IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 8, 2020, accepted December 19, 2020, date of publication January 1, 2021, date of current version January 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048711

Exploring Half-Duplex Communication of NFC Read/Write Mode for Secure Multi-Factor Authentication

MOHD SAMEEN CHISHTI[1], CHUNG-TA KING[2], (Senior Member, IEEE), AND AMIT BANERJEE[1], (Member, IEEE)

[1]Department of Computer Science, South Asian University, New Delhi 110021, India
[2]Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan

Corresponding author: Amit Banerjee (amit@cs.sau.ac.in)

ABSTRACT Near Field Communication (NFC) is a prominent short-range, contact-less communication technology, which is rapidly getting popular in modern smart devices. For communication between two active devices via NFC, applications generally choose the peer-to-peer operation mode. In this paper, we exploit the possibility of using the NFC read/write mode, designed primarily for unidirectional data transfer from an active NFC reader to a passive NFC tag, for bi-directional half-duplex communication between two active NFC devices. The advantages of using the NFC read/write mode include low protocol overhead and permitting different data formats. However, the challenges are avoiding the reader collision problem, maintaining a secure session, and completing all transactions in an acceptable time frame. In this paper, we address the above challenges and propose a methodology for efficient communication between active NFC devices using NFC read/write mode. To evaluate the scheme, we design a secure Multi-Factor Authentication (MFA) system that requires bi-directional communication for mutually authenticating two NFC devices. The proposed methodology is experimentally verified using NFC-enabled Android smartphones and a Kerberos server as the third-party authenticator.

INDEX TERMS Half-duplex communication, multi-factor authentication (MFA), near field communication (NFC), NFC read/write mode.

I. INTRODUCTION

Near Field Communication (NFC), based on radio-frequency identification (RFID) technology, is commonly integrated in many modern smart devices for short-range, contact-less communication [1]. NFC provides a mechanism for secure communication in close physical proximity ($< 10\ cm$). Today, NFC is used for providing various services, such as secure payment and opening car doors [2], [3]. In these applications, an active Radio frequency identification (RFID) reader (embedded in a smart lock) communicates with a passive RFID tag (a smartcard), to determine the access privileges and provide services to the user [4]. The primary modes of NFC communication are the peer-to-peer mode and read/write mode. The peer-to-peer mode supports bi-directional communication between two active devices. Whereas, the read-write mode is used for uni-directional data transfer between an active NFC reader and a passive tag, such as for reading/writing URL or business card information from/to a tag [5]. Nonetheless, the peer-to-peer mode various limitations, such as high protocol overhead, use of proprietary libraries, and the inability to handle multiple data formats. This is not true for the read-write mode [6], [7], which motivates us to investigate it further for enhancements.

In particular, the goal in this paper is to extend the boundaries of NFC read/write mode to perform half-duplex bi-directional communication between two active NFC devices. This can help the devices to overcome the limitation of tag size and can share data in different file formats, such as encrypted data. The proposed methodology can be used in most RFID technologies that support the read/write mode, such as low frequency (LF) RFID, high frequency (HF) RFID, and ultra high frequency (UHF) RFID [8]. Although, there are other wireless technologies,

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Yu.

such as Bluetooth and WiFi, that can be used for handling different file formats [9], [10], but these technologies are distinct in design and purposes. More specifically, as mentioned above, NFC is based on RFID, whereas Bluetooth/WiFi are based on spead-spectrum technology. In addition, NFC has a very short communication range and requires low power for functioning [11]. Moreover, NFC do not require pairing of the devices for sharing information [12], and provides more security in comparison to other wireless communication technologies [13].

The proposed model uses an intermediate tag in between two NFC enabled devices for bi-directional transfer of data in multiple cycles. The initiator of the communication write the contents (data) onto the intermediate passive tag and the target device reads the content from it. The communication between the two device is performed in multiple cycles depending upon the size of data. One of the most important issues in our implementation is to address the reader's collision problem (RCP), which occurs due to the collision of RF waves emitted by two active NFC devices [14]. To address this, a synchronized control of the NFC modules of both devices is required, such that the intermediate tag is not accessed simultaneously. Other challenges of the proposed methodology, including session maintenance and data transfer delay, is discussed in the following sections (Section IV-A).

To evaluate the concept, we implement a multi-factor authentication (MFA) system using the proposed methodology. MFA is a combination of multiple authentication techniques, where more than one user/device credentials are used for authenticating user and/or device [15]. MFA provides multiple layers of security by validating multiple user credentials. MFA plays an important role for the applications discussed above (payment system, car door opener or smart-lock), as they exchange personal user information between two devices. For these applications, in addition to verifying the authenticity of the key (e.g., PIN or password), we also need to verify the physical presence of the authentic user before providing the access to the lock.

To explain the framework, let us consider a scenario where a user having NFC enabled *smartphone* is trying to access a NFC enabled *smart-lock*, as shown in Fig. 1. We use the fingerprint sensor that is readily available in most modern smartphones as one of the factor for MFA. The proposed mechanism for MFA using NFC works as follows. Firstly, the user's smart-phone is authenticated using a third party authentication tool, i.e., Kerberos [16]. The third party issues a ticket for accessing the smart lock. Upon verification, the fingerprint sensor is used for user authentication. The use of two factor authentication allows the verification of the user's smartphone as well as the presence of a user in the close vicinity of the device or the smart lock under consideration.

We perform experimental evaluation of the system by using two NFC enabled smartphones and passive tags of different sizes. Furthermore, we deploy Kerberos as a third-party authentication tool, for generating tickets and device

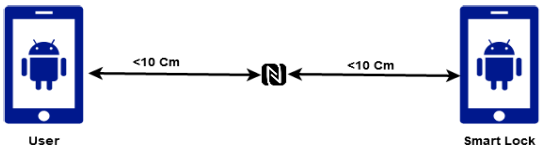


FIGURE 1. Bi-Directional Data Transfer using NFC read/Write Mode.

verification. In our implementation, the smart lock is another NFC enabled smartphone. The experimental results show that it takes about 1 second to connect and obtain ticket from the Kerberos system and 600 *ms* to unlock the door. In addition, we evaluate the resource requirements for implementing the algorithm, including the CPU usages, power consumption, and internal temperature of the smartphone. We also perform security analysis of the proposed MFA scheme against known attacks to understand the robustness of the system.

The rest of the paper is as follows. Section II discuss the previous works in the field of NFC and authentication process using NFC as communication media. Section III presents a study of different modes of NFC. In Section IV, we discuss the methodology of bi-directional communication using NFC read/write mode. Section V discuss the architecture, challenges, implementational details and security analysis of MFA system based on our proposed methodology. The evaluation of proposed methodology is performed in Section VI. Finally, the paper is concluded in Section VII.

II. RELATED WORKS

In the following, we begin our discussion by highlighting the applications of NFC technology in various domains. In particular, we focus on applications that use NFC for user/device authentication and finally discuss this paper's differences from the previous works.

A. NFC AND ITS APPLICATIONS

Near Field Communication (NFC) is a prominent contact-less communication technology, which is rapidly getting popular in modern smart devices. As published in *NFCW*, a leading online news platform on NFC, several governments are currently promoting the usages of NFC for mobile payments and other applications [17]. NFC can be easily integrate on a number of devices like smartphones, and sensors as it draws very little power for operation. The three basic modes of operation in NFC are (a) Read/Write mode, (b) Peer-to-peer mode and (c) Host Card Emulation (HCE) mode. NFC read/write mode is used for one way communication between NFC reader and writer which includes contact-less payment mechanism, pairing of devices, smart posters, sharing of short messages with the user of NFC enabled smart-phones [5]. The peer-to-peer mode is used for Bluetooth or WiFi pairing or to transfer very small amount of data, like beaming the URL to another smart-phone.

Researchers have also proposed the use of peer-to-peer mode for payment services such as IDA-Pay [18], secure credit transfer among smart-phone [19]. Reference [18]

proposes a micro-payment module for mobile to POS (Point Of Sale) transactions. In [19], authors discuss the potential use of NFC in payment mechanisms and propose an application for mobile to mobile credit transfer. A security framework in peer-to-peer mode is proposed in [20], to facilitate the use to the peer-to-peer mode for economic activities. In the mobile banking sector, several application like Swing-Pay, Samsung Pay, Apple Pay use NFC for payment purpose [21]–[23]. In addition, other application domain of NFC includes health-care sector [24]–[26]. Academicians are also interested in using NFC-enabled smartphones to sense bio-chemical, and gases like oxygen, carbon dioxide, and relative humidity [27], [28], semantic enrichment of children facing language disorder [29]. Reference [1] discuss the use cases and potential applications of NFC. NFC is also widely used in health-care sector [30]–[32] and payment and loyalty system [33].

