PSAP: Pseudonym-Based Secure Authentication Protocol for NFC Applications

Jie Xu, Kaiping Xue, *Senior Member, IEEE*, Qingyou Yang, and Peilin Hong

***Abstract*—Nowadays, near field communication (NFC) has been widely used in electronic payment, ticketing, and many other areas. NFC security standard requires the use of public key infrastructure (PKI) to implement mutual authentication and session keys negotiation in order to ensure communication security. In traditional PKI-based schemes, every user uses a fixed public/private key pair to implement authentication and key agreement. An attacker can create a profile based on user's public key to track and compromise the user's privacy. Recently, He *et al.* and Odelu *et al.* successively proposed pseudonym-based authentication key and agreement protocols for NFC after Eun *et al.*'s protocol (2013), which is first claimed to provide conditional privacy for NFC. They respectively claimed that their scheme can satisfy the security requirements. In this paper, first, we prove that their protocols still have security flaws, including the confusion of the user's identity and the random identity. Then, we propose a pseudonym-based secure authentication protocol (PSAP) for NFC applications, which is effective in lifetime and includes time synchronization-based method and nonce-based method. In our scheme, trusted service manager issues pseudonyms but does not need to maintain verification tables and it could reveal the user's identity of internal attackers. Furthermore, security and performance analysis proves that PSAP can provide traceability and more secure features with a little more cost.**

***Index Terms*—Authentication and key agreement, conditional privacy protection, near field communication (NFC), pseudonym, traceability.**

I. Introduction

NEAR field communication (NFC) is a communication standard, which is used to carry point-to-point wireless data interactions with a working distance no more than 10 cm from internal chips embedded in electronic devices to the receiver [1], [2]. In recent years, mobile phone manufacturers, banking institutions, and mobile network operators are trying to enable mobile phones and many other portable devices to support NFC. NFC devices have been widely used in contactless electronic payment, ticketing, and many other electronic commerce areas [3], [4]. Moreover, there is still a great potential for NFC applications, such as healthcare [5], [6] and rescue organization [7]. What they all have in common is that these applications usually require strict security to protect user privacy.

The existing international standards have specified the interface and protocol for simple wireless communication between close coupled devices, by which NFC devices communicate with bit rates of 106, 212, or 424 kbit/s [8], [9], which also define three different modes of operation: 1) card emulation mode; 2) reader/writer mode; and 3) peer-to-peer mode [10], [11]. In this paper, we focus on peer-to-peer mode, in which the device starting the communication is called initiator and the other is called target. Initiator and target exchange commands, responses, and data in alternating or half duplex communications. Moreover, the standard ISO/IEC 18092 (named NFCIP-1) [8] defines that in the initialization and single device detection steps, the initiator and the target must be able to read a random number (NFCID) of each other, which is dynamically generated and may be marked as ID_A or ID_B (A and B represent the initiator and the target) for identifying NFCIP-1 devices in a communication process. After obtaining the random ID_A or ID_B of the target (or the initiator) in the operating field, the initiator (or the target) may communicate with multiple targets (or initiators).

Since NFC is a short-range communication, it is considered to be more secure than the other long distance wireless communication technologies, but unfortunately, there still exists a series of challenging security and privacy problems. An attacker can eavesdrop or modify the data which is transmitted via the NFC-based wireless communication interface. Moreover, an attacker can impersonate one party to communicate with the other one.

In order to provide security protection, some NFC security standards have been proposed to define data exchange format, tag types, and security protocols (named NFC-SEC) [12]–[14]. The standard [14] specifies cryptographic mechanisms that use the elliptic curves Diffie-Hellman protocol for key agreement and the AES algorithm for data encryption and integrity. A public key infrastructure is introduced in these standards, where if two parties want to implement mutual authentication and key agreement protocol, they must first obtain other party's certificate to get the public key. The certificate is generated and signed by a certificate authority, and the user's fixed identity is included in it. Therefore, the user's identity can be tracked and his/her privacy information (e.g., actions) can be traced. Hence, we need an effective anonymous communication mechanism to authenticate each other and negotiate a session key.

Manuscript received January 1, 2018; revised February 12, 2018; accepted February 15, 2018. Date of publication March 7, 2018; date of current version March 29, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61379129, in part by the Youth Innovation Promotion Association CAS, and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Kaiping Xue.)*

The authors are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (e-mail: xj47@mail.ustc.edu.cn; kpxue@ustc.edu.cn; qingyou@mail.ustc.edu.cn; plhong@ustc.edu.cn).

Digital Object Identifier 10.1109/TCE.2018.2811260

TABLE I
NOTATIONS USED IN THIS PAPER

Symbol	Description
TSM	trusted service manager
Ep(a,b)	non-singular elliptic curve, such as $y^2 = x^3 + ax + b$ ($mod\ p$), over a prime field$GF(p)$
G	a base point on the elliptic curve Ep(a,b)
kG	G + G + G··· + G (*k* times), elliptic curve scalar multiplication over Ep(a,b), where k ∈ Z∗
d_X	the longtime private key of the user X
QX	the longtime public key of the user X
P_X^i	the i-th pseudonym of the user X
d_X^i	the i-th lifetime limited private key of the user X
Q_X^i	the i-th lifetime limited public key of the user X
UID_X	the identity of the user X
ID_A/ID_B	random ID for the activation of transport protocols, NFCID3 type [a]
KDF	a key derivation function
$MacTag_X$	a key verification tag received from X
SK	a shared secret key
r_X	a random number generated by user X
N_X	nonce of user X
TS_X	timestamp of user X
Enc(k,m)	asymmetric encrypting the message m using the key k
Sig(k,m)	signing the message m using the key k
h_1, h_2, h_3, h_4, h_5	secure hash function
$(P)_X$	the x coordinate of the point P
||	concatenation operation

[a] NFCIDn is a randomly generated number used by the RF Collision Avoidance and Single Device Detection sequence for both the Active and the Passive communication modes. NFCID3 is a Random ID for transport protocol activation [8]. This transmission process occurs at the underlying protocol.

Till now, anonymous protection mechanisms have been widely used in many applications, in which pseudonym is one of the most commonly used methods [15]–[18]. In pseudonym-based anonymous protection mechanisms, a trust third party generates multiple pseudonyms for a user, which have no relationship with the user's real identity. In order to provide privacy protection, Eun *et al.* [19] proposed a conditional privacy-preserving security protocol for NFC applications (CPPNFC). However, He *et al.* [20] pointed that Eun *et al.*'s protocol fails to prevent the impersonation attack, and they further proposed a pseudonym-based scheme (PBNFC) to address the security drawbacks in Eun *et al.*'s protocol. Moreover, Odelu *et al.* [1] further stated that He *et al.*'s protocol still fails to resist the impersonation attack[1] and they proposed a new security protocol named SEAP. However, we will explain that both He *et al.*'s and Odelu *et al.*'s protocol falsely confused the user's identity (UID, in order to distinguish it from ID used in NFC) and the randomly identity (ID), which is generated in the initialization phase and read in the single device detection step. Therefore, it results in that these two protocols cannot achieve real privacy protection. However, although existing security threats and design flaws in all above-mentioned three protocols, each one still has good parts to be learned from.

Inheriting the advantages but addressing security flaws of these three protocols, we propose a lifetime limited pseudonym-based conditional privacy protection protocol for NFC applications. Both NFC communication devices can only read each other's NFCID but without knowing the two involved parties' identities. Each party can determine the legitimacy of the other one through the verification of pseudonym provided by trusted service manager (TSM). The main contributions of this paper can be summarized as follows.

1) We analyze the security of two currently proposed NFC protocols, and propose a secure authentication and key agreement mechanism for preserving privacy in NFC with two variants, respectively, time-synchronization-based and nonce-based methods.
2) Pseudonym-based secure authentication protocol (PSAP) provides an efficient tracing mechanism, which can further reveal the identity of malicious users, to defend against internal attacks.
3) The TSM does not need to store users' identities and private keys, which reduces the risk of leaking users' confidential information stored on TSM.

