Two-Phase Security Framework for Network Layer Communication in Ad Hoc Clouds

Niroj Kumar Pani, Satyasundara Mahapatra, Rati Ranjan Dash

***Abstract*: An ad hoc cloud computing environment on the infrastructure point of view is formed by consuming resources from existing occasionally available computing setups that are primarily used for some other purposes, for example, personal computers, mobile phones and such similar devices connected to the internet. In this paper, we propose a mechanism to secure the network layer communications particularity routing and data packet forwarding in ad hoc cloud environment. The paper details the proposed scheme and analyses its robustness on the security perspective.**

***Keywords*: Cloud computing, ad hoc cloud, security, routing, forwarding.**

I. INTRODUCTION

According to the National Institute of Standards and Technology, there are two types of cloud architectures: private and public [1]. These are known as the data center cloud models. Here, the infrastructure for the cloud is built over completely dedicated machines specifically designed to provide the required could service. Even though both data center models are widely used by the industry, there may be instances where acquiring specialized resources to build the cloud set up, particularly in remote areas are not possible or cost-effective. In such a situation, an ad hoc cloud network [2, 6] may be considered as a good alternative. An ad hoc cloud network on the infrastructure point of view is formed by consuming resources from existing occasionally available computing setups that are primarily used for some other purposes, for example, personal computers, mobile phones and such similar devices connected to the internet. Ad hoc clouds, unlike the data center models (private/public), don't need the computing resources to be acquired a priori. A device that is a part of the ad hoc cloud may not be completely dedicated to providing the cloud service only, it may have its own task, for example, running some other applications. An ad hoc cloud is formed by the combination of a group of cloudlets, where each cloudlet is a set of possibly non-homogeneous mobile devices (nodes) that may change dynamically. A specific service is provided by each cloudlet that may be accessed by users through Web Services, or any other convenient protocol. Each device that builds the ad hoc cloud may provide one or several services, each of which corresponds to a cloudlet. The software running on a device that gives a specific service is called a cloud element. The cloudlets are dynamic as their size may be altered with a change in the number of nodes. There may be communication among the cloud elements to coordinate different activities related to network management or data transmission.

An enterprise that plans to deploy ad hoc clouds gets the advantage of minimizing the number of specialized device procurements and infrastructure set up that would have been otherwise required to build the cloud server based on the traditional model [7, 8]. This would result in a reduction in cost as well as the overall power consumption. Ad hoc cloud networks are very recent technology and their complex nature presents a variety of research issues [2-5]. The work, in this paper, deals with securing the network layer communication between nodes in the ad hoc cloud namely routing and data packet forwarding. We propose a technique that can establish a reliable route between two arbitrary ad hoc cloud nodes. It also secures data packet forwarding by providing a session key for packet encryption. The rest of the paper is organized as follows. Section 2 discusses the challenges against secure network layer communication in ad hoc clouds. The proposed technique is detailed in Section 3. The security analysis of the proposed technique is presented in Section 4. Section 5 concludes the paper.

II. CHALLENGES

In an ad hoc cloud, routing and data packet forwarding between the nodes is similar to that of a mobile ad hoc network (MANET) and hence the security concerns are also the same. Like MANETs, in an ad hoc cloud, the security exploits against routing or data packet can be passive or active [9, 10]. Passive attacks never hamper normal network functioning. Here, the attackers just snoop the packets communicated within the network with an intention to get valuable (required) information from the packets. Such attacking behavior makes passive attack nearly undetectable. Passive attacks defy the confidentiality security goal. Active attacks disrupt the normal functioning of networks. Therefore, all the security goals namely confidentiality, authenticity, integrity, availability, and non-repudiation may be violated.

Both passive and active attacks can be further classified as spoofing/impersonation, information disclosure, fabrication, modification, and replication depending upon their attacking behavior. In impersonation or spoofing, the attacker steals the identity of an authentic node either by guessing it or by snooping it from some earlier communication.

Revised Manuscript Received on August 09, 2019.

