Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 6 – No.4, December 2016 – www.caeaccess.org

A Highly Dynamic Locality Aware Secure Data Availability Routing Framework for Ad Hoc Clouds

Niroj Kumar Pani
Department of
Computer Science Engineering
& Applications
IGIT, Sarang, India

Bikram Keshari Ratha
Department of
Computer Science
Utkal University
Bhubaneswar, India

Sarojananda Mishra
Department of
Computer Science Engineering
& Applications
IGIT, Sarang, India

ABSTRACT
Ad hoc clouds in an integration of mobile ad hoc network over the cloud computing environment. In ad hoc clouds since the infrastructure setup is implemented over existing sporadically available, non-exclusive (primarily used for some other purpose) mobile devices they offer an attractive alternative to the existing data center cloud model in terms of cost and computational overhead. In this paper, we propose a secure routing framework for ad hoc cloud environments that optimizes itself depending upon the locality of nodes with respect to the cloudlets that form the ad hoc cloud. The scheme not only establishes a secure and reliable route between two nodes but also supports secure data packet forwarding. We specify the design of the protocol and examines its capacity to withstand possible security threats.

Keywords
Cloud computing; Ad hoc clouds; Security attacks; Secure routing.

1. INTRODUCTION
The use of cloud computing permits to overcome the lack of local resources. However, the services provided by the traditional models in which the cloud setup is built/run by a dedicated homogeneous collection of machines completely depends on the connection to the remote cloud. Such models are called the data center cloud models and may fail in low connectivity scenario. The main solution proposed for it is to use the surrounding mobile devices (mobile phones, PDAs, laptops etc.) as local resource providers and to exploit their capabilities as a mobile cloud. Such cloud models in which the cloud set up is distributed over non-exclusive (primarily used for some other purpose) heterogeneous local mobile devices is referred to as ad hoc clouds [1-3].

The architecture of ad hoc clouds composed of a set of cloudlets, each of which is formed by a dynamically changing set of possibly heterogeneous nodes (mobile devices). Each cloudlet provides a particular service or application. A node available to the ad hoc cloud may host a part or a number of services, each of which accounts for a cloudlet. The software running on a particular node that contributes to a particular service is termed as a cloud element. The cloudlets are dynamic in the sense that their size may be expanded or contracted by altering the number of nodes. The cloud elements comprising a given cloudlet communicate with themselves to coordinate their activity.

Implementation of ad hoc clouds can offer various benefits to individual enterprise [4, 5]. It not only reduces the numbers of machines that need to be procured by the enterprise to provide the cloud infrastructure but also minimizes the need for specialized infrastructure required for the cloud setup. This could yield significant cost saving. Moreover, because of the decrease in the procurement of the total number of dedicated machines, the overall power consumption reduces.

Ad hoc cloud networks host a number of research issues [6-9]. However, the biggest problem lies in its security [10, 11]. Our work is based on secure routing. We propose a secure routing framework called Locality Aware Secure Routing (LASR) that optimizes itself depending upon the locality of nodes with respect to the cloudlets that form the ad hoc cloud. The scheme not only establishes a reliable route between two nodes but also supports secure data packet forwarding.

The rest of the paper is organized as follows. Section 2, presents the possible security threats against routing and the security requirements in an ad hoc cloud environment. The proposed scheme is detailed in Section 3. In Section 4 we present the security analysis of LASR. Section 5 gives its implementation details. In Section 6, we conclude the paper.

2. ROUTING IN AD HOC CLOUDS: SECURITY THREATS AND NEEDS
Routing between nodes (mobile devices) in an ad hoc cloud is comparable to that of a mobile ad hoc network (MANET) and hence accounts for the similar level of security risks. Like MANETs, security exploits against routing in an ad hoc clouds may be passive or active [10, 11] both of which can be further classified as information disclosure, impersonation, modification of packet content, message fabrication and replication depending upon their attacking behavior.

Information disclosure attacks involve a compromised node disclosing confidential information such as the network topology, the location of nodes, the optimized route between two nodes etc. to other malicious nodes in the network. Eavesdropping, traffic analysis and monitoring, and location disclosure attacks come under this class. In the impersonation/spoofing attack, an attacker misrepresents an authentic node by stealing its identity. The intention of the attacker may involve the use the network resources which it may not get under non-adversarial situation or to havoc the normal network functioning by injecting illegal routing messages into the network. There are many ways to launch an impersonation attack, for instance, the attacker may guess the cryptographic details specifying the authenticity of the target node, or the attacker could even hear these details from some earlier communication. Examples of impersonation attack include man-in-the-middle attack and Sybil attack. In modification attack, the attacker doesn't snoop the packet but simply modifies the packet content. The intention might involve misrouting the packet or even fetch the nodes with

wrong information about the network topology, for example advertising a legitimate node as a malicious node. Some examples of modification attacks are blacklist attack, packet redirection by changing the route sequence number or hop count, the formation of routing loops, and the Denial of Service (DoS) attack. The packet redirection, routing loop, and DoS attacks can also be launched by other means without modifying the packet content. Fabrication attacks involve a set of unauthorized nodes generating wrong routing messages, for example, periodic route update packets or messages showing route errors etc. The objective behind the fabrication attack is to consume the resources of the authorized nodes in the network. The resource consumption attack, routing table poisoning, routing table overflow, rushing attack, blackhole attack, and DoS attacks are some examples of fabrication attacks. In the packet replication attack, an attacker or a group of attackers retransmits a packet over themselves or over a group of authorized nodes (without their notice). Examples include the tunneling and the Wormhole attacks.