B. MORE REAL LIFE APPLICATIONS OF NFC

Modern real-life applications, such as tracking tuna fish and agricultural products via blockchain [34], [35], uses Internet-of-Things (IoT) and NFC technologies to create virtual representation (digital twins) of the physical objects. These applications require data transmission in close physical proximity to ensure the existence of the physical object and to reduce the risk of unauthorized access. Similarly, information from these objects are retrieved in close proximity to reduce the chances of information leakage. RFID/NFC is a prime candidate for such communication, as the cost of implementing NFC based solution is very low and does not require sophisticated hardware infrastructure. The NFC network can be easily deployed in any terrain, such as in animal husbandry for tracking livestock and meat and establishing 'farm to plate' chain [36], [37].

C. AUTHENTICATION VIA NFC

Researchers have also used NFC for authentication purposes [4], [38]–[41]. Authentication of a user/device is important for maintaining the security and privacy of an application. [4], [38] use NFC only for unidirectional transmission of user credentials and for triggering the subsequent operations of a MFA system. On the other hand, [39]–[41], proposes an authentication protocol for NFC enabled devices for ensuring authenticity, security and privacy. Reference [42] reviews authentication techniques in wireless sensor networks. Researchers has also proposed authentication and access control mechanism in the field of wireless body area network [43], [44], IoT [45], road network [46] and smart home environment [47], [48]. Nowadays, exponential growth in cyber-attacks requires a chain of authentication mechanism, often referred as multi-factor authentication (MFA). In MFA, multiple user credentials are used to verify the legitimacy of a user, which can be the combination of text based password, bio-metrics, one time password (OTP) and others [49]. MFA is widely used in many areas such as smart cities [50], multi-server environment [51], [52], cyber physical systems [53].

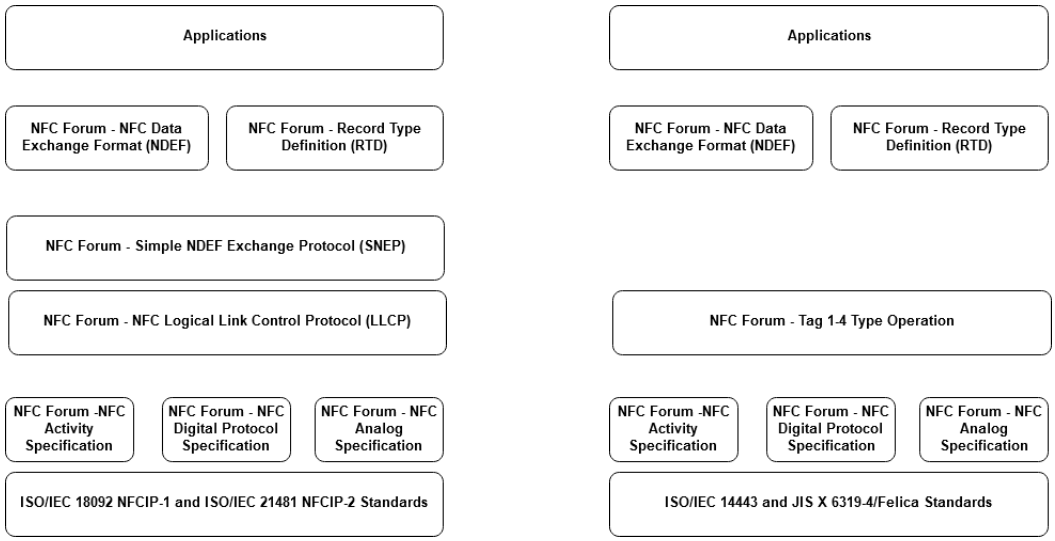
D. SUMMARY

In the literature, researchers have used NFC peer-to-peer mode to propose various authentication protocols, such as pseudonym-based secure authentication [41]. In [39], [40], authors consider similar protocols but do not discuss the mode of communication. Besides peer-to-peer mode, researchers have used NFC read/write mode to perform authentication. However, the mode is mainly used for one-way communication and for transmitting small data between two NFC devices. For example, commercial applications, such as [22], [23] use NFC read/write and HCE (Host Card Emulation) modes for financial transactions. In [56], the NFC read/write mode is applied for user identification and for triggering the second phase of the authentication process using face recognition.

In comparison to the previous works, in this paper, all communications take place via the NFC read/write mode using half-duplex communication in multiple cycles. This is an extension of our previous work [57]. In [57], we explore the possibility of bi-directional communication in the NFC read/write mode by controlling the screen of NFC enabled smartphones. For this, we set the sleep and wake time large enough (i.e., 4 and 2 secs, respectively) to avoid the readers-writers conflict problem. However, this can significantly increase the transmission delay of an application. Thus, in this paper, we try to address the problem by minimizing the sleep and wake time to reduce the overall transmission delay and evaluate its performance on an application that requires secure bi-direction communication.

III. NFC DATA TRANSFER MODES

NFC can operate in three different modes: (a) host-card emulation mode, (b) peer-to-peer mode and (c) read/write mode. The host-card emulation (HCE) mode allow virtualization of smart cards (credit card or identity card), so that a smartphone application can emulate the smart card and communicate directly with the NFC reader for payment or identity verification process. The peer-to-peer mode allows half-duplex communication between two active NFC devices. Whereas, the NFC read/write mode is used for communication between an active and a passive device. Here, an active NFC device means that the device is connected to a power source and can generate the magnetic field for communication. A passive device, on the other hand, do not have their own power source; but gets charged by magnetic field generated by an active device, based on the Faraday's principle of magnetic induction. From the definition, it is clear that the peer-to-peer mode can support bi-directional communication between two active NFC devices, and the read/write mode can only be used for uni-directional communications. To understand the advantages of using read/write mode, we compare it with NFC Peer-To-Peer mode.



(a) Protocol stack for peer-to-peer mode of communication [54] (b) Protocol stack for read/write mode of communication [55]

FIGURE 2. NFC Protocol stack.

Fig. 2, is the protocol stack of NFC peer-to-peer and read/write modes ([54], [55]). As shown in the figure, the protocol stack of NFC peer-to-peer mode is more complex than the read/write mode [54]. In read/write mode [55], the Tag Options for Type 1-4 are defined while in peer-to-peer mode, we have Logical Link Control Protocol (LLCP) and Simple NDEF Exchange Protocol (SNEP), which creates additional overhead in communication. LLCP is use for multiplexed communication between two NFC enabled devices. The next layer is of Simple NDEF Exchange Protocol which is a stateless request/response protocol and uses a client server architecture to exchange NDEF messages between two NFC-enabled devices. On the other hand, the read/write mode uses a single protocol for reading and writing data from the tags. Apart from protocol overhead, NFC peer-to-peer mode has the following limitations.

1) The peer-to-peer mode is only supported by proprietary software, such as Android Beam or S-Beam in Samsung smartphones. Thus, it requires the use of proprietary libraries, which is not open for cross platform applications [6].
2) The files or data to be shared in this mode, must reside in an external storage only in world-readable format [7]. Note that, storing data in the world readable format in an external storage of a smartphone can have major security implications, as it can be accessed by any other application and thereby compromising the privacy and security of the user.

IV. BI-DIRECTIONAL COMMUNICATION IN NFC READ/WRITE MODE

The goal of this paper is to design a methodology for bi-directional communication using NFC read/write mode. In this section, we first briefly present the challenges of performing bi-directional communication using NFC read/write mode. Next, we present the proposed methodology, analyze the delay for completing a transaction between two NFC devices, and finally discuss a methodology for minimizing the same.

A. CHALLENGES OF NFC READ/WRITE MODE FOR BI-DIRECTIONAL COMMUNICATION

As discussed in previous section, the NFC read/write mode do not support bi-directional communication. A possible solution for implementing the same can be to modify the protocol stack of the read/write mode. However, such modification means changing the fundamentals of read/write mode and modifying it to act as a peer-to-peer mode. Additionally, it can have other complications, such as OS dependencies and software up-gradation. Thus, one of the main challenges is to implement bi-directional communication, without changing the internals of the protocol stack. Other challenges in the proposed methodology are as follows:

- To avoid the reader collision problem, which occurs due to the collision of RF waves when a passive NFC device is placed in the interrogation zone of two active NFC devices [14].
- NFC tag are available in different sizes, such as 144 *bytes* to 4096 *bytes*. So, the methodology must work for all tags, i.e., the amount of data transferred must be independent of tag size.
- The communication must complete in an acceptable time frame.

In the following section, we try to address these challenges for the NFC read/write mode. The algorithm presented below can be applicable for most RFID based solutions that required bi-directional communication between two devices.

