A New Secure Lightweight Authentication Protocol for NFC mobile Payment

Reham Abdellatif Abouhogail

Electrical Quantities Metrology Dept., National Institute of standards (NIS), Egypt

***Abstract*:** As mobile applications grow, securing these applications become an important factor for their success. Especially, when these applications are related to financial transactions. Nowadays, mobile payment that is based on NFC technology is considered one of these important topics. In this paper, we propose A New Secure and Lightweight Authentication Protocol for NFC mobile Payment (NSLA) protocol. NSLA protocol presents a new method to update the users' identities and the valid session keys, which preserves the privacy and ensures the integrity of the system. The presented performance analysis shows that NSLA protocol satisfies low computation overhead. Moreover, the security analysis proves that NSLA protocol has an immunity against replay attack, brute force attack, denial of service attack, and others types of attacks.

***Keywords*:** Authentication protocols; Mobile payment; NFC technology; Non-repudiation; Privacy; Integrity.

1. Introduction

Near field communication (NFC) is a type of communication that facilitates the exchange and requests of data. NFC enables a little amount of data transfer through short range communication about ten centimeters. NFC has many applications. For example, applications in transportations, healthcare, and access control. In transportations, NFC has an application in e-ticketing like ticket booking, then link this booking with your electronic wallet. In healthcare, NFC can make the interaction between the patient and doctor, nurse, or pharmacist more easy. Like observing the amount of medicine, getting information about blood, and pressure. Moreover, NFC has an application in access control, which some authorities or universities can implement a system based on NFC technology to check the identities of the persons who have special access rights.

All these mentioned applications require at least verification for the three most important measuring security parameters, the availability, the integrity, and the confidentiality of the sent and received data. Otherwise, many problems can be occurred. Like mistake in airline booking in the transportation application, damage may affect the health of the patient in healthcare applications, and many other problems. However, the main application for NFC is the contactless payment. Nowadays, Security in contactless payment based NFC communication is considered a challenge. You have to protect your payment information and your identifying personal information [1], and you have to ensure the authenticity between the three shared entities. The three shared entities are the Authentication Server (*AS*), the Point of Sale (*PS*), and the NFC Mobile user (*M*). When *AS*, *PS*, and *M* can authenticate each other, this is called mutual authentication property. Mutual authentication property is very important factor in electronic payment, and is considered a metric of success for any transaction operation. This is because it ensures preventing fraudulent activity. Moreover, payment using mobile devices produces some limitations, like: limitations in mobile devices, and limitations in memory storage. To satisfy acceptable mobile payment procedures, many models are proposed [2, 3, 4, 5, 6]. In this paper, we focus in the authentication protocols in NFC mobile payments, which a new authentication protocol is presented. The new proposed protocol satisfies lightweight computation overhead, and satisfies the essential security requirements like mutual authentication, non-repudiation, availability, and preventing the famous types of possible attacks like replay attack, data desynchronization attack, and denial of service attack. The main contributions of this paper are summarized in the following points:

1. The proposed protocol is a lightweight authentication protocol, which its computation overhead is considered lower than the other similar proposed protocols, which is considered a challenge in NFC mobile payment applications. As will be declared in the performance analysis section.
2. The proposed protocol presents a new method for updating the session keys and the users' identities which keep them secure, and difficult to be expected. Revealing one of the previous valid keys or one of the previous valid identities doesn't mean the ability to reveal the following session keys and the following identities. Because they are randomly updated,
3. The proposed protocol satisfies the privacy of the shared users without losing the non-repudiation property, which is almost considered a difficult task for similar proposed protocols.
4. NSLA protocol has an immunity against most famous types of attacks (denial of service attack, brute force attack, and replay attack). As will be declared in the security analysis section. In the following section the related work is presented. In Section 3, the NSLA protocol is presented. In Section 4, the performance analysis is presented, the security analysis is discussed in Section 5, and finally the conclusion is presented in Section 6.

2. Related Work

Many electronic payment protocols based NFC technology are proposed in number of recent researches [7, 8, 9, 10, 11, 12, 13, 14, 15]. Some of them are based on public key encryption like in [7, 8, 14]. Others are based on symmetric key encryption like in [9, 10, 11, 12, 13]. In [7], Mohamad Badra and et. al. proposed "A lightweight security protocol for NFC-based mobile payments". This proposed protocol is for mobile payment applications based on the NFC technology using public key encryption technique. The proposed protocol discussed two cases. The first case, when the point of sale, *PS*

has an internet connection. The second case, when the *PS* has no internet connection. The protocol satisfies the mutual authentication property depending on three shared parties, the mobile device, *M*, *PS* device, and the trusted third party, *TTP*. In this case, there are five steps, and four sent messages. Both of the mobile device *M* and the *PS* device generate a random number *RVSE*, and *RVPOS* respectively. The *TTP* generates a session key by applying a pseudo random function on the generated random number by the mobile device, the generated random number by the *PS*, the received identity of the mobile device, and the XORing of the secret key between the *TTP* and the user *M*, SKTTP_SE with the secret key between *PS* and *M*, SKPOS_SE user. We are interested in the first case, because it's similar to our proposed protocol. In this case, this protocol proceeds as the following equations:

MSG#1: $M \rightarrow PS{:}ID_M, RV_{SE}.$ (1)

MSG#2:$PS \rightarrow TTP{:}ID_{PS}, RV_{SE}, Cert_{PS}.$ (2)

MSG#3:$TTP \rightarrow PS{:}EP(Session\_key||\ SKPOS\_SE)\ pubPS.$ (3);
where pub_{PS} is the public key of the point of sale.

