International Journal of Security and Its Applications
Vol.7, No.5 (2013), pp.321-342
http://dx.doi.org/10.14257/ijsia.2013.7.5.30

Trust-Based Clustering in Mobile Ad Hoc Networks: Challenges and Issues

Mehdi Maleknasab[1], Moazam Bidaki[2] and Ali Harounabadi[3]

[1]*Department of Computer Engineering, kohkiloye and BoirAhmad Sience and Research Branch, Islamic Azad University, Iran*
[2]*Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran*
[3]*Department of Computer Engineering, Islamic Azad University,Central Tehran Branch, Iran*
, [1]Maleknasab@Iausepidan.ac.ir, [2]Bidaki@Iau-neyshabur.ac.ir, [3]a.harounabadi@gmail.com

Abstract

Mobile ad hoc networks are prone to various security attacks of malicious nodes and attackers. To protect the network clustering from security attacks, numerous trust-based clustering schemes have been presented in the literature. Analyzing the existing trust-based clustering solutions, the researchers illustrated their primary features and properties in this paper and mainly discussed about the trust management mechanisms which are integrated in each trust-based clustering solution. Besides it was illuminated how trust and reputation are used in the cluster formation and maintenance phases. At the end, all of their trust computation methods were compared and concluded with open research issues.

Keywords: *Attack, Cluster Head, Direct Trust, Indirect Trust, Security*

1. Introduction

A Mobile ad hoc network or MANET is a collection of resource limited mobile nodes which does not rely on any fixed or centralized infrastructure. These nodes dynamically form a temporary network and communicate with each other through bandwidth limited and multi-hop wirelesses links [1]. From architectural point of view, MANETs can be classified into two types on networks: flat and hierarchical. All nodes have the same roles and responsibilities in the network of flat MANETs. They do not scale well but as the number of network nodes increases, the overheads of routing and other operations grow dramatically. Therefore, only small number of nodes and devices can be managed as flat MANETs. In order to support large number of devices, ad hoc networks should be organized hierarchically. Clustering organize the ad hoc networks hierarchically and create clusters of ad hoc nodes which are geographically adjacent. Each cluster is managed by a clusterhead(CH) and other nodes may act as cluster gateway or cluster member. Clusterheads nodes act as a local coordinator and perform intra-cluster transmission as well as other tasks such as data processing and forwarding operations. Since CHs need more resource than ordinary nodes, the heterogeneous MANETS use special nodes as CH, but in homogenous MANETs which all nodes have the same capabilities, any node can be selected as CH. In MANETs, all CHs are interconnected with each others to provide communication backbone to facilitate the data transmission between various network nodes. Also, inter-cluster communication is aided by

ISSN: 1738-9976 IJSIA

cluster gateways which have inter-cluster links and can direct data traffic between neighboring clusters [3]. It can be said that clustering is a very useful mechanism in managing various aspects of ad hoc networks and the following benefits can be specified for it in the literatures:

- Effective topology control.
- Spatial reuse of resources to increase the system capacity.
- Reduction of network traffic and communication overhead.
- Better coordination of transmission events.
- Reduced transmission collision.
- Better routing of data packets.
- energy saving
- Facilitating data aggregation.
- Increase of networks scalability, security and lifetime.

In [3] Yu *et al.*, studied and analyzed various clustering algorithms which are proposed for MANETs. In spite of numerous papers which studied the clustering in [3-11], security is one of the main items that is ignored in these surveys. Considering the vulnerability of MANETs to numerous passive and active security attacks, the clustering schemes can be classified as figure 1exhibits:

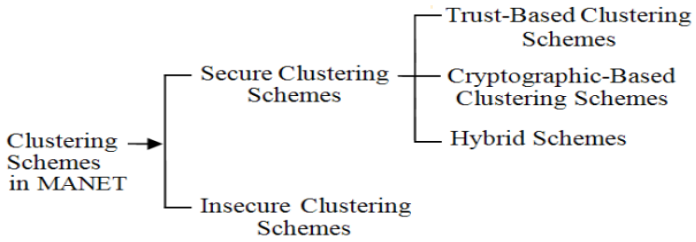

Figure 1. Classification of Clustering Schemes Form Security Perspective

As it is shown in the figure, secure clustering algorithms can be categorized as trust-based, cryptographic-based or hybrid schemes. Each of these categories protects the clustering process against special type of attackers but only hybrid scheme presents highest level of security. For example, although pure cryptographic–based clustering techniques increase the security of clustering operation against outsider and insider attackers, they are unable to detect the compromised nodes and insider attackers. To defend against insider malicious nodes, trust and reputation management systems should be used. Several general purpose reputation management systems have been proposed for MANETs in the literature but they have high overheads which decrease their effectiveness in the resource limited ad hoc networks. In this context, trust-based clustering algorithms integrate the trust management systems with the clustering algorithms and are aimed to reduce the overheads of reputation management. These schemes manage the trust related information for each node and prevent the election of a malicious or compromised node as the CHs or other cluster components. Hybrid trust-based clustering algorithms are complex security solutions that integrate the cryptography-based mechanisms and reputation management systems with clustering algorithms to defend against both internal and external attackers.

This paper presents a comprehensive analysis of well-known trust-based clustering schemes and illuminates their prominent features and capabilities. This study can be very

helpful to understand the limitations of existing solutions and design new and low overhead trust-based clustering schemes which can be resilient against internal misbehaving nodes as well as external attackers. Table 1 shows the acronyms and abbreviation used in this paper.

The rest of this paper is organized as follows: Section 2 discusses about the cryptographic-based clustering in mobile ad hoc Networks. Section 3 illustrates trust-based clustering schemes and cluster management operation of each scheme and determines their advantages and disadvantages in detail.

Table 1. Acronyms and Abbreviations

Acronym	Expansion
CH	Cluster Head
BS	Base Station
CA	Certificate Authority
DCA	Distributed Certificate Authority
PDCA	Partially Distributed Certificate Authority
FDCA	Fully Distributed Certificate Authority
DoS	Denial of Services
BM	Bad mouthing
BS	Ballot Staffing
SP	Self Promoting
TV	Trust Value

2. Cryptographic-Based Clustering

MANETs are vulnerable to many passive and active attacks. Considering passive attacks, the attacker can only eavesdrop or monitor the network traffic [12, 13] and tries to achieve valuable information. But in active attacks, the attacker is not only able to listen to the communications but also it is able to alter or manipulate it. Primarily, MANETs are vulnerable to the following active attacks:

- Sybil attacks that adversary presents more than one identity to network nodes [14].
- Replay attack that adversary replays the previously transmitted messages.
- Spoofed data attack that adversary intercepts, alters data and transmits them to the destination.
  - Wormhole attack: that an attacker receives packets at one point tunnels them and replays them into another point in the network. This tunnel between two colluding attacks is known as a wormhole.
  - Black hole attack that attacker advertises a zero metric for all destinations causing all nodes around it to route packets towards it. Then it drops all packets that receive instead of forwarding them [15].
  - Gray-hole attack which is a routing misbehaviour that leads to dropping of messages. This attack consists of two phases. Regarding first phase, attacker advertises as having a valid route to destination and in second phase, attacker drops received packets occasionally.
  - Sinkhole attack that a compromised node tries to attract and drops data from all neighboring nodes [16].
  - Denial of service attacks that are aimed to complete disruption of ad-hoc network.
  - Selfish nodes which use network for their advantage and do not participate in operations to save energy.

Most of these attacks can be launched independently or by colluding between multiple malicious and misbehaving nodes. It is obvious that detecting and defending against collusion attacks is harder than independent attacks. Also, attacks that are caused by mixture of insider and external attackers can be more catastrophic and more powerful, so more resources are needed to protect the clustering process against them. In hierarchical networks, CHs are the primary components of clusters and perform special tasks such as data processing, intra-cluster and inter-cluster routing. But these capabilities make them more vulnerable to security attacks which can be conducted against them. As a result, secure clustering schemes primarily focus on protecting the existing CHs and selecting valid and right node as new CH and the security of cluster maintenance has secondary importance.

