Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.022065
Article

Fully Authentication Services Scheme for NFC Mobile Payment Systems

Munefah Alshammari* and Shadi Nashwan

Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka, 42421, Saudi Arabia
*Corresponding Author: Munefah Alshammari. Email: 401205813@ju.edu.sa
Received: 26 July 2021; Accepted: 27 August 2021

Abstract: One commonly used wireless communication technology is Near-Field Communication (NFC). Smartphones that support this technology are used in contactless payment systems as identification devices to emulate credit cards. This technology has essentially focused on the quality of communication services and has somewhat disregarded security services. Communication messages between smartphones, the point of sale (POS), and service providers are susceptible to attack due to existing weaknesses, including that an adversary can access, block and modify the transmitted messages to achieve illegal goals. Therefore, there have been many research proposals in regards to authentication schemes for NFC communications in order to prevent various types of attacks. However, the proposed schemes remain inadequate to secure payment transactions in such systems. In this paper, we propose a fully authentication services scheme for NFC mobile payment systems in order to support a high security level. The proposed scheme has security services, such as a full authentication process, perfect forward secrecy, and simultaneous anonymity of the smartphone and POS. These security services have been validated using the BAN logic model and an automatic cryptographic protocol verifier (ProVerif) tool. A security analysis has clarified that the proposed scheme can prevent various types attacks. A comparison with recent authentication schemes demonstrates that the proposed scheme has an appropriate cost in different sides such as computation, communication and storage space. Therefore, the proposed scheme not only has appealing security features, but can also clearly be utilized in mobile payment systems.

Keywords: Near field communication; mutual authentication; anonymity service; BAN logic model; proVerif tool

1 Introduction

Near-Field Communication (NFC) is a wireless technology used to facilitate and speed up data transfer with a short range of within ten centimeters and 106–424 Kbps [1–3]. This technology has been developed based on radio frequency identification (RFID) technology [4–7]. One of the widely used systems developed based on NFC technology is the contactless payment system using a smartphone-called the mobile payment system [8–13]. Therefore, the world's largest smartphone and point of sale (POS) manufacturers have recently become supportive of NFC technology in all their productions, which are called NFC mobile and NFC POS, respectively.

In the mobile payment system, the process of payment can be summarized into the following steps [14–18]: the user places his/her NFC mobile within the range of the intended NFC POS in order to transmit the payment transaction request message; the NFC POS retransmits the transaction to the authentication center (AuC) of the payment serving provider (PSP); the AuC validates the POS NFC and NFC mobile; the AuC sends the transaction payment response message to the NFC POS; the NFC mobile is validated by the NFC POS; and the NFC mobile receives the transaction payment response message from the NFC POS. Upon receiving the response message, the NFC POS is then validated by the NFC mobile in order to complete the transaction.

NFC technology has essentially focused on the quality of communication services, and has somewhat disregarded security services. Additionally, all the messages from the transaction payment process between the NFC mobile, NFC POS, and AuC are susceptible to attack due to existing weaknesses [15,19–27]. An unauthorized party could access the transaction messages in order to collect secret data from the user's bank account. An unauthorized party could also block the transaction in order to prevent the delivery of payment services. In the same context, an unauthorized party could change the transaction messages in order to forward the incorrect payment transaction order.

Numerous types of attacks have recently been found that could exploit existing weaknesses, such as desynchronization, impersonation, stolen verifier table, replay, tracking, insider, spoofing, man-in-the-middle, and password guessing attacks [28–31]. Therefore, mobile payment systems require significant improvement in order to support appropriate security services, while at the same time being reasonable for use.

The authentication scheme is considered to be an optimal solution to improve such a system; researchers of mobile payment systems have recently proposed several authentication schemes. In 2012, Ceipidor et al. [32] proposed a scheme for a mutual authentication between NFC phones and NFC POS terminals for secure payment transactions. This protocol was based on the asymmetric method in order to conduct mutual authentication among NFC devices. Despite this, the protocol fulfilled security services such as confidentiality and mutual authentication, but it is susceptible to desynchronization attacks and cannot resist brute force attacks [14]. In 2015, Thammarat et al. [33] proposed a secure, lightweight protocol for NFC communications with mutual authentication, based on the limited-use of session keys. They claimed that their protocol could achieve some security aspects such as the forward/backward secrecy service, NFC mobile anonymity, and could defeat desynchronization attacks [14,15]. In 2017, Tung et al. [34] proposed a secure mutual authentication scheme for NFC mobile devices based on a set of hash functions. In this protocol, mutual authentication was partially satisfied, forward/backward secrecy was not achieved, it lacked anonymity in security services, and it could not defeat tracking attacks. In 2017, Nashwan [15] proposed the secure authentication protocol for NFC mobile payment systems. This protocol aimed to identify most of the security problems in Near-Field Communication (NFC) in order to achieve the highest levels of security. However, this protocol cannot fully support security services such as anonymity and the forward/backward secrecy services. In 2019, Abouhogail et al. [1] proposed an advanced authentication protocol for mobile applications using NFC technology in order to satisfy mutual authentication and to resist denial of services attacks. However, this protocol cannot support anonymity, forward/backward secrecy services or prevent desynchronization attacks.

Therefore, in this paper, we proposed an authentication scheme for NFC mobile payment systems in order to resolve security problems that were observed above. The major contributions of this paper can be summarized as follows: the proposed authentication scheme for mobile payment systems is discussed; security verification using Burrows et al. [35] logic and an automatic cryptographic protocol verifier

(ProVerif) tool [36–38] is used to verify the security services; comparative security analysis shows how the proposed scheme can fully support mutual authentication, full perfect forward security and full anonymity services and can resist all types of attacks; and a comparative performance analysis shows the proposed scheme's applicability.

This paper prepared as follows. In Section 2, we present our proposed authentication scheme. Security validation using BAN logic model and a ProVerif tool to verify the security features is performed in Section 3A. Comparative security analysis with recent authentication schemes for NFC mobile payment system is discussed in Section 4A. Performance analysis is presented in Section 5. Finally, a conclusion is given in Section 6.

2 Proposed Authentication Scheme

The proposed scheme consists of three entities: the NFC mobile, NFC POS and AuC. This scheme uses a set of pseudonym identities, symmetric cryptography functions and hash functions to securely exchange the authentication messages. The main notation of the proposed scheme is listed in Tab. 1.

2.1 Notation

Table 1: Notation of the proposed scheme

Notation	Description
Xi	NFC Mobile
IDi	User Identity (according to credit card)
PWi	User password (according to credit card)
Si	User security code (according to credit card)
AuC	Autdentication center of PSP.
Sk	Session key generated by AuC for specific Xi and Yj
x, y	The secret keys of AuC
TAuCi, TAuCj	Timestamps of tde AuC side
r1, r2, r3, r4, r5, r6, r9, r10	Random numbers generated by tde AuC
r7	Random number generated by Xi
Ti0, Ti1	Timestamps in tde user side
XCi	Session number update parameter
XIDi	Pseudonym identity in user side
XIDi0	Prefix user identity
XIDi1	Suffix user identity
Yj	NFC POS
IDj	Identity of POS
PWj	Password of POS
Sj	Security code of POS
YIDj	Pseudonym identity in POS side

(Continued)

Table 1 (continued)

Notation	Description
YIDj0	Prefix POS identity
YIDj1	Suffix POS identity
r8	Random numbers generated by Yj
Tj0, Tj1	Timestamps in tde POS side
EK,DK	Cryptography functions using key K
h	One-way hash function
Φ	Empty value
⊕	Exclusive-OR operation
{.}	Transmitted message

2.2 Phases of the Proposed Scheme

This section describes our proposed authentication scheme, which contains seven phases, namely, the NFC mobile registration phase, NFC POS registration phase, mobile log-in authentication phase, POS log-in authentication phase, authentication phase, user password change phase, and NFC POS password change phase.

2.2.1 NFC Mobile Registration Phase

Step 1: As shown in Fig. 1, the user of the NFC mobile device (Xi) inserts the credit identity number (IDi), selects password (PWi), and inputs the credit card code (Si) to the Xi according to the PSP specifications. Then, Xi generates ri, computes the Ci = h (ri ‖ PWi ‖ Si), and transmits a registration request message {M1: IDi and Ci} to AuC via a private channel.

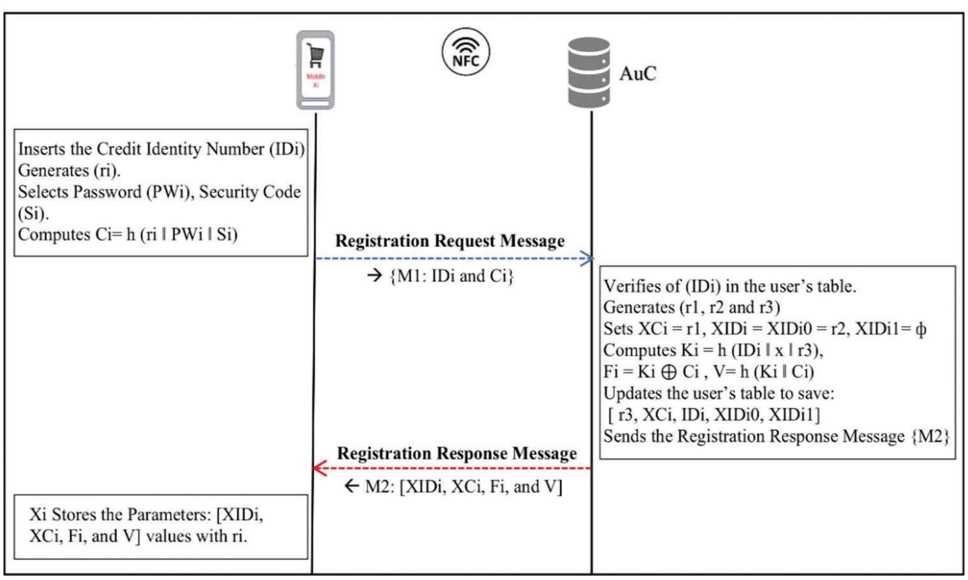


Figure 1: NFC mobile registration phase

Step2: In response to the Xi demand, the AuC node validates the presence of the identity (IDi) in the user's table, which includes the data from all the users that have already registered. If it exists, the AuC then refuses the registration request message {M1} and requests the Xi to re-enter a correct (IDi). Otherwise, the AuC generates three random numbers (r1, r2, and r3), then sets XCi = r1, XIDi = XIDi0 = r2, XIDi1 = ϕ. After that, the AuC computes Ki = h (IDi ‖ x ‖ r3), Fi = Ki ⊕ Ci and V = h (Ki ‖ Ci). Then, the AuC updates the user's table to save [r3, XCi, IDi, XIDi0, XIDi1] and sends the registration response message {M2} to Xi which includes [XIDi, XCi, Fi, and V].

Step 3: Upon receiving {M1} from AuC, Xi stores [XIDi, XCi, Fi, and V] values with ri.

2.2.2 NFC POS Registration Phase

Step 1: As shown in Fig. 2, the owner of the NFC POS device (Yj) inserts its POS identity number (IDj), selects the password (PWj), and inputs the POS secret code (Sj) according to the PSP specifications, The Yi generates rj, computes Cj = h (rj ‖ PWj ‖ Sj), and transmits a registration request message {M1: IDj and Cj} to AuC through a private channel.

