RUSHING ATTACK DEFENDING CONTEXT AWARE TRUSTED AODV IN AD-HOC NETWORK

Swarnali Hazra[1] and S.K.Setua[2]

[1]Department of Computer Science & Engineering, University of Calcutta, Kolkata, India
swarnali.hazra@gmail.com

[2] Department of Computer Science & Engineering, University of Calcutta, Kolkata, India
sksetua@gmail.com

ABSTRACT

This research aims to identify the security threats in on-demand routing protocol, AODV for Ad-hoc network. In this context we have extended the AODV protocol with trust and recommendation to secure the network. In AODV, establishment of routing path depends on the faster route request and route reply packets. Rushing attacker exploits such faster traversal activities to attack the network. Rushing attackers are identified based on their misbehaviour in comparison to other nodes of the network. Furthermore, trust value is assigned to the misbehaving node and the same is augmented with other aspects of trust like dependency pattern, context, previous history and dynamicity. Finally, based on threshold value of trust, trust evaluating node takes the decision to include or not to include the trustee node in routing path depending on the final trust value. To facilitate the trust computation and decision of our trust model, AODV is enhanced with different functional modules: Node Manager, Trust Module and Decision Manager. Trust based AODV secures the routing path by isolating the rushing attacker, based on their trust value. Our analysis and simulation results show the effectiveness of our proposal against rushing attack.

KEYWORDS

AODV, Rushing Attack, Trustor, Trustee, Direct Trust, Indirect Trust .

1. INTRODUCTION

Ad-hoc network is decentralized and dynamic in nature. To establish communication between two nodes in ad-hoc network, on-demand routing protocols are very popular in use. On-demand routing mechanism is victimised by various types of attacks during route discovery process. Rushing attack is such a vulnerable attack. This research aims to isolate the rushing attacker in on-demand routing protocol, AODV for Ad-hoc network.

In AODV, whenever source requires a communication with a destination, source initiates route discovery process by broadcasting route request packet (RREQ) to establish a communication route to that destination. Intermediate nodes or routers pass RREQ until the destination is found. Destination or intermediate node which has valid route to destination replies with RREP just after receiving of RREQ. To limit the overhead of RREQ flooding, every intermediate node only forward the 1st received routing packets (RREQ/RREP) and discards all late received identical routing packets. Rushing attacker takes the advantage of this mechanism. Rushing attacker

DOI : 10.5121/ijsptm.2012.1305

forwards rushed routing packets to target node more quickly than legitimate nodes. As a result, target or victimized node only considers the 1st received rushed routing packet for route discovery and discards all late arrived legitimate routing packets. In this way rushing attacker includes itself in discovered route and attracts all the data to exploit.

Rushing attacker sends rushed routing packets either by ignoring MAC/routing layer delay or by using higher transmission range. We have incorporated the trust concept to isolate rushing attacker from discovered route in AODV. We have introduced a new model i.e. Trusted On-demand Routing (TOR) model and based on this TOR model we have proposed CAT-AODV-R (Context Aware Trusted AODV against Rushing attack). Trust is belief or trust level between trustor and trustee. Trustor (CT) is the believing entity which evaluates the trust level for trustee (TE) on the basis of context dependent trust computation. The final trust computation is dependent on direct and indirect trust under timing constraint. Direct trust is evaluated on the basis of context based expectations and misbehaviour identification. On the other hand, indirect trust is evaluated depending on recommendations and notifications. If CT evaluated Trust level for TE is not up to the threshold limit, CT does not consider TE in route discovery process by discarding malicious TE forwarded routing packet.

In the section 2, the status of the considered domain is presented and the rushing attack is discussed in section 3. In section 4, TOR model architecture is explained. In section 5, functionalities of CAT-AODV-R are discussed on the basis of trust computation of underlying TOR model. Simulation results of our experiments are presented in section 6. Section 7 includes the conclusion part.

2. RELATED WORK

Trust becomes a popular approach to provide security in a distributed way for ad-hoc networks. Trust concept is proposed in different ways and in different aspects. In [7] , a trust monitoring architecture called TrAM (Trust Architecture for Monitoring), monitors trustworthiness of service users at run-time depending on trust rules and calculation mechanism for preventing occurrences of unwanted events. A system [4], based on path reputation and trust value, is proposed to enhances network throughput and reliability of discovered route. Here, trust value is incremented and decremented according to positive and negative observations, respectively. An integrated trust management model [8] is introduced to select trustworthy service by quantification of trust on platform of context-aware service. This model, addresses a basic set of trust aspects related to identity provisioning, privacy enforcement, and context provisioning activities.

Using trust concept AODV is secured by many proposed approaches. Trust-based SAODV [3] is proposed on the basis of intrusion detection mechanism (IDM) and trust-based mechanism (TBM) to penalize selfish nodes in AODV. In [1], a modified version of AODV is proposed on the basis of node trust and route trust to secure AODV. This work supports continuous node performance evaluation and neighbour node's recommendations collection. In [6], implicit trust relations between nodes are used to differentiate between trustworthy nodes from malicious nodes in AODV. Here, implicit trust relations of the AODV are formalized. Based on these relations, each node is able to reason on the actions performed by its neighbours and deduces information about their knowledge. Finally, deduced information is used to supervise the behaviour of neighbour and to detect malicious nodes. In [5], a trust-based framework is proposed for improving security and robustness of AODV. This mechanism is based on incentives and penalties depending on the behaviour of network nodes.

3. RUSHING ATTACK IN AODV

AODV routing protocol only considers the 1st received routing packets for route discovery. Rushing attacker sends rushed routing packets (RREQ or RREP) more quickly to target node in comparison to other legitimate sender node. Rushing attackers send rushed routing packets more quickly by ignoring MAC layer and/or routing layer delay, or by using higher transmission range. In Figure 1 (a), node R_1 is rushing attacker which sends rushed RREQ more quickly to target node 3, in comparison to node 2, by ignoring delays. Node 3 discards late received node 2 forwarded legitimate RREQ and forwards the first received R_1 forwarded rushed RREQ to destination D. Consequently, D replies with RREP towards source via R_1. As a result, source forwards all data packets towards R_1.

