Ad-hoc Network Secured via Trust Computing

Rashmi Singh
M.Tech [C.S] (Pursuing)

Sudhir Dawra, PhD
Associate Professo
Dept. CSE [C.S]

ABSTRACT

Ad hoc network consist of a collection of wireless nodes, all of which may be mobile, dynamically creates a wireless network topology amongst them without using any fixed infrastructure. Nodes communicate with each other by passing data and control packet from one node to another. The execution and survival of an Ad-hoc network is rely upon the co-operative and trusting nature of its nodes. Ad-hoc network are vulnerable to passive and active attacks by malicious nodes due to the independent movement of nodes. Several protocols have been developed to secure Ad-hoc networks using cryptographic schemes, but all dependent on the existence of central trust authority. The presence of central trust authority is an impractical requirement for Ad-hoc networks, so in this Paper we present a trust model that doesn't rely on central trust authority. In our model we make use of trust agents that reside on network nodes. Each agent operates independently and maintains its individual trust value. An agent gathers data from all events & assigns weights to each event and computes different trust levels based upon them. Each trust agent basically performs the three functions: Trust Derivation, Quantification, and Computation.

Keywords

AODV, Trust, Security, Ad-hoc, Networks, Protocols

1. INTRODUCTION

In Latin, '*ad hoc*' phrase means '*for this*', meaning '*for this special purpose only*', by expansion it is a special network for a particular application (in military, emergency and relief scenarios where, in spite of nonexistent infrastructure, a network can be established). Routing protocols in Ad-hoc network play an important role in the creation and maintenance of links between nodes. Since Ad-hoc routing is a cooperative process where route information is relayed between nodes; any secure routing mechanism must evaluate the trustworthiness of other nodes. A number of such protocols were developed to secure the routing process. All the protocols just gave the assurance of either the presence of 100% security or its absence. None of these had an intermediate level of security protection, because all are depend on the presence of central trust authority. This paper is focused on introducing a trust model suitable for application to ad-hoc networks. The rest of the paper is organized as follows. In Section 2 we discuss specific attacks against AODV implemented Ad-hoc network routing. In Section 3 we describe some relevant previous work that motivates our research. In Section 4 we describe our proposed trust computing algorithm and its application to the Ad-hoc On Demand Distance Vector Routing (AODV) protocol. An analysis & simulation result of the proposed model is presented in Section 5, followed by the conclusion in Section 6.

2. ATTACKS AGAINST AODV IMPLEMENTED AD-HOC NETWORKS

Ad Hoc on demand distance vector (AODV) implemented networks are subjected to two main kinds of attacks, passive attacks and active attacks. Passive attacks are those, wherein the attacker aims to obtain information that is in transit. A passive routing attack does not disrupt the operation of a routing protocol. Active attacks are based on modification of the original message in some manner. Other advanced routing attacks have been identified. The Black hole, Gray hole, Wormhole and Routing table overflow attacks are the typical examples. The **Black hole attack** has two properties. First, the node exploits the routing protocol, such as AODV, to advertise itself as having a valid route to a destination node, even though the route is false, with the intention of intercepting packets. Second, the attacker consumes the intercepted packets without any forwarding. In **Gray hole** an attacker forwards all RREQs and RREPs but forwards only a few data packets dropping all other data packets. In **Wormhole** an attacker records packets at one location in the network and tunnels them to another location. Routing can be disrupted when routing control messages are tunneled. In **Routing table overflow** a malicious node, by generating route requests to several non-existent destinations, causes other nodes to create several entries in their routing table, one for each desired (non-existent) destination to keep the address of the sender in the precursor field so that it can transmit RREP or RERR back to the originator, and leads to the overflow of their routing table.

3. PREVIOUS WORK

The work on trust computations can be broadly classified into the following categories:

- Distributed trust computations: Every node computes its own value of trust on its neighbor.
- Centralized trust computations: Central agent manages/helps the node in trust computations.

Table 3.1 Comparison of Different Direct Trust Computations Mechanisms

Authors and Year	Context in use	Trust and performance metrics	Performance and limitations
M. J. Probstet.al, 2007 [4]	Based on observing the neighbors Behavior over the time.	Trust is a fractional value in [0, 1]. Convergence Time, memory cache requirements are analyzed.	Trust computation is completely local and biased.

A. A. Pirzada et. al, 2006 [5]	Routing based direct trust calculations.	Trust is a fractional value in [0, 1]. Performance of AODV and DSR protocol have been analyzed with the proposed trust scheme.	Specific to routing. Nodes should monitor neighbors all the time to construct and update trust relations. Computed trust is biased.

The advantage of [4] is that it accumulates the past behaviours and weighs them based on time. Hence the trust computation is precise. No single point failure. The advantage of [5] is that it Works based on existing request and acknowledgement schemes in AODV and OLSR protocols.

