ASMA : Towards Adaptive Secured Multipath in MANETs

Vincent Toubiana and Houda Labiod
Ecole Nationale Supérieure des Télécommunication (ENST)
LTCI-UMR 5141 CNRS
GET/ENST/INFRES Department
46 rue Barrault – 75634 Paris Cedex 13 – France
Email : labiod@enst.fr, toubiana@enst.fr

Abstract. As they are used to create open communities, Mobile Ad hoc NETworks (MANETs) are not favourable environments to establish trust, which is necessary to provide security. Multipath routing mechanisms within infrastructureless networks environment seems appropriate and useful to enhance security protection. In fact, the level of trust can be increased so as many of potential security attacks are detected, revealed and stopped. Nevertheless an excessive control overhead is always generated. In this paper, we propose a global framework that integrates a set of concepts and mechanisms aiming at enhancing security in highly dynamic decentralized ad hoc networks. Our solution focuses on authentication, routing securing, trust management with reliable estimation of trust. A large panoply of attacks are prevented using our various mechanisms.

1 Introduction

Ad hoc networks rely on peer to peer architecture and collaboration between nodes to provide connectivity. Typically routing protocols proposed for these networks assume that every node collaborates and no node try to disturb the network. However, as this technology evolves rapidly and will be soon intensively deployed, threats have to be considered because ad hoc networks can be easily attacked by hackers and spammers. Furthermore being totally distributed, highly connected, open and implicit, MANETs can be used to spread virus. Remembering the Bluetooth virus appeared at the athletic games in Helsinki, it shows clearly that users are not aware of potential threat. Now imagine how fast the virus could spread in an ad hoc network and what could be the damage. Many attacks have already been considered for ad hoc networks and we can expect that a lot of specific attacks for future applications will be discovered whenever these new networks will be used. As these

networks provide no authentication and so no traceability, hackers could act easily without being detected. Actually, in wireless networks users must refer to peers to route and transmit packets, hence they have to establish trust relationship to operate in a secure context. To define trust, we refer to the definition of D. Gambetta in [1] : *" ...trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action"*.

To get feedback for the first time about a node, we distinguish two cases, 1) if we already know an agent we could use this agent to get feedback about the submitted action 2) we do not know an agent, so we could submit action which does not need an explicit feedback (i.e. : trying to reach a node that does not exist). Once we obtain a feedback about nodes, we can refer to them to estimate trust about other nodes, hence we refer to their reputation, which is defined by [1] as the *"perception that an agent creates through past actions about intentions and norms"*. This relation concerns only one agent and is defined as an aggregation of information collected by nodes on past interactions.

Based on these key concepts, we propose a global framework in order to meet security requirements through different functions such as authentication, efficient and low-cost multipath routing and trust management. Trust estimation is performed by close nodes and can be computed for either a node or a path, to maintain up-to-date trust information and to prevent gathering useless information. This paper is organized as follows. Section II gives a review on the security mechanisms proposed for ad hoc networks including trust management, authentication and routing protocols. Section III provides a description of our proposed framework. Section IV analyzes the robustness of our framework against some famous attacks. Finally, section V provides concluding remarks and highlights our future work.

2 Related Work

Security in MANETs is a critical research topic ; an intense research activity is undertaken. One of the major challenges of ad hoc networks is the authentication of the nodes as we may have to authenticate unknown nodes. Authentication can simply be provided by using a trusted third party common to all the nodes of the network. If nodes have different certification authorities, we have to build a PKI (Public Key Infrastructure). In a centralized mode, we use a unique authority of certification which is totally trusted, however if the authority is far this can create a high latency to get a certificate and the network comprises a single point of failure. To distribute the authority, one solution consists of using threshold cryptography[2] . Gathering at least "t" signatures of different certifications authorities is necessary to generate a certificate, hence the system is more distributed and robust to attackers, but this creates the problem to find the trusted entity and how to fix the threshold value to provide efficient security and low delay. To provide authentication and trust we can

