secure adhoc routing protocol for privacy reservation

J. Pandu Ranga Rao¹, P.S.S. Pavanganesh², K. Nagaraju³
¹,²,³ECE, Sreyas Institute Engg. And Technology

Abstract- Privacy preserving routing is crucial for some Ad hoc networks that require stronger privacy protection. A number of schemes have been proposed to protect privacy in Ad hoc networks. However, none of these schemes offer unobservability property since data packets and control packets are still linkable and distinguishable in these schemes. In this paper, we define stronger privacy requirements regarding privacy preserving routing in mobile ad hoc networks. Then we propose an Unobservable Secure Routing scheme (USOR) to offer complete unlinkability and content unobservability for all types of packets. USOR is efficient as it uses a novel combination of group signature and ID-based encryption for route discovery. Security analysis demonstrates that USOR can well protect user privacy against both inside and outside attackers. We implement USOR on Network Security (NS2), and evaluate its performance by comparing with Ad Hoc On demand Distance Vector Routing (AODV) and MASK. The simulation results show that USOR not only has satisfactory performance compared to AODV, but also achieves stronger privacy protection than existing schemes like Mask.

Key words- Routing protocols, Security, Privacy, USOR, Anonymity.

I. INTRODUCTION

An Ad-hoc network is a Local Area Network (LAN) that is built spontaneously as devices connect [1]. Instead of relying on a base station to coordinate the flow of messages to each node in the network, the individual network nodes forward packets to and from each other. Privacy protection of mobile ad hoc networks is more demanding than that of wired networks due to the open nature and mobility of wireless media. In wired networks, one has to gain access to wired cables so as to eavesdrop communications. In contrast, the attacker only needs an appropriate transceiver to receive wireless signal without being detected. In wired networks, devices like desktops are always static and do not move from one place to another. Hence in wired networks there is no need to protect user’s mobility behaviour or movement pattern, while this sensitive information should be kept private from adversaries in wireless environments. Otherwise, an adversary is able to profile users according to their behaviours, and endanger or harm users based on such information. Lastly, providing privacy protection for ad hoc networks with low-power wireless devices and low-bandwidth network connection is a very challenging task. A number of anonymous routing schemes have been proposed for ad hoc networks in recent years, and they provide different level of privacy protection at different cost. Most of them rely on public key cryptosystems (PKC) to achieve anonymity and unlinkability in routing. Although asymmetry of PKC can provide better support for privacy protection, expensive PKC operations also bring significant computation overhead. During the route discovery process, each intermediate node creates a one-time public/private key pair to encrypt/decrypt the routing onion, so as to break the linkage...
between incoming packets and corresponding outgoing packets. However, packets are publicly labelled and the attacker is able to distinguish different packet types, which fails to guarantee unobservability as discussed. Meanwhile, both generations of one-time PKC key pairs this can be done during idle time and PKC encryption/decryption present significant computation burden for mobile nodes in ad hoc networks. ARM considered reducing computation burden on one-time public/private key pair generation.

The proposed system is an efficient privacy-preserving routing protocol USOR that achieves content unobservability by employing anonymous key establishment based on group signature. The setup of USOR is simple; each node only has to obtain a group signature signing key and an ID-based private key from an offline key server or by a key management scheme like. The unobservable routing protocol is then executed in two phases. First, an anonymous key establishment process is performed to construct secret session keys. Then an unobservable route discovery process is executed to find a route to the destination.

A. Methodology

In this simulation of networking project we used Network Simulator (NS2.34) for implementation and simulation of our results. NS is an object oriented simulator, written in C++, with an Object Tool Command Language (OTCL) interpreter as a frontend. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy in this document), and a similar class hierarchy within the OTCL interpreter (also called the interpreted hierarchy in this document). The two hierarchies are closely related to each other; from the user’s perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. Users create new simulator objects through the interpreter; these objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy.

B. Traffic Management in Computer Networks

Traffic management is the set of policies and mechanisms that, allow a network to efficiency satisfy a diverse range of service request. The fundamental aspect of traffic management is diversity in user requirement and efficiency satisfying them. Traffic management subsumes many ideas traditionally classified them. Traffic management subsumes many ideas traditionally classified under congestion control as one aspect of traffic management.

Traffic management more general and includes mechanisms such as renegotiation, signalling etc, knowing the elements of network performance will help you better understand how the network performance tools work, and how to interpret the vast amount of information the tools provide.

Network performance is a complex issue, with lots of independent variables that affect hoe clients access servers across a network. However, most of the elements involved in the performance of networks can be boiled down to a few simple network principles that can be measured, monitors and controlled by network administrator with simple-often, free-software. Most network performance tools use a combination of five separate elements to measure network performances:

- Availability
- Response time
- Network utilization
- Network throughput
- Network bandwidth capacity
This section describes each of these elements, and explains how network performance tools use each to measure network performance. Once you establish that there are lost packets in the ping sequence, you must determine what caused the packet losses.

