CHAPTER 3

AN ANONYMITY-BASED SECURED ON-DEMAND ROUTING SCHEME FOR MOBILE AD HOC NETWORKS

3.1 INTRODUCTION

Many anonymous routing schemes have been suggested to provide privacy and security for MANET recently. A large number of these protocols use the on-demand routing approach. The operations of an on-demand routing scheme are initiated by the source nodes based on the communication demand. Most of the on-demand routing schemes rely on Public-key cryptography (PKC) to achieve anonymity, unlinkability and unobservability (pfitzmann & Hansen 2000) during route discovery and data forwarding. Although asymmetry of PKC can provide better support for privacy and security by enabling anonymity, unlinkability and unobservability features. However, existing anonymous routing protocols support any one or maximum of two features of anonymity, unlinkability and unobservability. As a result, none of the existing schemes provide privacy and security completely or in other words, the packets are not protected as a whole. In this research work, an Anonymity-based secured on-demand routing (ASOR) scheme is proposed to provide the features of anonymity, unlinkability and unobservability to achieve privacy and security in an adversarial environment. The ASOR scheme is designed based on the group signature (Boneh et al 2004), identity-based encryption (Boneh & Franklin 2001) and cryptographic trapdoor boomerang onion (TBO) (Kong & Hong 2003). Before ad hoc networks starts up, each node obtains a group signing key and an identity-based private key from OGM. After obtaining the cryptographic keys, each node interacts
with its neighbor nodes and generates local broadcast keys and session keys anonymously. The local broadcast key has been used for route discovery process and the session key has been used for the data forwarding anonymously.

3.2 PRELIMINARIES

The basic properties of the bilinear pairing (Boneh & Franklin 2001) and the complexity assumption are very much needed for privacy and security. This work has taken up Elliptic curve cryptography (ECC) (Dabholkar & Yow 2004). The notations and basic concept of bilinear pairing used in these works are discussed here:

**Notations:** Let \( p \) be a large prime, and \( Z_p^* \) be \( Z_p \setminus \{0\} \). Let \( N \) be a positive integer. Then write \( Z_N^* \) for the multiplicative group of integers modulo \( N \). Denote \( \varphi(n) \) as the Euler phi function. Let \( H \) and \( H_1 \) be two cryptographic hash functions: \( H : \{0,1\}^r \rightarrow G_1 \), and \( H_1 : \{0,1\}^r \times G_1 \rightarrow G_1 \).

**Bilinear Groups:** Let \( G_i \) and \( G_2 \) be two groups of order \( q \) for some large prime \( p \). The ASOR scheme makes use of bilinear map \( e : G_i \times G_i \rightarrow G_2 \) between these two groups. The map must satisfy the following properties:

1) **Bilinearity:** A map \( e : G_i \times G_i \rightarrow G_2 \) is bilinear if \( e(aP, bQ) = e(P, Q)^{ab} \) for all \( P, Q \in G_i \) and all \( a, b \in Z \).

2) **Non-degeneracy:** The map does not send all pairs in \( G_i \times G_i \) to the identity in \( G_2 \). Observe that \( G_i \) and \( G_2 \) are groups of prime order this implies that if \( P \) is a generator of \( G_i \) then \( e(P, P) \) is a generator of \( G_2 \).

3) **Computability:** There is an efficient algorithm to compute \( e(P, Q) \) for any \( P, Q \in G_i \)
A bilinear map satisfying the above three properties is said to be an admissible bilinear map. To define a bilinear pairing instance generator as a Probabilistic polynomial time (PPT) algorithm \( \mathcal{G} \) that takes an input security parameter \( t \) and returns a uniformly random tuple \( t = (p, G_1, G_2, e, P) \) of bilinear pairing parameters, including a prime number \( p \) of size \( t \), a cyclic additive group \( G_1 \) of order \( p \), a multiplicative group \( G_2 \) of order \( p \), bilinear map \( e: G_1 \times G_1 \rightarrow G_2 \) and a generator \( P \) of \( G_1 \).

### 3.3 SYSTEM DESIGN FOR ASOR SCHEME

#### 3.3.1 Network Scenario Assumptions

The ASOR scheme considers two entities. The first entity is “OGM” and the second one is “the users”. In this work, a user or node has been used alternatively and has the same meaning. It is assumed that each node has limited transmission and reception capabilities. A node can communicate with other nodes in the network directly if it is in the same communication range otherwise, the node communicates with other nodes via multi-hops. The wireless link is assumed to be symmetric means that if a node A can hear another node B’s transmission, the node B can also hear the node A’s transmission. In addition, this work assumes that each node can run its MAC for wireless interfaces in promiscuous mode of operation. In promiscuous mode (Marti et al 2000) of operation, a node can overhear communications of another node if it is in the same proximity even though it does not involve in the communications directly. For example, if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A. This promiscuous mode is not appropriate for all scenarios of ad hoc network, particularly in tactical networks but, it is useful for other scenarios such as ad hoc conferences so as to improve the routing performance. This prevents the traffic analysis based on the MAC addresses.
3.3.2 System Model

The system model for ASOR scheme is depicted in Figure 3.1. It consists of two phases such as initial setup and anonymous routing. During the initial setup, the ASOR scheme employs group signature and identity-based encryption scheme which then generates the group signing key and identity-based private key. Both the schemes are based upon the pairing of elliptic curve cryptography group of order of a large prime of 160-bit long, which is equivalent to the same security strength as the 1024-bit RSA algorithm (Bos et al 2009). Each node in the network establishes a session key and local broadcast key anonymously with each of its neighbors. Then it uses local broadcast key and cryptographic TBO for route discovery and the session key has been used for encryption and decryption process during data transmission.