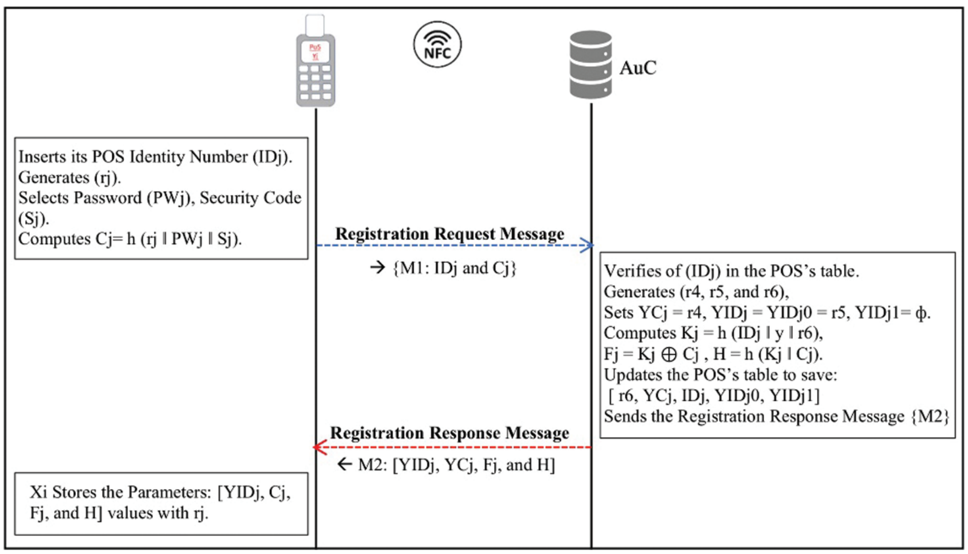


Figure 2: NFC POS registration phase

Step 2: In response to the Yj demand, the AuC node validates the presence of the (IDj) in the POS's table, which includes the data from all the POS deceives that have already been registered. If it exists, the AuC then refuses the registration request message {M1} and requests the Yi to re-enter a correct (IDj). Otherwise, the AuC generates three random numbers (r4, r5, and r6), then sets YCj = r4, YIDj = YIDj0 = r5, YIDj1 = ϕ. After that, the AuC computes Kj = h (IDj ‖ y ‖ r6), Fj = Kj ⊕ Cj and H = h (Kj ‖ Cj). Then, the AuC updates the POS's table to save [r6, YCj, IDj, YIDj0, YIDj1] authentication parameters and sends the registration response message {M2} to Yj, which includes [YIDj, YCj, Fj, and H].

Step 3: Upon receiving {M2} from AuC, Yj stores [YIDj, Cj, Fj, and H] values with rj.

2.2.3 Mobile Log-in Authentication Phase

Fig. 3 illustrates the mobile log-in authentication phase. When a user wants to put his/her NFC mobile (Xi) near the NFC POS in order to send the payment transaction request message, Xi needs to authenticate the user. The process of authentication can be described between the user and his/her Xi device as follows.

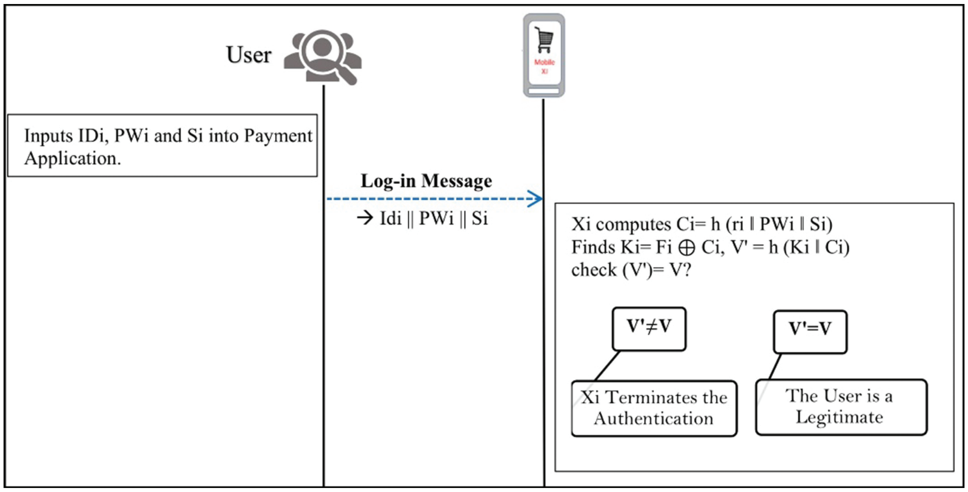


Figure 3: Mobile log-in authentication phase

Step 1: The user inserts PWi and Si into the payment application. The Xi fetches ri, computes Ci = h (ri ǁ PWi ǁ Si), then finds Ki = Fi ⊕ Ci and V' = h (Ki ǁ Ci)).

Step 2: The Xi verifies the authentication request by comparing the values of the computed (V') and the stored V. If they are not equivalent, Xi finishes the authentication. Otherwise, Xi approves that the user is legitimate.

2.2.4 POS Log-in Authentication Phase

Fig. 4 illustrates the POS log-in authentication phase: when the seller wants to activate his/her NFC POS (Yj) to receive the user payment request message, the Yj needs to authenticate the seller. The process of authentication can be described between the seller and his/her Yj device as follows.

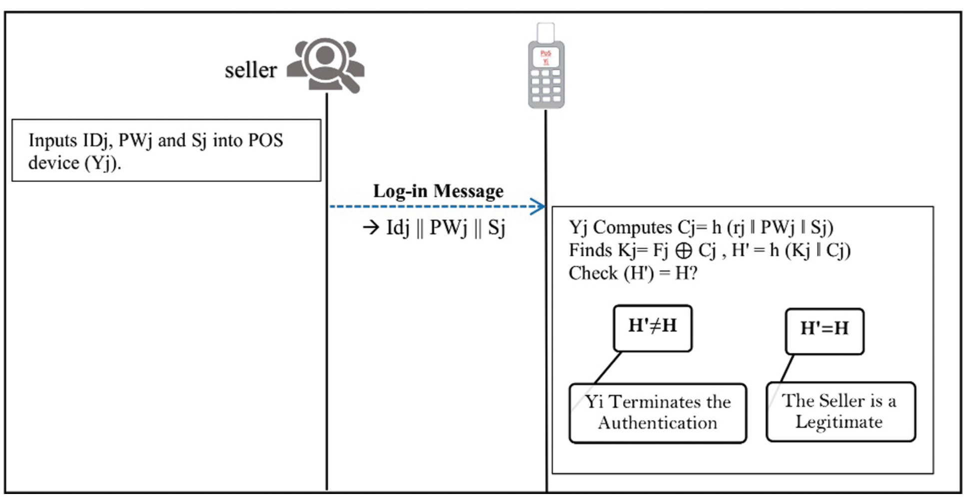


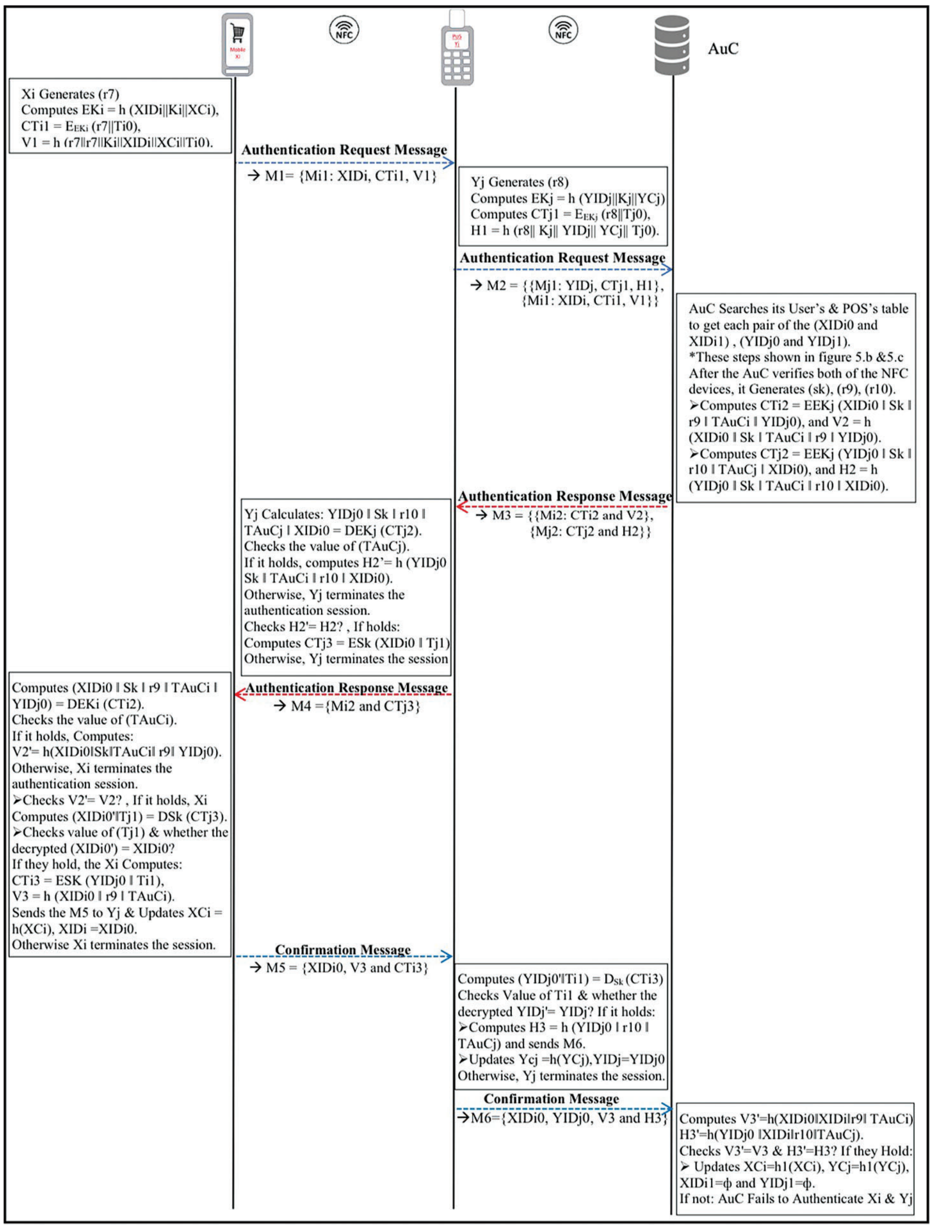
Figure 4: POS log-in authentication phase

Step 1: The seller inserts PWj and Sj into the POS device (Yj). The Yj fetches rj, computes Cj = h (rj ǁ PWj ǁ Sj), and finds the Kj = Fj ⊕ Cj and H' = h(Kj ǁ Cj).

Step 2: The Yj verifies the authentication request by comparing the values of the computed (H') and the stored value H. If they are not equivalent, Yj finishes the authentication. Otherwise, Yj approves that the seller is legitimate.

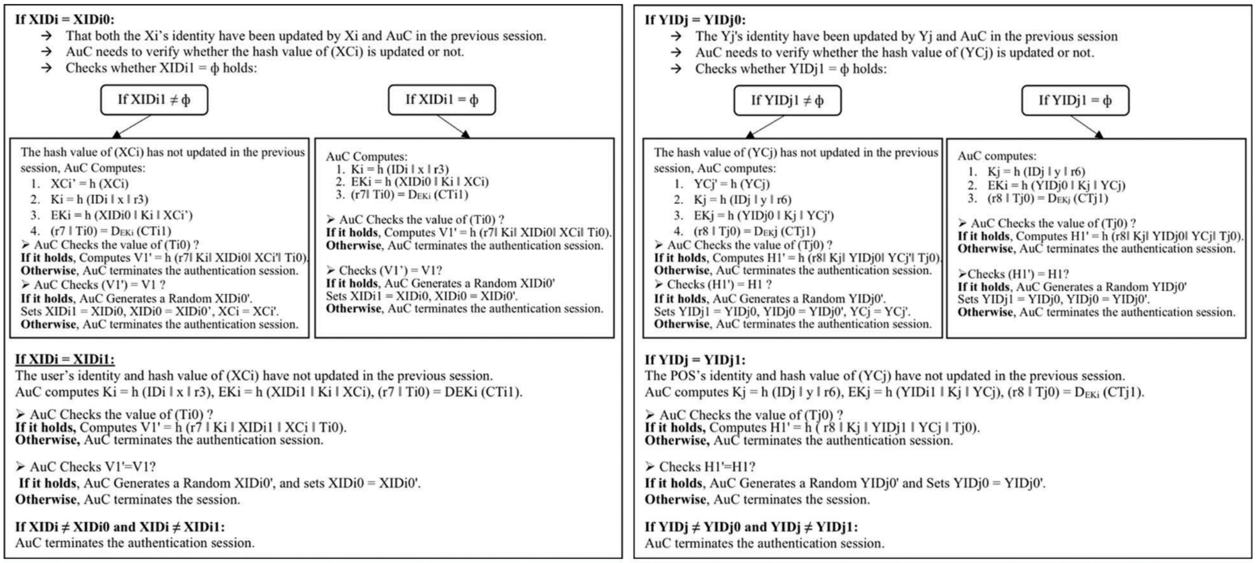
2.2.5 Authentication Phase

Figs. 5a–5c illustrate the authentication phase. The NFC mobile (Xi) can execute the payment transaction through a specific NFC POS device (Yj) by achieving mutual authentication with the AuC and Yj. It should be noted that this phase is executed after both the log-in authentication phases of the Xi and Yj have been completed. Thus, the following steps summarize the authentication process.



