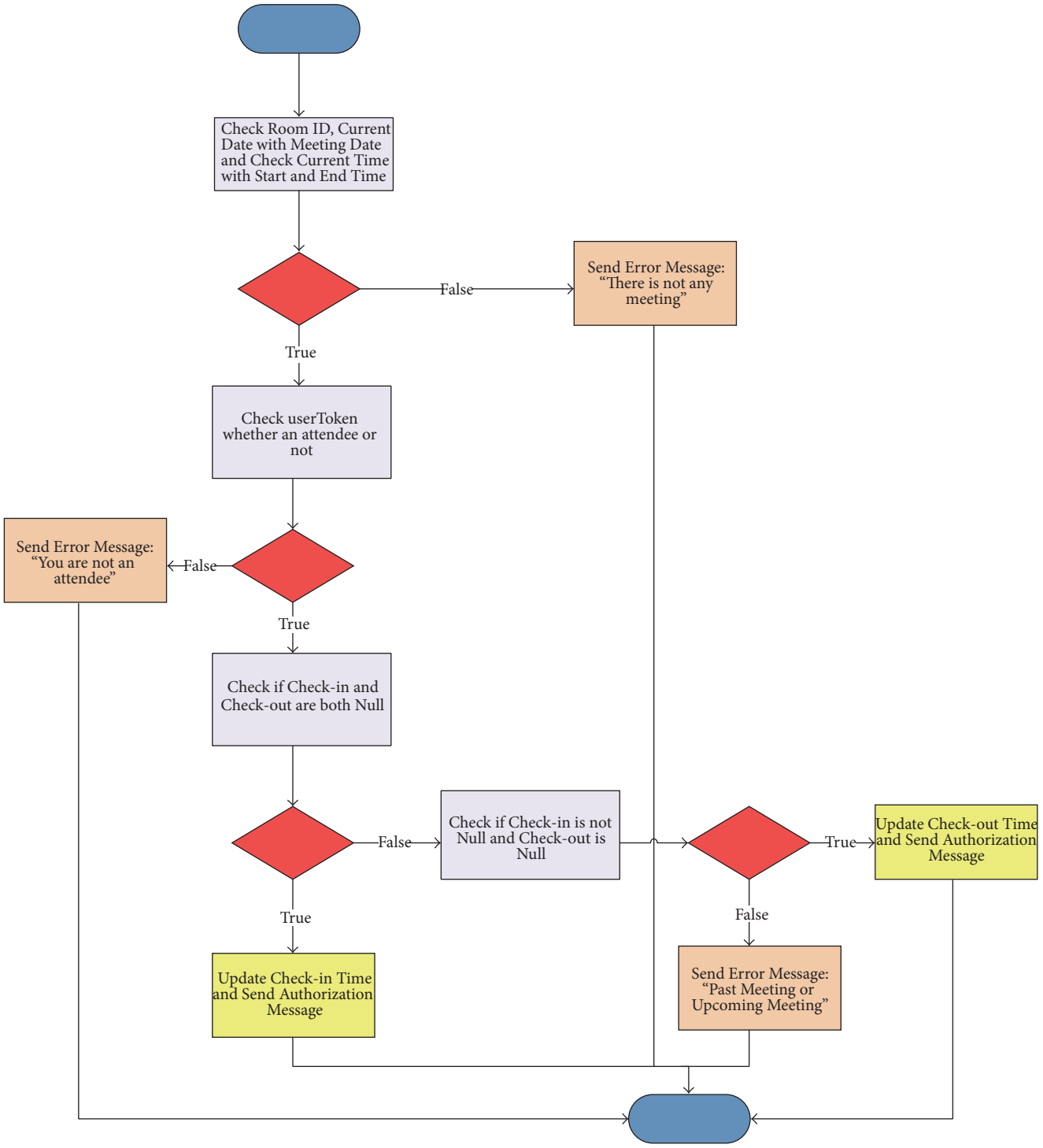


Figure 14: Flow of activities.

token value easily; hence it is obvious to say that the provided authentication and authorization scheme is scalable enough for NFC Smartphone users.

(3) *Physically Effortless*. This criterion measures how much physical effort is required of the user during the authentication process. In our case, the user only needs to enter a PIN or password to initiate the HCE functionality of her Smartphone and to touch her NFC Smartphone to the NFC reader. The user does not need to enter another username.