A robust secure routing framework for an ad hoc environment must be capable in withstanding each one of the above-discussed security breaches/attacks. We present resulting set of security requirements that would need to be addressed while designing a secure routing protocol for ad hoc clouds. First, the routing messages should not expose the network topology to any node in the network whether authorized or unauthorized, as a topology exposure may put a node trying to capture or destroy the routes in an advantageous position. Second, route signaling cannot be spoofed; every node needs to be authenticated. Third, the routing messages should remain unaltered throughout the transit between the nodes in the network except for the way as specified by the protocol. Fourth, injection of fabricated packets into the network should be completely prohibited. Fifth, care should be taken against the nodes with insufficient authentication to exclude them from route computation. Sixth, replication of packets should be forbidden. Seventh, packets cannot be redirected from the discovered shortest path unless that path becomes inactive, and finally, nodes should not be allowed to drop any packet unless it is found to be duplicate.

3. LASR

Locality Aware Secure Routing (LASR) is adaptive to the localization of the target node. We define the locality of a node as follows.

Definition: The locality of a node X in an ad hoc cloud comprises of those set of nodes that provide the same service ‘μ’ as X do (i.e. that are within the cloudlet of X) and are at a hop count of less than or equal to ‘β’ from X. Here, β is a number which is application dependent. The tuple (μ, β) is called the *localization factor* of the node X.

The above definition is explained with the help the following example. Let us consider the ad hoc cloud network given in Figure 1. Here, the nodes S, A, B, C, E, and F provides one service and hence belong to one cloudlet. If we consider β = 2, then the locality of the node S includes the nodes A, B, and C but not E or F. Similarly, the locality of A includes S, C, B, E and F. Notice that the locality of F does not include the node G even though it is within β hop counts from F. It’s because G is not a part of the cloudlet to which F belongs.

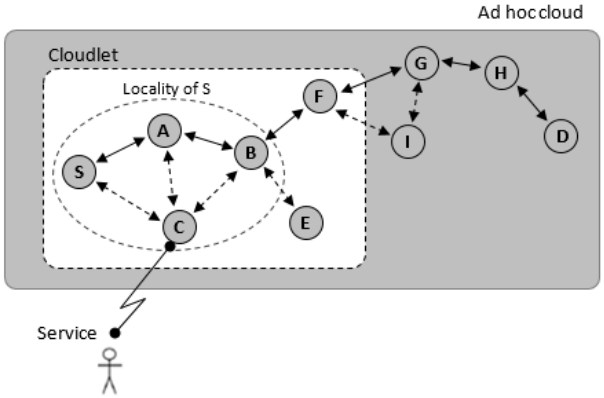

Figure 1. An ad hoc cloud formed by nine nodes.

In an ad hoc cloud, it can be safely considered that most of the communications take place between nodes close to each other within a cloudlet. LASR optimizes itself by deploying different versions of secure routing approaches based on whether a target node is present within the locality of the source or not. The concept is similar to the one presented in [12], however with significant security adaptations. LASR uses the following techniques in order to achieve the desired security requirements as discussed Section 2. (1) None of the packets carry any topology exposing information such as a route record or a hop count, as this information can be easily used by malicious nodes to launch a number of attacks. (2) Packet contents are unalterable. Any exploitable packet content, if present, is encrypted. (3) Hop-to-hop authentication and end-to-end integrity of packets are enforced. This is done by using the digital signature technique [13]. (4) A stringent hop-to-hop verification of packet traversal time is imposed. This is because even with hop-to-hop authentication and an end-to-end integrity check a malicious node or a group of malicious nodes collectively may launch a packet redirection or a routing loop attack in the absence of either a route record or a hop count information in the packet [14].

The process of encryption and digital signature requires private/public key pairs. Each node in LASR must keep two pairs of private/public keys, one pair for signing and verifying and the other for encrypting and decrypting. For a node X the signature and verification keys are respectively X.SKey and X.VKey while, encryption and decryption keys are respectively X.EKey and X.DKey. Among these X.SKey and X.DKey are private keys whereas X.VKey and X.EKey are public keys. LASR assumes the presence of trusted Certification Authorities (CAs) for authentic public key setups. The authors of [94] have discussed how key management could be done in a cloud environment by using public key infrastructure. A node before entering the network must obtain a certificate from a CA near to it. A certificate issued to a node must contain the two pubic keys: the encryption key and the verification key. In addition to this, for accurate verification of packet traversal time, we assume that all the nodes are tightly clock synchronized with a permitted error (clock difference) of Δ. The value of the parameter Δ must be on the order of a few milliseconds, and must be known to all the nodes in the network.