Niroj Kumar Pani, Department of CSEA, IGIT, Saranga, Odisha, India. Email: nirojpani@gmail.com

Satyasundara Mahapatra, Department of CSE, Pranveer Singh Institute of Technology, Kanpur, UP, India. Email: satyasundara123@gmail.com

Rati Ranjan Dash, Department of Mech Engg., CET, Bhubaneswar, Odisha, India. Email: ratiranjan.dash@gmail.com

Later these details are used by the attacker to represent itself as the authentic node. The intention of the attacker might be the injection of illegal/bogus route packets into the network or it might be getting those network resources that it may not get under normal circumstances. Popular examples of spoofing are man-in-the-middle and Sybil attacks. Information disclosure attacks involve a malicious node attempting to snoop confidential information present in the packets. The attacker may use this snooped information by itself to victimize the network or it may share this information with other malicious nodes with a similar intention. Eavesdropping is one of the popular examples of information disclosure attack. In modification attack, the attacker tries to modify the information present in the packets. With such an attacking behavior the attacker might want to give false information to other nodes in the network or even misroute the packets. Examples include the detour and blacklist attacks, packet redirections, routing loops creation, and the DoS attack. The packet redirection, routing loop, and DoS attacks can also be launched by other means without modifying the packet content. Fabrication involves the intentional generation and spreading of false routing messages into the network by a set of malicious nodes. Resource consumption, routing table poisoning and overflow, rushing, blackhole, and the DoS attacks are some popular examples of fabrications. In packet replications, such as the tunneling and the Wormhole attacks there is quick but intentional retransmission of packets by a set of malicious nodes over themselves. In some situations, the attackers may even make the packets to retransmit over some authentic nodes without their notice.

A robust communication technique designed for an ad hoc cloud should be able to protest each of the above-discussed security concerns. In this respect, while designing a mechanism intend to secure the network layer communication in ad hoc clouds, the following set of requirements need to be dealt with: (1) The route packets should not reveal the network topology at any means, because when the topology is exposed it puts attackers intending to use the network routes in a favorable condition; (2) Every node, from a source to a destination, need to be authenticated, as it would stop spoofing of route signaling; (3) The routing messages, in all cases, should remain unchanged throughout the transit except the particular way mentioned the security proposal; (4) Any fabricated packets should be blocked; (5) Nodes that are unauthentic should be excluded from route computation; (6) There should be no replication of packets; (7) Unless a discovered shortest path becomes inactive, packets cannot be redirected from it; (8) Unless a packet is not found to be duplicate, nodes can't drop it.

III. THE PROPOSED SCHEME

The proposed scheme has been designed to offer two-phase security for network layer communication in ad hoc clouds, firstly by establishing secure routing consisting of two sub-phases: secure route discovery and secure route reply, and then by securing data packet forwarding by providing a session key for data packet encryption, thereby ensuring a complete and secure network layer connectivity. The proposal is base on our work done in [11, 12].

The following techniques have been used in order to cope with the desired level of security requirements as discussed in Section II. First, no route packets in our proposed scheme contain any information, such as the number of hops or the routing path traveled, that may by-change leak the network topology. Second, packet content is made to remain unchanged from source to destination. Also, end-to-end integrity and hop-to-hop authentication of all packets are done. For these, digital signature [13, 14] is used. Third, any vulnerable packet content, if present, is encrypted. Finally, a precise verification of packet traversal time is done on a hop-to-hop basis. This is because, in the absence of hop count information or route record in the packet, a routing loop or a packet redirection attack could be launched by a set of malicious nodes even with end-to-end integrity check and hop-to-hop authentication [15].

A. Assumptions

The proposed scheme requires private/public key pairs for digital signature and encryption. Every node in the network needs two key pairs, the first pair is for signing/verifying and the second pair is for encrypting/decrypting. The keys can be easily subscribed through a public-key certificate. One of the methods is presented in [16]. A public-key certificate issued to a node must contain the encryption and the signature verification public keys. We assume that every node in the ad hoc cloud set up has these two public keys obtained through a public-key certificate.