Figure 3.1 System Model for ASOR
To provide privacy and security in MANETs, anonymity is the most required feature. The detailed discussion of anonymity in terms of unlinkability, unobservability and pseudonymity are based on Items of interest (pfitzmann & Hansen 2000) including sender, recipient, packet, and so on. The ASOR scheme is aims to offer the following anonymity properties:

(i) **Identity Anonymity**: Anonymity of a user means that the user is not identifiable within the network. Therefore, a sender may be anonymous only within a group of potential senders. At the same time, a recipient may be anonymous only within a group of potential recipients. Both anonymity groups may be disjoint, be the same, or they may overlap.

(ii) **Unlinkability**: A sender is anonymous with respect to sending, if and only if the sender is anonymous within the group of potential senders. Likewise, a receiver is anonymous with respect to receiving, if and only if the receiver is anonymous within the group of potential receivers. So, the relationship anonymity is ensured. As a result, the message is protected from outsider attackers and also there is no linkage between two messages as well.

(iii) **Unobservability**: A sender is unobservable means that it is sufficiently undetectable whether any sender within the unobservability group sends. In the same way it is sufficiently undetectable whether any recipient within the unobservability group receives. So, the relationship unobservability is ensured. As a result, any packet is indistinguishable from other packets to outside attackers.
3.4 SYSTEM DESCRIPTION FOR ASOR SCHEME

3.4.1 Initial Setup

In the initial setup, the MANET considers $n$ mobile nodes that have the trustable relationship with the same OGM. The OGM is a network planner who does not enter into the network during communication. The group signature and identity-based encryption scheme have been used to generate group signing key and identity-based private key. Table 3.1 describes the notations that are used in anonymous session key establishment.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>$OGM$</td>
<td>Key server</td>
</tr>
<tr>
<td>$gpk$</td>
<td>Group public key</td>
</tr>
<tr>
<td>$gsk$</td>
<td>Private group signing key only knows to the respective user</td>
</tr>
<tr>
<td>$gm_{sk}$</td>
<td>OGM's private key used to trace the signature of users</td>
</tr>
<tr>
<td>$PR_x$</td>
<td>Identity-based private key based on bilinear pairing</td>
</tr>
<tr>
<td>$PU_x$</td>
<td>Identity-based public key based on bilinear pairing</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Master secret key owned by OGM</td>
</tr>
<tr>
<td>$P$</td>
<td>Generator of elliptic curve group $G_1$</td>
</tr>
<tr>
<td>$p$</td>
<td>170-bit prime number</td>
</tr>
<tr>
<td>$sk_s$</td>
<td>Local broadcast key of node $S$</td>
</tr>
<tr>
<td>$sk_{sx}$</td>
<td>Pairwise session key shared between $S$ and $X$</td>
</tr>
<tr>
<td>$H(m)$</td>
<td>Secure one way hash function</td>
</tr>
<tr>
<td>$SIGN_{gsk}(\ast)$</td>
<td>Signature generation using node $S$ group public key</td>
</tr>
<tr>
<td>$Rtnym_{sx}$</td>
<td>Random route pseudonym shared between $S$ and $X$</td>
</tr>
</tbody>
</table>
3.4.1.1 Generation of group public/private key pairs

Before the start of ad hoc network, the OGM generates a group public key $gpk$ which is known to everyone in the network, and also generates a private group signing key $gsk_x$ for each node $X$ in the network. The group signature scheme ensures anonymity in which no one can reveal the signer’s identity but everyone can verify its validity.

The procedure to obtain group public key and private group signing key by the individual node is as follows:

Given as input a security parameter $l^i$,

Step 1: The bilinear pairing instance generator (Boneh et al 2004, Boneh & Franklin 2001) $\zeta$ generates a tuple of bilinear pairing parameters $t = (p, G_1, G_2, e, P) \leftarrow \zeta(l^i)$ that is also the publicly shared parameters.

Step 2: Choose a hash function $H_2: \{0,1\}^l \rightarrow \mathbb{Z}_p^*$, which is assumed to be a random oracle in the security proofs.

Step 3: Choose $P_0, G, H \in \mathbb{G}_1$ and $x, x_a, x_b \in \mathbb{Z}_p^*$.

Step 4: Compute $P_{pub} = sP$, $\ell_a = e(G,G)^{x_a}$ and $\ell_b = e(G,G)^{x_b}$

Step 5: The group public key is $gpk = (P, P_0, P_{pub}, H, G, \ell_a, \ell_b)$ and the group private key is $gsk = x$. 
3.4.1.2 Generation of public/private key pairs for users

OGM generates public/private key pair for each user based on identity-based encryption scheme. On input a security parameter $l^i$, each user $X$ obtains a private key $PR_X$ and the public key $PU_X$.

