

An ECC-Based Two-Party Authenticated Key Agreement Protocol for Mobile Ad Hoc Networks

Kavitha Ammayappan^{1,2}, Atul Negi², V. N. Sastry¹ and Ashok Kumar Das³

¹ Institute for Development and Research in Banking Technology, Hyderabad 500 057, India
E-mail: kavithaamayappan@gmail.com, vnsastry@idrvt.ac.in

² Department of Computer and Information Sciences
University of Hyderabad, Hyderabad 500 046, India
E-mail: atulcs@uohyd.ernet.in

³ Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad 500 032, India
E-mail: iitkgp.akdas@gmail.com

Abstract—Mobile ad hoc networks (MANETs) are known to be unprotected due to the nature of message propagation and the openness of public channel. Another important characteristic of MANETs is their being basically energy constrained. While it is known that symmetric key cryptography provides a high degree of secrecy and efficiency, but has a number of significant difficulties for the MANET domain in key distribution, key management, scalability and provision of non-repudiation. Public key cryptography (PKC) on other hand provides solutions to the problems inherent in symmetric key cryptography with authenticated key agreement protocols. However the constraints of MANETs such as mobility of nodes, lack of network services and servers make such a proposition difficult. In this paper, we propose a PKC based new energy efficient two-party mutual authenticated key agreement protocol suitable for MANETs. Its security is based on the elliptic curve discrete logarithm assumption. We provide proof here for the security of the proposed protocol and show its relative better performance when compared with other relevant protocols.

Index Terms—Elliptic curve cryptography, Two-party authentication, Key agreement, Hybrid crypto token, Security.

I. INTRODUCTION

Infrastructure-less networks such as mobile ad hoc networks (MANETs) appear to be more appropriate for communication in hostile environment due to its autonomous property. However, from a security point of view, MANET is vulnerable as various security attacks against such kind of network can be more easily performed than against a wired network. As MANET nodes are autonomous, each node should be self competent enough to prove

its authenticity as well as to verify the authenticity of the node to which it communicates without the assistance of any external infrastructure. Moreover, a node should be capable enough to establish security tunnels among themselves to achieve communication privacy. Through symmetric cryptography, security tunnels can be established among communicating nodes in MANETs to have energy efficiency. Key pre-distribution is one of the solutions for distributing symmetric secret keys among MANET nodes, but it has scalability and key maintenance problems. Therefore, the design of PKI based absolute authenticated two party key agreement protocol can enable each MANET node to be self competent.

Key agreement protocols are fundamental building blocks for ensuring authenticated private communications between participating entities over an untrusted network [1]. A key establishment (agreement) protocol allows two or more entities to establish a shared key for encrypted communications over an insecure network. A two-party key agreement protocol is used to establish a common key between two principals. Both principals contribute some information to derive the shared session key. The first key agreement protocol was proposed by Diffie-Hellman in 1976 [2]. However, the protocol does not enable authentication of the two principals and thus it is susceptible to active attack such as man-in-the-middle attack.

In general, authenticated key agreement protocols require entity authentication and key agreement to be appropriately linked to assure that the session key is established only between the intended principals. Therefore, it is more applicable for MANET environment [3]. Two party authenticated key agreement protocol has certain properties such as known key security, perfect forward

secrecy, key compromise impersonation, unknown key share, implicit key authentication, key confirmation and explicit key authentication which need to be satisfied while designing a protocol for efficacy. Therefore, the design of energy efficient two party authenticated key agreement protocol is essential for MANETs since these properties help MANET nodes in ensuring the completeness of the protocol purpose and thus, this type of protocol makes MANET nodes self competent in the absence of infrastructure.

Several authenticated key agreement protocols [4]–[13], [18] have been proposed in the literature, but most of them have been cryptanalyzed for their vulnerabilities and some of them have been enhanced to overcome the identified vulnerabilities.

Majority of the authenticated key agreement protocols [8], [11], [12], [23] are susceptible to key compromise impersonation attack. Long term secret key compromise can lead to undesirable consequences at least until the corrupted principal discovers that his/her key was compromised. A secure key agreement protocol needs to be resilient to key compromise impersonation (KCI) attack since this security attribute is also related to party corruption. This motivates us to come up with a new efficient and KCI-resilient two-party authenticated key agreement protocol for MANETs.

The rest of the paper is organized as follows. Related works are briefly discussed in Section II. Fundamental algorithmic problems, notations, assumptions and description of the proposed protocol are given in Section III. Section IV presents correctness of the protocol, security analysis with respect to key agreement properties and possible attacks. Performance comparison with respect to computational cost among our proposed scheme and other related schemes is given in Section V. Finally we conclude the paper in Section VI.

II. RELATED WORK

Generally authenticated key agreement protocols are based on various cryptographic techniques like identity-based cryptography, elliptic curve cryptography, etc. In turn, key agreement is based on RSA as well as Diffie-Hellman problem. Kumar et al. [4] have proposed elliptic curve cryptography based authentication and key agreement protocols for server and client environment. Their protocol does not address key compromise impersonation and full forward secrecy. M.A. Strangio [5] has proposed a password authenticated key exchange protocol. The ECMQV protocol [23] is efficient, but it does not prevent key compromise impersonation attack.

