Research Article

ACCENTS Transactions on Information Security, Vol 2(7)
ISSN (Online): 2455-7196
http://dx.doi.org/10.19101/TIS.2017.27002

An ID-based authenticated three-party key exchange protocol

Susmita Mandal*, Sujata Mohanty and Banshidhar Majhi
Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Odisha

Abstract
For secure communication in an open and distributed network, three-party authenticated key exchange (3PAKE) protocol establishes a secure session key between two users with the help of a trusted server to ensure transaction confidentiality and efficiency. Existing schemes fail to achieve privacy to user's identity also unable to ensure undeniability of a service request. Therefore, we propose an authenticated three-party key exchange scheme based on the elliptic curve computational diffie-hellman assumption (ECDH). The proposed scheme not only achieves anonymity, non-repudiation but also reduces the overall computational cost. The scheme is validated in automated validation of internet security protocols and applications (AVISPA) tool and is proved secure in the random oracle model. The scheme has huge applications in real life scenarios, namely, mobile-commerce; secure message transmission, and e-voting.

Keywords
Anonymity, Non-repudiation, ID-based, Elliptic curve cryptography, AVISPA tool.

1.Introduction
Key exchange protocols enable two parties to communicate securely over an untrusted network by exchanging a shared secret among them. Authentication and privacy are the two primary objectives of network security where privacy ensures that transmitted messages cannot eavesdrop. On the other hand, authentication assures that no unauthorized user can gain access maliciously. These two goals can be achieved simultaneously, using authenticated key exchange scheme where two or more parties can share a common secret to transmit a message securely in an open network.

Bellovin and Merritt [1] proposed a primitive two-party password-based authentication key exchange (2PAKE) protocol where each entity can authenticate one another via a public network for sharing a session key. On the basis of their protocol, several 2PAKE protocols [2-4] are proposed in the literature. However, the 2PAKE protocols are mostly suitable for client-server architectures, as they need to pre-share a common secret for mutual authentication and session key agreement. This restriction results in storage of huge amount of secret for communicating with a group of participants.

*Author for correspondence

To overcome this problem, *3PAKE* schemes are proposed where every user shares a single secret with a trusted server by eliminating the necessity of holding a huge amount of secrets to communicating with different group members. The first efficient three-party authenticated key exchange protocol based on *PKC* was proposed by Chen et al. [5]. The scheme proposes low round complexity to achieve mutual authentication. Yang and Chang [6] found, Chen's protocol suffer from the stolen-verifier attack and require more computation cost as it generates and verifies Schnorr's [7] digital signature based on modular exponentiation. Then, Yang et al. proposed an improved *3PAKE* protocol based on elliptic curve cryptography *(ECC)* without any pre-shared secrets. Later, Yang et al. proposed an improved *3PAKE* protocol based on elliptic curve cryptography *(ECC)*without any pre-shared secrets between client and server resulting in lower computation costs and low communication loads. In 2009,2010, Pu et al. and Tan found that Yang's scheme is vulnerable to unknown key-share attack, man-in-the-middle attack, impersonation, and parallel session attack [8, 9]. In the same paper, Tan et al. proposed a modified *3PAKE* using the *ECC* but recently, Nose [10] proved that Tan's protocol suffers from impersonation and the man-in-the-middle attack.

Abundant work has been done in *3PAKE*relevant to password-based authentication, traditional public key cryptosystem *(PKC)* and without server's public key

[11-14]. This causes very high computation loads and the communication costs of transmission. To simplify the complexity of certificate management Shamir [15] introduced ID-based cryptosystem where the public keys of each user are easily computed using user's identity. To solve the issue of high computational and communication load, elliptic curve cryptography *(ECC)* is a suitable solution. *ECC* was first proposed by Miller [16] and Koblitz [17] and its security was based upon the difficulty of solving elliptic curve discrete logarithm problem *(ECDLP)*. Compared with traditional public key cryptosystem, *ECC* provides better performance as it can achieve the same security with a smaller key size.

1.1Our contribution

In literature, several *3PAKE* protocols were proposed based on *PKC* and modular exponentiation which suffers from public key management and high computational cost. Therefore, we propose an ID-based authenticated three-party key exchange protocol based on *ECC* in this paper. The characteristics of our, scheme is described as follows:

1. Avoidance of certificate management: Since traditional public key cryptosystem used to verify and manage the certificates before establishing a connection between a client and server. However, it suffers from high computation cost; therefore, the ID-based cryptosystem is an ideal solution to eliminate the certificates.
2. Security analysis: The proposed *3PAKE* protocol employs simple hash function, signature analysis, and a single symmetric encryption/decryption process. The security is based on the hardness assumption of elliptic curve computational diffie-hellman *(ECDH)* also it is validated using random oracle model *AVISPA*.
3. Anonymity and Non-repudiation: It is very important to preserve user privacy in the applications like mobile-commerce, e-voting, and secure message transmission, etc., so that an adversary cannot track user's activity. Therefore, anonymity preserves user's personal information. Similarly, non-repudiation is one of the important requirements for a security protocol in any communication network to prevent denial of conducting an operation on a credit card purchase or any on-line transaction.

1.2Organization of the paper

The rest of the paper is organized as follows. A brief review of some basic concepts is described in Section 2. In section 3, we propose our ID-based three-party authenticated key exchange protocol. The security and analysis of the proposed protocol are presented in Section 4. In section 5 performance analysis is depicted. Finally, we conclude in Section 6.

2.Preliminaries

This section introduces two cryptographic techniques used in our proposed scheme: Elliptic curve cryptography and Computational problem.

2.1Elliptic curve cryptography

The security of ECC is based upon the difficulty of *(ECDLP)*.

