SECURE INITIALIZATION MODEL IMPROVEMENT FOR NFC-HCE SECURITY IN MOBILE PAYMENT SYSTEM

[1,2]LN HARNANINGRUM [3]AHMAD ASHARI [4]AGFIANTO EKO PUTRA

[1]PhD Student Department of Computer Science and Electronics, Gadjah Mada University, Yogyakarta, Indonesia
[2]STMIK AKAKOM, Yogyakarta, Indonesia
[3,4]Lecturer of Department of Computer Science and Electronics, Gadjah Mada University, Yogyakarta, Indonesia
Email: [1]ln.harnaningrum@mail.ugm.ac.id, [2]ningrum@akakom.ac.id,
[3]ashari@ugm.ac.id, [4]agfi@ugm.ac.id

ABSTRACT

Near Field Communication (NFC) has two payment ecosystems based on its secure element location: NFC-SIM-SE (Subscriber Identity Module Secure Element) and NFC-HCE (Host Card Emulation). Secure elements in the NFC-HCE ecosystem are currently in the cloud, so they have security weaknesses. One of these security weaknesses is a vulnerability in the delivery process because data is manipulated and used for fake transactions. This security issue of credential data arises because of the bulk amount of data transmission in the cloud. When the information is stored in the smartphone, there is no data transmission over the global network. This credential data stored in a smartphone will later be used for a faster transaction process. We propose a model that puts the secure element to the smartphone and performs multilevel verification to fix its security issues. The proposed model consists of two stages: the account and the card registration stage. The model was tested to determine the security level using random data with the 50.34%avalanche effect,3.99542 entropy, and 0.42048 P-value tests. The test also calculates the processing time for each step, and the result is 0.0025 seconds. The test results show that the encryption process can increase the NFC-HCE ecosystem's security; only verified data could be stored on APL-SE as credential data.

Keywords: *Initialization, Payment Card, NFC-HCE Ecosystem, Mobile Payment, Encryption Process, Security.*

1. INTRODUCTION

Near Field Communication (NFC) can be used for various purposes, including communication on the adaptation layer protocol, e-tolls payment system, payment systems at the station, presence system at school or work. The mobile payment system using NFC should be considered a future payment method to develop[1]. NFC makes it easy for buyers to shop online but still allows them to buy directly (offline). Buyers can store their payment information on NFC indirect purchases, while cashiers can retrieve it using an NFC reader[2].

NFC will also be widely used in the future. Point of Sales (POS) capable of NFC will increase in number over time. It is estimated that in 2022 it will reach 4.1 million[3]. This situation can, of course, be seized as an opportunity to develop further use of NFC, especially NFC-enabled smartphones in various fields.

NFC devices can be divided into two categories, including active and passive devices. An active NFC device is an NFC device that can send messages to other devices, usually using its power. Passive NFC devices are NFC devices that can send messages when another device emulates them. Active NFC devices have three modes: read/write, peer to peer, and Card Emulation. Passive NFC devices only have a read/write mode. The mode of NFC-enabled mobile can be peer-to-peer and card emulation. In both modes, it can be used for transactions. The use of NFC for commercial transactions based on the location of the Secure Element (SE) is divided into two ecosystems, namely the Subscriber Identity Module Secure Element (SIM-SE) and Host Card Emulation (HCE). On NFC-enabled smartphones with SIM-SE-based service (NFC-SIM-SE), communication is carried out through the NFC antenna's data. The data is forwarded to SIM-SE without requiring an Internet connection. It can even operate when the device is turned off. Meanwhile, an NFC-enabled smartphone with HCE-based service (NFC-HCE) forwards data from a paired device (such as an NFC reader) to the smartphone's Operating System (OS) and requires that the device should be turned on because it needs an Internet connection.

Payment transactions made via communication between devices require an SE. NFC-SIM-SE includes SE on the SIM card, while NFC-HCE does not have SE on the device but uses SE stored in the cloud. The NFC-SIM-SE ecosystem allows payment transactions without an internet connection because the credentials are already on the smartphone. The NFC-SIM-SE ecosystem has advantages in security, usability, and maturity compared with the NFC-HCE ecosystem[4]. The NFC-SIM-SE ecosystem allows payment transactions without an internet connection because the credentials are already on the smartphone. Major card issuers such as MasterCard, Visa, and American Express operate their mobile payment services using the SE approach[5].

Unlike the NFC-SIM-SE ecosystem, which has the main SE component present in the smartphone, this main component is removed from the smartphone and transferred to the cloud in the NFC-HCE ecosystem. SE being moved to the cloud requires sending credential keys from the cloud to the device. This condition makes security a significant problem, so that proper action is needed to protect payment security. However, the NFC-HCE ecosystem that does not need to use SIM in its implementation can be advantageous and more widely applied. NFC-HCE is also a solution for the NFC mobile payment system in several countries, which impose SIMs produced without SE.

This study develops an application-based NFC-HCE model to optimize NFC-HCE performance in a mobile payment system to allow SIM with or without SE to make NFC mobile payment transactions safe. The model implements SE in the NFC-SIM-SE ecosystem into the NFC-HCE ecosystem and remains in the smartphone. This model allows communication between devices (smartphones and NFC readers) by implementing a transaction system with a security system. Overall, the model consists of two main parts: initialization and transactions. The initialization stage involves preparing an NFC-enabled smartphone to make it possible for transaction usage by registering to the server, both user (smartphone) and card (one user can be many cards). The transaction stage contains transactions between NFC, from the card owner's smartphone to the point of sales (POS). This study focuses on improving the initialization section, which is the procedure to store SE data in smartphones securely.

Nowadays, there is more data flow through the internet. End-to-end communication is overgrowing. This development has many positive implications on all fronts. Examples of this positive influence are data for research and development in various fields, the development of big data, the Internet of Things, and the increasing number of electronic transactions. On the other hand, these changes have a less favorable impact because more essential data circulate on the internet. This situation results in attacks on the data traffics. This research aims to store that data in smartphones because the owner will protect their data. Data is stored in encrypted form and goes through several stages of encapsulation.