(4) *Nothing to Carry*. This refers to whether users need to carry an additional physical object [29]. The motivation for NFC services derives from this criterion as well. NFC enables users to carry all their sensitive data

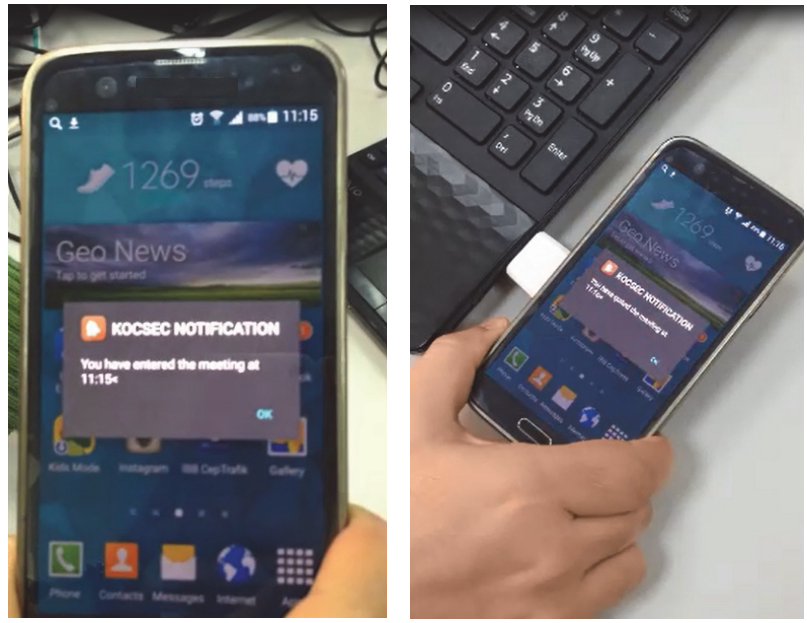


Figure 15: KocSEC Mobile Application Interfaces: attendees check-in and check-out.

on their Smartphones instead of carrying physical objects like smart cards, keys, credit card, debit cards, loyalty cards, membership cards, and so forth. The developed architecture will enable users to eliminate many physical objects by transferring those data to the SE on the Cloud.

(5) *Easy to Learn*. For prototype implementation, eight users participated in prototype testing. First, we requested that users register on the system as an existing user (i.e., with an existing ID and phone number and with a management role on the data server) through the mobile application; then participants scheduled a meeting and added attendees. Then the same users registered on the system as an existing user and performed check-in and check-out processes for the scheduled meeting. All participants performed registration processes without difficulty and scheduled a meeting successfully. All participants were already familiar with the NFC communication and NFC touch process; hence they easily learned and used the proposed system without requiring help.

(6) *Efficient to Use*. This parameter refers to the execution time that the user needs to spend for the authentication process. If user waits too long upon touching her NFC Smartphone to obtain authentication from the TSP, the system is evaluated as inefficient. During our case study implementation, the execution time between NFC touch and authorization retrieval from the TSP was approximately 2 seconds. For participants, this waiting duration was negligible. Moreover, this measurement shows that with the deployment of the proposed model by a data center provider, more solid and robust infrastructure with an efficient communication network, will be developed and supported.

(7) *Infrequent Errors*. This criterion pertains to system reliability; a reliable system does not reject requests from honest users. The proposed communication mechanism provides authentication and authorization depending on two different token values, userToken and appToken, and provides an efficient two-phased security mechanism, which increases the reliability of authentication and authorization.

(8) *Easy Recovery from Loss*. According to this criterion, if the credentials are forgotten, the user should conveniently regain the ability to authenticate [29]. If user loses her Smartphone, she can reregister for the system on a new Smartphone by choosing existing user option; userToken is generated again and updated on the Token Server of the TSP. The user can easily authenticate and reach her sensitive data on the Cloud in case of a loss; hence recoverability of the system in terms of users is high.

4.3.1. Comparison with Other Hardware-Based Token Mechanisms. Tokens enabling authentication are generally known as portable, hardware-based objects that store passwords, PINs, sensitive data, and so on. They are also known as hardware tokens, USB tokens, cryptographic tokens, software tokens, virtual tokens, key fob or smart card, and so forth.

The study [29] compares some specific hardware-based token schemes with traditional password based mechanism and emphasizes the drawbacks of hardware-based token mechanisms in terms of *Nothing to Carry* and *Ease Recovery from Loss* criteria. On the other hand, token based authentication schemes offer several benefits in terms of *Efficient to Use*, *Ease to Learn,* and *Infrequent Errors* criteria.

In our sense of the term, NFC Smartphone containing userToken value are construed as an *object-based authenticator* for users to authenticate themselves and obtain authorization for using their sensitive data on the Cloud. Especially in terms of *Nothing to Carry* and *Easy Recovery from Loss* criteria, our proposed model has considerable advantages when compared with hardware-based tokens.

In case of several services, the users may need to carry several physical, hardware-based objects as tokens; it is obvious that this situation is neither economical nor efficient for users. Due to the nature of NFC technology, Smartphones eliminate the need to carry a wallet or physical objects like smart cards, identification cards, keys, credit card, debit cards, loyalty cards, membership cards, wallet, and so forth. Indeed with the introduction of HCE technology, users can store their all these sensitive data on the remote environment of a Cloud Service Provider. At this point, the Tokenization system helps to create a secure communication and interaction environment. Our model proposes that an NFC Smartphone as a user authenticator can provide several NFC services.

