SECURED DATA TRANSMISSION USING WIRELESS NETWORKS

S.Nithya,D.Rajesh,S.Selvapriya,E.A.Jayanth,
UG Scholar,
Department of ECE,
Nandha College of Technology.

Ms.R.Sharmila,
Assistant Professor,
Department of ECE,
Nandha College of Technology.

Abstract — Authentication protocol plays an important role in the short-range wireless communications for the Near Field Communication (NFC) technology. Due to the shared nature of wireless communication networks, there are several kinds of security vulnerabilities. However, this paper further analyzes PBNFCP and shows that it still fails to prevent the claimed security properties, such as impersonation attacks against an adversary, who is a malicious registered user having a valid pseudonym and corresponding private key. In order to overcome these security drawbacks, this paper proposes a secure and efficient authentication protocol (SEAP) for NFC applications using lifetime-based pseudonyms.The proposed SEAP is simulated for the formal security verification using the widely-accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool. The simulation results show that SEAP is secure. The rigorous security and performance analysis shows that the proposed SEAP is secure and efficient as compared to the related existing authentication protocols for NFC applications[1].In Addition to that biometric sensor is used to provide the security for our transmission..Encryption and Decryption method is used.

I. INTRODUCTION

The market size of NFC-based payment services is expected to be increased to $3.572 and $180 billion in the years 2015 and 2017 separately [1], [2]. Since the rapid development of short-range wireless communication technology, there is a growing demand to design secure and efficient mobile applications, such as service discovery, e-payment, ticketing, and mobile healthcare systems, etc., in the area of the consumer electronics for NFC. In the NFC environment, the Trusted Service Manager (TSM) is responsible to distribute user keys to the registered users based upon the requests from the users and it does not involve in the authentication process. The authentication protocol involves only two parties, namely, an initiator user and a target user. The initiator user generates a radio frequency field and starts the NFC interface. After receiving communication signals, the target user sends a response message to the initiator user through the radio frequency field. After mutual authentication, both the initiator user and target user establish and agree on a secure session key. Due to the shared nature of wireless communication networks, there are several kinds of security vulnerabilities in NFC environment including impersonation and man-in-the-middle attacks. Thus, the security is one of the prerequisite for NFC applications . Moreover, transmission capacity of NFC technology is limited as its operating frequency is 13.56 MHz with transmission speed ranging from 106 Kbps to 424 Kbps up to 10 cm Since the widely use of mobile devices, such as smart phones and personal laptops, in combination of NFC technology, authentication protocol must ensure high security along with low computation and communication costs

A.Relates work

With the rapid development in mobile applications, the NFC is expected to become a very trendy technology for mobile services, more specifically for mobile payments In recent years, many researchers presented the assessment of NFC for future mobile payment systems [5], [6], [9], [10]. A public key infrastructure is used for the as initiator and target users. In this scenario, an adversary could track the user's activities by tracing its public key, and as a result, the user's privacy may be broken [11]-[13]. In order to overcome these drawbacks, the pseudonym technique is used in many authentication protocols include NFC and vehicular ad hoc networks (VANETs) [14]-[16]. In 2013, Eun et al. [17] presented a new conditional privacy preserving security protocol (CPPNFC) to protect the user's privacy. Later, in 2015, Kannadhasan et al. [14] proposed the similar approach as presented in CPPNFC. However, He et al. [18] pointed out that CPPNFC fails to prevent the impersonation attacks, and they further proposed a pseudonym based NFC protocol (PBNFCP) to withstand the security drawbacks found in CPPNFC with a marginal computational cost increase. The proposed security attacks are also applicable in Kannadhasan et al.'s protocol [14] as their approach remains same as that in CPPNFC where the user cannot identify the real identity of another user. This paper further revisits He et al.'s PBNFCP and shows that it still fails to prevent the proposed impersonation attacks on CPPNFC against an attacker (being an insider registered user), and discusses the

[1] V. Odelu is with the Department of Mathematics, Indian Institute of Technology Kharagpur, 721302, India. (e-mail: odelu.vanga@gmail.com, odelu.phd@maths.iitkgp.ernet.in).

A. K. Das is with the Center for Security, Theory and Algorithmic Research of the International Institute of Information Technology, Hyderabad, 500032, India. (e-mail: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in).

A. Goswami is with the Department of Mathematics, Indian Institute of Technology Kharagpur, 721302, India.
(e-mail: goswami@maths.iitkgp.ernet.in).

drawbacks of designed pseudonym in PBNFCP. This paper proposes a new secure and efficient authentication protocol (SEAP) for NFC applications using the new defined lifetime-based pseudonyms to withstand the security drawbacks found in PBNFCP.

TABLE I
NOTATIONS USED IN THIS PAPER

SYMBOL	DESCRIPTION
TSM	Trusted Service Manager
$E_p(a,b)$	Non-singular elliptic curve $y^2 = x^3 + ax + b(\text{mod } p)$ over a prime field $GF(p)$
G	A base point on the elliptic curve $E_p(a,b)$
kG	$G+G+...+G(k-times)$, elliptic curve scalar multiplication over $E_p(a,b)$, where $k \in Z_p^*$
(Q_{TSM}, d_{TSM})	Public-key and private key pair of TSM, $Q_{TSM} = d_{TSM}G$
ID_X	Identity of an entity X
(P_X^i, d_X^i)	i th pseudonym and private key pair of X issued by TSM
Q_X^i	Elliptic curve public key corresponding to P_X^i
RX	Compressed elliptic curve point computed by a user X
h, f	Two secure cryptographic one-way hash functions
KDF	Key derivation function
$Enc(k,m)$	Symmetric-key encryption of data m using the key k
$Sig(k,m)$	Signature on data m using the key k
$\|\|$	Concatenation operation

B. Contributions

The contributions of the paper are listed below:

(i) This paper analyzes and shows that the recently proposed PBNFCP fails to provide the claimed security properties, such as impersonation attacks against a malicious registered user being an attacker.

(ii) In this paper, a new secure and efficient authentication protocol (SEAP) is presented for the NFC applications using the lifetime-based pseudonyms. The proposed pseudonym and private key pair in SEAP is valid within its lifetime only. Thus, even if a pseudonym and private key pair is unexpectedly revealed to an adversary, he/she can use it within its expiry time on behalf of the corresponding user only. As a result, the vulnerability in this case is limited to the corresponding user only, whereas in PBNFCP, CPPNFC, and Kannadhasan et al.'s protocol, it causes to the impersonation attacks to any legitimate user in the system when the identity of that user is known to the adversary. Moreover, the size of the proposed pseudonym in SEAP is significantly reduced.