II. OVERVIEW OF RELATED PROTOCOLS

In this section, the brief review of two related security protocols for NFC applications is presented. Notations used in this paper are described in Table I.

A. Overview of He et al.'s Security Protocol

In this section, we give the brief overview of He *et al.*'s protocol [20], which is claimed to be a security improvement on Eun *et al.*'s protocol. The protocol also consists of two parts: 1) initialization and 2) mutual authentication and key agreement.

1) Initialization: The initialization phase includes the following steps:

Step 1: The user A requests TSM for some pseudonyms.

Step 2: Upon receiving the request, TSM generates n random numbers for the user A (q_A^i,$i = 1, 2, \ldots, n$). Then TSM computes the ith public key, pseudonym, signature, and private key, for $i = 1, 2, \ldots, n$, as follows:

$$\begin{aligned} Q_A^i &= q_A^i G \\ P_A^i &= \left\{Q_A^i \| \mathrm{Enc}\left(d_{\mathrm{TSM}}, \left\{\mathrm{ID}_A, Q_A^i\right\}\right) \| \mathrm{ID}_{\mathrm{TSM}} \| S_{\mathrm{TSM}}^i\right\} \\ S_{\mathrm{TSM}}^i &= Sig\left(d_{\mathrm{TSM}}, Q_A^i \| \mathrm{Enc}\left(d_{\mathrm{TSM}}, Q_A^i\right) \| \mathrm{ID}_{\mathrm{TSM}}\right) \\ d_A^i &= q_A^i + h\left(\mathrm{ID}_A, P_A^i\right) d_{\mathrm{TSM}}. \end{aligned}$$

TSM stores the identity of user A and n pseudonyms into its database and send all n pseudonyms and the corresponding private keys to A through a secure channel. The user A stores the received pseudonyms and the corresponding private keys.

2) Mutual Authentication and Key Agreement: When the user A wants to implement the mutual authentication and key agreement protocol with the user B, the following steps are executed.

[1] In this paper, we also state that the most fatal security drawback insider impersonation attack come from insider registered user, but still the external impersonation attacks.

Step 1: The user A randomly selects a pseudonym P_A^i and gets the corresponding private key d_A^i. Then A generates a random number r_A and a nonce N_A. Furthermore, A computes $Q'_A = r_A G$ and sends the message $M_1 = \{Q'_A, P_A^i, N_A\}$ to B.

Step 2: Upon receiving the message from A, B randomly selects a pseudonym P_B^j and gets the corresponding private key d_B^j. Then B generates a random number r_B and a nonce N_B. Furthermore, B computes $Q'_B = r_B G$ and sends the message $M_2 = \{Q'_B, P_B^j, N_B\}$ to the user A.

Step 3: Upon receiving the message from B, A computes

$$Z_A^1 = r_A Q'_B, Z_A^2 = d_A^i\left(Q_B^j + h\left(\mathrm{ID}_{\mathrm{TSM}}, P_B^j\right)Q_{\mathrm{TSM}}\right)$$
$$\mathrm{SK} = \mathrm{KDF}\left(N_A, N_B, \mathrm{ID}_A, \mathrm{ID}_B, Z_A^1, Z_A^2\right)$$
$$\mathrm{MacTag}_A = f\left(\mathrm{SK}, \mathrm{ID}_A, \mathrm{ID}_B, Q'_A, Q'_B\right).$$

Finally, A sends the authentication $M_3 = \{\mathrm{MacTag}_A\}$ to B.

Step 4: Upon receiving the message from A, B computes

$$Z_B^1 = r_B Q'_A, Z_B^2 = d_B^j\left(Q_A^i + h\left(\mathrm{ID}_{\mathrm{TSM}}, P_A^i\right)Q_{\mathrm{TSM}}\right)$$
$$\mathrm{SK} = \mathrm{KDF}\left(N_A, N_B, \mathrm{ID}_A, \mathrm{ID}_B, Z_B^1, Z_B^2\right).$$

Then B checks whether MacTag_A is equal to $f(\mathrm{SK}, \mathrm{ID}_A, \mathrm{ID}_B, Q'_A, Q'_B)$. If not so, B stops the session; otherwise, A is authenticated, then B further computes $\mathrm{MacTag}_B = f(\mathrm{SK}, \mathrm{ID}_B, \mathrm{ID}_A, Q'_B, Q'_A)$ and sets SK as the session key. B sends $M_4 = \{\mathrm{MacTag}_B\}$ to A.

Step 5: Upon receiving the message from B, A checks whether MacTag_B and $f(\mathrm{SK}, \mathrm{ID}_B, \mathrm{ID}_A, Q'_B, Q'_A)$ is equal. If not so, A stops the session; otherwise, B is authenticated, then A sets SK as the common session key.

B. Overview of Odelu et al.'s Security Protocol

In this section, we give the brief overview of Odelu *et al.*'s protocol [1], which is claimed to be a security improvement on He *et al.*'s protocol. The protocol can also be divided into two parts: 1) initialization and 2) mutual authentication and key agreement.

1) Initialization:

Step 1: The user A requests TSM for some pseudonyms.

Step 2: Upon receiving the request, TSM generates n random numbers for the user A (q_A^i, $i = 1, 2, \ldots, n$). Then TSM computes the ith public key, pseudonym, and private key, for $i = 1, 2, \ldots, n$, as follows:

$$Q_A^i = q_A^i G$$
$$P_A^i = \left\{Q_A^i \middle\| \mathrm{Enc}\left(d_{\mathrm{TSM}}, \left\{\mathrm{ID}_A, q_A^i\right\}\right) \middle\| \mathrm{ID}_{\mathrm{TSM}} \middle\| \mathrm{LT}_A^i \right\}$$
$$d_A^i = q_A^i + h\left(\mathrm{ID}_A, \mathrm{ID}_{\mathrm{TSM}}, P_A^i\right)d_{\mathrm{TSM}}$$

where LT_A^i is the lifetime window of P_A^i defined by TSM according to the security requirements. TSM stores the identity of user A and n pseudonyms into its database until expiration comes. Then TSM send all n pseudonyms and the corresponding private keys to A through a secure channel. User A stores the received pseudonyms and private keys.

2) Mutual Authentication and Key Agreement: When the user A wants to implement the mutual authentication and key agreement protocol with the user B, the following steps are executed.

Step 1: The user A randomly selects a pseudonym P_A^i and gets the corresponding private key d_A^i. Then A sends the message $M_1 = \{P_A^i\}$ to B.

Step 2: Upon receiving the message from A, B checks the validity of LT_A^i. If not so, A stops the session; otherwise, B randomly selects P_B^j and corresponding private key d_B^j. Then B generates a random number r_B and computes

$$R_B = h\left(r_B, d_B^j\right)\left(Q_A^i + h\left(\mathrm{ID}_A, \mathrm{ID}_{\mathrm{TSM}}, P_A^i\right)Q_{\mathrm{TSM}}\right).$$

Finally, B sends the message $M_2 = \{R_B, P_B^j\}$ to A.

Step 3: Upon receiving the message from B, A checks the validity of LT_B^j. If not so, B stops the session; otherwise, A generates a random number r_B and computes:

$$R_A = h\left(r_A^i, d_A^i\right)\left(Q_B^j + h\left(\mathrm{ID}_B, \mathrm{ID}_{\mathrm{TSM}}, P_B^j\right)Q_{\mathrm{TSM}}\right)$$
$$K_A = h\left(r_A, d_A^i\right)R_B/d_A^i = h\left(r_A, d_A^i\right)h\left(r_B, d_B^j\right)G$$
$$\mathrm{MacTag}_A = f\left(K_A, \mathrm{ID}_A, \mathrm{ID}_B, R_A, R_B\right).$$

Finally, A sends the message $M_3 = \{R_A, \mathrm{MacTag}_A\}$ to B.

