The importance of location on *Trust* in mobile networks

Dagmara Spiewak, Volker Fusenig, and Thomas Engel
University of Luxembourg
FSTC
6, r. Richard Coudenhove-Kalergi
L-1359 Luxembourg
Dagmara.Spiewak@uni.lu, Volker.Fusenig@uni.lu, and Thomas.Engel@uni.lu

Abstract: Mobile wireless networks introduce new challenges regarding security and privacy of data. On the other hand self-organization and independence of fixed infrastructure make these networks, such as mobile ad-hoc networks (MANETs) very attractive for military but also civilian application areas. They allow to extend the wireless link into areas with no readily available communication infrastructure. Additionally, these networks can also be used as a subsequent to the common communication environment in order to assure communication on-the-fly even if the regular network is overwhelmed, like for instance during emergency situation or during major sports or cultural events. With the purpose to overcome the security problem accompaingned with these networks, more and more research is launched in the area of *Trust* establishment in mobile wireless networks. However *Trust* in mobile network settings introduces new challenges compared to the conventional notions for infrastructure networks. Especially mobile behavior, which is enabled by wireless links, diversifies *Trust* research in multiple ways. Hence unfortunately, traditional security concepts, such as Public Key Infrastructures are no accurate solutions to protect sensitive communication and data in these autonomous network environments. Our paper discusses the *Trust* establishment in mobile wireless networks. We introduce the idea of *TrustRings* which enables the calculation of *Trust-Values* for nodes in mobile wireless networks based on an egocentric network model. Furthermore, the model takes the location and distance between communicating entities into account in order to obtain the accurate Trust-value.

Key–Words: Trust, Security, Mobility, MANETs, Network Model

1 Introduction

Mobile networks are booming in the sense that more and more people require the access to the Internet and data every-time and from every-where. In the event that the user has no direct access to the network, Mesh-Networks can be deployed in order to extend the wireless link toward the user‘s device.
The main characteristic of mobile wireless networks, including mobile ad-hoc networks (MANETs) and Mesh-Networks, is that these systems are able to interconnect in a dynamical self-organized way allowing the extension of common Wireless LAN technologies into areas with less or even no previously available network infrastructure.
However, the nature of mobile wireless networks with its resource-constrained devices makes them very vulnerable to malicious attacks and selfish actions . Particularly, due to the absence of pre-established communication infrastructures and the absence of continuously accessible central entities, security in mobile ad-hoc wireless networks is very difficult to reach and to maintain. Nevertheless, confidential data and sensitive applications transmitted within mobile wireless networks require a high degree of security. Therefore, more and more research topics are focusing on the establishment of *Trust-Metrics* in order to overcome this weakness and to ensure secured and reliable communications in these almost autonomous network scenarios of mobile wireless ad-hoc networks and Mesh-Networks.

The crucial point is, that *Trust* [24] in the field of network security is not clearly defined. The word *Trust* is mostly used intuitively frequent, serving as foundation for follow-on security concepts, like for example as a basis for public-key management infrastructures. So far, subjective interpretations about the meaning of the word *Trust* lead to big ambiguousness. Pradip Lamsal in [17] and Audun Josang in [15] present a wide expertise on the description of *Trust* as well as its relationship towards *Security*. Beyond, Pirzada and McDonald emphasize in [21] the interdependency of *Trust* and *Security*, while *Security* is highly dependent on trusted key exchange and trusted key exchange on the other side can only proceed with

the required security services. Furthermore, the notion of *Trust* in mobility settings is compared to *Trust* applied to the Internet in [5], for instance while thinking on the *Ebay recommendation system*, highlighting the independence of previously build *Trust* infrastructures.

The paper is organized as follows: sections 2 and 3 present a classification of mobile ad-hoc networks. Section 4 discusses relevant related work on the establishment and distribution of *Trust-Values* within fixed and mobile network settings. Subsequently, Section 5 introduces our concept of *TrustRings* to obtain and calculate the *Trust-Value*. Finally, section 6 concludes the paper.

2 Classification of mobile ad-hoc networks

Two categories of mobile ad-hoc networks can be identified. A managed environment is the first one, where a central trusted authority provides certification services. The second class of mobile ad-hoc networks is known as open environment. This category of mobile ad-hoc networks does not require a central authority to manage the network. The strength of these networks is it's self-organization, which means that the network can function without the pre-establishment of network infrastructure, configarations, and without any kind of external organization. Due to the fact that security strategies designed for the second category of mobile ad-hoc networks are also adequate to secure a managed mobile ad-hoc network, this paper will focus on open environment mobile ad-hoc networks.