Let E/F_q be a set of elliptic curve points over a finite field F_q, defined by an equation $y^2 = x^3 + ax + b$, $a, b \in F_p$ where $(4a^3 + 27b^2) \neq 0$. The additive elliptic curve group defined as $G_q = \{(x, y) : x, y \in F_q, (x, y) \in E / F_q\} \cup \{O\}$, where the point $"O"$ is known as *"point at infinity"* or *"zero point"*. The definitions about the elliptic curve group as follows.

1 Point Addition: Let *P*, *Q* be two points on the curve shown above, such that *P*+*Q*=*R*, where the line joining *P* and *Q* intersects the curve at negative *R*, and the reflection towards *x-axis* is *R*.

2 Scalar Point Multiplication: It is defined on a cyclic group G_q as $(rP = P + P + ...)(+P\ (r\ times))$, where $k \in Z_q^*$ is scalar.

2.2Computational problem

Definition1. (Elliptic curve discrete logarithm problem (ECDLP)) Given $P, R \in G_q$, where *R=xP* and $x \in Z_q^*$. It is difficult to compute *x* from *R*.

Definition2. (Computational Diffie-Hellman problem (ECDH)) Given $(P,xP,yP) \in (G_q)$ for $x, y \in Z_q^*$, where computation of *xyP* is hard from the group G_q

3.Proposed 3PAKE Protocol

In this section, we propose our ID-based 3PAKE protocol based on ECC. The scheme is divided into three phases: the system initialization phase, the user registration phase, the authenticated key exchange phase.

3.1Initialization phase

In this phase, the server S generates set of system parameters as follows:

1. For this, S chooses an elliptic curve equation $(E_p(a,b) \in F_q)$ as defined in Subsection 2.1.
2. The base point P over $\left(E_p(a,\ b)\right)$ with the order *n* and chooses a private key $(d_s \in Z_n^*)$ and the corresponding public key $\left(Q_s = d_s\right)$.
3. The server chooses three secure one-way hash functions $\left(H_1(.),\ H_2(.), H_3(.)\right)$. $\left(G_p\right)$ represents a cyclic addition group that is generated by $\left(P\right)$ *over* $\left(E_p(a,b)\right)$.
4. The server keeps $\left(d_s\right)$ secret and publishes $\left(\left\{E_p(a,b), P, Q_s, H_1(.), H_2(.), H_3(.)\right\}\right)$

3.2Registration

In this phase, any new user participating in the communication need to register with the server S. The user sends its identity in advance. Later, the server computes a temporary identity known only to designated user and server for preserving anonymity. The user and server perform following steps via a secure channel as follows.

1. The server chooses an integer randomly $(r_u \in Z_p^*)$. and computes, $\left(R_u = r_u P\right)$.
2. S computes $\left(h_u = H_1\left(ID_u \| ID_s \| R_u\right)\right)$, then computes client's private key $\left(U = r_u + h_u.d_s\right)$.
3. Thereafter, computes a temporary identity $(TID_u = H_2(d_s) \oplus H_2(ID_u))$ and generates a random nonce N_s.
4. Later, S sends $\left(\left\{ID_s, U, R_u, TID_u, N_s\right\}\right)$ to every client.
5. Upon receiving from server the client verifies $(U.P \stackrel{?}{=} R_u + H_1(ID_u \| ID_s \| R_u).Q_s)$ if it holds, the client keeps U and publishes $\left(R_u\right)$.

3.3Authentication phase

In this phase, two entities A and B wants to authenticate each other via a server S and generate a session key for future communication, assuming that A as a sender and B as a receiver. The steps are presented as follows.

Round 1.

1. In this round the initiator A randomly picks $(x_a \in Z_p^*)$ as its ephemeral-private-key, then computes $\left(P_{ua} = x_a.P\right)$. Thereafter, he computes $\left(t_1 = H_2\left(T_1\right)\right)$ where T_1 is a time-stamp which denotes the current time.
2. To preserve the anonymity the user A computes following parameters to hide its identity as, $(V_a = TID_a \oplus H_2(ID_a))$, $(M_a = V_a \oplus U_a)$, $(Z_a = H_1(V_a \| U_a) \oplus ID_a)$. Further, he computes $(F_a = V_a \oplus X_a)$ where, $\left(X_a = H_1\left(x_a \| ID_a\right)\right)$.
3. Then, a shared session key is computed as $\left(SK_{as} = H_3\left(TID_a \| ID_s \| R_a \| X_a \| V_a\right)\right)$ followed by a symmetric key encryption is generated as $\left(K_a = \left(K_a = H_3(TID_a \| TID_b \| ID_s \| R_a \| t_1)\right)\right)$.
4. Then user A sends $\left((TID_a,\ N_a,\ Request)\right)$, $((TID_a,\ TID_b, ID_s,\ T_1,\ E_{K_a}[V_a, M_a,)\ (F_a, SK_{as}])$ to B and S respectively.
5. Similarly, B computes and send $\left((TID_b,\ N_b,\ Response)\right)$, $\left((TID_b, TID_a, ID_s, T_2,)\ \left(E_{K_b}\right)\right)$ to A and S respectively.

Round 2.

1. Upon receiving A's initiation message the server S computes $(t_1 = H_2(T_1))$.
2. Then apply a decryption algorithm on $\left(D_{K_a}\right)$ as $(K_a \stackrel{?}{=} H_3(TID_a \| TID_b \| ID_s)\ (\| R_a \| t_1))$. If it holds then check if $(V_a \stackrel{?}{=} H_2(d_s))$, if verified then compute $(X_a = F_a \oplus H_2(d_s))$.
3. Upon confirmation S compute the shared common session key as $(SK_{as} = H_3(TID_a \| ID_s \| R_a \| X_a \| H_2(d_s)))$.
4. After successful verification S computes $\left(C_{sa} = H_3\left(TID_a \| TID_b \| R_a \| R_b \| P_{ua}\right)\ \left(\| P_{ub} \| SK_{sa}\right)\right)$ for user A. Similarly, compute $\left(C_{sb} = H_3\left(TID_a \| TID_b \| R_a \| R_b \|\right)\ \left(P_{ua} \| P_{ub} \| SK_{sb}\right)\right)$ for user B.
5. After which, S sends $\left(\left(ID_s, TID_b, P_{ub},\ E_{K_{sa}}[R_b, C_{sa}]\right)\right)$ to user A where the key $\left(K_{sa} = H_3\left(ID_s \| TID_b \| H_2\left(d_s\right) \| P_{ub}\right)\right)$. Similarly, sends

$\left((ID_s, TID_a, P_{ua},)\ \left(E_{K_{sb}}[R_a, C_{sb}]\right)\right)$ to user B where $\left(K_{sb} = H_3\left(ID_s \| TID_a \| H_1(d_s) \| P_{ua}\right)\right)$.