Step 4: Upon receiving the message from A, B computes $K_B = h(r_B, d_B^j)R_A/d_B^j$, and check whether MacTag_A is equal to $f(K_B, \mathrm{ID}_A, \mathrm{ID}_B, R_A, R_B)$. If not so, B stops the session; otherwise, B computes $\mathrm{SK}_B = \mathrm{KDF}(K_B, R_A, R_B)$, $\mathrm{MacTag}_B = f(\mathrm{SK}_B, \mathrm{ID}_B, \mathrm{ID}_A, R_B, R_A)$, and sets SK_B as the common session key. Finally, B sends $\{\mathrm{MacTag}_B\}$ to A.

Step 5: Upon receiving the message from B, A computes $\mathrm{SK}_A = \mathrm{KDF}(K_A, R_A, R_B)$ and checks whether MacTag_B is equal to $f(\mathrm{SK}_A, \mathrm{ID}_B, \mathrm{ID}_A, R_B, R_A)$. If not so, A stops the session; otherwise, B is authenticated, then A sets SK_A as the common session key.

III. Security Analysis of Related Protocols

A. Security Weakness Analysis of He et al.'s Protocol

In order to address the design flaw, He *et al.* first redesigned the pseudonyms and the signatures, where in P_A^i, $\mathrm{Enc}(Q_A, d_A^i)$ is replaced as $\mathrm{Enc}(d_{\mathrm{TSM}}, \{\mathrm{ID}_A, Q_A^i\})$, and in S_{TSM}^i, $\mathrm{Enc}(d_A^i, Q_A)$ is replaced as $\mathrm{Enc}(d_{\mathrm{TSM}}, Q_A^i)$. In the protocol description, He *et al.*'s did not mention about how to take advantage of this new redesigned elements. Moreover, as $\mathrm{Enc}(d_{\mathrm{TSM}}, \{\mathrm{ID}_A, Q_A^i\})$ is computed with d_{TSM}, anyone can decrypt it with TSM's public key, and further know ID_A. Therefore, no matter which pseudonym is used, the implemented protocol procedure can be easily linked to A, which disobey the security requirements defined in [19]. Moreover, in Eun *et al.*'s protocol, $\mathrm{Enc}(Q_A, d_A^i)$ is used to protect d_A^i, however, He *et al.*'s did not mention about how to securely store A's private keys.

Odelu *et al.* [1] pointed out that the adversary who has valid pseudonym and private key pair can launch the internal impersonation attack. It is important to note that the goals of internal impersonation attack prevention and anonymity protection are against each other. Therefore, in order to provide anonymity protection, the internal impersonation attack is not the key security threat to He *et al.*'s protocol. However, because of the misusage of ID_A and ID_B, anyone can know users' real identities, and He *et al.*'s protocol cannot provide

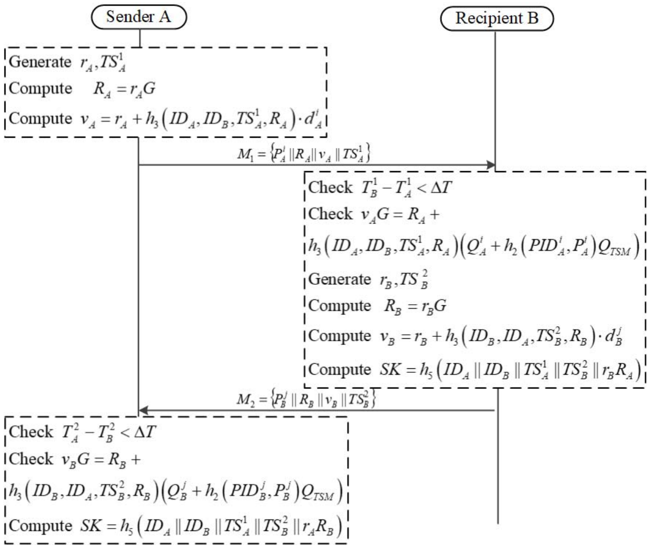

Sender A
Recipient B
Generate r_A, TS_A^1
Compute $R_A = r_A G$
Compute $v_A = r_A + h_3(ID_A, ID_B, TS_A^1, R_A) \cdot d_A^i$
$M_1 = \{P_A^i \| R_A \| v_A \| TS_A^1\}$
Check $T_B^1 - T_A^1 < \Delta T$
Check $v_A G = R_A +$
$h_3(ID_A, ID_B, TS_A^1, R_A)(Q_A^i + h_2(PID_A^i, P_A^i) Q_{TSM})$
Generate r_B, TS_B^2
Compute $R_B = r_B G$
Compute $v_B = r_B + h_3(ID_B, ID_A, TS_B^2, R_B) \cdot d_B^j$
Compute $SK = h_5(ID_A \| ID_B \| TS_A^1 \| TS_B^2 \| r_B R_A)$
$M_2 = \{P_B^j \| R_B \| v_B \| TS_B^2\}$
Check $T_A^2 - T_B^2 < \Delta T$
Check $v_B G = R_B +$
$h_3(ID_B, ID_A, TS_B^2, R_B)(Q_B^j + h_2(PID_B^j, P_B^j) Q_{TSM})$
Compute $SK = h_5(ID_A \| ID_B \| TS_A^1 \| TS_B^2 \| r_A R_B)$

Fig. 1. Proposed PSAP based on time synchronization.

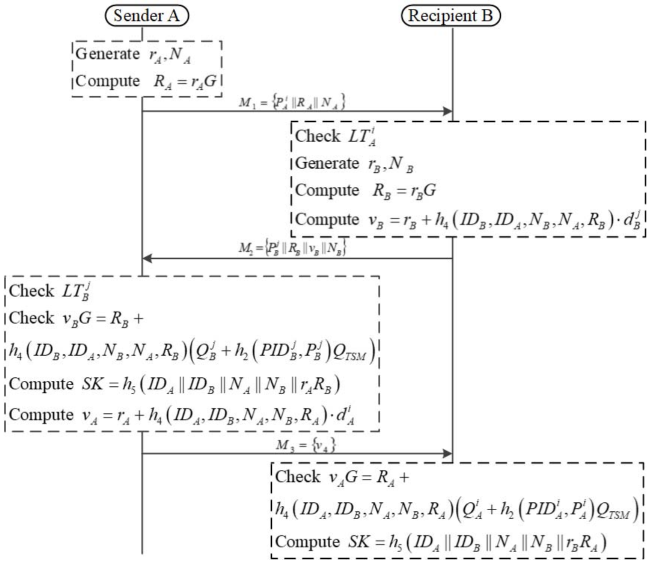

Sender A
Recipient B
Generate r_A, N_A
Compute $R_A = r_A G$
$M_1 = \{P_A^i \| R_A \| N_A\}$
Check LT_A^i
Generate r_B, N_B
Compute $R_B = r_B G$
Compute $v_B = r_B + h_4(ID_B, ID_A, N_B, N_A, R_B) \cdot d_B^j$
$M_2 = \{P_B^j \| R_B \| v_B \| N_B\}$
Check LT_B^j
Check $v_B G = R_B +$
$h_4(ID_B, ID_A, N_B, N_A, R_B)(Q_B^j + h_2(PID_B^j, P_B^j) Q_{TSM})$
Compute $SK = h_5(ID_A \| ID_B \| N_A \| N_B \| r_A R_B)$
Compute $v_A = r_A + h_4(ID_A, ID_B, N_A, N_B, R_A) \cdot d_A^i$
$M_3 = \{v_A\}$
Check $v_A G = R_A +$
$h_4(ID_A, ID_B, N_A, N_B, R_A)(Q_A^i + h_2(PID_A^i, P_A^i) Q_{TSM})$
Compute $SK = h_5(ID_A \| ID_B \| N_A \| N_B \| r_B R_A)$

Fig. 2. Proposed PSAP based on nonce.

anonymity protection. Actually, if ID_A and ID_B are defined as users' identities, and no need to provide anonymity protection, He *et al.*'s protocol is correct, and the attack case mentioned in Odelu *et al.*'s protocol [1] is invalid to it. There is no any internal adversary $\mathcal{C}$ can provide valid pseudonym including $\mathrm{Enc}(d_{\mathrm{TSM}}, \{\mathrm{ID}_A, Q_A^i\})$ to impersonate A, which can be verified by any corresponding user, although this verification has not been mentioned in He *et al.*'s work [20]. However, He *et al.*'s scheme cannot provide anonymity protection and unlinkability of context messages.