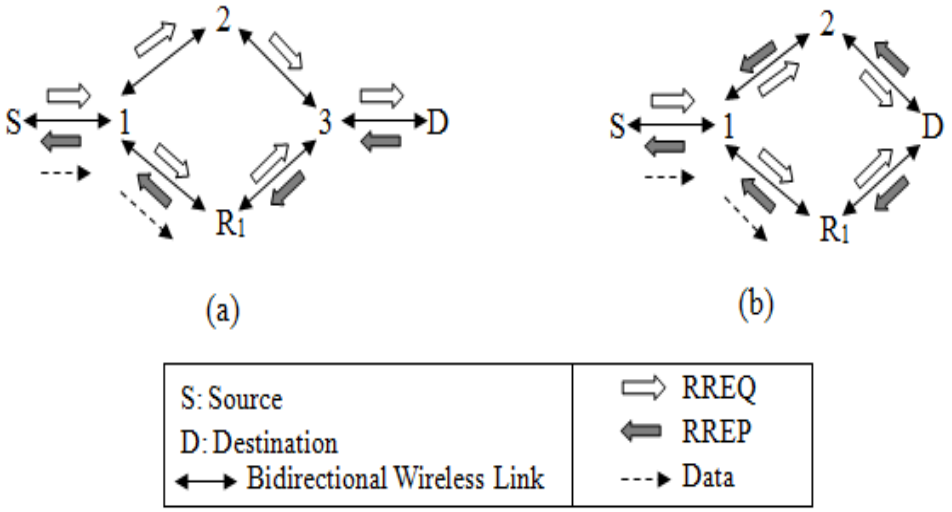

Figure 1. Rushing attacker ignores MAC/routing layer delay

In Figure 1 (b), node R_1 is rushing attacker which sends rushed RREP more quickly to target node 1, in comparison to node 2, by ignoring delays. Node 1 discards late received node 2 forwarded legitimate RREP and forwards the first received R_1 forwarded rushed RREP to destination S. As a result, S forwards all data towards R_1 and R_1 exploits those data.

Another type of rushing attacker sends rushed routing packets to target node using higher transmission range. Here the higher transmission range is at least twice the normal transmission range. In Figure 2 (a), node R_2 is rushing attacker which sends rushed RREQ to target node 4 more quickly in comparison to node2. Node 4 discards late received legitimate RREQ that reached form node 2 via node 3 and forwards the first received R_2 forwarded rushed RREQ to destination D. Consequently, D replies with RREP towards source via R_2. Since, R_2 is using higher transmission range, the wireless link between node 4 and R_2 is not bidirectional link. When D forwarded RREP reaches node 4, it can not be forwarded to R_2 because of shorter transmission range of node 4. As a result, node S can not get RREP and no route will be discovered between S and D.

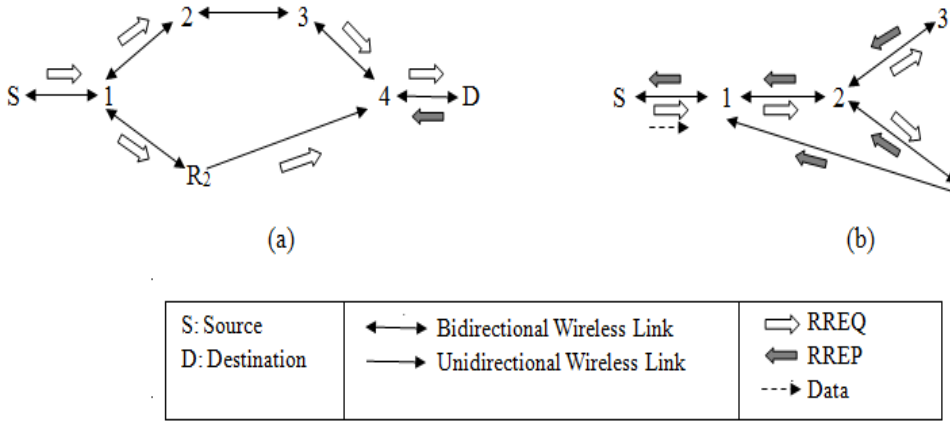

Figure 2. Rushing attacker uses higher transmission range

In Figure 2 (b), node R_2 is rushing attacker which sends rushed RREP using higher transmission range. R_2 sends rushed RREP to target node 1 more quickly in comparison to node3. Node 1 discards late received legitimate RREP that reached form node 3 via node 2 and forwards the first received R_2 forwarded rushed RREP to source S. Consequently, S sends all data to R_2 towards D. Since, R_2 is using higher transmission range, the wireless link between R_2 and node 1 is not bidirectional link. When S forwarded RREP reaches node 1, it can not be forwarded to R_2 since of shorter transmission range of node 1. As a result, node D can not get any data from S.

4. TOR MODEL

In this work, AODV is extended with TOR model to establish secure routing path between source and destination by avoiding malicious nodes. The following section describes the structural and functional components of the model. TOR model (Figure 3) consists of three functional modules (Node Manager, Trust Module and Decision Manager) along with the on-demand routing protocol, AODV.