This paper is organized as follows. The initialization section discusses the NFC payment system, its development, and its security system. The second part discusses previous studies related to NFC payment security protocols and systems, particularly NFC-enabled mobile. The third part presents a proposed model for initializing the NFC payment system with a layered security system and placing secure elements on mobile devices. The following section discusses the analysis of the model created.

2. RELATED WORKS

User identity along with random values for initialization is used by [6].Cryptography is used to protect payment applications and user account data. Another element in the mobile payment ecosystem, NFC-SE, replaces the physical credit card, which will emulate the card and store user data into the SE. Data stored in SE can be managed by a trusted third party (TTP), and multiple contactless applications can be stored and executed on SE. This payment system uses the NFC-SIM-SE ecosystem and uses smartphones and SIMs as a means to store SE data. This situation shows that the payment system protocol made still requires a third party other than the bank or financial institution.

Research on payment system protocols and their security continues. Security model using a hash function and multiple authentications include initialization. The secure authentication protocol for NFC (SAP-NFC) is proposed by[7], a protocol for overcoming replay attacks, impersonating attacks, tracking attacks, and desynchronizing attacks using the registration or initialization and authentication phases. The SAP-NFC protocol makes it more possible to overcome attacks because authentication is carried out for each data exchange. This protocol can be relied on to overcome attacks. But authentication that occurs all the time will cause high computational costs. At the registration stage, authentication is carried out on the smartphone that sends data. The authentication between the smartphone owner and the card data owner has not yet been shown. Scenario relay attack is made on a system involving a smartcard, smartphone on both sides, and is connected to the Internet network[8]. Relay attack attacking smartphones by changing the peer-to-peer mode into vogue Card Emulation and then reading on the smartcard. Messages are captured and forwarded to the proxy device, and the device sends messages as if it were the actual smartphone owner. This relay attack occurs at the application layer, so take real-time precautions must be taken. The study on relay attacks on NFC by analyzing the weaknesses of ISO / IEC 14443-4 when dealing with relay attacks was conducted by[5]. This drawback appears to be quite common for all Automated Fare Control (AFC) systems that adhere to this standard globally. Then designed the experimental relay method and perform the relay attack. The results show that the protocol is vulnerable to attack. Two countermeasures are then proposed and discuss the appropriateness and practicality of these countermeasures. The results show that the attack was successful by producing a delay during the transaction. In this study, the attacks studied are attacks during transactions. Attacks can also occur during registration and when data is stored in SE.

The initialization stage to ensure NFC devices are registered in the Authentication (AuC) database is used [9]. This initialization stage uses four steps. Firstly, it sends a registration request message containing a random identity and number. In response to the registration request, AuC generates a secret key. The confirmation message is then sent back to the NFC device by the AuC. Finally, the NFC device executes a derivative function to obtain a secret key. The statement that pure HCE is not yet trusted for payment still has attracted the attention of Visa Master Card is said by [10]. For that, it is necessary to make efforts to gain that trust. The statement that NFC is suitable for IoT devices is stated by[11]. This paper discusses the protocol for sending data using certificates in the NFC Data Exchange Format (NDEF) format. Security is carried out in several ways. The initialization will be terminated if the modified certificate preceded by the requested certificate via handshake does not match. Modified data is also traced. If the signatures do not match, it means there is a modification to the data. The data is discarded. There is a mechanism for checking the hash code between the message and the signed hash. Jamming attacks can be detected by interference. So, if there is interference, data is prohibited from being stored.

The attacks with reapplication tags by tracking the number of times the tag read are identified by[12]. When the goods are distributed and arrived at their destination, the tag should only be read once. If it is verified that the tag is fake, then the original tag is considered the second reading. The NFC protocol, which begins by making user initialization, was created by[4]. One can do this by checking the validation and the PIN. Then the data is stored on the tamper-resistance chip. All that process is done on the NFC-HCE ecosystem. The NFC communication mechanism to prevent eavesdropping using electronic circuits was created by[13]. As a result, it can reduce the threat of eavesdropping. The protocol preceded by initialization is made by[14], which at the initialization stage, a pseudorandom generator and key are generated. Then place them both in the valid and legitimate reader tags. The reader develops a random timestamp and random number at the tag identification stage and authenticates the tag. Time to receive must be greater than time to transfer; otherwise, the authentication was not successful. From the security side, attacks are detected with a timestamp for the anti-DoS attack scheme.

N. El. Madoun et al.[15] conducted a study on payment protocols. EMV (Europay Mastercard Visa) is a payment protocol between the client's payment device and POS with NFC. This protocol ensures important security such as initialization, authorization, and integrity. However, there are still weaknesses in terms of security. N. El. Madoun, et al. proposed an innovation protocol to overcome these weaknesses. This study produces a protocol that can overcome the shortcomings of EMV.

Automated Fare Collection (AFC) is a transportation payment system using NFC already widely used. However, F. Dang et al. [16] identified vulnerabilities that result in lower payments than they should be. The attack is called Lesspay. This study simulates these attacks and proposes their prevention.

Research on the EMV protocol was also carried out by A. Al-Haj, et al. [17]. Like N. El. Madoun, Al-Haj also identified a severe vulnerability to EMV. Al-Haj ensures communication between POS and NFC Mobile using a secure protocol. The proposed protocol provides mutual initialization between NFC Mobile-enabled mobile devices and POS, thus solving EMV's first vulnerability.

The study conducted by N Madhoun et al.[18] was still about the EMV protocol. This study discusses EMV and possible attacks that can occur. The analysis shows that many possible attacks can occur. This protocol has four sessions. First is the initialization session. The session is used to negotiate between the client and POS / ATM payment devices. Possible attacks that can occur are that the PIN can be cloned, and the one that verifies is the bank, not the POS. The second session is the initialization of the client's payment device. There are three authentication methods: SDA (Static Data Authentication), DDA (Dynamic Data Authentication), and CDA (Combined Data Authentication). These methods can only be done if POS / ATMs support them, and they are optional. A possible attack in this session is a downgrade attack. It is changing the ability of the client device to make it vulnerable. The third session is the authentication of the client. This session performs initialization with a PIN, manual signature, or a combination of both. The client payment device must support one of these CVM (Cardholder Verification Method). The possible attack that can occur is a downgrade attack. It is changing POS's ability to it vulnerable. The fourth session is the actual transaction (online and offline). Transactions between the client's payment device and POS and verify it with the bank. A possible attack in this session is the response of the client's payment device, indicating that the POS does not authenticate the PIN code.