B. Security Weakness Analysis of Odelu et al.'s Protocol

Based on the work of Eun *et al.* [19] and He *et al.* [20], Odelu further tried to propose a secure and efficient authentication protocol. However, the scheme is still not secure.

Odelu first redesigned the pseudonyms, where P_A^i and $\mathrm{Enc}(d_{\mathrm{TSM}}, \{\mathrm{ID}_A, Q_A^i\})$ in He *et al.*'s protocol [20] is replaced as $\mathrm{Enc}(d_{\mathrm{TSM}}, \{\mathrm{ID}_A, q_A^i\})$. It will result in that anyone can decrypt this element with TSM's public key, and further know ID_A and q_A^i. Therefore, no matter which pseudonym is used, the implemented protocol procedure can be easily linked to A, which disobeys the security requirements [19]. More seriously, from an arbitrary pseudonym, any internal adversary can compute to get d_{TSM}. For example, assume the adversary $\mathcal{C}$ disguises as a legitimate user, and obtains multiple pseudonyms, and the corresponding $d_{\mathcal{C}}^i$. Randomly choose a pseudonym $P_{\mathcal{C}}^i$, and decrypt it with Q_{TSM} to get $q_{\mathcal{C}}^i$. After that, based on $d_A^i = q_A^i + h(\mathrm{ID}_A, \mathrm{ID}_{\mathrm{TSM}}, P_A^i) d_{\mathrm{TSM}}$, the adversary can compute to get TSM's private key d_{TSM}, which will bring about potential security risk of the whole system.

Meanwhile, because of no signature protection, internal adversary can modify LT_A^i, the verification process of LT_A^i in the phase of "mutual authentication and key agreement" as described in Section II-C2 does not make sense.

IV. Our Proposed Protocol PSAP

In this section, we will describe an improved protocol to meet the requirements of mutual authentication and key agreement for NFC applications, while keeping untraceable and anonymous. Our protocol contains three phases.

1) Initialization phase.
2) Mutual authentication and key agreement phase.
3) Reveal the identity of internal attackers.

In mutual authentication and key agreement, we introduce two different methods: 1) timestamp-based and 2) nonce-based.

A. Initialization

Step 1: The user A sends his/her identity UID_A and public key Q_A to TSM and request for n pseudonyms.

Step 2: Upon receiving the request, TSM generate n random numbers for the user A (q_A^i, $i = 1, 2, \ldots, n$). Then TSM computes the ith public key, pseudonym identity, pseudonym, private key, and encryption protected private key, for $i = 1, 2, \ldots, n$, as follows:

$$
\begin{aligned}
Q_A^i &= q_A^i G \\
\mathrm{PID}_A^i &= h_1(\mathrm{UID}_A, \mathrm{ID}_{\mathrm{TSM}}, Q_A^i) \\
P_A^i &= \{Q_A^i \| \mathrm{PID}_A^i \| \mathrm{Enc}(Q_{\mathrm{TSM}}, \{\mathrm{PID}_A^i, \mathrm{UID}_A\}) \| \mathrm{ID}_{\mathrm{TSM}} \| \mathrm{LT}_A^i\} \\
d_A^i &= q_A^i + h_2(\mathrm{PID}_A^i, P_A^i) d_{\mathrm{TSM}} \\
Ed_A^i &= \mathrm{Enc}(Q_A, d_A^i)
\end{aligned}
$$

where LT_A^i is the lifetime window of P_A^i defined by TSM according to the security requirements. TSM does not need to store any information. Then TSM sends all n $\{P_A^i, Ed_A^i\}$ back to A. The user A stores the received $\{P_A^i, Ed_A^i\}$, $i = 1, 2, \ldots, n$.

B. Mutual Authentication and Key Agreement

Here, we give two different methods to implement mutual authentication and key agreement between users A and B. The first one relies on time synchronization (as shown in Fig. 1), while the second one relies on using nonce (as shown in Fig. 2).

1) Time Synchronization-Based: The phase of time synchronization based mutual authentication and key agreement includes the following steps:

Step 1: The user A randomly selects a pseudonym P_A^i and corresponding Ed_A^i. The user A decrypts Ed_A^i to get his/her private key d_A^i. A generates a random number r_A and a timestamp TS_A^i. Then A computes $R_A = r_A G$, $v_A = r_A + h_3(\text{ID}_A, \text{ID}_B, TS_A^1, R_A)d_A^i$. Finally, A sends the message $M_1 = \{P_A^i||R_A||v_A||TS_A^1\}$ to B.

Step 2: Upon receiving the message from A, B first verifies whether T_A^1 (related to TS_A^1) and the current time T_B^1 are involved in LT_A^i and makes sure $T_B^1 - T_A^1 < \Delta T$, where ΔT is the allowed time interval. If not so, B stops the session; otherwise, B checks whether $v_A G$ is equal to $R_A + h_3(\text{ID}_A, \text{ID}_B, TS_A^1, R_A)(Q_A^i + h_2(\text{PID}_A^i, P_A^i)Q_{\text{TSM}})$. If not so, B stops the session; otherwise, B generates a random number r_B and a timestamp TS_B^2. Then B randomly selects a pseudonym P_B^j and corresponding Ed_B^j B decrypts Ed_B^j to get his/her private key d_B^j. After that, B computes $R_B = r_B G$, $v_B = r_B + h_3(\text{ID}_B, \text{ID}_A, TS_B^2, R_B)d_B^j$. Finally, B sets the common session key as $\text{SK} = h_5(ID_A||ID_B||TS_A^1||TS_B^2||r_B R_A)$, and sends the message $M_2 = \{P_B^j||R_B||v_B||TS_B^2\}$ to A.

Step 3: Upon receiving the message from B, A first verifies whether T_B^j (related to TS_B^j) and the current time T_A^2 are involved in LT_B^j and makes sure $T_A^2 - T_B^2 < \Delta T$. If not so, A stops the session; otherwise, A checks whether $v_B G$ and $R_B + h_3(\text{ID}_B, \text{ID}_A, TS_B^2, R_B)(Q_B^j + h_2(\text{PID}_B^j, P_B^j)Q_{\text{TSM}})$ are equal. If not so, A stops the session; otherwise, A computes the common session key: $\text{SK} = h_5(\text{ID}_A||\text{ID}_B||TS_A^1||TS_B^2||r_A R_B)$.

2) Nonce-Based: The phase of nonce based mutual authentication and key agreement includes the following steps:

Step 1: The user A randomly selects a pseudonym P_A^i and corresponding Ed_A^i. The user A decrypts Ed_A^i to get his/her private key d_A^i. A generates a random number r_A and a nonce N_A. Then A computes $R_A = r_A G$. Finally, A sends the message $M_1 = \{P_A^i||R_A||N_A\}$ to B.

Step 2: Upon receiving the message from A, B first verifies whether the current time are involved in LT_A^i. If not so, B stops the session; otherwise, B generates a random number r_B and a nonce N_B. Then B randomly selects a pseudonym P_B^j and the corresponding Ed_B^j. B decrypts Ed_B^j to get his/her private key d_B^j. After that, B computes $R_B = r_B G$, $v_B = r_B + h_4(\text{ID}_B, \text{ID}_A, N_B, N_A, R_B)d_B^j$. Finally, B sends the message $M_2 = \{P_B^j||R_B||N_B||v_B\}$ to A.