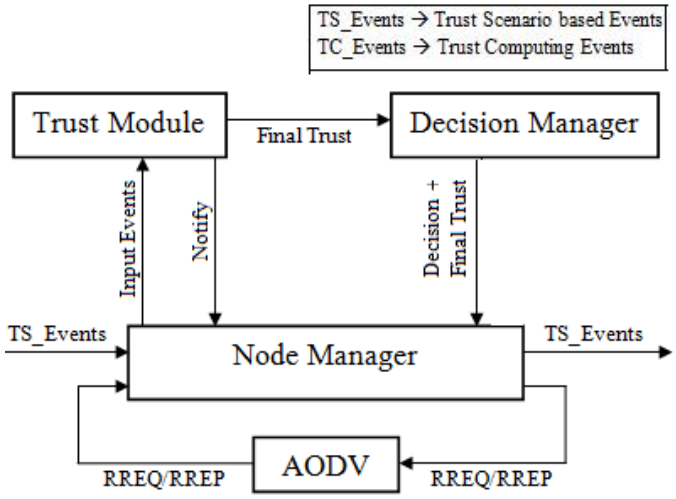

Figure 3. TOR Model

4.1. Node Manager

On receipt of AODV specified RREQ, RREP and TSB_Events, CT's Node Manager sends TC_Events to Trust Module for computing trust value for TE. On the other hand, based on received decision (based on trust value) from Decision Manager, Node Manager either considers TE in the route discovery (case of belief decision) or avoids TE (case of disbelief decision). It also broadcasts the computed final trust value as notification to other nodes. When a node receives this notification, it sends the value to Trust Module for storing in Final Trust Repository. When CT requires recommendation about TE, Node Manager of recommender sends recommendation to CT's Node Manager.

4.2. Trust Module

Trust Module (Figure 4) of TOR model is responsible for trust value computation of TE. Trust Module consists of Trust Engine, Direct Trust Manager and Indirect Trust Manager for computing different levels of trust values which are stored in respective repositories. Trust Module sends the computed final trust to decision Manager for taking belief-disbelief decision. Context Analyzer of Trust Module analyzes the contexts of incoming recommendations and trust notifications. Trust Module also has Event Analyzer for analyzing input events and Notifier for notifying output events. On the other hand, Trust Module sends recommendation from the Final Trust Repository and also stores the incoming notified final trust value for a TE.

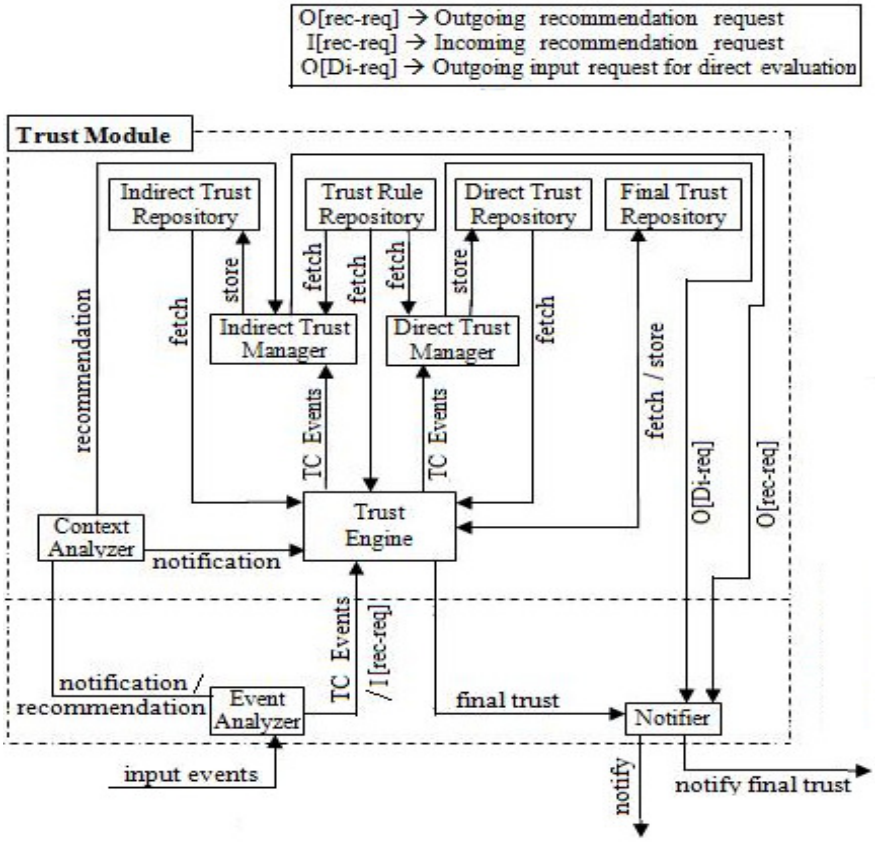

Figure 4. Trust Module of TOR Model

4.3. Decision Manager

On the basis of received trust value from Trust Module, Decision Manager takes either belief or disbelief decision for TE. If the computed trust value is greater than 0.5, it takes belief decision otherwise it takes disbelief decisio. It sends this decision to Node Manager for considering or not considering TE in route discovery. It also forwards the received final trust value to Node Manager for notifying other nodes in the network.

5. CAT-AODV-R (CONTEXT AWARE TRUSTED AODV AGAINST RUSHING ATTACK)

In CAT-AODV-R, context (C) is classified into context-1 and context-2, and on these considered contexts two types of misbehaviours (misbehaviour-1, misbehaviour-2) of rushing attacker is defined.

Context-1 (C1): C1 is defined with respect to ignorance of routing layer delay and/or MAC layer delay.

Context-2 (C2): C2 is defined with respect to usage of higher transmission range of a node in comparison to other nodes of the network.

Misbehaviour-1 (M1): Based on C1, if a particular node sends rushed RREQ packets by ignoring MAC and/or routing layer delays, this behaviour is considered as M1.

Misbehaviour-2 (M2): Based on C2, if a particular node sends rushed RREQ by using higher transmission range, this behaviour is considered as M2.