MSG#4:$PS \rightarrow M{:}E_S(Session\_key, SK_{PS\_SE})$ (4)

The *PS* sends its certificate ($Cert_{PS}$) to the *TTP*. The *TTP* sends an asymmetric encrypted message to *PS* contains the session key and SK_{POS_SE} using the *PS*'s public key. The *PS* sends the session key to *M* symmetrically encrypted by SK_{POS_SE}. In step number 5, as proposed in this protocol, the mobile device has to compute the session key. Here, there is a certain problem, *M* hasn't the required data to compute the session key. It hasn't the $SKPOS_{\_SE}$. Because, it's generated by the *TTP*, and the *TTP* is sent it to the *PS* asymmetrically encrypted by the *PS*'s public key in MSG#3. Moreover, *M* doesn't know the RV_{POS}. RV_{POS} is generated by the *PS* and is sent to the *TTP* only in MSG #2. To solve this problem, we propose that the *PS* replace the symmetric encryption used in MSG #4 by the asymmetric encryption using the mobile device public key. However, this will lead to a computation overhead problem due to the limited capability for the mobile devices. Moreover, sending the real identity of the NFC users violates the privacy. In [13], Ceipidor et.al.'s scheme proposed a protocol called KerNeeS, this protocol is based on Needham_Schroeder symmetric key protocol. It satisfied the mutual authentication. However, sending the identities of the entities makes it vulnerable to tracking attack. In [9], a Secure and Efficient Mutual Authentication Scheme for NFC Mobile Devices is proposed. The authentication phase of this protocol consists of five messages. The session keys are updated. Therefore, forward and backward secrecy are satisfied. The protocol is a lightweight protocol. But, this protocol has some drawbacks: 1- the privacy and the non-repudiation properties are not satisfied in this protocol. 2- the message sent from the server to the point of sale and the last message sent from the point of sale to the mobile have no identity. It's only an output of MAC function, which making it difficult for the receiver (either *PS* or *M*) to determine the identity of the sender especially for the *PS*, which deals with more than mobile user. As it's known, *PS* has to know this message is belonged to which mobile user to extract the required data from its database, and calculates the MAC function to make the required comparison with the received MAC. In [11], a SAP-NFC protocol is proposed. The protocol is performed between three entities, the NFC mobile, the point of sale, and the authentication server. The mutual authentication is satisfied between these three shared entities. This protocol proposes a solution to data desynchronization attack. The privacy property is satisfied in this protocol. However, SAP-NFC protocol has a problem with the non-repudiation property; it is not satisfied. Moreover, it's vulnerable to brute force attack, and denial of service attack.

3. The Proposed Protocol (NSLA Protocol)

In this section, we propose a new and lightweight authentication protocol applicable for mobile payments applications using NFC technology. The proposed NSLA protocol consists of three entities. They are two NFC devices, the NFC mobile device, *M*, and the point of sale device, *PS*, and the authentication server, *AS*. NSLA protocol contains two phases, the registration phase and the authentication phase as shown in Fig.1. Table1 presents the important notations in our protocol.

Table 1. Notations

Notation	Meaning
AS	Authentication Server
M_i	NFC mobile device *i*
PS_i	Point of sale *i*
ID_M^i, and ID_{PS}^i	Identity of the NFC mobile, and identity of the point of sale, respectively.
I_M^i, and I_{PS}^i	pseudonym of the NFC mobile *i* and pseudonym of the point of sale *i* respectively; where *i*=0, 1, ...*n*,.
P_M^i / P_{PS}^i	a pseudo random number assigned for the NFC mobile, and the point of sale, respectively.
K_M^i, and K_{PS}^i	NFC mobile session key , and point of sale session key generated by the authentication server, respectively; where *i*=0, 1, ...*n*,.
R1	Nonce random number generated by *M/PS* during the registration phase.
R2	Nonce random number generated by *AS* during the registration phase.
N1	Nonce random number generated by *PS* during the authentication phase.
N2	Nonce random number generated by *M* during the authentication phase.
H(*m*)	Hash function of the message *m*.
MAC(*m*, *k*)	Message authentication code function (MAC) for the message *m* using the key *k*.
$E_s(m)k$	Symmetric key encryption function for the message *m* using the key *k*.
Ds(*m*) k	Symmetric key Decryption function for the message *m* using the key *k*.
$E_P(m)k$	Asymmetric encryption function for the message *m* using the key *k*.
MSG#*n*	Message number *n*.
V_M	The verification message that is computed by the NFC mobile.
V_{PS}	The verification message that is computed by the point of sale.
C_M and C_{PS}	Two confirmation messages are computed by the *AS* for the *M* and *PS* respectively.