Step 3: Upon receiving the message from B, A verifies whether the current time is involved in LT_B^j. If not so, A stops the session; otherwise, A checks whether $v_B G$ is equal to $R_B + h_4(\text{ID}_B, \text{ID}_A, N_B, N_A, R_B)(Q_B^j + h_2(\text{PID}_B^j, P_B^j)Q_{\text{TSM}})$. If not so, A stops the session; otherwise, A computes the common session key $\text{SK} = h_5(\text{ID}_A||\text{ID}_B||N_A||N_B||r_A R_B)$, then computes $v_A = r_A + h_4(\text{ID}_A, \text{ID}_B, N_A, N_B, R_A)d_A^j$. Finally, A sends the message $M_3 = \{v_A\}$ to B.

Step 4: Upon receiving the message from A, B verifies whether $v_A G$ and $R_A + h_4(\text{ID}_A, \text{ID}_B, N_A, N_B, R_A)(Q_A^i + h_2(\text{PID}_A^i, P_A^j)Q_{\text{TSM}})$ are equal. If not so, B stops the session; otherwise, B computes the common session key $\text{SK} = h_5(\text{ID}_A||\text{ID}_B||N_A||N_B||r_B R_A)$.

Although the timestamp and nonce are used to resist replay attacks, they have their own advantages and disadvantages. The timestamp-based method requires time synchronization, but only has two rounds of message exchange. The nonce-based method needs three rounds of message exchange. Using nonce method does not require time synchronization but need to compare with the past nonce stored in the database. We need to use different methods under different conditions.

C. Reveal the Identity of Internal Attackers

Furthermore, once the communication party (user A) detects the other side's (user B) malicious behavior, A sends a request to TSM, which contents P_B^j and evidence of malicious behavior, to get the identity of B. If TSM confirms malicious behavior, TSM releases UID_B to A by decrypting the $\text{Enc}(Q_{\text{TSM}}, \{\text{PID}_A^i, \text{UID}_A\})$. The user A stores UID_B in its database, through the hash calculation of $\text{PID}_B^j = h_1(\text{UID}_B, \text{ID}_{\text{TSM}}, Q_B^j)$, user A can distinguish whether this is an illegal user afterwards. If so, A stops the session.

V. Security Analysis of Our Proposed PSAP

In this section, we analyze the security features of our proposed protocol as follows.

A. Communication Link Confidentiality

In PSAP, the session key can be, respectively, computed as $\text{SK} = h_5(\text{ID}_A||\text{ID}_B||TS_A^1||TS_B^2||r_A R_B) = h_5(\text{ID}_A||\text{ID}_B||TS_A^1||TS_B^2||r_B R_A)$, where r_A and r_B are randomly generated by initiator A and target B. Based on the hardness of elliptic curve discrete logarithm problem, even if attackers get the R_A or R_B, he cannot figure out the r_A or r_B. It means the adversary cannot compute SK without r_A or r_B. Therefore, the proposed protocol can provide session key security.

Furthermore, since users generate the new random number r_A or r_B each time, there is no correlation among all of the session keys. As the independence between different session keys, even if the disclosure of the current session key, the backward or the forward session keys still remain secure. The compromise of the current session key would not affect the security of the other keys. Hence, the agreement can provide perfect forward security and the perfect backward security.

B. Anonymity and Unlikability

The proposed scheme can preserve the user's identity from being exposed to any other entity, including the other side of communication. Pseudonyms, instead of the user's identity, are used for the communication between users. When initiator A communicates with target B, B only knows P_A^i, $P_A^i = \{Q_A^i||\text{PID}_A^i||\text{Enc}(Q_{\text{TSM}}, \{\text{PID}_A^i, \text{UID}_A\})||\text{ID}_{\text{TSM}}||\text{LT}_A^i\}$. The user A's identity is encrypted by the TSM and hashed with Q_A^i, ID_{TSM}. Due to the one-way property of hash function and the security of the TSM's private key, no one except TSM can decrypt P_A^i to get the user's identity UID_A.

Moreover, for each session, user A randomly selects P_A^i. P_A^i is composed of Q_A^i, PID_A^i, $\text{Enc}(Q_{\text{TSM}}, \{\text{PID}_A^i, \text{UID}_A\})$, ID_{TSM} and LT_A^i. When i takes different values, for example, u and v, there is no linkage between Q_A^u and Q_A^v, PID_A^u and PID_A^v, LT_A^u and LT_A^v, $\text{Enc}(Q_{\text{TSM}}, \{\text{PID}_A^u, \text{UID}_A\})$ and

$\mathrm{Enc}(Q_{\mathrm{TSM}}, \{\mathrm{PID}_A^v, \mathrm{UID}_A\})$ for attackers, and all users have the same $\mathrm{ID}_{\mathrm{TSM}}$. Thus, attackers cannot link two sessions which are initiated by the same user. To sum up, our scheme could provide user anonymity and unlikability.

C. Traceability

According to the phase of *reveal the identity of internal attackers,* in our proposed PSAP, TSM can find out who has launched an internal attack. Once an internal attack is detected, the attacked user sends the evidences of the internal attack and the suspected attacker's pseudonyms P_C^i to TSM, which contains the encrypted form of the suspected attacker's identity $\mathrm{Enc}(Q_{\mathrm{TSM}}, \{\mathrm{PID}_C^i, \mathrm{UID}_C\})$. After confirming the malicious behavior, TSM uses its private key to decrypt $\mathrm{Enc}(Q_{\mathrm{TSM}}, \{\mathrm{PID}_C^i, \mathrm{UID}_C\})$ to obtain the identity of the internal attacker (UID_C^i). Thus, TSM can reveal and trace the internal attacker's identity.

D. Resistance of Various Attacks

1) Impersonation Attack: Our proposed scheme can provide mutual authentication against impersonation attacks. In synchronization-based method or the nonce-based method, target B authenticates the initiator A by verifying

$$v_A G \overset{?}{=} R_A + h_3\left(\mathrm{ID}_A, \mathrm{ID}_B, TS_A^1, R_A\right)\left(Q_A^i + h_2\left(\mathrm{PID}_A^i, P_A^i\right)Q_{\mathrm{TSM}}\right) \text{ or}$$

$$v_A G \overset{?}{=} R_A + h_4(\mathrm{ID}_A, \mathrm{ID}_B, N_A, N_B, R_A)\left(Q_A^i + h_2\left(\mathrm{PID}_A^i, P_A^i\right)Q_{\mathrm{TSM}}\right).$$

Since the security of this authentication mechanism is based on elliptic curve discrete logarithm problem, the C without the authorized private key d_A^i cannot forge a feasible $v_A = r_A + h_3(\mathrm{ID}_A, \mathrm{ID}_B, TS_A^1, R_A)d_A^i$ to pass the verification by user B. Similarly, the attacker cannot forge a feasible $v_B = r_B + h_3(\mathrm{ID}_B, \mathrm{ID}_A, TS_B^2, R_B)d_B^j$ for the verification of identity legitimacy, since user A can verify the legitimacy of user B by checking

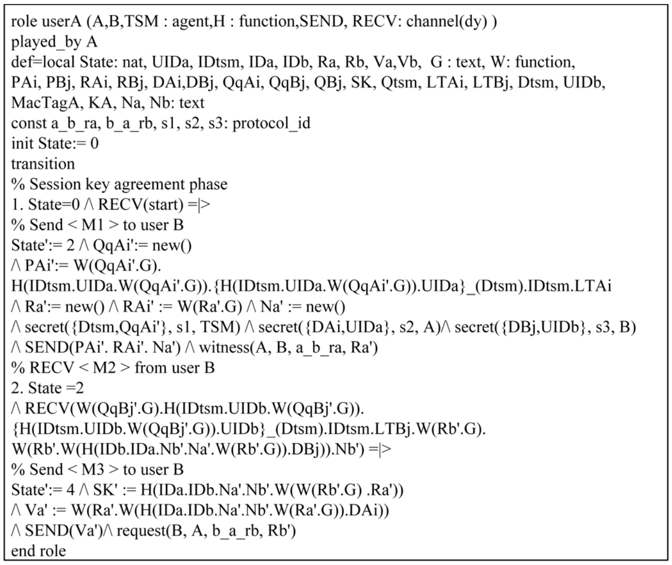
$$v_B G \overset{?}{=} R_B + h_3\left(\mathrm{ID}_B, \mathrm{ID}_A, TS_B^2, R_B\right)\left(Q_B^j + h_2\left(\mathrm{PID}_B^j, P_B^j\right)Q_{\mathrm{TSM}}\right) \text{ or}$$

$$v_B G \overset{?}{=} R_B + h_4(\mathrm{ID}_B, \mathrm{ID}_A, N_B, N_A, R_B)\left(Q_B^j + h_2\left(\mathrm{PID}_B^j, P_B^j\right)Q_{\mathrm{TSM}}\right).$$

Therefore, our scheme successfully resists impersonation attacks.