In CAT-AODV-R, every node broadcasts RQres packet (response packet of RREQ). In Figure 5, when TE receives RREQ, it broadcasts RQres and after receiving it, CT considers RQres receiving time for direct evaluation of TE. After necessary processing TE broadcasts RREQ and after receiving it, CT considers RREQ receiving time for direct evaluation. Next, CT broadcasts RQres in response of received RREQ. Against RQres, TS_Events are collected for direct trust evaluation, and also recommendations are collected for indirect trust evaluation.

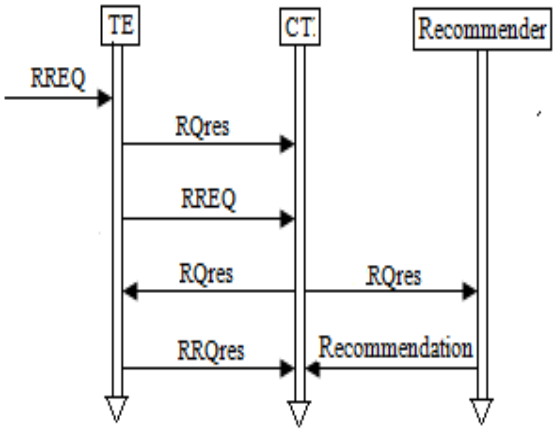

Figure 5. Packet transfer sequence

CAT-AODV-R deals with following set of symbols:

- T_{Packet}: (time taken for packet transmission and reception) + (packet travel time) + (MAC and routing layer delays) + (queuing time at receiver node). Here, Packet stands for RREQ / RQres / RRQres / recommendation packet.
- T_{Pi} : Processing time, where i = 1, 2,…. different levels of processing.
- $T_{Const\text{-}j}$: Different constant times for network. Where j = 1,2,… .

In this work, CT evaluates final trust of TE depending on different level of trusts. Inter-dependencies among trust levels are shown in Figure 6.

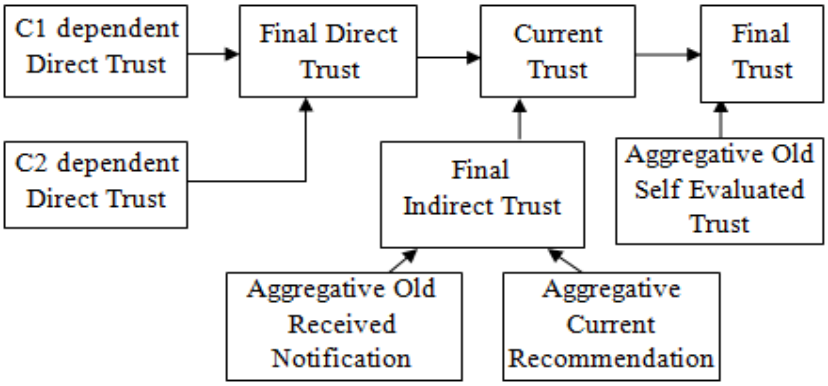

Figure 6. Trust Chain

In symbolic representation $[_XT_Y]$, T denotes trust of X on Y. Here C1, C2 and C are the contexts of computation; and t_{cur} or t_{old} are time instants at which respective trust values are computed. Symbols are used in CAT-AODV-R as follows:

- $[_{CT}T^{C1}{}_{TE}]t_{cur}D$: Context-1 dependent Direct Trust
- $[_{CT}T^{C2}{}_{TE}]t_{cur}D$: Context-2 dependent Direct Trust.
- $[_{CT}T^{C}{}_{TE}]t_{cur}D$: Direct Trust.
- $[_{CT}T^{C}{}_{TE}]t_{old}N$: Aggregated Old Received Notification
- $[_{CT}T^{C}{}_{TE}]t_{cur}R$: Aggregated Current Recommendation
- $[_{CT}T^{C}{}_{TE}]t_{cur}I$: Indirect Trust
- $[_{CT}T^{C}{}_{TE}]t_{cur}T$: Current Trust
- $[_{CT}T^{C}{}_{TE}]t_{old}S$:Aggregated Old Self Evaluated Final Trust
- $[_{CT}T^{C}{}_{TE}]t_{old}FT$: Current Final Trust,… .

CAT-AODV-R computes trust in phases. These are concerned with initiation, computation, decision and reaction phases. Phase-1 is for initiating Direct and Indirect Trust Manager to compute final direct trust and final indirect trust respectively. Considering phase-2, phase-3 and phase-4, final direct and indirect trust is computed in phase-5. In phase-6, current trust is computed by Trust Engine depending on final direct and indirect trust. Based on current trust and collaborative old self evaluated final trust, Trust Engine computes the final trust in this phase. Finally, on the basis of computed final trust, belief or disbelief decision is taken for TE in phase-7. Based on belief-disbelief decision, CAT-AODV-R avoids rushing attacker, and secures the route discovery.

5.1. Phase-1: Initiating Direct and Indirect Trust Manager

After receiving RREQ, TE broadcasts RQres and CT receives this RQres at time $T_1(TE)$. Next, after necessary processing, TE broadcast the received RREQ and CT receives it at time $T_2(TE)$. CT's Node Manager sends $T_1(TE)$ and $T_2(TE)$ as TC_Events to Trust Engine via Event Analyzer. Then Trust Engine initiates Direct Trust Manager by sending $T_1(TE)$ and $T_2(TE)$, for computing $[_{CT}T^{C1}{}_{TE}]t_{cur}D$. On the other hand, Trust Engine fetches stored notified trust values from Final Trust Repository and initiates Indirect Trust Manager by sending these fetched values for computing $[_{CT}T^{C}{}_{TE}]t_{old}N$.