(iii) The rigorous informal security analysis shows that SEAP is secure against possible well known attacks including the impersonation and man-in-the-middle attacks. In addition, the simulation results for the formal security verification using the widely accepted AVISPA tool shows that SEAP is secure against the passive and active attacks.

(iv) SEAP significantly reduces the computation and communication costs, and also provides more security functionalities as compared to the related existing protocols.

(v) Due to efficiency and more security functionalities, SEAP is very suitable for the short-range wireless communication applications, such as service discovery, e-payment, ticketing, and mobile healthcare systems, etc., in the area of the consumer electronic devices in the NFC environment.

C. Organization of the paper

The rest of the paper is sketched as follows. In Section II, a brief review of He et al.'s protocol is provided. In Section III, the security weaknesses of He et al.'s protocol are discussed. Section IV proposes a new authentication protocol (SEAP) for NFC applications. The rigorous security analysis of the proposed SEAP is presented in Section V. The simulation of SEAP for the formal security verification using the widely-accepted AVISPA tool is provided in Section VI. The performance of SEAP with related existing protocols is compared in Section VII. Finally, the paper is concluded in Section VIII.

II. REVIEW OF HE ET AL.'S PROTOCOL

This section briefly reviews He et al.'s proposed PBNFCP [18]. The notations used in this paper are listed in TABLE I. In PBNFCP, a user A requests the TSM for the pseudonyms to authenticate and establish a session with other users. Upon receiving the request, the TSM chooses n random secrets q_A^i, $i = 1, 2, ..., n$, and then computes n pseudonyms P_A^i as $P_A^i = \{Q_A^i \| Enc(d_{TSM}, \{ID_A, Q_A^i\}) \| ID_{TSM} \| S_{TSM}^i\}$, where $Q_A^i = d_A^i G$ and $d_A^i = q_A^i + h(ID_{TSM}, P_A^i)d_{TSM}$ are i-th public and private key pairs, respectively, and $S_{TSM}^i = Sig(d_{TSM}, Q_A^i \| Enc(d_{TSM}, Q_A^i) \| ID_{TSM})$ is the TSM's signature. Finally, the TSM sends the n pseudonym and private key pairs (P_A^i, d_A^i) to the user A via a secure channel, and then stores the identity ID_A and the corresponding pseudonyms P_A^i's of the user A in its database.

The initiator user A and target user B use the received pseudonyms in order to establish a session key $SK = SK_A = SK_B$ as follows:

1) A randomly selects a pseudonym and private key pair (P_A^j, d_A^j), and generates a nonce N_A and random number r_A. Then, A computes $Q_A' = r_A G$ and sends the request message $M_1 = \{Q_A', P_A^j, N_A\}$ to the user B via a public channel.

2) Upon receiving request M_1, B randomly selects a pseudonym and private key pair (P_B^j, d_B^j), and generates a nonce N_B and a random number r_B. Then, B computes $Q_B' = r_B G$, and sends the response message $M_2 = \{Q_B', P_B^j, N_B\}$ to the user A via a public channel.

3) Upon receiving M_2 from B, A computes $Z_A^1 = r_A Q_B'$, $Z_A^2 = d_A^i(Q_B^j + h(ID_{TSM}, P_B^j)Q_{TSM}$, $SK_A = KDF(N_A, N_B, ID_A, ID_B, Z_A^1, Z_A^2)$ and $f_A = f(SK_A, ID_A, ID_B, Q_A', Q_B')$. Finally, A

sends the authentication message $M_3 = \{f_A\}$ to B via a public channel.

4) Upon receiving M_3 from A, B computes $Z_B^1 = r_B Q_A^{'}$, $Z_B^2 = d_B^j (Q_A^i + h(ID_{TSM}, P_A^i) Q_{TSM})$ and $SK_B = KDF(N_A, N_B, ID_A, ID_B, Z_B^1, Z_B^2)$. B then checks whether the condition $f_A = f(SK_B, ID_A, ID_B, Q_A, Q_B)$ holds or not. If it does not hold, B terminates the session. Otherwise, B sets SK_B as the session key, and computes $f_B = f(SK_B, ID_B, ID_A, Q_B^{'}, Q_A^{'})$. Finally, B sends confirmation message $M_4 = \{f_B\}$ to A via a public channel.

5) Upon receiving M_4 from B, A checks whether the condition $f_B = f(SK_A, ID_B, ID_A, Q_B, Q_A^{'})$ holds or not. If it does not hold, A rejects the session. Otherwise, A confirms that the shared session key with the user B is $SK_A (= SK_B)$.

The summary of the session key agreement process of He et al.'s protocol is shown in Fig. 1.

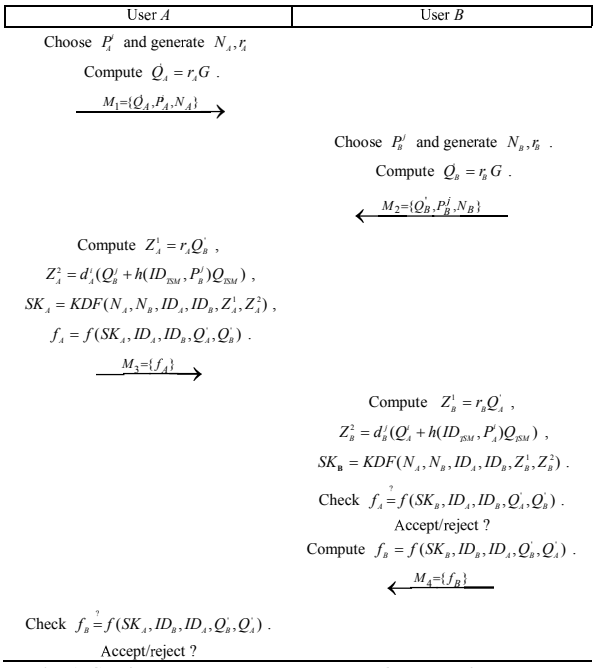

Fig. 1. Session key agreement process of He et al.'s protocol

III. SECURITY ANALYSIS OF HE ET AL.'S PROTOCOL

He et al. pointed out that Eun et al.'s protocol fails to prevent impersonation attack as the user cannot confirm the real identity of another party in Eun et al.'s protocol. In order to remedy these drawbacks, He et al. proposed a new protocol for NFC environment. However, He et al.'s protocol still fails to avoid such impersonation attacks against a malicious registered user. The attacks on He et al.'s protocol are discussed in the following subsections.