Table 3.2 Comparison of Different Hybrid Trust Computations Mechanisms

Authors and Year	**Context in use**	**Trust and performance metrics**	**Performance and limitations**
L. Xiong et. al, 2004 [6]	Based on feedback recommendation and own evaluations in P2P network.	Trust is measured in [0, 1]. Transaction success rate and malicious node detection rate are used as performance metrics.	The feedback can be represented only in binaries 0 or 1. Hence the feedback recommendations may not be accurate.
P. B. Velloso et. al, 2010 **[7]**	Based on Recommendation aggregation and also neighbor sensing.	Trust is measured in [0, 1]. Trust convergence and asymptotic error behaviour are analyzed.	This approach will be ineffective in spare networks

The advantage of [6] is that Feedbacks are weighted based on credibility factors and also community context is taken into account. This can provide accurate results. The advantage of [7] is that the recommendation aggregations and combining the recommendations with self measurement can increase the trust accuracy.

Table 3.3 Comparison of Different Recommendation Based Trust Computations Mechanisms

Authors and Year	**Context in use**	**Trust and performance metrics**	**Performance and limitations**
T. Jiang, 2006 G. Theodorakopoulos, 2006 [8]	Based on local voting.	Trust is measured in [−1, 1]. Bad nodes recognition rate is used as performance metric.	It does not consider the historical behavior of nodes.
Z. Liu et. al, 2004 [9]	Trust evaluation based on controlled flooding recommendations	Trust is measured in [0, 1].	The convergence time in trust computations and readjustments are high.

Table 3.4 Comparison of Different Centralized Trust Computing Mechanisms

Authors and Year	**Context in use**	**Trust and performance metrics**	**Performance and limitations**
S. S. Park et. al 2008 [10]	Clustering based trust computations	Trust is measured in the interval [0, 1] using Beta distribution.	The computed trust may not be precise with respect to single particular node. Cluster head can be single point of failure
A. Boukerche et. al 2008, Y. Ren et. al 2008 [11]	Nodes query the agents for the initial trust and then calculates the final trust value based on averaging.	Trust is defined in the interval [0, 1]. Malicious node handling, security over head and community sizes have been analyzed	This scheme will perform well as long as number of reputation agents are high
R. A. Shaikh et. al 2006 [12]	Cluster head aggregates the trust reports received from individual nodes and determines the final trust.	Trust is presented as fuzzy logic in the intervals $\{0-0.4, 0.4-0.6, 0.6-1\}$. Memory requirements have been analyzed	Cluster head can be single point of failure

4. PROPOSED TRUST COMPUTING ALGORITHM AND ITS APPLICATION TO THE AODV

```
/* Algorithm is divided into 3 function. Trust Derivation, Trust Quantification and Trust Computation.*/
// initialization
Rq= Route Request;
Rp=Route Reply;
Re=Route Error;
```

D=Data;

R_{qs} = Route Request Success =0;

R_{ps} = Route Reply Success =0;

R_{es} = Route Error Success=0;

D_s = Data Success=0;

R_{qf} = Route Request Failure=0;

R_{pf} = Route Reply Failure=0;

R_{ef} = Route Error Failure=0;

D_f = Data Failure=0;

P_A = Passive Acknowledgment; /* Trust Category 1. In this method the sender node places itself in promiscuous mode after the transmission of any packet so as to overhear the retransmission by the recipient node. */

P_P = Packet Precision; /* Trust Category 2. The accuracy of received data and routing packets offers a measure to compute trust levels.*/

S = Salvaging Route Error; // Trust Category 3.

S_s = Salvaging Route Error Success=0;

S_f = Salvaging Route Error Failure=0;

W= Weight assigned to the event;

T*x*(*y*) = Trust T in node *y* by node *x*;