Moreover, in the case of a lost or stolen token, the owner may misuse the sensitive data on a physical token, and these may not be easily recoverable. In such situation, the physical token must be revoked and a new token needs to be reissued, a time-consuming, costly, and undesirable process for the user. Also, enhanced and additional security mechanisms for HCE functionality on Smartphones [4] can be used to prevent undesirable outcomes. Popular examples for HCE security mechanisms are the usage of biometric factors (i.e., fingerprints, voice recognition, and facial recognition), entry of PIN or password, tamper proofing of software, and so on. Such implementations increase the durability and security of the proposed architecture in the event of a loss. Moreover, another important feature of HCE functionality on NFC Smartphones is that it becomes inoperable when the device screen is used in lock mode [30]. Hence, third parties cannot easily misuse the token value of a user on the NFC Smartphone. In cases of loss or theft, the user can update her userToken value on the data server of TSP by downloading the application on a new NFC Smartphone and registering as an existing user. The proposed system can be easily recovered in such a situation.

In accordance with the prototype testing and usability criteria evaluation, we regard the proposed communication architecture as a preferable and easy-to-use solution from the user's perspective. NFC Smartphones with HCE functionality can act as a significant object-based authenticator.

4.4. System Architecture and Efficiency Evaluation. As mentioned, the developed system architecture has four main applications: Token Server Application, Database Application, NFC Reader Application, and KocSEC Mobile Application. The developed system's architecture is illustrated in Figure 16.

The NFC Reader Application is developed using Java. ACR122U USB NFC Reader model is used for the prototype implementation which is a PC linked contactless smart card reader/writer and supports communication based on 13.56 MHz contactless technology; Mifare, ISO14443 A and B smart cards, NFC, FeliCa technologies.

The KocSEC Mobile Application is developed on the Android platform. Samsung Galaxy S5 Smartphones are used during testing which are HCE-enabled NFC Smartphones and using Android 5.0 (Lollipop) OS version.

The Token Server Application is developed using JSP (Java Server Pages). The GlassFish application server [31] is used for deployment of Token Server Application and Oracle Database is used for Database Application. The Token Server Application and Database Application are developed on remote servers separately by using Cloud service of Windows Azure [32]. Required firewall mechanisms are built between servers and applications for controlling the incoming/outgoing network traffic as well.

The Tokenization system including the Token Server Application and Database Application is developed by depending on the PCI DSS Tokenization Guidelines. A reliable token mapping process on database is provided by Token Server Application; after iterative development and testing processes an error-free performance is provided by the system. During case study implementation, it is obviously seen that when an attendee touches her Smartphone to the NFC reader of a meeting room, the Token Server Application performs mapping of userToken and Meeting Room ID values seamlessly all the time and sends the accurate results (i.e., authorization response).

In terms of time behavior and time performance, according to the system requirement analysis, the execution time between NFC touch and authorization retrieval from the TSP was designated as maximum 4 seconds. During case implementation, this execution time, including the mapping process of Token Server Application, is seen approximately 2 seconds; which can be considered as an efficient response time. This run time performance can be improved with advanced network architecture as well as with more efficient algorithm design.

5. Evaluation

The aim of this study was to provide valuable contributions towards the development of the card emulation mode of NFC services by making use of HCE technology. A promising communication architecture using Two-Phased Tokenization which accords to significant Tokenization standards is presented. Moreover, the prototype has been implemented as a real life case study.

The proposed architecture enables a centralized ecosystem environment for NFC technology. The architecture enables NFC Smartphone users to benefit from HCE-based NFC services as well as helping Service Providers provide diverse HCE-based NFC services through a centralized, independent communication infrastructure.

In this section, the proposed architecture is evaluated in terms of the ecosystem and business model perspective and, afterwards, its enablement of data protection is presented in order to understand the security aspects of the proposed model.