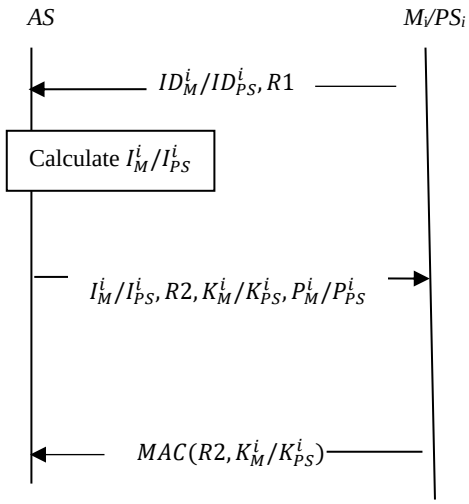

Figure 1. The registration phase in NSLA protocol.

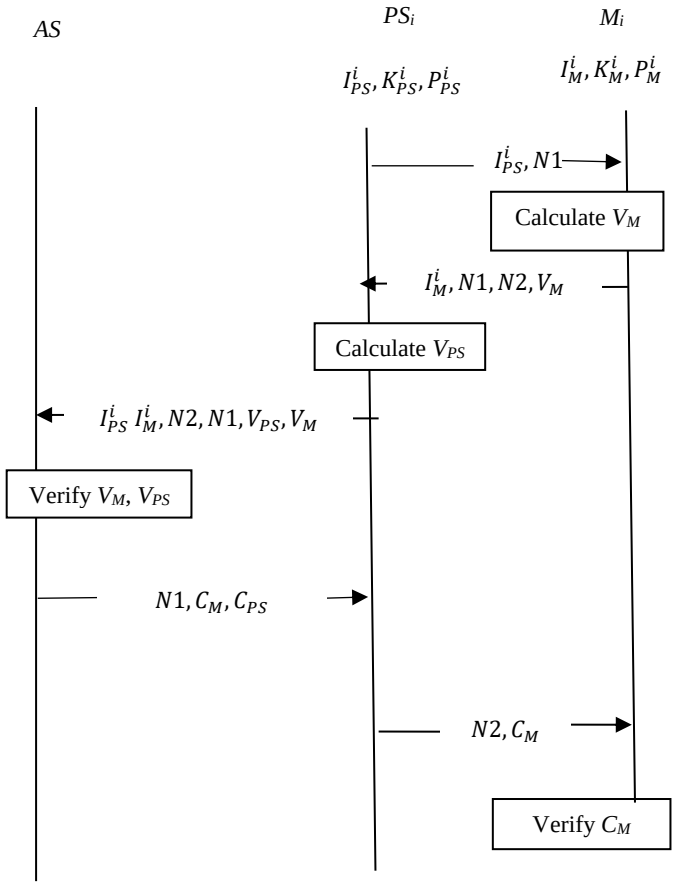

Figure 2. The authentication phase in the NSLA protocol.

A. Registration phase

In this phase, we assume that the NFC mobile device M_i or PS_i performs the registration procedures with the Authentication Server (*AS*) through a secure channel. The secure channel is used for the registration phase only. The registration phase isn't frequently repeated as the authentication phase. So this assumption will not cause heavy processing in the system. Moreover, it increases the confidence level in the obtained data. The steps of registration are as follows:

MSG#1: $M_i/PS_i \rightarrow AS: ID_M^i/ID_{PS}^i, R1$ (5)

MSG#2: $AS \rightarrow M_i/PS_i: I_M^i/I_{PS}^i, R2, K_M^i/K_{PS}^i, P_M^i/P_{PS}^i$ (6)

where; I_M^i is the pseudonym of the NFC mobile, I_{PS}^i is the pseudonym of the point of sale, and $I_M^i/I_{PS}^i = H(ID_M^i/ID_{PS}^i||RN)$; where *RN* is a generated random number by *AS*, and P_M^i/P_{PS}^i is a pseudo random number, its period (length) is determined according to the required level of security. Its period increases as the required level of security increases. However, you have a more communication overhead for long P_n. So here, there's a tradeoff. Hence, the *AS* has to have a good pseudo random number generator to generate random numbers for each NFC user. Each three bits in this P_n forms a number. The NFC device uses these numbers in order to update its own data.

For Example: if certain NFC device is given a pseudo random number, P_n; where P_n = 001 110 010 111 000 011 100 101. So the required number of hash operations are = 1, 6, 2, 7, 0, 3, 4, 5 in order after each successful payment operation. Therefore, according to this order of numbers, this NFC device has to hash its pseudonym and its session key after the first successful payment operation one time to get its new pseudonym and its new session key, and after the second successful payment operation, the NFC device has to apply the hash function six times to the last pseudonym and last session key to get its new pseudonym and its new session key, and after the third successful payment operation apply the hash function two times, and etc.