The two biggest causes of lost packets are:

- Collision on a network segment
- Packets dropped by a network device

Dropped packets can also result in packet losses. All network devices contain packet buffers. As packets are received from the network, they are placed in a packet buffer, waiting for their turn to be transmitted.

C. Network Parameters

- **Network Throughput**
  
  In computer networks, throughput is the number of useful bits per unit of time forwarded by the network from a certain source address to a certain destination, excluding protocol overhead, and excluding retransmitted data packets.

  \[
  \text{Transmission Time} = \frac{\text{File Size}}{\text{Bandwidth}} \quad \text{(sec)}
  \]

  \[
  \text{Throughput} = \frac{\text{File Size}}{\text{Transmission Time}} \quad \text{(bps)}
  \]

- **Packet Delivery Ratio (PDR)**
  
  In computer networks, PDR is the ratio of the number of delivered data packet to the destination. This illustrates the level of delivered data to the destination.

  \[
  \text{PDR} = \frac{\sum \text{Number of packet receive}}{\sum \text{Number of packet send}}
  \]

- **End-to-End Delay**
  
  In computer networks, End-to-End Delay is the average time taken by a data packet to arrive in the destination. It also includes the delay caused by route discovery process and the queue in data packet transmission. Only the data packets that successfully delivered to destinations that counted.

  \[
  \text{End-to-End Delay} = \frac{\sum (\text{arrive time} - \text{send time})}{\sum \text{Number of connections}}
  \]

II. MOBILE AD HOC NETWORKS

A Mobile Ad hoc Network (MANET's) is a group of wireless mobile computers in which nodes cooperate by forwarding packets for each other to allow them to communicate beyond direct wireless transmission range. Application such as military exercises, disaster relief, and mine site operation may benefit from ad hoc networking, but secure and reliable communication is a necessary prerequisite for such applications. MANET’s are more vulnerable to attacks than wired networks due to open medium, dynamically changing network topology, cooperative algorithms, lack of centralized monitoring and lack of clear line of defence [1], [2]. Security is a process that is as secure as its weakest link. So, in order to make MANET’s secure, all its weak points are to be identified and solutions to make all those weak points safe are to be considered.

A. MANET

Mobile Ad hoc Network (MANET) is a collection of independent mobile nodes that can communicate to each other via radio waves. The mobile nodes that are in radio range of each other can directly communicate, whereas others need the aid of intermediate nodes to route
their packets as shown in figure 1. These networks are fully distributed, and can work at any place without the help of any infrastructure. This property makes these networks highly flexible and robust. The characteristics of these networks are summarized as follows:

- Communication via wireless means.
- Nodes can perform the roles of both hosts and routers.
- No centralized controller and infrastructure.
- Intrinsic mutual trust.
- Dynamic network topology.
- Frequent routing updates.

“Figure 1. MANET Diagram”

➢ Advantages

The following are the advantages of MANETs:

- They provide access to information and services regardless of geographic position.
- These networks can be set up at any place and time.

➢ Disadvantages

Some of the disadvantages of MANETs are:

- Limited resources.
- Limited physical security.
- Intrinsic mutual trust vulnerable to attacks.
- Lack of authorization facilities.
- Volatile network topology makes it hard to detect malicious nodes.
- Security protocols for wired networks cannot work for ad hoc networks.

➢ Applications

Some of the applications of MANETs are

- Military or police exercises.
- Disaster relief operations.
- Mine cite operations.
- Urgent Business meetings.
B. Routing

The knowledge of routing protocols of MANETs is important to understand the security problems in MANETs. The routing protocols used in MANETs are different from routing protocols of traditional wired world. Some of the reasons are listed below:

- Frequent Route updates.
- Mobility.
- Limited transmission range.

Routing protocols in Mobile Ad hoc Networks are majorly of two categories:
- Proactive Protocols
- Reactive Protocols

Reactive Routing protocols are based on finding routes between two nodes, when it is required. This is different from traditional Proactive Routing Protocols in which nodes periodically sends messages to each other in order to maintain routes. Only Reactive Protocols are considered in this article, as they are extensively studied and used in MANETs. Among many Reactive Routing Protocols, only two of them are described below as they are mostly studied.

C. AODV (Ad Hoc On demand Distance Vector Routing)

It is a reactive routing protocol, meaning that it establishes a route to a destination only on demand. In contrast, the most common routing protocols of the Internet are proactive, meaning they find routing paths independently of the usage of the paths. AODV is, as the name indicates, a distance-vector routing protocol. AODV avoids the counting-to-infinity problem of other distance-vector protocols by using sequence numbers. AODV is capable of both unicast and multicast routing.