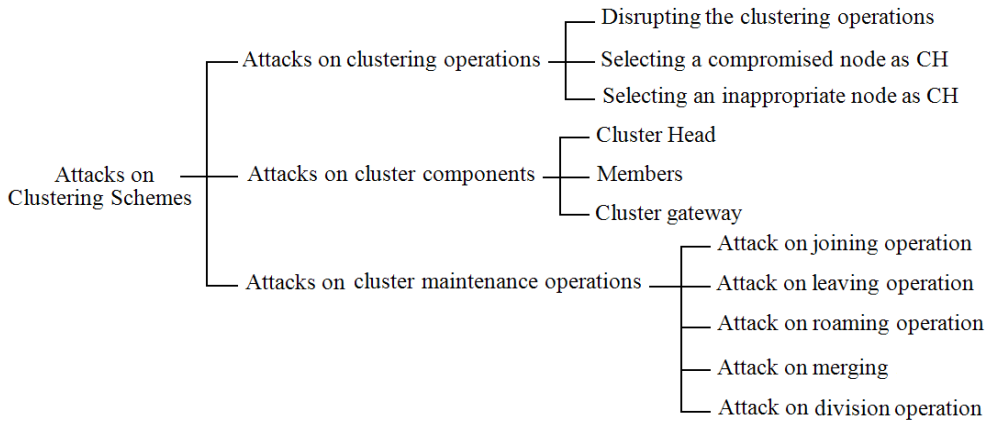

Figure 2. Classification of Attacks on Clustering Operations

Figure 2 shows various security attacks that can be conducted against clustering. These attacks are categorized into the following classes:

- Attacks on clustering operation.
- Attacks on cluster maintenance operations.
- Attacks on cluster components such CHs, cluster gateways or members.

Cryptographic-based clustering mechanisms [17-20] use cryptography to defend network against security threats and provid security services such as data privacy, authentication and digital signatures. The security of cryptographic solutions highly depends on the key management methods that they use. Generally, key management is defined as “the set of techniques and procedures supporting key establishment and the maintenance of keying relationships between authorized parties”. Normally, a key management system performs the followings tasks in a security solution:

- Generating and distributing of keying material.
- Controlling the use of keying material.
- Updating and revocation of keying material.
- Backup and recovery of keying material.

Figure 3 classifies the key management techniques in mobile ad hoc networks. Certificates in public key-based security schemes are managed by the web of trust and hierarchical trust method which can be achieved by certificate authority that issues and manages the digital certificates.

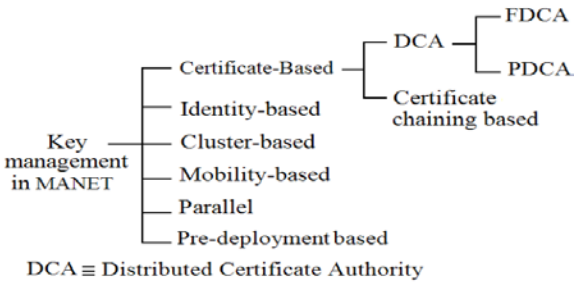

Figure 3. Key Management Methods in MANET

Certificate authorities can be classified as centralized and distributed forms. Distributed Certificate Authority or DCA is implemented through the distribution of the certificate authority's private key to a number of ad hoc nodes. As a result, when operations such as issuing or revoking certificates are required, a threshold of available shareholding DCA nodes should participate to perform the requested task. In MANET, Distributed CAs can be classified as partially or fully CAs. In Partially implemented DCA or PDCA, all tasks of the certificate authority are distributed to a set of specialized nodes using secret sharing. But in Fully CA or FDCA, services of certificate authority are distributed to all nodes and each node can generate partial certificates. Almost all DCA schemes use the threshold cryptography. In this method, cryptographic operation is divided among some nodes, so the action can be done if at least a certain number of parties collaborate. For example, a (t-1, n) threshold signature allows in a group of a total of n parties any t parties sign jointly, but no coalition of up to t -1 parties [21]. Therefore, any service provided by CA is performed jointly by t CA nodes. In this way, even if an attacker has compromised less than t CA nodes, it still cannot recover CA's secret key. On the other hand this scheme fails over a sufficiently long period. Determining the periods of private key and key share updates are very important and have direct impact on the security and performance of DCA. In addition, in hybrid MANETs some solutions rely on the CA of a conventional network. In [22, 23] we have analyzed various DCA schemes that are proposed for MANETs in the literature.

2. Trust-Based Clustering Schemes

According to Figure 1, the trust-based clustering algorithms are classified into two types of algorithms: pure and hybrid. Regarding the first one, algorithms pursue two main purposes: first they aim to improve the network security by electing trustworthy nodes as CHs, second they try to decrease the overheads of trust management systems by combing trust related operations with various phases of clustering algorithms. They consider the trust level of nodes as the main metric in the CH election and prevent the malicious nodes to be selected as CH. Generally, the pure trust-based clustering schemes consist of the following components:

- A clustering algorithm
- A trust management system

Normally, pure trust-based security is susceptible to various attacks such as bad mouthing and self promoting pattacks. Also, pure cryptographic-based security systems do not provide complete protection against attackers and mostly are vulnerable to internal malicious and misbehaving nodes. Slightly more complex schemes are the hybrid trust-based clustering schemes that can be considered as the ultimate secure clustering solutions. They use cryptographic-based security mechanisms in addition to the trust management systems and

this combination create complex and strong security solutions that successfully can protect clustering process against both internal and external attackers. Trust management systems and cryptographic-based security systems depend on each other to protect against various threats and attacks. However, they have highest resource consumption level and this issue should be considered in the resource limited ad hoc networks. This section analyzes cluster formation and management in the state of art pure and hybrid trust-based CH election algorithms which are presented for MANETs in the literature.

2.1. Proposed Schemes

This subsection describes the main characteristics of the trust-based clustering algorithms which are presented in Table 2. In [24] Elhdhili *et al.*, propose a clustering algorithm to elect trusted, stable and high-energy CHs called CASAN which creates one-hop members to minimize the overhead and consider the trust level of nodes, mobility, remaining energy and distance. To select cluster head**, e**ach node broadcasts a hello message with TTL 1 including its identification and mobility index. Then nodes with trust level less than a threshold trust executes the non-trustworthy nodes procedure and others execute the trustworthy nodes procedure. In this process, each node computes its connectivity degree which is equal to the total number of distinct hello messages that it has received. After that it broadcasts its metrics with TTL= 1 and uses received metric components of its neighbors to compute its weight and neighbor's weights. If the node compares the minimum weight to the weights in list, it will proclaim itself clusterhead by sending a role message CHMSG to its one-hop neighbors. Otherwise, it will launch a timer and wait for role messages from its neighbors with lower weights. If it receives at least one role message CHMSG, it will attach itself to the lowest weight CH and broadcast a role message to its one-hop neighbors to confirm its role as an ordinary node.

In [24] Xu *et al.*, present a trust evaluation based clustering which clusterheads jointly perform the tasks of a certification authority. In this solution, each cluster is first formed based on the trust values of the neighbor nodes. To create cluster, an ad hoc node evaluates its neighbor nodes and chooses one node that has the highest value as its trust guarantor. Then, the chosen node becomes the clusterhead and the chooser becomes a member of the cluster. After forming a cluster, the clusterhead plays the role of trust guarantor and evaluates the trust of the cluster members. When a member node requests it, CH issues the trust value certificate that contains the node's trust value. The member node uses the trust value certificate to show its trustworthiness to communicate with others.