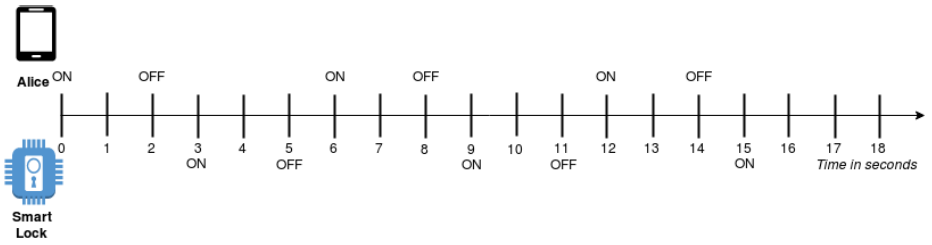
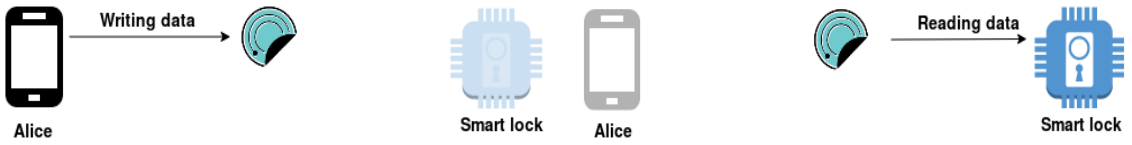


FIGURE 3. Timing diagram for NFC data transfer scheme.



(a) Alice writing data onto the tag

(b) Smart lock reading data from the tag

FIGURE 4. Half-Duplex communication in NFC Read/Write Mode.

B. BI-DIRECTIONAL DATA TRANSFER SCHEME

NFC works on the principle of magnetic induction. When a device wants to read/write from a tag, it first charges the tag by producing magnetic field. Problem occurs when the magnetic fields of two active NFC devices collide with each other (RCP problem). This can cause unpredictable errors or behavior in the devices. Such collisions frequently occur in applications that require bi-directional communication via RFID technology [58], [59]. To avoid this, we need to synchronize the NFC modules of the two active devices. Unfortunately, the Android operating system allow very limited access to the NFC module, and it can be only controlled manually through the setting menu. Note that, this restriction is enforced on NFC for security reasons and is not true for other RFID solutions. RFID is used for designing special purpose devices (e.g. for tracking and billing systems), and provides APIs to control the module according to the requirements of an application [60].

From our experiments with NFC-enabled smartphones, we find that the NFC module stops functioning if the device is locked or the screen is switched off (for saving battery). That is, although there are no APIs available to programmatically start/stop NFC, but we can control it via the screen-lock and power-management APIs [57]. We exploit this observation to devise a method for transferring data between two NFC enabled devices via an intermediate tag in multiple cycles. In the following, we use Fig. 3 to briefly discuss the procedure of half-duplex communication between two NFC enabled in read/write mode.

In Fig. 3, we assume that a user Alice is using her NFC enabled smartphone to transfer data to another NFC enabled device (smart-lock). As shown in the figure, we consider two different time intervals, i.e., sleep time and wake time. **Sleep time** is the duration for which the NFC module remains inactive (i.e., by locking screen) and **wake time** is the duration of an active NFC (by disabling screen-lock). In Fig. 3, at $t = 0$, Alice's smartphone wakes up and writes onto the passive tag (as shown in Fig. 4). The wake time for both smartphone and the smart lock is set to 2 seconds. Alice's smartphone sleeps at $t = 2$. The smart lock wakes up at $t = 3$ to reading the content of the tag, update it and turns off the NFC at $t = 5$. The sleep time for both devices is fixed to 4 seconds. So at $t = 6$, Alice wakes up again and read the contents and thereby completing one communication cycle. In the above discussion, we fix the sleep and wake time for simplicity, and it is significantly more than required for performing the read/write operation from a tag. In the following section, we discuss a procedure for selecting these time intervals.

C. DELAY ANALYSIS

In the following, we evaluate the overall communication delay of the proposed system. Equation (1) shows the NFC read/write delay ($T_{NFC'}$), that is the delay required for exchanging data via an intermediate NFC tag in multiple cycles. In the equation, T_{R_j} and T_{W_k} denote the delay required for reading and writing data from the NFC tag at any instance i in multiple cycles, respectively.

$$T_{NFC'} = \sum_{i} \left(\sum_{j} T_{R_j} + \sum_{k} T_{W_k} \right) \tag{1}$$

The above equation only considers the read and write operations, required for bi-directional communication between the two devices. However, in real-life applications, we need to consider the delay for other operations, as well. This can include the processing delays (T_i) of an application on the shared data. In in (2), T_ω aggregates these additional delays incurred by the both devices.

$$T_\omega = \sum_{l} T_l = T_1 + T_2 + \ldots.. + T_n \tag{2}$$

Thus, the overall communication delay can be written as (3). In the equation, T_{NFC} depends on the message size that needs to be transferred between two NFC devices and the available memory of the intermediate tag. The other two factors are depend on the network condition and processing capability of the smartphones.

$$T = T_{NFC} + T_{\omega} \tag{3}$$

Notice that the read/write operation can be performed, only when the NFC is in wake state. In this state, the device first reads from the tag, processes the data, writes response on the tag and finally, moves to the sleep mode. However, the sleep and wake time of two active NFC devices (say, A and B) at any i^{th} instance is related to each other and can be expressed using 4.

$$\begin{aligned} T^{A}_{wake_i} &= T^{B}_{sleep_i} = T_{NFC_i} + T_{\omega_i} \\ &= \sum_{j} T_{R_j} + \sum_{k} T_{W_k} + \sum_{l} T_l \end{aligned} \tag{4}$$

Thus, the total communication delay (T) can be calculated by aggregating the wake cycles of the two devices. However, as the sleep time of one device is equal to wake time of other, we can formulate 5 to compute the total delay of bi-directional communication. (5) shows that the communication delay primarily depends on the T_{sleep_i} and T_{wake_i} cycles and we can improve the performance of the system by minimizing the sleep and wake time of the two NFC devices. Notice that the tag size plays an important role in determining the number of cycles for bi-directional communication. Future innovations in NFC technology can increase the size of NFC tags, and can make the proposed concept more acceptable for close proximity applications.

$$T = \sum_{i} (T_{wake_i} + T_{sleep_i}) \tag{5}$$

As discussed in (5), the communication delay primarily depends on the T_{sleep_i} and T_{wake_i} cycles. Notice that we can improve the performance of the system, by minimizing the sleep and wake time of the two competing NFC devices. Furthermore, notice that the memory available in the NFC tag, also plays an important role in determining the number of read/write cycles required for transferring data between a smartphone and smart lock. Future innovations in NFC technology can increase the memory of NFC tags and reduce the read/write time, making the proposed concept more acceptable for close proximity communication applications.

As discussed in (5), the communication delay primarily depends on the T_{sleep_i} and T_{wake_i} cycles. Notice that we can improve the performance of the system by minimizing the sleep and wake time of the two competing NFC devices. Furthermore, notice that the memory available in the NFC tag, also plays an important role in determining the number of read/write cycles required for transferring data between a smartphone and smart lock. Future innovations in NFC technology can increase the memory of NFC tags and reduce the read/write time, making the proposed concept more acceptable for close proximity communication applications.

D. MINIMIZING TOTAL COMMUNICATION DELAY

Delay minimization is required to complete a transaction in an acceptable time frame. To address this, we first consider the minimization of a single sleep/wake cycle and discuss the synchronization of multiple cycles to complete di-directional communication between two NFC devices.

1) MINIMIZING SLEEP/WAKE CYCLE

In the previous section, we show that the overall delay for completing a bi-directional communication depends on the sleep and wake time of the NFC devices. Thus, the minimization of these two factors is important for improving its performance. From experiments (Section VI-A), we find that the read/write delay primarily depends on type of the NFC tags and version of the operating system (software) present in the NFC reader. However, due to small capacity of 144 - 868 bytes and transmission rate of 106 - 424 kbits/s, the commercially available NFC tags (type 1, 2, and 3) can perform the read/write operations on a tag in *milliseconds* (i.e., 4 - 6 ms for read and 60 - 100 ms for write operations). In addition to read/write delay, other dynamic factors like positioning of devices, communication distance, and surrounding interference, also affect the overall delay. These factors can differ for each read/write cycle and from one NFC device to another.