Let C be such an adversary, who have full control over the communication channel such that he/she can modify, replay, intercepts the messages transmitted between users A and B [17], [18]. Assume that C requests the TSM for pseudonyms as another legal user than A and B. C launches the following two kinds of impersonation attacks as defined in He et al.'s protocol using valid pseudonym and private key pair (P_C^k, d_C^k).

A. Impersonation attack against initiator object

In this attack, C can impersonate A to B as follows.

1) C randomly picks a pseudonym and private key pair (P_C^k, d_C^k), and generates a random number r_C and random nonce N_C. C computes $Q_C^{'} = r_C G$ and sends the message $M_1^{'} = \{Q_C^{'}, P_C^k, N_C\}$ to B.

2) Upon receiving request $M_1^{'}$, B randomly selects a pseudonym and private key pair (P_B^j, d_B^j), and generates a nonce N_B and a random number r_B. Then, B computes $Q_B^{'} = r_B G$ and sends the response $M_2 = \{Q_B^{'}, P_B^j, N_B\}$ to A.

3) C intercepts M_2, and computes $Z_{CB}^1 = r_C Q_B^{'}$, $Z_{CB}^2 = d_C^k (Q_B^j + h(ID_{TSM}, P_B^j) Q_{TSM})$, $SK_{CB} = KDF(N_C, N_B, ID_A, ID_B, Z_{CB}^1, Z_{CB}^2)$ and $f_{CB} = f(SK_{CB}, ID_A, ID_B, Q_C^{'}, Q_B^{'})$. Finally, C sends the message $M_3^{'} = \{f_{CB}\}$ to B.

4) Upon receiving the message $M_3^{'} = \{f_{CB}\}$ from C, user B computes $Z_{BC}^1 = r_B Q_C^{'}$, $Z_{BC}^2 = d_B^j (Q_C^k + h(ID_{TSM}, P_C^k) Q_{TSM})$ and $SK_{BC} = KDF(N_C, N_B, ID_A, ID_B, Z_{BC}^1, Z_{BC}^2)$. Then B checks whether the condition $f_{CB} = f(SK_{BC}, ID_A, ID_B, Q_C, Q_B)$ holds or not. If it does not hold, B terminates the session. Otherwise, B believes that the key SK_{BC} is the shared session key between A and B. Finally, B computes $f_{BC} = f(SK_{BC}, ID_B, ID_A, Q_B^{'}, Q_C^{'})$ and sends the message $M_4 = \{f_{BC}\}$ to A.

5) C intercepts the message M_4 and checks whether $f_{BC} = f(SK_{CB}, ID_B, ID_A, Q_B^{'}, Q_C^{'})$ holds. If it holds, C successfully shares session key $SK_{CB} = SK_{BC}$ with B.

B. Impersonation attack against target object

In this attack, C can also impersonate user B to A as follows.

1) A randomly selects a pseudonym and private key pair (P_A^i, d_A^i), and generates a nonce N_A and a random number r_A. Then, A computes $Q_A^{'} = r_A G$ and sends the request message $M_1 = \{Q_A^{'}, P_A^i, N_A\}$ to B.

2) C intercepts the message M_1. Then, C randomly picks a pseudonym and private key pair (P_C^k, d_C^k), and generates a random number r_C and random nonce N_C. C computes $Q_C^{'} = r_C G$ and sends the message $M_2^{'} = \{Q_C^{'}, P_C^k, N_B\}$ to A.

3) Upon receiving $M_2^{'}$ from C, A computes $Z_{AC}^1 = r_A Q_C^{'}$, $Z_{AC}^2 = d_A^i (Q_C^k + h(ID_{TSM}, P_C^k) Q_{TSM})$, $SK_{AC} = KDF(N_A, N_B, ID_A,$

$ID_B, Z^1_{AC}, Z^2_{AC})$ and $f_{AC} = f(SK_{AC}, ID_A, ID_B, Q^i_A, Q^i_C)$. Finally, *A* sends the authentication message $M_3 = \{f_{AC}\}$ to *B*.

4) *C* intercepts M_3 and computes $Z^1_{CA} = r_C Q_A$, $Z^2_{CA} = d^k_C(Q_A + h(ID_{TSM}, P^i_A)Q_{TSM})$, $SK_{CA} = KDF(N_A, N_B, ID_A, ID_B, Z^1_{CA}, Z^2_{CA})$. *C* checks whether the condition $f_{AC} = f(SK_{CA}, ID_A, ID_B, Q^i_A, Q^i_C)$ holds or not. If it holds, *C* further computes $f_{CA} = f(SK_{CA}, ID_B, ID_A, Q^i_C, Q^i_A)$ and sends the message $M^{'}_3 = \{f_{CA}\}$ to the user *A*.

5) Upon receiving $M^{'}_3$ from *C*, *A* checks whether the condition $f_{CA} = f(SK_{AC}, ID_B, ID_A, Q^i_C, Q^i_A)$ holds or not. If it does not hold, *A* terminates the session. Otherwise, *A* believes that the key $SK_{AC} = SK_{CA}$ is the shared session key between *A* and *B*.

C. Correctness of the proposed attacks

Since the pair (P^k_C, d^k_C) is valid, the following statements are true:

$d^k_C G = (q^k_C + h(ID_{TSM}, P^k_C)d_{TSM})G = Q^k_C + h(ID_{TSM}, P^k_C)Q_{TSM}$,

$Z^2_{AC} = d^i_A(Q^k_C + h(ID_{TSM}, P^k_C)Q_{TSM}) = d^i_A d^k_C G = d^k_C d^i_A G$

$= d^k_C(q^i_A + h(ID_{TSM}, P^i_A)d_{TSM})G = d^k_C(Q^i_A + h(ID_{TSM}, P^i_A)Q_{TSM}) = Z^2_{CA}$.

Similarly, the equality $Z^2_{CB} = Z^2_{BC}$ is also true. Thus, it is clear that the presented attacks on He et al.'s protocol are valid.