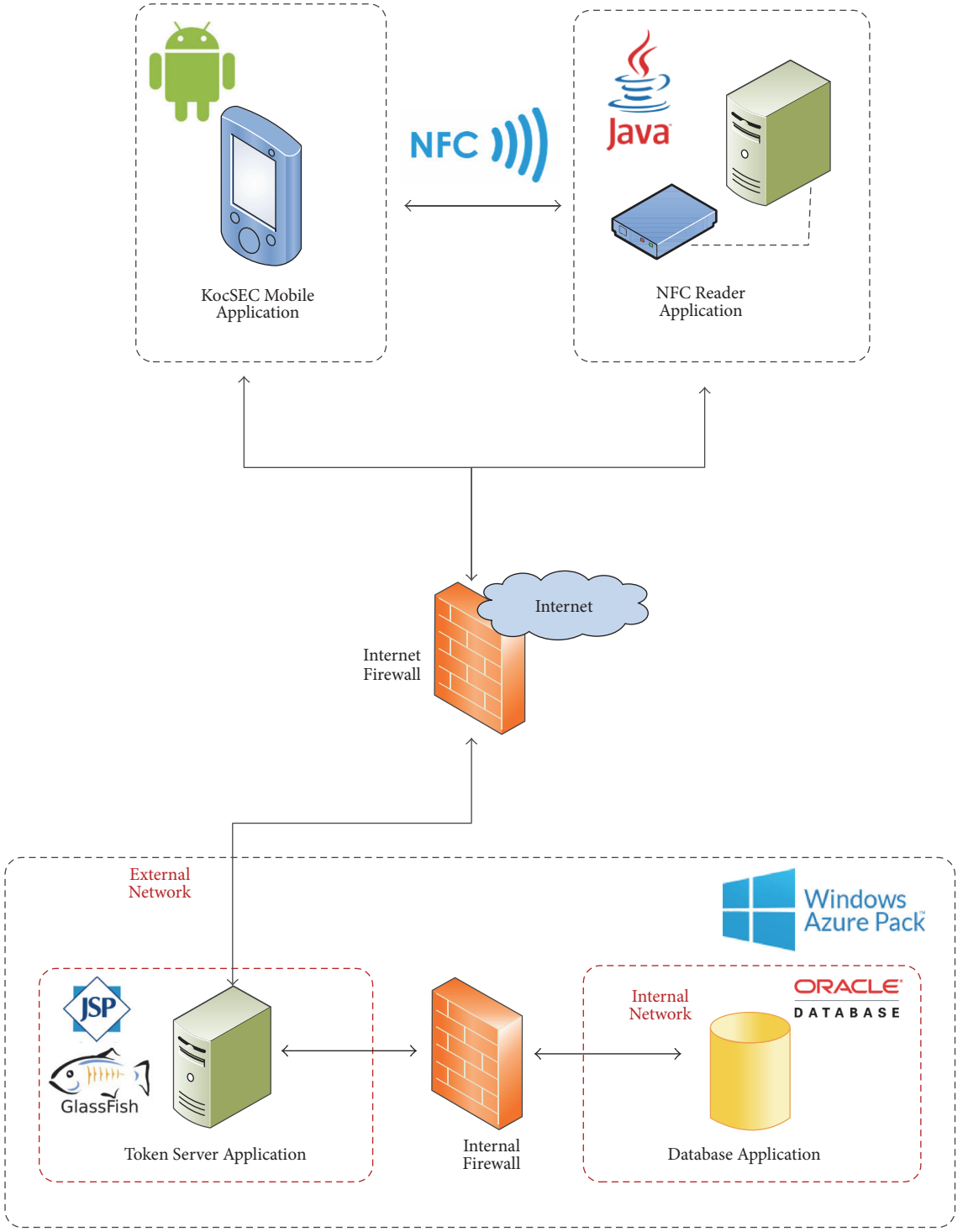


Figure 16: System development architecture.

5.1. Evaluation of Ecosystem and Business Perspective. The NFC ecosystem includes several organizations from diverse industries like MNOs, banking and payment services, device manufacturers, software developers, and other supplementary merchants including transport operators and retailers. All stakeholders in the NFC ecosystem agree on the fact that collaboration on an acceptable business model is crucial [1, 2]. Due to a lack of common understanding and vision in NFC technology among participating organizations, business models in the NFC ecosystem are underdeveloped. Companies need to cooperate with each other to create the added value first and compete with each other to take the biggest

share of it [33, 34]. A mutually beneficial business model especially for the card emulation services of NFC technology is yet to be sustained.

As mentioned in Section 1, the main reason behind this deficiency is the existence of different SE alternatives with different business models. Each stakeholder proposes a different business model that advantages that stakeholder above the others. Since SEs play important role in defining the business model for the NFC ecosystem, the stakeholders have attempted to develop more independent SE solutions. HCE technology on Smartphone OS is an important advancement in NFC technology and is referred to as a game changer by the Mobey Forum [14]. HCE functionality as an enabler of Cloud-based SE completely eliminates the need for an SE Issuer, SE Ownership, and MNO and creates considerable changes in the NFC ecosystem.