Saeedina [7] has proposed an improved key exchange protocol based on Gunther's protocol [6] which in turn has been improved by Hsieh et al. [8]. Tseng et al. [9] have showed that [8] is vulnerable to key compromise impersonation attack. Later in 2009, Holbl and Welzer [10] have proposed two improved two-party identity-based authenticated key agreement protocols based on

[8]. First protocol of [10] is immune against KCI of [8]. Second protocol of [10] is an enhancement of [9].

Wang et al. [11] have proposed an improved identity-based key agreement protocol. However, it does not support key compromise impersonation. Strangio [12] has proposed an efficient two-pass Elliptic Curve Diffie-Hellman key agreement protocol (ECKE-1) and it provides public key authentication and ensures explicit key agreement between communicating nodes. In addition, it has been claimed that the protocol satisfies all desirable attributes of a key agreement protocol. Later, Wang et al. [13] have found the vulnerability of [12] to KCI attack through cryptanalysis and have further proposed an improvement over ECKE-1, has resulted as ECKE-1N, which is KCI resilient. Strangio has revised his protocol [12] and has proposed its KCI-resilient version ECKE-1R [18] at the expense of increased computational overhead.

III. THE PROPOSED PROTOCOL

In this section, we present some fundamental algorithmic problems required for security analysis of our scheme, the set of notations that we make use of them in our scheme and then give description and significance of the proposed hybrid crypto token used in our scheme. Finally, we describe our proposed protocol.

A. Fundamental Algorithmic Problems

1) *Discrete logarithm problem*: The discrete logarithm problem (DLP) is as follows: given an element g in a finite group G whose order is n , that is, $n = |G|$ and another element $h \in G$, find an integer x such that $g^x = h \pmod{n}$. It is relatively easy to calculate discrete exponentiation $g^x \pmod{n}$ given g , x and n , but it is computationally infeasible to determine x given h , g and n , when n is large.

2) *Computational Diffie-Hellman problem*: The computational Diffie-Hellman problem (CDHP) is as follows: given a multiplicative group (G, \cdot) , an element $g \in G$ having order n , and $g^a \pmod{n}$, $g^b \pmod{n}$, find $g^{ab} \pmod{n}$. It is computationally infeasible to determine $g^{ab} \pmod{n}$ given g , n , $g^a \pmod{n}$ and $g^b \pmod{n}$, when n is large.

B. Notations

We use the following notations shown in Table I for describing our proposed protocol.

C. Description and Significance of the Proposed Hybrid Crypto Token

In the proposed approach, we assume that all MANET nodes obtain hybrid crypto token from resourceful TTP in registration phase. The purpose of hybrid crypto token is same as public key certificate. Therefore, hybrid crypto token is used by communicating nodes for ensuring their authenticity in active network phase. Based on the results of Potapally et al.'s experiment [14] we propose hybrid crypto token for achieving computational efficiency. The

TABLE I.
NOTATIONS USED IN THE PROPOSED PROTOCOL.

q	A large prime number
AREq	Authentication Request Packet
ARes	Authentication Response Packet
V_K	Signature verification using key K
TTP	Trusted Third Party
$Token_X$	Hybrid crypt token of node X
a	Long term secret of node A
b	Long term secret of node B
r_A	Ephemeral secret key of node A
r_B	Ephemeral secret key of node B
P	A base point on elliptic curve
Pub_A	$Pub_A = aP$ Long term public key of node A
Pub_B	$Pub_B = bP$ Long term public key of node B
SK_{AB}	Session key generated between node A and B
SK_{BA}	Session key generated between node B and A
$H(\cdot)$	Secure one way hash function
$HMAC(\cdot)$	Keyed message authentication code function
RN_A	Random nonce generated by node A
RN_B	Random nonce generated by node B

significance of hybrid crypto token lies in its architecture which uses two different cryptographic primitives such as

- 1) ECC (Key pair of MANET node is based on ECC) and
- 2) RSA (Key pair of TTP is based on RSA)

and hence it is known as hybrid crypto token.

TABLE II.
HYBRID CRYPTO TOKEN FORMAT

Field Name	Data Type
Version	Integer
Serial Number	Integer
Signature Algorithm	Hash with RSASignature
Issuer	String
Valid From	Time
Valid To	Time
Subject name (Node identifier)	String
Subject's public Key	Bit String
Thumbprint Algorithm	Hash
Thumbprint	Bit String

The proposed hybrid crypto token is shown in Table II. In this token, ECC-based public key of a MANET node is being signed by RSA-based private key of a TTP, whereas in normal digital certificates, the key pairs of both MANET node and TTP are based on same cryptographic algorithm. Hence both the public key of a MANET node and the private key of the TTP are based on same cryptographic algorithm.

In the active network phase of the proposed protocol, mutual authentication between the communicating nodes is achieved by mutually verifying their hybrid crypto tokens issued by TTP, explained in steps 1 and 3 of the next subsection D. This is carried out by verifying the digital signature of the hybrid crypto token with the public key of TTP using RSA verification algorithm. In the case of non-hybrid token, we use either RSA or ECC primitives

for generating the digital signature of the hybrid crypto token. If we use RSA primitives alone in the proposed protocol, only hybrid crypto token verification process consumes less energy. As per Figure 1, the remaining key agreement process performs signature generation and verification operation twice. This process consumes more energy in RSA based token over ECC based one, since RSA signature generation is energy intensive than ECC. If we use ECC primitives alone in the proposed protocol, hybrid crypto token verification process consumes more energy, since ECC verification is energy intensive than RSA. Hence we take the advantage of both ECC and RSA by proposing the hybrid crypto token.