Round 3.

1. Upon receiving, user A first applies a decryption algorithm as $(K_{sa} \stackrel{?}{=} H_3(ID_s \|))\ (TID_b \| H_2(d_s) \| P_{ub}))$.
2. Then verifies $(C_{sa} \stackrel{?}{=} H_3(TID_a \| TID_b \| R_a \| R_b \| P_{ua} \| P_{ub} \| SK_{as}))$ if it holds then user A generate a time-stamp T_3 then compute $\left(t_3 = H_2\left(T_3 \| N_b\right)\right)$ where N_b is a random nonce sent by B in Round 1.
3. A signature is generated to prevent repudiation by a legitimate but curious user as $(\sigma_a = (x_a + t_3).P_{ub})$ followed by a common shared key $\left(K_{ab} = x_a.P_{ub}\right)$.
4. Thereafter, A send $((TID_a, TID_b, E_{K_{ab}}[T_3, \sigma_a, N_b]))$ to B.
5. Later, compute the shared session key $(K = H_1(K_{ab} \| N_a \| N_b \| R_a \| R_b \| \sigma_a))$ and the session key $\left(SK_{ab} = H_3\left(TID_a \| TID_b \| P_{ua} \| P_{ub} \| R_a \| R_b \| K\right)\right)$ for further communication.

Round 4.

1. After receiving B computes $(K_{ab} = x_b.P_{ua})$ then applies a decryption algorithm.
2. Then computes $(t_3 = H_2(T_3 \| N_b))$ and verifies the signature as $(\sigma_a = (P_{ua} + t_3.P).x_b)$, if it holds then compute a session key $\left(SK_{as} = H_3\left(TID_a \| TID_b \| P_{ua}\right)\ \left(\| P_{ub} \| R_a \| R_b \| K\right)\right)$ where $(K = H_1(K_{ab} \| N_a \| N_b \| R_a \| R_b \| \sigma_a))$ for future communication.

4. Security model

4.1Security analysis

This section provides the provable security analysis in random oracle model [18] against chosen message attack and a formal security verification of the proposed 3PAKE scheme using AVISPA tool.

Theorem 1. The user with identity $(ID_u \in \{A, B\})$ accepts the private key U if $\left(U.P = R_u + H_1\left(ID_U \| ID_S \| R_u\right).Q_s\right)$ holds.

Proof. The correctness of the verification phase is proved as follows:

The user receive $\left(R_u, U\right)$ along with server's ID $\left(ID_s\right)$ then he computes

$$U.\mathrm{P} = R_u + H_1\left(ID_u \| ID_s \| R_u\right).Q_s)$$
$$= r_u.P + h_A.d_s.P\ )$$
$$= U.P$$

Theorem 2. The signer ID_a generates the signature (σ_a) then send to B who can accept if his authenticity is proved based on the condition if $(\sigma_a = \hat{\sigma}_a)$ holds.

Proof. The correctness of the verification phase is proved as follows:

B receives $([T_3, \sigma_a, N_a])$ using which he computes following parameters $(\hat{t}_3)$ then computes

$$\hat{\sigma}_a = (P_{ua} + \hat{t}_3.P).x_b)$$
$$= (x_a.x_b.P + \hat{t}_3.P.x_b)\ )$$
$$= (x_a.P_{ub} + \hat{t}_3.P_{ub})\ )$$
$$= (x_a + \hat{t}_3).P_{ub}\ )$$

Theorem 3 The identity of a participated user should not be revealed even if an adversary tries to read a transmitted message and also it should not be distinguishable if either two different sessions are initiated by same user then, the scheme achieves anonymity.

Proof. In our protocol, the identity of user $(\in \{A, B\})$ is hidden in $(Z_a = H_1(V_a \| U_a) \oplus ID_a)$, which contains the private key generated by S which is initiated at each session and is protected by secret key $\left(V_a = H_2\left(d_s\right)\right)$. Therefore, the message is fresh in each session and even if an adversary captures the identity TID_u he will be unable to link any two sessions executed before.

Theorem 4 The proposed 3PAKE protocol is provably secure against adaptively chosen message attack in the random oracle under the Elliptic curve computational diffie-hellman (ECDH) assumption. *Proof.* Assume that the proposed 3PAKE scheme can be forged under the adaptive chosen message attack by a probabilistic polynomial time adversary whose goal is to intrude into the system and break our scheme. For the same, we construct an algorithm C that helps an adversary A to solve the (ECDH) problem by responding with *(xyP)* as output from *(P,xP,yP)*, where $x, y \in Z_p^*$.