T*n* = The situational trust T*n* in node *n*;

```
// Start monitoring the event
Trust Derivation () {
    while (simulation doesn't end) {
        $P_A$(){                    // Trust Category 1
            if(Rq= = success ) then $R_{qs}$ ++;
            else  $R_{qf}$ ++;
            if(Rp = = success ) then $R_{ps}$ ++;
            else  $R_{pf}$ ++;
            if(Re = = success ) then $R_{es}$ ++;
            else  $R_{ef}$ ++;
            if(D= = success ) then $D_s$ ++;
            else  $D_f$ ++;
        }
        $P_P$ (){                   // Trust Category 2
            if(Rq= = success ) then $R_{qs}$ ++;
            else  $R_{qf}$ ++;
            if(Rp = = success ) then $R_{ps}$ ++;
            else  $R_{pf}$ ++;
            if(Re = = success ) then $R_{es}$ ++;
            else  $R_{ef}$ ++;
            if(D= = success ) then $D_s$ ++;
            else  $D_f$ ++;
        }
        S(){                        // Trust Category 3
            if(S= = success ) then $S_s$ ++;
            else  $S_f$ ++;
        }
    }
}
Trust Quantification () {
    T$n$ ($P_A$) = W(Rq)* Rq + W(Rp)* Rp +W(Re)* Re + W(D)*D;
    //Trust Category 1
    // where
    Rq= $R_{qs}$- $R_{qf}$ / $R_{qs}$+ $R_{qf}$ ;   for  $R_{qs}$+ $R_{qf}$ $\neq$0; else Rq=0;
    Rp = $R_{ps}$ - $R_{pf}$ / $R_{ps}$ + $R_{pf}$ ; for $R_{ps}$ + $R_{pf}$ $\neq$0; else Rp =0;
    Re = $R_{es}$ - $R_{ef}$ / $R_{es}$ + $R_{ef}$ ; for $R_{es}$ + $R_{ef}$ $\neq$0; else Re =0;
    D= $D_s$ - $D_f$ / $D_s$ + $D_f$ ;   for  $D_s$ + $D_f$ $\neq$0; else D=0;

    T$n$ ($P_P$) = W(Rq)* Rq + W(Rp)* Rp +W(Re)* Re + W(D)*D;
    //Trust Category 2
    T$n$ (S) = W(S)* S   //Trust Category 3
    //where
        S= $S_s$ - $S_f$ / $S_s$ + $S_f$; for $S_s$ + $S_f$ $\neq$0; else S=0;
}
Trust Computation() {

    T$x$($y$)=$W_x$($P_A$)*T$x$($P_A$)+$W_x$($P_P$)*T$x$($P_P$)+$W_x$ (S)*T$x$(S);
    if(T$x$($y$) == -1) then complete distrust;
        if(T$x$($y$) == 0) then non contributing event or uncertain;
        if(T$x$($y$) == 1) then complete trust;
        if(-1 <T$x$($y$) <0) then node considered as malicious;
        if(0<T$x$($y$)<0.5) then wait for some more event to occur;
        if(0.5<T$x$($y$)<1) then node may be considered as reliable;
        }
    }
```

5. SIMULATION SETUP AND RESULT ANALYSIS

A simulation environment for Ad-hoc network is developed to evaluate the performance of the AODV and TAODV (Trusted-AODV) protocol. Both the protocols were simulated over this environment and its performance was studied for various parameters. Our algorithm is implemented by modifying the original AODV source code in NS-2.

Table 5.1 Simulation Parameters

S. No.	Simulation Parameters	Values
1	Simulator Used	Network Simulator (version 2.31)
2	Number of Nodes	30
3	Total number of Faulty nodes(Black hole)	1
4	Transmission range	200m
5	Area Size	800 x 800
6	MAC	802.11
7	Simulation Time	100Secs
8	Packet Size	404 bytes
9	Propagation Model	Two ray ground model
10	Speed	20m/s

5.1 Packet Delivery Ratio

As shown in fig 5.1 we can clearly analyst that in the ideal case of the AODV the Packet delivery ratio is 95,000, but in the case of Black hole it become approximately 1900 only and in the case of the TRUSTEDODV the Packet delivery ratio increase to 90,000. The following result shows that TRUSTEDAODV which we have implemented gives better result than AODV and Black hole-AODV.

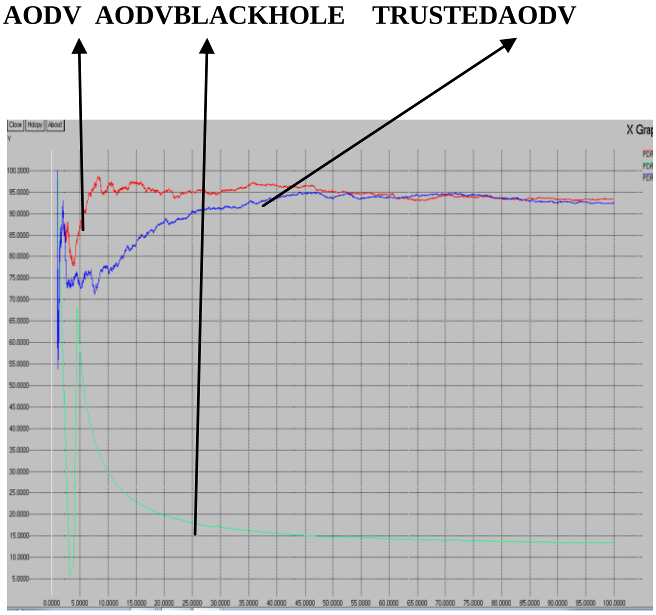
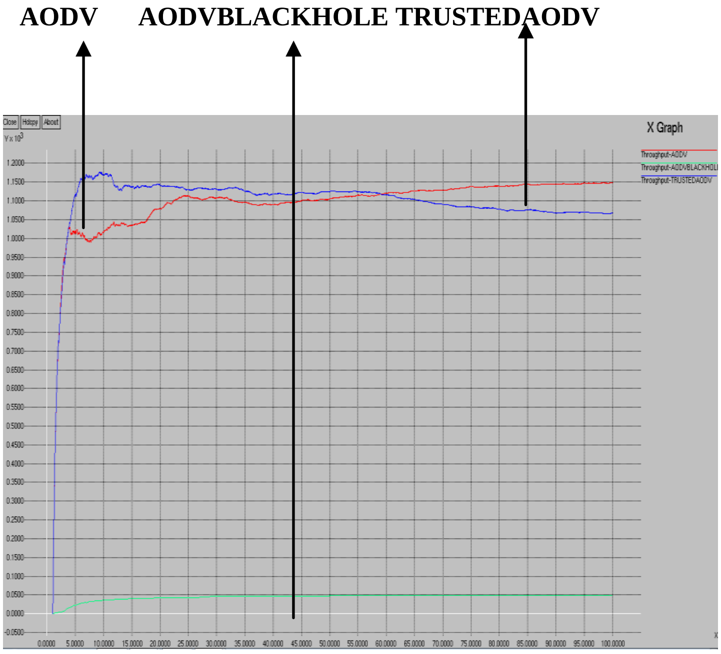