In brief, signature on the hybrid crypto token is generated once by TTP in registration phase and verified as and when required in active network phase. Therefore, in our protocol, expensive RSA signature generation is employed at resourceful TTP side to generate hybrid crypto tokens. Less intensive ECC primitives are employed by handheld nodes in active network phase. Table III shows the energy ratio of the signature generation and verification operations with respect to proposed hybrid token. From this table, it is clear that hybrid crypto token requires less energy consumption over non-hybrid tokens.

D. Description of our Two-Party Authenticated Key Agreement Protocol

We consider two different phases in the proposed protocol as follows:

- **Registration Phase:** In this phase, we consider a TTP known as certifying authority issues a certificate which we refer to as hybrid crypto token to the registered MANET nodes. This is used in the active phase for authenticated key agreement. TTP is not involved during the active phase of the network except in the initial registration phase. During the registration phase, we propose to use RSA primitives, especially for computing digital signature (Thumbprint field of Table II) during the generation of hybrid crypto tokens at CA. Therefore in hybrid crypto tokens, ECC based public key of a MANET node is being signed by RSA based private key of a TTP/CA.
- **Active Phase:** During the active phase of the network, communicating principals may not be connected with the TTP due to geographical separation. In this phase, to conserve the energy of the resource constrained MANET nodes, we propose to use ECC based public key cryptographic primitives for generating key pair and symmetric key among MANET nodes and also for generating and verifying signatures during authenticated key agreement process.

The detailed steps for the proposed two-party authenticated key agreement protocol are as follows.

- **Step 1:** Based on the reception of node B 's beacon, node A verifies its hybrid crypto token and sends an authentication request message to node B .
- **Step 2:** Node A selects r_A randomly, where $1 \leq r_A \leq q - 1$ and then computes $Q_A = r_A \cdot P$. Node

TABLE III.
CIPHER SUITE SELECTION VS ENERGY CONSUMPTION AT NODE LEVEL

Token Type	Node level Key Type and Size	TTP level Key Type and Size	Energy Required for SV(mJ)	Energy Ratio	Energy Required for SG(mJ)	Energy ratio
Proposed hybrid Token	ECC-163	RSA-1024	15.97	1:1	134.20	1:1
RSA based Token	RSA-1024	RSA-1024	15.97	1:1	546.50	1:4
ECC based Token	ECC-163	ECC-163	196.23	1:12	134.20	1:1

SV: Signature Verification, SG: Signature Generation

A also generates a random nonce RN_A . Nonce is a one-time random bit-string, usually used to achieve freshness. It unicasts AReq ($Token_A$, RN_A , Q_A) message to node B .

- **Step 3:** After receiving AReq message, node B first verifies node A 's token. If the signature contained in A 's token is verified using the public key of TTP, $V_{PK_{TTP}}$, then B ensures node A 's registration with TTP. B then generates a random nonce RN_B . Node B extracts the identifier of node A , ID_A from the token $Token_A$.
- **Step 4:** Node B selects randomly an integer r_B in the range $1 \leq r_B \leq q-1$ and computes $Q_B = r_B \cdot P$. It then computes $SK_{BA} = H((r_B + b) \cdot (Q_A + Pub_A) || ID_A || ID_B || RN_A || RN_B)$ as a session secret key between A and B .
- **Step 5:** Node B computes $HMAC_B = H(SK_{BA} || H((Q_{A.x} + Q_{B.x}) || (Q_{A.y} + Q_{B.y}) || ID_A || ID_B || RN_A || RN_B))$. It then constructs a message m consists of RN_A , RN_B , Q_B and $HMAC_B$, that is, $m = RN_A || RN_B || Q_B || HMAC_B$ and generates a signature $sig_B(m)$ on m as $sig_B(m) = (r, s)$ using the private long-term key b of B with the help of ECDSA signature generation algorithm [25], [27]. Node B finally sends ARep ($m, sig_B(m)$) as an authentication reply message to node A .
- **Step 6:** After receiving ARep message, node A first verifies the signature $sig_B(m)$ using the public key of node B with the help of ECDSA signature verification algorithm. If this verification holds, node A further checks whether the received RN_A is equal to the previously generated RN_A . If there is no mismatch between them, node A computes $SK_{AB} = H((r_A + a) \cdot (Q_B + Pub_B) || ID_A || ID_B || RN_A || RN_B)$ as a session secret key between A and B and also computes $HMAC_A = H(SK_{AB} || H((Q_{A.x} + Q_{B.x}) || (Q_{A.y} + Q_{B.y}) || ID_A || ID_B || RN_A || RN_B))$.
- **Step 7:** Node A finally compares computed $HMAC_A$ with received $HMAC_B$ for integrity check. If integrity check holds, as an initiator node A ensures the successful execution of the authenticated key agreement protocol with node B .
- **Step 8:** Node A sends an acknowledgment ($RN_B || HMAC_A || sig_A(RN_B || HMAC_A)$) to node

B . Here $sig_A(m)$ is the signature on message m generated using the long-term private key a of the user A .