Nour et al. continued their work in[19]. This study analyzes whether the EMV payment system is reliable. Nour identifies an attack that can occur in the initialization, authentication, and transaction phase as well. Attacks in the initialization phase are downgrade attacks and duplication of card owner data (cloning). An attack that may occur in the authentication phase is the cloning attack made in the initialization phase. POS detects the PIN sent by the client as the correct PIN, even though the attacker or client uses a signature. So, the PIN is not required here. The attack in the actual transaction phase is caused by the security system that does not detect a change in bytes, indicating an attack. So, transactions can still be carried out.

Research on the use of NFC for ATM access was conducted by D Mahansaria et al. [20]. There are two phases of secure authentication. The first phase is the initialization phase. This phase is used to register a client. Card and smartphone are used. The second phase is the transaction phase. This phase is used to make transactions at the ATMs using NFC, conditioned on card emulation mode. The use of NFC technology has advantages, namely on smartphones. In addition to security made for NFC cards, there is also security on the smartphone itself. By using a smartphone, there is no need to print a physical ATM card, which means no need for card delivery and can support going green.

Research on mobile payments continues. Ahamad et al. [3] proposed a secure and privacy-preserving mobile commerce (SPPMC) framework for NFC proximity payment. SPPMC ensures client anonymity by using traceable anonymous certificates (TAC) and a Grid of secure elements (GSE) used on banking servers. The computation and communication costs of the SPPMC are very minimal. SPPMC ensures end-to-end security. It also resists all known types of attacks, including multi-protocol attacks.

Anonymous payment scheme based on NFC is carried out by[21]. This schematic was made for use on payment by virtual account. The first phase is initialization. It is used to register a user account with the bank. The bank account's owner can do it. Key is generated and distributed to the user. The next phase is virtual Bank account generation. In this section, the user requests a bank to create anonymous accounts. After the bank verifies the user, the bank makes a certificate for that user ID.

Communication between device and smartphone with NFC is also made to control device from a smartphone [22]. Data exchange is done by using the NFC, but Bluetooth is used when there is data accumulation.

Credit card mobile payment protocol which is done offline, is made by [23]. This payment protocol is named EOPMA. This protocol has 5 phases: mutual initialization, function selection, offline certificate application, credit quota application, offline/online transaction, and merchant payment request. These protocols can solve the problem of lack of balance on offline payment.

The studies conducted above try to analyze and create a protocol or model for the NFC payment system to make it safe and reliable. There is a possibility that the attack can be overcome in various ways. Attacks that are trying to be overcome are relay attacks, authentication of client data, and transaction data. Several studies are trying to overcome the reapplication attack against the tags used. Securing credential data, particularly in the NFC-HCE ecosystem, is discussed by placing data in the cloud or on a financial institution's server. So that at the time of the transaction, the smartphone client must contact the server for verification.

On the other hand, POS also needs to verify client data to ensure safe transactions from the right side of client data. This study seeks to reduce the communication process to the server, which means reducing the risk of attack due to the process. The trick is to save client credential data on the client's smartphone and only access it when used for transactions.

The models and protocols made in the above study are used in the NFC-SIM-SE and the NFC-HCE ecosystem. The model applied to the NFC-HCE ecosystem uses the cloud as credential data storage. The smartphone must be connected to the cloud when a transaction is made.

The model proposed in this study is used in the NFC-HCE ecosystem, but the credential data is stored on the smartphone, not in the cloud storage. Therefore, when used for transactions, the smartphone does not need to connect to the internet to retrieve credential data. This model ensures that the credential data stored in the smartphone is safe. The model also ensures that at the time of the transaction, it will reduce one step from the existing NFC-HCE model or protocol, namely verification to a third party as a credential data store.

NFC Mobile Payment System

Payment with NFC-enabled mobile in the NFC-HCE ecosystem is performed by placing a secure element in the cloud, as described in the introduction. So far, the secure element can be placed on the SIM card in the NFC-SIM-SE environment. Secure elements in the cloud will be transferred to smartphones by creating a suitable mechanism and method of storing files on smartphones so that the secure data will be safe on the smartphone.

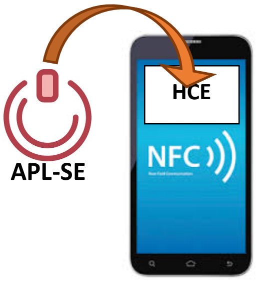


Figure 1. Architecture overview of the initialization model

Figure 1 shows the architecture of the initialization model. Model development of this NFC-HCE model will place the APL-SE as a secure element made so that it can store confidential data well. APL-SE acts like secure elements housed in the cloud. The difference is when it is used for transactions, a smartphone is not.

The mobile payment system model proposed has two stages. The first stage is the initial stage of initialization. This stage ensures that the client data and the smartphone device used are registered at a financial institution, and the same data is stored on the smartphone as credential data to be used for payments. The second stage is the transition stage. This stage is used to ensure transactions between client and POS devices are safe and correct.

This discussion is, however, focused only on the initialization section. In a mobile payment system, a reliable security system is a part that needs attention. The security system refers to the SE hardware inside the SIM card in the NFC-SIM-SE ecosystem. Identification element is made by synthesizing NFC-SIM-SE elements and NFC-HCE elements in the cloud into application elements. The elements present in both ecosystems are analyzed and adapted to the security system application (APL-SE) being developed. The initialization model is used to create flows and methods so that the data stored is safe. Safety will be ensured for the verification process and data storage format on the smartphone.