In an HCE-based NFC service business model (e.g., loyalty service), the Service Provider first needs to decide on a suitable HCE deployment model, which can be either Full Cloud-based model or Tokenization-based model. For all cases, a remote environment, Cloud infrastructure is required for enabling Cloud-based SE. When a Full Cloud-based model is used, the Service Provider needs to decide whether to develop the HCE and Cloud infrastructure in-house or engage expert HCE solution providers [14]. In case of the Tokenization-based model, which is the more secure and efficient HCE model, a robust Tokenization system depending on standards needs to be developed. Being a certificated and authorized entity for Tokenization is also important. In this context, TSP as a new actor engages with the ecosystem as mentioned in Section 3. TSP handles all Tokenization services depending on standards and enables a more secure HCE-based NFC service for the Service Provider. It is apparent that outsourcing a Tokenization system from an expert is a better option than performing in-house in terms of development, maintenance, and managerial costs.

On the other hand, a user who wants to benefit from a HCE-based NFC service (e.g., loyalty service) needs to obtain the application from her Service Provider. Since the user's sensitive data is stored on a remote environment, designated by the Service Provider, for both deployment models, the user needs to register for the application by contacting the Service Provider. If a Full Cloud-based model is used for HCE-based NFC service, the user certainly needs a fast Internet connection. In case of the Tokenization-based model, the card emulation operation is performed by the NFC reader and no Internet connection is needed by the Smartphone at the time of the NFC operation.

If we look at the big picture, there is a new, emerging, and competitive business environment between Service Providers. With the spread of HCE functionality on Smartphones, each Service Provider will try to impose its own HCE-based NFC service with a distinct deployment model and infrastructure; hence several TSPs or HCE solution providers may arise in the ecosystem. This chain will lead to a broad diversity of proprietary HCE-based NFC products and services in the market. On the other hand, NFC Smartphone users may be confused or reluctant to use these due to the diversity in the market. Some Service Providers have already started to provide proprietary HCE-based NFC services especially in payment, identity, and transit services using different TSP platforms [4, 12].

In our study, a centralized business model is provided for HCE-based NFC services through an efficient communication model. The proposed business environment is an important step for the development of centralized and structured ecosystem for HCE-based NFC model which includes three major actors: users, Service Providers, and TSP. We will now examine briefly the business model implications of our proposed model.

In the proposed business environment, only one TSP is assigned as the authorized, trusted, neutral entity that manages and secures the business environment as shown in Figure 6. One TSP performs the Two-Phased Tokenization operations and provides all required Cloud and data center services. Centralization of all communication on one TSP eliminates the possible conflicts between Service Providers and users which promote the development of more HCE-based NFC services. In this context, selection of appropriate TSP is important since the TSP will store all users' sensitive data as well as the Service Provider's applications data on its Cloud.

All Service Providers who want to provide HCE-enabled NFC services using the TSP's secure infrastructure need to make business agreements with their TSP. Each Service Provider must register itself and its application(s) to the TSP and obtain an Application Token for each application. This is an important step for Application Identity Management of the proposed Two-Phased Tokenization operation. For an HCE-based NFC Service of a Service Provider, the only cost is the NFC reader and backend server installation and maintenance. The Application Token value obtained from the TSP for that application needs to be uploaded to related server applications. All Service Providers' applications are supported by TSP's Tokenization system; hence the competitive environment between Service Providers will be handled. Another important advantage of the proposed architecture for Service Providers is that after de-Tokenization of User Token and Application Token values on TSP's server in each transaction, the authorization response is first directed to the Service Provider, enabling Service Providers to track transactions and retain their users' transaction details.

Also, users can easily register to the TSP. As mentioned, a User Token is generated during the registration phase for User Identity Management of the proposed Two-Phased Tokenization. The user needs to search for NFC services via the TSP application and apply for the NFC service that she wants to use through this. The user can easily manage her SE on the Cloud through mobile applications using the Two-Phased Tokenization operation. Also, in case of Smartphone loss, the user can easily get support from the TSP to disable her SE. She can register to the system as an existing user on a new Smartphone and generate a new token value; the old token value will be deleted from the Cloud.

The proposed centralized business model aims to provide a valuable road map for promoting HCE-based NFC services. It is crucial to say that HCE functionality carries important business opportunities for NFC ecosystem actors which