5.2. Phase-2: $[_{CT}T^{C1}{}_{TE}]t_{cur}D$ and $[_{CT}T^{C}{}_{TE}]t_{old}N$ Computation

After receiving initial TC_Events ($T_1(TE)$ and $T_2(TE)$) from Trust Engine, CT's Direct Trust Manager calculates T(TE). Here, TE = ($T_2(TE) - T_1(TE)$). Direct Trust Manager compares T(TE) with specific threshold time α. Here, $\alpha = (T_{RREQ} + T_{P1} + T_{Const\text{-}1})$ where, T_{P1} and $T_{Const\text{-}1}$ are constant for the network. For α, the time T_{RREQ} is the specified standard time, concerned with legitimate TE forwarded legitimate RREQ to CT. If $T(TE) < \alpha$, Misbehaviour M1 of TE is identified based on context C1. On the basis of M1 identification, CT's Direct Trust Manager assigns $[_{CT}T^{C1}{}_{TE}]t_{cur}D$ for TE as per equation (1).

$$f_1\{T(TE)\} = \begin{cases} For\ T(TE) < \alpha,\ [\ {}_{CT}T_{TE}^{C1}\ ]t_{cur}D = 0.1; & (disbelief) \\ For T(TE) \geq \alpha, [\ {}_{CT}T_{TE}^{C1}\ ]t_{cur}D = 0.9; & (belief) \end{cases} \quad (1)$$

Lemma 1. For $T(TE) < \alpha$, $[_{CT}T^{C1}{}_{TE}]t_{cur}D = 0.1$.

Proof: Here, T(TE) = ($T_2(TE) - T_1(TE)$) = $(T_{RREQ} + T_{P1} + T_{Const\text{-}1})$. For T(TE), the time T_{RREQ} is concerned with TE forwarded RREQ to CT. If $T(TE) < \alpha$, it implies that T_{RREQ} is taking less time than standard time, specified for legitimate TE forwarded legitimate RREQ to CT, since T_{P1} and $T_{Const\text{-}1}$ are constant for the considered network. It implies that, TE is forwarding rushed RREQ to CT by ignoring MAC and/or routing layer delays. As a consequence, CT's Direct Trust Manager identifies misbehavior-1 of TE on the basis of context-1 and therefore $[_{CT}T^{C1}{}_{TE}]t_{cur}D = 0.1$ as the case of disbelief.

On the other hand, after receiving stored notified trust values ($[Ti^{C}{}_{TE}]t_{old}$, where i=1,2…n; n=total numbers of old notified trust values) from Trust Engine, CT's Indirect Trust Manager computes $[_{CT}T^{C}{}_{TE}]t_{old}N$ for TE as per equation (2). Here $e^{-(told-t0)}$ is time decaying function, where t_0 is initial time and t_{old} is the old specified time at which received notified trust value is computed.

$$[\ {}_{CT}T_{TE}^{C}\ ]t_{old}N = \frac{1}{n} \times \{ \sum_{i=1}^{n} [Ti_{TE}^{C}\ ]t_{old} \times e^{-(t_{old} - t_o)} \} \quad (2)$$

Next, Indirect Trust Manager and Direct Trust Manager send recommendations request (O[rec-req]) and request of input for direct evaluation (O[Di-req]) respectively to Node Manager via Notifier. Against O[rec-req] and O[Di-req], CT's Node Manager broadcasts RQres packet after initializing time to zero. Against RQres, CT not only collects recommendations from recommenders for indirect evaluation, but also collects response packet of RQres i.e. RRQres from TE for direct evaluation.

5.3. Phase-3: 2nd time initiation of Direct and Indirect Trust Manager

After broadcasting RQres at time instant zero, CT waits for RRQres till time (T_{RQres} + T_{P2} + T_{RRQres} + $T_{Const-2}$). Here, T_{RQres} is concerned with CT forwarded RQres to TE, and TR_{RRQres} is concerned with TE forwarded RRQres to CT. T_{P2} is the time taken for processing at TE after receiving RQres from CT and before sending RRQres to CT. In response to RQres, if CT does not get back RRQres from TE within specified time, CT's Node Manager sends Nack[RRQres] as TC_Events to Direct Trust Manager via Event Analyzer and Trust Engine, otherwise sends Ack[RRQres], for computing $[_{CT}T^{C2}_{TE}]t_{cur}D$.

As well as, in response to RQres, CT gets recommendations about TE from Recommenders. CT's Node Manager sends these recommendations as TC_Events to Context Analyzer via Event Analyzer. If the contexts of incoming recommendations are in valid context set C (set of C1 and C2), Context Analyzer sends these recommendations to Indirect Trust Manager, for computing $[_{CT}T^{C}_{TE}]t_{cur}R$.

5.4. Phase-4: $[_{CT}T^{C2}_{TE}]t_{cur}D$ and $[_{CT}T^{C}_{TE}]t_{cur}R$ computation

If CT's Direct Trust Manager receives Nack[RRQres], it understands that CT did not get back RRQres in response of RQres, within specified time, since of CT's smaller transmission range than TE. In this case, CT's Direct Trust Manager identifies Misbehaviour M2 of TE on the basis of context C2, and it disbelieves TE. Based on M2 identification, CT's Direct Trust Manager assigns the value of $[_{CT}T^{C2}_{TE}]t_{cur}D$ for TE as per equation (3).

$$f_2\{\, Nack[RRQres], Ack[RRQres] \,\} = \begin{cases} For\ Nack[RRQres], \quad [\ {}_{CT}T^{C2}_{TE}\ ]t_{cur}D = 0.1\,; \quad (disbelief) \\ For Ack[RRQres], \quad [\ {}_{CT}T^{C2}_{TE}\ ]t_{cur}D = 0.9\,; \quad (belief) \end{cases} \tag{3}$$

Lemma 2. For Nack[RRQres], $[_{CT}T^{C2}_{TE}]t_{cur}D = 0.1$.