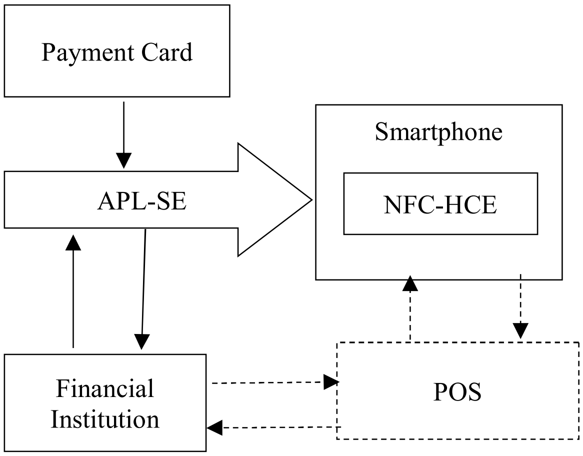


Figure 2. Secure initialization model (solid line) and transaction model (dashed line) in the proposed NFC mobile payment system

Figure 2 shows the overall architecture of the developed APL-SE model. The chart with the unbroken line is the initialization model discussed today, while the model depicted with the dotted line is the model that will be developed later. The initialization model consists of two stages: APL-SE account registration and payment card registration. The APL-SE account registration stage is the stage used to register an account with a financial institution. One smartphone is only for one account. This stage involves smartphone entities, APL-SE, and financial institutions. The card registration stage is the stage used to register the card that will be used for payment. Cards registered with APL-SE are the cards that have been registered as active at financial institutions. One smartphone can be used for more than one card. This stage involves payment card entities, smartphones, APL-SE, and financial institutions.

The procedure for the APL-SE account registration stage starts by running APL-SE. The user enters an ID number (ID) in payment card numbers, then APL-SE will retrieve the ID and took the specific device (SD) data from smartphones. Both data are processed, encrypted, and generate ID+SD data. The encryption result is then sent to the Financial Institution's server for verification. On the financial institution side, the data is decrypted, then verified. Financial institutions send a message to a smartphone containing the verification results, YES if the data is verified, and sends the message NO if it is not verified. If the information is not confirmed, the process will repeat from the first step. Meanwhile, if the data is verified, APL-SE will save the card data in an encrypted base 64 formats, after encrypted in AES encryption.

Data security is required in the initialization model; therefore, account data (username, password, and other required identity data) is encrypted. Then, the data is entered into JSON Object format and encrypted again. Enter it into the JSON object and then send it to the financial institution's server for verification. After the data is verified, the information is sent back to the smartphone in the JSON Object format. Once received by the smartphone, the data is encoded using Base64 for storage. This process can be seen in Figure 3.

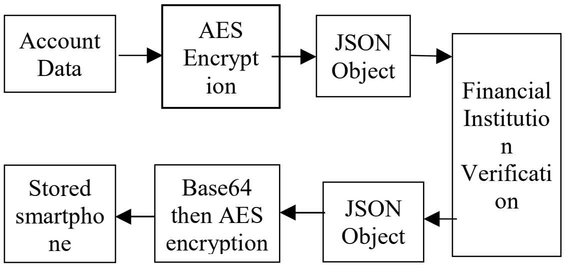


Figure 3. The encryption processes.

The card registration stage is the second part of the initial authentication model. The detailed step in the model initial authentication phase of card registration is explained as follows.

The first step is registering a user and password to a financial institution as the payment card provider. User password is used for logging into the system. This login is used to match user data and passwords with existing data in the financial institution system. There are three matching data, namely user, password, and id_tools. User data, passwords obtained from the input are then encrypted,