In addition to this, for the verification of exact packet traversal time, we assume that all the nodes present in the network are tightly clock synchronized. This is easily achievable, as all the devices are connected to the internet. The devices can sync their clocks with the internet clock. Despite this, we allow a clock error (time difference) of 'δ'. The value of 'δ' must be of the order of milliseconds and all nodes in the network should be known the value.

B. Data Structures

For the proposed scheme to work, a node in the ad hoc cloud is required to maintain two tables.

The first is the route discovery table (RDT). It stores information corresponding to the secure route discovery packets (SRD). The RDT contains seven fields: (1) Src.IP: IP address of the source that generated the SRD; (2) Dst.IP: IP address of the destination of the SRD; (3) Src.DT0: Time of departure of the SRD from the source; (4) Me.AT0: Time of arrival of the SRD from the current node; (5) Me.DT0: Time of departure of the SRD at the current node; (6) PHop.IP: IP address of the previous-hop from which the SRD is received; (7) NHop.IP: IP address of the hop from which the corresponding SRR is received.

The second is the routing table (RT). It is maintained only by the source node (in addition to the RDT). The RT stores information with respect to the secure route reply packet (SRR) after the completion of the secure route reply phase.

The RT contains three fields: (1) Dst.IP: Destination's IP address from which the SRR is received; (2) NHop.IP: IP address of the next-hop through which to reach the destination; (3) Me-Dst.TT0: Time of traversal of SRD from the concerned node to the destination.

Next, we present the working of our proposed two-phase security framework. First is the secure routing (secure route discovery and secure route reply) and second is the secure data packet forwarding. Notations used in our protocol are summarized in Table 1.

Table 1: Notations

Notation	*Description*
X_{IP}	Node X's IP address.
X_C	Node X's public key certificate.
X_{S+}	Signing key of node X (private key)
X_{S-}	Verifying key for node X (public key provided by X, through X_C to verify its signature made with X_{S+}).
X*	Digital signature made by node X by using X_{S+}.
X_{E-}	Encrypting key for X (public key provided by node X, through X_C for encrypting any information to be sent to X)
X_{E+}	Decrypting key of node X (private key used by X for decrypting information encrypted with X_{E-}).
XY_K	Session key (symmetric key) between node X and Y
X_{DT0}	Departure time of secure route discovery packet (SRD) from node X.
X_{DT1}	Departure time of secure route reply packet (SRR) from node X.
X_{AT0}	Arrival time of SRD at node X.
X_{AT1}	Arrival time of SRR at node X.
XY_{TT0}	Traversal time of SRD from node X to node Y.
XY_{TT1}	Traversal time of SRR from node X to node Y.
XY_{TT}	Traversal time of a control packet from node X to node Y.
<a, b>	A packet carrying values 'a' and 'b'.
[[<a, b>]] X_{S+}	Packet <a, b> signed with X_{S+}.
{m}X_{E+}	Message 'm' encrypted with X_{E+}. This can only be decrypted using X_{E-}.
X→Brd:<a, b>	Node X broadcasts the packet <a, b>.
X→Y:<a, b>	Node X unicasts the packet <a. b> to node Y.
δ	Clock difference.

C. Working

We first discuss the secure routing phase. This phase begins the network layer communication between two nodes in the ad hoc cloud. We use the ad hoc cloud model shown in Fig. 1 for illustration.

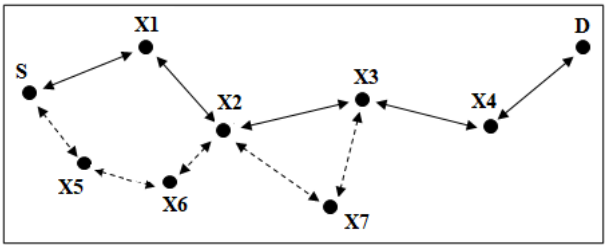

Fig. 1. An ad hoc cloud network under consideration.