3 Communication in mobile ad-hoc networks

The core concept of communication in networks is *routing*. The main functionality of routing is the process of discover and determine paths in the network in order to send packets. Routing tables store information about the routes towards many destinations. Keeping these tables as updated as possible in a very important task in order to realise fast communication. The following figure 1 presents the three basic routing schemes: unicast, multicast, and broadcast. These routing schemes differ in their sending method:

- unicast sends a message to a single node
- multicast sends a message to a group of nodes
- broadcast sends a message to all nodes in the network

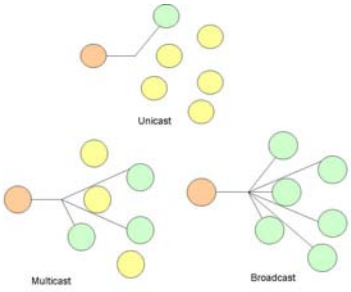

Figure 1: Routing schemes

In fixed networks routing problem is referred to as finding a route from a source node to the destination with the lowest cost. In this context, the network is represented as a graph with a set of nodes and edges. A cost-function is defined on the edges so that each edge has a certain cost.

Due to dynamic topology changes, routing in mobile ad-hoc networks is more tricky than in fixed networks. Within a mobile ad-hoc network devices, such as notebook computers, PDAs, cell phones, are able to communicate. Since no pre-established infrastructure in needed to set up a mobile ad-hoc network, routing needs to be generated in a distributed, self-organized manner and launched by the mobile devices by themselves and to act also as a router to transmit packets to nodes out of direct communication range. Numerous routing protocols for mobile ad-hoc networks have been proposed.

Basically, three different categories of routing algorithms can be identified:

- Reactive routing algorithms (on-demand)
- Proactive routing algorithms
- Hybrid routing algorithms

Routing protocols can keep routing information proactively (all the time) or can reactively compute them (on-demand). A hybrid routing protocol combine both technologies.

3.1 Reactive routing protocols

Mobile ad-hoc networks deploying a reactive routing protocol do not maintain up-to-date routing information on all nodes the whole time. Correct route are calculated on-demand and just in the event a node wishes to send a message. The advantage of reactive routing protocols is that the shared wireless medium is not overloaded with routing data, which is not used. On the other hand, in a scenario with high communication amount to different nodes, a reactive protocol impacts the performance of the mobile ad-hoc network

significantly due to a huge message overhead needed to obtain the correct routing information.

Famous reactive routing algorithms for mobile ad-hoc networks are:

- Associativity Based Routing - ABR
- Ad Hoc On Demand Distance Vector Routing Protocol - AODV
- Dynamic Source Routing - DSR

3.1.1 Associativity Based Routing - ABR

Associativity Based Routing - ABR [26] is a reactive routing algorithm initiated by the source node of the communication. The selection of routes is based on the stability of links between nodes. Periodical HELLO-messages allow nodes to advertise their existence to their neighbors. Each node maintains a table filled with associativity values, which help to rank a neighbor-link as stable or not. Hence, the core concept of ABR is to identify stable routes. The protocol operates in the following three steps:

- Route Discovery
- Route Repair/Reconstruction
- Route Delete

Route Discovery process

We suppose node A wishes to communicate with node E. In the event the route towards E is in A‘s routing table, than the communication starts immediately. Otherwise, the route discovery protocol is launched:

- Node A floods the network with *RouteRequest* messages
- Each intermediate node appends its address and its associativity value to the packet
- Destination node selects the best route by verifying the associativity values along each path
- Destination node sends a *Reply* packet to the source along the chosen path

Route Repair process

All neighbor nodes detect if a link is broken. In this event, the closest node to the source is the initiator of the route repair process:

- Node broadcasts a *Route Repair* message (Local Query - LQ) to his neighbors with a limited Time-to-Live stamp. Consequently, the broken link can be by-passed without flooding the mobile ad-hoc network again.

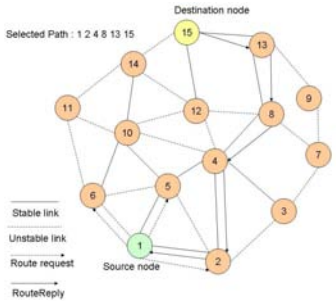

Figure 2: ABR Route Request

- Otherwise, the next node in direction to the source reinitiates the above process
- This process continues recursively.
- Until finally, the source is informed to start a new Route Discovery process.