D. Other drawbacks

The designed pseudonym in PBNFCP, CPPNFC, and Kannadhasan et al.'s protocol has no lifetime, that is, it never expires. In addition, the TSM does not involve in the authentication process. Once a valid pseudonym and private key pair of a user is unexpectedly revealed to an adversary, that adversary can use it in his/her entire lifetime to launch the impersonation attacks as discussed above. It is a serious issue in NFC-based authentication protocols for e-payments.

IV. THE PROPOSED SEAP PROTOCOL

In this section, a new secure and efficient pseudonym-based security protocol (SEAP) is proposed to withstand the security pitfalls found in He et al. and other protocols. The proposed SEAP consists of two phases, namely, pseudonym request phase and session key establishment phase.

A. Pseudonym request phase

A user *A* requests the TSM for the pseudonyms to authenticate and establish a session with other users. In order to overcome the security drawbacks found in He et al.'s protocol, the TSM generates *n* pseudonyms and private key pairs, say (P^i_A, d^i_A) using the elliptic curve cryptography (ECC) based El-Gammal type signature [19] as follows.

The TSM first chooses *n* random numbers $q^i_A, i = 1, ..., n$, and computes $P^i_A = \{Q_A \| Enc(d_{TSM}, \{ID_A, q^i_A\}) \| ID_{TSM} \| LT^i_A\}$, $d^i_A = q^i_A + h(ID_A, ID_{TSM}, P^i_A)d_{TSM}$, where $Q^i_A = q^i_A G$ is *i*th public key and LT^i_A the lifetime of P^i_A defined by the TSM according to the security requirement. Finally, the TSM sends the *n* pseudonym and private key pairs (P^i_A, d^i_A) to the user *A* via a secure channel, and stores the identity ID_A and corresponding pseudonyms P^i_A's of *A* in its database until expiration of the pairs. It is observed that even if a pseudonym and private key pair is unexpectedly revealed to an adversary, he/she can only use it within its expiry time on behalf of corresponding user. This implies that possibility of vulnerability is limited to the corresponding user only, whereas in PBNFCP and other protocols, it causes impersonation attacks to any legitimate registered user.

B. Session key establishment phase

In this phase, the process of authentication and key agreement between an initiator user *A* and a target user *B* of SEAP is discussed. In order to establish a session key $SK = SK_A = SK_B$, *A* and *B* need to execute the following steps. The summary of this phase is shown in Fig. 2.

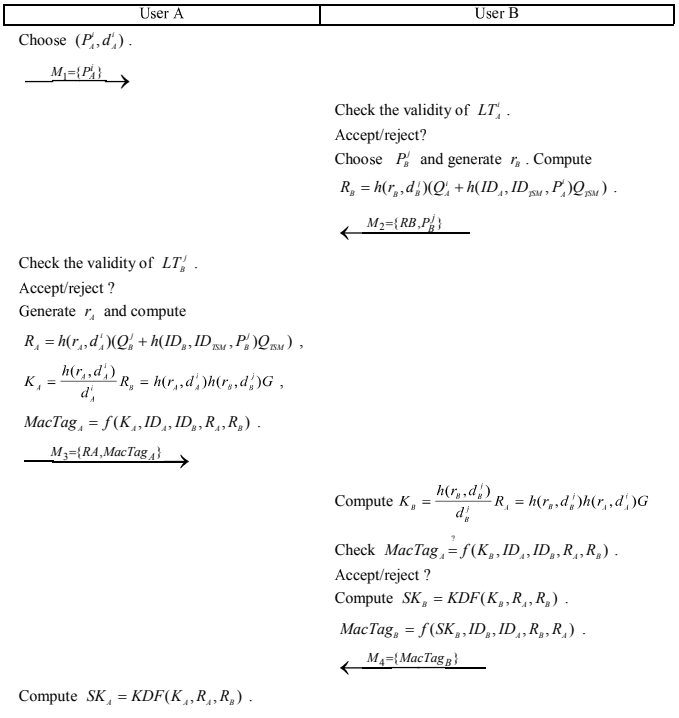

Fig. 2. Summary of the session key establishment of proposed SEAP

1) *A* randomly picks a pseudonym and private key pair (P^i_A, d^i_A), and sends the request $M_1 = \{P^i_A\}$ to *B* via a public channel.

2) Upon receiving M_1, *B* checks validity of the lifetime LT^i_A containing in P^i_A. If it is valid, *B* randomly picks a pseudonym and private key pair (P^j_B, d^j_B), and generates a

random number r_B and computes $R_B = h(r_B, d_B^j)(Q_A^i + h(ID_A, ID_{TSM}, P_A^i)Q_{TSM}) = h(r_B, d_B^j)d_A^i G$. Finally, B sends the response $M_2 = \{RB, P_B^j\}$ to A via a public channel.

3) Upon receiving M_2 from B, A checks validity of the lifetime LT_B^j containing in P_B^j. If it is valid, A generates a random number r_A. Then, A computes $R_A = h(r_A, d_A^i)(Q_B^j + h(ID_B, ID_{TSM}, P_B^j)Q_{TSM}) = h(r_A, d_A^i)d_B^j G$, $K_A = \frac{h(r_A, d_A^i)}{d_A^i} R_B = h(r_A, d_A^i)h(r_B, d_B^j)G$, $MacTag_A = f(K_A, ID_A, ID_B, R_A, R_B)$. Finally, A sends the authentication message $M_3 = \{RA, MacTag_A\}$ to B via a public channel.

4) Upon receiving M_3 from A, B computes $K_B = \frac{h(r_B, d_B^j)}{d_B^j} R_A = h(r_B, d_B^j)h(r_A, d_A^i)G$ and checks whether the condition $MacTag_A = f(K_B, ID_A, ID_B, R_A, R_B)$ holds. If it holds, B authenticates A, and then computes the session key $SK_B = KDF(K_B, R_A, R_B)$ and $MacTag_B = f(SK_B, ID_B, ID_A, R_B, R_A)$. Finally, B sends the confirmation message $M_4 = \{MacTag_B\}$ to A via a public channel.

5) Upon receiving M_4, A computes $SK_A = KDF(K_A, R_A, R_B)$ and checks whether the condition $MacTag_B = f(SK_A, ID_B, ID_A, R_B, R_A)$ holds. If it does not hold, A rejects the session. Otherwise, A authenticates B and confirms that SK_A is the shared session key with B.

V. SECURITY ANALYSIS OF SEAP PROTOCOL

In this section, SEAP is thoroughly analyzed and shown that it is secure against the well known attacks including the man-in-the-middle attack.