Initialization: Initially, C chooses a $Q_c \in Z_p^*$ to solve the (ECDH) problem, by setting $Q_s = Q_c$ and finally C gives returns the system parameters $\backslash\left(\left\{E_p(a,b), P, Q_c, H_1(.), H_2(.), H_3(.)\right\}\right)$ to A and responses the queries made by \(A\) as follows:\\

Hash queries to (H_1): Initially, C holds an empty $(L_{H_1}^{list})$ list. Each entry in the list is a tuple of the form $\left(\left(ID_u, ID_s, R_u, h_u\right)\right)$, $\left(\left(V_u, U_u, Z_u\right)\right)$, $\left(\left(x_u, ID_u, X_u\right)\right)$, $(K_{ij}, N_i, N_j, R_i, R_j, \sigma_i, K_i)$ .For each query $\left(\left(ID_u, ID_s, R_u, V_u, U_u, x_u, ID_u\right)\right)$ issued by adversary A to the oracle, C either returns a previous value else choose a number $(h_u, Z_u, X_u, K_i \in Z_p^*)$ such that there is no item $((.,.,.,h_u)),((.,.,Z_u)),\ ((.,.,X_u)),\ ((.,.,.,.,.,.,.,K_i))in(L_{H_1}^{list})$ and returns $\left(h_u, Z_u, X_u, K_i\right)$ to A. Now C inserts the tuple $\left(\left(ID_u, ID_s, R_u, h_u\right)\right)$, $\left(\left(V_u, U_u, Z_u\right)\right)$, $\left(\left(x_u, ID_u, X_u\right)\right)$, $(K_{ij}, N_i, N_j, R_i, R_j, \sigma_i, K_i)$ in the $(L_{H_1}^{list})$.

Hash queries to (H_2): Initially, C holds an empty $(L_{H_2}^{list})$ list. Each entry in the list is a tuple of the form $\left(\left(d_s, ID_u, TID_u\right)\right),\left(\left(T_i, N_i, t_i\right)\right),\left(\left(ID_u, V_u\right)\right)$. For each query (TID_u,t_i,V_u) issued by adversary A to the oracle, C either returns a previous value else choose a number $(TID_u, t_i, V_u \in Z_p^*)$ such that there is no item $((.,.,TID_u)),((.,.,t_i)),((.,V_u))inL_{H_2}^{list}$ and returns $\left(TID_u, t_i, V_u\right)$ to A. Now C inserts the tuple $\left(\left(d_s, ID_u, TID_u\right)\right),\left(\left(T_i, N_i, t_i\right)\right),\left(\left(ID_u, V_u\right)\right)$ in the $L_{H_2}^{list}$.

Hash queries to H_3: Initially, C holds an empty $L_{H_3}^{list}$ list. Each entry in the list is a tuple of the form $\left(\left(TID_u, ID_s, R_u, X_u, V_u, SK_u\right)\right)$, $\left(\left(TID_i, TID_j, ID_s, R_u, t_i,\right)\left(K_u\right)\right)$, $\left(\left(TID_i, TID_j, R_i, R_j, P_i, P_j, SK_{us}, C_{su}\right)\right)$, $\left(TID_i, TID_j, P_i, P_j, R_i, R_j, K_i, SK_{ij}\right)$. For each query $\left(\left(TID_u, ID_s, R_u, X_u, V_u\right)\right)$, $\left(\left(TID_i, TID_j, ID_s, R_u, t_i\right)\right)$, $\left(\left(TID_i, TID_j, R_i, R_j, P_i, P_j, SK_{us}\right)\right)$, $\left(\left(TID_i, TID_j, P_i, P_j, R_i, R_j\right)\right)$ issued by adversary A to the oracle, C either returns a previous value else choose a number $SK_i \in Z_p^*$ such that there is no item $((.,.,.,.,.,SK_u)),\ ((.,.,.,.,.,.,K_u)),((.,.,.,.,.,.,.,.)\ (C_{su})),\ ((.,.,.,.,.,.,.,.,SK_i))inL_{H_3}^{list}$ and returns $\left(\left(SK_u, K_u, C_{su}, SK_i\right)\right)$ to A. Now C inserts the tuple $\left(\left(TID_u, ID_s, R_u, X_u, V_u, SK_u\right)\right)$, $\left(\left(TID_i, TID_j, ID_s, R_u, t_i,\right)\left(K_u\right)\right)$, $\left(\left(TID_i, TID_j, R_i, R_j, P_i, P_j, SK_{us}, C_{su}\right)\right)$, $\left(\left(TID_i, TID_j, P_i, P_j, R_i, R_j, K_i, SK_{ij}\right)\right)$ in the $(L_{H_3}^{list})$.

Registration: Initially, C holds an empty list (L_R^{list}) consisting of tuples in the form $\left(\left(ID_u, r_u, R_u\right)\right),\backslash$ $\left(\left(H_2(d_s), H_2(ID_u), TID_u\right)\right)$. If the adversary A asks for *Registration query* with $\left(ID_u, TID_u\right)$, C searches the list $(L_{H_1}^{list})$. If a tuple $\left(\left(ID_u, ID_s, R_u,\right)\left(h_u\right)\right)$, $\left(\left(d_s, ID_u, TID_u\right)\right)$ is found then C does nothing else C selects $(r_u, h_u, d_s \in Z_p^*)$ and compute $(r_u = U_u - h_u.d_s)$, $(R_u = r_u.P)$, $((H_2(d_s) = TID_u \oplus H_2(ID_u))$ and answers as follows:

1. If identity of user $ID_u = ID_c$, for $(ID_u = ID_a, ID_b)$, C returns $((ID_u, \perp, r_c.P))$ to A.
2. Else, the algorithm C returns $\left(\left(ID_u, r_u, R_u\right)\right)$ to A.
3. If C tries to act as S=C, then C returns $((H_2(ID_u), \perp, d_a, TID_u))$ to A.
4. Else, the returns $((H_2(d_s), H_2(ID_u), TID_u))$ to user A.

Finally, C inserts the tuple $\left(\left(d_s, ID_u, TID_u\right)\right)$, $\left(\left(ID_u, r_u, R_u\right)\right), \left(\left(H_2\left(d_s\right), H_2\left(ID_u\right), TID_u\right)\right)$ into the list $(L_{H_1}^{list})$, $(L_{H_2}^{list})$ and (L_R^{list}).