- **Step 9:** When node B receives the acknowledgment from node A , B verifies A 's signature $sig_A(RN_B || HMAC_A)$ using the public key of node A . If the signature verification holds, it then checks whether the received RN_B is equal to its previously generated RN_B and the received $HMAC_A$ is equal to its previous $HMAC_B$. If these hold, B also ensures the successful execution of the authenticated key agreement protocol with node A . In this way, both nodes A and B use the secret key for future secret communications.

In summary, our protocol is briefly described in Figure 1.

IV. ANALYSIS OF THE PROPOSED PROTOCOL

In this section, we prove that in the proposed protocol, only the intended communicating principals generate the symmetric secret key using the exchanged public parameters and their own private keys through relevant mathematical proofs.

A. Correctness of the Proposed Protocol

In this subsection, the correctness of our proposed protocol are derived as follows.

Theorem 1: The proposed protocol ensures that the intended communicating nodes establish the identical session key at its end.

Proof: The correctness is verified as follows:

$$\begin{aligned}
 SK_{AB} &= H((r_A + a)(Q_B + Pub_B) \\
 &\quad || ID_A || ID_B || RN_A || RN_B) \\
 &= H(r_A Q_B + r_A Pub_B + a Q_B + a Pub_B) \\
 &\quad || ID_A || ID_B || RN_A || RN_B) \\
 &= H((r_A r_B \cdot P + r_A \cdot b \cdot P + a \cdot r_B \cdot P + \\
 &\quad a \cdot b \cdot P) || ID_A || ID_B || RN_A || RN_B) \\
 &= H((r_B + b)(r_A P + a P) \\
 &\quad || ID_A || ID_B || RN_A || RN_B) \\
 &= H((r_B + b)(Q_A + Pub_A) \\
 &\quad || ID_A || ID_B || RN_A || RN_B) \\
 &= SK_{BA}.
 \end{aligned}$$

■

Node A	Node B
<p>1. $AReq(Token_A, Q_A, RN_A)$</p> <p>Verifies B's signature $sig_B(m)$. Verifies received $RN_A = ?$ previous RN_A. If these hold, computes $SK_{AB} = H((r_A + a) \cdot (Q_B + Pub_B) ID_A ID_B RN_A RN_B)$, $HMAC_A = H(SK_{AB} H((Q_{A.x} + Q_{B.x}) (Q_{A.y} + Q_{B.y}) ID_A ID_B RN_A RN_B))$. Verifies whether $HMAC_A = ? HMAC_B$. If it holds sends acknowledgment.</p> <p>3. $(RN_B HMAC_A) sig_A(RN_B HMAC_A)$</p> <p>Stores SK_{AB} for secure communication with B.</p>	<p>Verifies A's token. If verification is successful, generates RN_B. Computes $SK_{BA} = H((r_B + b) \cdot (Q_A + Pub_A) ID_A ID_B RN_A RN_B)$, $HMAC_B = H(SK_{BA} H((Q_{A.x} + Q_{B.x}) (Q_{A.y} + Q_{B.y}) ID_A ID_B RN_A RN_B))$. Constructs a message $m = RN_A RN_B Q_B HMAC_B$. Generates $sig_B(m) = (r, s)$.</p> <p>2. $ARep(m, sig_B(m))$</p> <p>Verifies A's signature. If it holds, it then checks whether received $RN_B = ?$ previous RN_B and received $HMAC_A = ?$ previous $HMAC_B$. If these hold, it stores SK_{BA} for secure communication with A.</p>

Fig. 1. The proposed two-party authenticated key agreement protocol.

Theorem 2: If $HMAC_A = HMAC_B$ holds, as an originator of the proposed protocol, node A ensures authenticity of node B .

Proof: From Theorem 1, it follows that $SK_{AB} = SK_{BA}$. Now,

$$\begin{aligned}
HMAC_A &= H(SK_{AB} || H((Q_{A.x} + Q_{B.x}) \\
&\quad || (Q_{A.y} + Q_{B.y}) || ID_A || ID_B \\
&\quad || RN_A || RN_B)) \\
&= H(SK_{BA} || H((Q_{A.x} + Q_{B.x}) \\
&\quad || (Q_{A.y} + Q_{B.y}) || ID_A || ID_B \\
&\quad || RN_A || RN_B)) \\
&= HMAC_B.
\end{aligned}$$

As an originator of the proposed protocol, node A ensures authenticity of node B since $HMAC_A = HMAC_B$ holds. ■

B. Security Analysis of the Proposed Protocol

Followings are the important security attributes [3], [17] of a key agreement protocol. This section investigates the design compliance of the proposed protocol with respect to the following security attributes.

1) *Security analysis against security attributes of an authenticated key agreement protocol:*

- **Key security:** The adversary is unable to compute the session key established by two honest parties in a run of the protocol assuming the intractability of the computational Diffie-Hellman problem (CDHP) in the underlying group.

- **Known-key security:** Each run of a key agreement protocol between a specific pair of MANET nodes A and B should produce a unique session secret key. This property ensures that when the protocol has known key security the knowledge of previous session keys does not allow an adversary to compromise other previous session keys or future session keys.