(a)

Figure 5: Continued



(b) (c)

Figure 5: (a): Authentication phase. (b): Authentication phase: (Step 3). (c): Authentication phase (Step 4)

Step 1: When the Xi is placed near the Yj, the Xi generates a random number r7, then computes EKi = h (XIDi ‖ Ki ‖ XCi), CTi1 = EEKi (r7 ‖ Ti0), and V1 = h (r7 ‖ Ki ‖ XIDi ‖ XCi ‖ Ti0), where Ti0 is the timestamp from the user side. Finally, Xi transmits the authentication request message M1 = {Mi1: XIDi, CTi1, V1} to Yj through an NFC communication channel (public channel).

Step 2: With receiving M1, the Yj generates r8 randomly, computes EKj = h (YIDj ‖ Kj ‖ YCj), CTj1 = EEKj (r8 ‖ Tj0), and H1 = h (r8 ‖ Kj ‖ YIDj ‖ YCj ‖ Tj0), where Tj0 is the timestamp from the POS side. Finally, Yj transmits the payment authentication request message M2 = {{Mj1: YIDj, CTj1, H1}, and {Mi1: XIDi, CTi1, V1}} to AuC through a public channel.

Step 3: As shown in Fig. 5b, after receiving the (M2), AuC first searches its user's table to find a pair of the (XIDi0 and XIDi1) according to the XIDi value that has been received through (M1), then operates as follows in order to authenticate the Xi.

If XIDi = XIDi0, it means that both identities have been updated by Xi and AuC in the prior authentication session. Then, AuC wants to validate whether the hash value of (XCi) is updated or not. Therefore, the AuC checks whether XIDi1 = ϕ.

If it does not hold (i.e., XIDi1 ≠ ϕ), it means that the hash value of (XCi) has not been updated in the previous session (as in step 9). Thus, the AuC computes XCi' = h (XCi), Ki = h (IDi ‖ x ‖ r3), EKi = h (XIDi0 ‖ Ki ‖ XCi'), and (r7 ‖ Ti0) = DEKi (CTi1), then examines the value of (Ti0). If it holds, it computes V1' = h (r7 ‖ Ki ‖ XIDi0 ‖ XCi' ‖ Ti0). Otherwise, AuC terminates the authentication session. Then, AuC checks whether the computed (V1') matches with the V1 that has been received. If it holds, AuC generates a random XIDi0' and sets XIDi1 = XIDi0, XIDi0 = XIDi0', and XCi = XCi'. Otherwise, AuC terminates the authentication session.

In contrast, in XIDi1 = ϕ, the AuC computes Ki = h (IDi ‖ x ‖ r3), EKi = h (XIDi0 ‖ Ki ‖ XCi), and (r7 ‖ Ti0) = DEKi (CT1), and checks the value of (Ti0). If it holds, it computes V1' = h (r7 ‖ Ki ‖ XIDi0 ‖ XCi ‖ Ti0). Otherwise, AuC terminates the authentication session. Then, AuC examines whether the computed (V1') equals with the V1. If it holds, AuC generates XIDi0' randomly, and sets XIDi1 = XIDi0 and XIDi0 = XIDi0'. Otherwise, AuC terminates the authentication session.

End if

End if

If XIDi = XIDi1, it implies that the identity of the user and hash value of (XCi) have not been updated in the prior session. Thus, the AuC calculates Ki = h (IDi ‖ x ‖ r3), EKi = h (XIDi1 ‖ Ki ‖ XCi), and (r7 ‖ Ti0) = DEKi (CTi1), then examines the value of (Ti0). If it holds, it computes V1' = h (r7 ‖ Ki ‖ XIDi1 ‖ XCi ‖ Ti0). Otherwise, AuC finishes the authentication session. Then, AuC checks whether V1' matches the received V1. If it holds, AuC generates a random XIDi0', and makes XIDi0 = XIDi0'. Otherwise, AuC finishes the session.

End if

If XIDi ≠ XIDi0 and XIDi ≠ XIDi1, AuC terminates the authentication session.

End if

Step 4: According to Fig. 5c, as in the previous step, AuC determines Yj's identity. The AuC searches its POS's table to find a pair of (YIDj0 and YIDj1), according to the YIDj value, that has been received though (Mj1), and then executes the following steps to authenticate the Yj:

If YIDj = YIDj0, it means that both the Yj's identities have been updated by Yj and AuC in the prior session. Then, AuC wants to validate whether the hash value of (YCj) is updated or not. Therefore, the AuC checks whether YIDj1 = ϕ.

If it does not hold (i.e., YIDj1 ≠ ϕ), it implies that the hash value (YCj) has not updated in the prior session (as in step 9). Thus, AuC computes YCj' = h (YCj), Kj = h (IDj ‖ y ‖ r6), EKj = h (YIDj0 ‖ Kj ‖ YCj'), and (r8 ‖ Tj0) = DEKj (CTj1) and checks the value of (Tj0). If it holds, it computes H1' = h (r8 ‖ Kj ‖ YIDj0 ‖ YCj' ‖ Tj0). Otherwise, AuC terminates the authentication session. Then, AuC checks whether the computed (H1') equals the H1 that has been received. If it holds, AuC generates YIDj0' randomly, and makes YIDj1 = YIDj0, YIDj0 = YIDj0', YCj = YCj'. Otherwise, AuC terminates the authentication session.

In contrast, in YIDj1 = ϕ, the AuC computes Kj = h (IDj ‖ y ‖ r3), EKi = h (YIDj0 ‖ Kj ‖ YCj), and (r8 ‖ Tj0) = DEKj(CTj1) and checks the value of (Tj0). If it holds, it computes H1' = h (r8 ‖ Kj ‖ YIDj0 ‖ YCj ‖ Tj0). Otherwise, AuC terminates the authentication session. Then, AuC checks whether the computed (H1') equals the received H1. If it holds, AuC generates YIDj0' randomly, and makes YIDj1 = YIDj0 and YIDj0 = YIDj0'. Otherwise, AuC terminates the authentication session.

End if

End if

If YIDj = YIDj1, it implies that the identity of the POS and hash value of (YCj) have not updated in the prior session. Then, the AuC computes Kj = h (IDj ‖ y ‖ r6), EKj = h (YIDi1 ‖ Kj ‖ YCj), and (r8 ‖ Tj0) = DEKj (CTj1), and checks the value of (Tj0). If it holds, it computes H1' = h (r8 ‖ Kj ‖ YIDj1 ‖ YCj ‖ Tj0). Otherwise, AuC terminates the authentication session. Then, AuC checks whether H1' matches the received H1. If it holds, AuC generates YIDj0' randomly, and makes YIDj0 = YIDj0'. Otherwise, AuC finishes the session.

End if

If YIDj ≠ YIDj0 and YIDj ≠ YIDj1, AuC finishes the authentication session.

End if

It should be noted that, after this step, both the NFC mobile and NFC POS are considered to be either legitimate parties or not for the AuC to complete the authentication process.

Step 5: The AuC generates the (Sk) and (r9) randomly, then the AuC computes CTi2 = EEKi (XIDi0 ‖ Sk ‖ r9 ‖ TAuCi ‖ YIDj0) and V2 = h (XIDi0 ‖ Sk ‖ TAuCi ‖ r9 ‖ YIDj0), wherein (TAuCi) is a timestamp of the AUC. After that, it generates the (r10) randomly, and computes CTj2 = EEKj (YIDj0 ‖ Sk ‖ r10 ‖ TAuCj ‖ XIDi0) and H2 = h (YIDj0 ‖ Sk ‖ TAuCi ‖ r10 ‖ XIDi0), wherein (TAuCj) is a timestamp of the AUC. Finally, AuC transmits the payment authentication response message M3 = {{Mi2: CTi2 and V2}, {Mj2: CTj2 and H2}} to Yj. It should be noted that the value of sk represents the shared key between the Xi and Yj.

Step 6: Upon receiving (M3), Yj calculates YIDj0 ‖ Sk ‖ r10 ‖ TAuCj ‖ XIDi0 = DEKj (CTj2) and checks the value of (TAuCj). If it holds, it computes H2' = h (YIDj0 ‖ Sk ‖ TAuCi ‖ r10 ‖ XIDi0). Otherwise, Yj terminates the authentication session. Then, it checks whether the computed (H2') matches the received H2. If it holds, it computes CTj3 = ESk (XIDi0 ‖ Tj1), wherein (Tj1) is a timestamp of NFCPOS, and sends the payment authentication response message M4 = {Mi2 and CTj3} to Xi. Otherwise, Yj terminates the session. It should be noted that, after this step, the AuC is considered to be a legitimate party for the NFC POS.

Step 7: After receiving (M4), the Xi computes (XIDi0 ‖ Sk ‖ r9 ‖ TAuCi ‖ YIDj0) = DEKi (CTi2) and checks the value of (TAuCi). If it holds, it computes V2' = h (XIDi0 ‖ Sk ‖ TAuCi ‖ r9 ‖ YIDj0). Otherwise, Xi terminates the authentication session. Then, it checks whether the computed (V2') matches the received V2. If it holds, Xi computes (XIDi0' ‖ Tj1) = DSk (CTj3) and checks the value of (Tj1) and whether the decrypted (XIDi0') matches the XIDi0. If they hold, the Xi computes CTi3 = ESK (YIDj0 ‖ Ti1), wherein (Ti1) is a timestamp of the NFCMobile, V3 = h (XIDi0 ‖ r9 ‖ TAuCi), and sends the confirmation message M5 = {XIDi0, V3 and CTi3} to Yj. Then, it updates XCi = h (XCi) and XIDi = XIDi0. Otherwise, Xi terminates the session. It should be noted that, after this step, both AuC and POS are considered to be legitimate parties for the NFC mobile.

Step 8: Upon receiving the (M5) from Xi, the Yj computes (YIDj0' ‖ Ti1) = DSk(CTi3) and checks the value Ti1 and whether the decrypted YIDj' matches the YIDj. If it holds, it computes H3 = h (YIDj0 ‖ r10 ‖ TAuCj) and sends the confirmation message M6 = {XIDi0, YIDj0, V3, and H3} to AuC. Then, it updates YCj = h (YCj) and YIDj = YIDj0. Otherwise, Yi terminates the session.

Step 9: After receiving (M6) from Yi, the AuC computes V3' = h (XIDi0 ‖ r9 ‖ TAuCi), computes H3' = h (YIDj0 ‖ r10 ‖ TAuCj) and checks whether V3' and H3' match the received V3 and H3, respectively. If they hold, AuC updates XCi = h1(XCi), YCj = h1(YCj) and XIDi1 = ф and YIDj1 = ф. Otherwise, AuC fails to authenticate Xi and Yj.

2.2.6 User Password Change Phase

Fig. 6 illustrates the user password change phase: where a user of Xi needs to change the password. Therefore, he/she requires to execute the following steps:

Step 1: The user inserts PWi and Si to the NFC mobile (Xi). Then, Xi fetches the stored ri, computes Ci = h (ri ‖ PWi ‖ Si), finds the Ki = Fi ⊕ Ci, and V' = h (Ki ‖ Ci)), then verifies whether the computed (V') and the stored (V) are equivalent. If not, Xi cannot authenticate the user, and rejects the request of the password change. Otherwise, the user inserts an updated password PWi*.

Step 2: Xi computes Ci* = h (ri ‖ PWi* ‖ Si), Fi* = Ki ⊕ Ci ⊕ Ci* and V* = h (Ki ‖ Ci*)

Step 3: Finally, Xi replaces computed Fi* and V* with Fi and V, respectively.