Fig 5.1 Comparison graph for PDR between AODV, AODVBLACKHOLE & TRUSTEDAODV

5.2 Throughput

Fig 5.2 shows the comparison graphs of throughput for AODV AODVBLACKHOLE & TAODV which we had implemented. We can clearly analyze that TAODV gives much batter result as compare to the AODV AODVBLACKHOLE.



5.2 Comparison graph for Throughput between AODV, AODVBLACKHOLE & TRUSTEDAODV

6. CONCLUSION

In this paper we have implemented a Trusted AODV that establishes and manages trust in pure Ad-hoc network. By means of pure Ad-hoc network we want to say that the Ad-hoc network that doesn’t rely on presence of central trust authority. AODV does not specify any special security measures. The proposed protocol, TAODV would be considered to enhance the security requirements of AODV. Thus the application area of TAODV includes where the secure communication among the mobile nodes is crucial. Our protocol is best suited to emergency ad hoc networks like in military, emergency and relief scenarios where, in spite of nonexistent infrastructure, a network can be established.

7. REFERENCES

[1] Z. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Network **13** (1999), no. 6, 24–30.

[2] L. Zhou and Z. Haas, Securing Ad Hoc Networks, *IEEE Network Magazine*, Nov. 1999.

[3] Asad Amir Pirzada and Chris McDonald, ”Establishing Trust In Pure Ad-hoc Networks”

[4] M. J. Probst and S. K. Kasera, “Statistical trust establishment in wireless sensor networks,” in *Proceedings of the 13th International Conference on Parallel and Distributed Systems*, pp. 1–8, 2007.

[5] A. A. Pirzada and C. McDonald, “Trust establishment in pure ad-hoc networks,” *Wireless Personal Communications*, vol. 37(1-2), pp. 139–168, 2006.

[6] L. Xiong and L. Liu, “PeerTrust: Supporting reputation-based trust in peer-to-peer communities,” *IEEE Transactions on Knowledge and Data Engineering, Special Issue on Peer-to-Peer Based Data Management*, vol. 16, no. 7, pp. 843–857, July 2004.

[7] P. B. Velloso, R. P. Laufer, D. O. Cunha, O. C. M. B. Duarte and G. Pujolle1, “Trust management in mobile ad hoc networks using a scalable maturity-based model,”

IEEE Trans. Netw. Service Manag., vol. 7, no. 3, pp. 172–185, Sep. 2010.

[8] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 318–328, February 2006.

[9] Z. Liu, A. W. Joy, and R. A. Thompson, "A dynamic trust model for mobile ad hoc networks," in *IEEE International Workshop on Future Trends of Distributed Computing Systems, FTDCS'04*, pp. 80–85, May 2004.

[10] S. S. Park, J. H. Lee, and T. M. Chung, "Cluster-based trust model against attacks in ad-hoc networks," in *Third International Conference on Convergence and Hybrid Information Technology*, pp. 526–532, 2008.

[11] Y. Ren and A. Boukerche, "Modeling and managing the trust for wireless and mobile ad hoc networks," in *IEEE International Conference on Communications, ICC '08*, pp. 2129 – 2133, 19-23 May 2008.

[12] R. A. Shaikh, H. Jameel, S. Lee, S. Rajput and Y. J. Song, "Trust management problem in distributed wireless sensor networks," in *12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 411–414, 2006.