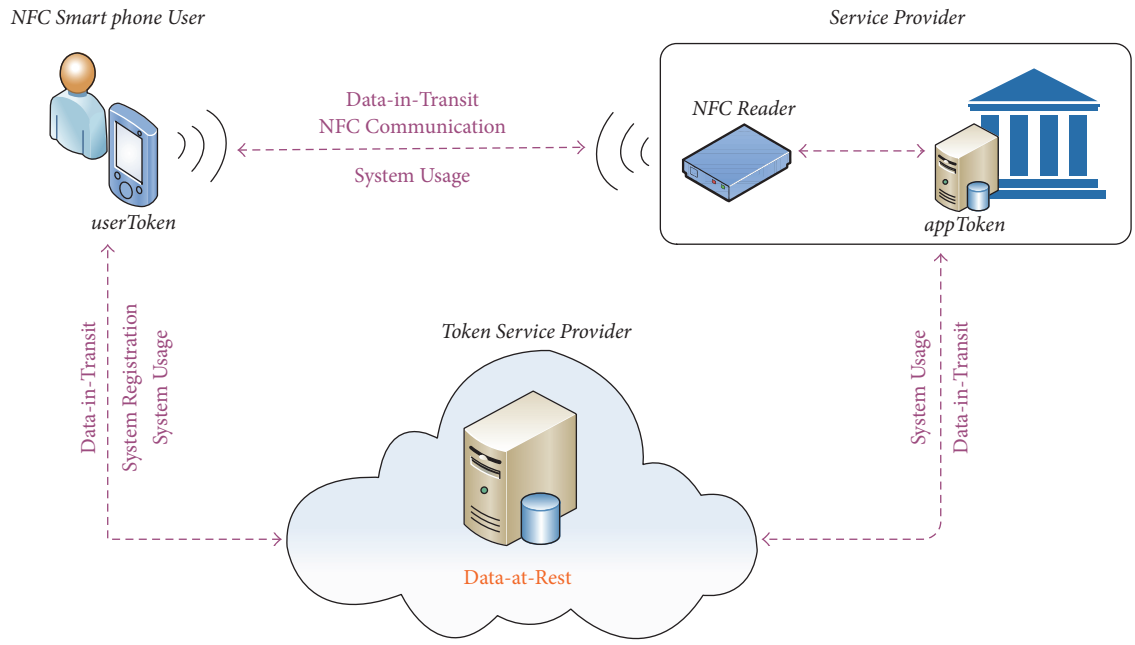


Figure 17: Data protection in transit and at rest.

should be evaluated in-depth. With the spread of diverse HCE-based NFC services, a centralized business model of HCE-based NFC services will become inevitable over time.

5.2. Discussion on Data Protection and Privacy. When users move and store their private data on a Cloud system, confidentiality becomes a critical issue. The Cloud-based system concept naturally carries diverse questions as to the risks of unauthorized disclosure, loss, modification, and unavailability of data. The vulnerability of sensitive data should be considered in terms of confidentiality, availability, and integrity [35–37].

In the proposed architecture, TSPs as Cloud Service Providers should satisfy ISO 27001 and ISO 27002 standards. In addition, there exist two new standards for Cloud Computing: ISO 27017 and ISO 27018. ISO 27017 deals with the application of the ISO 27002 standard to the use of Cloud services and to the provision of Cloud services [38]. ISO 27018 deals with the application of 27002 to the handling of Personally Identifiable Information (PII) [39]. TSPs should conform to these standards for enabling information security and protecting privacy in the Cloud.

Besides those standards, in the proposed architecture, data protection should be considered and discussed from Data-in-Transit and Data-at-Rest perspectives. Although in-depth security analysis is beyond the scope of this study, some suggestions pertaining to data protection in transit and at rest are outlined.

Data-in-Transit (DIT) is also referred in the literature as Data in Motion and Data in Flight [27, 37]. DIT refers to the protection of the confidentiality, integrity, and availability of sensitive data as they are moving or transiting across storage network, LAN, WAN, and so on. There are diverse and mature security standards for enabling encrypted data transmission: HTTPS for regular connections from Cloud service customers over the Internet to Cloud services, SFTP for bulk data transfers, and VPN using IPSec or Secure Sockets Layer (SSL) for connections to Cloud services [23, 27, 37].

As illustrated in Figure 17, data is in transit during system registration and system usage. In the case of system usage, userToken data transmission between Smartphone and NFC reader is initiated by NFC touch; NFC provides inherent security due to very close distance communication (up to few centimeters). Data transmission between the NFC reader and Service Provider's backend system and between the Service Provider's backend system and the TSP's data server is crucial; it should be performed over a secure communication link. The communication channel between the client and server can be supported with security standards such as TLS and SSL.

Similarly, the communication between the user and TSP for system registration and system usage should be secured by TLS or SSL. The communication should also benefit from an encrypted data transmission; a secure channel should be established between entities.

Data at Rest (DAR) is the protection of the confidentiality, integrity, and availability of the data residing on servers, storage, arrays, NAS appliances, and other media [23, 27, 37]. Along with secure data transmission, encrypted data storage is an important requirement for enabling data protection at rest (Figure 15).

There exist diverse encryption and key management mechanisms as described in Section 2. Appropriate encryption mechanisms should be applied according to standards such as US Federal Information Processing Standards Publication and the FIPS 140-2 Security Requirements for Cryptographic Modules [20].