To overcome the above challenges in our implementation, we pre-compute the average delay of read/write operations for different NFC tags. Notice that, since the read/write operations are in milliseconds, we can incorporate the above dynamics (i.e., tag size, software/hardware dependencies) and can complete multiple cycles of bi-directional communication without any perceptible difference in the performance of an application. (6) can be used to determine the sleep and wake duration of the NFC devices. In (6), D_i is the average time required for a read/write operation on a particular tag, and δ is an additional delay (in milliseconds) to incorporate the dynamic factors and to avoid RF-wave collision problem (RCP problem). In our implementation, as the variations in delay for the read and write operation on different tag types are not very significant (in msec), we fix the sleep/wake delays for the active NFC devices (as discussed below).

$$D = max(D_1, D_2, \ldots D_n) + \delta \tag{6}$$

2) SYNCHRONIZING THE SLEEP AND WAKE CYCLES

In the literature, researchers have proposed different ways for synchronizing the read and write cycles of RFID devices. Such as, in [61] authors perform synchronization of densely populated RFID readers (e.g., in warehouse) via a centralized command center. Similarly, [62] uses a rotating disk with LEDs to determine the order for performing the read/write operations on multiple NFC tags and readers/writers. Such strategies can be used in our implementation for

Algorithm 1 Bi-Directional Data Transfer Using NFC R/W Mode

```
Function InitTransfer()
    if (Collision Detected) then
        UsrNFC.State = TarNFC.State = OFF ;
        DataTransfer(UsrNFC,Data);
        DataTransfer(TarNFC,NULL);
Function DataTransfer(Dev, Data)
    if (Dev == TarNFC) then
        Dev.Sleep(OffTime) ;     // TarNFC sleep
    while (EndOf(Data)) do
        Dev.State = ON ;        // NFC module ON
        Dev.Read() ;             // Read NFC tag
        Dev.Write() ;           // Write NFC tag
        Dev.Wait(OnTime) ;          // Dev wake
        Dev.State = OFF ;     // NFC module off
        Dev.Sleep(OffTime) ;        // Dev sleep
```

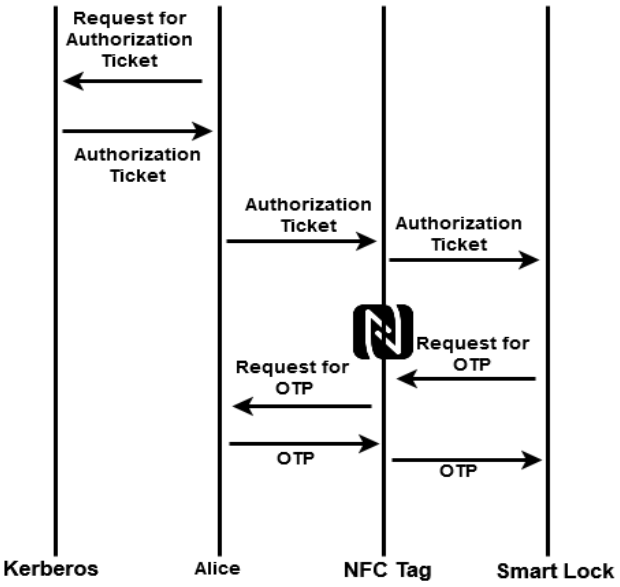


FIGURE 5. MFA via NFC technology.

synchronizing the order of bi-directional communication for the NFC devices. However, notice that communication with external servers can be insecure [63] and can increase the complications of sensitive NFC applications (e.g., financial transactions). Thus, our goal is to synchronize the communicating NFC devices without having direct or indirect communication between the two.

To address this, we assume that the NFC module of one of the communicating device (or a target device), such as a payment terminal or smart-look, is continuously scanning the passive NFC tag. When Alice brings her smartphone (initiator NFC device) near the target, the RF fields of the NFC devices get modulated due to collision, and both devices can know the presence of each other. Immediately after the collision, the devices turn-off their screen to deactivate the NFC module and the NFC module of the target device remains off for next 100 ms. However, the smartphone immediately activates its NFC by switching on the screen and writes data onto the tag and moves to sleep mode. Finally, the smart-lock switch on its NFC after 100 ms and reads the data from the tag. The communication between the two parties can continue in a cyclic manner. Algorithm 1, is a generalized procedure for transferring data between two NFC devices using the read/write mode, where the *OffTime/OnTime* correspond to the sleep/wake durations, respectively.

V. MULTI-FACTOR AUTHENTICATION USING NFC READ/WRITE MODE

In the following discussion, we consider a scenario where Alice again with her smartphone trying to open a smart-lock (as target), where both are active NFC devices. The proposed model uses an intermediate tag between two active NFC devices for transferring data bi-directionally in multiple cycles (Fig. 5). We begin the our discussion with brief discussion on architecture of MFA using NFA followed by implementation details of the system and end our discussion with a brief security analysis of the system.

A. SYSTEM ARCHITECTURE

Fig. 6 depicts a scenario of of our proposed MFA system. Note that, the term ''smart lock'' is only for representational purpose only. The proposed methodology can be applied to any device for verification of user and his/her smartphone. In Fig. 6, we assume that Alice is the user, and wants to interact with the smart electronic device (smart lock). In our implementation, we assume that both smartphone and the smart lock is registered with a third-party authenticator, in our case, Kerberos. Although, any third party authenticator such as SAML [64], WebAuth [65] can be used for this. Alice uses her smartphone to connect to the Kerberos server to receive an authentication ticket and write it to the intermediate passive NFC tag. The intermediate NFC tag, in Fig. 6, facilitates the data transfer between Alice's smartphone and the reader device embedded in the smart lock. The NFC reader in the smart lock reads the content and acknowledges back in same fashion and thereby completing the first phase of authentication.

In the next step, Alice needs to verify her fingerprint using the fingerprint sensor in her smartphone. For security reasons, the fingerprint module in smartphones do not allow the fingerprint to be shared between applications or store in the internal storage of the devices. The scanned fingerprint can only be compared with the stored fingerprint. The fingerprint is stored in a secure area and cannot be accessed for comparison by any application without user permission. In our implementation, when Alice's fingerprint is verified by her smartphone, the smartphone generates

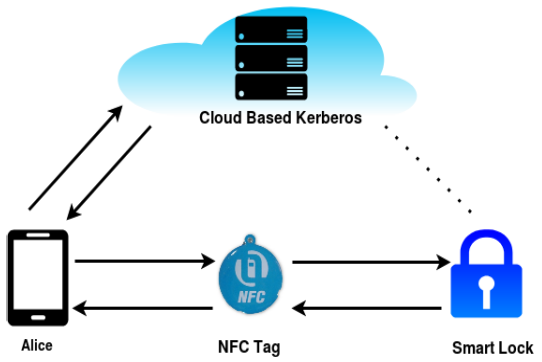


FIGURE 6. Proposed smart lock architecture.

an one-time password (OTP) using time-based one-time password (TOTP) algorithm [66]. Alice then transfers the generated OTP to the smart lock in the final phase of authentication. The smart lock generates and compares the OTP by using the same parameters to the TOTP algorithm. On successful verification, the smart lock gets unlocked.

The challenges in implementing a MFA system using NFC read/write mode are as follows.

(a) We need
o NFC devices. Notice, that there are many available session maintenance protocols, such as [67]–[69]. However, these protocols may not be suitable for our purposes, as the NFC read/write mode has limitations of the tag size. So, we need to device a lightweight session maintenance protocol for the proposed MFA.

(b) The MFA between two NFC enabled devices must be completed in an acceptable time frame.

(c) Since the NFC is generally used for transferring data between two resource constraint devices, the proposed technique must be efficient in terms of computation and battery requirements.

B. IMPLEMENTATION OF PROPOSED MFA

The authentication procedure discussed above consists of two different phases, namely the registration and login phase. In the registration phase, the users/devices need to register with the Kerberos server and share the public key. In the login phase, user tries to gain access of the smart lock by using his/her smartphone. We implement the proposed methodology using NFC enabled smartphones. The implementation is divided into two parts, i.e., 1) Deployment of Kerberos and 2) Secure session maintenance, discussed below. Lastly, we summarize the process of authentication using the NFC read/write mode in algorithmic form.

1) DEPLOYMENT OF KERBEROS

We use a third-party authenticator, namely, Kerberos [16] to mutually authenticate Alice's smartphone and smart lock. Kerberos v5 appeared as RFC 1510 in 1983 and later replaced by RCF 4120 in 2005. In our implementation, we deploy the Kerberos server (K_S) locally in our lab and use a Kerberos

Algorithm 2 MFA Using NFC R/W Mode

```
Function MFAuth()
    UsrNFC.InitTransfer() ;
    UsrNFC.MFA_NFC() ;
Function MFA_NFC()
    authTkt = UsrNFC.getAuthTicket() ;      // Get Ticket
    DataTransfer(UsrNFC, authTkt) ;         // Send Ticket
    if (TarNFC.VerifyTicket() == Success) then
        // Verify ticket and request fingerprint
        DataTransfer(TarNFC, "Request for fingerprint");
    if (UsrNFC.VerifyFingerprint() == Success) then
        // If fingerprint OK, send OTP
        DataTransfer(UsrNFC, UsrNFC.GenerateOTP());
    if (TarNFC.VerifyOTP() == Success) then
        // Check OTP and Return success
        DataTransfer(TarNFC, "Auth Success");
```

Client (K_C) (deployed locally) as an interface between K_S and external users. Note that, all communications are facilitated through the K_C. The reason for using K_C as an interface is to improve the security of the system. As suggested in the Kerberos documentation, K_S should not be placed in the public domain or with a public IP address, as it contains confidential information about users/devices [70]. We assume that the interaction between K_C and K_S is secure and is not susceptible to an internal/external attack. Another advantage of using K_C is that it can filter network traffic to prevent attacks, such as Denial Of Service (DOS). Moreover, K_C can hide the internal security architecture and can be easily replaced by any third party authentication tools, without any hindrance in the services that our system provides to the user.