Proof: If, Direct Trust Manager receives Nack[RRQres], it implies that CT did not receive RRQres from TE within specified time (T_{RQres} + T_{P2} + T_{RRQres} + $T_{Const-2}$). In this case, TE forwarded packet reaches CT (CT receives TE forwarded RREQ) but CT forwarded packet could not reach TE because of CT's smaller transmission range than TE. That means TE is forwarding rushed RREQ packet to CT using higher transmission range. As a consequence, CT's Direct Trust Manager identifies misbehavior-2 of TE on the basis of context-2 and therefore $[_{CT}T^{C2}_{TE}]t_{cur}D$ = 0.1 as the case of disbelief.

On the other hand, based on received recommendations ($R[_{TRi}T^{C}_{TE}]t$, where TRi is i^{th} recommender and i=1,2,…n; n=total number of recommendations), Indirect Trust Manager computes $[_{CT}T^{C}_{TE}]t_{cur}R$ for TE as per equation (4). Here, $[_{CT}T^{C}_{TRi}]t$ is CT's trust for TRi , and $e^{-(t-t0)}$ is time decaying function. The time: t is computing time instant. Adviser

$$[\,{}_{CT}T^{C}_{TE}]t_{cur}R = \frac{\sum_{i=1}^{n}\{R[\,{}_{TRi}\,T^{C}_{TE}]t \times e^{-(t\,-t_o)}\} \times \{[\,{}_{CT}T^{C}_{TRi}]t \times e^{-(t\,-t_0)}\}}{\sum_{i=1}^{n}\{[\,{}_{CT}T^{C}_{TRi}]t \times e^{-(t\,-t_0)}\}} \quad (4)$$

5.5. Phase-5: $[_{CT}T^{C}_{TE}]t_{cur}D$ and $[_{CT}T^{C}_{TE}]t_{cur}I$ Computation

CT's Direct Trust Manager computes $[_{CT}T^{C}_{TE}]t_{cur}D$ for TE as per equation (5), and stores it in Direct Trust Repository. Then it sends storing acknowledgement of $[_{CT}T^{C}_{TE}]t_{cur}D$ to Trust Engine.

$$[\,{}_{CT}T^{C}_{TE}]t_{cur}D = \{W_1 \times [\,{}_{CT}T^{C1}_{TE}]t_{cur}D\} + \{(1-W_1) \times [\,{}_{CT}T^{C2}_{TE}]t_{cur}D\} \quad (5)$$

If misbehaviour M1 is identified in context C1 and misbehaviour M2 in context C2 is not identified, value of W1 is 0.9. If misbehaviour M2 in context C2 is identified and misbehaviour M1 in context of C1 is not identified, value of W_1 is 0.1. On the other hand, if both M1 and M2 are identified or both are not identified in contexts C1 and C2 respectively, W1 is of value 0.5.

On the other hand, Indirect Trust Manager computes $[_{CT}T^{C}_{TE}]t_{cur}I$ for TE as per equation (6), and store it in Indirect Trust Repository. Then it sends the storing acknowledgement of $[_{CT}T^{C}_{TE}]t_{cur}I$ to Trust Engine.

$$[\,{}_{CT}T^{C}_{TE}]t_{cur}I = \{0.5 \times [\,{}_{CN}T^{C}_{TE}]t_{cur}R\} + \{0.5 \times [\,{}_{CN}T^{C}_{TE}]t_{old}N\} \quad (6)$$

5.6. Phase-6: $[_{CT}T^{C}_{TE}]t_{cur}T$, $[_{CT}T^{C}_{TE}]t_{old}S$ and $[_{CT}T^{C}_{TE}]t_{cur}FT$ Computation

After getting storage acknowledgement of Direct trust and Indirect trust into their respective repository, CT's Trust Engine fetches $[_{CT}T^{C}_{TE}]t_{cur}D$ and $[_{CT}T^{C}_{TE}]t_{cur}I$ from Direct Trust Repository and Indirect Trust Repository respectively. Then Trust Engine computes $[_{CT}T^{C}_{TE}]t_{cur}T$ for TE with the help of fetched $[_{CT}T^{C}_{TE}]t_{cur}D$ and $[_{CT}T^{C}_{TE}]t_{cur}I$ as per equation (7).

$$[\,{}_{CT}T^{C}_{TE}]t_{cur}T = 0.8 \times [\,{}_{CT}T^{C}_{TE}]t_{cur}D\} + \{0.2 \times [\,{}_{CT}T^{C}_{TE}]t_{cur}I\} \quad (7)$$

Then Trust Engine fetches old self evaluated final trust values($[_{CT}Ti^{C}_{TE}]t_{old}FT$, where i=1,2,…n ; n=total numbers of old self computed values) for Final Trust Repository. Next, it computes $[_{CT}T^{C}_{TE}]t_{old}S$ for TE as per equation (8).

$$[\,{}_{CT}T^{C}_{TE}]t_{old}S = \frac{1}{n} \times \{\sum_{i=1}^{n}[\,{}_{CT}Ti^{C}_{TE}]t_{old}FT \times e^{-(t_{old}-t_o)}\} \quad (8)$$

Finally, Trust Engine computes $[_{CT}T^{C}_{TE}]t_{cur}FT$ for TE, with the help of $[_{CT}T^{C}_{TE}]t_{cur}T$ and $[_{CT}T^{C}_{TE}]t_{old}S$ for TE as per equation (9).

$$[\,{}_{CT}T^{C}_{TE}]t_{cur}FT = \{0.7 \times [\,{}_{CT}T^{C}_{TE}]t_{cur}T\} + \{0.3 \times [\,{}_{CT}T^{C}_{TE}]t_{old}S\} \quad (9)$$

Trust Engine stores this Final $[_{CT}T^{C}_{TE}]t_{cur}FT$ in Final Trust Repository, and sends it to Decision Manager for taking belief-disbelief decision.