The procedure to obtain private and public key by each user $X$ as follows:

Step 1: OGM runs $\zeta$ on input $l^i$ to generate a prime $q$, two groups $G_1$ and $G_2$ of order $p$, and an admissible bilinear map $e : G_1 \times G_1 \rightarrow G_2$.

Step 2: OGM chooses a random generator $P \in G_1$.

Step 3: OGM picks a random number $\gamma \in \mathbb{Z}_p^*$ and set $P_{pub} = \gamma P$.

Step 4: OGM chooses a cryptographic hash function $H_1 : \{0,1\}^* \rightarrow G_1^*$.

Choose a cryptographic hash function $H_2 : G_2 \rightarrow \{0,1\}^n$ for some $n$.

The security analysis will view $H_1, H_2$ as random oracles.

Step 5: Then the identity-based private key for node $X$ is $PR_X = \gamma \times H_1(X)$ and the corresponding public key is $PU_X = (q, G_1, G_2, e, n, P, P_{pub}, H_1, H_2)$.

3.4.2 Anonymous Routing Scheme

The anonymous routing scheme consists of anonymous session key establishment, anonymous route discovery and anonymous data forwarding. During anonymous session key establishment phase, each user establishes a session key anonymously with every neighbor. Then by employing cryptographic TBO the source node initiates the route discovery process to find out a path to the destination node anonymously. After establishing the
anonymous route between source and destination node the packet will be forwarded to the destination anonymously using session keys.

### 3.4.2.1 Anonymous session key establishment

During this phase, each node communicates with its direct neighbor within its proximity. There are $n$ mobile nodes in the network, among the $n$ nodes the mobile node $S$ with a private group signing key $gsk_s$ and identity-based private key $PR_s$ is surrounded by a number of neighboring nodes within its proximity. The mobile $S$ establishes a session key anonymously with the neighboring mobile node $X$. Figure 3.2 illustrates the anonymous session key establishment process.

![Figure 3.2 Anonymous Session Key Establishment](image)

The following procedure shows the process of anonymous session key establishment by the mobile node $S$ and $X$:

Node $S$ - generates a signature and sends to a neighbor node $X$.

1: $S$ generates a random number $d_s \in \mathbb{Z}_p^*$ and calculates $d_sP$, where $P$ is the generator of $G_1$. 
2: It calculates \( r = d_s P \times x_i \pmod{n} \), where \( d_s P \times x_i \) denotes \( x_i \) coordinate of \( d_s P \).

3: It creates a signature of \( r \) using its group private signing key \( gsk_s \) to obtain \( SIGN_{gsk_s}(r) = k^{-1}(H(m) + x_r)(\pmod{n}) \), where \( H \) a secure hash function SHA1. Any one can verify this signature using group public key \( gpk \).

4: It broadcasts \( <r, SIGN_{gsk_s}(r)> \) to its neighborhood.

Neighbor Node \( X \) - verifies a signature received from node \( S \). It also generates its own signature and sends to node \( S \).

5: \( X \) receives the message from \( S \) and verifies the signature in that message. If the verification is successful, \( X \) chooses a random number \( d_x \in \mathbb{Z}_p^* \) and computes \( d_x P \).

6: \( X \) calculates \( k = d_x P \times x_2 \pmod{n} \) where \( d_x P \times x_2 \) denotes \( x_2 \) coordinate of \( d_x P \).

7: It creates a signature \( SIGN_{gsk_x}(r \mid k) \) using its own group signing key \( gsk_x \).

8: Finally \( X \) computes a session key \( sk_{sx} = H(d_s d_x P) \) and replies to \( S \) with a message \( <k, SIGN_{gsk_x}(r \mid k), E_{sk_{sx}}(sk_{sx} \mid r \mid k)> \), where \( sk_{sx} \) is \( X \)'s local broadcast key.

Node \( S \) - verifies a signature received from node \( X \) and computes its own session key.
9: Upon receiving a reply from \(X\), \(S\) verifies the signature. If the signature is valid, \(S\) proceeds to compute session key between \(X\) and itself as \(sk_{sx} = <d_x P>\). \(S\) also generates a local broadcast key \(sk_s\), and sends \(E_{sk_{sx}}(sk_s, |sk_x| r |k|)\) to its neighbor \(X\) to inform \(X\) about the established local broadcast key.

10: \(X\) receives the message from \(S\) and computes the same session key as \(sk_{sx} = H(d_x P)\) and decrypts the message to get the local broadcast key \(sk_s\).

The anonymous session key establishment is designed based on the combination of Elliptic Curve Diffie-Hellman (ECDH) (Blake-Wilson & Menezes 1999) – Group Signature Algorithm. This combination of ECDH – Group Signature has been used to provide authentication and certificates verification purpose. In addition, it inherits the security and implementation properties of the elliptic curve cryptosystems and offers the highest cryptographic strength than all other existing public-key cryptosystems. The smaller key sizes result in smaller system parameters, smaller public-key certificates, bandwidth savings, faster implementations, lower power requirements, and smaller hardware processors.