2) SECURE SESSION MAINTENANCE

As discussed above, we use the Kerberos (third-party authentication tool) in the first phase, to mutually authenticate the smartphone and smart lock. In the second phase, Alice uses the fingerprint scanner in her smartphone to verify her legitimacy. A successful fingerprint verification triggers the TOTP (Time based One Time Password) function to generate the OTP (One Time Password). The parameters passed to the TOTP function are the TGS session key and current time, as shown below. In this, the TGS session key is obtained from the previous step, generated by the ticket generating server (TGS) in Kerberos.

$$OTP = f(TGS\_session\_key, current\_time)$$

We use the time-based OTP function to avoid replay attacks. Typically the time-based OTP is valid only for 2-3 minutes, depending upon the algorithm. Also, Alice's smartphone and smart lock are time-synchronized using Network Time Protocol (NTP), as it is a prerequisite for all devices connected to the Kerberos server. Thus, there is no inconsistency in the OTP generated by smartphone and smart lock, as both devices use the same TGS session key obtained in the first phase of the authentication process and time as input to the TOTP function. A little delay in executing TOTP function has no consequence and results in the same output. To maintains secure session in multiple cycles of data transfer, we append hash of session key and iteration number at the end of every message. In OTP verification, we again use session key to encrypt the OTP, so that, it can be ensured that, the OTP came from legitimate user only. This makes the communication secure between these two entities and also maintain the session.

Algorithm 2 is a procedure for MFA using NFC read/write mode. The authentication begins by invoking the *MFAuth()* function. It calls the *InitTransfer()* function (discussed in Algorithm 1), to initialize the NFC devices for transferring data via the read/write mode. In *MFA_NFC()*, the user sends request to Kerberos for the authentication ticket (one-time process in 24 hours, or as configured), and transfers the ticket using the *DataTransfer()* method (Algorithm 1). Notice that if the ticket size is more than the tag size, then the user device may need multiple cycles to transfer the ticket to the target device. The target device verifies the ticket and on successful verification, it initiates the second phase by requesting fingerprint verification from the user. If the fingerprint is successfully verified by the user's device, it generates an OPT and transfers it to the target using *DataTransfer()*. Successful OTP verification completes the authentication process.

C. SECURITY ANALYSIS

In the following, we analyse the security of the proposed system, in detail. The idea is to explore the vulnerabilities and robustness of the system against popular attacks.

1) MAN-IN-THE-MIDDLE (MITM) ATTACK

The proposed methodology uses two different communication channels. Firstly, communication via traditional network takes place between the smartphone and Kerberos infrastructure for obtaining the authentication ticket. The design principles of Kerberos makes it resilient to common attacks, such as man-in-the-middle (MITM). In addition, we can use any secure transport layer protocol to enhance the security of the system.

The other channel used is the NFC technology, for communication between smartphone and smart-lock. The communication is facilitated using NFC read/write mode. The use of NFC makes is resilient to MITM attacks as it is difficult to intercept the communication without causing any disturbance because the interceptor must be in close physical proximity [71]. Although, MITM attack is difficult in our setup, but it is not impossible. Researchers have studied this problem [72] and proposed cryptographic solutions such as [72] for handling the same. In the proposed system, the session key generated by the Kerberos server is used for encrypting all communications. As a result, it may be difficult for an attacker to get any meaningful information by performing the MITM attack.

2) DENIAL-OF-SERVICE (DoS) ATTACK

The denial-of-service attack can be initiated by using a high power NFC reader/writer for causing disturbance in the communication channel of the two NFC devices. Such attacks can be prevented by using Faraday cage [73], which can block unwanted magnetic waves generated by an adversary.

3) REPLAY OR IMPERSONATION ATTACK

In reply attack, an adversary tries to impersonate a legitimate user by eavesdropping the communication channel to gain access to the system. Due to the use of the NFC technology, such attacks are difficult in the proposed system. However, the problem is more significant for RFID communication. In [73], authors successfully intercepted the RFID communication from a distance of 6.5 meters by eavesdropping, which can be used by an adversary to modify the RFID tag to elicit the Kerberos tickets and perform impersonation attack [74]. In our implementation, the above issue can be addressed by using the security keys of the Kerberos server. Notice that, even if an adversary steals the ticket exchanged between Alice and smart lock, it cannot decrypt its content, as it is encrypted by the private key known to the intended parties. The ticket contains a session key generated by the Kerberos server for facilitating mutual authentication of the devices and for securing future communications between the two parties. Thus, replay/impersonation attack can be difficult in the proposed system.

4) SECURITY OF SMARTPHONE AND K_C/K_S

The security of the proposed system depends on the security of the smartphone and the Kerberos system. The smartphone performs various security operations, such as encryption/decryption, scanning of finger-print, generating OTP, and storing the private key. Thus it can be a potential target for an attacker. However, in this paper, we do not focus on the security of smartphones, which is mainly the responsibility of the operating system and is currently one of the major research areas [75]. In our implementation, we assume that the smartphone performing the above operations is secure. Similarly, in our implementation, we assume that the Kerberos server platform is secure against all internal and external attacks.

5) SECURITY AGAINST SMARTPHONE THEFT

The proposed MFA technique uses smartphone as the key but smartphone theft is nowadays is not uncommon. In case of theft, an unauthorized user can pass the first step of authentication by using a cached authorization ticket from

Kerberos, which generally has a lifetime of 24 hours. But, the second step requires the fingerprint biometric information of an authentic user, which can not be faked.

VI. EXPERIMENTAL RESULTS

In the following section, we discuss the performance of the proposed system. Our goal is to understand factors such as delay incurred by the Kerberos server in generating ticket, network latency and overall delay required for completing the proposed two-factor authentication. Also, we analyse the resource requirements in the smartphones for the implementation. In Table 1, we show the specifications of the tags used in our experiments. We implement the Kerberos locally and use it to generate and retrieve the authentication tickets. We conduct our experiments for a week at 4 different times in a day. The results below are the average of all the data collected from multiple experiments.

TABLE 1. Specfication of NFC tags.

Tag Type	Total Memory (bytes)	Usable Memory (bytes)
Type1	168	144
Type2	512	462
Type3	924	868

A. READ/WRITE DELAY

In the first set of experiments, we try to find the time a smartphone requires for reading and writing on an NFC tag. The idea is to select an NFC tag for implementing the smart lock and also to reduce the overall authorization delay by minimizing the communication time. Initially, we set the data size as 100 bytes and calculate the time spent in transferring of data in a single cycle. The results are shown in Fig. 7 and Fig. 8 are an average of our experiment performed manually for 60 times in an indoor location. Referring to Fig. 7, the smartphone take 4.256 ms in reading data from a Type3 or NXP MIFARE Ultralight C tag with a standard deviation of 0.99 ms and an upper limit of 6 ms. Similarly, Fig. 8 shows that a smartphone takes 68 ms on average to write 100 bytes of data on a Type3 tag.

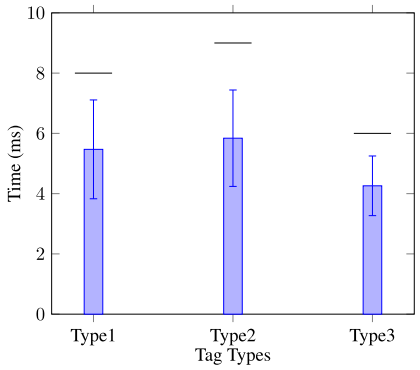


FIGURE 7. Read Time in single cycle.

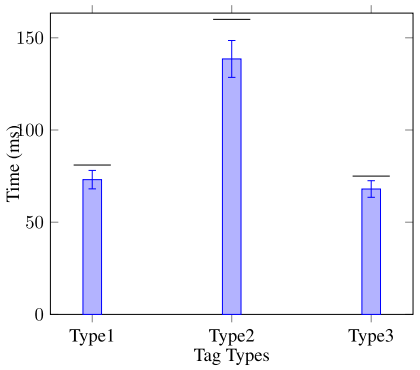


FIGURE 8. Write Time in single cycle.

The next experiment calculates the time spent in reading and writing data transferred in multiple cycles. For this, we transfer data from 500 bytes to 2000 bytes. From the previous experiment, we find that writing takes about 100 ms and reading data takes only 10 ms. The results lead us to set sleep and wake time of M_{INI} to 10 ms and 100 ms and M_{TAR} to 100 ms and 10 ms, respectively. Note that, this time also includes time spent writing or reading data to or from the tag.

As shown in Fig. 9, Type3 tag takes minimum time for transferring data from one NFC enabled device to another. This is due to its fast read and write speed and also large usable memory of 868 bytes, which limits the number of cycles to transfer data. For transferring 2000 bytes of data, Type3 tag needs only three cycles while Type1 takes 14 cycles.

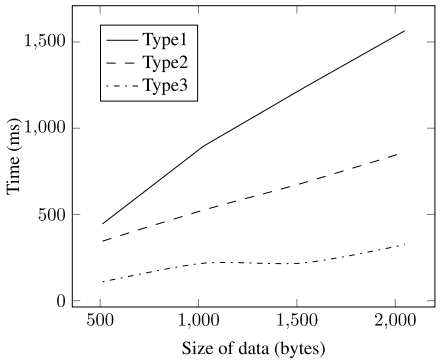


FIGURE 9. Data transfer time in multiple cycles.