5.7. Phase-7: Decision and Reaction

If CT's Decision Manager receives $[_{CT}T^{C}_{TE}]t_{cur}FT$, having value greater than 0.5, it takes belief decision for TE, otherwise it takes disbelief decision. Decision Manager sends the final trust value and taken decision to Node Manager. If CT's Node Manager receives belief decision, it broadcasts TE forwarded RREQ, and if it receives disbelief decision, it avoids TE by discarding the TE forwarded RREQ. Finally, CT's Node Manager notifies final trust value.

If CT believes TE and broadcasts TE forwarded RREQ, CT appends the IP address of TE in RREQ. Only the IP address of current TE is maintained in RREQ. If CT discards a TE forwarded RREQ, CT stores a tag which indicates TE as malicious RREQ sender. If later a RREQ of same route discovery reaches to that CT with appended IP address of CT evaluated malicious TE, CT discard that RREQ immediately.

When destination node receives RREQ, it evaluates the trust of the RREQ sender node (destination is CT and RREQ sender node is TE) by the same process.

6. SIMULATION RESULT

We conducted simulation experiments to evaluate the performance of the proposed routing algorithm CAT-AODV-R, in presence of rushing attack which behaves according to Misbehavior-1 (M1) or Misbehavior-2 (M2). Our CAT-AODV-R is also compared with existing routing algorithm, AODV. The traffic type is CBR. Here considered network is over a 1000m×1000m terrain. Presented results are evaluated with 100 simulation runs.

Figure 7 shows that CAT-AODV-R detects rushing attacker efficiently as the detection rate is efficient, with respect to network of 100 nodes. Detection rate denotes the rate of detection of rushing attacker among total rushing attackers present in the network.

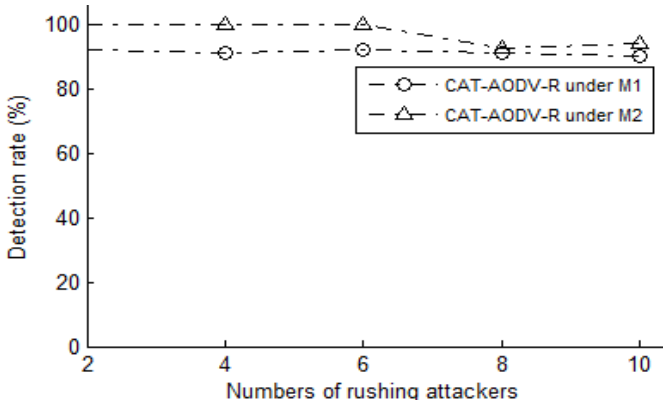

Figure 7. Detection rate vs. number of rushing attackers

Figure 8 shows that the legitimate RREQ success in CAT-AODV-R is much higher than AODV, with respect to network of 50 nodes. Legitimate RREQ success denotes the win of legitimate RREQ against rushed RREQ.

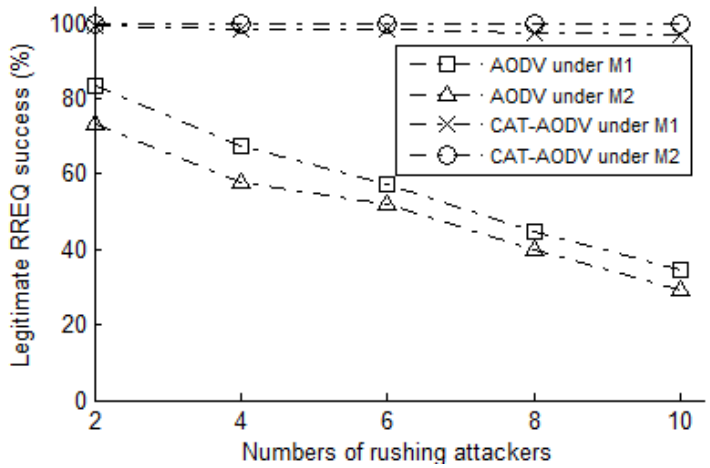

Figure 8. Legitimate RREQ success vs. number of rushing attackers

We have considered the positions of rushing attackers at near source, near destination and anywhere. In Figure 9, we have considered the positions of rushing attackers at near source; in Figure 10, the positions of rushing attackers at near destination and in Figure 11, the positions of rushing attackers at anywhere. From Figure 9, Figure 10 and Figure 11 we can see that rushing attackers can not get success in our CAT-AODV-R where as AODV is highly victimised in presence of rushing attackers. Attack success rate denotes the number(s) of discovered route included rushing attacker(s) with respect to the attacker free legitimate discovered route. Here the considered network is of 50 nodes.