The other trust-based clustering scheme is designed by park et al in [25]. In this scheme, each node evaluates the trust value of neighbor nodes and recommends the one with the highest trust value as its trust guarantor. Then the recommender node becomes a member of cluster head which is one-hop away. When nodes recommend a cluster head, they give a recommendation certificates called R-Certificate to cluster head which are used to authenticate. So, the cluster head which has many recommendation certificates is more trustable and the new cluster in the new place refers the node's trust value by the previous cluster head for the trust evaluation.

Voting-based clustering algorithm [26] is another trust-based clustering scheme which evaluates stability of node through computing the neighbor change ratio and the residual power of nodes. In this scheme, each node votes other nodes only if the node is the most trustful one among its neighbor nodes. Votes are propagated only to one hop neighbors and they are not forwarded by other nodes.

For clustering the following steps are followed:

- Each node computes its stability.
- Computes the trust of node with respect to its neighbors.
- Each node votes its neighbors according to the voting algorithm. Choose the largest $V(i)$ as the CH but if the number of votes is the same, choose the best stability as CH. If the number of votes and stability are the same, choose the smallest ID as CH.

Secured weight-based clustering algorithm or SCA is another scheme which is presented by the kadri *et al.*, in [27]. This algorithm elects clusterheads according to their weight computed by combining a set of parameters such as stability, battery, degree and *etc*. To create or maintain clustering architecture, first the *Discovery stage* should be done which information about the neighborhood should be retrived. For this purpose, nodes desiring to be CH send *CH_ready* beacons to their *D* hops radius. Then nodes receiving this beacon, estimate a trust value and send it back. After a discovery period, nodes having initiated this operation, can derive from the received responses information such as *degree, stability and trust value*. Then each node adds to the previous parameters the state of its *battery* and the *max value* and combines them to computes the *global weight*. After nodes choose the cluster head which has the maximum weight, each newly elected cluster head needs to discover each other to construct a virtual backbone for inter-cluster communication. Thus every new elected cluster head broadcasts a discovery request over the network. Cluster heads receiving this request register the certificate of the new cluster head and send their certificate.

Table 2. General Features of the Presented Trust-based Clustering Schemes for Mobile Ad Hoc Networks

Ref#	Intra-Cluster Topology	Clustering Factors			
		Stability	Distance	**Mobility**	**Energy**
24	One-Hop & Multihop	----	☑	☑	☑
33	One-Hop	----	----	----	----
26	One-Hop	☑	----	----	☑
25	Multihop	☑	----	☑	☑
28	Multihop	----	----	----	----
29	One-Hop	----	----	----	----
30	One-Hop	----	----	☑	☑
31	----	----	----	☑	----

In [28] wang *et al*., present a secure clustering scheme protocol that divides the MANET into several clusters and apply mesh topology structure. The CH is selected within the cluster according to the number of the trust connections and the nodes which have trust connection with CH will be the core nodes. At first the cluster nodes set their trust values as 0. The cluster service group is made up of CH and core nodes which can join together to be the service group which is in charge of providing service for various requests from cluster members. The nodes which connect to the service group will be periphery nodes and do nothing except forwarding the received messages. The messages between different clusters will be forwarded by the CHs and due to the existence of the session keys between the CHs, the messages can be transmitted in the common channel.

In [29] Ferdous *et al.*, present a clusterhead selection algorithm based on an efficient trust model. After deployment, the nodes broadcast their *ID* and trust value to their neighbors

along with the REQ/REPLY flag. When the participating nodes have discovered their neighbors, they exchange information about the number of one hop neighbors. The node which has maximum neighbors from the trust interaction table is selected as the TA and other nodes become members of the Cluster or local nodes.

In [30] Chatterjee *et al.*, present a solution which evidences of trustworthiness are captured from direct recommendations. After deployment each node sends "HELLO" beacon and each node receiving this beacon replies with his ID and public key and increases the counter of its neighbor list. Then an efficient secure distributed leader election algorithm called SEC-LEAD is executed which can adapt to arbitrary topological changes. In this algorithm, first a node that wants to be CH broadcasts "START-ELECTION" message with its mobility and battery power value to all its one hop neighbors. Each node that receives this message calculates the global weight of that candidate node using a global function as follows:

$$G_w = w_1 * TV + w_2 * MV + w_3 * BP$$

Where w1, w2, w3 are different weights such that w1+w2+w3=1. If this value is greater than a predefined threshold, the node will vote for M by signing a Leader Certificate. After a certain time interval, the candidate node will count how many certificates it has already received. If this is greater than half of the neighbor nodes, it advertises itself as leader and broadcasts the leader message with the set of node-ids who has voted for it. If any node finds that its id is falsely included, it will generate a warning message to all its neighbors. After certain time, neighbor nodes will sign a TrustCert for Leader. Thus M becomes a Leader and the elector nodes who have signed the certificate become its member.

SCAR is another secure clustering algorithm designed by Yu *et al.*, in [31]. It takes into account a combined weight metrics including the reputation value, the nodes' degree and the relative mobility. In this solution, each node broadcasts Hello message to its neighbor nodes periodically for connectivity and the weight information is carried in Hello message. When the node receives its neighbor nodes' Hello messages, it updates the related nodes' reputation value. In addition, the node can update its degree and mobility according to the number of Hello messages received and the transmission power. After receiving Hello message in some period, the node gets its initial weight and sends its weight through the broadcasted Hello message. Compared with other nodes' weight, the node that has the highest weight is elected as CH. If the node A receives the CH message from its neighbor node B and node B's reputation value is higher than A's, A will send the message to node B to join its cluster. If node A hasn't received the CH's message during a period, it becomes an isolate CH which has no cluster member. In [32] palanisamy *et al.*, address the key management issues in ad hoc networks and present a scheme called TBCMKDS. The system builds clusters and spans the entire network. The nodes within the cluster select a CH using degree of neighbors, node identification and trust. The proposed scheme uses the modified version of web of trust and elliptic curve cryptography (ECC) is used for key pair and certificate generation, since ECC provides higher security with less key size.

2.2. Cluster Management

After the clusters are established, the topology of network may change due to some factors such as: mobility, reduction of CH's power and so on [31]. In addition, MANET nodes often have short lifetime and may join or leave the network more frequently than conventional networks. A perfect clustering is not limited to just initial clustering of network. It should respond to the dynamic situation of MANET. Therefore such schemes should support roaming; joining and leaving operations and these should be done securely in secure

clustering schemes. It is clear that the performance of a clustering scheme highly depends on the efficiency of nodes that join and leave operations. In the following subsections we analyze the solution that each scheme presents the following questions:

- How small or large clusters are handled in sparse and dense partitions of MANET?
- How trust and reputation mechanism are combined in these schemes?

If the CH moves fast, it may cause frequent re-clustering and result in the increase of control packets. Thus, an efficient and stable clustering will decrease the cluster management overheads.

2.2.1. Joining to Clusters

Join operation is one of the important parts of each clustering algorithm which should be considered in the following issues:

- Preventing the attackers to join the clusters.
- Producing symmetric clusters which do not need further cluster division and merging.
- Having impact only on one cluster to prevent ripple effect.

This subsection analyzes the joining operation in secure clustering schemes from security perspective. In [33], when a node joins to the network, it broadcasts a hello message. Any CH that receives it, sends a respond message. The respond message of the head contains the number nodes and the CH's trust value. After receiving the response message, the join node sends the join message to the CH. This step is also called the log-on procedure in which a new node joining the network becomes a guest node and later a full member. In order to log on, the new node needs the trust value from its neighboring nodes. Each of these trust value is signed by the neighbor to guarantee its authenticity and also includes a period of validity. When CHs are being asked to share certificates by a new node, they first have to make sure that the issuers of trust value are really authorized to vouch for a guest, then they check the trust value to see whether it is above the threshold or not. After that the CHs send their shares of an identity certificate if all the certificates are valid. After the new node collected enough certificate shares, it can complete its identity certificate. Now having its key signed, the new node is a full member. The CH sends the symmetric cluster key to the new node. If the joining node is not able to find any CH in a one-hop range, it has to construct the cluster with the neighbor node. Since, the CHs run NS3, the new CH can hold the secret share.