B. SUCCESS RATE

In this experiment, we try to find the success rate for both reads and write operations on the tags. The success rate is defined as the number of times the smartphone successfully reads the content from the tag or writes data to it. The results are shown in Fig. 10, are the average of our experiment performed manually for 60 times. The goal is to choose the tag with the highest success rate so that our smart lock works smoothly. The success rate is little higher in an outdoor location (97%) as compared to indoor location (94%) in

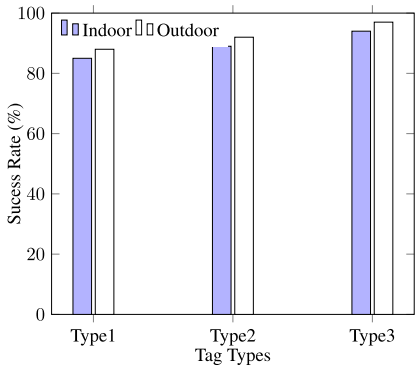


FIGURE 10. Success Rate.

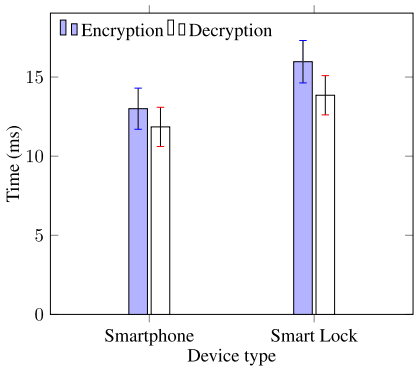


FIGURE 11. Encryption and Decryption time.

case of Type3 tag. The reason for the low success rate in indoor location is due to the presence of various radio and electromagnetic waves generated by smartphone, computer systems and other electrical devices, causing interference to the NFC communication.

C. DELAY OF AUTHENTICATION PROCESS

In the following experiments, we try to find the delay incurred for fetching ticket from Kerberos server and the time required for completion of authentication process of our proposed MFA mechanism. The average time spent in acquiring the authentication ticket from K_S via K_C is 1.001 seconds with a mean deviation of only 0.048 seconds, which also includes network delay. For our experiment, we use the university WiFi network with bandwidth 100 Mbps.

After acquiring the ticket from K_S, the next step is to prepare authenticator ticket and transfer it to the smart lock via NFC tag. For this purpose, the session key is used, which is present in the ticket obtained from the K_S. The smartphone first decrypts the ticket in order to get the session key and prepare the authenticator ticket by encrypting it with the acquired session key. The time spent for encryption and decryption of ticket by the smartphone and the smart lock is given in Fig. 11. The smartphone on an average takes 16 ms to encrypt and 15.85 ms to decrypt the ticket. Smart lock also performs encryption/decryption and takes about 15.97 and 13.85 ms on an average.

For the second phase of authentication, the smart lock requires a TOTP which is a function of time and the session key obtained from the previous step. The smartphone use the fingerprint scanner that triggers the generation of OTP, and at the same time, smart lock also generates the OPT, using same session key. The fingerprint scanner takes about 0.2 seconds to verify Alice's fingerprint and triggers the OTP function. The time spent for generation of OTP by smart -phone and the smart lock is 2.71 and 3.50 ms respectively.

Finally, the proposed system works as follows. First, Alice obtains an authorization ticket from the Kerberos server (Section V-B1). After obtaining ticket and preparing authorization ticket for the smart lock, data transfer takes place between Alice's smartphone and smart lock. Initially, both smartphone and smart lock NFC is switched on, and upon collision, both switch off their NFC. In Fig. 12, this duration is indicated as the Collision Zone. After the initial collision, both device switch off their NFC and the smartphone switches it ON again at time $t = 0$. It takes about 100 ms to write the ticket and authenticator message on the tag and switches OFF it's NFC. At $t = 100$, smart lock switches ON its NFC and read the content of tag which takes about 10 ms and decryption takes around another 30 ms. At $t = 140$, smart lock starts writing request for OTP as an acknowledgement of the first step of authentication. Smartphone switches ON it's NFC at $t = 240$, read the request, use fingerprint scanner and generate OTP and write it onto the tag. this whole exercise takes about 360 ms. The smart lock then read the OTP at $t = 600$ and upon successful verification, unlock the smart lock. This whole process takes about 600 ms to unlock the smart lock.

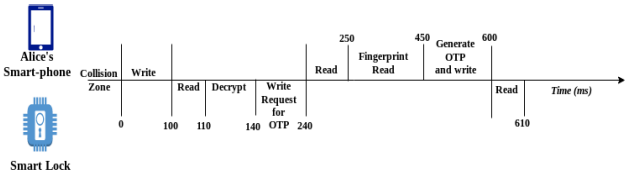


FIGURE 12. Timing diagram for data transfer using passive NFC tag between user and smart lock.

D. RESOURCE ANALYSIS

In this section, we discuss the resource consumption like battery and CPU usage for the proposed implementation. First, we consider the battery consumption of the smart lock as the smart look keeps the NFC on for a prolonged period. We keep NFC module of smart lock opened for about 6 hours. Fig. 13 shows the battery consumption by various application including NFC service. The figure shows that NFC services took only 1.42% of the total battery. The typical current consumption of NFC reader device is less than 100mA. So for 10 hours continuous operation with 80% battery efficiency of a NFC reader device, we require $100mA * 10h * (1/0.8) = 1250\ mAh$ battery.

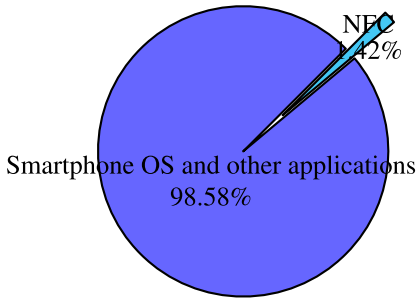


FIGURE 13. Battery Consumption of smart lock.

We also analyse the battery and CPU temperature of the smart lock to check if there is any heating effect when NFC is switched on for a long period of time. Fig. 14 shows that the battery temperature is nearly constant throughout the experiment but the CPU temperature has variations. This is because smartphone operating system runs various background services, which changes the CPU temperature. The effect of NFC reading or writing does not have much effect on the temperature.

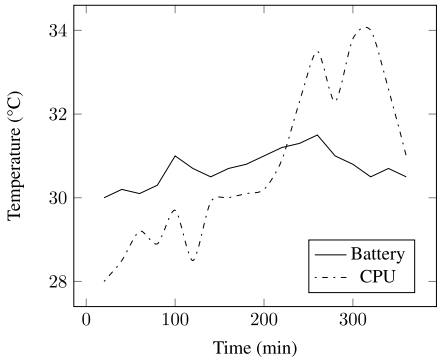


FIGURE 14. Battery and CPU temperature.

Fig. 15 shows that the CPU utilization shoots to about 20-25% whenever NFC is used. Whenever there is a collision between RF ways of both the devices involved, we experience a spike in CPU utilization The decryption algorithm and OTP generation also leads to these spikes in the graph.

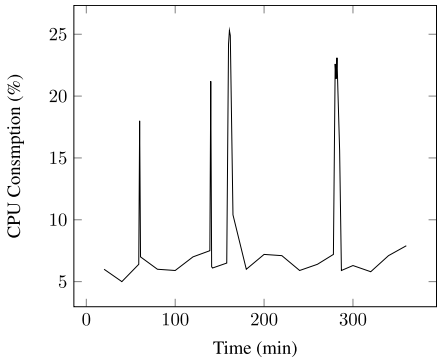


FIGURE 15. CPU consumption.

VII. CONCLUSION AND FUTURE WORK

In this paper, we investigate the use of NFC read/write mode as a bi-directional communication channel in close physical proximity. For demonstration, we implement a multi-factor authentication system for accessing a smart-lock via NFC enabled smartphone. The proposed system requires two-factor authentication, for which the methodology uses a third-party authenticator (Kerberos) and user's fingerprint. The paper explores the advantages of using NFC read/write mode and address the challenges of implementing the MFA. This includes, bi-directional communication between two active NFC devices, session maintenance, and completing the process in an acceptable time-frame with low energy consumption. The proposed methodology can be extended to any RFID system, as most RFID system supports the read/write mode as its basic functionality. We evaluate the security by analysing the different possible attacks on the NFC system. Finally, our experimental evaluation shows that the MFA takes about one second to acquire authentication ticket from Kerberos and about 600 ms to unlock the smart-lock. This is huge improvement from our previous work on transferring data using NFC Read/Write mode. In that work, the time complexity was high (experimental setup discussed in Section IV-B) and the communication was unsecured, resulting in non-applicability of the work. Our future work includes the enhancement of user experience by minimizing the sleep and wake time of the smart-phones to minimize the overall time. We are also working on the problem of accessing a single NFC tag via multiple NFC devices.