TSPs also need to consider database encryption, field level encryption, or transparent data encryption [22, 23, 40]. Cell level encryption provides more granular level encryption; if the amount of data to be encrypted is very small or if the request can be custom-designed and the performance level is not a concern, cell level encryption is recommended over TDE. TDE encryption is recommended for encrypting existing high performance applications or for sensitive applications [21–23].

Both implementations require appropriate key management mechanism; encryption keys should be managed appropriately using a standard. OASIS KMIP provides useful guidelines to management of encryption keys across diverse infrastructures [24].

6. Conclusions

To date, several SE alternatives have been deployed for the development of card emulation based NFC services; however, each alternative creates advantages and opportunities in terms of SE ownership and management for different stakeholders in the ecosystem. To eliminate these dependencies and to reach a more acceptable solution, the Cloud-based SE concept emerged with the introduction of HCE technology.

HCE completely separates the card emulation functionality from hardware-based SE and enables storage of sensitive data on the Cloud with different deployment models. In this context, Tokenization as a security method has important contributions for promoting HCE-based NFC services. Diverse efforts have been made to standardize (i.e., ASC X9, PCI DSS, Visa, and EMVCo) the Tokenization method, especially in the domain of mobile payment services.

Due to the novelty of Cloud-based SE using the HCE functionality concept, a comprehensive architecture for diverse HCE-based NFC services is indubitably required to fill the gap in this emerging area of research. In accordance with these standards, we propose an innovative Tokenization-based communication for HCE-based NFC services: access control, identification, loyalty, and membership applications. The proposed model aims to provide an efficient authentication mechanism for both users and Service Providers through a Two-Phased Tokenization model and enables NFC Smartphone users to store, manage, and use their sensitive data on the Cloud for NFC services. Our study presents two main phases of the architecture, namely, system registration and system usage. It also includes a prototype implementation and testing to study the viability of proposed architecture and a brief usability evaluation. The study concludes with an evaluation of the ecosystem and business perspective and suggestions for data protection issues.

Together with an efficient authentication using the Two-Phased Tokenization model, the proposed communication architecture provides a centralized, win-win business model for promoting diverse card emulation based NFC services. NFC Smartphone users can benefit from diverse HCE-enabled NFC services by storing their sensitive data on the remote environment of TSP, and Service Providers can easily deploy diverse HCE-based card emulation NFC services, such as access control, identification, loyalty, and membership, by using the Tokenization-based communication architecture of TSP.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work is funded by KocSistem Information and Communication Services Inc. and Turkish Ministry of Science, Industry and Technology under SAN-TEZ Project no. 0726.STZ.2014.

References

[1] V. Coskun, B. Ozdenizci, and K. Ok, "The survey on near field communication," *Sensors*, vol. 15, no. 6, pp. 13348–13405, 2015.

[2] V. Coskun, K. Ok, and B. Ozdenizci, *Near Field Communication (NFC): From Theory to Practice*, John Wiley & Sons, London, UK, 1st edition, 2012.

[3] B. Ozdenizci, V. Coskun, and K. Ok, "NFC internal: an indoor navigation system," *Sensors*, vol. 15, no. 4, pp. 7571–7595, 2015.

[4] Smart Card Alliance Mobile and NFC Council, "Host Card Emulation (HCE) 101, White Paper," 2014, http://www.smartcardalliance.org/wp-content/uploads/HCE_Webinar_FINAL_061815.pdf.

[5] N. Prakash, "Host card emulation," *International Journal of Scientific and Research Publications*, vol. 5, no. 8, pp. 1–3, 2015.

[6] P. Urien, "Cloud of secure elements: an infrastructure for the trust of mobile NFC services," in *Proceedings of the 10th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '14)*, pp. 213–218, Larnaca, Cyprus, October 2014.

[7] P. Urien, "Towards token-requestor for epayment based on cloud of secure elements and HCE mobiles," in *Proceedings of the IEEE 1st Conference on Mobile and Secure Services (MOBISECSERV '15)*, pp. 1–2, Gainesville, Fla, USA, February 2015.

[8] N. Saparkhojayev, A. Nurtayev, and G. Baimenshina, "Access control and management system based on NFC-technology by the use of smart phones as keys," *Middle-East Journal of Scientific Research*, vol. 21, no. 7, pp. 1130–1135, 2014.

[9] GlobalPlatform, https://www.globalplatform.org/mediaguideSE.asp.

[10] M. Reveilhac and M. Pasquet, "Promising secure element alternatives for NFC technology," in *Proceedings of the 1st International Workshop on Near Field Communication (NFC '09)*, pp. 75–80, Hagenberg, Austria, February 2009.

[11] L. Kanniainen, "Alternatives for banks to offer secure mobile payments," *International Journal of Bank Marketing*, vol. 28, no. 5, pp. 433–444, 2010.