A. Impersonation attack

During computation of private key, unlike He et al.'s protocol, SEAP computes it using three fields in hash function, that is, d_A^i as $q_A^i + h(ID_A, ID_{TSM}, P_A^i)d_{TSM}$. Thus, the pseudonym and private key pair (P_A^i, d_A^i) becomes an El-Gamal type ECC-based signature on the identity ID_A of user A generated by the TSM's private key d_{TSM} [19]. Assume that an attacker C is a registered user with a valid pseudonym and private key pair (P_C^k, d_C^k), and users A and B are two communicating parties. C fails to authenticate at both A and B by launching the impersonation attack. The argument is given below:

a) Suppose C intercepts the message $M_1 = \{P_A^i\}$ which is sent to B by A, and computes the response $R_C = h(r_C, d_C^k)(Q_A^i + h(ID_A, ID_{TSM}, P_A^i)Q_{TSM}) = h(r_C, d_C^k)d_A^i G$. C sends $M_2' = \{RC, P_C^k\}$ to A.

b) Upon receiving M_2', A computes $R_A = h(r_A, d_A^i)(Q_C^k + h(ID_B, ID_{TSM}, P_C^k)Q_{TSM}) \neq h(r_A, d_A^i)d_C^k G$, where $ID_C \neq ID_B$, $K_A = \frac{h(r_A, d_A^i)}{d_A^i} R_C = h(r_A, d_A^i)h(r_C, d_C^k)G$, and $MacTag_A = f(K_A, ID_A, ID_B, R_A, R_B)$. A sends the message $M_3 = \{RA, MacTag_A\}$ to B.

c) After intercepting M_3, and C computes $K_C = \frac{h(r_C, d_C^k)}{d_C^k} R_A \neq h(r_C, d_C^k)h(r_A, d_A^i)G = K_A$. Since $K_C \neq K_A$, C is never authenticated by A on behalf of B without valid pseudonym and private key pair (P_B^j, d_B^j) of B. Similarly, C is also never authenticated by B on behalf of A without valid pseudonym and private key pair (P_A^i, d_A^i) of A. In addition, (P_A^i, d_A^i) of A is the El-Gamal type ECC-based signature on ID_A, and thus, generating a new valid such a pair without the private key d_{TSM} of TSM is computational hard problem for C [19]. As a result, SEAP successfully prevents such attacks.

B. Secure mutual authentication

Since (P_A^i, d_A^i) is the El-Gamal type ECC-based signature on ID_A, it is computationally hard for an adversary C to generate such a valid pair due to the difficulty of solving elliptic curve discrete logarithm problem (ECDLP) [19]. Thus, C does not have any ability to compute the valid $MacTag_A$ to be authenticated by B and $MacTag_B$ to be authenticated by A. This implies that SEAP prevents unauthorized modifications, and thus, the users A and B mutually authenticate each other by validating $MacTag_B$ and $MacTag_A$, respectively. Hence, SEAP provides secure mutual authentication.

C. User anonymity

It ensures that an adversary C cannot trace the user activities by intercepting the transmitted messages. C has full control over the communication due to wireless network used in NFC applications. Assume that C intercepts all the messages $M_1 = \{P_A^i\}$, $M_2 = \{RB, P_B^j\}$, $M_3 = \{RA, MacTag_A\}$ and $M_4 = \{MacTag_B\}$ transmitted between the users A and B. The user identity is involved in the corresponding pseudonyms P_A^i and P_B^j, which are then encrypted by the TSM's private key. Thus, except the TSM, no adversary can compute the real identity of a user from given pseudonym. From the above discussion (Section V-A), no adversary can verify whether the pseudonym corresponds to the given user identity due to the difficulty of solving ECDLP. On the other hand, no adversary can retrieve the real identity from $MacTag_A$ and $MacTag_B$ due to the one-way collision-resistance hash function property [18]. Thus, the adversary cannot trace the original user identity from the intercepted communications. As a result, SEAP provides the user anonymity property.

D. Session key security

An authentication protocol should ensure the security of the session key in the following two cases [3], [18], [19]: (i) when the session-specific temporary information is unexpectedly revealed to an adversary by session exposure attack and (ii) when valid pseudonym and private key pair corresponding to session key is unexpectedly revealed to an adversary. Assume that any one of these cases may arise, but not both. In SEAP, using $K_A = K_B = h(r_A, d_A^i)h(r_B, d_B^j)G$ the session key is computed, which involves both the random numbers pair (r_A, r_B) and private keys pair (d_A^i, d_B^j). It is clear that computing session key using only one of the pairs (r_A, r_B) and (d_A^i, d_B^j) is computationally infeasible for an adversary due to the difficulty of solving ECDLP. Hence, the adversary has no ability to derive any session key in the two cases (i) and (ii). This implies that SEAP provides perfect forward security and is secure against session-specific temporary information leakage attack. Hence, SEAP provides session key security.

E. Replay attack

From the above arguments, no adversary can compute valid authentication and confirmation messages to be authenticated by users A and B using intercepted messages as SEAP prevents unauthorized modifications. No adversary can then successfully establish the session by replaying intercepted messages without corresponding valid pseudonym and private key pair. As generating valid pseudonym and private key pair is computationally hard problem due to solving ECDLP, the adversary cannot launch the replay attack. Hence, SEAP is secure against the replay attack.

F. Man-in-the-middle attack

In this attack, an adversary tries to impersonate the legal users by intercepting the messages between communicating users using available public information. However, from above discussion, SEAP prevents impersonation attacks and provides secure mutual authentication between two communicating parties. As a result, SEAP is secure against this attack.

G. Modification attack

An adversary does not have any ability to compute valid $MacTag_A = f(K_A, ID_A, ID_B, R_A, R_B)$ to be authenticated by B and $MacTag_B = f(SK_B, ID_B, ID_A, R_B, R_A)$ to be authenticated by A due to the difficulty of generating El-Gamal type ECC-based signature on given identity. Thus, SEAP successfully prevents the unauthorized modifications.

VI. SIMULATION FOR FORMAL SECURITY VERIFICATION USING AVISPA TOOL

In this section, SEAP is simulated using the widely-accepted AVISPA tool to show that SEAP is secure.