2.1 Working

In AODV, the network is silent until a connection is needed. At that point the network node that needs a connection broadcasts a request for connection. Other AODV nodes forward this message, and record the node that they heard it from, creating an explosion of temporary routes back to the needy node. When a node receives such a message and already has a route to the desired node, it sends a message backwards through a temporary route to the requesting node. The needy node then begins using the route that has the least number of hops through other nodes. Unused entries in the routing tables are recycled after a time. When a link fails, a routing error is passed back to a transmitting node, and the process repeats. Much of the complexity of the protocol is to lower the number of messages to conserve the capacity of the network. For example, each request for a route has a sequence number. Nodes use this sequence number so that they do not repeat route requests that they have already passed on. Another such feature is that the route requests have a "time to live" number that limits how many times they can be retransmitted. Another such feature is that if a route request fails, another route request may not be sent until twice as much time has passed as the timeout of the previous route request. The advantage of AODV is that it creates no extra traffic for communication along existing links. Also, distance vector routing is simple, and doesn't require much memory or calculation. However AODV requires more time to establish a connection, and the initial communication to establish a route is heavier than some other approaches.

2.2 Technical description
The AODV Routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path [3]. The major difference between AODV and Dynamic Source Routing (DSR) stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. In an on-demand routing protocol, the source node floods the Route Request packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single Route Request. The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number to determine an up-to-date path to the destination. DSR includes source routes in packet headers. Resulting large headers can sometimes degrade performance—particularly when data contents of a packet are small; AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes. AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate.

2.3 Message Routing

Route Requests (RREQ) are forwarded in a manner similar to DSR. When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source-AODV assumes symmetric (bi-directional) links. When the intended destination receives a Route Request, it replies by sending a Route Reply (RREP). Route Reply travels along the reverse path set-up when Route Request is forwarded. Route Request (RREQ) includes the last known sequence number for the destination. An intermediate node may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender. Intermediate nodes that forward the RREP, also record the next hop to destination as shown in figure 2.

A routing table entry maintaining a reverse path is purged after a timeout interval. A routing table entry maintaining a forward path is purged if not used for an active route timeout interval. A neighbour of node X is considered active for a routing table entry if the neighbour sent a packet within active route timeout interval which was forwarded using that entry. Neighbouring nodes periodically exchange hello message. When the next hop link in a routing table entry breaks, all active neighbours are informed. Link failures are propagated by means of Route Error (RERR) messages, which also update destination.
sequence numbers. When node X is unable to forward packet P (from node S to node D) on link (X, Y), it generates a RERR message. Node X increments the destination sequence number for D cached at node X. The incremented sequence number N is included in the RERR. When node S receives the RERR, it initiates a new route discovery for D using destination sequence number at least as large as N. When node D receives the route request with destination sequence number N, node D will set its sequence number to N, unless it is already larger than N. Routes need not be included in packet headers. Nodes maintain routing tables containing entries only for routes that are in active use. At most one next-hop per destination maintained at each node—DSR may maintain several routes for a single destination. Sequence numbers are used to avoid old/broken routes. Sequence numbers prevent formation of routing loops. Unused routes expire even if topology does not change.

III. SECURE AD HOC ROUTING PROTOCOL

A. Security Basics

Before proceeding further, the reader should have the knowledge of following terminologies of Network Security:

- Symmetric Key Cryptography.
- Public Key Cryptography.
- Authentication and Digital Signatures.
- Hash and Message Authentication Codes (MAC)
- Man-in-the-middle attack, Denial of Service Attack

B. Security Analysis

- Passive attack: Malicious nodes cannot find the sender, receiver and other intermediate node just by eavesdropping on path discovery messages.
- Active attack: Any modification of the path discovery messages will be detected by receiver because of signatures appended, which preserves integrity of message.
- Denial of Service Attack: The protocol is incapable of resisting DOS attack involving flooding the network with meaningless path discovery messages. It is because verification of these messages involves complex computations which are resource consuming. Also it consumes network bandwidth. In fact DOS attack is very difficult to resist in any protocol.

C. Security Problems in MANETs

MANETs are much more vulnerable to attack than wired network. This is because of the following reasons:

- Open Medium - Eavesdropping is easier than in wired network.
- Dynamically Changing Network Topology – Mobile Nodes comes and goes from the network, thereby allowing any malicious node to join the network without being detected.
- Cooperative Algorithms - The routing algorithm of MANETs requires mutual trust between nodes which violates the principles of Network Security.
- Lack of Centralized Monitoring - Absence of any centralized infrastructure prohibits any monitoring agent in the system.
- Lack of Clear Line of Defence - The only use of I line of defence - attack prevention may not suffice. Experience of security research in wired world has taught us that we need to deploy layered security mechanisms because security is a process that is as
secure as its weakest link. In addition to prevention, we need II line of defence -
detection and response.

The possible security attacks in MANETs can be divided into two categories:

- **Route Logic Compromise**: Incorrect routing control messages are injected into
  the network to damage routing logic.
- **Traffic Distortion Attack**: All attacks that prohibit data packets to transfer from
  the source to the destination, either selectively or collectively comes under the
  category of Traffic Distortion Attack. This type of attack can snoop network
  traffic, manipulate or corrupt packet header or contents, block or reply
  transmissions for some malicious purposes.

The list of some of the attacks in MANETs is as follows:
- Jamming.
- Snooping.
- Flood Storm attack.
- Packet Modifications and Dropping.
- Repeater attack.
- Identity Impersonation.
- Black Hole attack.
- Wormhole attack.
- Rushing attack.