In the event a node is moving and the topology of the mobile ad-hoc networks is changed, the last node before the destination deletes its route. Consequently, a Local Query process is launched to determine if the node is still reachable. If the node is reachable, it selects the most efficient route and replies. Otherwise, the Local Query process is forwarded to the subsequent node. *Route Notification* messages inform the next node to delete the invalid route. This process continues until more than the half route is backtracked. Finally, the source will have to launch a new broadcast query process. Figure 3 demonstrates the Route Repair process.

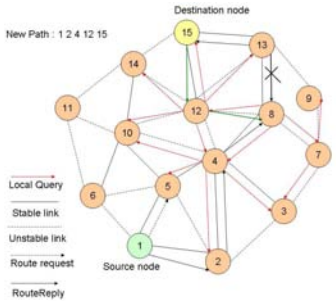

Figure 3: ABR Route Repair

Route Deletion process

Reactive routing algorithms compute the routes just on-demand. If a route is no longer needed, the source node launches a *RouteDelete* broadcast. Each node on the route deletes the route from their routing table.

ABR provides no security. Hence malicious nodes might request routes for non-existing nodes disturbing the communication. A high amount of fake requests can result in the breakdown of the whole mobile ad-hoc network.

3.1.2 Ad Hoc On Demand Distance Vector Routing Protocol - AODV

Another famous reactive routing protocol is the Ad Hoc On-Demand Distance Vector Routing Protocol [18] introduced in 1997 where routes are calculated only when needed. Figure 4 demonstrates the steps of the protocol.

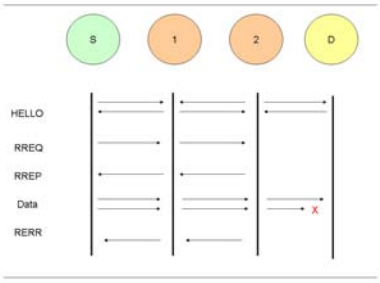

Figure 4: AODV protocol messaging

Like in the ABR protocol HELLO messages are utilized to discover and maintain links to neighbor nodes. These messages are sent periodically and in the event a node fails to receive some HELLO messages from a neighboring node, he assumes the breakage of the link. *Route Request* (RREQ) messages are used by a source node in order to discover the destination of the node the source wishes to communicate to.

Each intermediate node establishes a route to the source, if it receives a RREQ message. The intermediate can detect if it is the destination node immediately due to the fact that if the node did not receive this RREQ message before it knows that it is not the destination node the source wishes to communicate to. In this case, the intermediate node rebroadcasts the RREQ message. On the other hand, if the intermediate node is not the designated destination node, but it knows the accurate route toward the destination nodes, it responses the source with a *Route Reply* (RREP) message by the use of a unicast routing strategy. Consequently, the RREP message is spread allowing the establishment of a route to the designated destination node. After receiving the RREP message, the source starts sending data. Each node has a routing table. The crucial entries of a RREP message are:

- Destination IP Address
- Destination Sequence Number
- Next Hop IP Address
- Time-To-Live showing the expiration time of the route
- Hop Count showing the amount of hops to reach the destination node
- State and routing flags, such as valid, invalid

The protocol allows the establishment of multiple route towards the destination and the source decides to choose the shortest route in order to send the data. In order to keep the route in their routing table, each node updates the timer, which is binded with the source and the destination. After a certain period of time in which the timer was not updated the node knows that the route was not used. Due to the mobile character of the network, the node is not sure about the validity of the route and the deletes it from his routing table.

Another functionality of AODV is *Route Error* detection. *Route Error* (RERR) messages is sent to the route if a link break is detected. During the hop-by-hop propagation of the RERR message, each intermediate node on the route towards the source marks this route as invalid. The source it-self, marks this route as invalid as well and reinitiates route discovery again. Figures 5, 6, 7, and 8 show an example of this protocol.
AODV provides no security. It is very easy to impersonate a node *i* by forging a RREQ with its address as a originator address or to impersonate a node *j* by forging a RREP with its address as a destination address. Furthermore a malicious node may selectively not forward certain RREQs and RREPs, or not answer several RREPs, and do not forward certain data messages. A secure version of the AODV protocol is the Secure Ad hoc On-Demand Distance Vector (SAODV) [27]. It protects the route discovery mechanism ensuring security aspects like integrity and authentication. Integrity protection in this protocol relies on hash-chains that allow to protect hop-count information, which is the only changeable information within the messages. Malicious nodes often attempt to decrease the hop-count of a RREQ message in order to increase the life-time of the message. The trick behind this action is to gain more time to analyze the communication. A hash-chain is generated by using a one-way hash function repeatedly to a seed. Authentication in SAODV is realized by digital signatures by which all fields of the message are signed except the hop-count and the hash-chain fields. The price for the protection of integrity and the establishment of authentication is the maintenance of an asymmetric cryptosystem, in the sense that each node has its own key private-/public-key pair, which is not a flexible solution and which contradicts to the nature of mobile ad-hoc network with its self-organized character.