use the "Small World Theory" as proposed in PGP[3], however such model may generate a high overhead due to certification operations and in practice the size of the connected graph of trust remains small compared to the size of the whole network [4]. So, the probability to get the wanted certificate through a chain is very low. One solution to create more certification links may be to use the mechanism of [5] to distribute certificates in the network, however it is mainly based on the existence of an on-line certification authority ; once the authority is off-line nodes can not know really who to trust. Authentication can also be based on the identity mechanisms; this is the principle of encryption based identity. One of them is proposed in SUCV (Statistically Unique and Cryptographically Verifiable) [6], but the principal threat for this scheme is that anyone can create its own identity and so there is no way to know if an identity is legitimate or not. Consequently, this authentication is highly vulnerable to Sybil attacks [7]. And even if the problem of binding identity and key is solved, authentication is not a problem anymore ; identification becomes the real key challenge. A hybrid solution has been proposed as the composite key management [8] protocol which uses the trusted certification authority and PGP-like chain to provide short chain of trust in the network. However this solution does not clearly define which node takes part of the certificate chaining and this may create a high overhead as all the certification chain may remain long.

Once they are authenticated, nodes use secured routing protocols to communicate. These protocols often rely on powerful cryptography algorithms. ARIADNE[9] provides end-to-end security for both routing and forwarding ; however it relies on TESLA and symmetric cryptography. The mechanism of key distribution is not clearly defined in the protocol. ARAN[10] uses asymmetric cryptography for hop-by-hop encryption, but such mechanism assumes that the source trusts every node in the route because if there is at least one untrusted node, the whole route is untrusted. Recently some proposal use multipath routing [11] such as SecMR (Secure Multipath Routing) and the Secure MultiPath Protocol. Both provide security through redundancy : SecMR uses list in the routing packet to prevent node from being in two paths and Secure MultiPath Protocol uses information added to the RREQ (Route REQuest) by intermediate nodes to select the paths. As they do not refer to a pre-established trust, they have to use totally distinct paths resulting in high overload of the network and resource consumption ; these solutions are not scalable.

Obviously, security in ad hoc networks is a young research domain, no standard has been adopted yet, many issues have to be addressed and more studies are needed. Authentication and routing are tightly bound and creates a deadlock situation. However, only SUCV clearly define both and others suppose that there is a bootstrapping phase which avoids the problem. Moreover, most of the proposed solutions are not adaptive and so highly vulnerable to especially Denial of Service (DoS) attacks. In this paper, we propose a complete framework named Adaptive Secure Multipath for Ad hoc Networks(ASMA) which deals with necessary functions to resolve the critical security issues. We propose an approach which can be seen as a suitable candidate to make a balance between security requirements and system flexibility in the case of highly dynamic ad hoc networks. As it is based on a

dynamic trust, this framework provides security without requiring a bootstrapping phase.

3 ASMA

This section gives an overview of the framework.

3.1 Definition

Without loss of generality, we define our functions in the [0;1] interval. Doing so, we can use probability and fuzzy logic theory [12].

a) Trust. Our aim is to use a trust model which is coherent and computable. Then we refer to the definition of [1] to propose the trust function : *Trust(A,B,F,C)* is the trust that node *A* has in an agent *B* to perform an action *F* in a context *C*. Most of the definitions consider the agent *B* as a single node but we adopt a more general concept, as we use multipath routing, we consider that agent *B* is either a node or a path. As an attacker could act honestly as long as it knows we have a feedback and become malicious when it knows it is not observed. So a node should not be able to know if we have a feedback and this is an important part of the context which relies on multipath capabilities. Other parameters of the context include density of the network. We suppose that the trust function is continue for *F* and *C*, so we could estimate the trust of a node for a particular value *(F,C)* knowing the trust of the same node for *(F',C')*. Trust is composed of two values, the measured trust (*Mtrust*) and the reputation trust (*Rep*). The trust is a weighted average of these two values : with *a* in [0;1]. *Mtrust* is computed as the number of successful actions divided by the number of submitted actions. It is clear that the number of actions submitted is an integer but the number of successful actions may be a float as some actions may not totally succeed or failed (for example recommendation). This definition requires considering only action on which we received a feedback. In order to be more flexible to recent changes, we can give a higher weight to recent actions.