**D. Network layer Attacks**

Network layer protocols extend connectivity from neighbouring 1-hops nodes to all
other nodes in MANET. The connectivity between mobile hosts over a potentially multi-hop
wireless link strongly relies on cooperative reactions among all network nodes.

"Figure 3. Illustration of Routing Attack"

A variety of attacks targeting the network layer have been identified and heavily
studied in research papers. By attacking the routing protocols, attackers can absorb network
traffic; inject themselves into the path between the source and destination, and thus control
the network traffic flow, as shown in Figure 3. (a) and (b), whereas malicious node M can
inject itself into the routing path between sender S and receiver D.

The traffic packets could be forwarded to a non-optimal path, which could introduce
significant delay. In addition, the packets could be forwarded to a nonexistent path and get
lost. The attackers can create routing loops, introduce severe network congestion, and
channel contention into certain areas. Multiple colluding attackers may even prevent a source node from finding any route to the destination, causing the network to partition, which triggers excessive network control traffic, and further intensifies network congestion and performance degradation.

E. Anonymous Key Establishment

In this phase, every node in the ad hoc network communicates with its direct neighbours within its radio range for anonymous key establishment [6]. Suppose there is a node S with a private signing key and a private ID-based key KS in the ad hoc network and it is surrounded by a number of neighbours within its power range. S generates a signature ID and sends it the neighbourhood node. A neighbour X of S receives the message from S and verifies the signature in that message. If the verification is successful X computes the session key and replies to S with message. Upon receiving the reply from X, S verifies the signature inside the message. If the signature is valid, S proceeds to compute the session key between and itself also generates a local broadcast key, and sends to its neighbour X to inform X about the established local broadcast key. X receives the message from S and computes the same session key. It then decrypts the message to get the local broadcast key. As a result of this phase, a pair wise session key is constructed anonymously, which means the two nodes establish this key without knowing who the other party is. Meanwhile, node S establishes a local broadcast key, and transmits it to all its neighbours. It is used for per-hop protection for subsequent route discovery.

“Figure 4. Anonymous Key Establishment”

From figure 4 illustrates that the anonymous key establishment process. Note that the messages exchanged in this phase are not unobservable, but this would not leak any private information like node identities. As a result of this phase, a pair wise session key kSX is constructed anonymously, which means the two nodes establish this key without knowing who the other party is. Meanwhile, node S establishes a local broadcast key, and transmits it to all its neighbours. Our key establishment protocol uses elliptic curve Diffie-Hellman (ECDH) key exchange to replace Diffie-Hellman key exchange, and uses group signature to replace MAC code.
F. Encryption and Decryption

In cryptography, encryption is the process of encoding messages (or information) in such a way that eavesdroppers or hackers cannot read it, but that authorized parties can. In an encryption scheme, the message or information (referred to as plaintext) is encrypted using an encryption algorithm, turning it into an unreadable cipher text. This is usually done with the use of an encryption key, which specifies how the message is to be encoded. Any adversary that can see the cipher text should not be able to determine anything about the original message. An authorized party, however, is able to decode the cipher text using a decryption algorithm that usually requires a secret decryption key that adversaries do not have access to. For technical reasons, an encryption scheme usually needs a key generation algorithm to randomly produce keys.

There are two basic types of encryption schemes: Symmetric-key and public-key encryption. In symmetric-key schemes, the encryption and decryption keys are the same. Thus communicating parties must agree on a secret key before they wish to communicate. In public-key schemes, the encryption key is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key and is capable of reading the encrypted messages. Public-key encryption is a relatively recent invention: historically, all encryption schemes have been symmetric-key (also called private-key) schemes. Encryption has long been used by militaries and governments to facilitate secret communication. It is now commonly used in protecting information within many kinds of civilian systems. Encryption is also used to protect data in transit, for example data being transferred via networks (e.g. the Internet, e-commerce), telephones, wireless microphones, wireless intercom systems, Bluetooth devices and bank automatic teller machines. There have been numerous reports of data in transit being intercepted in recent years. Encrypting data in transit also helps to secure it as it is often difficult to physically secure all access to networks.

G. ROUTE DISCOVERY

3.1 Privacy-Preserving Route Discovery

This phase is a privacy-preserving route discovery process based on the keys established in previous phase. Similar to normal route discovery process, the discovery process also comprises of route request and route reply [7]. The route request messages flood throughout the whole network, while the route reply messages are sent backward to the source node only.

Suppose there is a node S (source) intending to find a route to a node D (destination), and S knows the identity of the destination node D. Without loss of generality, assume the three intermediate nodes between S and D. The route discovery process executes as shown in following figure 5.
3.2 Route Request (RREQ)

S (source) chooses a random number, and uses the identity of node D (destination) to encrypt trapdoor information that only can be opened with D’s private ID based key. S then selects a sequence number sequential number for this route request, and another random number NS as the route pseudonym, which is used as the index to a specific route entry. To avoid RREQ broadcasting storm, A will check if he has received the same request before by looking up in his cache, which includes a list of NS and sequential number as shown in the figure 6.