After the NFC mobile device receives MSG#2, it stores I_M^i/I_{PS}^i, K_M^i/K_{PS}^i, and P_M^i/P_{PS}^i in its database, and prepares MSG#3 to send it to *AS*.

MSG#3: $M_i/PS_i \rightarrow AS: MAC(R2, K_M^i/K_{PS}^i)$ (7)

After receiving MSG#3, *AS* calculates $MAC(R2, K_M^i/K_{PS}^i)$, and compares the result with the received one to verify the integrity of K_M^i/K_{PS}^i. To be sure that the following authentication verification steps will be happened correctly, you must be sure on the correctness of this key.

Note that I_M^i/I_{PS}^i and K_M^i/K_{PS}^i are changed after each successful payment operation or after number of successful payment operations according to the management of the system. By computing its hash value *l* number of times; where *l* equals certain digit number in the pseudo random number for this NFC device according to the order. After registration, the *AS* stores I_M^i/I_{PS}^i, K_M^i/K_{PS}^i, and P_M^i/P_{PS}^i in its database. Moreover, the *AS* still keep I_M^i, and K_M^i after updating them to satisfy the non_repudiation property. The three shared entities in the protocol *AS*, PS_i, and M_i are ready now to proceed into the authentication phase as presented in the following Subsection.

B. Authentication phase

After the registration phase, both of M_i and PS_i own the required data to begin the authentication phase as shown in Fig.2, the procedures will be as follows.

The PS_i sends MSG#1 to M_i, which contains a generated random number *N1* and the *PS*'s pseudonym I_{PS}^i.

1. MSG#1: $PS_i \rightarrow M_i: I_{PS}^i, N1$ (8)

After receiving MSG#1, M_i generates *N2*, and computes the verification message, V_M; where $V_M = MAC(I_M^i||N1||N2, K_M^i)$. Then, M_i sends MSG#2.

2. MSG#2: $M_i \rightarrow PS_i: I_M^i, N1, N2, V_M$ (9)

After receiving MSG#2, PS_i computes the verification message, V_{PS}; where $V_{PS} = MAC(I_{PS}^i||N1||N2, K_{PS}^i)$. Then, PS_i sends MSG#3.

3. MSG#3: $PS_i \rightarrow AS: I_{PS}^i\ I_M^i, N2, N1, V_{PS}, V_M$ (10)
After receiving MSG#3, *AS* computes V_M and V_{PS} using the stored I_M^i/I_{PS}^i, K_M^i/K_{PS}^i, and P_M^i/P_{PS}^i in its database. Then, *AS* compares the received values with the computed values, and checks the results. If the computed V_M or the computed V_{PS} is not equal the received V_M and the received V_{PS}, the *AS* ends the session, else M_i and PS_i are verified. Then, *AS* prepares a confirmation messages, C_{PS} and C_M; Where $C_{PS} = MAC(I_{PS}||I_M||N1||N2, K_{PS}^0)$, $C_M = MAC(I_{PS}||I_M|N1||N2, K_{MS}^0)$. Then, *AS* sends MSG#4.

4. MSG#4: $AS \rightarrow PS_i: N1, C_M, C_{PS}$ (11)
After receiving MSG#4, PS_i computes C_{PS}, and checks if the computed value equals the received one. If the check is fail, PS_i closes the session else PS_i authenticates M_i, and *AS*. Then, *PSi* sends MSG#5 to M_i.

5. MSG#5: $PS_i \rightarrow M_i : N2, C_M$ (12)
After receiving MSG#5, M_i computes C_M and compares the computed C_M with the received C_M. If they are not equal, *Mi* closes the session else M_i verifies PS_i and *AS*. Note: to proceed into a new authentication phase, M_i and PS_i has to update their pseudonyms I_M^i/I_{PS}^i and their session keys K_M^i/K_{PS}^i by hashing them number of times according to their pseudo random number P_M^i/P_{PS}^i as declared before. Also, the *AS* has to update the required data for each NFC device participates in the system using the stored NFC device's pseudo random number. The following section presents a performance analysis for the proposed protocol with a comparison to its similar lightweight authentication protocols.

Table 2 The computation time for different cryptographic operations [16].

The Cryptographic function	The cryptographic algorithm	Computation time (ms)/ byte
H/ MAC	SHA-1	1.28
Es/ Ds	AES	1.71
Ep/ Dp	RSA	15.21

Table 3 Performance comparison between different NFC authentication protocols.