A. Overview of AVISPA

AVISPA is a push-button tool for automated validation of Internet security-sensitive protocols and applications, which formally verifies whether a security protocol is safe or unsafe [3], [19]-[24].

```
role userA (A, B, TSM : agent,   H, F : hash_func, SEND, RECV: channel(dy)
                % H, F are hash function )
played_by A
def=
 local State : nat, lDa, lDtsm, Ra, Rb, G : text,  KDF, W : hash_func,  PAi, PBj, RAi, RBj, DAi,
    DBj, QqAi, QqBj, QBj, SK, Qtsm, LTAi, LTBj,  Dtsm, lDb, MacTagA, KA: 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' := 1  /\ QqAi' := new()   /\ PAi' := W(QqAi'.G).{lDa.QqAi'}_(Dtsm). lDtsm.LTAi
          /\ secret({Dtsm,QqAi'}, s1, TSM)   /\ secret({DAi,lDa}, s2, A)
          /\ secret({DBj,lDb}, s3, B)   /\ SEND(PAi')
% Receive < M2 > from user B
2. State = 1 /\ RECV(W(H(Rb'.DBj).W(W(QqAi'.G).W(H(lDa.lDtsm.W(QqAi'.G).
                        {lDa.QqAi'}_(Dtsm).lDtsm.LTAi).W(Dtsm.G))).W(QqBj'.G).
                        {lDb.QqBj'}_(Dtsm). lDtsm.LTBj)) =|>
% Send < M3 > to user B
State' := 2 /\ Ra' := new()  /\ RAi' := W(H(Ra'.DAi).W(F(QqBj'.G). H(lDb.lDtsm.W(QqBj'.G).
          {lDb.QqBj'}_(Dtsm).lDtsm.LTBj).W(Dtsm.G)))
 /\ KA' := W(H(Ra'.DAi).H(Rb'.DBj).G)  /\ MacTagA' := F(KA'.lDa.lDb.RAi'. W(H(Rb'.DBj).
W(W(QqAi'.G).W(H(lDa.lDtsm. W(QqAi'.G).{lDa.QqAi'}_(Dtsm).lDtsm.LTAi). W(Dtsm.G)))))
          /\ SEND(RAi'.MacTagA')    /\ witness(A, B, a_b_ra, Ra')
% A has freshly generated the value Ra for B
% Receive < M4 > from user B
3. State = 2 /\ RECV(F(SK'.lDb.lDa.W(H(Rb'.DBj). W(W(QqAi'.G). W(H(lDa.lDtsm.
W(QqAi'.G).  {lDa.QqAi'}_(Dtsm).lDtsm.LTAi). W(Dtsm.G))).W(H(Ra'.DAi).W(F(QqBj'.G).
 H(lDb.lDtsm.W(QqBj'.G). {lDb.QqBj'}_(Dtsm).lDtsm.LTBj).W(Dtsm.G))))) =|>
% A's acceptance of the value Rb generated for A by B
State' := 3 /\ request(B, A, b_a_rb, Rb')
end role
```

Fig. 3. Role for the user *A*

In AVIPSA, the protocols need to be specified in HLPSL (High Level Protocols Specification Language) [20], [21]. HLPSL is a role-oriented language in which each specified role is independent from other roles. The role system defines the number of sessions, number of principals and roles. In addition, in HLPSL an intruder (which is always denoted by *i*) is modeled using the Dolev-Yao model [25] with the possibility for the intruder to assume a legitimate role in a protocol run. The HLPSL code is converted to the intermediate format (IF) using the HLPSL2F translator. The IF is then fed into one of the following four backends: (i) OFMC (On-the-fly Model-Checker); (ii) CL-AtSe (Constraint-Logic-based Attack Searcher); (iii) SATMC (SAT-based Model-Checker); and (iv) TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols). For more details on these backends, one can refer to [20], [21]. Finally, these backends produce the output format (OF), which has following sections [21]: SUMMARY tells whether the tested protocol is safe, unsafe, or if the analysis is inconclusive. DETAILS either explains under what condition the tested protocol is declared safe, or what conditions have been used for finding an attack, or finally why the analysis was inconclusive. PROTOCOL, GOAL and BACKEND are the name of the protocol, the goal of the analysis and the name of the backend used, respectively. After some comments and statistics, the trace of an attack (if any) is also displayed in the standard Alice-Bob format.

Various basic types supported by HLPSL are as follows [21]: *agent, symmetric_key, public_key, hash_func, nat,* and *text* represent the principal names, secret keys in a symmetric-key cryptosystem, public keys in a public-key cryptosystem, cryptographic hash function, natural numbers in non-message contexts, and a nonce. Note that if a given public (respectively private) key *ku*, its inverse private (respectively public) key is denoted by *inv_ku*, respectively. In addition, if *N* is a type text (fresh), *N*' is a fresh value which an intruder cannot guess it.

B. Specifying the protocol

In the implementation of SEAP in HLPSL for the session key agreement phase: *userA* and *userB* represent for the basic roles for user *A* and user *B*, respectively, and role for the session, and role for the goal and environment are defined. In Fig. 3, role for *A* is shown. *A* first receives the start signal and then changes its initial state (denoted by State) from 0 to 1. *A* sends message $M_1 = \{P_A^i\}$ to *B* via a public channel using the SEND( ) operation. After receiving message $M_2 = \{RB, P_B^j\}$ from *B* via a public channel by the RECV( ) operation, *A* changes its state from 1 to 2. *A* then sends the message $M_3 = \{RA, MacTag_A\}$ to *B* via a public channel. Finally, *A* waits for acknowledgment $M_4 = \{MacTag_B\}$ from *B*. The *played_by A* declaration indicates that the agent named in variable *A* will play in a specific role. If a variable *V* needs to be permanently kept secret, it is expressed by goal *secrecy_of V*. Therefore, if *V* is ever obtained or derived by an intruder, a security violation will result immediately. The declaration witness (A, B, a_b_ra, Ra') tells that *A* has freshly generated random number r_A for *B* characterized by protocol id a_b_ra. By the declaration request (B, A, b_a_rb, Rb'), *A* authenticates *B* based on r_B. In a similar way, the role for *B* is implemented in HLPSL and is shown in Fig. 4.

Finally, implementations of roles for session, and goal and environment for SEAP are provided in Fig. 5. In session segment, all basic roles including roles for *A* and *B* are instanced with concrete arguments. The top-level role, called the environment, contains global constants and a composition of one or more sessions. As shown in Fig. 5, an intruder (*i*) can play some roles as legitimate users. In the current version of HLPSL, the standard authentication and secrecy goals are supported. In the implementation, three secrecy goals and two authentications are verified.