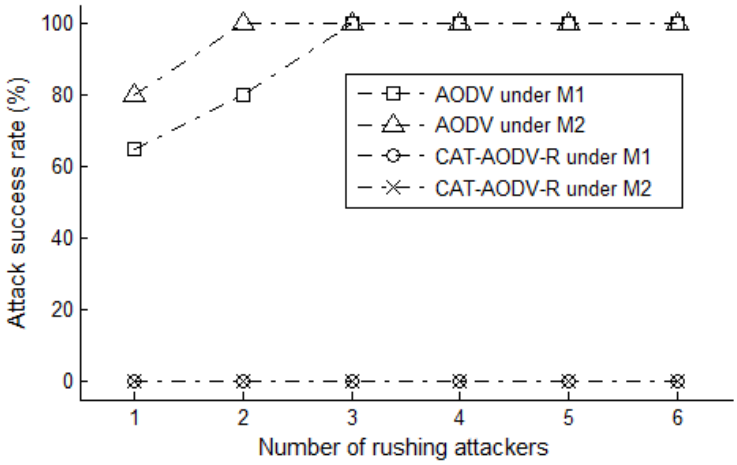

Figure 9. Attack success rate vs. number of rushing attackers at near source

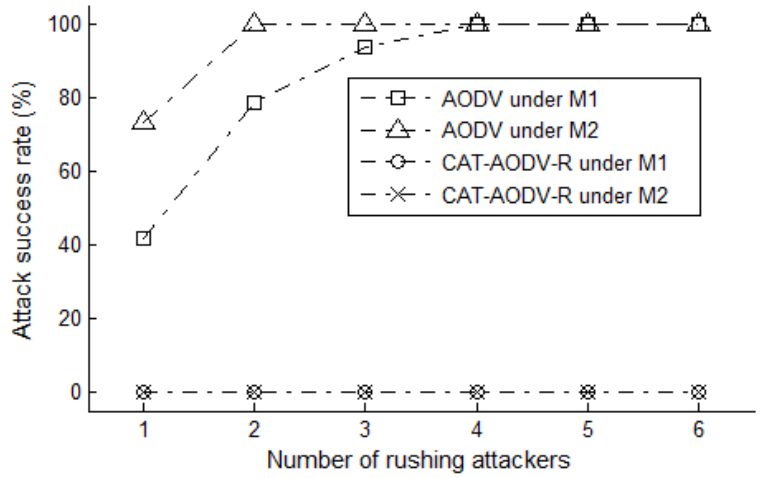

Figure 10. Attack success rate vs. number of rushing attackers at near destination

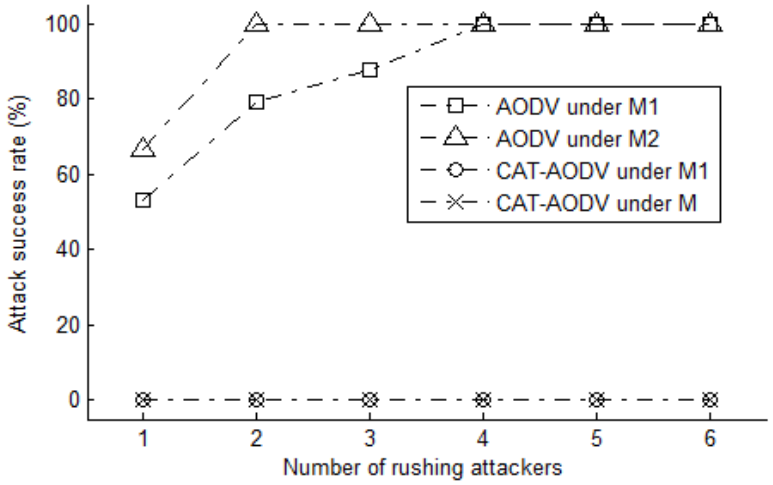

Figure 11. Attack success rate vs. number of rushing attackers at anywhere

7. CONCLUSIONS

CAT-AODV-R protocol for ad hoc network has been presented. We proposed a new solution using trust concept, against the rushing attack problem for existing on demand routing protocol AODV, in ad hoc networks. With the help of proposed trust model, all CAT-AODV-R supported nodes cooperate together to detect and avoid misbehavior-1 (M1) or misbehavior-2 (M2) behaving rushing attacker nodes in a more reliable fashion. Our detection-avoidance scheme detects the misbehaving rushing attacker nodes and isolates them from the active data forwarding and routing on the basis of belief-disbelief decision which comes from evaluated trust value. More research into this novel mechanism for secure routing is necessary. For further research, we will work on improving the proposed trust model, which may provide a solution to other attacks in ad-hoc network.

REFERENCES

[1] A.Menaka Pushpa M.E., "Trust Based Secure Routing in AODV Routing Protocol", IMSAA, Bangalore, 1-6, 2009.

[2] Daoxi Xiu and Zhaoyu Liu, "A Formal Definition for Trust in Distributed", SpingerLink, Lecture Notes in Computer Science,Volume 3650/2005, 482-489, 2005.

[3] Floriano De Rango and Salvatore Marano, "Trust-based SAODV Protocol with Intrusion Detection and Incentive Cooperation in MANET", IWCMC'09, Leipzig, Germany, 2009.

[4] Ji Li, Teng - Sheng Moh, and Melody Moh, "Path-Based Reputation System for MANET Routing", SpingerLink, Volume 5546/2009, 48-60, 2009.

[5] Kamal Deep Meka, Mohit Virendra and Shambhu Upadhyaya, "Trust Based Routing Decisions in Mobile Ad-hoc Networks", CiteSeerX, 2009.

[6] Mohamed Ali Ayachi, Christophe Bidan, Tarek Abbes and Adel Bouhoula, "Misbehavior Detection using Implicit Trust Relations in the AODV Routing Protocol", International Conference on Computational Science and Engineering, 2009.

[7] Mohammad Gias Uddin and Mohammad Zulkernine, "A Trust Monitoring Architecture for Service Based Software", Springer Science+Business Media, LLC 2009.

[8] Ricardo Neisse, Maarten Wegdam1, Marten van Sinderen1, and Gabriele Lenzini, "Trust Management Model and Architecture for Context-Aware Service Platforms", Springer-Verlag Berlin, Heidelberg, 2007.

[9] Stefano Basagni, Marko Conti, Silvia Giordano and Ivan Stojmenovic. Mobile Ad Hoc Networking (chapter: 10). A JOHN WILEY & SONS, INC., PUBLICATION.

[10] YihChun Hu, Adrian Perrig and David B. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocol", WiSe 2003, San Diego, California, USA, September 19, 2003.

Authors

Swarnali Hazra, received her B.Tech from West Bengal University of Technology, M.Tech from University of Calcutta Computer Science and Engineering. Currently she is Ph.D scholar of University of Calcutta. Her research interests include ad-hoc network, network security, sensor network, routing, clustering, image processing, graph theory etc. She has published many research papers in international journals and conference.

Sanjit Kr. Setua is Associate Professor in the University of Calcutta. His research interests include network security, sensor network, routing, clustering, image processing, graph theory, distributed computing, cloud computing, database security, DNA computing etc. He has published many research papers in international journals and conference.