In [28], when a new node wants to join a cluster, it sends a join request and public key to the service group of the cluster. After permission, the service group will encrypt the secret share and cluster key by public key of node and they will send back to the node. Subsequently, the service group will inform all the nodes within the cluster about the joining of new node while the trust value of it will be set as zero. After that, the newly added node can communicate with the cluster member successfully. By this process, the cluster members will refuse to communicate with any stranger nodes.

In [32], a join protocol is presented in which CH evaluates the trust value of totally new nodes, because node does not have a trust value certificate and recommendation certificate. Furthermore, nodes gives new CH the trust value certificate and recommendation certificates which are received from the previous CH. Using these certificates, new CH establish an initial trust value of the new member node and authenticate the previous CH of new node. In [30], to register node, each CH starts to broadcast CH beacon and attracts some nodes to join its cluster. As the node M gets the CH beacon, it sends “JOIN” beacon to join the network with its public key. CH checks whether it is a duplicate message or not. If it is not a duplicate, CH stores the public key of M as its id and generates a pair wise shared key to communicate

between CH and M. Also it sends a secret key to secure intra cluster communication. Initially CH gives the node suspicious status and allows it to register subject for periodic review. Then CH sends a "WhocanSense" message along with the status of the newly joined node to its member nodes to review the status of it. CH calculates its direct trust about M and asks its one hop members of M to send their recommendation for M. Then CH tries to combine evidences to find the most probable belief of M. If Trust is higher than a threshold, CH will send a trust certificate CERT. Thus M becomes a Trusted Member of the cluster.

Leaving from Cluster

When some node is missing from cluster, it may leave the MANET or roamed to the other clusters. Mobility is one of the issues that have negative impact on the network architecture and clustering. To decrease the impact of mobility on clustering the following solutions have been proposed in the literature:

- Use of more stable nodes as CH.
- Use of group-mobility based clustering solutions such as [34, 35].

In fact missing node may have responsibilities such as CH or cluster gateway and its absence should be detected and handled somehow. Generally, when the CH leaves the cluster, the cluster members can select a new CH or join to nearby clusters. For example, in [33] when CH wants to leave the network, it chooses the node with the highest trust value among the cluster members as its successor. Then the old CH securely migrates its state to the successor and sends a signed broadcast message containing the new CH's identity. The new CH holds the share of the network key and has to notify the members of the CHs about the CH delegation. During the next refresh of the key shares, the new CH will be updated. However, if no CH successor can be found by the old CH, members have to join neighboring clusters or form a new cluster. Table 3 shows the missing nodes handling methods in MANET.

Table 3. Handling Missing Nodes in MANET

Method		Advantages	Disadvantages
Node inform its departure from network		Has the lowest overhead. Leaving node can select best successor for itself.	It does not work for nodes that disappear suddenly by failures
Someone should detect node's absence	**Source Initiated**	In high traffic this method can be implemented by overhearing the traffic of monitored node, In low traffic, monitored node should inform its presence by sending periodic hello messages.	High messaging overhead for transmitting periodic hello messages.
	Destination Initiated	This is done based on a simple request reply protocol that may be performed in a secure manner.	

Also in the [26], after clustering, nodes can change their cluster by moving to the new position and should execute the join and leave protocols. In the leave protocol, the CH gives to the leaving node a trust value certificate and recommendation certificates. The trust value certificate guarantee the node's trust value and recommendation certificates prove that the CH is not malicious.

Furthermore, to detect the missing nodes, in the [30] each node has to send an ALIVE beacon to CH at a certain time interval. If the CH cannot hear from a node at a certain time out, there will be two possible reasons: One is due to mobility of the silent node which may move to such place, from where it cannot sense the CH or the node is damaged. CH broadcasts a "WhocanSense" message and tries to sense the Silent node. If any node gives any reply to this message, the CH will try to establish a path to the silent node through that answering node and ask its location about new CH. Then it sends the Trust CERT of missing node to the new CH, Otherwise CH removes the information of the silent node from its list.

2.2.2. Cluster Division

In dense or large MANETs some clusters may have more members than the others. This issue depletes the energy of the CH and increases the collisions in MAC layer which decreases the network throughput. Thus some schemes propose to divide more populated clusters. For example in [25], whenever CH becomes busy and cannot support the increasing number of nodes, it launches a cluster division procedure to divide the cluster into two small clusters. Therefore, the CH broadcasts *Cluster_Division* request to its members. Whenever this request is received, each member computes its weight and sends it back to the CH which saves them. Then CH chooses the farthest node with the maximum weight as a new CH and sends it a grant response. Then new CH begins to create its cluster.

2.3. Trust Management Systems

This section focuses on general attributes and properties of general trust management schemes that are designed for MANET. However, before analyzing any trust based system, it is significant to understand trust and reputation concepts. Generally, the following definitions have been presented for reputation in literature:

- The reputation of a node can be defined as its quality in terms of its behavior [36].
- It is what is generally said or believed about a person's or thing's character or standing.
- It is an expectation about an agent's behavior which is based on information about or observations of its past behavior.
- It is a perception that an agent creates through past actions about its intentions and norms.

Although there is no clear consensus on the definitions of trust, numerous definitions are proposed:

- Trust is a subjective opinion in the reliability of other entities or functions including veracity of data, connectivity of path, processing capability of node and availability of service etc.
- Trust in general is defined as belief, subjective probability, and reputation etc. As a natural consequence of combining trust within specific application environments, trust has been put forward with quite different meanings and corresponding features [36].
- Trust can be defined generally as the expectation of one person about the actions of others [36].

Generally, trust describes a subjective relation between an entity and another entity (or group of entities) while reputation is what is generally said about an entity. Thus, the reputation of an entity is based on the opinions provided by all entities. Trust may be used to determine the reputation of an entity. In [36] Cho *et al.*, present a complete survey on trust management in MANET and specify the following properties for trust:

- Trust is *dynamic*. Trust establishment should be based on temporally and spatially local information. Because due to node mobility or failure, information is typically incomplete and can change rapidly.
- Trust is *subjective*. A node may determine different levels of trust against the same trustee node due to different experiences.
- Trust is *not necessarily transitive*. For example, if *A* trusts *B*, and *B* trusts *C*, it does not guarantee *A* trusts *C*.
- Trust is *asymmetric*, not necessarily reciprocal. Nodes with higher capability may not trust weaker nodes at the same level that weaker nodes trust nodes with higher capability.
- Trust is *context-dependent*. Depending on the given task, different types of trust e.g., trust in unselfishness, trust in forwarding versus trust in reporting are required.

Also, a reputation management should perform the following tasks:

- Reputation gathering
- Trust Computation
- Trust Revocation

Considering first stage, trust management systems attempt to collect the reputation ratings from the whole MANET. Figure 4 shows the reputation gathering steps in a trust management system. To compute trust value, most of these systems utilize first hand information and second hand information. First hand information is the data that each node collects about its neighboring nodes by directly observing its behaviors such as routing and forwarding packets. This can be done by eavesdropping broadcasted data in the wireless channel. Also, the second hand information is reports that are sent by neighboring nodes and should be accepted only from the trustworthy senders [37], because as Figure 4 shows the second hand information may be sent by malicious and normal nodes. Furthermore in [38] Zhang specified the following attacks against the trust-based systems:

- *Sybil Attack* that occurs when a malicious node creates and uses fake identities.
- *Newcomer Attack* that occurs when a malicious node register itself as a new user.
- *Betrayal Attack* that occurs when a trusted node suddenly turns into a malicious and starts to attack.
- *Inconsistency Attack* that is also called on-off attack and happens when a malicious peer repeatedly changes its behavior from honest to dishonest.