3.4 Route Reply (RREP)

After node D (destination) finds out he is the destination node, he starts to prepare a reply message to the source node. For route reply messages, unicast instead of broadcast is used to save communication cost. D chooses a random number D and computes a cipher text showing that he is the valid destination capable of opening the trapdoor information. After decryption using the right key, A knows this message is a data packet and should be forwarded to B according to route pseudonym NS as seen in figure 7.
Hence it composes and forwards the following packet to B. A session key is computed for data protection. Then he generates a new pair wise pseudonym between c and him and him. At the end, using the pair wise session key computes and sends the following message by looking up in his route table, D knows himself is the destination of this packet. So it is able to decrypt the encrypted payload with the session key as seen in figure 8.

H. Anonymity

User anonymity is implemented by group signature which can be verified without disclosing one’s identity [4], [10]. Group signature is used to establish session keys between neighbouring nodes, so that they can authenticate each other anonymously. And subsequent routing discovery procedure is built on top of these session keys. Hence it is easy to see that USOR fulfills the anonymity requirement under both passive and active attacks, as long as the group signature is secure [9].

3.5 Unlinkability

Let’s consider the three types of packets. In these packets, they are identified by pseudonyms which are generated from random nodes and secret session keys. The nodes are only used once and never reused, and so are the pseudonyms. Except the random nonce and the pseudonym, the remaining part of the message, including the trapdoor information in the route request, is decrypted and encrypted at each hop. Hence even for global adversaries who can eaves drop every transmission within the network, it is impossible for him to find linkage between messages without knowing any encryption key. Even has no idea of the type of the packet being transmitted in the network, and he cannot relate different packets in terms of packet type. The only way to gain information on relationship between transmissions is that the attacker has access to some encryption keys, i.e., has compromised one or more valid nodes.

3.6 Unobservability
In USOR, RREQ, RREP and data packets are indistinguishable from dummy packets to a global outside adversary. Meanwhile, nodes involved in the routing procedure are anonymous to other valid nodes [11]. Consequently, USOR provides unobservability as defined for ad hoc networks. First of all, a global adversary cannot distinguish different packet types, and neither can he distinguish a meaningful cipher text from random noise. Moreover, a node chooses the nonce randomly and never reuses it. Only those mobile nodes with valid session keys can recognize valid pseudonyms and decrypt the corresponding cipher texts to obtain meaningful plaintexts from them. Secondly, a node and its next-hop node or previous-hop node on route establish a session key anonymously, hence no one is able to know real identities of its next-hop node or previous-hop node. Even the source and the destination node do not know real identities of the intermediate nodes on route. As a result, USOR offers content unobservability for ad hoc networks according to the definition. Based on the content unobservability provided by USOR, traffic padding can be introduced into the network to traffic analysis and provide traffic pattern unobservability.

3.7 Node Compromise

Node compromise is easy for the adversary and highly possible in ad hoc networks; hence it is crucial for a privacy-preserving routing protocol to withstand security attacks due to node capture. In this case, privacy information leakage is unavoidable due to secret exposure, while our routing protocol can protect user privacy against serious node compromise. Suppose a node is compromised by an attacker, his private signing key and id-based encryption key are disclosed to the attacker. The attacker now is able to establish keys with neighbouring nodes, but only the following information can be obtained by the attacker:

- The type of a received packet;
- Data/RREP packets sent to/via the compromised node;
- Headers of packets relayed by the compromised node;
- RREQ packets sent from the compromised node’s neighbours.

The attacker is not able to gain more beyond this information. From this information, he cannot infer:

- The location of the source/destination node.
- Real identities of source/destination node of the relaying packets.
- Source/destination node of the RREQ packets.

IV. NETWORK SIMULATION OVERVIEW

A. Basic Simulator Classes

Network Simulator is a discrete-event object simulator to emulate packet-switched networks. It is written in C++. With the Object Tool Command Language (OTCL) interpreter as a front-end. The simulator supports a class hierarchy within the OTCL interpreter. The two hierarchies are closely related to each other; there is one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. The root of this hierarchy is the class OTCL. The interpreted class hierarchy is automatically established through methods defined in the class TCLClass. New simulator objects are created through the interpreter; these objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy. Instantiated objects are mirrored through methods defined in the class TCLObject. There are other hierarchies in C++ code and OTCL scripts; these other hierarchies are not mirrored in the manner of TCLObject.
NS use two languages because simulator has two different tasks to perform. A
detailed simulation of protocols requires a system programming language which can
efficiently manipulates bytes, packet headers, and implement algorithms that run over large
data sets. For these tasks run-time speed is important and turn-around time (run simulation,
find bug, fix bug, recompile, re-run) is less important. Second task is that, a large part of
network research involves slightly varying parameters or configurations, or quickly exploring
a number of scenarios. In these cases, iteration time (change the model and re-rum) is more
important [5]. Since configuration runs once (at the beginning of the simulation), run-time of
this part of the task is less important

OTCL is used for:

- For configuration, setup and simulation of various network topologies.
- To test the effect of various network parameters by manipulating existing C++
  objects.
- Analyze the impact of varying such parameters by tracing and monitoring.