	KerNeeS [13]	Tung and Juang's protocol [9]	SAP-NFC protocol [10]	NSLA
[1] Computation overhead	7Es+ 6Ds	8*MAC*	2Es +2Ds +10 *H*	1*H* +8*MAC*
[2] Computation time (ms)	22.23	10.24	19.64	11.52
Number of mutual messages during the authentication phase.	7	5	5	5

4. Performance Analysis and Comparison

In this section, analysis for the performance of NSLA protocol from several characteristics is presented, including computation overhead, computation time, and number of required messages, *M*, by comparing it with other schemes which are the most relevant to ours [9, 10, and 13]. The three selected authentication protocols [9, 10, and 13] are based on NFC technology. In addition to, they didn't resort to use an asymmetric key technique in their proposed design. So they are considered lightweight authentication protocols. Moreover, they contain three shared entities, two NFC devices, and an authentication server which made them more suitable for comparison with our proposed protocol. To calculate the computation cost (ms), we used the estimated computation time for different cryptographic operations, which are presented in Table 2 [16]. Performance comparison is presented in Table 3. We assume some assumptions as shown in Table 2: 1- The data size for all functions are assumed to be the same, 2- The processing time for the symmetric encryption function E_s equals the processing time for the symmetric decryption function D_s, 3- The processing time for the asymmetric encryption function E_P equals the processing time for the asymmetric decryption function D_P ,4- The processing time for the hash function equals the processing time for the MAC function.
According to the results presented in Table 2, we can see that NSLA protocol has good results between the other presented schemes. Our protocol doesn't require the mobile device to do any heavy processing during payments, which is considered more suitable for mobile devices which have limited power. From Table 2, we can see that Tung and Juang's protocol [9] has the same number of mutual messages as the NSLA proposed protocol, and it has less computation overhead. But, Tung and Juang's protocol [9] lacks two important security features, which are the privacy, and the non-repudiation as was declared in Section 2.

5. Security Analysis and Comparison

In this section, the security analysis is presented, the analysis is based on the main security requirements.

5.1 Mutual Authentication

The presented protocol assumes that the authentication channel between the shared entities is susceptible to attacks. Therefore, the NSLA protocol follows some steps to defend itself. To satisfy authentication between the three shared entities, the protocol uses a MAC functions, verification messages, and confirmation messages. In the beginning, *AS* searches in its pseudonym database list for the received pseudonyms, I_M^i and I_{PS}^i. If they are existed the *AS* can get the corresponding data for these pseudonyms like the session keys, K_M^i, K_{PS}^i and the pseudo random numbers P_M^i/P_{PS}^i. Otherwise, the M_i or the PS_i is considered not legitimate. After that, the *AS* verifies if the received *VPS* or V_M is equal to the received ones. If the calculated V_M or V_{PS} doesn't equal to the received values then the *AS* will consider the M_i or the PS_i is not legitimate, and the authentication session is terminated. On the other side, PS_i verifies if the received C_{PS} equals the calculated value. If the verification fails, the *AS* will be considered not legitimate, and the PS_i terminates the session. On the same approach, M_i verifies that the received C_M equals the calculated one. If the verification fails, PS_i will be considered not legitimate, and M_i terminates the session.

5.2 Preserving user anonymity

No adversary can verify which pseudonym corresponds to the given user identity due to the difficulty of solving the hash function; where the identity of the user is hashed with certain random number. Furthermore, these pseudonyms are updated after number of success payment operations as mentioned before.

5.3 Integrity

Integrity is satisfied by using MAC functions with secure and periodically updated session key.

5.4 Forward and Backward secrecy for key

The forward and backward secrecy for the user's key is satisfied. The adversary can't know the user's key. Because, the user's key is transmitted through a secure channel. Moreover, it's updated after each successful payment as mentioned before. However, Kernees protocol, which is presented in [13], the old sessions keys are valid.

5.5 Immunity against attacks

In this subsection, the most important and related types of attacks to the subject of the presented protocol are analyzed to assess their potential.

5.5.1 Denial of service attack

Other protocols like SAP-NFC protocol [11] suffer from a denial of service attack. Because upon the *AS* server receives the challenge messages from either the NFC mobile or the *PS*, it computes a hash function for all the stored mobiles' identities or all the stored *PSs'* identities until it finds the required match. Hence, the denial of service attack is possible by making the *AS* busy all the time in calculating a lot of hash functions responding to these erroneous received messages. In our presented protocol, this problem is solved; which the *AS* receives the hashed message (the pseudonym) and stores the pseudonyms in its database. Therefore, the denial of service attack is improbable.

5.5.2 Man in the middle attack

As long as the mutual authentication is satisfied as mentioned before, the adversary hasn't the ability to impersonate the legitimate users.

5.5.3 Replay attack

The replay attack to be probable, the adversary must know the valid session key with its corresponding pseudonym, and this's considered very difficult. Because, the session keys and the pseudonyms are updated regularly.

5.5.4 Desynchronization attack

NSLA protocol doesn't need time synchronization. Therefore, data desynchronization attack is improbable.

5.5.5 Brute force attack

Our presented scheme has high level of immunity against brute force attack. This is due to updating the used session key after number of success payments as mentioned. Moreover, our proposed design updating the key according to random rule. So even if the key is revealed, the adversary still can't guess the used key in the following session, and so on. In other presented protocols [10,11], which use the KDF function or hash function only to generate the key, if the current session key is revealed, it will be easy to know the following session key by making simple hash function.

5.6 Non-repudiation

In the presented NSLA protocol, the *AS* stores the pseudonyms and their corresponding valid session keys after updating them for a certain time to satisfy the non-repudiation property. Other protocols like SAP-NFC protocol [10] the non-repudiation property is not satisfied. Where, the *AS* updates its database immediately as soon as the NFC user's data are updated.