Next, we discuss the two different routing approaches adopted by our proposed scheme LASR depending upon the locality of the target node. We use the ad hoc cloud network shown in Figure 1 for the illustration.

3.1 Intra-Locality Routing

Intra-locality routing is applicable if the target node happens to be within the locality of the source node, for example, in Figure 1, if node S needs a route to node B. Since, within a locality the call to mobility ratio is very high, a proactive routing scheme is well suited for this situation. Therefore, intra-locality routing is implemented in terms of a variation of limited depth proactive link-state routing technique [16, 17] with additional security features. It works as follows.

Step 1: Each node within its locality periodically advertises (broadcasts) a *link-state packet (LSP)*. For example, node S within it locality broadcasts the following LSP.

S→ broadcast: <<LSP, S.IP, $S.T_{D0}$, S.Cert, (μ, β), TTL, NeighbourList[n], LinkMetric[n]>, S.Sign>
Where, S.Sign= [<LSP, S.IP, $S.T_{D0}$, S.Cert, (μ, β), TTL, NeighbourList[n], LinkMetric[n]>]^S.SKey

The packet contains the packet type identifier (LSP), the IP address of the source node (S.IP), the departure time of the LPS from source ($S.T_{D0}$), S's certificate (S.Cert), the localization factor (μ, β), a time-to-live value (TTL) which is used to control the traversal scope of the packet, the list of neighbors of S (NeighbourList[n]), and a set of link metrics (in terms of traversal time) to the neighbors of S (LinkMetric[n]), all appended by the signature S.Sign of S. The signature S.Sign is made on <LSP, S.IP, $S.T_{D0}$, S.Cert, (μ, β), TTL, NeighbourList[n], LinkMetric[n]> by using the signature key S.SKey of S. The tuple (S.IP, $S.T_{D0}$, μ, β) uniquely identifies an LSP. The TTL field is initialized to β times the *average packet traversal time* within the locality of S. Upon receipt the packet, every node checks the value of the TTL field and until the value of $S.T_{D0}$+TTL±Δ is less than the current time, the LSP is rebroadcasted. NeighbourList[n] and the average packet traversal time within the locality can be obtained with the help of suitable MAC layer protocols such as the *neighborhood discovery protocol* (NDP) [18].

Step 2: When the LSP is received by a neighbor of S, the neighbor first examines the packet's authenticity using S.VKey extracted from S.Cert. It then, checks whether the value of $S.T_{D0}$+TTL±Δ is less than the current time. If it is, the node adds the LSP's information to its *link-state table (LST)*, and again forwards it. Otherwise, the LSP is dropped. The LST in LASR has five fields. (1) Src.IP. It is the IP address of LSP's source. (2) $Src.T_{D0}$. It is the departure time of the LSP from the source. (3) $Me.T_{A0}$. It is the arrival time of the LSP at the receiving node. (4) NeighbourList. It is the list of neighbors of the source node corresponding to the field NeighbourList[n] in the LSP. (5) LinkMetric. It is the set of metrics that corresponds to the field LinkMetric[n] in the LSP. Since all the nodes within the locality of S receive the same LSPs, each of them builds/maintains the same link-state table.

Step 3: After the LST is built, each node applies the Dijkstra algorithm [16] to its LST in order to compute the optimal route to every other node present within its locality. This information is stored in its *routing tables (RTs)*. For example in Figure 1 node S computes the route to nodes A, B, and C and stores this information in its RT. The RT has three fields. (1) Dst.IP. It is the IP address of the destination node. (2) NHop.IP. It is the IP address of the next hop through which to reach the destination. (3) $Me\sim Dst.T_T$. It is the traversal time of the LSP from the concerned node to the destination.

Step 4: Once the routing table is built, a node, in order to facilitate secure data packet delivery, requests a session key from its desired destination by sending it a *session key request packet* (SKREQ) along the route specified by its routing table before the data transfer phase. For example, node S sends the following SKREQ to node B through A before transferring any data packet to B.

S→B: <<SKREQ, S.IP, B.IP, S.Cert>, S.Sign>
Where, S.Sign = [<SKREQ, S.IP, B.IP, S.Cert>]^S.SKey

The packet contains a packet type identifier (SKREQ), the IP address of the source (S.IP), the IP address of the destination (B.IP), and S's certificate (S.Cert), all appended by the signature S.Sign made on <SKREQ, S.IP, B.IP, S.Cert> by using S.SKey of S.

Step 5: Node B on receiving this SKREQ first verifies the signature of S using S.VKey, which it extracts from S.Cert. It then creates a session key S-B.SesKey, encrypts it using S.EKey which is present in S.Cert and sends the encrypted session key to S as a session key request packet (SKREP) along the reverse route.

B→S: <<SKREP, B.IP, S.IP, B.Cert, {S-B.SesKey}^S.EKey>, B.Sign>
Where, B.Sign = [<SKREP, B.IP, S.IP, B.Cert, {S-B.SesKey}^S.EKey>]^B.SKey

The packet contains a packet type identifier (SKREP), the IP addresses of B and S (B.IP and S.IP), the certificate of B and the encrypted session key ({S-B.SesKey}^S.EKey), all appended by the signature B.Sign made on [<SKREP, B.IP, S.IP, B.Cert, {S-B.SesKey}^S.EKey> by using B.SKey of B.