The overall simulator is described by a TCL class simulator. It provides a set of
interfaces for configuring a simulation and for choosing the type of event scheduler used to
drive the simulation. A simulation script generally begins by creating an instance of this class
and calling various methods to create nodes, topologies and configure other aspects of the
simulation.

B. Network Simulator Directory Structure

Suppose that NS2 is installed in directory nsallinone-2.30. Here, directory nsallinone-
2.30 is on the Level 1. On the Level 2, directory tclcl-1.18 contains classes in TCL (e.g.,
TCL, TCLObject, and TCLclass). All NS2 simulation modules are in directoryns-2.30 on the
Level 2. On Level 3, the modules in the interpreted hierarchy are under directorytcl. Among
these modules, the frequently-used ones (e.g., ns-lib.tcl, ns-node.tcl, ns-link.tcl) are stored
under directorylibon Level 4.

![Figure 9. Structure of NS-2 Directory](image)

4.1 Class TCL

The class TCL encapsulates the actual instance of the OTCL interpreter, and provides
the methods to access and communicate with that interpreter. The methods described in this
section are relevant to the ns programmer who is writing C++ code.
The class provides methods for the following operations:

- Obtain a reference to the Tcl instance
- Invoke OTcl procedures through the interpreter
- Retrieve, or pass back results to the interpreter
- Report error situations and exit in an uniform manner
- Store and lookup “OTCL”.
- Acquire direct access to the interpreter.

4.2 Class TCL Object

Class TCLObject is the base class for most of the other classes in the interpreted and compiled hierarchies. Every object in the class TclObject is created by the user from within the interpreter. An equivalent shadow object is created in the compiled hierarchy. The two objects are closely associated with each other. The class TCLclass, described in the next section, contains the mechanisms that perform this shadowing.

4.3 Class TCL Command

This class (class TCL command) provides just the mechanism for ns to export simple commands to the interpreter that can then be executed within a global context by the interpreter.

C. NS2 ARCHITECTURE:

As shown in the simplified user's view of, NS is an Object-oriented Tcl (Otcl)script interpreter that has a simulation event scheduler and network component object libraries, and network set-up (plumbing) module libraries.

* Object-oriented (C++, OTCL)
* Modular approach
* Fine-grained object composition
* Reusability
* Maintenance
* Performance (speed and memory)
* Careful planning of modularity
D. NS2 PROGRAMMING:
* Create the event scheduler
* Turn on tracing
* Create network
* Setup routing
* Insert errors
* Create transport connection
* Create traffic
* Transmit application-level data

E. CHARACTERISTICS OF NS-2:
5 NS-2 implements the following features
1. Router queue Management Techniques Drop Tail, RED, CBQ,
2. Multicasting
3. Simulation of wireless networks
   i. Developed by Sun Microsystems + UC Berkeley (Daedalus Project)
   ii. Terrestrial (Cellular, Ad-hoc, GPRS, WLAN, BLUETOOTH), Satellite
   iii. IEEE 802.11 can be simulated, Mobile-IP, and Adhoc protocols such as DSR, TORA, DSDV and AODV.
4. Traffic Source Behaviour, CBR, VBR
5. Transport Agents- UDP/TCP
6. Routing
7. Packet flow
8. Network Topology

F. Running Simulations using NS-2
To run a network simulation using NS-2 it is imperative to do the following
- Define the Network Topology
  - Define Nodes
  - Define links
  - Set node and link configuration
- Define Protocol/ Application Agents on the various nodes
- Set packet size and transmission intervals
- Define Events
- Capture Trace Variables for analysis
- Visualize using NAM

“V. PROJECT IMPLEMENTATION & RESULTS”
The computation cost of USOR, and compare it with existing schemes. And then describe the implementation and performance evaluation of protocol. USOR requires a signature generation and two point multiplications in the first process. In the route discovery process, each node except the source node and destination node needs one ID-based decryption, while the source node and destination node have to do two ID-based encryption/decryption and two point multiplications. The proposed system ignore symmetric operations as they are negligible compared to PKC operations. MASK is not listed in the table as they do not need public key operations during the route discovery process. However, MASK does not offer sender anonymity or receiver anonymity. It can observe that USOR can achieve unobservability without too much computation cost. System implement both USOR and MASK on NS2, and evaluate their performance by comparing with AODV.