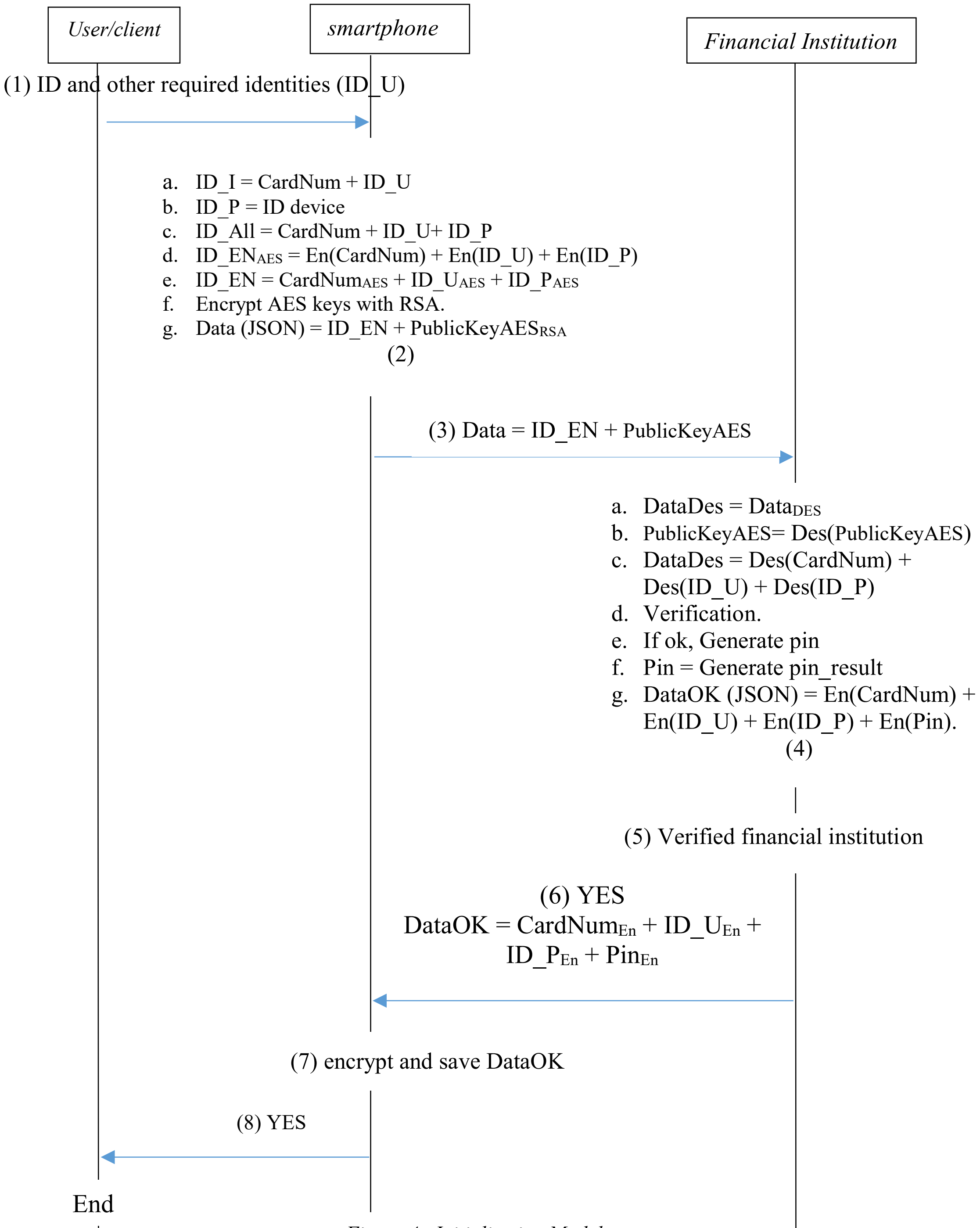
$$UsrAES = En\ (user) \quad (1)$$

And

$$PaswAES = En\ (password) \quad (2)$$

The Android_id obtained from reading the device being used is also encrypted,

$$AdrAES = En\ (device\_id) \quad (3)$$


User/client
smartphone
Financial Institution
(1) ID and other required identities (ID_U)
(2)
(3) Data = ID_EN + PublicKeyAES
(4)
(5) Verified financial institution
(6) YES
(7) encrypt and save DataOK
(8) YES
End

a. ID_I = CardNum + ID_U
b. ID_P = ID device
c. ID_All = CardNum + ID_U+ ID_P
d. ID_EN$_{AES}$ = En(CardNum) + En(ID_U) + En(ID_P)
e. ID_EN = CardNum$_{AES}$ + ID_U$_{AES}$ + ID_P$_{AES}$
f. Encrypt AES keys with RSA.
g. Data (JSON) = ID_EN + PublicKeyAES$_{RSA}$

a. DataDes = Data$_{DES}$
b. PublicKeyAES= Des(PublicKeyAES)
c. DataDes = Des(CardNum) + Des(ID_U) + Des(ID_P)
d. Verification.
e. If ok, Generate pin
f. Pin = Generate pin_result
g. DataOK (JSON) = En(CardNum) + En(ID_U) + En(ID_P) + En(Pin).

DataOK = CardNum$_{En}$ + ID_U$_{En}$ + ID_P$_{En}$ + Pin$_{En}$

Figure 4. Initialization Model

UsrAES is user data that has been encrypted using the AES encryption algorithm. PaswAES is a password data that has been encrypted using the AES encryption algorithm. And AdrAES is a unique identity data from smartphone device that has been encrypted using the AES encryption algorithm.

Referring to equations (1), (2), and (3), these three data are entered into the JSON Object, then encrypted. The encryption result is inserted into the JSON Object. Abbreviations used in this paper as in Table 1.

Table 1. Abbreviations used in this paper

Abbr.	Description
ID	Identity
ID_U	Identity of user
CardNum	Card number used for transactions
ID_I	Combination of user identity and card number
ID_En	Encrypted ID_I
ID_P	Identity of smartphone device
Pin	Transactions use the pin

The next step from the card registration stage is to register and prepare the card data to be used on a smartphone. This step is shown in Figure 4, the initial authentication model. Data names and other required identities, card numbers, smartphone IDs are encrypted using AES. The AES key is generated by the smartphone and sent to the financial institution's server. The data is then processed and produces output data in the JSON object format. Card number, other required identity data are combined into equation (4).

$$ID\_I = CardNum + ID\_U \quad (4)$$

User data used as an identity can tailor to the needs of financial institutions. However, user ID and smartphone ID data as part of the security system must exist. When the data is processed, the device ID is added, becoming equation (5).

$$ID\_All = ID\_I + ID\_P \quad (5)$$

Then, all data is encrypted separately, as in equation (6)

$$ID\_EN_{AES} = En\ (CardNum) + En\ (ID\_U) + En\ (ID\_P) \quad (6)$$

The AES encryption key is encrypted with RSA. The RSA public key is known to all parties involved in this process, the RSA of AESKey = RSA(AESKey). The encrypted data is then stored in the JSON Object format as in equation (7).

$$JSON\ (Data) = ID\_EN_{AES} + AESKey_{RSA} \text{ or } JSON\ (Data) = CardNum_{AES} + ID\_U_{AES} + AESKey_{RSA} \quad (7)$$

Then, the JSON Object data is encrypted and entered to the JSON Object. The encrypted data in the JSON Object is then sent to the financial institution's server for verification. If it has been verified, the data is stored on the server, and the server provides notifications to the customer to be stored on the customer's smartphone. The verification conducted by financial institutions is as follows:

a. DataDes = $Data_{DES}$;
b. AESKey = Des($AESKey_{RSA}$);
c. DataDes = Des(CardNum) + Des(ID_U) + Des(ID_P);
d. Verified;
e. If ok, Generate pin;
f. Pin = Generate pin_result;
g. DataOK (JSON) = En(CardNum) + En(ID_U) + En(ID_P) + En(Pin).

The server will again send verified data to the smartphone. Before sending data, the data is encoded using Base64. After the data reaches the smartphone, the data is decoded with Base64, data = encode Base64(dataOK). Then the data is broken down into each attribute, as in equation (8).

$$dataVer = En\ (CardNum) + En\ (ID\_U) + En(ID\_P) \quad (8)$$

Then the data is converted back to JSON format and encoded. Then save on the smartphone in the form of a file with the Base64 format.

Testing and results are as follows. Experiments are carried out by implementing the system on a smartphone with the Android operating system 8thand9thversion with 2GB, 4GB, 6GB memory size. The experiments use dummy data for input. Data is created by changing one bit for the current data and the following data. The data randomness analysis was then carried out by calculating the avalanched effect, entropy, and mono bit test. Besides the three analyzes, the experiment is carried out to evaluate the testing time for each process.

The first test is done by getting the avalanche effect of encrypted data, and the avalanche effect results with the best parameters. The avalanche effect is obtained by comparing the different bits between the data before and after encryption. Tests are done by making dummy data representing the card data and the user, then calculating the bit length of the data. Furthermore, the data is encrypted, and the size of the data is also calculated. The two data are compared for each corresponding bit value. The avalanche effect is declared good if it is more than 50%[24]. Avalanche effect in this experiment, which was conducted 50 times, the average avalanche effect was 50.34%. This test shows that the encrypted data is random.