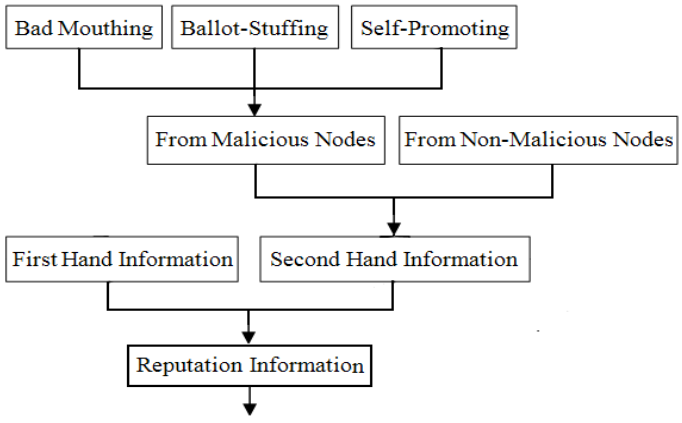

Figure 4. Reputation Gathering in MANET

Table 4 specifies the attacks on trust management systems in MANETs. To prevent these attacks, trust management system should apply some filtering methods to decrease the impact of these security attacks. In addition, all of these attacks can be controlled by careful considerations in trust value initialization and update procedures. After gathering reputation information, the trust value is computed which reflects the degree of the trust between the trustor node and trustee node. As Figure 5 exhibits, a trust computation that can be done using a centralized, distributed or hybrid approach. In centralized computation, observations about a node's behaviors are propagated to a central authority that derives reputation values for each node and subsequently updates nodes with new reputation values.

Table 4. Comparison of Trust-Based Clustering Schemes in MANETs

Attack	Description
Bad-mouthing	Maliciously trying to lower the reputation of well-behaving nodes
Ballot-stuffing	Maliciously trying to increase the reputation of some
Self-promoting	Attackers seek to falsely augment their own reputation. Such attacks are only possible in systems that consider positive feedback in the formulation.
Sybil Attack	This type of attacks occurs when a malicious node creates and uses fake identities.
Newcomer Attack	These attacks occur when a malicious node register itself as a new user.
Betrayal Attack	Such attacks occur when a trusted node suddenly turns into a malicious one and starts to attack.
Inconsistency Attack	These attacks are also called on-off attacks and happen when a malicious peer repeatedly changes its behavior from honest to dishonest.

However, the centralized structures can become a single point of failure and have low scalability. In distributed structure, each node propagates its observations to neighboring nodes which then calculate the reputation values individually. Also, it avoids single points of failure in the system and balances load across multiple nodes.

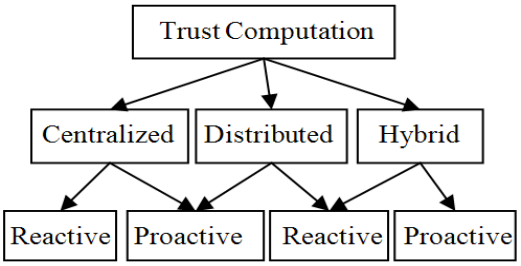

Figure 5. Methods of Trust Computation and Distribution

Finally in hybrid structures, reputation values are calculated by more than one entity *e.g.*, each CH can collect and compute the trust values of its members. This method which has high scalability, distributes the trust computation overheads on multiple nodes and prevents the single point of failure in the network. Normally, the second hand information can be disseminated by proactive or reactive methods. In the former, reputation values are broadcasted periodically, although there are no changes to reputation values since last update. In the latter, reputation values are only broadcasted when there are sufficient changes to these reputation values, such as the occurrence of a specific event, or that a request for a reputation

value is received. After trust computation, reputation system should decide that the trustworthiness of node is enough for a certain interaction or not, i.e. it is trusted or not. This is usually done based on a threshold value which can be static and dynamic. If the trust value of a node is above the predefined threshold, the node will be trusted, otherwise it will not be trusted and should be isolated from network. Figure 6 shows possible threshold values for trust-based clustering schemes in MANETs that can be static and dynamic and we can consider multiple thresholds for various tasks and operations. Although, most of the trust-based solutions use one threshold for evaluation of trustworthiness, more complicated trust-based clustering schemes consider multiple thresholds, for example, some scheme may define different thresholds for CHs, gateway nodes and cluster members. When the trust values are computed, they should be stored in trust tables. However, these trust values should not be remained intact for very long time and must be updated somehow. Aging is one of the methods that can be used to decrease the stored trust values, if no first and second hand information is achieved on some period.

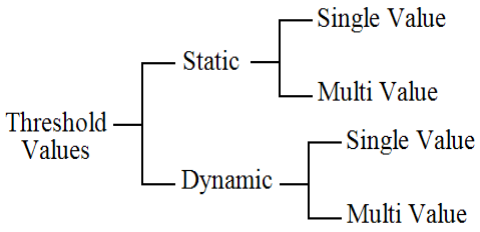

Figure 6. Type of Threshold Values

The computed trust value can be positive, negative or both of them. Also, it can be continuous or discrete. As a result, determining the type and range of trust values is an important issue which has profound impact on the security and performance of trust management system. For example, by considering only positive values, the bad mouthing attack can be prevented, because in this case malicious nodes cannot affect the trust value of good nodes by propagating the negative reputations. But, malicious nodes can collude and falsely praise misbehaved nodes to launch ballot staffing attacks. Propagating positive feedback also exhausts the network's limited resources since the number of nodes that behave correctly is supposed to be larger than those which do not.

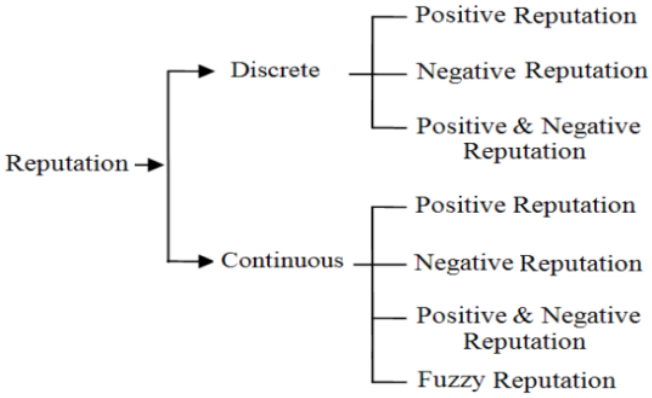

Figure 7. Trust Type in Trust Management Systems

Thus, the number of transmissions required to update reputation values is high, which depletes the limited energy of nodes. Furthermore, using only negative trust values prevent the malicious nodes from colluding and praising misbehaving nodes which are called ballot staffing attack. It also helps to minimize the number of transmission required to update the reputation values. However, malicious nodes can assign negative reputation ratings/feedback to trustworthy nodes in order to affect their trust level (BM attack).

2.3.1. Trust Computation

Most trust-based security schemes rely on Bayesian formulation as Beta reputation system for trust evolution. Beta reputation system is presented by Jøsang *et al.*, in [39]. In this scheme, prior probabilities of binary events can be represented as beta distributions which are composed of two parameters α and β. The beta distribution $f(p|\alpha, \beta)$ can be expressed by the gamma function Γ as:

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

$$0 \leq p \leq 1, \alpha > 0, \beta > 0$$

The probability expectation value of the beta distribution is given By $E(p)=\alpha/(\alpha+\beta)$

To show how Beta functions can be employed, let us consider the task of target detection as an action with two possible outcomes, namely ‘‘correct” and ‘‘false”. Let r be the observed number of ‘‘correct” target detections and s be the observed number of ‘‘false” target detections by a sensor node. The beta function takes the integer number of past observations of ‘‘correct” and ‘‘false” target detections to predict the expected frequency of ‘‘correct” target detections by that sensor node in the future which is achieved by setting:

$$\alpha = r + 1 \qquad \beta = s + 1$$

The variable p represents the probability of “correct” target detections and $f(p|\alpha, \beta)$ represents the probability that p has a specific value. The probability expectation value is given by $E(p)$ which is interpreted as the most likely value of p. Hence, a sensor node’s reliability can be predicted by beta distribution function of its previous actions as long as the actions are represented in binary format.