The phase starts when a source node 'S' wants to communicate with a destination node 'D', but in the routing table (RT) of S there is no path for D. In such a case, S begins the secure route discovery sub-phase by broadcasting a secure route discovery packet (SRD) as follows.

$$S \rightarrow Brd: \langle\langle SRD, S_{IP}, D_{IP}, S_{DT0}, S_C\rangle, S^*\rangle$$
$$\text{Where, } S^* = [[\langle SRD, S_{IP}, D_{IP}, S_{DT0}, S_C\rangle]]S_{S+}$$

Here, SRD is the packet type identifier. The tuple $\langle S_{IP}, D_{IP}, S_{DT0}\rangle$ is used by the nodes to identify the SRD uniquely thereby the replay attack is prevented.

Before broadcasting, S keeps the required information about this SRD in its RDT.

$$\text{Src.IP} = S_{IP}, \text{Dst.IP} = D_{IP}, \text{Src.DT0} = S_{DT0}, \text{Me.AT0} = \text{NULL},$$
$$\text{Me.DT0} = S_{DT0}, \text{PHop.IP} = \text{NULL}, \text{NHop.IP} = \text{NULL}$$

Node S also sets a timer before the SRD is broadcasted. If S doesn't receive an SRR from D before the expiry of the set timer, a secure route discovery sub-phase may again be initiated by S. This re-initiation of the secure route discovery sub-phase by S may also occur if S receives an SRR within the expiry of the set timer but observes that the security of the SRR has been violated i.e. the SRR has arrived at S traveling through a non-authentic path. The source node S can make up to η (the value is application dependent, normally based on the round-trip-time of the packets) number of secure route discovery attempts.

An intermediate node in the network upon receiving the SRD validates the signature of the previous node and determines the uniqueness of the SRD by looking into the tuple $\langle S_{IP}, D_{IP}, S_{DT0}\rangle$. If the SRD is found to be authentic and unique the intermediate node removes the previous node's signature from SRD provided the previous node doesn't happen to be the source. It then signs the SRD, appends its own certificate, inserts a new row in its RDT for this SRD, and rebroadcasts it. For example, node X1 broadcasts the following SRD that it receives from S after conforming its authenticity and uniqueness.

$$N1 \rightarrow Brd: \langle\langle\langle SRD, S_{IP}, D_{IP}, S_{DT0}, S_C\rangle, S^*\rangle X1^*, X1_C\rangle$$
$$\text{Where, } X1^* = [[\langle\langle SRD, S_{IP}, D_{IP}, S_{DT0}, S_C\rangle, S^*\rangle]]X1_{S+}$$

And the X2 broadcasts the following SRD that it receives from X1 after validating its authenticity and uniqueness.

$$N1 \rightarrow Brd: \langle\langle\langle SRD, S_{IP}, D_{IP}, S_{DT0}, S_C\rangle, S^*\rangle N2^*, X2_C\rangle$$
$$\text{Where, } X2^* = [[\langle\langle SRD, S_{IP}, D_{IP}, S_{DT0}, S_C\rangle, S^*\rangle]]X2_{S+}$$

Ultimately, the SRD reaches D through the path S-X1-X2-X3-X4-D.

The destination D on receipt of the SRD first verifies the signatures of its previous node X4 and that of the source S to determine the authenticity of the SRD. If either of the signatures is found invalid, the SRD is rejected. Otherwise, D compares the tuple $\langle S_{IP}, D_{IP}, S_{DT0}\rangle$ of the SRD with its RDT's tuple <Src.IP, Dst.IP, Src.DT0> to determine whether the received SRD is the first one for the current route discovery attempt (note that, S can attempt up to η route requests). There may be the following three cases.

1. There is a complete match between the tuples <S_{IP}, D_{IP}, S_{DT0}> and <Src.IP, Dst.IP, Src.DT0>. It implies that the current SRD is just a duplicate of an already received SRD. Therefore, the current SRD is rejected.
2. There is a partial match i.e. only the tuple <S_{IP}, D_{IP}> matches with the tuple <Src.IP, Dst.IP>, but S_{DT0} does not match with the value in the field Src.DT0. It implies that the recently received SRD is the first secure route discovery packet of second or onwards secure route discovery attempts made by the source S. An earlier attempt has already been made by S for which an SRR has been sent by D. However, the SRR has been rejected by S because the SRR has reached the S through a non-authentic path. Therefore, in this situation, D needs to find out whether recently received SRD has arrived at D through the same hostile path along which the previous SRR (that was dropped out by S) has arrived at S. For this, node D takes the following set of actions:

   It first calculates SD_{TT0} which is equal to the difference between current SRD's arrival time and S_{DT0}. Then, it determines the traversal time of that SRD (SD_{TT*}) for which the tuple <S_{IP}, D_{IP}> matches the tuple <Src.IP, Dst.IP> of an existing row in the RDT. It is the difference between the matched row's Me.AT0 and Src.DT0.