In this second test, the encrypted data is processed to search the entropy and P-value. The entropy value is obtained to get the frequency of the possibility of data occurrence. The entropy can show the level of randomness of the data. The P-value is obtained to get a value that indicates the level of data randomness. This test is based on the monobit test. In addition to finding the entropy and P-value values, this second test is also used to calculate the time used by existing processes, the encryption process, and the process of sending data to the financial institution server.

Monobit test and entropy calculation are shown in Figure 5 and Figure 6. The data is retrieved from testing the process in the model Figure 4. The tested AES key lengths are 128, 192, and 256, each tested on the same data. The values of entropy and P-value are then calculated for each data. By using 50 data records, the average value obtained is as seen in Figure 5 and Figure 6.

From Figure 5 and Figure 6, the P-value is in the range of 0.411 - 0.425 for various lengths of different encryption keys. This value is more excellent from 0.01, that is, the value specified as the limit so that the encrypted data is declared random. Meanwhile, the entropy value is in the range of 3.99. This value approximates the value for the declared data random, which is 4 (for a change of 1 bit, then the value entropy is 2^2). When viewed from the existing data, the average P-Value for API variations and memory size from smartphones, the largest average P-Value value is API 28 and memory size is 2GB, while the smallest is API 28 and size 4GB memory. Comparison for the AES key length, the largest P-Value value is at AES 128 key length and the smallest at AES 256 key length.

When the data is sent to the financial institution's server, the AES key generated on the smartphone is encrypted first with the RSA encryption method. The RSA encryption process carried out on each data is done with the following steps.

a. Specifies the bit length for encryption
b. Generates two random numbers (p and q)
c. Get the modulus value n = p * q
d. Calculating phi = (p-1) * (q-1)
e. Get the value of the public key e by generating a random number, where 0 <= e <= phi and e are prime numbers.
f. Get the private key value d = $(e/phi)^{-1}$.
g. Encrypt data, AES key
   i. String data changed to BigInteger,
      a. keyAESBA = keyAES.toByteArray.
      b. keyAESBI = BigInteger(keyAESBA)
   ii. EN(keyAES) = $(keyAESBI/n)^d$

This encrypted AES key then looks for the entropy value and P-value. The entropy and P-values also indicate that the data encryption results are declared random. When P-value≥ 0.01, then the data is declared random [24].

Figure 5 and Figure 6 show that smartphone variations also affect. Legend shows the variation of the API and memory size of the smartphone. Apart from variations in memory size, the amount of memory remaining also affects. Another thing that affects is the existence of applications that run together on the smartphone. However, based on the trial results, the variations affect but are not significant enough to make the results differ from one another.

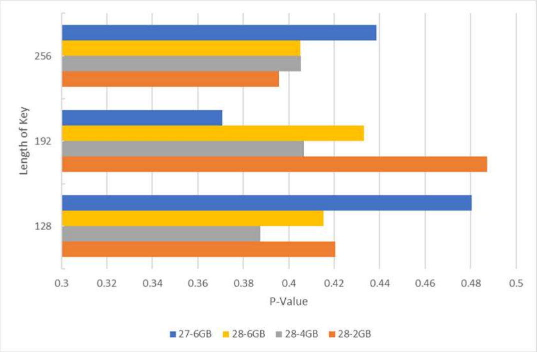


Figure 5. The P-value for various smartphone situations

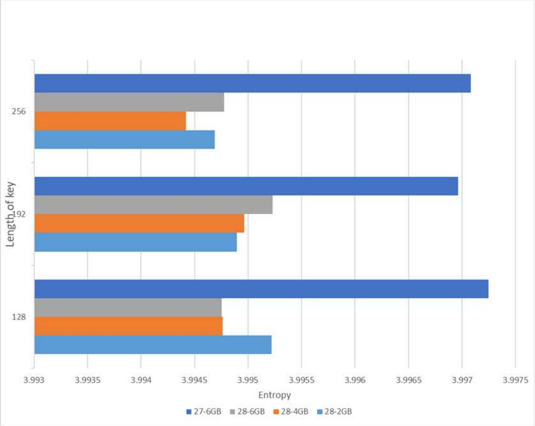


Figure 6. Entropy values for various smartphone situations

The third test is carried out to see the time used to carry out the processes. The time required is shown in Figure 7. From Figure 7, the time needed is no more than 1 second for each process. The mean time to encrypt the whole data is 0.565 milliseconds, which is<0.001 seconds. This 0.565 millisecond processing time is very short compared to the time the attacker retrieves and translates data that takes more than one millisecond. This time is very short compared to the time it takes for an attacker to retrieve and translate encrypted data, which takes more than one millisecond. When P-value≥ 0.01, then the data is declared random [18].The authentication time to the financial institution server is shown in Figure 8, and the time to store data on the smartphone is shown in Figure 9. The average time used to authenticate to the financial institution server is 404.99 milliseconds. This time varies widely because the speed depends not only on the processing speed on the smartphone but also on the data speed when passing through the internet network. Meanwhile, the average time to store data on a smartphone is 2.51 milliseconds or 0.0025 seconds.