Suppose that an adversary knows a previous session key derived as $SK1_{AB} = H((r_{A1} + a)(Q_{B1} + Pub_B) || ID_A || ID_B || RN_A || RN_B) = SK1_{BA} = H((r_{B1} + b)(Q_{A1} + Pub_A) || ID_A || ID_B || RN_A || RN_B)$ between node A and B and suppose there is another key established between the same nodes which is $SK2_{AB} = H((r_{A2} + a)(Q_{B2} + Pub_B) || ID_A || ID_B || RN_A || RN_B) = SK2_{BA} = H((r_{B2} + b)(Q_{A2} + Pub_A) || ID_A || ID_B || RN_A || RN_B)$. $SK1_{AB}$ is the product of two terms. One of the terms is a sum of long term and ephemeral private key values and the other term is the sum of long term and ephemeral public key values. The adversary with known previous session key has a negligible probability to compute the present as well as future session keys, since session keys are uncorrelated. Hence the proposed protocol can withstand known-key attack.

- **Perfect forward secrecy:** This property ensures that the compromise of the long term private keys of one or more entities does not lead to the compromise of previously agreed session keys established by honest entities in the presence of a passive adversary.

Suppose the long term secret keys a and b are disclosed and the adversary tries to compute the key $SK_{AB} = H((r_A + a)(Q_B + Pub_B) || ID_A || ID_B || RN_A || RN_B) = SK_{BA} = H((r_B + b)(Q_A + Pub_A) || ID_A || ID_B || RN_A || RN_B)$. Here forward secrecy is achieved by means of the term $r_A r_B P$. However, in order to compute the session key, adversary needs the knowledge of ephemeral private keys r_A and r_B . Solving Q_A and Q_B in order to get r_A and r_B is equivalent to solving elliptic curve discrete logarithm problem (defined in Definition 1). Therefore, the proposed protocol satisfies perfect forward secrecy.

- **Key-compromise impersonation:** When an adversary compromises long term private key a of node A , then an adversary can, of course, impersonate node A . However, a protocol is said to be resistant to key compromise impersonation attack after capturing long term private key a of node A , if an adversary cannot impersonate other entities to node A in a key agreement protocol and obtain the resulting session secret key.

For example, an adversary E which has the knowledge of long term private key a of node A at hand, attempts to establish a valid session key with A by masquerading as another legitimate entity say B . Note that key compromise impersonation attack represents a serious threat since a party may not be immediately aware that his/her private key is compromised.

A detailed description of the KCI attack scenario is examined with respect to proposed protocol. Lets assume $E(B)$ denotes that adversary E is impersonating B to node A . $E(B)$ has the knowledge on the following:

TABLE IV.
INTRUDER KNOWLEDGE

Parameters	Status
a	Compromised
Pub_A, Pub_B	Known
Q_A, A_B	Known
$Token_A, Token_B$	Known
RN_A, RN_B	Known
ID_A, ID_B	Known

- $E(B)$ intercepts Q_A, RN_A and relays it to B without modifications.
- $E(B)$ intercepts $ARep(m, sig_B(m))$ sent from node B to node A .
- Suppose the attacker $E(B)$ computes $Q_{E(B)} = e \cdot P - Pub_B$ for some random $e \in [1, q-1]$. Then $E(B)$ can easily compute the secret key shared with node A , SK_{EA} using node A 's private key a and the known knowledge mentioned in Table IV as $SK_{EA} = H((r_A + a) \cdot (Q_E + Pub_B) || ID_A || ID_B || RN_A || RN_B) = H((r_A + a) \cdot (eP - Pub_B + Pub_B) || ID_A || ID_B || RN_A || RN_B) = H((r_A + a) \cdot eP || ID_A || ID_B || RN_A || RN_B) = H(e(r_A P + aP) || ID_A || ID_B || RN_A || RN_B) =$

$H(e(Q_A P + Pub_A) || ID_A || ID_B || RN_A || RN_B)$. $E(B)$ then can compute $HMAC_{E(B)} = H(SK_{EA} || H((Q_{A,x} + Q_{B,x}) || (Q_{A,y} + Q_{B,y}) || ID_A || ID_B || RN_A || RN_B))$. Possibilities of KCI attack on the proposed protocol is detailed in Figure 2.

- Thus, $E(B)$ can replace Q_B and $HMAC_B$ in the message with $Q_{E(B)}$ and $HMAC_{E(B)}$. Let the modified message be $m' = RN_A || RN_B || Q_{E(B)} || HMAC_{E(B)}$. However, $E(B)$ cannot compute $sig_B(m')$ on the modified message m' on behalf of node B as $E(B)$ does not know the node B 's long-term private key b . As a result, $E(B)$ does not have any ability to change B 's transmitted message sent to node A .

Therefore, impersonating B to A even after knowing the long term secret key a of node A is impossible by the adversarial node E and thus proposed protocol is resilient against KCI.

- **Unknown key-share (UKS):** A key agreement protocol is resistant to unknown key-share attack if a node cannot be coerced into sharing a session key with a different node rather than the one intended without their knowledge. For example, node A cannot be coerced into sharing a key with node B when in fact node A believes the key is shared with node C .

Proposed protocol achieves strong partnering as the symmetric key calculation includes the identities ID_A, ID_B of the participating nodes A and B along with the random numbers RN_A, RN_B generated at their end. As per the proposed protocol, a confirmation message is sent from node A to node B by sending $(RN_B || HMAC_A) || sig_A(RN_B || HMAC_A)$. Confirmation message includes a $HMAC_A$, which is computed using the generated symmetric secret key SK_{AB} and a signature computed over $m = RN_B || HMAC_A$ using private key a . Verification of $sig_B(m)$, $sig_A(RN_B || HMAC_A)$ at node A and B confirms the generation of same symmetric key and thus the proposed protocol prevents unknown key share attack.