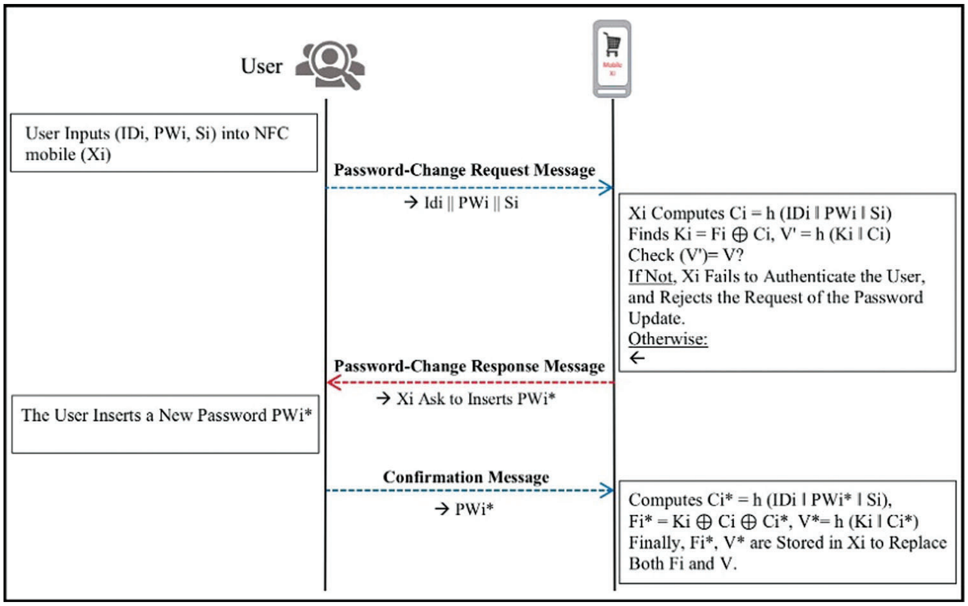


Figure 6: NFC mobile password change phase

2.2.7 NFC POS Password Change Phase

Fig. 7 illustrates the NFC POS password change phase, where a seller of Yj needs to change the password. Therefore, he/she requires to execute the following steps:

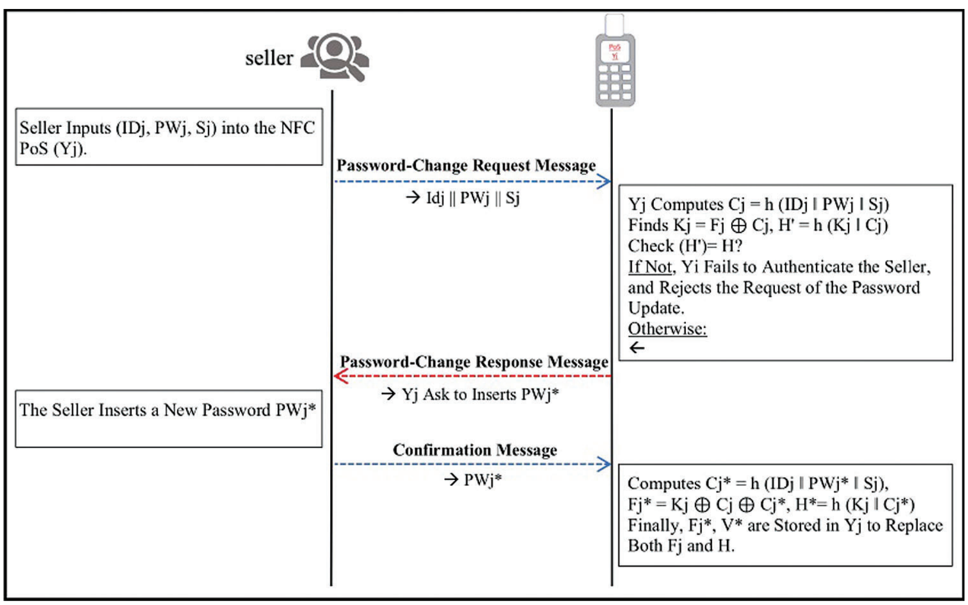


Figure 7: NFC POS password change phase

Step 1: The seller inserts PWj and Sj to the NFC POS (Yj). Then, Yj fetches the stored rj, computes Cj = h (rj ∥ PWj ∥ Sj), then finds the Kj = Fj ⊕ Cj, and H' = h (Kj ∥ Cj)), and verifies whether the computed (H') and the stored (H) are equivalent. If not, Yj cannot authenticate the seller, and rejects the request of the password change. Otherwise, the seller inserts an updated password PWj*.

Step 2: Yj computes Cj* = h (IDj ‖ PWj* ‖ Sj), Fj* = Kj ⊕ Cj ⊕ Cj* and H* = h (Kj ‖ Cj*)

Step 3: Finally, Yj replaces computed Fj* and H* with Fj and H, respectively.

3 Formal Security Verification

We can observe from the above description that the phases of the proposed scheme are either not used frequently or are executed through a secure communication channel, except for the authentication phase. Therefore, we will concentrate on the soundness of the authentication phase by verifying it based on the BAN logic model and a ProVerif tool in the following subsections.

3.1 Security Verification Using BAN Logic

In this section, we apply the BAN logic model [16,35] to examine the freshness and originality of the authentication messages exchanged between the NFC mobile, NFC POS and AuC in the authentication phase. To apply the BAN logic model, the basic notation and believing rules that we will used are listed in the Tabs. 2 and 3, respectively.

Table 2: Main notation

Notation	Description
P, T	Statements
F, Q	Communication principles
K	Shared key
$F \mid\equiv P$	F can consider P is true.
$F \triangleleft P$	F sees P.
$F \mid\sim P$	F says P, then F can send a message contains P.
$F \Longrightarrow P$	F jurisdiction over P.
#(P)	P is a fresh.
(P, T)	P or T is a part of tde formula (P, T).
$\langle P \rangle T$	P is combined witd tde T.
{P} K	P is encrypted by K.
$F \stackrel{K}{\leftrightarrow} Q$	F and Q communicate witd each otder using K.
$F \stackrel{P}{\Leftrightarrow} Q$	A secret P is known only for F and Q.
SK	A session key.

The NFC mobile (Xi), NFC POS (Yj), and the AuC are considered to be the principles that are mainly involved in the verification process for our proposed authentication scheme. During the authentication phase, EKi, EKj and Sk are the cipher keys used to symmetrically cipher authentication messages, while groups of unrepeatable timestamps (Ti0, Ti1, Tj0, Tj1, TAuCi, and TAuCj) and random numbers (r7, r8, r9, and 10) are employed to ensure the freshness of the authentication session. The listed of our goals, the idealized form and the verification assumptions for the authentication phase are shown in Tabs. 4–6, respectively.

Table 3: The believing rules

Rules	Formulas
Message meaning rule (R1)	$\frac{F\vert\equiv F \overset{K}{\leftrightarrow} Q,\ F \triangleleft \langle P \rangle_K}{F\vert\equiv Q\vert\sim P}$
Freshness conjuncatenation rule (R2)	$\frac{F\vert\equiv \#(P)}{F\vert\equiv \#(P, T)}$
Belief rule (R3)	$\frac{F\vert\equiv P,\ F\vert\equiv T}{F\vert\equiv (P,\ T)}$
Nonce verification rule (R4)	$\frac{F\vert\equiv \#(P),\ F\vert\equiv Q\vert\sim P}{F\vert\equiv Q\vert\equiv P}$
Jurisdiction rule (R5)	$\frac{F\vert\equiv Q\Rightarrow P,\ F\vert\equiv Q\vert\equiv P}{F\vert\equiv P}$
Session key rule (R6)	$\frac{F\vert\equiv \#(P),\ F\vert\equiv Q\vert\equiv P}{F\vert\equiv F \overset{K}{\leftrightarrow} Q}$

Table 4: The main goals of Authentication phase

Goals	Description
G1	$AuC \vert\equiv AuC \overset{SK}{\leftrightarrow} Xi$
G2	$AuC \vert\equiv Xi \vert\equiv AuC \overset{SK}{\leftrightarrow} Xi$
G3	$Yj \vert\equiv Yj \overset{SK}{\leftrightarrow} AuC.$
G4	$Yj \vert\equiv AuC \vert\equiv Yj \overset{SK}{\leftrightarrow} AuC$
G5	$AuC \vert\equiv AuC \overset{SK}{\leftrightarrow} Yj$
G6	$AuC \vert\equiv Yj \vert\equiv AuC \overset{SK}{\leftrightarrow} Yj$
G7	$Xi \vert\equiv Xi \overset{SK}{\leftrightarrow} AuC$
G8	$Xi \vert\equiv AuC \vert\equiv Xi \overset{SK}{\leftrightarrow} AuC$
G9	$Xi \vert\equiv Xi \overset{SK}{\leftrightarrow} Yj$
G10	$Xi \vert\equiv Yj \vert\equiv Xi \overset{SK}{\leftrightarrow} Yj$
G11	$Yj \vert\equiv Yj \overset{SK}{\leftrightarrow} Xi$
G12	$Yj \vert\equiv Xi \vert\equiv Yj \overset{SK}{\leftrightarrow} Xi$

Table 5: Idealized form of the authentication phase messages

Authentication messages	Idealized form
M2	(YIDj, CTj0, H1: 〈(Tj0, r8)〉 EKj, XIDi, CTi0, V1: 〈(Ti0, r7)〉 EKi).
M3	(CTj1, H2: 〈(Sk, TAuCj, r10)〉 EKj, CTi1, V2: 〈(Sk, TAuCi, r9)〉 EKi).
M4	(CTi1, V2: 〈(Sk, TAuCi, r9)〉 EKi), CTj3: 〈XIDi0’,Tj1〉 Sk).
M5	(V3, CTi3: 〈YIDj0’,Ti1〉 Sk).

Table 6: Initial verification assumptions of the authentication phase

Assumptions	Description
As1	Xi |≡ # (r9, TAuCi, Sk)
As2	Xi |≡ AuC ⇒ (r9, TAuCi, Sk)
As3	AuC |≡ # (r7, r8, Ti0, Tj0)
As4	AuC |≡ Xi ⇒ (r7, Ti0)
As5	AuC |≡ Yj ⇒ (r8, Tj0)
As6	Yj |≡ # (r10, TAuCj, Sk)
As7	Yj |≡ AuC ⇒ (r10, TAuCj, Sk)
As8	Xi |≡ # (Tj1)
As9	Xi |≡ Yj ⇒ (Tj1)
As10	Yj |≡ # (Ti1)
As11	Yj |≡ i ⇒ (Ti1)
As12	Xi |≡ Xi $\overset{EKi}{\leftrightarrow}$ AuC.
As13	AuC |≡ AuC $\overset{EKi}{\leftrightarrow}$ Xi
As14	AuC |≡ AuC $\overset{EKj}{\leftrightarrow}$ Yj
As15	Yj |≡ Yj $\overset{EKj}{\leftrightarrow}$ AuC
As16	Xi |≡ Xi $\overset{Sk}{\leftrightarrow}$ Yi.
As17	Yj |≡ Yj $\overset{Sk}{\leftrightarrow}$ Xi

The following steps summarize the validation process of the authentication phase:

Consider the second part of M1, then (A1) can be seen as (AuC ⊲ XIDi, CTi0, V1: 〈 (Ti0, r7) 〉 EKi). Thus, using the A1, As13, R3, and R1, then (A2) can be acquired as (AuC|≡ Xi |~〈 (Ti0, r7) 〉 EKi). Next, using As3 and R2, (A3) can also then be obtained as (AuC|≡ # (〈 (Ti0, r7) 〉 EKi)). Then, using A2, A3 and R4, the (A4) can also be obtained as (AuC|≡ Xi|≡〈 (Ti0, r7) 〉 EKi). Therefore, from A3, A4 and R6, then (A5) can be inferred (AuC |≡ AuC $\overset{SK}{\leftrightarrow}$ Xi), which represents (**G1**). Besides, using As4, A5 and R4, then (A6) can be inferred (AuC|≡ Xi |≡ AuC $\overset{SK}{\leftrightarrow}$ Xi), which represents (**G2**) as well.