REFERENCES

[1] V. Coskun, B. Ozdenizci, and K. Ok, "A survey on near field communication (NFC) technology," *Wireless Pers. Commun.*, vol. 71, no. 3, pp. 2259–2294, 2013.

[2] X. Chen, K. Choi, and K. Chae, "A secure and efficient key authentication using bilinear pairing for NFC mobile payment service," *Wireless Pers. Commun.*, vol. 97, no. 1, pp. 1–17, Nov. 2017.

[3] A.-M. Lesas and S. Miranda, "NFC use cases," in *Proc. Art Sci. NFC Program.*, 2017, pp. 107–120.

[4] G. K. Verma and P. Tripathi, "A digital security system with door lock system using RFID technology," *Int. J. Comput. Appl.*, vol. 5, no. 11, pp. 6–8, Aug. 2010.

[5] InfoWorld. *6 Cool Uses of Near-Field Communication*. Accessed: Dec. 10, 2018. https://www.infoworld.com/article/2607039/mobile-technology/mobile-technology-6-cool-uses-of-near-field-communication.html

[6] Android Developers. *Near Field Communication Overview*. Accessed: Oct. 2, 2019. [Online]. Available: https://developer.android.com/guide/topics/connectivity/nfc/

[7] Mostly-Tech. (2012). *What Samsung & Google Don't Tell You About Beaming*. Accessed: Oct. 19, 2018. [Online]. Available: https://mostly-tech.com/2012/10/08/what-samsung-google-dont-tell-you-about-beaming/

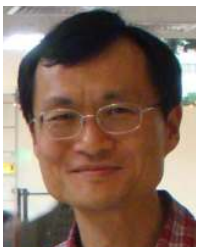
[8] Lowry Solutions. *What Are the Different Types of RFID Technology*. Accessed: Jan. 16, 2020. [Online]. Available: https://lowrysolutions.com/blog/what-are-the-different-types-of-rfid-technology/

[9] J. Potts and S. Sukittanon, "Exploiting Bluetooth on Android mobile devices for home security application," in *Proc. Southeastcon*, Dec. 2012, pp. 1–4.

[10] A. Kassem, S. E. Murr, G. Jamous, E. Saad, and M. Geagea, "A smart lock system using Wi-Fi security," in *Proc. 3rd Int. Conf. Adv. Comput. Tools for Eng. Appl. (ACTEA)*, Jul. 2016, pp. 222–225.
[11] M. Pulipati and S. Phani, "Comparison of various short range wireless communication technologies with nfc," *Interface J. Sci. Res*, vol. 2, pp. 87–91, 2013.
[12] M. Roland, *Basics*. Cham, Switzerland: Springer, 2015, pp. 13–31.
[13] K. Shabana, N. Fida, F. Khan, S. R. Jan, and M. U. Rehman, "Security issues and attacks in wireless sensor networks," *Int. J. Adv. Res. Comput. Sci. Electron. Eng.*, vol. 5, no. 7, pp. pp–81, 2016.
[14] W. Yoon and N. H. Vaidya, "RFID reader collision problem: Performance analysis and medium access," *Wireless Commun. Mobile Comput.*, vol. 12, no. 5, pp. 420–430, Apr. 2012.
[15] C. Mears. (2016). *Why Organizations Need Adaptive Multi-factor Authentication (MFA)*. Accessed: Nov. 20, 2018. [Online]. Available: https://blog.centrify.com/adaptive-multi-factor-authentication-mfa
[16] MIT Univeristy. (2017). *Kerberos: The Network Authentication Protocol*. Accessed: Nov. 10, 2018. [Online]. Available: https://web.mit.edu/kerberos/
[17] SJB Research. *NFCW*. Accessed: Oct. 4, 2019. [Online]. Available: https://www.nfcw.com//
[18] L. Mainetti, L. Patrono, and R. Vergallo, "IDA-Pay: A secure and efficient micro-payment system based on peer-to-peer NFC technology for Android mobile devices," *J. Commun. Softw. Syst.*, vol. 8, no. 4, pp. 117–125, 2012.
[19] D. M. Monteiro, J. J. P. C. Rodrigues, and J. Lloret, "A secure NFC application for credit transfer among mobile phones," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, May 2012, pp. 1–5.
[20] P. Urien, "LLCPS: A new security framework based on TLS for NFC P2P applications in the Internet of Things," in *Proc. IEEE 10th Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2013, pp. 845–846.
[21] S. Ghosh, A. Majumder, J. Goswami, A. Kumar, S. P. Mohanty, and B. K. Bhattacharyya, "Swing-pay: One card meets all user payment and identity needs: A digital card module using NFC and biometric authentication for peer-to-peer payment," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 82–93, Jan. 2017.
[22] Samsung. *Samsung Pay*. Accessed: Dec. 1, 2018. [Online]. Available: http://www.samsung.com/in/samsung-pay/
[23] Apple. *Pay: Cashless Made Effortless*. Accessed: Dec. 1, 2018. [Online]. Available: https://www.apple.com/apple-pay/
[24] J.-S. Sheu, H.-N. Shou, F.-M. Luo, and G.-C. Zeng, "The realization of healthcare combined with Bluetooth and nfc technology," *Eng. Technol. Innov.*, vol. 4, pp. 22–24, Dec. 2016.
[25] J. Miranda, J. Cabral, S. Wagner, C. Fischer Pedersen, B. Ravelo, M. Memon, and M. Mathiesen, "An open platform for seamless sensor support in healthcare for the Internet of Things," *Sensors*, vol. 16, no. 12, p. 2089, Dec. 2016.
[26] G. Alex, B. Varghese, J. G. Jose, and A. Abraham, "A modern health care system using IoT and android.," *Int. J. Comput. Sci. Eng.*, vol. 8, no. 4, pp. 117–121, 2016.
[27] P. Escobedo, M. M. Erenas, N. López-Ruiz, M. A. Carvajal, S. Gonzalez-Chocano, J. de Orbe-Payá, L. F. Capitán-Valley, A. J. Palma, and A. Martínez-Olmos, "Flexible passive near field communication tag for multigas sensing," *Anal. Chem.*, vol. 89, no. 3, pp. 1697–1703, Feb. 2017.
[28] G. Xu, Q. Zhang, Y. Lu, L. Liu, D. Ji, S. Li, and Q. Liu, "Passive and wireless near field communication tag sensors for biochemical sensing with smartphone," *Sens. Actuators B, Chem.*, vol. 246, pp. 748–755, Jul. 2017.
[29] M. Luisa Lorusso, E. Biffi, M. Molteni, and G. Reni, "Exploring the learnability and usability of a near field communication-based application for semantic enrichment in children with language disorders," *Assistive Technol.*, pp. 1–12, 2017.
[30] R. Shaw and K. Govinda, "Automation of patient information in healthcare system," in *ICT Based Innovations*, A. K. Saini, A. K. Nayak, and R. K. Vyas, Eds. Singapore: Springer, 2018, pp. 93–103.
[31] C. Kloch, T. Jørgensen, and N. Boye, "User driven Innovation—Involving the users of the global information multimedia communication village in the creation of a device for personal healthcare: Maxi," *Wireless Pers. Commun.*, vol. 49, no. 3, pp. 431–443, May 2009.
[32] N. N. Khah Razmi and A. B. Sangar, "The use of NFC technology to record medical information in order to improve the quality of medical and treatment services," *Modern Appl. Sci.*, vol. 10, no. 6, p. 136, Apr. 2016.
[33] B. Ozdenizci, K. Ok, and V. Coskun, "NFC loyal for enhancing loyalty services through near field communication," *Wireless Pers. Commun.*, vol. 68, no. 4, pp. 1923–1942, Feb. 2013.
[34] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in agri-food supply chain management: A practical implementation," in *Proc. IoT Vertical Topical Summit Agricult.*, May 2018, pp. 1–4.
[35] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.
[36] STARNFC. *RFID Application in Animal Husbandry*. Accessed: Jan. 3, 2020. [Online]. Available: https://www.starnfc.com/property-fairness-launch-going-to-advantage-everyone
[37] D. Pigini and M. Conti, "NFC-based traceability in the food chain," *Sustainability*, vol. 9, no. 10, p. 1910, Oct. 2017.
[38] S. M. Nasution, E. M. Husni, and A. I. Wuryandari, "Prototype of train ticketing application using near field communication (NFC) technology on Android device," in *Proc. Int. Conf. Syst. Eng. Technol. (ICSET)*, Sep. 2012, pp. 1–6.
[39] H. Eun, H. Lee, and H. Oh, "Conditional privacy preserving security protocol for NFC applications," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 153–160, Feb. 2013.
[40] V. Odelu, A. K. Das, and A. Goswami, "SEAP: Secure and efficient authentication protocol for NFC applications using pseudonyms," *IEEE Trans. Consum. Electron.*, vol. 62, no. 1, pp. 30–38, Feb. 2016.
[41] J. Xu, K. Xue, Q. Yang, and P. Hong, "PSAP: Pseudonym-based secure authentication protocol for NFC applications," *IEEE Trans. Consum. Electron.*, vol. 64, no. 1, pp. 83–91, Feb. 2018.
[42] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Netw.*, vol. 27, pp. 159–194, Apr. 2015.
[43] D.-E.-S. Agha, F. H. Khan, R. Shams, H. H. Rizvi, and F. Qazi, "A secure crypto base authentication and communication suite in wireless body area network (WBAN) for IoT applications," *Wireless Pers. Commun.*, vol. 103, no. 4, pp. 2877–2890, Dec. 2018.
[44] L. Ma, Y. Ge, and Y. Zhu, "TinyZKP: A lightweight authentication scheme based on zero-knowledge proof for wireless body area networks," *Wireless Pers. Commun.*, vol. 77, no. 2, pp. 1077–1090, Jul. 2014.
[45] Y. Yang, H. Cai, Z. Wei, H. Lu, and K.-K. R. Choo, "Towards lightweight anonymous entity authentication for IoT applications," in *Proc. Australas. Conf. Inf. Secur. Privacy*, 2016, pp. 265–280.
[46] I. Memon, Q. A. Arain, H. Memon, and F. A. Mangi, "Efficient user based authentication protocol for location based services discovery over road networks," *Wireless Pers. Commun.*, vol. 95, no. 4, pp. 3713–3732, Aug. 2017.
[47] P. Gope, "Anonymous mutual authentication with location privacy support for secure communication in M2M home network services," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 1, pp. 153–161, Jan. 2019.
[48] C.-J. Chae and H.-J. Cho, "Enhanced secure device authentication algorithm in P2P-based smart farm system," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 6, pp. 1230–1239, Nov. 2018.
[49] I. Velsquez, A. Caro, and A. Rodrguez, "Authentication schemes and methods," *Inf. Softw. Technol.*, vol. 94, pp. 30–37, Feb. 2018.
[50] X. Li, J. Niu, S. Kumari, F. Wu, and K.-K.-R. Choo, "A robust biometrics based three-factor authentication scheme for global mobility networks in smart city," *Future Gener. Comput. Syst.*, vol. 83, pp. 607–618, Jun. 2018.
[51] R. S. Pippal, C. D. Jaidhar, and S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Pers. Commun.*, vol. 72, no. 1, pp. 729–745, Sep. 2013.
[52] Neha and K. Chatterjee, "An efficient biometric based remote user authentication technique for multi-server environment," *Wireless Pers. Commun.*, vol. 97, no. 3, pp. 4729–4745, Dec. 2017.
[53] S. Ibrokhimov, K. L. Hui, A. Abdulhakim Al-Absi, H. J. lee, and M. Sain, "Multi-factor authentication in cyber physical system: A state of art survey," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 279–284.
[54] Texas Instruments. *NFC Activeand Passivepeer-to-Peer Communication Using the TRF7970A*. . Accessed: Oct. 2, 2019. [Online]. Available: http://www.ti.com/lit/an/sloa192b/sloa192b.pdf/
[55] Texas Instruments. *NFC/HFRFIDreader/Writerusingthe TRF7970A*. Accessed: Oct. 2, 2019. [Online]. Available: http://www.ti.com/lit/an/sloa227b/sloa227b.pdf/
[56] D. B. Mei Yin, M. I. Kamal, N. S. Azmanuddin, S. H. S. Ali, A. T. Othman, and R. Z. W. Chik, "Electronic door access control using MyAccess two-factor authentication scheme featuring near-field communication and eigenface-based face recognition using principal component analysis," in *Proc. 10th Int. Conf. Ubiquitous Inf. Manage. Commun.*, Jan. 2016, p. 1.