After the first hand information (direct trust) and second hand information (indirect trust) are collected, they are combined together by the following function to compute the trust value:

$$New\ Trust\ Value = \alpha * (Direct\ Trust) + \beta * (Indirect\ Trust) + Initial\ Trust\ Value$$

In this formula, α and β are the coefficients that control the importance and impact of the direct and indirect trust. In addition, direct trust is computed by this formula:

$$Direct\ Trust = f(\sum_{i=1}^{n} event_i)$$

$$Indirect\ Trust\ Value = f(\sum_{j=1}^{m} report_j)$$

As we stated in Section 3.1.1 some clustering schemes use beta reputation system in these formulas. This subsection shows how trust based clustering schemes compute trust value. According to Figure 8 shows, in [27] both the direct trust and recommendation trust is used. In this scheme, $TV_{i,j}$ represents the trust value between the trustor node *i* and trustee node *j*, T_D indicates the direct trust, and T_R indicates the recommendation trust.

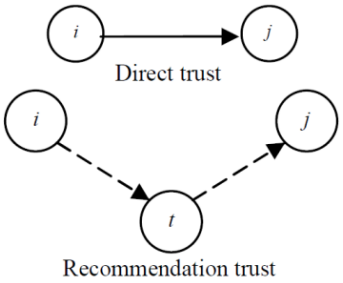

Figure 8. Direct and Recommendation Trust (indirect trust)

In this scheme, direct trust between *i* and *j* can be calculated as:

$$T_{D_{ij}} = \tanh(\sum_{K=1}^{n} \mu_k * w_k * p_k)$$

Which *pk* represents the number of experience *k* upon the trustee node, *n* is the total number of various experiences and w_k is the weight of this experience to represent the importance of the level of experience. μ_k is +1 if experience *k* is positive and -1 if it is a negative experience. Moreover, the recommendation trust value can be calculated as follows:

$$T_{R_{ij}} = 1/\mathrm{n}(\sum_{K=1}^{n} TV_{it} * TV_{tj})$$

which TV_{it} is the trust value between the trustor node *i* and the third node *t*, TV_{tj} is the trust value between the third node *t* and the trustee node *j*, *n* is the number of the third nodes. They believe that how much recommendation value will be taken depends on how much direct trust value the trustor node gets. The bigger value a trustor node gets from its own direct experience, the smaller value the trustor node will consider from the third node. If the trustor has no direct experience about the trustee node, the trust evaluation will rely on the recommendations. To satisfy these properties, the following equation is defined to compute the trust value:

$$TV_{ij} = TD_{ij} + (1 - |TD_{ij}|) * TR_{ij},$$
$$-1 \leq TD_{ij} \leq 1, -1 \leq TR_{ij} \leq 1,$$

Scheme such as [28] use Node-based Trust Management (NTM) which compute TEs (Trust Evaluators) from the following equation:

$$T_{ni,nj} = \alpha_{1ni} T_s^{nj} + \alpha_{2ni} T^{nj}{}_o$$

Where, $0 \leq \mathrm{T_{ni,nj}} , niT_s^{nj} ,\mathrm{niT^{nj}}\ \mathrm{o} \leq 1$. In this equation, niT_s^{nj} is the node ni's self evaluated trust on nj and node ni computes this by directly monitoring nj. Also, $\mathrm{niT^{nj}o}$ is the weighted sum of other nodes' trust on nj evaluated by ni. α1 and α2 are weighting factors such that α1 + α2 = 1. Thus, by varying α1 and α2, ni can vary the weight of self evaluated vs. others trust in calculating its total trust on nj. To update trust value, this scheme uses the

concept of trust decay. Assume that node ni has trust value on node nj is Tnij (t1) at t1 time. After a certain period, node nj may leave the cluster temporarily. The trust value of node nj decays over this time gap. Then Tnij the new trust value of nj given by ni at time t2 computed as follows:

$$T_{nij}\,(t2) \;=\; T_{nij}\,(t1) \;*\; exp[-(T_{nij}\,(t1)\Delta t)]^{2k}$$

Where $\Delta t = t2 - t1$ and k is an integer such that $k \geq 1$.

$$T_{ni}(t) = \sum_{i=0.5}^{i=1} T_{ni} + \Delta t + \beta_{ni}$$

Where β_{ni} is the step value for node n−i which is assigned as a small fractional value during simulation. Also T_{ni} (t) is $0.5 \leq T_{ni}$ (t) ≤ 1.

In [26] Peng *et al.*, established direct trust $T_d(i, j)$ by directly observing another node's behavior. Also, when one node receives recommendations from other nodes, recommendation trust $Tr(i, j)$ can be established which denotes that there is no direct interaction between nodes *i* and *j*, while there may be indirect interactions between them. In this scheme, the total trust degree of successful interaction probability $T(i, j)$ between nodes *i* and *j* by integrating $Td(i, j)$ and $Tr(i, j)$.

$$T(i,j) = \omega_1 T_d(i,j) + \omega_2 T_r(i,j)$$

In this scheme, direct trust is established by the following equation:

$$Td(i,j) = E\big(Beta(\theta|s+1, f+1)\big) = \frac{s+1}{s+f+2}$$

Table 5. Direct Trust Computation in Trust-Based Clustering

Ref#	Direct Trust	Description
28	$T_{D_{ij}} = \tanh(\sum_{K=1}^{n} \mu_k * w_k * p_k)$	*Pk*, the number of experience *k* upon the trustee node, *n* the total number of various experiences, w_k , the weight of experience to its importance, μ_k is +1 if experience is positive and -1 if it is negative.
26	$Td(i,j) = E\big(Beta(\theta\|s+1, f+1)\big) = \frac{s+1}{s+f+2}$	described by the number of total interactions, number of successful interactions, number of failed interactions.
40	$T_D^{ij} = \frac{m + \lambda/2}{n+\lambda}$ $m, n \geq 0$ *and* $\lambda > 0$	Considers the individual experience of the past transaction with other nodes, It is taken as 0.5 if there is no previous interaction.

Where $0 < \theta < 1, s > 0$ *and* $f > 0$. The recommendation trust is achieved as follows:

$$T_r(i,j) = E\big(Beta(\theta|s1+s2+1, f1+f2+1)\big) = \frac{s1+1}{n1+2} * \frac{s2+1}{n2+1}$$

In [25], four features are considered as evaluation factors that are communication value, data value, recommendation value, and malicious black list. Communication value expresses a success or fail that are communications between two nodes. It consists of two categories,

one is data delivery rate and the other is communication rate. Malicious black list contains the id of malicious nodes which have been detected by data value or trust value. This value set in 1 or 0. Value 0 means that the node is malicious and value 1 means it isn't. The following equations show how a node computes the trust value on other nodes.

$$V_T(i,j) = \frac{\sum_{v(j)\varepsilon S}\left(1-(1-Vi(j))^{v(\dot{j})}\right)}{|S|} * \mathrm{V_B(j)}$$
$$v_i(j) = max(w * V_R(j), 1/2)$$

Where S is the set of evaluation factors and $V_T(i,j)$ is the trust value of node j evaluated by node i. $V_R(i)$ is the trust value of node j evaluated by the node j's CH and $V_B(j)$ is a value that shows whether node j is malicious or not. Furthermore, $V_i(j)$ is the initia1 trust value of node j and w is the weight of node. The scheme proposed by Rani *et al.*, in [37] uses both direct trust and recommendation trust. Node Ni takes into account the experience of the past transactions with *Nj*. If *Ni* and *Nj* have *n* times transaction with *m* times success, the direct value is calculated as:

$$T_D^{ij} = \frac{m + \lambda/2}{n + \lambda} \quad m, n \geq 0 \; and \; \lambda > 0$$