In the simulation, 50 nodes are randomly distributed within a network field of size1500mx300m as such a rectangle field can make the number of hops between two nodes larger. Mobile nodes are moving in the field according to the random waypoint model, and adopt the speed ranges used in so that the average speeds range from 0 to 10m/s. The local session keys are updated every 40 seconds in the simulation, and each update involves a complete anonymous key establishment procedure. To simulate cryptographic operations on each node, force each node to delay for some time according to the benchmarks given. Evaluate the performance of USOR in terms of packet delivery ratio, packet delivery latency, and normalized control bytes. With following results demonstrate the performance of USOR, MASK and AODV at different moving speeds for two different traffic loads. Two traffic loads are selected according to performance of the standard AODV implementation of ns2. AODV has the highest packet delivery ratio for both types of traffic loads; and Mask’s performance is between AODV and USOR. The packet delivery ratio decreases as nodal speed increases and traffic load becomes heavier. Under the light traffic load (2 packets/s), USOR has more than 90% packet delivery ratio at high node speeds, only slightly lower than MASK and AODV. Under the heavy traffic load (4 packets/s), performance of all three protocols has downgraded greatly. The biggest difference between USOR and AODV on packet delivery ratio is less than 10%. Apparently, the performance drop of both protocols when node speed goes up due to more frequent route disruption at higher speeds. Route disruption leads to packet drop and retransmission, and a new route has to be constructed before remaining packets can be sent out.

Finally, compare USOR with MASK in terms of privacy protection. And make use of the information theoretic privacy metric shown in following tables 1, 2, 3, 4, 5, 6, 7, 8, 9 and figure 11, 12, 13, 14, 15, 16, 17, 18 respectively. Then alter the number of eavesdropping nodes in the network and compute the sender anonymity of RREQ packets. The sender anonymity is the obtained by calculating entropy of probability distribution of possible sender of RREQ packets. USOR provides best privacy protection regardless of the number of eavesdroppers, while MASK provides better privacy for less eavesdropping nodes. However, when the number of eavesdropper increases to 8 or larger, the privacy entropy does not decrease significantly. This is reasonable since the anonymity set of possible senders cannot be reduced any more by introducing more eavesdroppers.

A. Simulation Scenario

<table>
<thead>
<tr>
<th>SIMULATION</th>
<th>PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Nodes</td>
<td>50</td>
</tr>
<tr>
<td>Traffic rate (packets per second)</td>
<td>2,4</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Area (m x m)</td>
<td>1500 x 300</td>
</tr>
<tr>
<td>Simulation time (seconds)</td>
<td>600</td>
</tr>
<tr>
<td>Type of traffic</td>
<td>Constant Bit Rate (CBR)</td>
</tr>
<tr>
<td>Number of connections</td>
<td>20</td>
</tr>
<tr>
<td>Speed (meters per seconds)</td>
<td>2,4,6,8,10</td>
</tr>
<tr>
<td>Speed</td>
<td>varying 2 m/s to 10 m/s</td>
</tr>
<tr>
<td>Pause Time</td>
<td>0s</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>600s</td>
</tr>
<tr>
<td>Packet size</td>
<td>512 bytes</td>
</tr>
</tbody>
</table>

Table 1: Simulation Scenario

B. Steps for Implementing the Dissertation in NS2:

- Step 1: Creating scene files
- Step 2: Creating Traffic file
- Step 3: Writing TCL script and running the Tcl script for the created scene and traffic file
- Step 4: Write AWK file and run it for the .tr file obtained in the previous step to extract the Results
- Step 5: Executing Nam &Trace files

C. Network Animator (NAM) Steps:

- Step 1: Run droptail.tcl file
- Step 2: This will generate xxxx.tr and xxxx.nam files
- Step3: Through the terminal go to the directory where tr and NAM files are generated
- Step4: type namxxxx.nam file
- Step5: Network Animator will open, click the play button and vary the speed depending upon the simulation
A packet consists of two kinds of data: control information and user data (also known as payload). The control information provides data the network needs to deliver the user data, for example: source and destination network addresses, error detection codes, and sequencing information. Typically, control information is found in packet headers and trailers, with payload data in between. The above graph shows the comparison between the AODV, MASK and USOR protocols. The graph is plotted with Variation of Control packets send as a function of Node Speed (m/s). It is observed that Control packets sent for MASK is more than AODV and USOR protocols. This is because of the computational cost for key establishment is more in MASK compared to USOR and AODV.
D. Average End-to-End Delay

In computer networks, End-to-End Delay is the average time taken by a data packet to arrive in the destination. It also includes the delay caused by route discovery process and the queue in data packet transmission. Only the data packets that successfully delivered to destinations that counted. In the above graph the Average End to End delay is more in USOR compare to AODV and MASK. This is because of generation of the group signature and ID based signature.

![Figure 13. Packet Delivery Ratio with Variable Node Speed](image)

E. Packet Delivery Ratio (PDR)

In computer networks, PDR is the ratio of the number of delivered data packet to the destination. This illustrates the level of delivered data to the destination.
PDR = \( \sum \frac{\text{Number of packet received}}{\sum \text{Number of packet send}} \)

AODV has the highest packet delivery ratio for both types of traffic loads, and Mask’s performance is between AODV and USOR. The packet delivery ratio decreases as nodal speed increases and traffic load becomes heavier. Under the light traffic load (2 packets/s), USOR has more than 90% packet delivery ratio at high node speeds, only slightly lower than MASK and AODV.