Signature queries: Suppose A submits a signature query with sender's identity ID_a and the receiver's identity ID_b, and then he searches the $L_{H_1}^{list}$ list then generates the signature as given below:

1. If $(TID_u \neq ID_A)$ or ID_b, then C selects a random integers $x_a, x_b \in Z_p^*$ then compute the signature as $(\sigma_c = (x_a + t_i)P_b)$ where $(P_b \in \{Receiver\})$ is the public key of user ID_u.
2. Otherwise, outputs abort the protocol.

Finally, C returns a signature (σ_c) to A for the signer TID_a and the verifier $\left(ID_b\right)$.

Verify queries: When A makes this query to C for the verification of the signature (σ_c) for the signer TID_a and the verifier TID_b, C first checks whether $TID_a, TID_b = TID_i, TID_j$ holds.

1. If it holds, C terminates the session and reports abort $\perp$.
2. Else, recovers the secret key of receiver x_b of TID_b and verifies the algorithm using the verification algorithm.

Send $(\pi_{a,b}^s, m)$**:** Initially, C holds an empty list (L_S^{list}) consisting of $(\pi_{a,b}^s, Trans_{a,b}^s, N_i)$ where $Trans_{a,b}^s$ is the transcript of $\pi_{a,b}^s$, C answers the query as follows:

```
\% OFMC
\% Version of 2006/02/13

SUMMARY
  SAFE
DETAILS
BOUNDED\_NUMBER\_OF\_SESSIONS
PROTOCOL
  C:\SPAN\testsuite\results\edited.if

GOAL
  as\_specified
BACKEND
  OFMC
COMMENTS
STATISTICS

  parseTime: 0.00s
  searchTime: 1.56
  visitedNodes: 499 nodes
  depth: 9 plies
```

Figure 1 Simulation result of OFMC

```
SUMMARY
  SAFE

DETAILS
BOUNDED\_NUMBER\_OF\_SESSIONS
  TYPED\_MODEL

PROTOCOL
  C:\SPAN\testsuite\results\edited.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS

  Analysed  : 0 states
  Reachable  : 0 states
  Translation: 0.08 seconds
  Computation: 0.00 seconds
```

Figure 2 Simulation result of CL-AtSe

1. If $((\pi_{a,b}^s = \pi_{a,b}^T))$ and m is the first message, C searches for the tuple in the form $((t_i, \sigma_i, N_i))$ into the (L_S^{list}). Later, C inserts the tuple $(\pi_{a,b}^T, Trans_{a,b}^T, N_i)$ to the list (L_S^{list}).
2. Else, C looks for (t_i, σ_i, N_i) into (L_S^{list}), chooses a nonce $N_r \in Z_p^*$ and returns (N_r, σ_i, T_i) as the

answer. C updates the tuple indexed by $(\pi)^s_{a,b}$ in the list (L_S^{list}).

Corrupt: This query is responded by C for searching a long lived key as follows:

1. If the identity of user $(TID_u = TID_i)$ or (TID_j), (C) aborts.
2. Else, the algorithm C searches for the list (L_R^{list}) for a tuple $((ID_u, r_u, R_u))$, $((H_2(d_s), H_2(ID_u), TID_u))$ and return $(r_u, H_2(d_s))$ to A.

Reveal $(\pi^s_{a,b}, m)$**:** Initially C holds an empty list L_{RVL}^{list} containing the tuples in the form $(ID_s, ID_r, \pi^s_{a,b}, T^s_{a,b}, SK^s_{a,b})$, where ID_s in the identification of the sender in the session which $\pi^s_{a,b}$ and ID_r represents the receiver identification. C answers the query as follows:

1.If $(\pi^s_{a,b} = \pi^T_{a,b})$, then C aborts the session.

2.If $(ID_u \neq ID_a)$ or ID_b then C searches in the lists L_S^{list}, L_R^{list} and $L_{H_1}^{list}$ for corresponding tuples $((\pi^s_{a,b}, Trans^s_{a,b}, N_i))$, $((ID_u, r_u, R_u))$, $(H_2(d_s), H_2(ID_u), (TID_u))$, $((ID_u, ID_s, R_u, h_u))$, $((V_u, U_u, Z_u))$, $((x_u, ID_u, X_u))$, and $(K_{ij}, N_i, N_j, R_i, R_j, \sigma_u, K_i)$ receptively.

Then C computes $K^s_{a,b} = (x_i.P_{ub})$.

1. C makes a H_3 query. If $(\pi^s_{a,b})$ is the sender oracle A query as $(TID_i, TID_j, P_i, P_j, R_i, R_j, K_i)$.
2. Else, C chooses $SK^s_{a,b} \in Z^*_p$.

Test $(\pi^T_{a,b})$**:** This query is allowed once the adversary asks any of the above mentioned queries for which the oracle $(\pi^T_{a,b})$ must be fresh. If it does not choose any oracles then C aborts the session.
Else, it randomly responds with a value from the set (Z^*_p).

4.2AVISPA

AVISPAs a push-button tool for the automated validation of the Internet security-sensitive protocols and applications [19]. It is considered as a widely accepted simulation tool for formal security verification, which measured whether the security protocol is *SAFE* or *UNSAFE*. It uses a special language called High-Level Protocol Specification Language and integrates the different back-ends that implement a variety of state-of-the-art automatic analysis techniques [20].

The architecture of AVISPA is demonstrated in *Figure 3* This language is based on roles where the AVISPA tool mechanically translates the HLPSL into a lower level specification using a HLPSL2IF translator. Afterward, it generates an intermediate format (IF). The present version integrates four back-ends namely, on-the-fly model-checker (OFMC), CL-based attack searcher (CL-AtSe), SAT-based model-checker (SATMC), and tree-automata-based protocol analyzer (TA4SP).