Similarly, consider the first part of M1, then (B1) can be seen as (AuC ⊲ YIDj, CTj0, H1: 〈 (Tj0, r8) 〉 EKj). Thus, using the B1, As14, R3, and R1, then (B2) can be acquired as (AuC|≡ Yj |~〈 (Tj0, r8) 〉 EKi). Next, using As3 and R2, then (B3) can also be obtained as (AuC|≡ # (〈 (Tj0, r8) 〉 EKj)). Then, using B2, B3 and R4, the (B4) can also be obtained as (AuC|≡ Yj|≡〈 (Tj0, r8) 〉 EKj). Therefore, from B3, B4 and R6, then (B5) can be inferred (AuC |≡ AuC $\overset{SK}{\leftrightarrow}$ Yj), which represents (**G5**). Besides this, using As5, B5 and R4, then (B6) can be inferred (AuC|≡ Yj |≡ AuC $\overset{SK}{\leftrightarrow}$ Yj) which represents (**G6**) as well.

Now, consider the first part of M2, then (C1) can be seen as (Yj ⊲ CTj1, H2: 〈 (Sk, TAuCj, r10) 〉EKj). Thus, from the (C1), As15, R3, and R1, then (C2) can be obtained as (Yj|≡ AuC |~〈 (Sk, TAuCj, r0) 〉EKj). Next, using As6, and the R2, the (C3) can be obtained as (Yj|≡ # (〈(Sk, TAuCj, r10)〉 EKj). Then, using (C2), (C3), and the R4, the (C4) can be acquired: (Yj|≡ AuC|≡〈 (Sk, TAuCj, r10) 〉EKj). Therefore, from (C3), (C4), and R6, the C5: (Yj|≡ Yj $\overset{SK}{\leftrightarrow}$ AuC) can be inferred and this represents (**G3**). Besides this, using As7, (C5), and the R4, then the C6: (Yj|≡AuC |≡ Yj $\overset{SK}{\leftrightarrow}$ AuC) can be inferred and this also represents (**G4**).

Similarly, consider the first part of M3, then (D1) can be seen as (Xi $\triangleleft$ CTi1, V2: $\langle$ (Sk, TAuCi, r9) $\rangle$ EKi). Thus, from the (D1), As12, R3, and R1, then (D2) can be obtained as (Xi$|\equiv$ AuC $|\sim\langle$ (Sk, TAuCi, r9) $\rangle$ EKi). Next, using As1, and the R2, the (D3) can be obtained as (Xi$|\equiv$ # ($\langle$ (Sk, TAuCi, r9) $\rangle$ EKi). Then, using (D2), (D3), and the R4, the (D4) can be acquired: (Xi$|\equiv$ AuC$|\equiv\langle$ (Sk, TAuCi, r9) $\rangle$EKi). Therefore, from (D3), (D4), and R6, the D5: (Xi $|\equiv$Xi $\overset{SK}{\leftrightarrow}$ AuC) can be inferred and this represents (**G7**). Besides this, using As2, (D5), and the R4, then the D6: (Xi$|\equiv$AuC $|\equiv$Xi $\overset{SK}{\leftrightarrow}$ AuC) can be inferred and this also represents (**G8**).

Now, consider the second part of M3, then (E1) can be seen as (Xi $\triangleleft$ CTj3: $\langle$ (XIDi', Tj1) $\rangle$ Sk). Thus, from the (E1), As16, R3, and R1, then (E2) can be obtained as (Xi$|\equiv$ Yj $|\sim\langle$(XIDi', Tj1) $\rangle$ Sk). Next, using As8, and the R2, the (E3) can be obtained as (Xi$|\equiv$ # ($\langle$(XIDi', Tj1)$\rangle$ Sk). Then, using (E2), (E3), and the R4, the (E4) can be acquired: (Xi$|\equiv$ Yj$|\equiv$ $\langle$ (XIDi', Tj1) $\rangle$ Sk). Therefore, from (E3), (E4), and R6, the E5: (Xi $|\equiv$Xi $\overset{SK}{\leftrightarrow}$ Yj) can be inferred and this represents (**G9**). Besides this, using Assumption 9, (E5), and the R4, then the E6: (Xi$|\equiv$Yj $|\equiv$Xi $\overset{SK}{\leftrightarrow}$ Yj) can be inferred and this also represents (**G10**).

Finally, consider the second part of M4, then (F1) can be seen as (Yj $\triangleleft$ CTj3: $\langle$ (YIDj', Ti1) $\rangle$ Sk). Thus, from the (F1), As17, R3, and R1, then (F2) can be obtained as (Yj$|\equiv$ Xi $|\sim\langle$(YIDj', Ti1)$\rangle$ Sk). Next, using As10, and the R2, the (F3) can be obtained as (Yj$|\equiv$ # ($\langle$(YIDj', Ti1) $\rangle$ Sk). Then, using (F2), (F3), and the R4, the (F4) can be acquired: (Yj$|\equiv$ Xi$|\equiv$ $\langle$ (YIDj', Ti1) $\rangle$ Sk). Therefore, from (F3), (F4), and R6, the F5: (Yj $|\equiv$Yj $\overset{SK}{\leftrightarrow}$ Xi) can be inferred and this represents (**G11**). Besides this, using As11, (F5), and the R4, then the F6: (Yj$|\equiv$ Xi$|\equiv$ Yj $\overset{SK}{\leftrightarrow}$ Xi) can be inferred and this also represents (**G12**).

Therefore, the main goals of the authentication phase have been successfully proven and mutual authentication can be granted between the Xi, Yj and AuC throughout this phase.

3.2 Security Verification Using ProVerif Tool

In this section, we validate our authentication scheme in terms of achieving mutual authentication and secure authentication sessions using one of the common automated verifier tools, called the ProVerif tool [16,35]. This tool is used to verify the main security features of the cryptographic protocol, such as authentication, secrecy, anonymity by supporting numerous cryptographic techniques, including symmetric/asymmetric cryptography, hash functions, and digital signatures. Besides this, the ProVerif tool assumes that the adversary can modify, eavesdrop, and delete the communication messages that are exchanged between the authentication nodes. Thus, if the proof is true, as a result of the verification process, then all possible attacks are checked and the communication messages are in a safe state. Otherwise, traces of attacks are provided.

As to verify the security of our authentication scheme, we defined a set of premises for our verification code statements, as shown in Fig. 8. The publicchMM and publicchMP are public communication channels that are used by the NFC mobile. However, the publicchPP. publicchPM, and publicchPA are used by the NFC POS and AuC 2. Besides, we declared four data types: type key for symmetric encryption, type timestamp to set the timestamps, type coins to generate random numbers, and type host to define the participants of our authentication scheme as the NFCMobile, NFCPOS, and AuC. Then, four free names, namely, sec1, sec2, sec3 and sec4, were declared to analyze the secrecy of the session key. After that, we defined eight events in order to show the start and end of the authentication processes to review mutual authentication between principles. Finally, we defined a set of queries to check if the proposed scheme could achieve the authentication and secrecy of the session key.

Fig. 9 illustrates the main functions that are defined to implement the authentication events. The h, xor, con1, con3, con4, and con5 represent the hash, exclusive-or, and concatenation functions. Besides this, the encrypt and isFresh symbols for encryption and freshness functions were used, wherein the isFresh function is used to check if the timestamp is a fresh value or not. Furthermore, we defined a set of data-type converter functions from line 44 to line 48.

```
free publicchMP, publicchMM: channel.
free publicchPM, publicchPA, publicchPP : channel.
type key.
type host.
type coins.
type timestamp.
free NFCMobile, NFCPOS, AuC: host.
free x, y: key[Private].
table NFCMoA(coins, key, bitstring, bitstring).
table NFCPoA(coins, key, bitstring, bitstring).
table NFCmoM(bitstring, key, bitstring, bitstring, coins).
table NFCPoP(bitstring, key, bitstring, bitstring, coins).
free XID: bitstring.
free YID: bitstring.
free sec1, sec2, sec3, sec4: bitstring [private].
event StartPMparam (host).
event endPMparam(host).
event StartMPparam(host).
event endMPparam(host).
event StartAPparam(host).
event endAPparam(host).
event StartPAparam(host).
event endPAparam(host).
query z: host; inj-event(endPMparam(z))==> inj-event(StartPMparam(z)).
query z: host; inj-event(endMPparam(z))==> inj-event(StartMPparam(z)).
query z: host; inj-event(endAPparam(z))==> inj-event(StartAPparam(z)).
query z: host; inj-event(endPAparam(z))==> inj-event(StartPAparam(z)).
query attacker(sec1).
query attacker(sec2).
query attacker(sec3).
query attacker(sec4).
not attacker(new XCi).
not attacker(new YCj).
```

Figure 8: The code premises

```
fun h(bitstring): bitstring.
fun xor(bitstring, bitstring): bitstring.
equation forall x: bitstring, y: bitstring; xor(xor(x, y), y) = x.
fun con2(bitstring, bitstring): bitstring.
fun con3(bitstring, bitstring, bitstring): bitstring.
fun con4(bitstring, bitstring, bitstring, bitstring): bitstring.
fun con5(bitstring, bitstring, bitstring, bitstring, bitstring): bitstring.
fun encrypt (bitstring, key): bitstring.
reduc forall m:bitstring, k:key decrypt(encrypt(m, k), k)= m.
fun isFresh(timestamp, bool): bool
reduc forall T: timestamp; isFresh(T, true) = true
otherwise forall T: timestamp; isFresh(T, false) = false.
fun keytostring(key): bitstring [data, typeConverter].
fun randtostring(coins): bitstring [data, typeConverter].
fun stringtokey(bitstring): key [data, typeConverter].
fun timestamptostring(timestamp): bitstring [data,typeConverter].
fun coinstostring(coins): bitstring [data,typeConverter].
```

Figure 9: The main functions code

The proposed authentication scheme is emulated as the concurrent execution of three separate processes in order to emulate the NFC mobile, NFC POS, and AuC. Fig. 10 illustrates the code of the NFC mobile, called the processNFCMobile process. The first part of the process represents the code of the NFC mobile log-in phase (lines 52 to 59). While the second part represents the code of the authentication phase in the NFC mobile side (line 60 to 78). The (StartPMparam) event of NFC POS is set at line 55 and the (endMPparam) event of the NFC mobile is set at line 78. In the last line, the analysis query code of the session key secrecy (Sk), through publicchMP, is set.

Fig. 11 shows the code of the NFC POS, called the processNFCPOS process. The first part of the process represents the code of the NFC POS log-in phase (lines 82 to 89), while the second part represents the code of the authentication phase on the NFC POS side (line 90 to 113). The (StartAPparam) event of AuC is set at line 85, and the (endPAparam) and (endPMparam) events of the NFC POS are set at line 112 and line 113, respectively. The analysis queries code of the of session key secrecy (Sk) through the publicchPM and publicchPA is set at line 115 and 116. Fig. 12 shows the code of the AuC, called the processAuC process. This process represents the authentication phase in the NFC POS side (lines 118 to 147). The (StartPAparam) event of NFC POS is set at line 157, and the (endAPparam) event of the AuC is set at line 184. The analysis query code of the of session key secrecy of (EKi) through the publicch2 is set at line 185.

```
let processNFCMobile =
in (publicchMM, XIDx :bitstring);
if XIDx = XIDi then
get NFCmoM(=XIDi, XCi, Fi, V, ri)in
event StartPMparam(NFCPOS);
Let Ci' = h(con3(Si, PWi , coinstostring (ri)))in
let xKi = xor(Fi, Ci')in
let V' = h(con2(xKi, Ci'))in
if V'= V then
let EKi = h(con3(XID, xKi, keytostring(XCi)))in
new r7: coins;
new Ti0: timestamp;
let v1 = h(con5(randtostring(r7), xKi, XID, keytostring(XCi),timestamptostring(Ti0)))in
let CTi1 = encrypt (con2 (randtostring(r7), timestamptostring(Ti0)), stringtokey (EKi))in
out(publicchMP, (XID, CTi1, v1, isFresh(Ti0, true)));
in(publicchPM, (CTi2:bitstring, v2: bitstring, CTj3:bitstring, checkTi: bool, checkTj: bool ));
if checkTi = true then
let (YID: bitstring, Sk : key, rA9: coins, TAuCi: timestamp)= decrypt(CTi2, stringtokey(EKi))in
let v2'= h (con5 (XID, keytostring(Sk), timestamptostring(TAuCi), coinstostring(rA9), YID))in
if v2'= v2 then
If checkTj = true then
Let (XID': bitstring, Tj1: timestamp)= decrypt(CTj3, Sk)in
if  XID' = XIDi then
new Ti1: timestamp;
let CTi3 = encrypt (con2 (YIDj, timestamptostring(Ti1)), Sk)in
let v3 = h (con4 (YIDj, XIDi, coinstostring (r9), timestamptostring(TAuCi)))in
out(publicchMP,(XIDi, CTi3, v3, isFresh(Ti1, true));
event endMPparam(NFCMobile);
out (publicchMP, encrypt (sec1, Sk)).
```

Figure 10: The NFC mobile code

As we mentioned, our authentication scheme is emulated as the concurrent execution of the processNFCMobile, processNFCPOS, and processAuC. Fig. 13 shows the code of the main process used to execute the parallel processes. The code in the first part represents the registration phases for either the NFC POS or NFC mobile (lines 151 to 168), wherein all the relevant parameters used to emulate the registration phase are initiated, while the second part of the code is used to Launch an unbounded number of sessions between the processes.