[12] SmartCard Alliance, "Host Card Emulation: An Emerging Architecture for NFC Applications," 2015, http://www.smartcardalliance.org/activities-events-host-card-emulation-an-emerging-architecture-for-nfc-applications/.

[13] M. Alattar and M. Achemlal, "Host-based card emulation: development, security and ecosystem impact analysis," in *Proceedings of the IEEE International Conference on High Performance Computing and Communications*, Paris, France, August 2014.

[14] Mobey Forum, The Host Card Emulation in Payments: Options for Financial Institutions, White Paper, http://www.mobeyforum.org/the-host-card-emulation-in-payments-options-for-financial-institutions-3/.

[15] SIM Alliance, Secure Element Deployment & Host Card Emulation, Version 1.0., 2015, http://simalliance.org/wp-content/uploads/2015/03/Secure-Element-Deployment-Host-Card-Emulation-v1.0.pdf.

[16] PCI DSS, "Tokenization Guidelines Version 2.0," 2011, https://www.pcisecuritystandards.org/documents/Tokenization_Guidelines_Info_Supplement.pdf.

[17] EMVCo, EMV Payment Tokenization Specification, Technical Framework, 2014, https://www.emvco.com/specifications.aspx?id=263.

[18] D. Ortiz-Yepes, "A critical review of the EMV payment tokenisation specification," *Computer Fraud & Security*, vol. 2014, no. 10, pp. 5–12, 2014.

[19] B. R. Williams, "How tokenization and encryption can enable PCI DSS compliance," *Information Security Technical Report*, vol. 15, no. 4, pp. 160–165, 2010.

[20] FIPS140-2 Security Requirements for Cryptographic Modules, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

[21] W. Hubis, An Introduction to Key Management for Secure Storage, White Paper, http://www.snia.org/sites/default/files/Hubis-W_Introduction_to_Key_Management.pdf.

[22] SANS, Transparent Data Encryption: New Technologies and Best Practices for Database Encryption, White Paper, https://www.sans.org/reading-room/whitepapers/analyst/transparent-data-encryption-technologies-practices-database-encryption-34915.

[23] "Securosis, Understanding and Selecting a Database Encryption or Tokenization Solution," White Paper, https://securosis.com/assets/library/reports/Securosis_Understanding_DBEncryption.V_.1_.pdf.

[24] OASIS KMIP, Key Management Interoperability Protocol, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=kmip.

[25] "Google Wallet," https://www.google.com/wallet/.

[26] Apple Pay, https://developer.apple.com/library/content/ApplePay_Guide/index.html#//apple_ref/doc/uid/TP40014764-CH1-SW1.

[27] M. Ramachandran and V. Chang, "Towards performance evaluation of cloud service providers for cloud data security," *International Journal of Information Management*, vol. 36, no. 4, pp. 618–625, 2016.

[28] C. Tang and J. Liu, "Selecting a trusted cloud service provider for your SaaS program," *Computers & Security*, vol. 50, pp. 60–73, 2015.

[29] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: a framework for comparative evaluation of web authentication schemes," in *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, pp. 553–567, San Francisco, Calif, USA, May 2012.

[30] Android Developers, Host-based Card Emulation, http://developer.android.com/guide/topics/connectivity/nfc/hce.html.

[31] GlassFish Server, https://glassfish.java.net/.

[32] Windows Azure, https://azure.microsoft.com.

[33] G. Madlmayr, J. Langer, and J. Scharinger, "Managing an NFC ecosystem," in *Proceedings of the 7th International Conference on Mobile Business*, pp. 95–101, Barcelona, Spain, July 2008.

[34] K. Ok, V. Coskun, B. Ozdenizci, and M. N. Aydin, "A role-based service level NFC ecosystem model," *Wireless Personal Communications*, vol. 68, no. 3, pp. 811–841, 2013.

[35] N. Kshetri, "Privacy and security issues in cloud computing: the role of institutions and institutional evolution," *Telecommunications Policy*, vol. 37, no. 4-5, pp. 372–386, 2013.

[36] Cloud Standards Customer Council, Security for Cloud Computing Ten Steps to Ensure Success, White Paper, http://www.cloud-council.org/deliverables/CSCC-Security-for-Cloud-Computing-10-Steps-to-Ensure-Success.pdf.

[37] Cloud Security Alliance, "SecaaS Implementation Guidance Category 8: Encryption," https://cloudsecurityalliance.org/download/secaas-category-8-encryption-implementation-guid-ance/.

[38] ISO 27017, Information Technology—Security Techniques—Code of practice for information security controls based on ISO/IEC 27002 for cloud services.

[39] ISO 27018, Information Technology—Security Techniques—Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors.

[40] E. Shmueli, R. Vaisenberg, E. Gudes, and Y. Elovici, "Implementing a database encryption solution, design and implementation issues," *Computers & Security*, vol. 44, pp. 33–50, 2014.