The OFMC is responsible for symbolic techniques for exploring the state place in a demand driven way. CL-AtSe provides a translation from any security protocol wrote into an intermediate format IF into a set of constraints mainly used to find whether there are attacks on protocols. SAT generates a propositional form then input into a SAT solver and any model found is translated back into an attack.

Finally, TA4SP is responsible for approximating the intruder knowledge using regular tree languages.

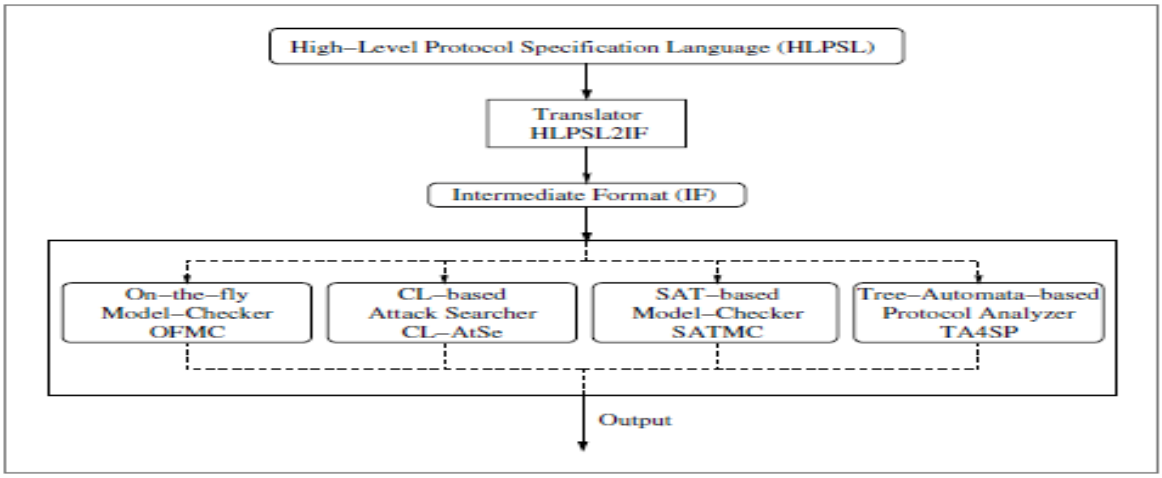

Figure 3 Architecture of AVISPA

Table 1 Security comparisons of the protocol with others

Attribute	Yang Et Al.	Pu Et Al.	Tan Et Al.	Islam Et Al.	Our Scheme
Anonymity	No	No	No	No	Yes
Impersonation attack	No	No	No	Yes	Yes
Non-repudiation	No	No	No	No	Yes
Parallel session attack	No	Yes	Yes	Yes	Yes
Man-in-the-middle-attack	No	No	No	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes	Yes
Known-key security	Yes	Yes	Yes	Yes	Yes
Server-impersonation attack	No	No	No	No	Yes

4.2.1Analysis and specification of result

We have implemented our scheme using the HLPSL language where, we assign three primary roles namely, alice, server, and bob represented as A, S and B respectively. The results ensure that our proposed scheme is safe under OFMC and CL-AtSe back-ends. It is tested using SPAN for AVISPA in *Figure 1* and *Figure 2*. The results ensure that our scheme can withstand popular active attacks, such as the masquerading, replay, and man-in-the-middle attacks, and passive attacks.

4.3BAN Logic

BAN is logic of authentication proposed by Burrows-Abadi-Needham [21, 22].

The goal of BAN logic is to provide trust among communicating parties. It is used to analyse the security of authentication and key distribution protocols. In this section, we first briefly describe the notations used in the \ (BAN\) logic, and after that, we provide the authentication proof.

The notations of the BAN logic are as follows:

- {P, Q}: are the participating entities.
- {X}: message sends in channel.
- {K}: the secret key.
- $\{X\}_K$: message is encrypted with the secret key.
- $P \mid\equiv Q$: P believes in Q.
- $P \lhd X$: P received message X.
- $P \mid\sim X$: P once said X.
- $Q \Rightarrow X$: Q has jurisdiction on X.
- $\#(X)$: X is fresh.
- $P \overset{K}{\leftrightarrow} Q$: K is a shared key between P and Q.

Some rules used in BAN logic as follows:

Message-meaning rule: $\dfrac{P \mid\equiv P \overset{K}{\leftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$

Nonce-verification rule: $\dfrac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$

Jurisdiction rule: $\dfrac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$

Belief-joint rule: $\dfrac{P \mid\equiv Q \sim (X,Y)}{P \mid\equiv Q \sim X}$

Freshness-joint rule: $\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)}$

Additional rule: $\frac{\#(K), P \lhd \{X\}_K, P \mid\equiv P \overset{K}{\leftrightarrow} Q}{P \mid\equiv Q \mid\sim X, P \mid\equiv Q \mid\equiv P \overset{K}{\leftrightarrow} Q}$

Deduction of the proposed protocol

1. Idealization

$MS2: A \rightarrow S, ID_s, TID_b, T_1, \{V_a, M_a, F_a, SK_{as}\}_{K_a}$

$MS4: B \rightarrow S, ID_s, TID_a, T_2, \{V_b, M_b, F_b, SK_{bs}\}_{K_b}$

$MS5: S \rightarrow A, ID_s, TID_b, Pub, \{R_b, C_{sa}\}_{K_{sa}}$

$MS6: S \rightarrow B, ID_s, TID_a, Pua, \{R_a, C_{sb}\}_{K_{sb}}$

$MS7: S \rightarrow A, TID_a, TID_b, \{T_3, \sigma_a, N_b\}_{K_{ab}}$

Note: MS1, MS3 does not contribute to the logic therefore it is omitted from the proof.