Fig. 14. Illustrates the results of our verification code, wherein the first four results demonstrate that the attacker has not been traced as resetting the sec1, sec2, sec3, and sec4. Hence, the session key sk is secure against the various attacks that are emulated by the ProVerif tool, while the rest the four results show that the eight events have been executed in stable orders. Hence, mutual authentication is achieved between all the participants and our proposed scheme is secure according to the formal verification.

```
let processNFCPOS(Sj: bitstring, PWj: bitstring, rj: bitstring, Fj: bitstring, H: bitstring) =
in(XID: bitstring, CTi1: bitstring, v1: bitstring, checkTjj: bool);
in(publicchPP, YIDy :bitstring);
if YIDy = YIDj then
get NFCPoP(=YIDj, YCj, Fj, H, rj)in
event StartAPparam(AuC);
Let Cj' = h(con3(Sj, PWj , coinstostring (rj)))in
let yKj = xor(Fj, Cj')in
let H' = h(con2(yKj, Cj'))in
if H'= H then
if checkTjj = true then
let EKj = h(con3(YIDj, yKj, keytostring(YCjj)))in
new r8: coins;
new Tj0: timestamp;
let h1 = h(con5(randtostring(r8), keytostring (KgY),YID, keytostring (YCj), timestamptostring(Tj0))) in
Let CTj1 = encrypt (con2 (randtostring(r8), timestamptostring(Tj0)), stringtokey(EKj))in
out(publicchPA, (YID, CTj1, h1, isFresh(Tj0, true), XID, CTi1, v1, isFresh(Ti0, true)));
in(publicch2, (CTi2:bitstring, v2: bitstring, CTj2:bitstring, h2: bitstring, checkTAuCi: bool, checkTAuCj: bool));
event StartMPparam(NFCMobile);
if checkTAuCi = true && checkTAuCj = true then
let (Sk : key, r10: coins, TAuCj: timestamp)= decrypt(CTj2, stringtokey(EKj))in
let h2'= h (con4 (YIDj, keytostring(Sk), timestamptostring(TAuCj), coinstostring (r10), XIDi))in
if h2'= h2 then
new Tj1: timestamp;
let CTj3 = encrypt (con2 (XIDi, timestamptostring(Tj1)), Sk)in
out(publicchPM, (CTi2, v2, CTj3, isFresh(TAuCi, true), isFresh(Tj1, true)));
in(publicch1, (XIDi : bitstring, CTi3: bitstring, v3:bitstring, checkTm: bool) );
If checkTm = true then
Let (YID': bitstring, Ti1: timestamp)= decrypt(CTi3, Sk) in
if YID'= YID then
let h3 = h (con4 (YID, XID, coinstostring (rA10), timestamptostring(TAucjj)))in
out(publicchPA(YID, XID, h3, v3);
event endPAparam(NFCPOS);
event endPMparam(NFCPOS);
(*-End NFCPOS Authentication Phase*)
out(publicchPM, encrypt(sec2, Sk));
out(publicchPA, encrypt(sec3, Sk)).
```

Figure 11: The NFC POS code

```
let processAuC =
in(publicch2, (YIDj: bitstring, CTj1: bitstring, h1: bitstring, checkTj0: bool, XIDi: bitstring, CTi1: bitstring,
v1: bitstring, checkTi0: bool ));
if checkTi0 = true && checkTj0 = true then
event StartPAparam(NFCPOS);
get NFCMoA(=XIDi, r3, XCi, IDi)in
let Ki = h(con3(XIDi, keytostring(x), randtostring(r3)))in
let EKi = h(con3(XIDi, Ki, keytostring(XCi)))in
let (r7: coins, Ti0: timestamp)= decrypt(CTi1, stringtokey(EKi))in
let v1'= h (con5 (coinstostring(r7), Ki, XIDi, keytostring(XCi), timestamptostring(Ti0))in
if v1'= v1 then
get NFCPoA(=YIDj, r6, YCj, IDj)in
let Kj = h(con3(YIDj, keytostring(y), randtostring(r6)))in
let EKj = h(con3(YIDj, Kj, keytostring(YCj)))in
let (r8: coins, Tj0: timestamp)= decrypt(CTj1, stringtokey(EKj))in
let h1'= h (con5 (coinstostring(r8), Kj, YIDj, keytostring(YCj), timestamptostring(Tj0))in
if h1'= h1 then
new TAuCi: timestamp;
new Sk: key;
new r9: coins;
let CTi2 = encrypt (con4 (YIDj, keytostring(Sk), coinstostring (r9), timestamptostring(TAuCi)), stringtokey(EKi))in
let v2 = h (con5 (XIDi, keytostring(Sk), timestamptostring(TAuCi), coinstostring(r9), YIDj)in
new TAuCj: timestamp;
new r10: coins;
let CTj2 = encrypt (con3 (keytostring(Sk), coinstostring (r10), timestamptostring(TAuCj)), stringtokey(EKj))in
let h2 = h (con5 (YIDj, keytostring (Sk), timestamptostring(TAuCj), coinstostring(r10),XIDi)in
out(publicchPA, (CTj2, h2,CTi2, v2,isFresh(TAuCj, true), isFresh(TAuCi, true)));
in (publicchPA, (YID: bitstring, XID: bitstring, h3: bitstring, v3, bitstring);
Let v3' = h (con4(YID, XID, coinstostring(r9), timestamptostring(TAuCi)))in
if v3'= v3 then
Let h3' = h (con4(YID, XID, coinstostring(r10), timestamptostring(TAuCj)))in
if h3'= h3 then
event endAPparam(AuC);
out(publicchPA, encrypt(sec4, Sk)).
```

Figure 12: The AuC code

```
process
new IDi, PWi, Si, Fi, V; Ci: bitstring;
new ri, r1, r3: coins;
new IDj, PWj, Sj, Fj, H, Cj: bitstring;
new rj, r1, r6: coins;
new XCi, Ki: key;
new YCj, Kj: key;
let Ki = h(con3(IDi, keytostring(x),coinstostring(r3)))in
let Kj = h(con3(IDj, keytostring(y),coinstostring(r6)))in
let Ci = h(con3(Si, PWi , coinstostring (ri)))in
let Cj = h(con3(Sj, PWj , coinstostring (rj)))in
let Fi = xor(Ki, Ci)in
let V = h(Con2(Ki, Ci)in
let Fj = xor(Kj, Cj) in
let H = h(Con2(Kj, Cj) in
insert NFCMoA(r3, XCi, IDi, XIDi);
insert NFCPoA(r6, YCj, IDj, YIDj);
insert NFCmoM(XIDi, XCi, Fi, V, ri);
insert NFCPoP(YIDj, YCj, Fj, H, rj);
(
(!processNFCMobile)|
(!processNFCPOS)|
(!processAuC)
)
```

Figure 13: The main process code

```
1- RESULT not attacker(sec1[]) is true.
2- RESULT not attacker(sec2[]) is true.
3- RESULT not attacker(sec3[]) is true.
4- RESULT not attacker(sec4[]) is true.
5- RESULT inj-event(endMAparam(NFCPOS)) ==>  inj-event(StartMAparam(NFCPOS))is true.
6- RESULT inj-event(endMPparam(NFCMobile)) ==> inj-event(StartMPparam(NFCMobile))is true.
7- RESULT inj-event(endAPparam(AuC)) ==> inj-event(StartAPparam(AuC))is true.
8- RESULT inj-event(endPAparam(NFCPOS)) ==> inj-event(endPAparam(NFCPOS))is true.
```

Figure 14: ProVerif output results

4 Informal Security Verification Analysis

In this section, we discuss the security of the proposed authentication scheme through the achievement of security services. Besides this, the informal analysis is demonstrated to show how our proposed authentication scheme can prevent the related attacks types. Finally, the security features comparison of our proposed authentication scheme with other related schemes is presented.

4.1 Security Services Achievements

The Proposed Scheme Supports Full Mutual Authentication.

Proof. Our authentication scheme can fully achieve mutual authentication among NFC mobile (Xi), NFC POS (Yj) and AuC during the authentication phase through the following authentication messages.

The computed value of (H1') by the AuC matches the received value of (H1) from Yj via (M2). Besides this, the computed value of (H2') by the Yj matches the received value of (H2) from AuC via (M3). In addition, the transmitted value of (H3) by Yj via (M6) matches the computed value of (H3') by AuC. Thus, mutual authentication can be supported among Yj and AuC by the exchanging M2, M3 and M6 messages.

When the computed value of (V1') by the AuC matches the received value of (V1) from Xi via (M1). Besides this, the computed value of (V2') by the Xi matches the received value of (V2) from AuC via (M4). Furthermore, the transmitted value of (V3) by Xi via (M6) matches the computed value of (V3') by AuC. Thus, mutual authentication can be supported among Xi and AuC by exchanging M2, M4 and M6 messages.

On the other side, the value of (XIDi') that was decrypted by the Xi matches the value of (XIDi) that was encrypted by Yj using the shared (Sk) within received M4. Besides this, the value of (YIDj') that was decrypted by the Yj matches with the value of (YIDj) that was encrypted by Xi using the same shared (Sk) within the received M5. Thus, mutual authentication can be achieved between Xi and Yj via the exchange of M4 and M5 messages.

Therefore, the proposed authentication scheme is able to support full mutual authentication services among all the communication entities.

The Proposed Scheme Supports a Full NFC Devices Anonymity.

Proof. To protect NFC mobile and NFC POS identities, the proposed scheme employs pseudonym identities (XIDi and YIDj) as transmitted messages instead of the NFC device's real identities. The pseudonym identities are generated in random manner and updated after finishing each authentication session. Thus, the pseudonym identities are distinct for both NFC mobile and NFC POS devices in every authentication session. Furthermore, it is unattainable for an adversary to obtain the NFC mobile and NFC POS real identities from the messages that have been transferred between the communication entities.

Therefore, the proposed authentication scheme is able to achieve full anonymity and the untraceability of services.

The Proposed Scheme Supports a Full Perfect Forward Secrecy.

Proof. According to our proposed authentication scheme, assume that the adversary has gained the long-term keys of the NFC devices, which are (Ki, XCi) and (Kj, YCj) of the NFC mobile and NFC POS, respectively. Then, the adversary still cannot obtain the session key (Sk) that has been generated by the AuC. This is because, after each succeeded authentication session, the keys XCi and YCj will be changed by one-way hash functions, as XCi' = h(XCi) and YCj' = h(YCj) in both the NFC mobile and NFC POS, respectively. The reason for this is that the used hash functions are one-way functions, and so the adversary cannot obtain the XCi and YCj from XCi' and YCj'.

Therefore, our proposed authentication scheme is able to support a perfect forward secrecy service during the authentication stage.

4.2 Resistance to Related Attacks

The Proposed Scheme Resists De-Synchronization Attack.

Proof. Since our authentication scheme uses XIDi, YIDj and has a group of one-way hash functions in order to support full anonymity for NFC devices and perfect forward secrecy features. Therefore, it also wants a method to preserve the synchronization of the hash values of the NFC mobile, NFC POS, and the AuC.