5.7 Security features comparison

In this subsection, we summarize all the results that we have reached to after the discussion, that is presented in the previous subsections. Table 4 presents a comparison between the presented NSLA protocol, and the authentication protocols which are presented in [9, 10, and 13]. The proposed protocol in [9] mentioned that the session keys are updated without determine the method that is used to update them. So we use the notation (=) in Table 4 to refer to the security feature that isn't exactly determined.

Table 4. Security features comparison between different NFC authentication protocols.

Security feature	[9]	[10]	[13]	NSLA protocol
Mutual Authentication	satisfied	satisfied	satisfied	satisfied
Privacy	Not satisfied	satisfied	Not satisfied	satisfied
Non-repudiation	Not satisfied	Not satisfied	satisfied	satisfied
Forward and backward secrecy	satisfied	satisfied	Not satisfied	satisfied
Immunity against Brute force attack	=	Not satisfied	Not satisfied	satisfied
Immunity against Denial of service attack	satisfied	Not satisfied	satisfied	satisfied
Immunity against desynchronization attack	=	satisfied	satisfied	satisfied

5.8. Formal analysis using BAN logic

In this subsection a formal verification of the proposed protocol is presented to ensure its validity. BAN Logic [17] for verification of authentication protocols is used. BAN logic is selected because of its excellence to prove the mutual authentication between the shared entities. Moreover, its ability to detect number of attacks like the replay attack.

Rules of BAN Logic

1. Rule 1 : the interpretation rule, $\frac{P \mid\equiv (Q \mid\sim (X,Y))}{P \mid\equiv (Q \mid\sim X), P \mid\equiv (Q \mid\sim Y)}$
2. Rule 2 : the message meaning rule, $\frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd [X]_K}{P \mid\equiv Q \mid\sim X}, P \neq Q$
3. Rule 3 : the nonce verification rule, $\frac{P \mid\equiv \#(X), P \mid\equiv Q \sim X}{P \mid\equiv Q \mid\equiv X}$
4. Rule 4: the jurisdiction rule, $\frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$
5. Rule 5: the freshness rule, $\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)}$
6. Rule 6: the synthetic rule, $P \mid\equiv (Q \mid\sim X) \rightarrow P \mid\equiv (Q \mid\sim (X,Y))$
7. Rule 7: $\frac{P \equiv (X,Y)}{P \equiv (X), P \equiv (Y)}$

The authentication protocol is completed between *AS* and *PS*, if for certain data *X*:

AS ⊨ PS ⊨ *X* , AS ⊨ X: they mean that *AS* believes that *X* is sent by *PS*; where symbol ⊨ means believes. Also, the authentication protocol is completed between *AS* and *M*, if for certain data *Y*: AS ⊨ M ⊨ Y , AS ⊨ Y: they mean that *AS* believes that *Y* is sent by *M*. The messages of the

authentication phase of the NSLA protocol can be transformed into the following formulas:
The first message is omitted, because all its components are plaintext.

$$M \rightarrow PS: I_M, \#N1, \#N2, (I_M, \#N1, \#N2)_{K_m} \quad (13)$$

$$PS \rightarrow AS: I_M, I_{PS}, N1, \#N2, (I_M, \#N1, \#N2)_{K_m,}, \left(I_{ps}, \#N1, \#N2\right)_{K_{ps}} \quad (14)$$

$$AS \rightarrow PS: N1,\ (I_M, I_{PS}, \#N1, \#N2)_{K_m,}, (I_M, I_{PS}, \#N1, \#N2)_{K_{ps}} \quad (15)$$

$$PS \rightarrow M: \#N2,\ (I_M, I_{PS}, \#N1, \#N2)_{K_m} \quad (16)$$

These are the initial assumptions:

$$AS \equiv AS \xrightarrow{\mathrm{K_m}} \mathrm{M} \quad (17)$$

$$M \equiv AS \xrightarrow{\mathrm{K_m}} \mathrm{M} \quad (18)$$

$$PS \equiv \mathrm{AS} \xrightarrow{\mathrm{K_{ps}}} \mathrm{PS} \quad (19)$$

$$AS \equiv \mathrm{AS} \xrightarrow{\mathrm{K_{ps}}} \mathrm{PS} \quad (20)$$

$$AS \equiv \#\ \mathrm{N1} \quad (21)$$

$$AS \equiv \#\ \mathrm{N2} \quad (22)$$

$$AS \equiv \mathrm{PS} \Rightarrow I_{PS} \quad (23)$$

$$AS \equiv \mathrm{M} \Rightarrow I_M \quad (24)$$

$$M \equiv \mathrm{AS} \Rightarrow I_M \quad (25)$$

$$PS \equiv \mathrm{AS} \Rightarrow I_{PS} \quad (26)$$

$$PS \equiv \#\ \mathrm{N1} \quad (27)$$

$$M \equiv \#\ \mathrm{N2} \quad (28)$$

Using Equation (13) and Equation (17) and after applying the message meaning rule, we obtain:

$$AS \equiv M|\sim\ (I_M, N1, N2) \quad (29)$$

Using Equation (14) and Equation (20) and after applying the message meaning rule, we obtain:

$$AS \equiv PS|\sim (I_{PS}, N1, N2) \quad (30)$$

Using Equation (30) and applying the interpretation rule, we obtain:

$$AS \equiv PS|\sim (N1, I_{PS}) \quad (31)$$

Using Equation (31) and applying the freshness rule, we obtain:

$$AS \equiv \#(N1, I_{PS}) \quad (32)$$

Using Equation (31 and 32) and applying the nonce verification rule, we obtain:

$$AS \equiv PS \equiv (N1, I_{PS}) \quad (33)$$

From Equation (33) and from rule 7

$$AS \equiv PS \equiv I_{PS} \quad (34)$$

From Equation (34, and 23) and from the jurisdiction rule, we obtain:

$$AS \equiv I_{PS} \quad (35)$$

From Equations (34 and 35), we can say that *PS* is the actual sender of the messages. So *AS* authenticates *PS*. The same steps will be done to proof that *AS* authenticates *M* as follows:
Using Equation (30) and applying the interpretation rule, we obtain:

$$AS \equiv M|\sim (N2, I_M) \quad (36)$$

Using Equation (36) and applying the freshness rule, we obtain:

$$AS \equiv \#(N2, I_M) \quad (37)$$

Using Equations (37 and 36) and applying the nonce verification rule, we obtain:

$$AS \equiv M \equiv (N2, I_M) \quad (38)$$

From Equation (38) and from rule 7, we obtain

$$AS \equiv M \equiv I_M \quad (39)$$

From Equations (39, and 24) and from the jurisdiction rule, we obtain:

$$AS \equiv I_M \quad (40)$$

From Equations (39 and 40), we can say that *M* is the actual sender of the messages. So *AS* authenticates *M*.
Now, we will use BAN logic to verify that *M* and *PS* authenticate *AS* to verify that the mutual authentication is verified in NSLA protocol.
By dividing Equation (15) into three parts, and by using Equation (16) we can get the following three Equations:

$$AS \rightarrow PS:\ \#N1,\ \left(I_m, I_{ps}, \#N1, \#N2\right)_{K_m} \quad (41)$$

$$AS \rightarrow M:\ \left(I_m, I_{ps}, \#N1, \#N2\right)_{K_m} \quad (42)$$

$$AS \rightarrow PS:\ \left(I_m, I_{ps}, \#N1, \#N2\right)_{K_{ps}} \quad (43)$$

From Equation (42) and Equation (19) and after applying the message meaning rule, we obtain:

$$M \equiv AS|\sim\ (I_M, I_{PS}, N1, N2) \quad (44)$$

Using Equation (44) and applying the interpretation rule, we obtain:

$$M \equiv AS|\sim\ (I_M, N2) \quad (45)$$

Using Equation (45 and 28) and after applying the freshness rule, we obtain:

$$M \equiv \#(I_M, N2) \quad (46)$$

Using Equations (45 and 46) and applying the nonce verification rule, we obtain:

$$M \equiv AS \equiv (I_M, N2) \quad (47)$$

From Equation (47) and from rule 7, we obtain

$$M \equiv AS \equiv I_M \quad (48)$$

From Equations (48, and 25) and from the jurisdiction rule, we obtain:

$$M \equiv I_M \quad (49)$$

From Equations (48 and 49), we can say that *AS* is the actual sender of the messages. So *M* authenticates *AS*.
From Equation (43) and Equation (19) and after applying the message meaning rule, we obtain:

$$PS \equiv AS|\sim\ (I_M, I_{PS}, \#N1, \#N2) \quad (50)$$

Using Equation (50) and applying the interpretation rule, we obtain:

$$PS \equiv AS|\sim\ (I_{PS}, N1) \quad (51)$$

Using Equations (51, and 27) and after applying the freshness rule, we obtain:

$$PS \equiv \#(I_{PS}, N1) \quad (52)$$

Using Equations (51 and 52) and applying the nonce verification rule, we obtain:

$$PS \equiv AS \equiv (I_{PS}, N1) \quad (53)$$

From Equation (53) and from rule 7, we obtain

$$PS \equiv AS \equiv I_{PS} \quad (54)$$

From Equations (54, and 26) and from the jurisdiction rule, we obtain:

$$PS \equiv I_{PS} \quad (55)$$

From Equations (54 and 55), we can say that *AS* is the actual sender of the messages. So *PS* authenticates *AS*. From the previous analysis and from the properties of BAN logic, we can say that the mutual authentication between the shared entities in the NSLA protocol is verified. Moreover, we can say that NSLA protocol has no redundancy, and it's free from

any type of known attacks like, replay attack, and man in the middle attack.