Table 6. Indirect Trust Computation in Trust-Based Clustering

Ref#	Indirect Trust	Description
26	$T_r(i,j) = E\left(Beta(\theta \mid s1 + s2 + 1, f1 + f2 + 1)\right) = \frac{s1+1}{n1+2} * \frac{s2+1}{n2+1}$	n_1 and n_2, number of total interactions between nodes *i* and *k* and between nodes *k* and *j* respectively; s_1 and f_1, the number of successful and failed interactions between nodes *i* and *k* respectively; s_2 and f_2, number of successful and failed interactions between nodes *k* and *j* respectively. Recommendation trust: the successful probability of the $(n_1+n_2+2)th$ interaction.
28	$T_{R_{ij}} = 1/\mathrm{n}(\sum_{K=1}^{n} TV_{it} * TV_{tj})$	TV_{it} is the trust value between the trustor node i and t, TV_{tj} is the trust value between the third node and the trustee node j, n is the number of the third nodes
40	$T_R^{ij} = \sum_{i=1}^{n} T_D^{hi} T_D^{ij} / \sum_{i=1}^{n} T_D^{hi}$ $where \; T_D^{hi} > H, i \neq j$	N, number of nodes in the current cluster, Aggregation weight, direct trust value of node Ni by CH. CH collects the recommendations and calculates recommendation trust

Table 7. Trust Value Computation in Trust-Based Clustering Schemes

Ref#	Trust Computation	Description
26	$T(i,j) = \omega_1 T_d(i,j) + \omega_2 T_r(i,j)$	ω_1 and ω_2 weighting factors are used to tune the impact of the direct and indirect trust.
29	$T_{ni,nj} = \alpha_{1ni} T_s^{nj} + \alpha_{2ni} T^{nj}{}_o$	α1 and α2 are weighting factors, α1 +α2 = 1 niT_s^{nj}: *ni*'s evaluated trust on *nj* computed by direct monitoring. $niT^{nj}{}_o$: Weighted sum of other nodes' trust on *nj* evaluated by *ni*.
28	$TV_{ij} = TD_{ij} + (1 - \lvert TD_{ij} \rvert) * TR_{ij}$, $-1 \leq TD_{ij} \leq 1, -1 \leq TR_{ij} \leq 1$	$0 < TV_{ij} < 1$, node j is a trusted node $-1 < TV_{ij} < 0$, node j is suspected, if $TV_{ij} < T_{threshold}$, node j is convicted, it will be added into the CNL (convicted nodes list)

40	$V_T(i,j) = \frac{\sum_{v(j)\varepsilon S}\left(1-(1-Vi(j))^{v(j)}\right)}{\lvert S\rvert} * V_B(j)$ $v_i(j) = max(w * V_R(j), 1/2)$	S: Set of evaluation factors. $V_R(i)$: The trust value by node j evaluated by the nodej's CH. $V_B(j)$ the value whether node j is malicious or not. $v_i(j)$: initia1 trust value of node j, w: weight value about node.

The direct trust value is taken as 0.5 if there is no previous interaction between Ni and Nj. If the first interaction is success, the direct trust value will increase rapidly. Otherwise it will decrease quickly. Also, recommendation trust is calculated by the following equation, to the unknown or unfamiliar nodes by the CH for every node in the cluster.

$$T_R^{ij} = \sum_{i=1}^{n} T_D^{hi} T_D^{ij} / \sum_{i=1}^{n} T_D^{hi} \; where \; T_D^{hi} > H, i \neq j$$

Where n is the number of nodes in the current cluster. Note that the recommendation trust of cluster head is always 1 in the proposed model. Table 5 and 6 specifies the direct and indirect trust computations and Table 7 shows the trust computation methods in trust-based clustering schemes.

2.4. Security Overheads

Figure 9 shows the security overheads in MANET clustering solutions. It shows that as the level of security increased, more processing should be made and more storage should be considered for storing cryptographic materials and trust tables. In addition, it shows that trust-based systems have the highest energy consumptions and delay because gathering information from spatially remote areas will consume more resources. Therefore, they save energy needed for future retransmissions and re-clustering operations, by electing trusted routes and selecting trustworthy nodes as CH.

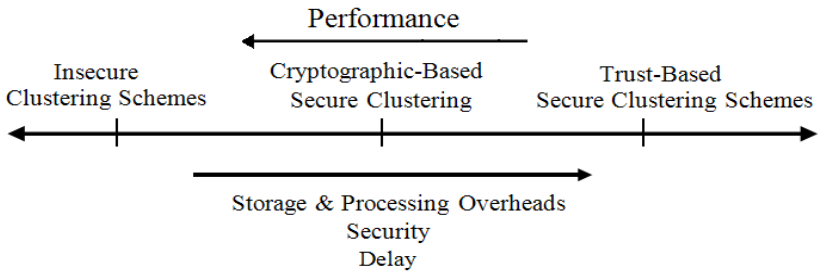

Figure 9. Security Overheads of CH Selection

Table 8: Overheads of first and second hand information	
First Hand Information	**Second Hand Information**
Storage Overhead to maintain trust table	Storage Overhead to maintain trust table
Power consumption for monitoring events	Power consumption for transmissions and computations
	Communication overheads
	Vulnerable to various security attacks of external attackers
	Susceptible to lying attacks such as BM, BS and SP

Thus, from energy perspective, trust-based clustering solutions decrease the energy consumption of sensors in hostile environments and this issue increases the lifespan of WSNs. On the other hand, trust-based security systems consume a lot of energy to collect and distribut reputations of sensor nodes. Therefore, mechanisms and solutions should be designed to decrease the overheads of reputation management. Table 8 shows the overheads of first and second hand information. As this table specifies because second hand information have more overheads, trust-based clustering solution should mostly rely on first hand information and use second hand information occasionally and less frequently.

3. Conclusion

Clustering is one of the techniques that are used to organize more scalable networks. Although, providing security issues in clustering algorithms had been neglected in the literature, recently trust and reputation based solutions have been proposed to protect clustering algorithms against insider attacks and malicious behaviors of compromised nodes. As these schemes use nodes' previous actions as their reputations and apply nodes that have specific threshold of trust value, they prevent the attackers to be elected as CH or join to existing clusters. It should be noted that pure clustering algorithms are not totally secure and they are susceptible to various attacks such as bad mouthing, ballot staffing and self promotion. To compensate the vulnerabilities of pure trust-based clustering schemes, hybrid trust-based clustering algorithms have been presented which are resilient against both internal and external attackers. In this paper, we discussed about pure and hybrid trust-based clustering algorithms and analyzed their properties and features. We mainly focused on the integration of trust related issues in the cluster formation and maintenance process and compare the techniques which each scheme is applied for direct and indirect trust computations. Finally, although numerous secure clustering schemes have been presented for mobile ad hoc networks, there is a lack of clustering scheme that is able to operate in both trusted and hostile environments. Such schemes present low overheads in the trusted environments and provide high security in more hostile conditions. As a result, designing such scheme should be considered in the future researches and studies.