In our authentication scheme, the consistency of the (XIDi) and hash chain value of h(XCi) will be guaranteed by exploiting two pseudonym identities (XIDi0) and (XIDi1) for the connection between the Xi and AuC. Similarly, for the connection between the Yj and AuC, our authentication scheme uses two pseudonym identities, (YIDj0) and (YIDj1), to ensure the consistency of YIDj and the hash chain value h (YCj). Since the hash functions that have been considered in our scheme are one-way hash functions, even if the adversary can block the authentication messages, the Xi, Yj and AuC can re-synchronize the hash values between them. With a view to make our discussion more precise, Fig. 15 illustrates different desynchronization attack scenarios.

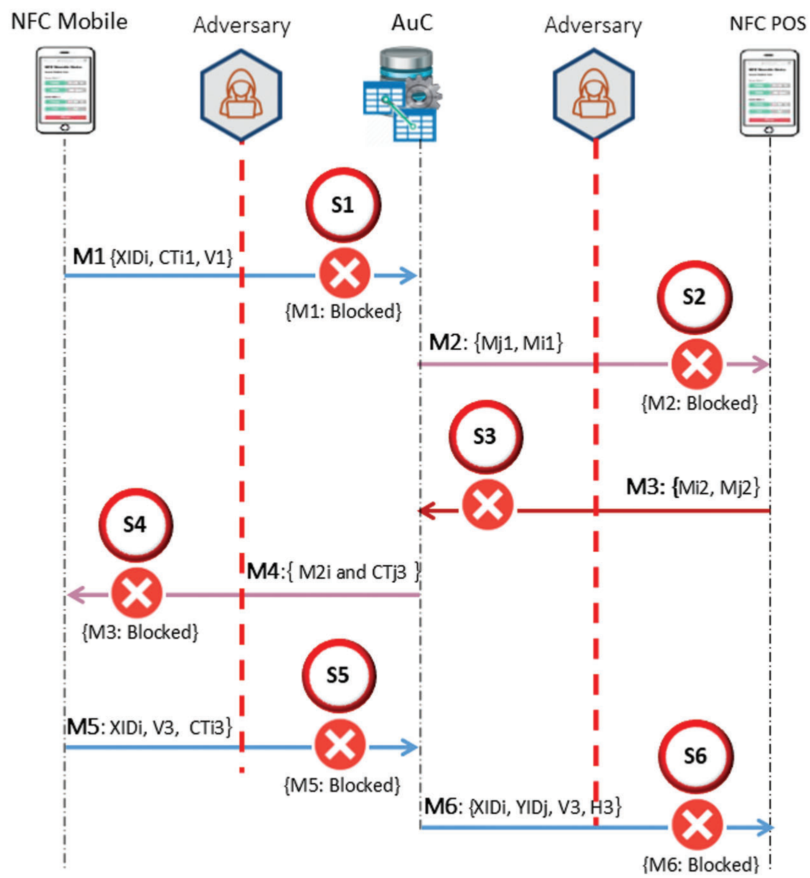


Figure 15: Desynchronization attack scenarios

Scenario (S1): Assume the adversary has blocked (M1) message, clearly will not affect the synchronization between Xi, Yj and AuC, wherein the authentication entities have not started updating the value of XIDi nor the hash chain value of h(XCi). Therefore, this scenario will not be taken into account.

Scenario (S2): Assume the adversary has blocked (M2) message, obviously will not affect synchronization between Xi, Yj and AuC, wherein the authentication entities have not even started updating the YIDj nor the hash chain value h (YCj). Therefore, this scenario is the same as (S1) and will not be taken into account.

Scenario (S3): Assume the adversary has blocked (M3) message, the asynchronous pseudonym identities of the NFC POS will be considered between the Yj and AuC. In this case, since the hash chain values the Yi and AuC are not updated, then the synchronization of these identities only needs to be considered. The value of YIDj0 in the AuC has been renewed, while the value of YIDj in the Yj does not update. Fortunately, the previous pseudonym identity is saved in YIDj1 in the AuC, which is YIDj1 = YIDj. Thus, when the next authentication session is started by the Xi by unchanged XIDi, then the Yj will use the unchanged Yj, the AuC is still able to distinguish the Yj and completes the authentication.

Similarly, the asynchronous pseudonym identities of the NFC mobile will be considered between the Xi and AuC. Since both the hash chain values the Xi and AuC are not updated, then the synchronization of these identities only needs to be considered. The value of XIDi0 in the AuC is a fresh pseudonym identity, while XIDi value in the Xi does not update. Fortunately, the previous pseudonym identity is saved in XIDi1 in the AuC, which is XIDi1 = XIDi. Thus, when the next authentication session is started by the Xi by unchanged XIDi, the AuC is still able to distinguish the Xi and completes the authentication. In general, this scenario will cause asynchronous pseudonym identities between the <Yj and AuC> and <Xi and AuC>, but it will not have any effect on the next sessions.

Scenario (S4): Assume the adversary has blocked (M4) message, clearly this attack will not affect the synchronization of pseudonym identities of the NFC devices and AuC. Therefore, this scenario is the same as (S3) and will be ignored.

Scenario (S5): Assume the adversary has blocked (M5) message, it is like scenario (S3). However, the pseudonym identities values in the Xi and AuC have updated, and this means XIDi = XIDi0, and so we only want to consider the synchronization of two hash chain values in Xi and AuC. In this scenario, the hash chain value in the Xi is updated, while the value hash chain in the AuC is unchanged. When Xi uses a changed hash chain value, it initiates a new session, and the AuC will update its hash chain value through checking whether it is the value of XIDi1 = ϕ or not. Therefore, even if this scenario will cause asynchronous hash chain value between the Xi and the AuC, the two pseudonym identities will synchronize the hash chain values again.

Scenario (S6): Assume the adversary has blocked (M6) message, this scenario is similar to (S5) regarding to the value of YIDj0, if the pseudonym identities values in the Yj and AuC have updated, it means that YIDj0 = YIDj, and so we only need to consider synchronization of two hash chain values in Yj and AuC. The hash chain value in the Yj is updated, while the value hash chain in the AuC is unchanged. When Yj, using changed hash chain values, initiates a new session, the AuC will update its hash chain value through checking whether it is the value of YIDj1 = ϕ or not. Therefore, even if this scenario will cause asynchronous hash chain value between the Yj and the AuC, the two pseudonym identities will synchronize the hash chain values synchronize again.

Therefore, according to the analysis of the above-mentioned de-synchronization attack scenarios, the proposed authentication scheme can resist de-synchronization attacks.

The Proposed Scheme Resists Stolen Password Table Attack.

Proof. In the proposed authentication scheme, no password table of the NFC mobile or a NFC POS is stored in the AuC. Therefore, the proposed authentication scheme will not be subjected stolen verifier table attacks and can prevent such attacks.

The Proposed Scheme Resists Impersonation Attack.

Proof. In the proposed authentication scheme, the adversary cannot forge the NFC mobile or NFC POS devices. The adversary should be able to generate a valid value of {Ti0, XIDi, CTi1, and V1} in order to forge the Xi. It is impracticable where the adversary does not identify the secret keys XCi and Ki. Similarly, the adversary should be able to generate a valid value of {Tj0, YIDj, YCj, and H1} in order to forge the Xi. it is impracticable where the adversary does not identify the secret keys Kj and YCj. Therefore, our authentication scheme can prevent both NFC mobile and NFC POS impersonation attacks.

The Proposed Scheme Resists Spoofing Attack.

Proof. The adversary cannot forge the legitimate authentication messages of the NFC mobile or NFC POS without the pair of secret keys (Ki and XCi) or (Kj and YCj), respectively. Therefore, the NFC mobile or NFC POS device cannot spoof any other NFC mobile or NFC POS devices in in our authentication scheme.

The Proposed Scheme Resists Replay Attack.

Proof. The proposed authentication scheme uses a set of the timestamps to resist replay attacks. For the connection between the NFC mobile and AuC, message (M1) contains a current timestamp (Ti0) of the Xi, and other message flows deploy the challenge-response messages in order to resist reply attacks. For the connection between the NFC POS and AuC, message (M2) contains a current timestamp (Tj0) of the Yj, and other message flows deploy the challenge-response messages to resist reply attacks. For the connection between the Xi and Yj, message (M4) contains a current timestamp (T1j) of the Yj, and the (M5) message includes a current timestamp (T1i) of the Xi. As a result, when the NFC mobile and NFC POS devices admit each other, they should be in the current authentication session and not in the prior authentication session. Therefore, our authentication scheme can resist replay attack.

The Proposed Scheme Resists Man-in-the-Middle Attack.

Proof. In our proposed authentication scheme, authentication messages that have been transmitted are protected by the secret values of the NFC mobile (Ki and XCi) and NFC POS (Kj and YCj), and anyone without these pairs of keys cannot forge legitimate authentication messages. Thus, our authentication scheme prevents man-in-the-middle attack.

The Proposed Scheme Resists Wrong Password Login Attack.

Proof. In our proposed authentication scheme, the password verification data V = h (Ki ‖ Ci)) is saved in the NFC mobile, which is incorrect to validate the correctness of the password. If the user inserts the incorrect password PWi', the verification data V and V' will not be equal. The same thing occurs for the NFC POS, which is the password verification data H = h (Kj ‖ Cj)) that is saved in the NFC POS, and is incorrect to validate the correctness of the password. If the seller inputs the wrong password PWj', the verification data H and H' will not be equal. Therefore, the proposed authentication scheme can prevent unauthorized logins.

4.3 Security Comparisons

This section compares the security features of our proposed authentication scheme with recent related schemes [1, 15, 34]. Tab. 7 lists the comparison results.

Table 7: The security features the comparison between the proposed scheme and other related schemes

Security features	[1]	[15]	[34]	Ours
Support the full mutual authentication	No	Yes	No	Yes
Support the full NFC devices anonymity and untraceability	No	Yes	No	Yes
Support the full perfect forward secrecy	No	Yes	No	Yes
Support the NFC mobile login function	No	No	No	Yes
Support the NFC POS login function	No	No	No	Yes
Support the NFC mobile password change function	No	No	No	Yes
Support the NFC POS password change function	No	No	No	Yes
Prevent the De-synchronization Attack	Yes	Yes	Yes	Yes
Prevent the NFC mobile impersonation attack	No	Yes	No	Yes
Prevent the NFC POS impersonation attack	No	Yes	No	Yes
Prevent the NFC mobile spoofing attack	Yes	Yes	Yes	Yes
Prevent the NFC POS spoofing attack	Yes	Yes	Yes	Yes

(Continued)

Table 7 (continued)

Security features	[1]	[15]	[34]	Ours
Prevent the replay attack	Yes	Yes	Yes	Yes
Prevent the Man-in-the-middle Attack	Yes	Yes	Yes	Yes
Prevent the Wrong password login/update attack	n/a	n/a	n/a	Yes
Prevent the Password table attack	n/a	n/a	n/a	Yes

The results in Tab. 7 illustrate that the proposed authentication scheme can achieve all the listed security features. Where the authentication schemes in [1,15,34] did not support the security features such as full mutual authentication, full NFC devices anonymity and untraceability, and full perfect forward secrecy. Furthermore, the only scheme that supported the NFC mobile login function, NFC POS Login function, NFC mobile password change function, and NFC POS password change function was the proposed authentication scheme. That means that our scheme offers more security features than the other related authentication schemes.

5 Performance Analysis

This section compares the storage space, communication, and communication costs of our authentication scheme with recent related schemes [1,15,34]. We will only focus on comparing the authentication phase where the other phases are not used frequently.

As pointed out in [39,40], the size of all the parameters to 128 bits, and the input and output block sizes of symmetric cryptography functions are multiples of 128 bits; the output of the hash functions is equal to 160 bits, the running time of AES cryptographic function is ($T_{E/D}$≅0.0056 s), and the running time of the one-way hash function are SHA-1, MAC and HMAC is ($T_{mac} \approx T_{hmac} \approx T_h \cong 0.00032$ s).

5.1 Storage Space Costs Analysis

Tab. 8 illustrates the storage space costs of the NFC devices of the proposed authentication scheme and other authentication schemes [1,15,34]. In the proposed authentication scheme, storage space costs for the NFC mobile {XIDi, XCi, Fi, and V} are required (128 + 128 + 128 + 160) = 544 bits, the NFC POS {YIDj, Cj, Fj, and H} are required (128 + 128 + 128 + 160) = 544 bits, while for the AuC, it requires 1024 bits. We note that our authentication scheme has the highest cost. The reason for this is that the NFC devices of our authentication scheme store a set of pseudonym identities in order to preserve the anonymity service.