b) Risk. We define risk of an action as the minimum of trust that we can accept to perform this action. Risk depends on the function, the user and the context of the action. Risk is defined by the function : *R(A,F,C'); C'* can be different of the context define for trust. Since *A* has all the knowledge to estimate the risk, we do not require the function to be continue for *F* and *C'*. Risk must be higher than 0,5. We settle that *A* trust *B*, and so that *B* is trusted by *A,* for the function *F* in the context *C* if : *Trust(A,B,F,C) > R(A,F,C').*

c) Knowledge. In our mechanisms we use only asymmetric cryptography and certificate to bind entity to key. The certification management is symmetric and will be explained for centralized and distributed networks. Here we define the

"knowledge relation" as follow : *node A knows a node B if A and B have exchanged there certificates*. This relation is symmetric as in [5]: if *A* knows *B* then *B* knows *A*. The relation is not extended to be transitive because if a node forges many identities it could use this transitivity to know the entire network. We also define *K(X)* as the set of nodes which know *X*.

d) Reputation. Parameters of both trust and reputation are the same, however as reputation may be an aggregation of values, it can take less arguments. So we define the global reputation of node *B* for a function *F* as : $Rep(B,F) = \frac{1}{card(Net)} \sum_{A \in \text{Net}} \sum_{C \in S} Trust \frac{(A,B,F,C)}{card(S)}$. Where *Net* represents the set of nodes in the network and *S* the set of all the contexts. When we use recommendation from an other node, we use the trust we allow to the node which sign the recommendation. So the perceived reputation is the product of the recommendation and the trust we have in the recommender for recommendations. For node *A* the reputation of *B* for action *F* in context *C* is defined as : $Rep(A,B,F,C) = \frac{1}{card(Known(A))} \sum_{B \in \text{Known}(A)} Trust(A,D,Rec,C) \times Trust(B,D,F,C)$; C' is the common context between *D* and *A*. From this value we can sum the reputation on different contexts to get an aggregate reputation.

3.2 Different Procedures of the Framework

We focus our work on on-demand routing approach. We based our routing protocol on DSR (Dynamic Source Routing) and just few packet extensions are needed to propagate "trust information". When a node wants to establish a communication, it first applies Multipath Key Management to get the certificate of the destination, then it executes the Multipath Adaptive Routing Protocol to find the route it will use. Finally, it uses the forwarding mechanism to send packets.

The Multipath Key Management : This solution is derived from composite key management which provides certificate chaining using graph of trust [8]. However this mechanism creates a high load due to the certificates chain when the destination is far (more than two hops). But in our solutions we assume that some known nodes get the certificate that we are looking for, then they can sign it and send it to us. So instead of starting a chain from the source to the destination, we create a chain from the destination to the first node which knows the source. Known nodes receiving the request may reply if they know the destination too. We also improve the reliability of the received certificates using multipath : in case we receive multiple responses, we use Dempster-Shafer [12] theory to compute the trust associated to the certificate. As nodes move and because we establish communication with different nodes, the set of nodes known by a node will be distributed among the network and the probability is low for two nodes to get the same set of known nodes.

Multipath Adaptive Routing protocol : Multipath routing provides redundancy and robustness, but implies drawbacks like increased load of the network and resource consumption which may be used to generate DoS. However when a node just arrives in the network, it knows no node and so can not have security requirement, hence multipath is a good solution to establish trust relations. Consequently we use adaptive multipath routing to allow new nodes to use multipath routing so they have good requirement on security. Hence nodes store feedback of different paths and then establish trust relation with other nodes in the network. Once the node has established trust relation, it decreases the number of paths it uses to get higher priority and slower delay. When we trust some intermediate nodes, we do not have to use totally disjoint paths, as this may not be possible, and may generates high load. It is enough to have just disjointed paths between trusted nodes. To prevent excessive flood of request, node monitor the number of RREQ they forward. When an intermediate node forwards too many RREQ, it stops to forward RREQ unless they are signed by a trusted node, thus preventing DoS. We define the macro graph *MG(S,D)* as the oriented graph connecting *S* to *D* which vertexes are nodes in *K(S)* and an edge is the set of paths between two vertexes. The weight of an edge is the trust associated to it by its input vertex. If an edge contains multiple paths, nodes use Dempster-Shafer theory to compute the weight. Similarly to reactive approach, the route request phase remains the same. Modifications concern reply phase and forwarding.