- **Implicit key authentication (IKA):** Because of this property, a communicating node say A is sure that no other node besides a specific second node say B can learn the value of a particular session secret key. Note that the property of implicit key authentication does not necessarily mean that A is assured of B actually possessing the key.

Node A initiates the protocol by sending the $ARep(Token_A, Q_A, RN_A)$ which is destined to node B . According to the protocol design, except intended communicating nodes (i.e., A and B) no other node can derive the particular session secret key. Nodes A and B ensure the generation of the symmetric secret key SK_{AB} by verifying the signatures sig_A, sig_B

Node A	Node E (Attacker)	Node B
1. $AReq(Token_A, Q_A, RN_A)$		<p>Verifies A's token. If verification is successful, generates RN_B. Computes $SK_{BA} = H((r_B + b) \cdot (Q_A + Pub_A) ID_A ID_B RN_A RN_B)$, $HMAC_B = H(SK_{BA} H((Q_{A.x} + Q_{B.x}) (Q_{A.y} + Q_{B.y}) ID_A ID_B RN_A RN_B))$. Constructs a message $m = RN_A RN_B Q_B HMAC_B$. Generates $sig_B(m) = (r, s)$.</p> <p>2. $ARep(m, sig_B(m))$</p>
	<p>E intercepts $ARep(m, sig_B(m))$. E computes $Q_{E(B)} = e \cdot P - Pub_B$, $SK_{EA} = H((r_A + a) \cdot (Q_E + Pub_B) ID_A ID_B RN_A RN_B)$ $= H((r_A + a) \cdot (eP - Pub_B + Pub_B) ID_A ID_B RN_A RN_B)$ $= H((r_A + a) \cdot (eP) ID_A ID_B RN_A RN_B)$ $= H((r_A + a) \cdot (eP) ID_A ID_B RN_A RN_B)$ $= H((r_A eP + a eP) ID_A ID_B RN_A RN_B)$ $= H(e(r_A P + a P) ID_A ID_B RN_A RN_B)$ $= H(e(Q_A P + Pub_A) ID_A ID_B RN_A RN_B)$, and $HMAC_{E(B)} = H(SK_{EA} H((Q_{A.x} + Q_{B.x}) (Q_{A.y} + Q_{B.y}) ID_A ID_B RN_A RN_B))$. Constructs $m' = RN_A RN_B Q_{E(B)} HMAC_{E(B)}$. However, E cannot compute the signature $sig_B(m')$ on behalf of B using B's private key b.</p>	

Fig. 2. Analysis of KCI in the proposed scheme.

respectively.

- **Key confirmation (KC):** Because of this property, the intended communicating parties can ensure that they have actually computed the session secret key. In the proposed protocol, after the successful execution of the two-party authenticated key agreement protocol, nodes A and B ensure the successful key agreement.
- **Explicit key authentication:** A key establishment protocol is said to provide key confirmation if entity A is assured that the second entity B can compute or actually computed the session key. If both implicit key authentication and key confirmation are provided, then the key establishment protocol is said to provide explicit key confirmation. As per our analysis, the proposed protocol satisfies key confirmation and implicit key authentication and thus it also satisfies explicit key authentication.

2) *Security analysis against possible attacks:* In this subsection, we prove that our proposed protocol is secure. An attacker cannot obtain the established session secret key by eavesdropping the messages transmitted over the public channel. We need a security assumption to prove this. Here, we adopt the ECDLP and the collision

resistance property of one-way function to prove the security of our protocol. Several works have proved the security of the ECDLP [15], [16], [19], [22] and is defined as follows.

Definition 1: Let $E_p(a, b)$ be an elliptic curve modulo a prime p : $y^2 = x^3 + ax + b \pmod{p}$. Given two points $P \in E_p(a, b)$ and $Q = kP \in E_p(a, b)$, for some positive integer k . $Q = kP$ represents the point P on elliptic curve $E_p(a, b)$ is added to itself k times. The elliptic curve discrete logarithm problem (ECDLP) is to determine k given P and Q . It is relatively easy to calculate Q given k and P , but it is computationally infeasible to determine k given Q and P , when the prime p is large.

Theorem 3: Under the above assumption of ECDLP, the proposed two party authenticated key agreement protocol is secure. An attacker cannot obtain the established session key by eavesdropping the messages transmitted over the public channel.

Proof: If an attacker needs to compute the session key SK_{AB} computed between nodes A and B , he/she needs to find out the long term private key $a(b)$ and ephemeral

private key r_A (r_B) of either node A or node B from the exchanged transcripts ($Token_A, Q_A, Token_B, Q_B, HMAC_B$). Computing $a(b)$ from Pub_A (Pub_B) is equivalent to solving the elliptic curve discrete logarithm problem (ECDLP). ■

Theorem 4: Under the above security assumption, the proposed protocol achieves mutual authentication and key agreement between the sender (node A) and the intended receiver (node B).