C. Analysis of results

The widely-accepted OFMC and CL-AtSe backends are chosen for the execution tests and a bounded number of sessions model checking. For replay attack protection, these backends verify whether the legitimate agents (users) can execute the specified protocol by means of performing a search of a passive intruder. For the Dolev-Yao check, the backends check if there is any man-in-the-middle attack possible by the intruder. The proposed SEAP is simulated using SPAN (Security Protocol ANimator for AVISPA) [21] for OFMC and CL-AtSe. The simulation results for the formal security verification of SEAP shown in Fig. 6 ensure that SEAP is secure against the replay and man-in-the-middle attacks. The summary of the results reported under OFMC and CL-AtSe backends reports that SEAP is safe.

```
role userB (A, B, TSM : agent, H, F : hash_func, SEND, RECV: channel(dy)
            % H, F are hash function )
played_by B
def=
 local State : nat, IDa, IDtsm, Ra, Rb, G : text, KDF, W : hash_func, PAi, PBj, RAi, RBj, DAi,
    DBj, QqAi, QqBj, QBj, SK, Qtsm, LTAi, LTBj, Dtsm, KB, IDb, MacTagB: text
 const a_b_ra, b_a_rb, s1, s2, s3: protocol_id
init State := 0
 transition
% Session key agreement phase
 % Receive < M1 > from user A
1. State = 0 /\ RECV(W(QqAi'.G).{IDa.QqAi'}_(Dtsm). IDtsm.LTAi) =|>
State' := 1  /\ secret({Dtsm,QqAi'}, s1, TSM)  /\ secret({DAi,IDa}, s2, A) /\ secret({DBj,IDb}, s3, B)
% Send < M2 > to user A
/\ Rb' := new()   /\ QqBj' := new()
/\ PBj' := W(QqBj'.G).{IDb.QqBj'}_(Dtsm).IDtsm.LTBj  /\ RBj' := W(H(Rb'.DBj).W(W(QqAi'.G).
W(H(IDa.IDtsm. W(QqAi'.G).{IDa.QqAi'}_(Dtsm).  IDtsm.LTAi).W(Dtsm.G))))
 /\ SEND(RBj'.PBj')
% B has freshly generated the value Rb for A
        /\ witness(B, A, b_a_rb, Rb')
% Receive < M3 > from user A
2. State = 1 /\ RECV(W(H(Ra'.DAi).W(F(QqBj'.G). H(IDb.IDtsm.W(QqBj'.G).{IDb.QqBj'}_(Dtsm).
IDtsm.LTBj).W(Dtsm.G))).F(W(H(Ra'.DAi).H(Rb'.DBj).G).IDa.IDb. W(H(Ra'.DAi).W(F(QqBj'.G).
H(IDb.IDtsm.W(QqBj'.G).{IDb.QqBj'}_(Dtsm).IDtsm.LTBj).W(Dtsm.G))). W(H(Rb'.DBj).
W(W(QqAi'.G).W(H(IDa.IDtsm.W(QqAi'.G).{IDa.QqAi'}_(Dtsm).IDtsm.LTAi).W(Dtsm.G)))))) =|>
% Send < M4 > to user B
State' := 2 /\ KB' := W(H(Rb'.DBj).H(Ra'.DAi).G)  /\ SK' := KDF(KB'.Ra'.Rb')
/\ MacTagB' := F(SK'.IDb.IDa.W(H(Rb'.DBj).W(W(QqAi'.G).W(H(IDa.IDtsm.W(QqAi'.G).
  {IDa.QqAi'}_(Dtsm).IDtsm.LTAi).W(Dtsm.G))).W(H(Ra'.DAi).W(F(QqBj'.G).
  H(IDb.IDtsm.W(QqBj'.G).{IDb.QqBj'}_(Dtsm) .IDtsm.LTBj).W(Dtsm.G)))))
        /\ SEND(MacTagB')
% B's acceptance of the value Ra generated for B by A
        /\ request(A, B, a_b_ra, Ra')
end role
```

Fig. 4. Role for the user *B*

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

```
role environment()
def=
 const a, b, tsm : agent,   h, f,  kdf, w :
hash_func,   idtsm: text,   a_b_ra, b_a_rb,
 s1, s2, s3: protocol_id
 intruder_knowledge =
       {a, b, tsm, idtsm, h, f, w, kdf}
composition
 session(a, b, tsm, h, f) /\ session(i, b, tsm, h,
f)
/\ session(a, i, tsm, h, f)
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
  C:\progra~1\SPAN\testsuite\results\SEAP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.06s
  visitedNodes: 24 nodes
  Depth: 4 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\SEAP.if
GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS
  Analysed   : 8 states
  Reachable  : 6 states
  Translation: 0.02 seconds
  Computation: 0.01 seconds
```

Fig. 6. Analysis of results using OFMC and CL-AtSe backends

VII. PERFORMANCE COMPARISON WITH RELATED SCHEMES

This section analyzes the performance of the proposed SEAP, and compares it with the existing related Eun et al.'s

protocol [17], Kannadhasan et al.'s protocol [14] and He et al.'s protocol [18]. The notations are defined as follows [18], [19]: T_m, T_{em}, T_{ea}, T_h, T_{kdf} and T_{inv} denote the time to execute a modular multiplication, an elliptic curve point multiplication, an elliptic curve point addition, a hash function, a key derivation function, and a modular inverse operations, respectively; and approximate costs for each operation in terms of modular multiplication are as follows: $1T_{em} \approx 1200T_m$, $1T_{ea} \approx 5T_m$, $1T_h \approx 0.36T_m$, $1T_{kdf} \approx 1T_h \approx 0.36T_m$, and $1T_{inv} \approx 3T_m$. It is clear that the elliptic curve point multiplication operation is significantly costly which is approximately $1200T_m$, and the elliptic curve point addition, hash function and modular inverse operations require approximately $5T_m$, $0.36T_m$ and $3T_m$, respectively. Since the key derivation function is generally constructed through a hash function, it is assumed that the execution time T_{kdf} of KDF is same as the time to execute one hash operation T_h, that is, $1T_{kdf} \approx 1T_h \approx 0.36T_m$ [18].