References

[1] Y. Zhang, J. Mee Ng and C. Ping Low, "A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks", Journal of Computer Communications, vol. 32, **(2009)**, pp. 189-202.
[2] C. Konstantopoulos, D.Gavalas and G. Pantziou, "Clustering in mobile ad hoc networks through neighborhood stability-based mobility prediction", Computer Networks, vol. 52, **(2008)**, pp. 1797-1824.
[3] J. Y. Yu and P. H. J. Chong, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks", IEEE Communications Surveys & Tutorials, **(2005)**.
[4] D. Wei and H. A. Chan, "Clustering Ad Hoc Networks: Schemes and Classifications", 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, **(2006)**, pp. 920-926.
[5] S. Chinara and S. K. Rath, "A Survey on One-Hop Clustering Algorithms in Mobile Ad Hoc Networks", Journal of Network and Systems Management archive, vol. 17, no. 1-2, **(2009)** June, pp. 183-207.
[6] K. Erciyes, O. Dagdeviren, D. Cokuslu and D. Ozsoyellery, "Graph theoretic clustering algorithms in mobile ad hoc networks and wireless sensor networks", Applied and Computational Mathematics, vol. 6, no. 2, **(2007)**, pp. 162-180.
[7] R. C. Hincapie, B. A. Correa and L. Ospina, "Survey on Clustering Techniques for Mobile Ad Hoc Networks", **(2006)**.
[8] G. Kumar, K. K. Tripathi and N. Tyag, "Research Survey of Load Balancing Clusters in Wireless Ad hoc Network", International Journal of Electronics Engineering, **(2011)**, pp. 305-307.
[9] P. Rai and S. Singh, "A Survey of Clustering Techniques", International Journal of Computer Applications, vol. 7, no. 12, **(2010)** October.
[10] I. G. Shayeb, A. R. H. Hussein and A. B. Nasoura, "A Survey of Clustering Schemes for Mobile Ad-Hoc Network (MANET)", American Journal of Scientific Research, **(2011)**, pp. 135-151.

[11] R. Agarwal and M. Motwani, "Survey of clustering algorithms for MANET", International Journal on Computer Science and Engineering, vol. 1, no. 2, **(2009)**, pp. 98-104.
[12] P. Goyal, S. Batra and A. Singh, "A Literature Review of Security Attack in Mobile Ad-hoc Networks", International Journal of Computer Applications, vol. 9, **(2010)** November, pp. 11-15.
[13] B. Wu, J. Chen, J. Wu and M. Cardei, "Wireless/Mobile Network Security", Springer, chapter 12, **(2006)**.
[14] K. F. Ssu, W. T. Wang and W. C. Chang, "Detecting Sybil attacks in Wireless Sensor Networks using neighboring information", Computer Networks, vol. 53, no. 18, **(2009)** December 24, pp. 3042-3056.
[15] M.-Y. Su, "Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems", Computer Communications, vol. 34, no. 1, **(2011)** January 15, pp. 107-117.
[16] H. C. Tseng and B. Jack Culpepper, "Sinkhole intrusion in mobile ad hoc networks: The problem and some detection indicators", Journal of Computers & Security, **(2005)**, pp. 561-570.
[17] Y. Zeng, J. Cao, S. Guo, K. Yang and L. Xie, "SWCA: A Secure Weighted Clustering Algorithm in Wireless Ad Hoc Networks", IEEE Wireless Communications and Networking Conference WCNC, **(2009)**.
[18] I. Nishimura, T. Nagase, Y. Takehana and Y. Yoshioka, "Secure Clustering for Building Certificate Management Nodes in Ad-Hoc Network", International Conference on Network-Based Information Systems, **(2011)**, pp. 685-689.
[19] H. Rifà-Pous and J. Herrera-Joancomartí, "A Fair and Secure Cluster Formation Process for Ad Hoc Networks", Journal Wireless Personal Communications: An International Journal archive, vol. 56, no. 3, **(2011)** February.
[20] V. Sivaranjani and D. Rajalakshmi, "Secure Cluster Head Election for Intrusion Detection in MANET", Journal of Computer Applications, vol. 5, no. EICA2012-4, **(2012)** February 10.
[21] N. Saxena, G. Tsudik and J. H. Yi, "Threshold cryptography in P2P and MANETs: The case of access control", journal of Computer Networks, vol. 51, **(2007)**, pp. 3632-3649.
[22] M. Masdari and J. Pashaei, "Distributed Certificate Management in Mobile Ad Hoc Networks", International Journal of Applied Information Systems, **(2012)** November 6.
[23] M. Masdari, S. Jabbehdari, M. Ahmadi, S. M. Hashemi, J. Bagherzadeh and A. Khadem-Zadeh, "A survey and taxonomy of distributed certificate authorities in mobile ad hoc networks", EURASIP Journal on Wireless Communications and Networking, **(2011)**.
[24] M. E. Elhdhili, L. B. Azzouz and F. Kamoun, "CASAN: Clustering algorithm for security in ad hoc networks", Computer Communications, vol. 31, **(2008)**, pp. 2972-2980.
[25] C. Park, Y. Lee, H. Yoon, S. Jin and D. Chio, "Cluster based Trust Evaluation in Ad Hoc Networks", pp. 503-507.
[26] S. Peng, W. Jia and G. Wang, "Voting-Based Clustering Algorithm with Subjective Trust and Stability in Mobile Ad-Hoc Networks", IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, **(2008)**, pp. 3-9.
[27] B. Kadari, A. Mhamed and M. Feham, "Secured Clustering Algorithm for Mobile Ad Hoc Networks", IJCSNS International Journal of Computer Science and Network Security, vol. 7, no. 3, **(2007)** March, pp. 27-34.
[28] L. Wang and F. Gao, "A Secure Clustering Scheme Protocol for MANET", International Conference on Multimedia Information Networking and Security (MINES), **(2010)**, pp. 785-789.
[29] R. Ferdous, V. Muthukkumarasamy and E. Sithirasenan, "Trust-based Cluster head Selection Algorithm for Mobile Ad hoc Networks", International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, **(2011)**, pp. 589-596.
[30] P. Chatterjee, "Trust Based Clustering And secure routing Scheme for Mobile Ad Hoc Networks", International Journal of Computer Networks & Communications, vol. 1, no. 2, **(2009)** July, pp. 84-97.
[31] Y. Yu and L. Zhang, "A Secure Clustering Algorithm in Mobile Ad Hoc Networks", IPCSIT, vol. 29, **(2012)**.
[32] V. Palanisamy and P. Annadurai, "Trust-based clustering for multicast key distribution scheme in ad hoc network (TBCMKDS)", Journal International Journal of Internet Protocol Technology, vol. 6, no. 1-2, **(2011)**, pp. 46-64.
[33] L. Xu, X. Wang and J. Shen, "Strategy and Simulation of Trust Cluster Based Key Management Protocol for Ad hoc Networks", Proceedings of 4th International Conference on Computer Science & Education, **(2009)**, pp. 269-274.
[34] Y. Zhang, J. M.ng and C. P. Low, "A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks", Journal Computer Communications archive, vol. 32, no. 1, **(2009)** January.
[35] Y. Zhang, J. M. Ng and C. P. Low, "A distributed group mobility adaptive clustering algorithm for mobile ad hoc".
[36] J. H. Cho, A. Swami and I. R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks", IEEE Communications Surveys & Tutorials, **(2011)**, pp. 562-583.

[37] J. Duan, Y. Qin, S. Zhang, T. Zheng and H. Zhang, "Issues of Trust Management for Mobile Wireless Sensor Networks", 7th International Conference on Wireless Communications, Networking and Mobile Computing, **(2011)**, pp. 1-4.
[38] J. Zhang, "A Survey on Trust Management for VANETs", International Conference on Advanced Information Networking and Applications, **(2011)**, pp. 105-112.
[39] A. Jøsang and R. Ismail, "The beta reputation system", 15th Bled Electronic Commerce Conference, **(2002)**.
[40] V. G. Rani and M. Punithavelli, "Optimizing on Demand Weight -Based Clustering Using Trust Model for Mobile Ad Hoc Networks", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) vol. 1, no. 4, December.

Authors

Mehdi Maleknasab, is teaching at the department of Computer Science at spidan University of spidan - Iran. He has received his master degree in Software Engineering from Azad University southern Tehran branch. His main research interests include Mobile Ad Hoc Networks and Sensor networks.

Moazam Bidaki, is a researcher at the research center of Computer Science at Azad University of Neyshabur - Iran. She has received his master degree in Software Engineering from Azad University southern Tehran branch. Her main research interests include Distributed Systems and cryptography.

Ali Haron Abadi was born in Iran; He is a PHD in the department of computer engineering Islamic Azad University Branch of North Tehran.