### 3.4.2.2 Anonymous route discovery

Anonymous route discovery process uses local broadcast key and cryptographic onion to establish a privacy preserving route between source and destination. The route discovery process consists of anonymous route request and anonymous route reply. The anonymous route request messages will be broadcast to the whole network, where the anonymous route reply message is unicast in nature and sent back to the source node only.
Figure 3.3 shows mobile nodes in MANET, a node $S$ wants to communicate with another mobile node $D$ in the network. In this case, the mobile node $S$ is a source and the mobile node $D$ is a destination. The mobile nodes $X$, $Y$ and $Z$ are intermediate forwarding nodes and these nodes are acts as a routers.

![Figure 3.3 Communications Scenario of Mobile Nodes in MANET](image)

The implementation of route discovery and route reply process has been discussed as follows:

**Anonymous Route Request**: The source node $S$ initiates the route discovery procedure and broadcasts the route request packet locally. The format of the route request packet is as follows:

$$\langle ARREQ, seqno, tr_{de}, TBO \rangle$$

where

- $ARREQ$ - Label for anonymous route request packet
- $seqno$ - Random route pseudonym
- $tr_{de}$ - Cryptographic global trapdoor
- $TBO$ - Cryptographic TBO

The cryptographic global trapdoor (Menezes et al 1996) that can only be opened with $D$’s identity-based private key. The cryptographic global trapdoor yields $E_{pu,D}(D, K_{commit})$ and $K_{commit}(D)$ where $D$ is the tag for destination and $K_{commit}$ is a trapdoor commitment key. The concept of
“trapdoor commitment” is one-way function are collision resistant – given a message digest $K_{comm}(D)$, it is computationally hard to find the pre-image of the digest or another pre-image collision that can produce the same digest. The Cryptographic TBO and initially it is constructed by the source node $S$ as $TBO_s = E_{sk_s}(S)$ by using its local broadcast key $sk_s$, which is shown in Figure 3.4.

On receiving the route request message from $S$, the node $X$ tries to open the trapdoor information using its identity-based private key to see whether it is the destination node. To avoid $ARREQ$ broadcasting storm, the node $X$ checks if it has received the same request before by looking up $seqno$ in its cache. If it is not a duplicate $ARREQ$, $X$ caches $seqno$ for a given time to detect multiple receipt of the same $ARREQ$ packet.

Here, the node $X$ is not the destination and its trial fails, so it acts as an intermediate forwarding node. It embeds a random nonce $Nonce_x$ to the cryptographic TBO, encrypts the result by using its own local broadcast key $sk_x$ then broadcasts the $ARREQ$ locally. The format of the cryptographic TBO the node $X$ broadcasts to all of its neighbors is:

$$TBO_x = E_{sk_x}(Nonce_x, E_{sk_s}(S)).$$ (3.1)

The intermediate nodes $Y$ and $Z$ do the same as $X$ does. Finally, the $ARREQ$s reach the destination node $D$ and it successfully decrypts the trapdoor information using its identity-based private key to find out it is the destination node. If the destination node $D$ receives more than one $ARREQ$, then it replies only to the first arrived message and drops the following ones.
**Anonymous Route Reply:** Once the mobile node $D$ knows that it is the destination, then it starts to prepare a reply message to the source node $X$. The route reply messages are forwarded as unicast instead of broadcast to save the communication cost. The format of the route reply message is as follows:

\[ \langle ARREP, Rnym, sk, (pr_{dest}, TBO) \rangle \]

where,

- **ARREP** - Label for anonymous route reply packet
- **Rnym** - Random route pseudonym
- **sk** - Session key. Where $sk_{ij}$ are established by the pair of nodes such as $sk_{sx}, sk_{xy}, sk_{yz}, sk_{zd}$ by the pair of nodes $(S, X), (X, Y), (Y, Z)$ and $(Z, D)$ respectively.
- **pr_{dest}** - Anonymous proof of global trapdoor opening. Which yields $K_{ckey}$ created by the destination node $D$. Any intermediate forwarding node can verify the anonymous proof of trapdoor opening by checking $K_{comm}(D) = K_{ckey}(D)$.
- **TBO** - Cryptographic TBO. A cryptographic onion associated with the route reply message.

The destination node $D$ bounces back the $ARREP$ packet with the above field as unicast message to the source node $X$. In Figure 3.4, the destination node $D$ encrypts the cryptographic onion by using the session key $sk_{zd}$ and forwards the $ARREP$ packet to the neighbor node $Z$ as unicast packet.
The intermediate node $Z$ receives the $ARREP$ packet and verifies the anonymous proof of trapdoor opening by checking $K_{commit}(D) = K_{ckey}(D)$. Then it selects another random route pseudonym and replaces the older pseudonym with newer one and stores the mapping between them in its forwarding table. Also, it peels of the onion and remove its random nonce $Nonce_Z$ which has been sent during route discovery process. After that it sends the $ARREP$ packet to another intermediate node $Y$. In the same way, all the intermediate forwarding nodes repeats the same procedure until the source receives the route reply packet.

Finally the $ARREP$ packet reaches the node $S$. It results in the opening of the cryptographic onion using the local broadcast key which has been established during the anonymous session key establishment phase.