   If $SD_{TT0} - \delta \leq SD_{TT*} \leq SD_{TT0} + \delta$, it implies that the recently received SRD has arrived at D traveling through the same hostile path along which the previous SRR (that was dropped out by S) has arrived at S. Therefore, in such case the recently received SRD is not accepted. Otherwise, D accepts the SRD, deletes the existing row in the RDT whose <Src.IP, Dst.IP> fields match the tuple <S_{IP}, D_{IP}>, inserts a new row in the RDT for the recently received SRD and initiates the secure route reply.
3. No match is found between the tuples <S_{IP}, D_{IP}, S_{DT0}> and <Src.IP, Dst.IP, Src.DT0> for any row in the RDT. This means that the recently received SRD is the first secure route discovery packet sent by S for the first route discovery attempt. So, in this case, D accepts the SRD. D then makes a new entry in its RDT corresponding to this SRD and initiates the secure route reply sub-phase.

Destination D starts the secure route reply sub-phase by creating an SRR that contains the encrypted session key SD_K, along with other information as given below. D unicasts the SRR back to S along the reverse path D-X4-X3-X2-X1-S.

$$D \rightarrow S: \langle\langle SRR, D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}, \{SD_K\}S_{E-}, D_C\rangle, D^*\rangle$$
$$\text{Where, } D^* = [[\langle SRR, D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}, \{SD_K\}S_{E-}, D_C\rangle]]D_{S+}$$

Here, SRR is the packet type identifier. The tuple <D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}> is used by the intermediate nodes to uniquely identify the SRR. So, the replay attack is prevented. The value S_{DT0} is collected from the RDT corresponding to the SRD against which this SRR is generated.

An intermediate node, say N (anonymous), in the reverse path when receives this SRR, verifies its previous node's signature from whom the SRR is received. If the signature is found invalid the SRR is dropped out. Otherwise, N compares the SRR's traversal time from D to itself (DN_{TT1}) with the traversal time of the SRD from itself to D (ND_{TT0}) against which this SRR is forwarded by D.

$$DN_{TT1} = N_{AT1} - D_{DT1}$$
$$ND_{TT0} = D_{AT0} - N_{DT0}$$

N_{AT1} is the time at which the SRR reached N, D_{DT1} and D_{AT0} are present in the SRR itself, whereas N_{DT0} can be obtained from N's RDT by matching the SRR's tuple <D_{IP}, S_{IP}, S_{DT0}> against the RDT's tuple <Dst.IP, Src.IP, Src.DT0>. The Me.DT0 field of the matching row gives the value. If there is a difference between DN_{TT1} and ND_{TT0} is found to be more than δ it implies that, the SRR has been through a routing loop or it has been redirected through some invalid path. Therefore, the SRR is rejected. However, if the difference between DN_{TT1} and ND_{TT0} happens to be less than or equal to δ, the SRR is treated to be non-victimized and hence accepted for further processing by the concerned node. The intermediate node then removes the signature of its previous node from which the SRR is received provided the previous node doesn't happen to be the destination D. It then signs the SRR by using its own signing key, puts its public key certificate in the SRR, and forwards the SRR to the next node. For example, node X4 after receiving the SRR from D unicasts the following SRR to node X3.

$$X4 \rightarrow X3: \langle\langle\langle SRR, D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}, \{SD_K\}S_{E-}, D_C\rangle, D^*\rangle X4^*, X4_C\rangle$$
$$\text{Where, } X4^* = [[\langle\langle SRR, D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}, \{SD_K\}S_{E-}, D_C\rangle, D^*\rangle\rangle]]X4_{S+}$$

And, node X3 unicasts the following SRR to X2.

$$X3 \rightarrow X2: \langle\langle\langle SRR, D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}, \{SD_K\}S_{E-}, D_C\rangle, D^*\rangle X3^*, X3_C\rangle$$
$$\text{Where, } X3^* = [[\langle\langle SRR, D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}, \{SD_K\}S_{E-}, D_C\rangle, D^*\rangle\rangle]]X3_{S+}$$

Finally, the SRR arrives at S through the path D-X4-X3-X2-X1-S.