Step 6: Node S on receipt of the SKREP verifies the authenticity of the packet using B.VKey and decrypts the session key using B.DKey extracted from B.Cert. Once S gets the session key S-B.SesKey, it can encrypt the data packet using S-B.SesKey and send it to B along the same route as specified in its routing table. All further communication between S and B takes place similarly, using this session key.

3.2 Inter-Locality Routing

This approach of routing is initiated by a node (the source node) when it searches the path to another node (the destination node) in its routing table (RT). However, the RT doesn't contain the path to the destination (because the destination is not within the locality of the source), for example, in Figure 1, when S wants to send a packet to D. For routing to a target situated outside the locality of the source where the call to mobility ratio is very low, a reactive (on-demand) routing approach [19, 20] is well suited. The implementation of routing outside the locality is done on the basis of our work in [21] with minor modifications to the packet content. It works as follows.

Step 1: The source node S initiates the route request phase by broadcasting a route request packet (RREQ).

S→Broadcast: <<REQ, S.IP, D.IP, $S.T_{D1}$, (μ, β), S.Cert>, S.Sign>
Where, S.Sign = [<REQ, S.IP, D.IP, $S.T_{D1}$, (μ, β), S.Cert>]^S.SKey

The RREQ contains seven fields. REQ is the route request packet type identifier, S.IP and D.IP are respectively the IP addresses of the source and destination, $S.T_{D1}$ is the departure time of the RREQ from S, (μ, β) is the localization factor, S.Cert is the certificate of S, and S.Sign is the signature of S made on <REQ, S.IP, D.IP, $S.T_{D1}$, S.Cert> by using its signature key S.SKey. This signature can only be verified by

using verification key S.VKey of S present in its certificate S.Cert. The localization factor (μ, β) is included in the RREQ to let the destination D ensure about the type of service provided by the source S. The tuple <S.IP, D.IP, $S.T_{D1}$> uniquely identifies an RREQ which prevents the replay attack.

Before broadcasting the RREQ, S keeps this packet's information in a data structure called *Route Discovery Table (RDT)*. The RDT is maintained by every node in the ad hoc cloud and is used to store information on the route request packets (RREQs) and the route reply packets (RREPs) processed by the nodes during the route request and route reply phases respectively. A row in the RDT of a node corresponds to one RREQ and its respective RREP processed by the node. The RDT is made of seven fields: (1) Src.IP. It is the IP address of the source from which the RREQ is originated. (2) Dst.IP. It is the IP address of the final destination of the RREQ. (3) $Src.T_{D1}$. It is the departure time of the RREQ from the source. It is set by the source node and is present in the RREQ itself. (4) $Me.T_{A1}$. It is the arrival time of the RREQ at the concerned node. (5) $Me.T_{D1}$. It is the departure time of the RREQ from the concerned node. (6) PHop.IP. It is the IP address of the hop from which the RREQ is received. (7) NHop.IP. It is the IP address of the hop from which the respective RREP is received. In our case, the source S inserts the following row in its RDT.

Src.IP = S.IP, Dst.IP = D.IP, $Src.T_{D1} = S.T_{D1}$, $Me.T_{A1}$ = NULL, $Me.T_{D1} = S.T_{D1}$, PHop.IP = NULL, NHop.IP = NULL

The source S also sets a timer before broadcasting the RREQ. If S doesn't receive a route reply packet (RREP) from the destination D before the timer expires, it may reinitiate a route request phase. A route request phase is also be reinitiated by the source if it receives an RREP and finds that the security of the RREP has been compromised (the RREP has reached S through a malicious path). The source can make up to η route request attempts. The value of η is application dependent (normally based on the round-trip-time of the packets).

Step 2: Every intermediate node upon receiving the RREQ validates the previous node's signature and examines the uniqueness of the RREQ. If the packet is found to be authentic and unique the intermediate node removes the previous node's signature from the packet if the previous node is not the source, then signs the RREQ, appends its own certificate to the packet, inserts a new row in its RDT for this RREQ, and rebroadcasts it. For example, node A broadcasts the following RREQ after verifying the authenticity and uniqueness of the same received from S.

A→Broadcast: <<<REQ, S.IP, D.IP, $S.T_{D1}$, (μ, β), S.Cert>, S.Sign>, A.Sign, A.Cert>
Where, A.Sign = [<<REQ, S.IP, D.IP, $S.T_{D1}$, (μ, β), S.Cert>, S.Sign>]^A.SKey

And node B broadcasts the following RREQ after validating the authenticity and uniqueness of the same received from A.

B→Broadcast: <<<REQ, S.IP, D.IP, $S.T_{D1}$, (μ, β), S.Cert>, S.Sign>, B.Sign, B.Cert>
Where, B.Sign = [<<REQ, S.IP, D.IP, $S.T_{D1}$, (μ, β), S.Cert>, S.Sign>]^B.SKey

Eventually, the RREQ reaches the destination D through the path S-A-B-F-G-H-D.