The entire process of constructing the cryptographic onion by the nodes $S$, $X$, $Y$ and $Z$ respectively and the process of opening the constructed cryptographic onion by the nodes $Z$, $Y$, $X$ and $S$ is shown in the Figure 3.4.

![Figure 3.4 TBO Constructions and Opening in ASOR Scheme](image)

Figure 3.4 TBO Constructions and Opening in ASOR Scheme
### 3.4.2.3 Anonymous data forwarding

Once the source node $S$ finds out a path anonymously to the destination node $D$, then it starts sending data packets by using the random route pseudonym and session keys. Figure 3.5 shows that, the data packets have to travel from the source node $S$ to the destination node $D$ via the intermediate nodes $X$, $Y$ and $Z$. The general format of the data packet which is sent by the source node to the destination is as follows:

$$< Nonce , E_{sk_{ij}} (Rnym, TData, E_{sk_{SD}} (payload)) >$$

where

- **Nonce** - Random nonce
- **Rnym** - Random route pseudonym
- **TData** - Packet type
- **$sk_{ij}$** - Session keys. Here the keys such as $sk_{SX}, sk_{XY}, sk_{YZ}, sk_{ZD}$ by the pair of nodes $(S, X)$, $(X, Y)$, $(Y, Z)$ and $(Z, D)$ respectively. These keys are established during the anonymous session key establishment phase
- **payload** - Transmission data

In this example, the data packets sent by the source node $S$ have the following format:

$$< Nonce_{S}, E_{sk_{SX}} (Rnym_{SX}, TData, E_{sk_{SD}} (payload)) >$$
Once the above message is received by the node $X$ from $S$, and the node $X$ knows that the received message is for itself according to the random route pseudonym $Rnym_{SX}$. Then it compares the $Rnym_{SX}$ with the relative random route pseudonym in the forwarding table. After comparison, it knows that the data packet is not for itself and it has to be forwarded to the neighbor $Y$. Then it computes and forwards the following data packet to $Y$. The forwarded data packet is of the following format:

$$< \text{Nonce}_X, E_{sk_{XY}}(Rnym_{XY}, TData, E_{sk_{XD}}(\text{payload}))>$$

In the same way the data packet is further forwarded by all the intermediate nodes $Y$ and $Z$ until it reaches the destination node $D$. At the end, the destination node $D$ receives the following data packet:

$$< \text{Nonce}_Z, E_{sk_{ZD}}(Rnym_{ZD}, TData, E_{sk_{XD}}(\text{payload}))>$$

Finally, the destination node $D$ knows that the received data packet for itself by looking up the routing table. So, it decrypts the payload by using the session key $k_{XD}$. The entire process of data packet transmission from the source node $S$ to the destination node $D$ is illustrated in the Figure. 3.5.

![Figure 3.5 Anonymous Data Forwarding in ASOR Scheme](image-url)
3.5 PRIVACY AND SECURITY ANALYSIS

3.5.1 Privacy Analysis

3.5.1.1 Comparative discussion

The ASOR scheme is compared with MASK. The ASOR scheme authenticates the neighboring nodes by using group signatures, whereas MASK protocol uses one-time pairing-based keys for preserving the privacy. This may be vulnerable to key pair depletion attacks in MASK. In ASOR protocol, per-hop protection provides complete anonymity in terms of unlinkability and unobservability, where as in MASK one-time pairing-based keys are generated by a trusted party in advance, thus it has to face the problem of one-time key depletion. Moreover, the identity information is well protected in ASOR protocol using random route pseudonym, but MASK leaks identity information of the recipient during route discovery process.

3.5.1.2 Discussion on privacy related goals

Anonymity: The ASOR scheme assigns pseudonyms as identities for the mobile nodes instead of real identities. The anonymity is achieved through group signature by using pseudonyms without disclosing the user’s real identity. In addition, the group signature scheme establishes the session keys and local broadcast keys anonymously with per-hop nodes. The route discovery process uses local broadcast keys for route establishment and data forwarding phase uses session keys for data transmission. Hence, the ASOR scheme satisfies anonymity requirement as long as the group signature is secure.

Unlinkability: The cryptographic onion production is implemented by using local broadcast key encryption function which ensures that the cryptanalysts do not understand the relation between the input onion and the output onion.
Only the forwarding mobile node knows that the onion which has been received by and it is produced by the respective predecessor. It is very hard for the cryptanalysts to discover the relation between the producer and recipient of the particular onion. It is proved that the cryptanalysts cannot correlate the route pseudonyms established by cryptographic onions.

**Unobservability:** In ASOR scheme, the mobile nodes involved in routing procedure are anonymous to the other nodes. A mobile node chooses the nonce randomly and uses it only once; there is no relation between pseudonyms which are computed from nonces, because, the mobile nodes with valid session keys can recognize the respective pseudonyms and obtain the plain text by decrypting the corresponding cipher text. Moreover, a mobile node establishes the local broadcast keys and session keys anonymously with its previous or next mobile node. So, no one can know the real identities of the intermediate nodes. So, ASOR scheme preserves the content unobservability according to (pfitzmann & Hansen 2000).