The source S, on receipt of the SRR, first verifies the digital signatures of X1 as well as of the destination D. S, then compares DS_{TT1} with SD_{TT0} in a similar way as that of the intermediate nodes.

$$DS_{TT1} = S_{AT1} - D_{DT1}$$
$$SD_{TT0} = D_{AT0} - S_{DT0}$$

If there is a difference of more than δ, the SRR is rejected and a new route discovery phase is initiated. Otherwise, S accepts the SRR and updates its routing table (Dst.IP = D_{IP}, NHop.IP = $X1_{IP}$, Me-Dst.TT0 = SD_{TT0}). Thereafter S extracts the session key SD_K by decrypting it with S_{E+}. The session key SD_K can now be used to encrypt the data packets to be communicated between S and D.

After the establishment of a secure route between the source S and the destination D, S can now start the second phase, the secure data packet forwarding. In this, S unicasts the secure data packet (SDP) to D along the discovered path. The SDP contains the information 'm' that S wants to send to D encrypted using SD_K.

$S \rightarrow D$: $<<SDP, D_{IP}, S_{IP}, S_{DT0}, \{m\}SD_K, S_C>, S^*>$

Where, $S^* = [[<SDP, D_{IP}, S_{IP}, S_{DT0}, \{m\}SD_K, S_C>]]S_{S+}$

The intermediate nodes simply forward the SDP. The destination D, on receipt of the SDP, verifies the signature S by using S_{S-} retrieved from S_C and decrypts the message from the packet by using the session key SD_K. In this way, the secure data packet delivery is achieved.

IV. ANALYSIS

The blend of security techniques used in our proposed scheme makes it resilient against different network-layer security threats as discussed in Section II that may be launched by internal or external malicious nodes in the cloud setup. This section thoroughly analyses the strength of our proposal on the security perspective.

Impersonation: The proposed scheme allows only those packets to be accepted and processed that are signed by using a certified signature key by its originator as well as the previous hop from which it is received by a node. The SRDs are digitally signed by the source and the and SRRs digitally signed by the destination, thereby making it sure that they are only created by their authentic generators.

The SDP also include the signature and certificate of its originator. This enforcement of hop-to-hop and end-to-end authentication prevents the possibility of any node in the network to be impersonated.

Information Disclosure: In the secure routing phase, none of the control packets contain any information that might by-chance expose the network topology. The only exploitable information present is the session key in the SRR. It is encrypted by using the private key of the destination. Moreover, all SDPs in the data forwarding phase are encrypted using the session key established between the source and the destination only. So, disclosure of the information is impossible.

Fabrication: No intermediate node in the network can generate a control packet either during the secure route discovery or the secure route reply phases. Only the source is authenticated to create an SRD and the destination is authenticated to create an SRR. The intermediate node only forwards the control packets. This technique prevents fabrication attacks such as resource consumption, RDT/RT poisoning, and overflow, rushing or the blackhole attack.

Modification: All packets are signed by their originators (SRDs have the source's signature, SRRs have the destination's signature, and SDPs are signed by the source or destination whoever generates it). It ensures that when a packet is altered by some node along the path between the source and destination, it is instantly caught and successively dropped out. Hence, it prevents modification breaches such as the detour or the blacklist attack.