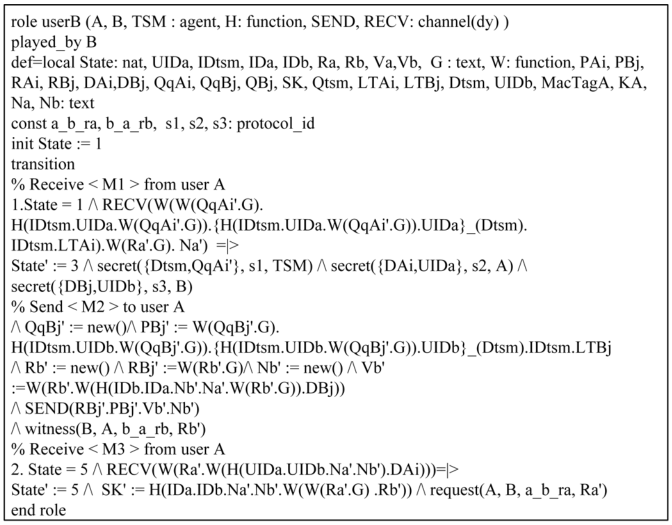
2) Man-in-the-Middle Attack: In a man-in-the-middle attack, two parties are tricked into a three-party communication. From the proof of impersonation attack, a malicious attacker both fails to impersonate user A to user B and impersonate user B to user A. Therefore, the protocol can withstand man-in-the-middle attacks.

3) Replay Attack: A replay attack refers to that the adversary sends any messages, which have been transmitted already, to the target again. In our scheme, we introduce timestamp or nonce to address the replay attack. For example, in time synchronization-based method, TS_A^1 is included in $v_A = r_A + h_3(\mathrm{ID}_A, \mathrm{ID}_B, TS_A^1, R_A)d_A^i$, which cannot be generated without d_A^i and modified since the one-way hash function. Once an attacker launches replay attacks, the target B can detect attack by checking the validity of T_A^1 (related to TS_A^1). If the current time T_B^1 does not satisfy $T_B^1 - T_A^1 < \Delta T$, the target will interrupt the session immediately. Similarly, in nonce-based method, the replay attack would be detected by checking the nonce. Therefore, our scheme can resist the replay attack.



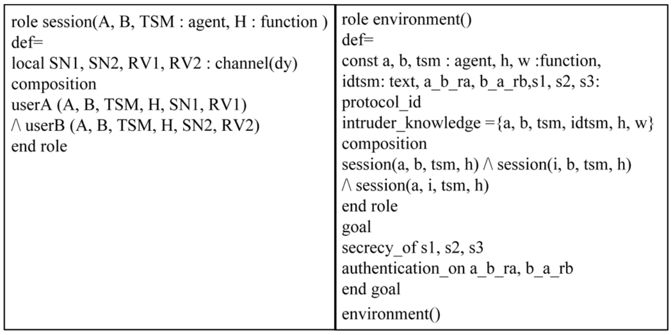
```
role userA (A,B,TSM : agent,H : function,SEND, RECV: channel(dy) )
played_by A
def=local State: nat, UIDa, IDtsm, IDa, IDb, Ra, Rb, Va,Vb,  G : text, W: function,
PAi, PBj, RAi, RBj, DAi,DBj, QqAi, QqBj, QBj, SK, Qtsm, LTAi, LTBj, Dtsm, UIDb,
MacTagA, KA, Na, Nb: text
const a_b_ra, b_a_rb, s1, s2, s3: protocol_id
init State:= 0
transition
% Session key agreement phase
1. State=0 /\ RECV(start) =|>
% Send < M1 > to user B
State':= 2 /\ QqAi':= new()
/\ PAi':= W(QqAi'.G).
H(IDtsm.UIDa.W(QqAi'.G)).{H(IDtsm.UIDa.W(QqAi'.G)).UIDa}_(Dtsm).IDtsm.LTAi
/\ Ra':= new() /\ RAi' := W(Ra'.G) /\ Na' := new()
/\ secret({Dtsm,QqAi'}, s1, TSM) /\ secret({DAi,UIDa}, s2, A)/\ secret({DBj,UIDb}, s3, B)
/\ SEND(PAi'. RAi'. Na') /\ witness(A, B, a_b_ra, Ra')
% RECV < M2 > from user B
2. State =2
/\ RECV(W(QqBj'.G).H(IDtsm.UIDb.W(QqBj'.G)).
{H(IDtsm.UIDb.W(QqBj'.G)).UIDb}_(Dtsm).IDtsm.LTBj.W(Rb'.G).
W(Rb'.W(H(IDb.IDa.Nb'.Na'.W(Rb'.G)).DBj)).Nb') =|>
% Send < M3 > to user B
State':= 4 /\ SK' := H(IDa.IDb.Na'.Nb'.W(W(Rb'.G) .Ra'))
/\ Va' := W(Ra'.W(H(IDa.IDb.Na'.Nb'.W(Ra'.G)).DAi))
/\ SEND(Va')/\ request(B, A, b_a_rb, Rb')
end role
```

Fig. 3. Role for the user *A* of PSAP in HLPSL.



```
role userB (A, B, TSM : agent, H: function, SEND, RECV: channel(dy) )
played_by B
def=local State: nat, UIDa, IDtsm, IDa, IDb, Ra, Rb, Va,Vb,  G : text, W: function, PAi, PBj,
RAi, RBj, DAi,DBj, QqAi, QqBj, QBj, SK, Qtsm, LTAi, LTBj, Dtsm, UIDb, MacTagA, KA,
Na, Nb: text
const a_b_ra, b_a_rb,  s1, s2, s3: protocol_id
init State := 1
transition
% Receive < M1 > from user A
1.State = 1 /\ RECV(W(W(QqAi'.G).
H(IDtsm.UIDa.W(QqAi'.G)).{H(IDtsm.UIDa.W(QqAi'.G)).UIDa}_(Dtsm).
IDtsm.LTAi).W(Ra'.G). Na')  =|>
State' := 3 /\ secret({Dtsm,QqAi'}, s1, TSM) /\ secret({DAi,UIDa}, s2, A) /\
secret({DBj,UIDb}, s3, B)
% Send < M2 > to user A
/\ QqBj' := new()/\ PBj' := W(QqBj'.G).
H(IDtsm.UIDb.W(QqBj'.G)).{H(IDtsm.UIDb.W(QqBj'.G)).UIDb}_(Dtsm).IDtsm.LTBj
/\ Rb' := new() /\ RBj' :=W(Rb'.G)/\ Nb' := new() /\ Vb'
:=W(Rb'.W(H(IDb.IDa.Nb'.Na'.W(Rb'.G)).DBj))
/\ SEND(RBj'.PBj'.Vb'.Nb')
/\ witness(B, A, b_a_rb, Rb')
% Receive < M3 > from user A
2. State = 5 /\ RECV(W(Ra'.W(H(UIDa.UIDb.Na'.Nb').DAi)))=|>
State' := 5 /\  SK' := H(IDa.IDb.Na'.Nb'.W(W(Ra'.G) .Rb')) /\ request(A, B, a_b_ra, Ra')
end role
```