2. Initial state assumption

The initial state assumption of S are:

$S \mid\equiv S \overset{K_a}{\leftrightarrow} A$

$S \mid\equiv S \overset{K_b}{\leftrightarrow} B$

$S \mid\equiv S \overset{K_{sa}}{\leftrightarrow} A$

$S \mid\equiv S \overset{K_{sb}}{\leftrightarrow} B$

$S \mid\equiv A \mid\Rightarrow S \overset{SK_{as}}{\leftrightarrow} A$

$S \mid\equiv B \mid\Rightarrow S \overset{SK_{bs}}{\leftrightarrow} B$

$S \mid\Rightarrow C_{sa}$

$S \mid\Rightarrow C_{sb}$

The initial state assumption of A are:

$A \mid\equiv A \overset{K_a}{\leftrightarrow} S$

$A \mid\equiv A \overset{K_{sa}}{\leftrightarrow} S$

$A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

$A \mid\Rightarrow SK_{as}$

$A \mid\equiv S \mid\Rightarrow A \overset{C_{sa}}{\leftrightarrow} S$

The initial state assumption of B are:

$B \mid\equiv B \overset{K_b}{\leftrightarrow} S$

$B \mid\equiv B \overset{K_{sb}}{\leftrightarrow} S$

$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} A$

$B \mid\Rightarrow SK_{bs}$

$B \mid\equiv S \mid\Rightarrow B \overset{C_{sb}}{\leftrightarrow} S$

3. Annotation

$S \lhd \{V_a, M_a, F_a, A \overset{SK_{as}}{\leftrightarrow} S, \#(A \overset{SK_{as}}{\leftrightarrow} S)\}_{K_a}$

$S \lhd \{V_b, M_b, F_b, B \overset{SK_{bs}}{\leftrightarrow} S, \#(B \overset{SK_{bs}}{\leftrightarrow} S)\}_{K_b}$

$A \lhd \{R_b, A \overset{C_{sa}}{\leftrightarrow} S, \#(A \overset{C_{sa}}{\leftrightarrow} S)\}_{K_{sa}}$

$B \lhd \{R_a, B \overset{C_{sb}}{\leftrightarrow} S, \#(B \overset{C_{sb}}{\leftrightarrow} S)\}_{K_{sb}}$

$B \lhd \{T_3, \sigma_a, N_b, A \overset{K_{ab}}{\leftrightarrow} B\}$

4. Derivation process

According to MS2

$$\frac{S \mid\equiv S \overset{K_a}{\leftrightarrow} A, S \lhd \{V_a, M_a, F_a, SK_{as}\}_{K_a}}{S \mid\equiv A \sim \{V_a, M_a, F_a, SK_{as}\}}$$

$$\frac{S \mid\equiv A \mid\sim \{V_a, M_a, F_a, SK_{as}\}}{S \mid\equiv A \sim \{SK_{as}\}}$$

$$\frac{S \mid\equiv A \mid\sim \{SK_{as}\}, S \mid\equiv A \mid\Rightarrow SK_{as}}{S \mid\equiv SK_{as}}$$

According to MS4

$$\frac{S \mid\equiv S \overset{K_b}{\leftrightarrow} B, S \lhd \{V_b, M_b, F_b, SK_{bs}\}_{K_b}}{S \mid\equiv B \sim \{V_b, M_b, F_b, SK_{bs}\}}$$

$$\frac{S \mid\equiv B \mid\sim \{V_b, M_b, F_b, SK_{bs}\}}{S \mid\equiv B \sim \{SK_{bs}\}}$$

$$\frac{S \mid\equiv B \mid\sim \{SK_{bs}\}, S \mid\equiv B \mid\Rightarrow SK_{bs}}{S \mid\equiv SK_{bs}}$$

According to MS5

$$\frac{A \mid\equiv A \overset{K_{sa}}{\leftrightarrow} S, A \lhd \{R_a, C_{sa}\}_{K_{sa}}}{A \mid\equiv S \sim \{R_b, C_{sa}\}}$$

$$\frac{A \mid\equiv S \mid\sim \{R_b, C_{sa}\}}{A \mid\equiv S \sim \{C_{sa}\}}$$

$$\frac{A|\equiv S|\sim\{C_{sa}\}, A|\equiv S|\Rightarrow C_{sa}}{A|\equiv C_{sa}}$$

According to MS6

$$\frac{B|\equiv B\overset{K_{sb}}{\leftrightarrow}S, B\triangleleft\{R_a, C_{sb}\}_{K_{sb}}}{B|\equiv S\sim\{R_a, C_{sb}\}}$$

$$\frac{B|\equiv S|\sim\{R_a, C_{sb}\}}{B|\equiv S\sim\{C_{sb}\}}$$

$$\frac{B|\equiv S|\sim\{C_{sb}\}, B|\equiv S|\Rightarrow C_{sb}}{B|\equiv C_{sb}}$$

According to MS7

$$\frac{B|\equiv B\overset{K_{ab}}{\leftrightarrow}A, B\triangleleft\{T_3, \sigma_a, N_b\}_{K_{ab}}}{B|\equiv A\sim\{T_3, \sigma_a, N_b\}}$$

$$\frac{B|\equiv\#(N_b)}{B|\equiv\#\{N_b, T_3, \sigma_a\}}$$

$$\frac{B|\equiv A|\sim\{N_b, T_3, \sigma_a\}, B|\equiv\#\{N_b, T_3, \sigma_a\}}{B|\equiv A|\equiv\{N_b, T_3, \sigma_a\}}$$

$$\frac{\#K_{ab}, B|\triangleleft\{N_b, T_3, \sigma_a\}_{K_{ab}}, B|\equiv A\overset{K_{ab}}{\leftrightarrow}B}{B|\equiv A|\sim(N_b), B|\equiv A|\equiv A\overset{K_{ab}}{\leftrightarrow})B}$$

5. Performance evaluation

In this section, we compare the proposed scheme with some existing systems already discussed in the literature with respect to computation cost efficiency.