Replication: Every packet in our proposed scheme contains tuple that defines the uniqueness of the packet (SRD has $<S_{IP}, D_{IP}, S_{DT0}>$, SRR has $<D_{IP}, S_{IP}, S_{DT0}, D_{AT0}, D_{DT1}>$, and SDP has $<SDP, D_{IP}, S_{IP}, S_{DT0}>$). These tuples are checked by the intermediate nodes between the source and destination before forwarding the packets (SRD/SRR/SDP). It ensures replication behavior such as tunneling, or the Wormhole be caught. *Packet Redirection:* In the proposed security solution, every node in the path, from source to destination, checks the packet traversal time on the hop-to-hop basis. This approach prevents packet redirections and the routing loop attacks (discussed next) from being launched. To prove our claim, let us consider the cloud set up in Fig. 1. Let μ_X and ξ_{XY} are the packet (SRD/SRR) processing time of an arbitrary node X and packet (SRD/SRR) traversal time between two arbitrary node X and Y (bidirectional) respectively. Packet processing time for SRD and SRR has been considered equal because the action taken by nodes to process either of these packets is identical. Suppose the intermediate node X3 becomes malicious and launches a packet redirection attack as follows. During the secure route reply phase, when the SRR is forwarded by X4, to X3, X3 forwards the SRR to X7 instead of X2 i.e. the SRR reaches X2 through X7, not directly from X3 and here X2 is unaware of such behavior of X3.

This kind of attacking approach, in our proposed scheme, is instantly detected by X2 as follows. When the SRR is received by X2 from X7, X2 compares the $X2D_{TT0}$ with $DX2_{TT1}$ a difference of more than δ is found. So, the SRR is rejected by X2. The proof is presented below.

For the secure route discovery phase,

$$X2D_{TT0} = \xi_{X2X3} + \mu_{X3} + \xi_{X3X4} + \mu_{X4} + \xi_{X4D} \quad \textbf{(1)}$$

For the secure route reply phase, when the SRR is received by X2 through X7,

$$DX2_{TT1} = \xi_{X4D} + \mu_{X4} + \xi_{X3X4} + \mu_{X3} + \xi_{X3X7} + \mu_{X7} + \xi_{X2X7} \pm \delta \quad \textbf{(2)}$$

$$\Rightarrow DX2_{TT1} - X2D_{TT0} = (\xi_{X3X7} + \mu_{X7} + \xi_{X2X7} - \xi_{X2X3} \pm \delta) \quad \textbf{(3)}$$

In equation (3), the value of $\xi_{X3X7} + \mu_{X7} + \xi_{X2X7}$ is always greater than $\xi_{X2X3} \pm \delta$, by a value larger than δ because, if it is not, then during secure route discovery the SRD forwarded by X2 would have reached X3 through X7 earlier than directly. ■

Routing Loops: Let us consider the same set of metrics as we have taken for the discussion of packet redirection attack. But this time lets us assume the nodes X2 and X3 to be malicious. In the course of the SRR phase, they form a routing loop as follows. When the SRR reaches X2 from X3, it sends the SRR back to X3, which again forwards it to X2. X2 which may continue this looping behavior for some time or deliver the SRR to X1.

The proposed scheme can catch such behavior easily. When X1 gets the SRR from X2 (after, say β number of loops) and compares $X1D_{TT0}$ with $DX1_{TT1}$, it finds a difference of more than δ, and hence the SRR is rejected.

For the secure route discovery phase,

$$X1D_{TT0} = \xi_{X1X2} + \mu_{X2} + \xi_{X2X3} + \mu_{X3} + \xi_{X3X4} + \mu_{X4} + \xi_{X4D} \quad \textbf{(1)}$$

For the secure route reply phase, when the SRR is received by X1 after β number of loops,

Time taken by each loop between X2 and X3 = $2 \times \xi_{X2X3}$

(2)

$\Rightarrow DX1_{TT1}$ via β number of loops formed between X2 and X3 = $\xi_{X4D} + \mu_{X4} + \xi_{X3X4} + \mu_{X3} + \xi_{X2X3} + \mu_{X2} + (2\beta \times \xi_{X2X3}) + \xi_{X1X2} \pm \delta$ **(3)**

$\Rightarrow DX1_{TT1} - X1D_{TT0} = (2\beta \times \xi_{X2X3} \pm \delta) > \delta$ ■

Selective Packet Drops: In our proposed scheme, whenever an SRR is dropped by some intermediate malicious node, the source doesn't receive an SRR within the timer expires. Hence, another secure route discovery is initiated by the source. However, this time, the SRD won't be accepted by the destination because it will reach the destination through the same malicious route through the earlier SRD has reached the destination. Therefore, automatically the malicious node performing the attack is excluded from the secure route.