Step 3: The destination D on receipt of the RREQ verifies the signatures of the previous node as well as the source S to determine the authenticity of the packet. If either of the signatures is found invalid, the RREQ is dropped. On the other hand, if both the signatures are valid, destination D compares the tuple (S.IP, D.IP, $S.T_{D1}$) of the RREQ with its RDT's tuple (Src.IP, Dst.IP, $Src.T_{D1}$) to determine whether the received RREQ is the first route request packet for this route request attempt (recall that, a source can make up to η route request attempts). There may be three cases.

*Case 1***:** The tuple (S.IP, D.IP, $S.T_{D1}$) completely matches the tuple (Src.IP, Dst.IP, $Src.T_{D1}$). It means that the current RREQ is a duplicate of an already processed RREQ. So, in this case, the current RREQ is rejected.

*Case 2***:** Only the tuple (S.IP, D.IP) matches with the tuple (Src.IP, Dst.IP). However, $S.T_{D1}$ does not match. It means that the current RREQ is the first route request packet of another (2nd and onwards) route request attempt made by the source. An earlier attempt has already been made by the source for which an RREP has been sent by the destination D. However, the RREP has been rejected by the source because the packet has reached the source S through a malicious route. In such case, it becomes necessary for the destination D to determine whether the current RREQ has reached D through the same malicious route through which the earlier RREP (the rejected one) has reached the source S. To know this, D takes the following steps.

First, it calculates the current RREQ's traversal time ($S{\sim}D.T_{T1}$).

$S{\sim}D.T_{T1}$ = Current RREQ's T_{A1} - Current RREQ's $S.T_{D1}$

Then, it determines the traversal time of that RREQ ($S{\sim}D.T_{T1*}$) for which the tuple (Src.IP, Dst.IP) of an existing row in the RDT matches the tuple (S.IP, D.IP) of the current RREQ.

$S{\sim}D.T_{T1*}$ = Matched row's $Me.T_{A1}$ - Matched row's $Src.T_{D1}$

If $S{\sim}D.T_{T1}-\Delta \leq S{\sim}D.T_{T1*} \leq S{\sim}D.T_{T1}+\Delta$, it means that the current RREQ has reached the destination D through the same malicious path through which the earlier RREP (the rejected one) has reached the source S. Hence, the current RREQ is rejected. Instead, D accepts the RREQ, deletes the existing row in the RDT whose (Src.IP, Dst.IP) matches the tuple (S.IP, D.IP) of the current RREQ, inserts a new row in the RDT for the current RREQ and initiates a route reply phase.

Case 3: The tuple (S.IP, D.IP, $S.T_{D1}$) doesn't match with the tuple (Src.IP, Dst.IP, $Src.T_{D1}$) for any row in the RDT. It means that the received RREQ is the first route request packet of the first route request attempt made by S. So, D accepts this RREQ, inserts a new row in its RDT for this RREQ and initiates a route reply phase.

Step 4: The destination D initiates the route reply phase by creating a route reply packet (RREP) containing the encrypted session key S-D.SesKey, and unicasting it back to the source S along the reverse path i.e. D-H-G-F-B-A-S.

D→H: <<REP, D.IP, S.IP, $S.T_{D1}$, $D.T_{A1}$, $D.T_{D2}$, {S-D.SesKey}^S.EKey, (μ, β), D.Cert>, D.Sign>
Where, D.Sign = [<REP, D.IP, S.IP, $S.T_{D1}$, $D.T_{A1}$, $D.T_{D2}$, {S-D.SesKey}^S.EKey, (μ, β), D.Cert>]^D.SKey

The RREP contains ten fields. REP is the route reply packet type identifier, D.IP and S.IP are respectively the IP addresses

of the destination and source nodes, $S.T_{D1}$ is the departure time of the RREQ from the source S against which this RREP is generated, $D.T_{A1}$ is the arrival time of the RREQ at D against which this RREP is generated, $D.T_{D2}$ is the departure time of the RREP from D, {S-D.SesKey}^S.EKey is the encrypted session key, (μ, β) is the localization factor, D.Cert is the certificate of D, and D.Sign is the signature of D made on <REP, D.IP, S.IP, $S.T_{D1}$, $D.T_{A1}$, $D.T_{D2}$, {S-D.SesKey}^S.EKey, D.Cert> by using its signature key D.SKey. This signature can only be verified by using D.VKey which is present in D.Cert

Step 5: An intermediate node, say X, in the reverse path on receiving the RREP, verifies the signature of the node from which the RREP is received. If it is found invalid the packet is rejected. Instead, X compares the RREP's traversal time from the destination to itself ($D\sim X.T_{T2}$) with the traversal time of the RREQ from itself to the destination ($X\sim D.T_{T1}$) against which this RREP is generated.

$$D\sim X.T_{T2} = X.T_{A2} - D.T_{D2}$$
$$X\sim D.T_{T1} = D.T_{A1} - X.T_{D1}$$