Table 2 Conversion of various operations

Notations	Definition and conversion
T_{ML}	Time complexity for executing the modular multiplication
T_{PM}	Time complexity for executing the elliptic curve scalar point multiplication $T_{PM}\approx 29T_{ML}$
T_{BP}	Time complexity for executing the billinear pairing operation, $T_{BP}\approx 87T_{ML}$
T_{IN}	& Time complexity for executing the modular inversion operation, $T_{IN}\approx 11.6T_{ML}$
T_H	Time complexity for executing the hash function is negligible

In *Table 1*, we represent the security comparison of existing schemes with ours whereas, *Table 2*. enlist the conversion notations of various operation units with respect to modular multiplication. The computation cost of a symmetric en/decryption algorithm almost takes same computation cost as a hash function therefore it can be overlooked with the ECC point multiplication. The comparison is depicted in *Table 3*.

Table 3 Comparison of communication cost in each protocol

Protocol	User A	User B	Server S	Total cost
Yang et al.		$5T_{PM}$	$2T_{PM}$	$12T_{PM}\approx 348\$T_{ML}$
Pu et al.	$5T_{PM}$	$5T_{PM}$	$2T_{PM}$	$12T_{PM}\approx 348\$T_{ML}$
Tan et al.	$3T_{PM}$	$3T_{PM}$	$T_{PM}+4T_{BP}$	$8T_{PM}+4T_{BP}\approx 580T_{ML}$
Islam et al.	$4T_{ML}$	$4T_{ML}$	$2T_{ML}$	$10T_{PM}\approx 290T_{ML}$
Our Scheme	$3T_{PM}$	$3T_{PM}$	$2T_{PM}$	$8T_{PM}\approx 232T_{ML}$

6. Conclusion

This paper proposes an ID-based authenticated three-party key exchange protocol based on elliptic curve cryptography, where the security of the session lies on ECDHP assumption in the random oracle. The scheme achieves anonymity and non-repudiation with efficiency. Furthermore, the overall computation cost is lesser than existing schemes; also it is validated in AVISPA toolkit and formally verified using BAN logic. The results confirmed that the proposed scheme is secure under OFMC and CL-AtSe back-ends.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Bellovin SM, Merritt M. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In proceedings of IEEE computer society symposium on research in security and privacy 1992 (pp. 72-84). IEEE.

[2] Bellare M, Rogaway P. Entity authentication and key distribution. In annual international cryptology conference 1993 (pp. 232-49). Springer Berlin Heidelberg.

[3] Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks. In international conference on the theory and applications of cryptographic techniques 2000 (pp. 139-55). Springer Berlin Heidelberg.
[4] Choie YJ, Jeong E, Lee E. Efficient identity-based authenticated key agreement protocol from pairings. Applied Mathematics and Computation. 2005; 162(1):179-88.
[5] Chen TH, Lee WB, Chen HB. A round-and computation-efficient three-party authenticated key exchange protocol. Journal of Systems and Software. 2008; 81(9):1581-90.
[6] Yang JH, Chang CC. An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments. Journal of Systems and Software. 2009; 82(9):1497-502.
[7] Schnorr CP. Efficient signature generation by smart cards. Journal of Cryptology. 1991; 4(3):161-74.
[8] Pu Q, Zhao X, Ding J. Cryptanalysis of a three-party authenticated key exchange protocol using elliptic curve cryptography. In international conference on research challenges in computer science 2009 (pp. 7-10). IEEE.
[9] Tan Z. An improvement on a three-party authentication key exchange protocol using elliptic curve cryptography. Journal of Convergence Information Technology. 2010; 5(4):120-9.
[10] Nose P. Security weaknesses of authenticated key agreement protocols. Information Processing Letters. 2011; 111(14):687-96.
[11] Lin CL, Sun HM, Hwang T. Three-party encrypted key exchange: attacks and a solution. ACM SIGOPS Operating Systems Review. 2000; 34(4):12-20.
[12] Chang CC, Chang YF. A novel three-party encrypted key exchange protocol. Computer Standards & Interfaces. 2004; 26(5):471-6.
[13] Lu R, Cao Z. Simple three-party key exchange protocol. Computers & Security. 2007; 26(1):94-7.
[14] Farash MS, Attari MA. A secure and efficient identity-based authenticated key exchange protocol for mobile client–server networks. The Journal of Supercomputing. 2014; 69(1):395-411.
[15] Shamir A. Identity-based cryptosystems and signature schemes. In workshop on the theory and application of cryptographic techniques 1984 (pp. 47-53). Springer Berlin Heidelberg.
[16] Miller VS. Use of elliptic curves in cryptography. In conference on the theory and application of cryptographic techniques 1985 (pp. 417-26). Springer Berlin Heidelberg.
[17] Koblitz N. Elliptic curve cryptosystems. Mathematics of Computation. 1987; 48(177):203-9.
[18] Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols. In proceedings of the ACM conference on computer and communications security 1993 (pp. 62-73). ACM.
[19] AVISPA Web tool: automated validation of internet security protocols and applications. www.avispa-project.org/web-interface/. Accessed 16 June 2016.
[20] HLPSL tutorial, 2006. http://www.avispa-project.org/package/tutorial.pdf. Accessed 16 June 2016.
[21] Burrows M, Abadi M, Needham RM. A logic of authentication. In proceedings of the royal society of London a: mathematical, physical and engineering sciences 1989 (pp. 233-71). The Royal Society.
[22] Wen J, Zhang M, Li X. The study on the application of BAN logic in formal analysis of authentication protocols. In Proceedings of the 7th international conference on electronic commerce 2005 (pp. 744-7). ACM.

This paper is selected from proceedings of National Workshop on Cryptology-NWC 2016 organized at JNN College of Engineering Shimoga, Karnataka, India during 11-13, August 2016.