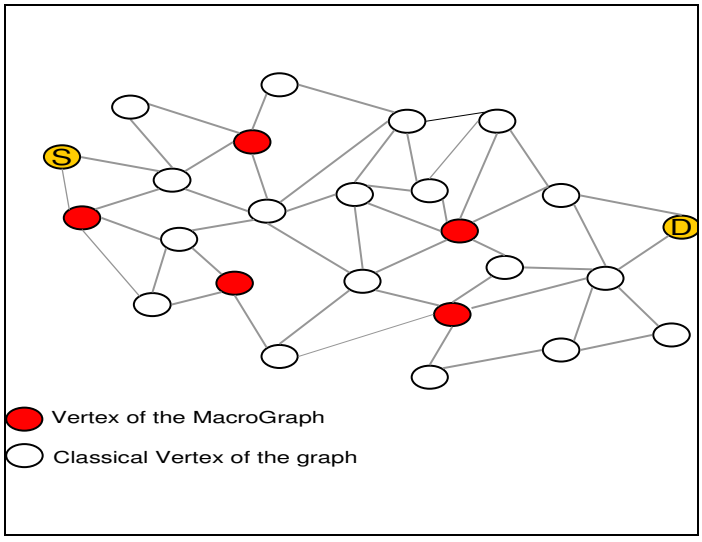

Fig. 1. Example of MacroGraph

Multipath response flooding : Before sending a RREQ, the source *S* estimates the risk *R* it can accept for the transaction knowing the context of the action. Then it

sends the RREQ which includes *R*, a special byte to notify that the RREP (Route REPly) has to use multiple paths and add a list of nodes that can not be neighbor in the macro graph and a list of nodes that can not be on the macro graph, then it signs and broadcast the RREQ. Intermediate nodes known by the destination *D* append a signature of the packet in an extended destination list (this provide integrity of the route). Intermediate nodes in *K(S)* and not blacklisted indicate that they forward the RREQ in the extended source list. Receiving the RREQ packet, *D* checks signatures of known nodes. If it has up-to-date information about nodes in the extended source list, it appends them in a reputation extension packet. Then it waits for a random timer *t1* in $\left[\frac{R \times T \times HopCount}{2}; R \times T \times HopCount\right]$ where *T* is adapted to the network and HopCount is the average number of hop records in RREQ. Once *t1* expired, it signs RREP and sends it through all paths which have propagated the RREQ. Intermediate nodes do the same using a timer *t2* in $[0; R \times T \times HopCount]$ and just nodes in *K(S)* sign the RREP. Intermediate nodes in *K(S)* check the RREP to find other nodes in *K(S)*, then they remove from the RREP all nodes which are not in *K(S)* and just indicate the trust allowed to path to their neighbor in *MG(S,D).* This prevent useless overhead as other nodes are not concerned by these nodes and allow trusted nodes to manage paths between them without referring to *S*. If a node is connected to nodes in *K(S)* through many paths (the exact value depend on the density of the network) and is not the direct neighbor, it acts as a known node and requests an exchange of certificates with *S*.

We propose an extension to limit the multipath flooding. If *S* already knows a lot of nodes, it does not have to ask for flooding RREP and so can reach lower delay reducing the number of paths it uses. The source proceeds as previously but does not indicate that the RREQ must be flooded. Thus instead of sending the RREP by all the routes it received a RREQ, intermediate nodes always use the shortest path (or the trustiest, the metric can be defined by the user).

Forwarding adaptation : Upon receiving the packet, the source computes *MG(S,D)* to achieve the required trust level (which may be higher than R but must correspond to the value returned by known nodes). Thanks to the concept of dynamic trust, for non critical packets, the source could select nodes which have been malicious and may now act honestly, offering them a chance to be trusted again. Then *S* sends a packet describing *MG(S,D)* using the list of vertexes and edges. Every node in *MG(S,D)* use the information it collected during the routing phase to achieve the security requirements of *S*. An advantage of such mechanism is that if a node changes often its identity, it will not know a lot of nodes and so can not require trust.