### 3.5.2 Security Analysis

#### 3.5.2.1 Timing and Data Analysis

Data transmission is assumed to be observable, and the adversary can monitor the traffic based-on timing information which is recorded during its transmission.

Let $M$ and $N$ are sets of explicit attributes of a temporal relation schema, $R$. A temporal functional dependency, denoted $M \rightarrow N$, exists on $R$ if, for all instances $r$ of $R$, all snapshots of $r$ satisfy the functional dependency $M \rightarrow N$. Based on the definition, the adversary can use temporal dependency between transmissions to trace the victim message’s forwarding path.
In ASOR scheme, the forwarding mobile node uses random pseudonyms while forwarding the data packets. To prevent timing and data analysis, the forwarding mobile node forwards dummy packets associated with pseudonyms in addition to the original data packets. The pseudonyms associated with original data packets are different from the pseudonyms associated with dummy packets. When all the transmissions mix together, then it is very difficult to the adversary for timing and data analysis.

3.5.2.2 Node compromise

In compromised node attack, firstly the attacker can secretly enter into the network and compromise individual node. Then the attackers can extract cryptographic secrets such as private signing key and identity-based encryption key to establish key with neighboring nodes. This kind of privacy information leakage is unavoidable due to the nature of mobile ad hoc networks. In ASOR scheme, even though the private signing key and identity-based encryption key is compromised by adversary, it does not get useful privacy information from the compromised node. The privacy information only contains the cryptographic secrets of compromised nodes one-hop neighbor. The ASOR scheme implements per-hop authentication and cryptographic TBO for route discovery. So, the compromised node cannot extract location and real identities of the source / destination node of the relaying packets.

3.5.2.3 Collusion attack

The ASOR scheme implements per-hop authentication and key establishment through group signature. In addition, the forwarding mobile node generates meaningful dummy packets depending on the load of the network. The proposed protocol also supports for unobservability, so, it is
impossible for the colluding insiders/outsiders to infer any useful information from the compromised node.

3.5.2.4 Sybil attack

MANET consists of autonomous mobile nodes which forms a decentralized network. Due to its decentralization the mobile nodes in ad hoc network is prone to Sybil attack. In Sybil attack, a mobile node can create multiple fake identities to the other nodes in the network. In ASOR scheme, the centralized key server OGM generates group signing key and identity-based private key for the mobile nodes. This makes it impossible for the adversary to obtain the real identities except the compromised nodes.

3.6 IMPLEMENTATION AND PERFORMANCE ANALYSIS

3.6.1 Simulation Setup

The routing protocols ASOR and MASK are implemented with ns2 simulator version 2.32 for MANET. The DCF of IEEE 802.11 is used as MAC layer in this simulation. The radio propagation model used in this work is two-ray ground model. The radio propagation range of the each mobile node is 250 meters and the channel capacity is 2 Mbits/sec. In the simulation scenario an ad hoc network of size 1500m × 1500m consists of 100 mobile nodes uniformly deployed. The mobile nodes are moving in the field according to the random waypoint model, and their average speeds varied from 0 to 50 m/s. CBR sessions are used to generate data traffic and all the data packets are 512 bytes long. Simulation is done with the benchmarks on a 2-GHz Pentium Dual Core platform. The network scenario parameters and value used for simulation are listed in Table 3.2.
Table 3.2 Simulation Parameters for ASOR and MASK

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>500 s</td>
</tr>
<tr>
<td>Network Size</td>
<td>1500m × 1500m</td>
</tr>
<tr>
<td>Wireless radio Range</td>
<td>250m</td>
</tr>
<tr>
<td>Mobile nodes</td>
<td>100</td>
</tr>
<tr>
<td>Node speed</td>
<td>Varied from 0 to 50 m/s</td>
</tr>
<tr>
<td>Pause time</td>
<td>10 s</td>
</tr>
<tr>
<td>Traffic type</td>
<td>CBR</td>
</tr>
<tr>
<td>Packet size</td>
<td>512-byte</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random waypoint model</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>ASOR and MASK</td>
</tr>
</tbody>
</table>

The performance of ASOR scheme is analyzed based on the performance metrics (1.9) and compared with MASK. The traceable ratio has been used to analyze the efficiency of ASOR scheme. The packet delivery ratio, end-to-end delay, routing overhead and throughput are chosen as routing performance metrics and the results are demonstrated in this section. The routing performance is evaluated with by varying mobility, malicious nodes and pause time.

3.6.2 Results and Discussions

3.6.2.1 Traceable ratio

In the simulation, a percentage of 5, 10 and 50 nodes are marked as eavesdroppers. Figures 3.6, 3.7 and 3.8 depict the traceable ratios over different path lengths of routes for ASOR and MASK. When the path length is only one-hop, where the two protocols expose different amount of information for varying percentage of eavesdroppers.
Figure 3.6 shows the traceable ratio of ASOR and MASK for 5% of eavesdroppers with the number of hops varies from 1 to 10. For ASOR, the number of hops increases the traceable ratio decreases. Whereas in MASK, the traceable ratio is nonlinear that is the traceable ratio is getting up and down while the number of hops increases. In the Figure 3.7 and 3.8, the traceable ratio is decreases for ASOR with 10% and 50% of eavesdroppers when the number of hops increases. In the same way, it is nonlinear for MASK. As a result, when the path grows longer, the traceable ratio does not exceed the percentage of eavesdroppers in case of ASOR, whereas it is not the same for MASK. Because, in MASK the routing information is secured only from external adversaries, once the adversaries become internal then it can send bogus routing messages that are difficult to verify by legitimate nodes. The proposed ASOR scheme is well protected from insider and outsider adversaries through group signature and identity-based encryption scheme. As an outcome, the proposed ASOR scheme outperforms the MASK.