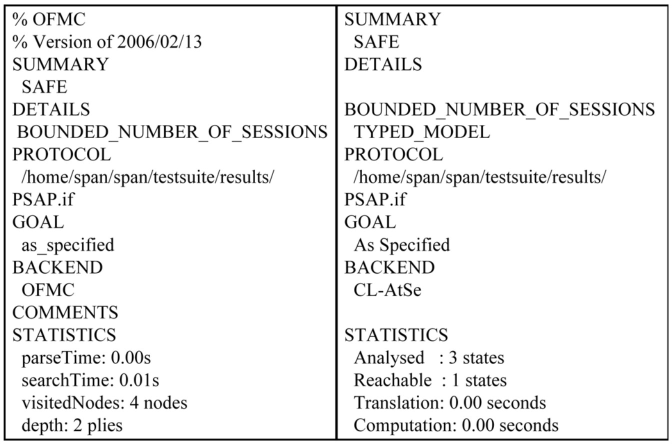
Fig. 4. Role for the user *B* of PSAP in HLPSL.



```
role session(A, B, TSM : agent, H : function )
def=
local SN1, SN2, RV1, RV2 : channel(dy)
composition
userA (A, B, TSM, H, SN1, RV1)
/\ userB (A, B, TSM, H, SN2, RV2)
end role
```

```
role environment()
def=
const a, b, tsm : agent, h, w :function,
idtsm: text, a_b_ra, b_a_rb,s1, s2, s3:
protocol_id
intruder_knowledge ={a, b, tsm, idtsm, h, w}
composition
session(a, b, tsm, h) /\ session(i, b, tsm, h)
/\ session(a, i, tsm, h)
end role
goal
secrecy_of s1, s2, s3
authentication_on a_b_ra, b_a_rb
end goal
environment()
```

Fig. 5. Role for the session, and goal and environment.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/
PSAP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 4 nodes
  depth: 2 plies
```

```
SUMMARY
  SAFE
DETAILS

BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/
PSAP.if
GOAL
  As Specified
BACKEND
  CL-AtSe

STATISTICS
  Analysed   : 3 states
  Reachable  : 1 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Fig. 6. Simulation result of the PSAP using OFMC and CL-AtSe backends.

4) Modification Attack: The private key d_A^i issued by TSM is related to q_A^i, UID_A, $\mathrm{ID}_{\mathrm{TSM}}$, LT_A^i, Q_A^i, and d_{TSM}. Malicious users cannot modify any part of them without the help of TSM. Take a concrete example as an illustration. If a malicious user A modifies the lifetime LT_A^i, P_A^i should also be modified, but he/she cannot compute the corresponding modification of d_A^i without the knowledge of d_{TSM}. As the user cannot generate a legal key pair $\{Q_A^i, d_A^i\}$ without TSM's assistance, he/she cannot pass the mutual authentication with an illegal key pair. Therefore, the proposed security protocol can withstand the modification attack.

VI. Security Verification Using AVISPA Tool

In addition to proving the security features of our proposed protocol, in this section, we also provide a formal analysis using automated validation of Internet security protocols and applications (AVISPAs) [21]. The AVISPA is aimed at specifying cryptographic protocols and analyze their security properties by looking for attacks on specified scenarios. We choose this tool because it has the following advantages: providing a modular and expressive form language [high-level protocol specification language (HLPSL)], integrating different back-ends that implement a variety of automatic analysis techniques ranging from protocol falsification and no other tool exhibits the same scope and robustness while enjoying the same performance and scalability [22].

In AVISPA, the protocol should be specified in HLPSL, including each role participating in the protocol, the protocol session and the execution environment, and so on. On-the-fly model-checker (OFMC) and constraint-logic-based attack searcher (CL-AtSe) are two backends integrated in AVISPA, which assume that there is an active Dolev–Yao intruder.

As shown in Figs. 3–5, we translate our proposed protocol into HLPSL, and run with the security protocol animator for AVISPA. The experimental result in Fig. 6 shows that our proposed PSAP succeeds in resisting Dolev-Yao intruder, which can launch eavesdrop, interception, modification, or replay attacks.

TABLE II
Some More Notations and Description in This Section

Notation	Description
T_m	The time of running a modular multiplication operation
T_{em}	The time of running an elliptic curve point multiplication operation
T_{ea}	The time of running an elliptic curve point addition operation
T_h	The time of running a hash function operation
T_{kdf}	The time of running a key derivation function operation
T_{inv}	The time of running a modular inverse operation.

TABLE III
Comparison of Computation Overhead (Initiator and Target)

Protocol	Performance cost	Total (ms)
[14]	$2T_h$+$2T_{kdf}$	0.0004
[19]	$4T_m$+$6T_{em}$+$2T_{ea}$+$4T_h$+$2T_{kdf}$	3.6030
[20]	$8T_{em}$+$2T_{ea}$+$6T_h$+$2T_{kdf}$	4.8028
[1]	$6T_{em}$+$2T_{ea}$+$8T_h$+$2T_{kdf}$+$2T_{inv}$	3.6032
PSAP (Timestamp)	$2T_m$+ $8T_{em}$+ $4T_{ea}$+ $6T_h$	4.8048
PSAP (Nonce)	$2T_m$+ $8T_{em}$+ $4T_{ea}$+ $6T_h$	4.8048

VII. Performance Analysis

In this section, we analyze the computational and communication performance of PSAP. For clarity, notations, and descriptions are defined in Table II.

A. Computation Overhead Analysis

Obviously, the computation overheads of the above operations are different. According to the work of Chatterjee *et al.* [23], we can quantify the computational cost of the main operation. More specifically, if T_m is the benchmark, then, we can further get $T_{\mathrm{em}} \approx 1200T_m$, $T_{\mathrm{ea}} \approx 5T_m$, $T_h \approx T_{\mathrm{kdf}}$ and $T_h \approx 0.36T_m$, $T_{\mathrm{inv}} \approx 3T_m$.

Because the speed of verification is mainly composed of six operations referred to in Table II, we ignore other operations. We chose to test the time of protocol operations with our computer, Intel P IV 3.0 GHz Machine. Here, we adopt the experiment in [24] for an MNT curve of embedding degree $k = 6$ and 160-bit q. Through multiple tests and taking the average value, the following experiment results are obtained: T_{em} is 0.6 ms, T_{ea} is 0.001 ms, T_m, T_h, T_{kdf}, and T_{inv} are 0.0001 ms. The comparison with the related protocols is shown in Table III.

From Table III, the computation cost of our proposed PSAP protocol is close to PBNFC, but a little more than CPPNFC and SEAP. The main reason for the increase in computational overhead is that we use the difficulty of the elliptic curve discrete logarithm problem to guarantee the security of the protocol, which costs computation overhead a lot.

B. Communication Overhead Analysis

According to the work of Odelu *et al.* [1] and Eun *et al.* [19], assuming the length of timestamp is 32 bits, the size of the parameters used in NFC protocol is shown as follows: $\mathrm{ID}_{\mathrm{TSM}}$

TABLE IV
COMPARISON OF COMMUNICATION OVERHEAD

Protocol	Pseudonym Size (bit)	Communication overhead (bit)
[14]	0	784
[19]	1200	1184
[20]	1200	3184
[1]	624	1840
PSAP (Timestamp)	912	3040
PSAP (Nonce)	912	3168

TABLE V
SECURITY FUNCTIONALITY COMPARISON BETWEEN PSAP AND RELATED WORK

Protocol	[14]	[19]	[20]	[1]	PSAP
Mutual Authentication	YES	NO	NO	NO	YES
Anonymity	NO	NO	NO	NO	YES
Untraceability	NO	NO	NO	NO	YES
Conditional Privacy Preservation	NO	NO	NO	NO	YES
Session Key Security	YES	NO	NO	NO	YES
User Information Storage	NO	YES	YES	YES	NO
Easily scheduled revocation	NO	NO	NO	NO	YES
Dynamic scheduled revocation	NO	NO	NO	NO	YES

is 16 bits, LT$_X$ is 32 bits, N_X,R_X, and Mactag$_X$ are 96 bits, SK is 16 bits, D_X, z, and v_X are 192 bits, QX, QX', and QX'' are 200 bits, $\mathrm{Enc}(Q_X, d_X)$ is 352 bits, Q_X is 384 bits, and S_{TSM} is 448 bits. We can compute the total communication cost of PSAP, NFC-SEC [14], CPPNFC [19], PBNFC [20], and SEAP [1]. The comparison results are shown in Table IV.