6. Conclusions

Today, many people use electronic payment. Later, mobile payment especially using NFC technology will spread widely as it's expected. Therefore, it's necessary to provide a safe environment for this use. With the large number of vulnerabilities in this media, it's considered not an easy task. In this paper, NSLA authentication protocol is proposed. The design of NSLA protocol assume updating for the used session keys, and the users' pseudonyms which making NSLA satisfy some essential properties, the privacy, and forward backward secrecy. A new method to update the session keys, and the users' pseudonyms is presented. This method is characterized by its randomness in generating the session keys, and the users' pseudonyms, which gives the NSLA protocol immunity against brute force attack. The security analysis proves that NSLA satisfies the mutual authentication between the shared entities, and it has an immunity against most types of attacks. The performance analysis proves that NSLA is lightweight compared to other recent proposed authentication protocols based NFC technology. So it's more suitable for NFC mobile payment application.

7. Acknowledgment

This work was supported by National Institute of Standards (NIS), Egypt.

References

[1] Derek Johnson, Mohammed Ketel, "IoT: Application Protocols and Security", IJCSNS International Journal of Computer Science and Network Security, VOL.11, No.4, pp. 1-8, April 2019.

[2] Teppo Halonen, "A System for Secure Mobile Payment Transactions**",** Master's Thesis, HELSINKI UNIVERSITY OF TECHNOLOGY, Department of Computer Science, Jan. 2002.

[3] Phone Lin, Hung-Yueh Chen, Yuguang Fang, Jeu-Yih Jeng, and Fang-Sun Lu, "A Secure Mobile Electronic Payment Architecture Platform for Wireless Mobile Networks", IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, vol. 7, no. 7, pp.2705-2713, July 2008.

[4] Osama Faridoon, Abdul Ghafoor, " Security Protocol for NFC Enabled Mobile Devices Used in Financial Applications", NUST Journal of Engineering Sciences, Vol 9, No 2, pp 35-41, 2016.

[5] Huda Ubaya, " Design of Prototype Payment Application System with Near Field Communication (NFC) Technology based on Android ", Computer Engineering and Applications Vol. 1, No. 1, pp.1-12, June 2012.

[6] Ahmed H. Ali, Reham Abdellatif Abouhogail, Ibrahim F. Tarrad and Mohamed I. Youssef, "A new design of Mobile Payment system based on NFC Technology", International Journal of Engineering & Technology IJET-IJENS Vol.17 No.03, pp.7-18, June 2017.

[7] Mohamad Badra, Rouba Borghol Badra, "A lightweight security protocol for NFC-based mobile payments", The 7th International Conference on Ambient Systems, Networks and Technologies, (ANT 2016), Procedia computer Science 83, pp. 705-711, 2016.

[8] S.S. Ahamad, S.K. Udgata, and M. Nair, "A Secure Lightweight and Scalable Mobile Payment Framework", International Conference on Frontiers of Intelligent Computing, Theory and Applications; Vol. 247, pp. 545-553, 2013.

[9] You-Han Tung, Wen-Shenq Juang, " Secure and Efficient Mutual Authentication Scheme for NFC Mobile Devices ", Journal of Electronic Science and Technology, Vol.15, No.3, pp. 240-245, 25 Sept. 2017.

[10] Mustafa Al-Fayoumi, Shadi Nashwan, "Performance Analysis of SAP-NFC Protocol", International Journal of Communication Networks and Information Security (IJCNIS), Vol. 10, No. 1, pp. 125-130, April 2018.

[11] Shadi Nashwan, "Secure Authentication Protocol for NFC Mobile Payment Systems ", IJCSNS International Journal of Computer Science and Network Security, VOL.17, No.8, pp. 256-263, August 2017.

[12] C. Thammarat, R. Chokngamwong, and C. Techapanupreeda, " A secure lightweight protocol for NFC communications with mutual authentication based on limited-use of session keys," in Proc. of Intl. Conf. on Information Networking, pp. 133-138, 2015.

[13] U. B. Ceipidor, C. M. Medaglia, S. Sposato, and A. Moroni, " KerNeeS: A protocol for mutual authentication between NFC phones and POS terminals for secure payment transactions" in Proc. of the 9th Intl. ISC Conf. on Information Security and Cryptology, pp. 115-120, 2012.

[14] S. Kungpisdan and S. Metheekul, "A Secure Offline Key Generation with Protection Against Key Compromise" Proceedings of the 13th World Multi-conference on Systemics, Cybernetics, and Informatics, Orlando, USA, 2009.

[15] Ahamad SS, Udgata SK, and Nair M., "A Secure Lightweight and Scalable Mobile Payment Framework", International Conference on Frontiers of Intelligent Computing: Theory and Applications, p. 545-553, 2013.

[16] Chalee Thammarat , Werasak Kurutach," A lightweight and secure NFC-base mobile payment protocol ensuring fair exchange based on a hybrid encryption algorithm with formal verification", International Journal of Communication Systems, June 2019, https://doi.org/10.1002/dac.3991.

[17] M. Burrows, M. Abadi and R. Needham, "A Logic of Authentication", ACM Transactions on Computer Systems, Vol. 8, No. 1, pp.18-36, 1990.