Proof: As per the protocol design, authentication response from node B includes $sig_B(m)$ which is signed by node B 's private key b . Therefore verification of $sig_B(m)$ at node A using B 's public key Pub_B ensures the binding of b with Pub_B and thus authenticates node B and confirms that the symmetric secret key generation at node B . Confirmation message from A to B incorporates $sig_A(m)$ which is signed using node A 's private key a . Successful verification of $sig_A(m)$ using node A 's public key Pub_A ensures the binding of private key a with pub key Pub_A . In addition to that, node B ensures the generation of symmetric secret key at node A . Therefore, the proposed protocol ensures mutual authentication and key agreement between communicating principals. ■

V. PERFORMANCE COMPARISON WITH RELATED SCHEMES

In this section, we compare the computational overhead of the proposed protocol with $ECKE - 1$ [12], $ECKE - 1N$ [13], $ECKE - 1R$ [18] and MQV [23]. The computational overhead is measured in terms of number of energy intensive operations such as scalar multiplications, signature generation and verification used in the protocols.

TABLE V.
NODE LEVEL COMPUTATIONAL OVERHEAD OF VARIOUS PROTOCOLS

Protocols	SM	SG	SV
Proposed scheme	2(3)	1	1
M. A. Strangio [12]	3	—	—
S. Wang et al [13]	2.5	—	—
M. A. Strangio [18]	3(4)	—	1
L. Law [23]	2.5	—	—

SM: Scalar Multiplication
SG: Signature Generation
SV: Signature Verification

Table V presents the computational overhead of the proposed protocol with the existing ECC based two party authenticated key agreement protocols [12], [13], [18], [23]. In the first column of this table, figures within the brackets represent computational overhead without pre-computation and figures outside the brackets represent

with pre-computation. Moreover, our protocol employs RSA based signature verification for validating the Token obtained from TTP. However, it is less energy intensive compared to scalar multiplication and ECC based signature verification operations. Therefore, we do not consider RSA based signature verification as part of computational overhead calculation.

The proposed protocol satisfies common security properties of a two party authenticated key agreement protocol such as known key security, perfect forward secrecy, key compromise impersonation resilience, unknown key share, implicit key agreement, key confirmation and explicit key confirmation. In the proposed protocol, key confirmation is achieved at both communicating parties whereas in other protocols key confirmation is achieved at one end. Overall, we conclude that the proposed protocol is efficient compared with the existing protocols [12], [13], [18], [23].

VI. CONCLUSION

In this paper we have proposed a new two-party authenticated key agreement protocol for mobile ad hoc networks, based on the mix of ECC and RSA. This hybrid cryptographic approach requires less computational overhead and its suitability for authority based MANET architecture has been proved through our protocol design. Proposed hybrid crypto token offers increased level of security, compared with that of other protocols. Moreover, the proposed protocol is scalable and has better tradeoff between computational overhead and security, compared to the existing protocols. These advantages make the proposed protocol appropriate for securing MANET communication scenarios. In addition, our proposed protocol provides mutual authentication between two parties in order to establish a symmetric secret key shared between them.

ACKNOWLEDGMENT

The authors are grateful for the constructive suggestions and comments of the anonymous reviewers which have improved the content and the presentation of this paper.

REFERENCES

- [1] M. A. Strangio, On the Resilience of Key agreement protocols to Key Compromise Impersonation, EuroPKI'06, Vol. 4043, pp. 233-247, LNCS, 2006.
- [2] W. Diffie and M. E. Hellman, New directions in cryptography, IEEE Transactions Information Theory, Vol. 22, pp. 644-654, 1976.
- [3] W. Diffie, P. V. Oorschot and M. Wiener, Authentication and Authenticated Key Exchange Designs, Codes and Cryptography, LNCS, pp. 107-125, 1992.
- [4] K. V. Mangipudi R. S. Katti and H. Fu, Authentication and Key Agreement Protocols Preserving Anonymity, International Journal of Network Security, Vol. 3, No. 3, pp. 259-270, 2006.
- [5] Maurizio A. Strangio, Password-authenticated key exchange using efficient MACs, Journal of Computers, Vol.1, No.8, pp. 27-35,2006.