For the following packets, *S* does not have to include the list of trusted nodes, but just the list of edges and their weights, thus we reduce considerably the size of the header. As every trusted node knows the required weight for the edge, it can adapt the transmission of packets. During communications *S* gets feedback about paths and nodes, it can adapt the number of paths it uses to reach the destination. As this is done dynamically, the source can easily adapt security and/or QoS requirement. When a node does not receive packet for a long time, it simply removes the recorded path corresponding to the route. Updating trust is a problem when two different

nodes declare that the other one is malicious and that we have neither feedback nor reputation to know which one is the liar. Thus we may try to obtain reputation about one of the nodes and so trust the one which has the best reputation. If there is no way to get feedback by different paths (and so to solve the problem), we can not know which node is lying, so these two nodes can not be neighbor on the macro graph anymore. For next RREQ, we add their names in the neighbor blacklist (two nodes in the list can not be neighbor). Afterwards if feedback is obtained about one of them, trust adaptation is applied as previously described. To get feedback, source requires that trusted nodes send cumulative acknowledgment. We also can use end-to-end acknowledgment to get a global feedback, then the destination signs the feedback and sends it by a trusted path. In case of link breakage detection, node signs the RERR (Route ERRor). If the upstream trusted node does not know the signature, it tries to recover it through another path. Otherwise, it sends the error to the source and signs it.

3.3 Operations

The framework ASMA goes through several steps :

- Initialization of nodes,
- Exchange of certificates before a communication,
- Route and trust establishment,
- Forwarding and route adaptation.

To illustrate the operation of ASMA, we give two different scenarios associated to the centralized and the distributed modes.

Centralized Mode : We suppose that a Certification Authority (CA) is present in the visited network. It can be either centralized or distributed among some nodes.
We distinguish four steps :
A) The node *S* broadcasts a Certification Request (CReq) which contains a sequence number and its public key (PK). Only nodes between *S* and the CA forward it.
B) The responding CA replies by sending a certificate of the public key. It adds a list of nodes' certificates which have forwarded the CReq and then sends it by the same path it received it. When a node forwards this packet, if its certificate is in the list, it stores the certificate of the requester and considers it as known.
C) In the case of reception of the same certificate, the source records it. Otherwise it considers the certificate with the best reputation. In both cases it associates to the certificate an “unknown value”.
D) When interacting with an authenticated peer, the node asks for confirmation the CA's certificate. If it is confirmed, it associates to all the corresponding certificates the trust of the confirming peer. Otherwise, the node has to send the real certificate of the CA and to sign it with its own key (for non repudiation); it must have received it from what it claims to be the real authority. The source then asks other nodes to confirm the certificate. Then it will associate the certificate with the trust corresponding to the node that established the certificate.

Distributed Mode : A node connecting for the first time to the network broadcasts its PK. A forwarding node records the key and generates a certificate with probability *P(Cert,Path)* which depends on the number of distinct certificates already recorded and the number of path by which the node received the key. When it records a PK, the node signs it and sends it as a certificate. It then sends its own PK to be signed and returned by the requesting node. In that case, unless there is at least one trusted path, a malicious node could use a Man in The Middle attack to get certificates. As previously, the node associates the "unknown value" to this certificate. Later, during a communication with a node, we can get its certificates with trust value higher than an associated threshold.

4 Analyzing Robustness Against Attacks

This gives a good overview of the capabilities of the protocol to resist to attacks. For every attack, we consider that at least one path from the destination to the source is trustful, even if this is a strong requirement, it is necessary and there can not be network communication without this assumption.

4.1 Passive Attacks

Traffic Analyzes : Multipath routing approach make it very difficult for an attacker to guess where is the expected destination of packet and what are the critical paths. This still possible, but this kind of threat is not dangerous in domestic networks and is treated in military networks.
Information Leak : The information leak depends on the captured packet. If the first packet is captured then the information leak is critical, but if it concerns another packet, the leak is small as the header just contains source and destination information. We plan to reduce this leak using address substitution in future work.

4.2 Active Attacks

Wormhole Attack : If a wormhole exists between two nodes, the feedback about these nodes will remove the route from the macro-graph.
Byzantine Attack : If the Byzantine node is a trusted node we must refer to another path of the macro graph to prevent the attack. If the attack is performed by an untrusted node, there is a high probability of detection by a trusted node and the impact of the attack is low because the packet is routed by another path.
Resource consumption : If a node uses different identities during flooding its packet will be dropped. However a node could consume resource on a path, pretending forwarding traffic. For example the node may use two identities and pretends forwarding packets from the first identity to the second one using a loop. Nodes on the loop are not supposed to know neither the source nor the destination, hence they have no reason to not forwarding. To prevent this attack, intermediate nodes could

ask for the source's certificate when they suspect this attack. If nodes can forge identities, there is no other solution than to limit the number of forwarded packets for nodes that we do not know.