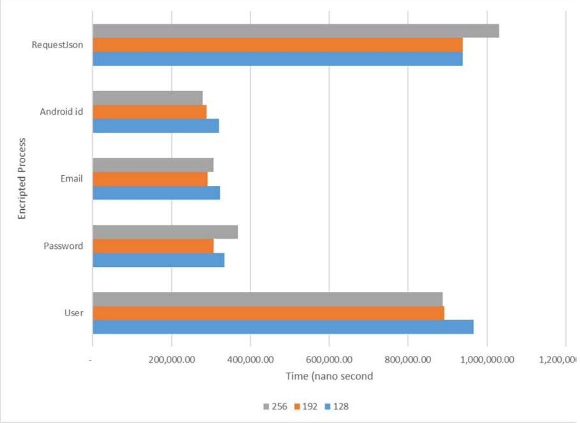


Figure 7. Encryption time for various processes

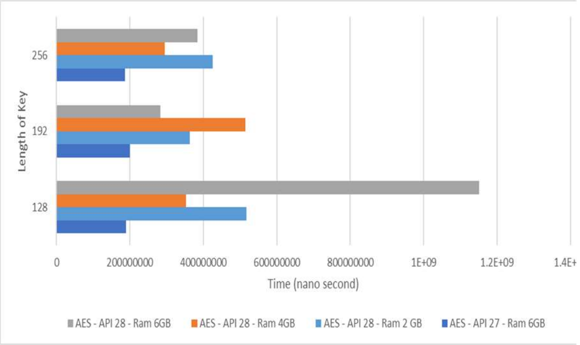


Figure 8. Authentication Time to the server for various smartphone situations

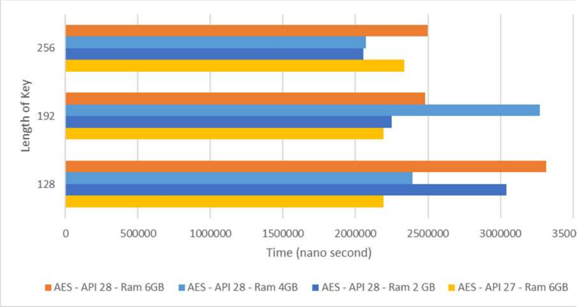


Figure 9. Data saving time for various smartphone situations

Based on the results of this evaluation, the model was then compared with other similar studies. Some of the things analyzed are as follows.

(1) Mutual Authentication. [13] identified the tag as the original tag by sharing the ID between the client and the POS. [14] Authentication from small merchant to client payment device and vice versa. [16] The proposed protocol provides Authentication on NFC-enabled mobile devices. [17] uses three EMV authentication methods: SDA (Static Data Authentication), DDA (Dynamic Data Authentication), and CDA (Combined Data Authentication). [19] enters a PIN and is sent to the server for authentication, using OTP (one-time-password). [20] validate the client using a digital signature. The initial authentication model in this study ensures the client by registering client data, card data, and device data. The client ensures the correctness of the data by entering a pin and smartphone identity data used. POS provides clients by verifying data to financial institution servers.

(2) Security level and authentication layer. [13] has four layers of authentication, but only one layer in the initialization section. [14] uses 7-step encryption and decryption with a two-phase process. This model is pretty much encrypted and decrypted. In the research, the enrollment phase for merchants is done with the user, password, and other personal data. Data encryption is performed using the RSA method, and the key is generated because of the keystore[15]. Thus, in[18], security is carried out in phase initialization to confirm the clients' device for payment. The initialization model in this study ensures that data is safe when sent and received back by the client device. The data is encrypted by each item and encrypted again as a single record before being sent.

The evaluation of possible attacks is as follows. Attacks can retrieve data at the time of a transaction, both during account registration and card registration, and then use it another time. The problem ensures that the transaction is done by the right person (the actual account owner). This attack can be prevented from encrypting data twice before sending it to and from servers and smartphones. This process is shown in the second step of Figure 4. Data is encrypted per item, then entered to JSON format and sent to the server. Possible attacks can also be detected by the length of time required for the process. If enough time is used, there is a possibility that the data is attacked. By knowing this possibility, existing data can be evaluated and tested for authenticity. Based on the test results, it takes less than 1 second for each process.

This initialization model can prevent attacks. The comparison of this initialization model with other studies in terms of attack prevention can be seen in Table 2.

Table 2. Comparison Attack Prevention

Paper	Attack Prevention
[14]	Reduce denial of service attack
[15]	Replay attack, the man in the middle attack
[16]	relay attack
[17]	the impersonation and replay attack, the session key security attack, the brute force attack, and the Man-in-the-middle attack
[25]	Phishing attack
Proposed model	An attack that takes data and then uses it is an attack that uses duplicated data to manipulate the original data.

Prevention of attacks that can retrieve data and use it is carried out by processing two sub-process d, e, and f. This process ensures that data is encrypted after the encryption key is generated while the public key is encrypted before transmission. An encryption key is generated for each process. It will differ from one process to another. Prevention is also done by making the process fast and straightforward. This prevention is evidenced by the time it takes to perform the encryption and transmission process far below one second.

Attacks prevention that uses duplicate data to manipulate the original data is done by doing two-level encryption. The data is then saved in a base64 format. The data encryption results evidence this is declared random.

Smartphones with different memory, different versions, and the memory allocation used when the test was conducted. The trial time required varies with the state of the smartphone. From this quite different situation, we can see that the result is not much different and is still in a relatively small range, far under one second.

An attack that can retrieve data and can analyze data. The problem is how the data can remain safe even though the attacker successfully retrieves the data. Such attacks can be prevented by storing data by encoding it first.

Table 3. Secure Element Location Comparison to the ecosystem

Research	Ecosystem	Secure Element Location
[20]	NFC-HCE	cloud
[17]	NFC-HCE	cloud
[15]	NFC-SIM-SE	Smartphone
[6]	NFC-SIM-SE	Smartphone
[5]	NFC-SIM-SE	Smartphone
This research	NFC-HCE	Smartphone

Based on the ecosystem and the location of the secure elements, this model can be compared with other models or protocols. Table 3 shows that comparison. This study uses the NFC-HCE ecosystem, but puts the secure element on the smartphone, not the cloud or a trusted third party.

3. CONCLUSION

The mobile payment system is an important part to be researched and studied continuously. The security system required in the mobile payment system model is continuously developed to answer the user's trust. The mobile payment system model has two ecosystems, NFC-SIM-SE and NFC-HCE. This research has succeeded in modeling a mobile payment system in a secure NFC-HCE ecosystem by placing SE on a smartphone. The data stored on the smartphone is declared safe by testing the randomness of the data with avalanche effect, entropy and monobit test. The test results show that the encrypted data is declared random, thus data security is guaranteed. In addition, the data is also authenticated before the data is stored on the smartphone. Thus, the data stored on the smartphone is safe and reliable.

Further research will create a transaction model on a mobile payment system based on this initialized data.

REFERENCE

[1] J. Ondrus and Y. Pigneur, “An assessment of NFC for future mobile payment systems,” in *Conference Proceedings - 6th International Conference on the Management of Mobile Business, ICMB 2007*, 2007, no. May 2014, pp. 43–49, doi: 10.1109/ICMB.2007.9.

[2] S. J. Olivieri, “An Investigation of Security In Near Field Communication Systems,” 2015.