![Figure 14. Control Packets with Variable Node Speed](image1)

<table>
<thead>
<tr>
<th>NodeSpeed m/s</th>
<th>AODV</th>
<th>MASK</th>
<th>USOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>105407</td>
<td>44946</td>
<td>95185</td>
</tr>
<tr>
<td>4</td>
<td>69748</td>
<td>47123</td>
<td>70231</td>
</tr>
<tr>
<td>6</td>
<td>122223</td>
<td>54344</td>
<td>114645</td>
</tr>
<tr>
<td>8</td>
<td>105458</td>
<td>51115</td>
<td>100609</td>
</tr>
<tr>
<td>10</td>
<td>120760</td>
<td>54430</td>
<td>127081</td>
</tr>
</tbody>
</table>

*Table 5. Control Packets Sent for AODV, MASK and USOR (4 packets/second)*

F. Latency with Node Speed

A packet consists of two kinds of data: control information and user data (also known as payload). The control information provides data the network needs to deliver the user data, for example: source and destination network addresses, error detection codes, and sequencing information. Typically, control information is found in packet headers and trailers, with payload data in between. In the above graph the control packets sent for AODV and USOR are more compare to MASK. This is because of there is no need to establish anonymous key establishment again in MASK.

![Rate 4 pkt/sec](image2)
In computer networks, End-to-End Delay is the average time taken by a data packet to arrive in the destination. It also includes the delay caused by route discovery process and the queue in data packet transmission. Only the data packets that successfully delivered to destinations that counted. In the above graph the Average End to End delay is reduced in USOR compare to Rate 2 packets/second. This is because of generation of the group signature and ID based signature is done at 2 packets/second no need to generate again at 4 packets/second.

Under the heavy traffic load (4 packets/s), performance of all three protocols has downgraded greatly. The biggest difference between USOR and AODV on packet delivery ratio is less than 10%. Apparently, the performance drop of both protocols when node speed
goes up due to more frequent route disruption at higher speeds. Route disruption leads to packet drop and retransmission, and a new route has to be constructed before remaining packets can be sent out. So USOR is better protocol in the performance and Security point of view compare to other two protocols MASK and AODV.

G. Results with Flood Strom Attack

![Figure 17. Packet Delivery Ratio with Variable Node Speed](image)

<table>
<thead>
<tr>
<th>Node Speed (m/s)</th>
<th>FLOOD 2 packets/second</th>
<th>FLOOD 4 packets/second</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>8.37209</td>
<td>7.78302</td>
</tr>
<tr>
<td>6</td>
<td>13.9151</td>
<td>14.3188</td>
</tr>
<tr>
<td>8</td>
<td>9.15493</td>
<td>8.79905</td>
</tr>
<tr>
<td>10</td>
<td>29.7105</td>
<td>27.5904</td>
</tr>
</tbody>
</table>

![Table 8. Packet Delivery Ratio for FLOOD Strom Attack](image)

![Figure 18. Latency with Variable Node Speed](image)

<table>
<thead>
<tr>
<th>Node Speed (m/s)</th>
<th>FLOOD 2 packets/second</th>
<th>FLOOD 4 packets/second</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7.89345</td>
<td>5.59582</td>
</tr>
</tbody>
</table>
Table 9. End to End Delay for FLOOD Storm Attack

<table>
<thead>
<tr>
<th>Packet Size</th>
<th>End to End Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7.96249</td>
</tr>
<tr>
<td>6</td>
<td>6.20119</td>
</tr>
<tr>
<td>8</td>
<td>8.2551</td>
</tr>
<tr>
<td>10</td>
<td>2.3145</td>
</tr>
</tbody>
</table>

Under the Flood Storm Attack, the performance of the system degraded greatly. The above graphs shown the Packet Delivery Ratio of the system is very low and Average End to End delay is very high. So to avoid such Flood Strom Attacks the security must be needed so that the USOR protocol is proposed.

**H. Overview of USOR:**

The proposed protocol is an efficient privacy-preserving routing protocol USOR that achieves content unobservability by employing anonymous key establishment based on group signature.

The unobservable routing protocol is then executed in two phases.

- First, an anonymous key establishment process is performed to construct secret session keys.
- Then an unobservable route discovery process is executed to find a route to the destination.
- The security analysis demonstrates that USOR not only provides strong privacy protection, it is also more resistant against attacks due to node compromise.
- The design of USOR offers strong privacy protection complete unlinkability and content unobservability for Ad hoc networks.
- USOR provides enhanced privacy protection for mobile ad-hoc networks. While performing the operation it takes less time compared to AODV protocol.

**VI. CONCLUSION AND FUTURE SCOPE**

In this dissertation, the proposed system is an unobservable routing protocol USOR based on group signature and ID-based crypto system for ad hoc networks. The design of USOR offers strong privacy protection complete unlinkability and content unobservability for ad hoc networks. The security analysis demonstrates that USOR not only provides strong privacy protection, it is also more resistant against attacks due to node compromise. USOR is implemented on NS2 and examined performance of USOR, which shows that USOR has satisfactory performance in terms of packet delivery ratio, latency and normalized control bytes.

Future work along this direction is to study how to defend against wormhole attacks, which cannot be prevented with USOR. Also how to make the unobservable routing scheme resistant against DoS attacks is a challenging task that demands sin-depth investigation.

**REFERENCES**