Table 8: Storage space cost analysis

Authentication Scheme	NFC mobile (bits)	NFC POS (bits)	AuC (bits)
[1]	544	416	416
[15]	256	256	256
[34]	288	288	160
Ours	544	544	1024

5.2 Communication Costs Analysis

The communication costs are computed according to the total bits of authentication messages size, which transmit between the NFC parties through the authentication phase. The communication costs of the proposed scheme can be summarized as follows: M1: {XIDi, CTi1 and V1} requires (128 + 128 + 160) = 416 bits, M2: {M1, YIDj, CTj1 and H1} requires (416 + 128 + 128 + 160) = 832 bits, M3: {CTi2, V2, CTj2 and H2} needs (160 + 128 + 160 + 128) = 576 bits, M4: {CTi2, V2 and CTj3} requires (128 + 160 + 128) = 416 bits, M5: {XIDi0, V3 and CTi3} requires (128 + 160 + 128) = 416 bits, and M6: {XIDi0, YIDj0, V3, H3} requires (128 + 128 + 160 + 160) = 576 bits. Tab. 9 illustrates the communication costs of our authentication scheme and other authentication schemes [1,15,34]. We note that our authentication scheme has the highest cost. The reason for this is that the proposed authentication scheme is executed with six authentication messages in order to preserve the full mutual authentication service, but the total communication costs are still in the applicable range.

Table 9: Communication cost analysis.

Scheme	Authentication phase						Total (bits)
	M1	M2	M3	M4	M5	M6	
[1]	384	512	768	256	128	N/A	1792
[15]	128	512	1152	576	416	N/A	2784
[34]	256	416	832	320	160	N/A	1984
Ours	416	832	576	416	416	576	3232

5.3 Computation Costs Analysis

Tab. 10 shows the computation costs of our authentication scheme in comparison to the other recent authentication schemes [1,15,34]. The computation costs are computed based on the total execution time of the encryption, decryption, MAC, and hash functions that are executed during the authentication phase. We note that the proposed authentication scheme has the highest cost. The reason for this is that our authentication scheme executes both encryption/decryption functions in all authentication messages to preserve security services, but the total computation costs are still in the applicable range.

Table 10: Total of executed cryptographic functions

Scheme	Total cryptographic functions	Total execution time
[1]	$8\ T_{hmac} + 1T_h$	0.0122 s
[15]	$14T_h + 4\ T_{E/D}$	0.0333 s
[34]	$8\ T_{mac}$	0.0126 s
Ours	$22T_h + 12\ T_{E/D}$	0.1428 s

6 Conclusion

We proposed a new authentication scheme for the NFC mobile payment system in order to overcome current security deficiencies and to make such a system more secure. The proposed authentication scheme has significant security services, such as full mutual authentication between all communication entities, full anonymity for NFC devices, and full perfect forward secrecy services. The ProVerif tool was used to verify the mutual authentication and the shared key secrecy. The BAN logic model was performed to confirm the mutual authentication between the communication entities. According to the several attack scenarios that were discussed, the highest security features of our authentication scheme were illustrated. Therefore, it can not only achieve full security features, but can also prevent numerus attacks such as password table, smartcard loss, replay, wrong login information, man-in-the-middle, insider, impersonation, and desynchronization attacks. Furthermore, a performance analysis showed that the proposed authentication scheme has an applicable cost range in the storage space, computation, and communication. Finally, our proposed authentication scheme is applicable in NFC mobile payment systems in order to execute payment transactions in a safe manner.

Acknowledgement: The authors express their gratitude to all members of the Computer and Information Sciences College at Jouf University for their support.

Funding Statement: The authors would like to thank the Deanship of Graduate Studies at Jouf University for funding and supporting this research through the initiative of DGS, Graduate Students Research Support (GSR) at Jouf University, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. A. Abouhogail and A. H. Ali, "Design and development of an advanced authentication protocol for mobile applications using NFC technology," *Journal of Computer Science*, vol. 15, no. 12, pp. 1809–1819, 2019.

[2] V. Odelu, A. K. Das and A. Goswami, "SEAP: Secure and efficient authentication protocol for NFC applications using pseudonyms," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 1, pp. 30–38, 2016.

[3] A. Rahul, G. Gokul Krishnan, H. Unni Krishnan and S. Rao, "Near field communication (NFC) technology: A survey," *International Journal on Cybernetics & Informatics (IJCI)*, vol. 4, no. 2, pp. 133–144, 2015.

[4] S. Nashwan, "SE-H: Secure and efficient hash protocol for RFID system," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 9, no. 3, pp. 358–365, 2017.

[5] M. Badra and R. Badra, "A lightweight security protocol for NFC-based mobile payments," in *Proc. the 7th International Conf. on Ambient Systems, Networks and Technologies*, Madrid, Spain, pp. 705–711, 2016.

[6] S. K. Timalsina, R. Bhusal and S. Moh, "NFC and its application to mobile payment: overview and comparison," in *Proc. the 8th Int. Conf. on Information Science and Digital Content Technology*, Jeju, South Korea, 2012.

[7] N. El Madhoun, F. Guenane and G. Pujolle, "An online security protocol for NFC payment: Formally analyzed by the scyther tool," in *Proc. the IEEE 2nd Int. Conf. on Mobile and Secure Services (MobiSecServ)*, FL, USA, pp. 1–7, 2016.

[8] A. Allyson, V. Lakshmi and A. Packialatha, "Mobile devices using NFC in payment applications," *International Journal of Innovative Research in Technology & Science*, vol. 3, no. 1, pp. 32–36, 2015.

[9] A. Khan, M. Gandhi, A. Jain and N. Kacholia, "Emerging markets driving the payments transformation," Ahmedabad, Gujarat, India: PWC Network, 2016. [Online]. Available: https://www.pwc.com/emergingmarketspayments.

[10] J. Ahn, S. Lee and H. Kim, "NFC based privacy preserving user authentication scheme in mobile office," *International Journal of Computer and Communication Engineering*, vol. 5, no. 1, pp. 61–70, 2016.

[11] P. Pourghomi, M. Q. Saeed and G. Ghinea, "A proposed NFC payment application," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 8, pp. 173–181, 2013.

[12] H. Du, "NFC technology: Today and tomorrow," *International Journal of Future Computer and Communication*, vol. 2, no. 4, pp. 351–354, 2013.

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

[14] M. Al-Fayoumi and S. Nashwan, “Performance analysis of SAP-NFC protocol,” *International Journal of Communication Networks and Information Security*, vol. 10, no. 1, pp. 125–130, 2018.

[15] S. Nashwan, “Secure authentication protocol for NFC mobile payment systems,” *International Journal of Computer Science and Network Security*, vol. 17, no. 8, pp. 256–263, 2017.

[16] S. Nashwan and B. Alshammari, “Formal analysis of MCAP protocol against replay attack,” *British Journal of Mathematics & Computer Science*, vol. 22, no. 1, pp. 1–14, 2017.

[17] R. Sivaranjani, R. Sujitha, D. Sindhu and T. Tharani, “Secure and efficient authentication protocol using pseudonym,” *Journal of Chemical and Pharmaceutical Sciences, Special Issue*, vol. 5, pp. 104–108, 2017.

[18] M. Rahman and H. Elmiligi, “Classification and analysis of security attacks in near field communication,” *International Journal of Business & Cyber Security*, vol. 1, no. 2, pp. 1–14, 2017.

[19] R. Heeks, *In Information and Communication Technology for Development (ICT4D)*, London, UK: Routledge, 2017. [Online]. Available: DOI 10.4324/9781315652603.

[20] D. Nelson, M. Qiao and A. Carpenter, “Security of the near field communication protocol: An overview,” *Journal of Computing Sciences in Colleges*, vol. 29, no. 2, pp. 94–104, 2013.

[21] M. Roland, *In Security Issues in Mobile NFC Devices*, New York, United States: Springer International Publishing, 2015. [Online]. Available: DOI 10.1007/978-3-319-15488-6.

[22] M. P. Bosamia, “Positive and negative impacts of information and communication technology in our everyday life,” in *Proc. Int. Conf. on Disciplinary and Interdisciplinary Approaches to Knowledge Creation in Higher Education: CANADA & INDIA (GENESIS 2013)*, Bhavnagar, India, 2013.

[23] C. B. Chew, K. C. Wei, T. W. Sheng, M. Mahinderjit-Singh, N. H. A. H. Malim *et al.*, “Security challenges and mitigations of NFC-enabled attendance system,” in *Proc. the IEEE Int. Conf. in Swarm Intelligence*, Springer, Cham, pp. 160–167, 2015.

[24] N. El Madhoun, F. Guenane and G. Pujolle, “A cloud-based secure authentication protocol for contactless NFC payment,” in *Proc. of the IEEE 4th International Conference on Cloud Networking, Niagara Falls*, Canada, pp. 328–330, 2015.

[25] B. Seo, S. Lee and H. Kim, “Authenticated key agreement based on NFC for mobile payment,” *International Journal of Computer and Communication Engineering*, vol. 5, no. 1, pp. 71–78, 2016.

[26] J. Ling, Y. Wang and W. Chen, “An improved privacy protection security protocol based on NFC,” *International Journal of Network Security*, vol. 19, no. 1, pp. 39–46, 2017.

[27] S. Nashwan, “SAK-AKA: A secure anonymity key of authentication and key agreement protocol for LTE network,” *International Arab Journal of Information Technology*, vol. 14, no. 5, pp. 790–801, 2017.

[28] M. Al-Zewairi, S. Hamdan and M. Al-Fayoumi, “Enhanced multi-key model for risk adaptive hybrid RFID access control system,” in *Proc. Int. Arab Conf. on Information Technology (ACIT 2017)*, Yasmine Hammamet, Tunisia, 2017.

[29] A. Alshehri and S. Schneider, “Addressing NFC mobile relay attacks: NFC user key confirmation protocols,” *International Journal of RFID Security and Cryptography*, vol. 3, no. 2, pp. 137–147, 2014.

[30] F. Ota, M. Roland, M. Holzl, R. Mayrhofer and A. Manacero, “Protecting touch: Authenticated app-to-server channels for mobile devices using NFC tags,” *Information*, vol. 8, vo. no. 3, pp. 1–18, 2017.

[31] J. Ling, Y. Wang and W. Chen, “An improved privacy protection security protocol based on NFC,” *International Journal of Network Security*, vol. 19, no. 1, pp. 39–46, 2017.

[32] U. B. Ceipidor, C. M. Medaglia, A. Marino, S. Sposato and A. Moroni, “Kernees: A protocol for mutual authentication between NFC phones and POS terminals for secure payment transactions,” in *Proc. IEEE 9th Int. ISC Conf. on Information Security and Cryptology (ISCISC)*, pp. 115–120, Tabriz, Iran, 2012.

[33] 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. the IEEE Int. Conf. on Information Networking*, Siem reap, Cambodia, pp. 133–138, 2015.

[34] Y. Tung and W. Juang, “Secure and efficient mutual authentication scheme for NFC mobile devices,” *Journal of Electronic Science and Technology*, vol. 15, no. 3, pp. 1–6, 2017.

[35] M. Burrows, M. Abadi and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[36] H. M. N. Al Hamadi, C. Y. Yeun, M. J. Zemerly, M. A. Al-Qutayri and A. Gawanmeh, "Verifying mutual authentication for the DLK protocol using ProVerif tool," *International Journal for Information Security Research*, vol. 3, no. 1, pp. 256–265, 2013.

[37] B. Blanchet, "Modeling and verifying security protocols with the applied Pi calculus and ProVerif," *Foundations and Trends in Privacy and Security*, vol. 1, no. 1, pp pp. 1–135, 2016.

[38] D. Baelde, S. Delaune and S. Moreau, "A method for proving unlinkability of stateful protocols," in *Proc. 33rd IEEE Computer Security Foundations Symposium*, Boston, USA, pp. 169–183, 2020.

[39] S. Nashwan, "An End-to-end authentication scheme for healthcare IoT systems using WMSN," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 607–642, 2021.

[40] S. Nashwan, "AAA-WSN: Anonymous access authentication scheme for wireless sensor networks in big data environment," *Egyptian Informatics Journal*, vol. 22, no. 1, pp. 15–26, 2021.