[3] S. S. Ahamad and A. S. K. Pathan, “Trusted service manager (TSM) based privacy preserving and secure mobile commerce framework with formal verification,”

Complex Adapt. Syst. Model., vol. 7, no. 1, pp. 1–19, 2019, doi: 10.1186/s40294-019-0064-z.

[4] P. Pourghomi, P. E. Abi-char, and G. Ghinea, “Towards a mobile payment market: A Comparative Analysis of Host Card Emulation and Secure Element,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 13, no. 12, pp. 156–164, 2015.

[5] P. Pourghomi, S. E. Seker, G. Ghinea, and W. Masri, “Java Implementation of a Cloud-based SIM Secure Element NFC Payment Protocol,” *Int. J. Secur. Its Appl.*, vol. 10, no. 12, pp. 261–282, 2016, doi: 10.14257/ijsia.2016.10.12.21.

[6] M. Badra and R. B. Badra, “A Lightweight Security Protocol for NFC-based Mobile Payments,” *Procedia Comput. Sci.*, vol. 83, no. Ant, pp. 705–711, 2016, doi: 10.1016/j.procs.2016.04.156.

[7] S. Nashwan, “Secure Authentication Protocol for Mobile Payment,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 8, pp. 256–263, 2017, doi: 10.26599/tst.2018.9010031.

[8] D. Cavdar and E. Tomur, “A practical NFC relay attack on mobile devices using card emulation mode,” *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc.*, no. May, pp. 1308–1312, 2015, doi: 10.1109/MIPRO.2015.7160477.

[9] M. Al-fayoumi and S. Nashwan, “Performance Analysis of SAP-NFC Protocol,” *Int. J. Commun. Networks Inf. Secur.*, vol. 10 No 1, no. April, p. 125, 2018.

[10] M. Alattar and M. Achemlal, “Host-based card emulation: Development, security, and ecosystem impact analysis,” *Proc. - 16th IEEE Int. Conf. High Perform. Comput. Commun. HPCC 2014, 11th IEEE Int. Conf. Embed. Softw. Syst. ICESS 2014 6th Int. Symp. Cybersp. Saf. Secur.*, pp. 506–509, 2014, doi: 10.1109/HPCC.2014.85.

[11] A. Asaduzzaman, S. Mazumder, and S. Salinas, “A Security-Aware Near Field Communication Architecture,” *2017 Int. Conf. Networking, Syst. Secur.*, no. January, 2017.

[12] N. Alzahrani, “Securing Pharmaceutical and High-Value Products Against Tag Reapplication Attacks Using NFC Tags,” *2016 IEEE Int. Conf. Smart Comput.*, 2016, doi: 10.1109/SMARTCOMP.2016.7501715.

[13] O. Wenxing, W. Lei, Z. Yu, and Y. Changhong, “Research on Anti-eavesdropping Communication Mechanism for NFC,” *Proc. - 2015 7th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2015*, pp. 839–841, 2015, doi: 10.1109/ICMTMA.2015.206.

[14] K. Fan, P. Song, and Y. Yang, “ULMAP : Ultralightweight NFC Mutual Authentication Protocol with Pseudonyms in the Tag for IoT in 5G,” *Mob. Inf. Syst.*, vol. 2017, no. April, 2017.

[15] N. El Madhoun, E. Bertin, and G. Pujolle, “For Small Merchants: A Secure Smartphone-Based Architecture to Process and Accept NFC Payments,” *Proc. - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018*, pp. 403–411, 2018, doi: 10.1109/TrustCom/BigDataSE.2018.00067.

[16] F. Dang, E. Zhai, Z. Li, P. Zhou, and A. Mohaisen, “Pricing Data Tampering in Automated Fare Collection with NFC-equipped Smartphones,” vol. 1233, no. c, pp. 1–14, 2018, doi: 10.1109/TMC.2018.2853114.

[17] A. Al-Haj and M. A. Al-Tameemi, “Providing security for NFC-based payment systems using a management authentication server,” *2018 4th Int. Conf. Inf. Manag. ICIM 2018*, pp. 184–187, 2018, doi: 10.1109/INFOMAN.2018.8392832.

[18] N. El Madhoun, E. Bertin, and G. Pujolle, “An overview of the EMV protocol and its security vulnerabilities,” *2018 4th Int. Conf. Mob. Secur. Serv. MOBISECSERV 2018*, vol. 2018-Febru, no. February, pp. 1–5, 2018, doi: 10.1109/MOBISECSERV.2018.8311444.

[19] N. El Madhoun, E. Bertin, and G. Pujolle, “The EMV Payment System: Is It Reliable?,” *2019 3rd Cyber Secur. Netw. Conf. CSNet 2019*, no. 2, pp. 80–85, 2019, doi: 10.1109/CSNet47905.2019.9108846.

[20] D. Mahansaria and U. K. Roy, “Secure authentication for ATM transactions using NFC technology,” *Proc. - Int. Carnahan Conf. Secur. Technol.*, vol. 2019-Octob, pp. 1–5, 2019, doi: 10.1109/CCST.2019.8888427.

[21] J. N. Luo, M. H. Yang, and S. Y. Huang, “An Unlinkable Anonymous Payment Scheme based on near field communication,” *Comput. Electr. Eng.*, vol. 49, pp. 198–206, 2016, doi: 10.1016/j.compeleceng.2015.08.007.

[22] D. Sethia, D. Gupta, and H. Saran, “NFC Secure Element-Based Mutual Authentication and Attestation for IoT Access,” *IEEE Trans. Consum. Electron.*, vol. 64, no. 4, pp. 470–479, 2018, doi: 10.1109/TCE.2018.2873181.

[23] J. N. Luo and M. H. Yang, "EMV-compatible offline mobile payment protocol with mutual authentication," *Sensors (Switzerland)*, vol. 19, no. 21, pp. 1–24, 2019, doi: 10.3390/s19214611.

[24] W. Stallings, *Cryptography and Network Security: Principles and Practice, International Edition: Principles and Practice*. 2014.

[25] S. Bojjagani, D. R. D. Brabin, and P. V. V. Rao, "PhishPreventer: A Secure Authentication Protocol for Prevention of Phishing Attacks in Mobile Environment with Formal Verification," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1110–1119, 2020, doi: 10.1016/j.procs.2020.04.119.