[57] A. Banerjee, M. S. Chishti, and G. Kumar, "On exploring NFC for half-duplex communication in read/write mode," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWNeT)*, May 2017, pp. 1–8.
[58] C. Lee, H. Cho, and S. W. Kim, "An adaptive RFID anti-collision algorithm based on dynamic framed ALOHA," *IEICE Trans. Commun.*, vols. E91–B, no. 2, pp. 641–645, Feb. 2008.
[59] Z. Xuan and Y. Gang, "Research and simulate of the optimization anti-collision technology in UHF RFID system," in *Proc. Int. Conf. Electric Inf. Control Eng.*, Apr. 2011, pp. 643–646.
[60] ORACLE. *Programming With the ALE and ALEPC APIs*. Accessed: Apr. 20, 2019. [Online]. Available: https://docs.oracle.com/cd/E13197_01/rfid/edge_server/docs21/prog/read_write_tags.html/
[61] K. S. Leong, L. Ng, A. R. Grasso, and P. H. Cole, "Synchronization of rfid readers for dense rfid reader environments," in *Proc. Int. Symp. Appl. Internet Workshops*, Jan. 2006, pp. 51–54.
[62] M. Abbas. *Device-to-Device Communication via NFC Demo*. Accessed: Mar. 23, 2020. [Online]. Available: https://www.element14.com/community/groups/industrial/blog/2018/08/10/device-to-device-communication-via-nfc-demo/
[63] P. Gope, R. Amin, S. K. Hafizul Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Future Gener. Comput. Syst.*, vol. 83, pp. 629–637, Jun. 2018.
[64] SAML. (2017). *SAML Single Sign-On (SSO)*. Accessed: Nov. 10, 2018. [Online]. Available: https://www.onelogin.com/saml/
[65] Stanford Univeristy. (2017). *Stanford WebAuth*. Accessed: Nov. 10, 2018. [Online]. Available: http://webauth.stanford.edu/
[66] IETF RFC. (2011). *TOTP: Time-Based One-Time Password Algorithm*. Accessed: Dec. 12, 2018. [Online]. Available: https://tools.ietf.org/html/rfc6238
[67] M. S. Farash, "An improved password-based authentication scheme for session initiation protocol using smart cards without verification table," *Int. J. Commun. Syst.*, vol. 30, no. 1, p. e2879, Jan. 2017, doi: 10.1002/dac.2879.
[68] Y. Lu, L. Li, H. Peng, and Y. Yang, "A secure and efficient mutual authentication scheme for session initiation protocol," *Peer Netw. Appl.*, vol. 9, no. 2, pp. 449–459, Mar. 2016.
[69] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card," *Peer Netw. Appl.*, vol. 9, no. 1, pp. 171–192, Jan. 2016.
[70] Kerberos Consortium. *Kerberos: Documentation*. Accessed: Dec. 13, 2018. [Online]. Available: https://kerberos.org/docs/index.html/
[71] E. Haselsteiner and K. Breitfu, "Security in near field communication (NFC)," in *Proc. Workshop RFID Secur.*, 2006, pp. 12–14.
[72] M. Badra and R. B. Badra, "A lightweight security protocol for NFC-based mobile payments," *Procedia Comput. Sci.*, vol. 83, pp. 705–711, 2016.
[73] P.-H. Thevenon, O. Savry, S. Tedjini, and R. Malherbi-Martins, "Attacks on the hf physical layer of contactless and rfid systems," *Current Trends Challenges RFID*, vol. 4, p. 415, Dec. 2011.
[74] S. Guizani, "Relay attacks concerns in wireless ad hoc, sensors, and RFID networks," *Wireless Commun. Mobile Comput.*, vol. 16, no. 11, pp. 1431–1435, Aug. 2016,
[75] D. Goel and A. K. Jain, "Overview of smartphone security: Attack and defense techniques," *Comput. Cyber Security: Princ., Algorithm, Appl., Perspect.*, vol. 4, p. 249, Nov. 2018.

MOHD SAMEEN CHISHTI received the B.Sc. degree in computer application from Aligarh Muslim University, in 2010, and the M.C.A. degree from Jamia Millia Islamia, New Dehi, India, in 2013. He is currently a Research Scholar with the Department of Computer Science, South Asian University, New Delhi. His research interests include the IoT, fog computing, and blockchain.



CHUNG-TA KING (Senior Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1980, and the M.S. and Ph.D. degrees in computer science from Michigan State University, East Lansing, MI, USA, in 1985 and 1988, respectively. From 1988 to 1990, he was an Assistant Professor of computer and information science with the New Jersey Institute of Technology, Newark, NY, USA. In 1990, he joined the faculty with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, where he is currently a Professor of the department. His research interests include parallel and distributed processing and networked embedded systems.

AMIT BANERJEE (Member, IEEE) received the Ph.D. degree in computer science from National Tsing-Hua University, Hsinchu, Taiwan, in 2009.

He worked for two years as an Engineer with the SoC Technology Center, Industrial Technology Research Institute (ITRI), Taiwan. He is currently working as an Assistant Professor with the Department of Computer Science, South Asian University (SAU), New Delhi, India. He has authored or coauthored papers in peer-reviewed journals and conferences, including IEEE TRANSACTIONS. His current research interests include distributed computing, the Internet of Things, and edge computing.