Obviously, the communication cost of PSAP is a little more than SEAP and CPPNFC, but a little less than PBNFCP. The main reason is that messages include pseudonym, which contains $\mathrm{Enc}(Q_{\mathrm{TSM}}, \{\mathrm{PID}_A^i, \mathrm{UID}_A\})$ to get the user's identity, to prevent any internal attackers.

C. Storage Overhead Analysis

In CPPNFC, the pseudonym is composed of the user's public key, the encrypted privacy key, the identity of the TSM and TSM's signature.

The length of the pseudonym = the length of user's public key + encrypted privacy key + identity of the TSM + TSM's signature = 1200 bits

The TSM must store the identity of users and n pseudonyms into its database in CPPNFC, PBNFC, and SEAP. When TSM contains a large number of users, users' pseudonym is a large storage overhead. However, in our scheme, TSM does not require the storage of user's identity and corresponding pseudonyms. For TSM, pseudonym can be calculated by user's identity and public key. User's identity can also be calculated from anyone of his/her pseudonym.

D. Security Functionality Comparison

Finally, Table V shows the security functionality comparisons of PSAP and the existing protocols [1], [13], [19], [20]. It can be drawn the conclusion from Table V that only PSAP can meet mutual authentication, user anonymity and untraceability, and other security functionalities, without the need to store user information.

VIII. CONCLUSION

In this paper, we analyze the security of the related NFC application protocols that they still have security flaws, including the confusion of the user's identity and the random identity in NFC. We further propose a PSAP for NFC applications. In comparison with the related NFC security protocols, our proposed PSAP can not only protect legal user's anonymity but also reveal the identity of internal attackers. PSAP with acceptable computation and communication overhead and less storage overhead contribute to the promotion of NFC communication.

REFERENCES

[1] V. Odelu, A. K. Das, and A. Goswami, "SEAP: Secure and efficient authentication protocol for NFC applications using pseudonyms," *IEEE Trans. Consum. Electron.*, vol. 62, no. 1, pp. 30–38, Feb. 2016.

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

[3] S. Ghosh *et al.*, "Swing-pay: One card meets all user payment and identity needs: A digital card module using NFC and biometric authentication for peer-to-peer payment," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 82–93, Jan. 2017.

[4] T. Dahlberg, J. Guo, and J. Ondrus, "A critical review of mobile payment research," *Electron. Commerce Res. Appl.*, vol. 14, no. 5, pp. 265–284, 2015.

[5] J. Morak and G. Schreier, "Design and evaluation of near field communication (NFC) technology based solutions for mHealth challenges," in *Mobile Health*. Cham, Switzerland: Springer, 2015, pp. 813–838.

[6] W. D. Yu, H. Hansrao, K. Dhillon, and P. Desinguraj, "NFC based m-healthcare application focusing on security, privacy and performance," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, 2013, pp. 4104–4109.

[7] E. Kyriacou *et al.*, "Health and rescue services management system during a crisis event," *Healthcare Technol. Lett.*, vol. 3, no. 3, pp. 205–211, 2016.

[8] Information Technology—Telecommunications and Information Exchange Between Systems—Near Field Communication—Interface and Protocol (NFCIP-1), ISO/IEC Standard 18092-2, 2013.

[9] J. Fischer, "NFC in cell phones: The new paradigm for an interactive world [near-field communications]," *IEEE Commun. Mag.*, vol. 47, no. 6, pp. 22–28, Jun. 2009.

[10] W. Fan, W. Huang, Z. Zhang, Y. Wang, and D. Sun, "A near field communication (NFC) security model based on OSI reference model," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, 2015, pp. 1324–1328.

[11] N. Alexiou, S. Basagiannis, and S. Petridou, "Formal security analysis of near field communication using model checking," *Comput. Security*, vol. 60, pp. 1–14, Jul. 2016.

[12] Information Technology—Security Techniques—Cryptographic Techniques Based on Elliptic Curves—Part 1: General, ISO/IEC Standard FDIS 15946-1, 2008.

[13] Information Technology—Telecommunications and Information Exchange Between Systems—NFC Security—Part 1: NFC-SEC NFCIP-1 Security Services and Protocol, ISO/IEC Standard 13157-1, 2014.

[14] Information Technology—Telecommunications and Information Exchange Between Systems—NFC Security—Part 2: NFC-SEC Cryptography Standard Using ECDH and AES, ISO/IEC Standard 13157-2, 2016.

[15] K. Xue, P. Hong, and C. Ma, “A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture,” *J. Comput. Syst. Sci.*, vol. 80, no. 1, pp. 195–206, 2014.

[16] P. Gope and T. Hwang, “A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7124–7132, Nov. 2016.

[17] R. Yu *et al.*, “MixGroup: Accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks,” *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 1, pp. 93–105, Jan./Feb. 2016.

[18] D. Huang, S. Misra, M. Verma, and G. Xue, “PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 736–746, Sep. 2011.

[19] H. Eun, H. Lee, and H. Oh, “Conditional privacy preserving security protocol for NFC applications,” *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 153–160, Feb. 2013.

[20] D. He, N. Kumar, and J.-H. Lee, “Secure pseudonym-based near field communication protocol for the consumer Internet of Things,” *IEEE Trans. Consum. Electron.*, vol. 61, no. 1, pp. 56–62, Feb. 2015.

[21] *AVISPA. Automated Validation of Internet Security Protocols and Applications*. Accessed: Mar. 2018. [Online]. Available: http://www.avispaproject.org/

[22] A. Armando *et al.*, “The AVISPA tool for the automated validation of Internet security protocols and applications,” in *Proc. Int. Conf. Comput. Aided Verification*, Edinburgh, U.K., 2005, pp. 281–285.

[23] S. Chatterjee, A. K. Das, and J. K. Sing, “A survey on user access control in wireless sensor networks with formal security verification,” *Int. J. Trust Manag. Comput. Commun.*, vol. 2, no. 3, pp. 259–295, 2014.

[24] M. Scott. *Efficient Implementation of Cryptographic Pairings*. Accessed: Mar. 2018. [Online]. Available: http://ecrypt-ss07.rhul.ac.uk/Slides/Thursday/mscottsamos07.pdf

Jie Xu received the B.S. degree from the Department of Information Security, University of Science and Technology of China, Hefei, China, in 2017, where she is currently pursuing the graduation degree in communication and information system with the Department of Electronic Engineering and Information Science.

Her current research interest includes network security protocol design and analysis.

Kaiping Xue (M’09–SM’15) received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), Hefei, China, in 2003 and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007.

He is currently an Associate Professor with the Department of Information Security and Department of EEIS, USTC. His current research interests include future Internet, distributed networks, and network security.

Qingyou Yang received the B.S. degree in information security from the School of the Gifted Young, University of Science and Technology of China, Hefei, China, in 2016, where he is currently pursuing the graduation degree in communication and information system with the Department of Electronic Engineering and Information Science.

His current research interests include network security and cryptography.

Peilin Hong received the B.S. and M.S. degrees from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), Hefei, China, in 1983 and 1986, respectively.

She is currently a Professor and an Advisor for Ph.D. candidates with the Department of EEIS, USTC. She has published two books and over 150 academic papers in several journals and conference proceedings. Her current research interests include next-generation Internet, policy control, IP QoS, and information security.