TABLE II
COMPARISON OF COMPUTATIONAL COST

PROTOCOL	COMPUTATIONAL COST
Eun et al.	
Initiator user	$3T_{em}+1T_{ea}+2T_h+2T_m+1T_{kdf} \approx 3608T_m$
Target user	$3T_{em}+1T_{ea}+2T_h+2T_m+1T_{kdf} \approx 3608T_m$
Total cost	$6T_{em}+2T_{ea}+4T_h+4T_m+2T_{kdf} \approx 7216T_m$
Kannadhasan et al.	
Initiator user	$3T_{em}+1T_{ea}+2T_h+2T_m+1T_{kdf} \approx 3608T_m$
Target user	$3T_{em}+1T_{ea}+2T_h+2T_m+1T_{kdf} \approx 3608T_m$
Total cost	$6T_{em}+2T_{ea}+4T_h+4T_m+2T_{kdf} \approx 7216T_m$
He et al.	
Initiator user	$4T_{em}+1T_{ea}+3T_h+1T_{kdf} \approx 4806T_m$
Target user	$4T_{em}+1T_{ea}+3T_h+1T_{kdf} \approx 4806T_m$
Total cost	$8T_{em}+2T_{ea}+6T_h+2T_{kdf} \approx 9612T_m$
Proposed SEAP	
Initiator user	$3T_m+1T_{ea}+4T_h+1T_{kdf}+1T_{inv} \approx 3609T_m$
Target user	$3T_m+1T_{ea}+4T_h+1T_{kdf}+1T_{inv} \approx 3609T_m$
Total cost	$6T_m+2T_{ea}+8T_h+2T_{kdf}+2T_{inv} \approx 7218T_m$

In the proposed SEAP, both the initiator and target users require $3T_{em}+1T_{ea}+4T_h+1T_{kdf}+1T_{inv}$ operations, which is approximately $3609T_m$. Thus, the total computational cost required in SEAP is $6T_{em}+2T_{ea}+8T_h+2T_{kdf}+2T_{inv} \approx 7218T_m$. The computational costs of SEAP, Eun et al.'s protocol, Kannadhasan et al.'s protocol and He et al.'s protocol are compared in TABLE II. It is clear that the cost required in SEAP remains approximately equal to the cost required in Eun et al.'s protocol and Kannadhasan et al.'s protocol. However, this cost is significantly less as compared to He et al.'s protocol. Moreover, SEAP provides a secure mutual authentication and is secure against possible well known attacks including the impersonation and man-in-the-middle attack, whereas Eun et al.'s protocol, Kannadhasan et al.'s protocol and He et al.'s protocol fail to provide the secure mutual authentication as they are vulnerable to the impersonation attacks.

In order to compare the communication efficiency, the bit-length sizes of the parameters are given as follows [16]: ID_X is 16 bits, random number r_X is 96 bits, $MacTag_X$ is 96 bits, Q_X is 384 bits, RX is 200 bits, d_X is 192 bits, and session key SK is 128 bits. In addition, it is assumed that if q^i_X is 128 bits and LT^i_X is 32 bits, the symmetric ciphertext $Enc(d_{TSM}, \{ID_X, q^i_X\})$ becomes 192 bits.

TABLE III
COMPARISON OF COMMUNICATION COST

PROTOCOL	PSEUDONYM SIZE (BITS)	COMM. COST (BITS) /NO. OF MESSAGES
Eun et al.	1200	1184 (4 messages)
Kannadhasan et al.	1200	1184 (4 messages)
He et al.	1200	3184 (4 messages)
SEAP	624	1840 (4 messages)

The size of pseudonym in SEAP is (384+ 192 + 16 + 32) = 624 bits as the pseudonym in the proposed SEAP is $P^i_A = \{Q^i_A \,||\, Enc(d_{TSM}, \{ID_A, q^i_A\}) \,||\, ID_{TSM} \,||\, LT^i_A\}$. The size of the pseudonym is 1200 bits in Eun et al.'s protocol, Kannadhasan et al.'s protocol and He et al.'s protocol. Moreover, SEAP significantly reduces the communication cost as compared to He et al.'s protocol. The communication cost required for Kannadhasan et al.'s protocol and Eun et al.'s protocol is little less. However, Kannadhasan et al.'s protocol, Eun et al.'s protocol, and He et al.'s protocol are insecure as they do not prevent the impersonation attacks. Moreover, the pseudonym defined in Kannadhasan et al.'s protocol, Eun et al.'s protocol, and He et al.'s protocol never expires. Thus, if it leaks once to an adversary, he/she can launch the impersonation attack against any legal user during the pseudonym's entire lifetime, whereas in SEAP, it is limited to the corresponding user and valid within its lifetime only. The comparison of communication cost of SEAP with Eun et al.'s protocol, Kannadhasan et al.'s protocol and He et al.'s protocol is shown in TABLE III. The proposed SEAP needs significantly less communication cost as compared to He et al.'s protocol, where it is compared with other protocols. It is then clear that SEAP provides more security functionalities along with low computation and communication costs as compared to those for existing Eun et al.'s protocol, Kannadhasan et al.'s protocol and He et al.'s protocol.

VIII. CONCLUSION

The recently proposed He et al.'s protocol is first analyzed and then shown that it is vulnerable to two kinds of

impersonation attacks. A novel secure and efficient authentication protocol (SEAP) for NFC applications is proposed using the lifetime-based pseudonyms with significantly low computation and communication costs as compared to existing related authentication protocols. Through the rigorous security analysis, it is shown that SEAP is secure against possible known attacks including the impersonation attacks found in He et al.'s protocol. In addition, the simulation results for the formal security verification using the widely-accepted AVISPA tool clearly show that the proposed SEAP is secure. Thus, SEAP provides high security along with low computation and communication costs as compared to the related existing protocols.

REFERENCES

[1] Gartner, "Market Insight: The Outlook on Mobile Payment," *Market Analysis and Statistics*, May 2010.

[2] Juniper Research, "NFC Mobile Payments & Retail Marketing-Business Models & Forecasts 2012-2017," May 2012.

[3] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953-1966, Jun. 2015.

[4] R. Want, "Near field communication," *IEEE Pervasive Comput.,* vol.10, no.3, pp. 4 - 7, July. 2011.

[5] V. Patil, N. Varma, S. Vinchurkar, and B. Patil, "NFC based health monitoring and controlling system," *in Proc. IEEE Global Conference on Wireless Computing and Networking*, Lonavala, India, pp. 133-137, Dec. 2014.

[