- [6] C. G. Gunther, An identity-based key-exchange protocol, EuroCrypt'89, Vol. 434, LNCS, pp. 29-37, 1990.
- [7] S. Saeedina, Improvement of Gunther's identity-based key exchange protocol, Electronics Letters, Vol. 36, pp. 1535-1536, 2000.
- [8] B. T. Hsieh, H. M. Sun, T. Hwang and C. T. Lin, An improvement of Saeednia's identity-based key exchange protocol, Information Security Conference, pp. 41-43, 2002.
- [9] Y. M. Tseng, J. K. Jan and C. H. Wang, Cryptanalysis and improvement of an identity-based key exchange protocol, Journal of Computers, Vol. 14, pp. 17-22, 2002.
- [10] M. Holbl and T. Welzer, Two improved two-party identity based authenticated key agreement protocols, Computer Standards and Interfaces, Vol. 31, pp. 1056-1060, 2009.
- [11] S. Wang, Z. Cao, K. K. R. Choo and L. Wang, An improved identity-based key agreement protocol and its security proof, Information Sciences, Vol. 179, pp. 307-318, 2009.
- [12] M. A. Strangio, Efficient Diffie-Hellman two-party key agreement protocols based on elliptic curves, In Proc of 20th ACM Symposium on Applied Computing (SAC), pp. 324-331, 2005.
- [13] S. Wang, Z. Cao, M. A. Strangio and L. Wang, Cryptanalysis and Improvement of an Elliptic Curve Diffie-Hellman Key Agreement Protocol, IEEE Communications Letters, IEEE, Vol. 12, Issue 2, pp. 149-151, 2008.
- [14] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, A study of the Energy Consumption Characteristics of cryptographic algorithms and security protocols, IEEE Transactions on Mobile Computing, Vol. 5, pp. 128- 143, 2006.
- [15] ANSI X9.42-2003, Public key cryptography for the financial services industry: Agreement for symmetric keys using discrete logarithm cryptography, ANSI, 2003.
- [16] V. Shoup, Lower bounds for discrete logarithms and related problems, Proceedings of Advances in Cryptology, EuroCrypt'97, LNCS, Vol. 1233, pp. 256-266, 1997.
- [17] S. B. Wilson, D. Johnson and A. Menezes, Key agreement protocols and their security analysis, Proceedings of the 6th IMA International conference on cryptography and Coding, pp. 30-45, 1997.
- [18] M. A. Strangio, Revisiting an efficient elliptic curve key agreement protocol, Cryptology eprint Archive, IACR, Report 081, 2007.
- [19] H. C. Lin and Y. M. Tseng, A scalable ID based pairwise key establishment protocol for wireless sensor networks, Journal of Computers, Vol.18, No. 2, pp. 13-24, 2007.
- [20] N. Koblitz, Elliptic Curves Cryptosystems, Mathematics of computation, Vol. 48, pp. 203-209, 1987.
- [21] R. W. D. Nickalls, A new approach to solving the cubic: Cardan's solution revealed, The Mathematical Gazette, vol. 77, No. 480, pp. 354-359, 1993.
- [22] J. Kar and B. Majhi, A secure deniable authentication protocol based on Bilinear Diffie Hellman algorithm, Cryptology eprint Archive, IACR, 2010.
- [23] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, An efficient protocol for authenticated key agreement, Designs, Codes and Cryptography, Vol. 28, pp. 119 - 134, 2003.
- [24] W. Stallings, Cryptography and Network Security: Principles and Practices, Prentice Hall publisher, 3rd edition, 2003.
- [25] D. Johnson and A. Menezes, The Elliptic Curve Digital Signature Algorithm (ECDSA), Technical Report CORR 99-34, Dept. of C & O, University of Waterloo, Canada, August 23, 1999.
- [26] Digital Signature Standard. FIPS PUB 186-3, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, June 2009.
- [27] H. Z. Liao and Y. Y. Shen, On the Elliptic Curve Digital Signature Algorithm, Tunghai Science, Vol. 8, pp. 109-126, 2006.

Kavitha Ammayappan received her BE in Computer Science and Engineering from the College of Engineering, Anna University, Chennai, in 2001 and her ME in Wireless Technologies from Thiagarajar College of Engineering, (Madurai) Anna University, Chennai in 2005. She is currently a PhD Scholar with the Department of Computer and Information Sciences, University of Hyderabad, India. She works as a research fellow at Institute for Development and Research in Banking Technology, Hyderabad. Her current research interests include security in mobile ad hoc and wireless sensor networks, key management protocols, secure routing protocols and formal verification methods.

Dr. Atul Negi is currently working as an Associate Professor in the Department of Computer and Information Sciences, AI Lab, University of Hyderabad, India. He has research interests in Document Analysis: Handwriting Segmentation and Recognition, Optical Character Recognition of machine printed Telugu Script. Pattern Recognition and its applications: to Systems Security, and Systems Research: Linux system architecture and applications, Mobile Adhoc networks. Dr.Negi is a Senior Member of IEEE, a Co-Founder Member and Moderator of Linux User group of Hyderabad, Life Member of Indian Unit of International Association for Pattern Recognition. He has been associated as an investigator with funded projects from the Ministry of Home Affairs, Ministry of Communications and Information Technology and with Indian Space Research Organization.

Dr. V. N. Sastry is currently working as an Associate Professor at the Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, India since 1999. Prior to this he served at the National Institute of Technology, Tiruchirappalli, Tamilnadu, India for seven years as a faculty member. Dr. Sastry obtained his PhD Degree from the Indian Institute of Technology, Kharagpur in 1994. His areas of research interest are Routing Algorithms, Mobile Adhoc Networks, Access Control Models, Multi-objective optimization, Fuzzy Control and Risk Modelling.

Dr. Ashok Kumar Das is currently working as an Assistant Professor in the Center for Security, Theory and Algorithmic Research of the International Institute of Information Technology (IIIT), Hyderabad 500 032, India. Prior to joining IIIT Hyderabad, he held academic position as an Assistant Professor in Department of Computer Science and Engineering of the International Institute of Information Technology, Bhubaneswar 751 013, India from July 2008 to May 2010. He received his Ph.D. degree in Computer Science and Engineering from the Indian Institute of Technology, Kharagpur, India in April 2009. He received his M.Tech. degree in Computer Science and Data Processing from the Indian Institute of Technology, Kharagpur, India in January 2000. He also received his M.Sc. degree in Mathematics from the Indian Institute of Technology, Kharagpur, India, in 1998. Prior to joining Ph.D, he worked with C-DoT (Centre for Development of Telematics), a premier telecom technology centre of Govt. of India at New Delhi, India from March 2000 to January 2004. His current research interests include cryptography, security in wireless sensor networks, mobile adhoc networks and vehicular adhoc networks, proxy ring signature and remote user authentication. He has published over 20 papers in international journals and conferences in these areas.