![Figure 3.6 Traceable Ratio with 5 % of Eavesdroppers for ASOR and MASK](image-url)
Figure 3.7  Traceable Ratio with 10 % of Eavesdroppers for ASOR and MASK

Figure 3.8  Traceable Ratio with 50 % of Eavesdroppers for ASOR and MASK
3.6.2.2 Routing performance under mobility

The performance of ASOR scheme is analyzed and the observations are made with respect to the parameters of PDR, end-to-end delay, routing overhead and throughput based on varying mobile node speed from 0 meters per second (m/s), 10 m/s, 20 m/s, 30 m/s, 40 m/s and 50 m/s.

Figure 3.9 demonstrates that, the PDR of ASOR and MASK is 0.96 and 0.95 respectively for no mobility. When the mobility increases the PDR decreases for both in ASOR and MASK due to more frequent route disruption at higher speeds which leads to retransmission and a new route has to be constructed before remaining packets can be sent out. However, in ASOR the PDR is high compared to MASK because the route is anonymous between source node and destination node in ASOR. So, the route disruptions is low when compare to MASK. This leads to high PDR for ASOR than MASK.

![Figure 3.9 Packet Delivery Ratio Vs Mobility for ASOR and MASK](image-url)
Figure 3.10 shows that, the ASOR has the low end-to-end delay than the MASK. When there is no mobility the end-to-end delay is 0.02s for ASOR and 0.05s for MASK. As the mobility increases the delay also is gradually increases for both the protocols. For the mobile node speed 50 m/s the end-to-end delay for ASOR is 0.21s whereas it is 0.23s for MASK. However, there is a delay difference between both the protocols the end-to-end delay for ASOR is lower than the MASK. Because, the local key construction and non-optimal paths, the delay result in longer latency for both the protocols, but comparatively the ASOR scheme has lower delay than the MASK.

![Figure 3.10 End-to-End Delay Vs Mobility for ASOR and MASK](image)

Figure 3.11 shows the number of control packets transmitted for each data packets to deliver successfully. Normally, when mobility increases the routing overhead also increases intern, because both the protocols generate more control packets due to the need of secure route discovery.
From the figure, it has been observed that the routing cost for the MASK and ASOR is very close because both the schemes use almost similar number of control packets for route discovery. However, routing overhead is high for ASOR than MASK because, the complexity is high for ASOR due to cryptographic trapdoor function and onion routing.

Figure 3.12 demonstrates that, the throughput for ASOR and MASK has significant difference. When there is no mobility the throughput for ASOR is $3.65 \times 10^4$ bps whereas it is $3.59 \times 10^4$ bps for MASK. When mobility increases the throughput for both the schemes are decreases. But, there is drastic decrease in the throughput under varied mobile node speed for MASK whereas it is linear decrease for ASOR. Under the mobile node speed of 50 m/s the ASOR protocol provide the throughput of $3.52 \times 10^4$ bps whereas it is $3.01 \times 10^4$ bps for MASK. As a result, the ASOR scheme outperforms the MASK.

![Figure 3.11 Routing Overhead Vs Mobility for ASOR and MASK](image-url)
3.6.2.3 Routing performance under malicious nodes

Obviously as the number of malicious node increases in the network, the number of genuine users decreases. This decreases the possibility of route establishment. As a result, the source node may slow down or even stop sending packets. Both ASOR and MASK protocols are implemented with varying number of malicious node.

Figure 3.13 shows that, comparatively the ASOR gives better performance than MASK when there is no malicious node in the network. In case of MASK, when there is no malicious node the PDR for ASOR is 0.96 whereas it is around 0.92 for MASK. As the number of malicious nodes increases the PDR decreases significantly around 0.74 for MASK whereas it is better for ASOR as 0.85. When the malicious node
increases, there is a poor performance in PDR of MASK because it suffers from compromised nodes. Whereas for ASOR, even though the private signing key and identity-based encryption key is compromised by adversary, it does not get useful privacy information from the compromised node. The privacy information only contains the cryptographic secrets of compromised nodes one-hop neighbor. So, the compromised nodes do not degrade the packet delivery ratio of the proposed ASOR scheme.