Spoofing : Although forging identities is penalizing because using identities reduces the number of nodes known by a single identity, it can not be totally prevented. In distributed networks, a node can generate as many IP addresses that it wants, but in a centralized mode, this situation depends on the capacity of the server to check the identity of the node requesting for an IP address.

Denial Of Service : Using many identities, nodes can flood the node with RREQ thus generate DoS as some legitimates RREQ are dropped. However nodes which are trusted by a lot of nodes are not concerned by this attack as their RREQ are not dropped.

Routing Attack : If all routing attacks can not be prevented, at least their impacts are drastically reduced as they have to impact all the paths to achieve their goal. *Source route modification* can only be performed by nodes on paths between trusted nodes and will be detected by trusted nodes. *Rushing attack* is not efficient in multipath routing because nodes do not only consider the first arrived RREQ. *Packet replication* is prevented thanks to the use of sequence numbers and signatures by the source. As we use DSR as base routing protocol, we can use its improvement on cache utilization. However the best way to secure DSR is to disable cache optimizations or to use it only if the packet is signed by a trusted node.

5 Conclusion

In this paper we give the main guidelines for designing a new framework to deal with security in MANETs. This framework called ASMA provides authentication and security routing using trust dynamic relations and based on multipath communication. We adopt an appropriate trust model suitable to our multipath reactive routing approach. Since dynamic trust is supported, no bootstrapping phase is needed. ASMA combines efficiently key management, routing and forwarding operations to securely transmit data packets through the network. Moreover, an other strength of our solution is that it can be used either in a centralized or a distributed mode. An analysis of the robustness of ASMA is illustrated by considering a large number of passive and active known attacks. For future work, we intend to complete and to evaluate the cited mechanisms.

Acknowledgement

This work was carried out through a France Telecom R&D Research Program. We would like to thank Laurent Reynaud for his helpful and very interesting comments.

6 References

1. D. Gambetta, "Can we trust trust?", Trust, Making and Breaking Cooperative Relations. basil Blackwell (1990) p. 213-237
2. J. Kong, P. Zerfos, H. Luo, S. LU and L. Zhang. "Providing robust and ubiquitous security support for mobile ad-hoc networks", in Proc. of the 9th IEEE International Conference on Network Protocols, 2001
3. P. Zimmermann. "The official PGP user's guide". MIT Press 1995
4. S. Capku, L. Buttyan, and J.-P Hubaux. "Small worlds in security systems: an analysis of the pgp certificate graph", in Proc. of the 2002 workshop on New security paradigms, 28-35, 2002
5. T. Li, Z. Wan , F. Bao , K. Ren, R. H. Deng, K. Kim, "Highly reliable trust establishment scheme in ad hoc networks",Computer Networks 45 (2004) : 687699
6. G. Montenegro and C. Castelluccia. "Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses". in Proc. of the Network and Distributed System Security Symposium (NDSS'02), 2002
7. J.R. Douceur," The Sybil Attack", in Proc. of the 1st International Workshop on Peer-to-Peer Systems, 2002
8. S. Yi, R. Kravets, "Composite Key Management For Ad Hoc Networks", in Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), 2004
9. Y.-C. Hu, A. Perrig, and D. B. Johnson. "Ariadne : A secure on-demand routing protocol for ad hoc networks". In MOBICOM, 2002.
10. K. Sanzgiri, D. LaFlamme, B. Dahill, B. Neil Levine, C. Shields, E. M. Belding-Royer, "Authenticated Routing for Ad Hoc Networks,IEEE Journal on Selected Areas in Communications, vol. 23", no. 3, march 2005
11. R. Mavropodi, P. Kotzanikolaou, C. Douligeris, "Performance Analysis of Secure Multipath Routing Protocols for Mobile Ad Hoc Networks", in Springer WWIC 2005, p.269-278, 2005
12. T. M. Chen, V. Venkataramanan, "Dempster-Shafer Theory for Intrusion Detection in Ad Hoc Networks", IEEE Internet Computing Volume: 9 Issue: 6 Date: Nov.-Dec. 2005