$X.T_{A2}$ is the time at which the RREP reached X, $D.T_{D2}$ and $D.T_{A1}$ are present in the RREP itself, whereas $X.T_{D1}$ can be obtained from X's RDT by matching the RREP's tuple (D.IP, S.IP, $S.T_{D1}$) against the RDT's tuple (Dst.IP, Src.IP, $Src.T_{D1}$). The $Me.T_{D1}$ field of the matching row gives the value. If there is a difference between $D\sim X.T_{T2}$ and $X\sim D.T_{T1}$ is found to be more than Δ it means that, either a packet redirection attack or a routing loop attack has occurred on the RREP. Hence, the RREP is rejected. This approach has the advantage that the other nodes along the reverse path from the current node up to and including the source node don't have to unnecessarily process a compromised RREP. On the other hand, if $D\sim X.T_{T2}$ and $X\sim D.T_{T1}$ have a difference of less than or equal to Δ, the intermediate node considers the RREP as a valid one. In this case, the intermediate node removes the signature of the node from which the RREP is received if it is not the destination D, then signs the RREP, appends its own certificate, and unicasts the RREP to the next node in the reverse path. For example, node H unicasts the following RREP to G.

H→G: <<<REP, D.IP, S.IP, $S.T_{D1}$, $D.T_{A1}$, $D.T_{D2}$, {S-D.SesKey}^S.EKey, (μ, β), D.Cert>, D.Sign> H.Sign, H.Cert>

Where, H.Sign = [<<REP, D.IP, S.IP, $S.T_{D1}$, $D.T_{A1}$, $D.T_{D2}$, {S-D.SesKey}^S.EKey, (μ, β), D.Cert>, D.Sign>]^H.SKey

And, node G unicasts the following RREP to F.

G→F: <<<REP, D.IP, S.IP, $S.T_{D1}$, $D.T_{A1}$, $D.T_{D2}$, {S-D.SesKey}^S.EKey, (μ, β), D.Cert>, D.Sign> G.Sign, G.Cert>

Where, G.Sign = [<<REP, D.IP, S.IP, $S.T_{D1}$, $D.T_{A1}$, $D.T_{D2}$, {S-D.SesKey}^S.EKey, (μ, β), D.Cert>, D.Sign>]^G.SKey

Ultimately, the RREP reaches the source S through the path D-H-G-F-B-A-S.

Step 6: The source S, on receiving the RREP, verifies the signatures of the node from which it receives the RREP as well as of the destination D. S, then compares the traversal time of the RREP from destination to itself ($D\sim S.T_{T2}$) with that of the corresponding RREQ from itself to the destination ($S\sim D.T_{T1}$) in the similar manner as the intermediate nodes do.

$$D\sim S.T_{T2} = S.T_{A2} - D.T_{D2}$$
$$S\sim D.T_{T1} = D.T_{A1} - S.T_{D1}$$

If they have a difference of more than Δ, the RREP is rejected and a new route request phase is initiated. Otherwise, S accepts the RREP and updates its routing table (Dst.IP = D.IP, NHop.IP = A.IP, $Me\sim Dst.T_{T1} = S\sim D.T_{T1}$). Thereafter S extracts the session key S-D.SesKey by decrypting it with S.DKey. The session key can now be used to encrypt the data packets thereby ensuring secure data packet delivery.

4. SECURITY ANALYSIS

In this section, we analyze the strength of LASR in resisting multiple possible security threats presented in Section 2.

4.1 Information Disclosure

None of the packets (LSP, RREQ, or RREP) in LASR contain a route record or a hop count information. This prevents malicious nodes from getting/deriving any kind of information regarding the network topology. In the case of inter-locality routing, a node can't deduce the network topology except for knowing about its neighbors. So far intra-locality routing is concerned, the nodes determine the topology indirectly. However, the topology is restricted to the locality only. This is safe because the nodes accept only those packets that have a verified sender's signature.

Further, the session key present in the RREP, which is the only packet content that might be considered exploitable, is encrypted by using highly secure asymmetric key cryptography. Therefore, it cannot be snooped.

4.2 Impersonation

In LASR only those packets are accepted by the nodes that are signed with a certified public key. In intra-locality routing, the LSPs are signed by the source. Inter-locality routing follows both hop-to-hop and end-to-end authentication during route request as well as route reply phases. Since the RREQs and the RREPs include the certificates and signatures of the source and the destination respectively, it ensures that only the source can generate the RREQ and the destination the RREP. This prevents breaches where the source, the destination or an intermediate node may be impersonated, for example, man-in-the-middle attack and Sybil attack.

4.3 Modification

Since all packets are signed by the initiating nodes (LSPs and RREQs are signed by the source whereas RREPs are signed by the destination), any alterations in the route between the source and the destination are immediately detected and the altered packets are subsequently discarded. This prevents modification attacks such as the detour or the black list attack.

4.4 Fabrication

Fabrication attacks such as such as resource consumption, routing table/route cache poisoning, routing table overflow, rushing attack or the blackhole attack are prevented, because in the proposed protocol no intermediate node is allowed to generate a routing packet either in the case of localized routing or in the case of routing outside a locality.