Figure 3.14 depicts that the end-to-end delay of the ASOR is low than MASK when the malicious nodes increases from 0 % to 50 %. For both ASOR and MASK the delay is increases when the number malicious node increases in the network. When there is no malicious node in the network, the end-to-end delay for ASOR scheme is 0.06s and for MASK it is 0.07s. While the percentage of malicious node increases in the network, the delay also increases for both the protocols. As a result, under 50% of malicious nodes the end-to-end delay for ASOR is around 0.33s whereas it is 0.35 s for MASK. However, the delay for ASOR scheme is low than that of MASK. This tells that when there is a large number of a malicious node in the networks the performance degradation is high for MASK, because it suffers from DoS attacks induced by internal adversaries. So, the mobile nodes have to spend more time in processing the bogus packets whereas, the ASOR exclude the inside and outside attacker’s from the network through anonymity features.
Figure 3.13  Packet Delivery Ratio Vs Malicious Nodes for ASOR and MASK

Figure 3.14  End-to-End Delay Vs Malicious Nodes for ASOR and MASK
Figure 3.15 shows that the routing overhead on transmitting each data byte. It can be seen from the figure that the proposed ASOR scheme needs more control packet compared to MASK. To send one data byte, the ASOR protocol need 0.20 – 0.35 control packets, whereas the MASK needs to send 0.20–0.33 control packet. The controls packets for ASOR is slightly high than MASK because the cryptographic complexity is high for ASOR.

Figure 3.15 Routing Overhead Vs Malicious Nodes for ASOR and MASK

Figure 3.16 illustrates that the ASOR gives better throughput than MASK when there is no malicious nodes in the network. In case of ASOR, when there is no malicious node the throughput is $3.58 \times 10^4$ bps whereas it is $3.45 \times 10^4$ bps for MASK. As the number of malicious nodes increases the throughput decreases significantly. As a result, when the malicious nodes in the networks are 50% the throughput of ASOR is $3.42 \times 10^4$ bps where as it is
2.96 × 10^4 bps for MASK comparatively very low. Even though there is a gradual increase of malicious nodes in the networks, the ASOR scheme outperforms the MASK and gives better throughput.

![Figure 3.16 Throughput Vs Malicious Nodes for ASOR and MASK](image)

**Figure 3.16 Throughput Vs Malicious Nodes for ASOR and MASK**

### 3.6.2.4 Routing performance under pause time

The performance of ASOR and MASK is analyzed with the parameters of PDR, end-to-end delay, routing overhead and throughput with varying pause time.

Figure 3.17 demonstrates that the packet delivery ratio is increases to maximum when the pause time is increases. The PDR of ASOR is 0.87 and for the MASK it is 0.83 when the pause time is 0s and mobility is 10m/s. When there is an increase in pause time; there is a positive sign on the
performance of the both schemes. The PDR for ASOR is 0.93 and for MASK it is 0.89 for the pause time maximum of 100s. However, the ASOR outperforms the MASK due to the efficient group signature scheme used in ASOR.

Figure 3.18 shows that the end-to-end delay for both the protocols ASOR and MASK is getting low when the pause time is increased. From the figure it has been seen that the end-to-end delay for ASOR is getting decreases from 0.03s to 0.005s and also decreases from 0.03s to 0.008s for MASK when the pause time is increases from 0 sec to 100 sec. Comparatively, the delay is low for ASOR than MASK.

![Figure 3.17 Packet Delivery Ratio Vs Pause Time for ASOR and MASK](image-url)
Figure 3.18 End-to-End Delay Vs Pause Time for ASOR and MASK

Figure 3.19 shows that the routing overhead for transmitting each data byte. The ASOR scheme has need more control packets than MASK. For sending one data packet, the needed control packets for MASK is from 0.35 bytes to 0.05 bytes when there is an increase in the pause time from 0s to 100s. Whereas for ASOR is from 0.38 bytes to 0.08 bytes. When the pause time increases the control bytes required for discovery is decreases for both the protocols because the mobile nodes are in immobility.

Figure 3.20 depicts that the ASOR gives high throughput than MASK when there is an increase in the pause time of the mobile nodes. From the figure it has been seen that, in case of MASK when the pause time is 0stuhe throughput is $3.50 \times 10^4$ bps whereas it is still better for ASOR is $3.66 \times 10^4$ bps. As the pause time increases the throughput of the both schemes increases. As a result, when the pause time of mobile node is 100s the throughput for ASOR is $4.35 \times 10^4$ bps whereas for MASK it is $3.77 \times 10^4$ bps.
Figure 3.19  Routing Packets Overhead Vs Pause Time for ASOR and MASK

Figure 3.20  Throughput Vs Pause Time for ASOR and MASK
3.7 SUMMARY

In this research work, the proposed ASOR scheme is achieved the privacy and security for data communication by adopting anonymity features such as unlinkability, unobservability and pseudonymity during route discovery and data forwarding phase in MANET. To achieve anonymity, the ASOR scheme implements the group signature, identity-based encryption and cryptographic onion. Based on the group signature scheme, each node in the network established the session keys and local broadcast keys anonymously with its one-hop neighbors. Then the anonymous route discovery process is executed to find out a route from source to destination by implementing local broadcast key and onion routing scheme. Finally, the data packets are transmitted anonymously by using session keys and random route pseudonyms. The performance enhancement of the ASOR scheme is ensured based on the results obtained for the metrics traceable ratio, PDR, end-to-end delay and throughput. The routing overhead is high for ASOR than MASK because of its high cryptographic complexity. The simulation results proved that the importance of anonymity for providing privacy and security to enhance routing performance during route discovery and data forwarding phase. Besides route discovery and data forwarding, the anonymity is also needed during event reporting.