V. CONCLUSION

In this paper, we presented a new two-phase security technique for governing network layer communication in ad hoc clouds. While building the solution, chosen and required cryptographic techniques have been integrated into it in such a way that it becomes robust against different possible network layer breaches that can be launched in ad hoc cloud environments, but at the same time, the efficiency is not degraded. Together with the solutions for lower-layer security, the proposed scheme can build a framework for secure end-to-end communication in an ad hoc cloud environment.

REFERENCES

1. P. Mell and T. Grance, "The NIST definition of cloud computing," Communications of the ACM, vol. 53, no. 6, p. 50, 2010.
2. G. Kirby, A. Dearle, A. Macdonald, and A. Fernandes, "An approach to ad hoc cloud computing," arXiv preprint arXiv: 1002.4738, 2010.
3. 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.
4. 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.
5. 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.
6. Gary A. McGilvary, Adam Barker, and Malcolm Atkinson, "Ad hoc cloud computing." 2015 IEEE 8th International Conference on Cloud Computing. IEEE, 2015.
7. 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.
8. H. Wang, W. He, and F.-K. Wang, "Enterprise cloud service architectures," Information Technology and Management, vol. 13, no. 4, pp. 445–454, 2012.
9. S. Srinivasamurty and D. Q. Liu, "Security and Privacy in Cloud Computing: A Survey," Parallel and cloud computing, vol. 2, no. 4, pp. 126-149, 2013.
10. 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, pp. 1-7, 2010.
11. 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.
12. Niroj Kumar Pani, Bikram Keshari Rath, Sarojananda Mishra. "Localization Adaptive Secure Routing for Ad-Hoc Cloud Networks Based on Synchronized Timestamp Approach", Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies - ICTCS '16, 2016.
13. B. A. Forouzan and D. Mukhopadhyay, Cryptography and Network Security (Special Indian Edition). McGraw-Hill Education, 2011.
14. A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996.
15. 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.
16. M. H. Kharche, and M. D. S. Chouhan, "Implementing trust in cloud using public key infrastructure," International Journal of Advanced Computer Science and Applications, vol. 3, no. 3, 2012.

AUTHORS PROFILE

Niroj Kumar Pani received his M.Tech in Computer Science with specialization in Information security from National Institute of Technology, Rourkela, India in 2009 and Ph.D. in computer science from Utkal University, Odisha, India in 2018. He had worked as Assistant Professor in Indian Institute of Science and Information Technology and as a senior analyst in PMAP India. At present, he is working as Assistant Professor in the Department of Computer Science Engineering and Applications in Indira Gandhi Institute of Technology, Saranga, Odisha, India. His research interests include applied cryptography, network security, wireless ad hoc and sensor networks, clouds, and Internet of Things.

Satyasundara Mahapatra is an Associate Professor in the Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur, Uttar Pradesh. He received his master's degree and Ph.D. in Computer Science from Utkal University, Bhubaneswar, Odisha in 2006 and 2016 respectively. He also holds an MCA degree from Utkal University, Bhubaneswar, Odisha in 1997. His current research interests include Machine Learning, Image Processing, Scheduling, and IoT. He has authored or co-authored in international scientific journals, two book Chapters and three patents approval in his field of expertise.

Rati Ranjan Dash is Professor of Mechanical Engineering Department, College of Engineering and Technology, Bhubaneswar, Odisha, India. He received the bachelor's degree in mechanical engineering from the University College of Engineering, Burla, Odisha in 1987, master's degree in applied mechanics from Indian Institute of Technology, Delhi, India in 1993 and the Ph.D. degree in Robotics from Utkal University, Bhubaneswar, India in 2004. His main research interests are in control of mechanical systems, robotics, multibody dynamics, robot manipulation, scheduling, and soft computing.