4.5 Replication

In the case of intra-locality routing, every node checks the TTL field. Since an LSP is not allowed to traverse beyond its TTL time the packet can't be replayed. In the case of routing outside the locality, every node checks the tuple <S.IP, D.IP, $S.T_{D1}$> for the uniqueness of the RREQ. This completely prevents the packet replication attacks such as tunneling and the Wormhole attacks.

4.6 Packet Redirection and Routing Loops

In LASR, the deployment of the stringent hop-to-hop verification of the packet traversal time prevents packet redirection and a routing loop attacks from being launched. This approach also eliminates the possibility of Denial of Service (DoS) attack that may take place as a consequence of the packet redirection or routing loop attacks. We prove our claim with the help of the ad hoc cloud network given in Figure 1. Let us consider that δX is the packet (RREQ/RREP) processing time of any arbitrary node X in the network and $\Omega_{X\text{-}Y}$ is the packet traversal time between two arbitrary nodes X and Y (from X to Y and from Y to X) in the network. We have considered the processing time for both RREQ and RREP to be same because the actions performed by a node upon receipt of either of these packets is identical. Moreover, since the links are symmetric, the packet traversal times between two nodes in any direction are always same.

4.6.1 Packet Redirection

Let us suppose the intermediate node G is malicious. The attacking behavior exhibited by G is as follows: during the route reply phase, when H unicasts the RREP through the reverse path D-H-G-F-B-A-S, node G launches a packet redirection attack by unicasting the RREP to I instead of unicasting it to F. The RREP reaches F via I, not directly from G (where I is unknown about this behavior of G).

In LASR, this attacking behavior of G is immediately caught by F when it receives the packet from I, because when F compares the traversal times of this RREP with that of the respective RREQ, it finds a difference of more than the permitted error Δ. The proof is given below.

The traversal time of the RREQ from F to D ($F\sim D.T_{T1}$) = $\Omega_{F\text{-}G} + \delta G + \Omega_{G\text{-}H} + \delta H + \Omega_{H\text{-}D}$ **(1)**

The traversal time of the corresponding RREP from D to F ($D\sim F.T_{T2}$) through node I = $\Omega_{H\text{-}D} + \delta H + \Omega_{G\text{-}H} + \delta G + \Omega_{I\text{-}G} + \delta I + \Omega_{F\text{-}I} \pm \Delta$ **(2)**

$\Rightarrow D\sim F.T_{T2} - F\sim D.T_{T1} = (\Omega_{I\text{-}G} + \delta I + \Omega_{F\text{-}I} - \Omega_{F\text{-}G} \pm \Delta)$ **(3)**

In equation (3), the value of $\Omega_{I\text{-}G} + \delta I + \Omega_{F\text{-}I}$ is always greater than $\Omega_{F\text{-}G} \pm \Delta$, by a value larger than Δ because, if it is not the case, then the RREQ forwarded by node F would not have reached node G directly, earlier than the same copy of the RREQ that reached G via node I.

$\Rightarrow D\sim F.T_{T2} - F\sim D.T_{T1} > \Delta$ ■

4.6.2 Routing Loops

This time let us consider the nodes F and G to be malicious. They launch a routing loop attack in the route reply phase as follows. When F receives the RREP from G, it sends the packet back to G, which again forwards it to F. Node F may continue this looping or deliver the RREP to B.

This type of attack is easily detected in LASR, because when B receives the RREP from F (after few loops) and compares the traversal times of this RREP with that of the respective RREQ, it finds a difference of more than the allowed error Δ.

The traversal time of the RREQ from B to D ($B\sim D.T_{T1}$) = $\Omega_{B\text{-}F} + \delta F + \Omega_{F\text{-}G} + \delta G + \Omega_{G\text{-}H} + \delta H + \Omega_{H\text{-}D}$ **(1)**

Time taken by each loop between F and F = $2*\Omega_{F\text{-}G}$ **(2)**

$\Rightarrow$ The traversal time of the corresponding RREP from D to B ($D\sim B.T_{T2}$) via 'n' loops formed between F and G = $\Omega_{H\text{-}D} + \delta H + \Omega_{G\text{-}H} + \delta G + \Omega_{F\text{-}G} + \delta F + 2n*\Omega_{F\text{-}G} + \Omega_{B\text{-}F} \pm \Delta$ **(3)**

$\Rightarrow D\sim B.T_{T2} - B\sim D.T_{T1} = (2n*\Omega_{F\text{-}G} \pm \Delta) > \Delta$ ■

4.7 Selective Packet Drops

LASR successfully prevents selective packet drops because when an RREP is dropped by some intermediate node, the source doesn't receive an RREP within the timer expires. Hence, it reinitiates the route request phase. But this time, the destination doesn't accept the RREQ which has traversed through the same malicious path that the earlier RREQ does. So, automatically the malicious node is excluded.

5. IMPLEMENTATION

The implementation of our proposed scheme in an ad hoc cloud environment primarily depends upon two factors. (1) The deployment of the key management approach that we have adopted. The authors of [15] have discussed how key setup could be done in a cloud environment with the help of public key infrastructure. (2) The implementation of tight clock synchronization among the nodes in an ad hoc cloud. This kind of time synchronization can be achievable with the use of hardware based on LORAN-C, WWVB, or GPS [22].

We carried out a primary simulation using the NS-2 [23]. A 16 byte RSA digital signature with 512 bit key was used for the digital signature. We took 50 mobile nodes and 5-10 numbers of attackers which drop the packets forwarded to them in the route reply phase. We calculated the packet delivery ratio (ratio of packets received by the destination to the packets generated by the source) as the performance metric. Results indicate that the packet delivery ratio of LASR is always greater than its non-secure counterpart (the protocol used without the digital signature).

6. CONCLUSION

In this paper, we presented a new secure routing framework for cloud networks where the cloud setup is formed on an ad hoc basis. In devising the scheme, we used selective cryptographic majors to its functionality to create an effective and practical protocol that can withstand multiple possible attacks against ad hoc cloud routing. Together with the implementations for securing the lower layer in the network protocol stack, our proposal provides a foundation for secure communication in ad hoc clouds.

7. REFERENCES

[1] G. Kirby, A. Dearle, A. Macdonald, and A. Fernandes, "An approach to ad hoc cloud computing," arXiv preprint arXiv: 1002.4738, 2010.

[2] S. K. Pippal, S. Mishra, and D. S. Kushwaha, "Architectural Design and Issues for Ad-Hoc Clouds," Advances in Communication, Network, and Computing, pp.291-296, 2012.

[3] A. R. Mohammad, A. S. Elham, and Y. Jararweh, "AMCC: Ad-hoc based Mobile Cloud Computing Modeling," Procedia Computer Science, vol. 56, pp.580-585, 2015.

[4] P. Gupta, A. Seetharaman, and J. R. Raj, "The usage and adoption of cloud computing by small and medium businesses," International Journal of Information Management, vol. 33, no. 5, pp. 861–874, 2013.

[5] H. Wang, W. He, and F.-K. Wang, "Enterprise cloud service architectures," Information Technology and Management, vol. 13, no. 4, pp. 445–454, 2012.

[6] G. Kirby, A. Dearle, A. Macdonald, and A. Fernandes, "An approach to ad hoc cloud computing," arXiv preprint arXiv: 1002.4738, 2010.

[7] S. K. Pippal, S. Mishra, and D. S. Kushwaha, "Architectural Design and Issues for Ad-Hoc Clouds," Advances in Communication, Network, and Computing, pp.291-296, 2012

[8] A. R. Mohammad, A. S. Elham, and Y. Jararweh, "AMCC: Ad-hoc based Mobile Cloud Computing Modeling," Procedia Computer Science, vol. 56, pp.580-585, 2015.

[9] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," Future Generation Computer Systems, vol. 29, no. 1, pp.84-106, 2013.

[10] S. Srinivasamurty and D. Q. Liu, "Survey on cloud computing security," In Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science. IEEE, pp. 1–8, 2010.

[11] M. Almorsy, J. Grundy, and I. M¨uller, "An analysis of the cloud computing security problem," in Proceedings of the 17th Asia Pacific Software Engineering Conference (APSEC), Cloud Workshop, Sydney, Australia, 2010, pp. 1–7.

[12] Z. J. Haas, M.R. Pearlman, and P. Samar, "The zone routing protocol (ZRP) for ad hoc networks," IETF Internet Draft, 2002.

[13] B. A. Forouzan and D. Mukhopadhyay, "Cryptography and Network Security," Special Indian Edition, McGraw-Hill Education, 2011.

[14] S. Ghazizadeh, O. Ilghami, E. Sirin, and F. Yaman, "Security-aware adaptive dynamic source routing protocol," In Proceedings of 27th Annual IEEE Conference on Local Computer Networks (LCN 2002), IEEE, pp. 751-760, 2002.

[15] M. H. Kharche, and M. D. S. Chouhan, "Building trust in cloud using public key infrastructure," International Journal of Advanced Computer Science and Applications, vol. 3, no. 3, 2012.

[16] Behrouz A. Forouzan, "Data Communications and Networking," 4th Edition, Tata McGraw, 2013.

[17] M. O. Pervaiz, M. Cardei, and J. Wu, "Routing security in ad hoc wireless networks," Network Security, Springer US, pp. 117-142. 2010.

[18] Z. J. Haas, M. R. Pearlman, and P. Samar, "Intrazone routing protocol (iarp)," IETF Internet Draft, 2002.

[19] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks," Computer networks, vol. 56, no. 2, pp. 940-965, 2012.

[20] H. Bakht, "Survey of routing protocols for mobile ad-hoc network," International Journal of Information and Communication Technology Research, vol. 1, no. 6, 2011.

[21] Niroj Kumar Pani, Bikram Keshari Rath, and Sarojananda Mishra, "A Topology-Hiding Secure On-Demand Routing Protocol for Wireless Ad Hoc Network," International Journal of Computer Applications, vol. 144, no. 4, pp. 42-50, 2016.

[22] Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A defence against Wormhole attacks in Wireless Ad Hoc networks," In INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol. 3, pp. 1976-1986, 2003

[23] http://www.isi.edu/nsam/ns