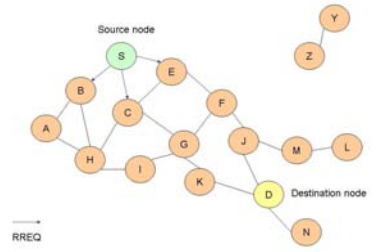

Figure 5: AODV Route Request

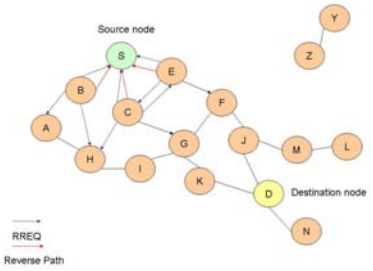

Figure 6: AODV Reverse Path

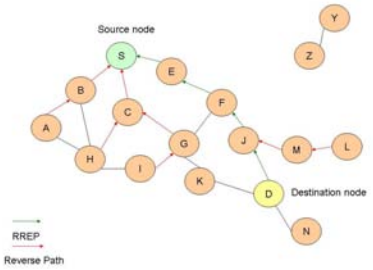

Figure 7: AODV Route Reply

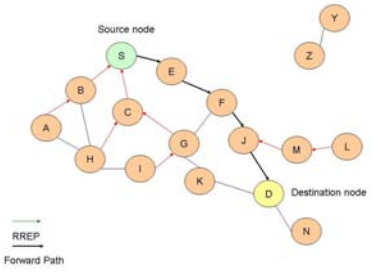

Figure 8: AODV Forward Path

3.1.3 Dynamic Source Routing - DSR

The Dynamic Source Routing is another famous reactive routing protocol [12] and as the name indicates, the source node initiates the communication process. DSR functions in two steps:

- Route Discovery
- Route Maintenance

The header of the packets contains the whole route to its destination (from the source node to the destination node).

Route Discovery process

The first steps of the *Route Discovery process* is the broadcast of a *Route Request* message by the source node, that wishes to start a communication. All nodes within the node's transmission range receive this message. The RREQ message has the following entries:

- Unique *requestID*
- *Record-list* to store each node on the path
- Hop-limit showing the amount of nodes the message is allowed to be routed through

Figures 9 shows the action flow of a node after it received a *Route Request* message.

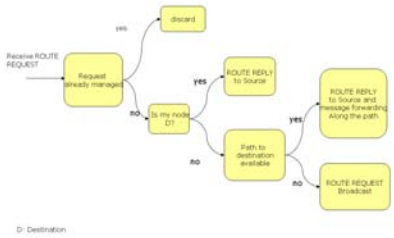

Figure 9: DSR Route Request handling

At first, each node checks if this *Route Request* message was already processed. In the negative case, the node controls, if the desired communication destination node is equal to his node ID. If the node ID's are equal he sends a unicasts *Route Reply* message back to the sender of the *Route Request* message indicating that he is the destination. Otherwise, the node checks, if the route to the required requestID of the destination is stored in his cache. Finally, the node either sends a unicasts *Route Reply* message

back to the sender of the *Route Request* message with the required route to the destination node, or he rebroadcasts the *Route Request* message. This process continues until the destination node is reached.

Route Maintenance process

Due to the dynamic nature of mobile ad-hoc networks routes between nodes are changing. Therefore, the route maintenance mechanism is very important for finding incorrect routes. Nevertheless, route maintenance in wireless networks can easily be realized on a hop-to-hop basis. At every hop, the node sending the message for that hop is able to notice if this link of the route is still existing. For example Figures 10 presents, that node 1 is able to hear node 2's transmission of a message to node 3.

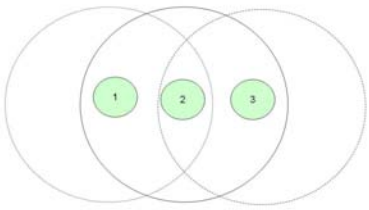

Figure 10: Ad-hoc network with three wireless mobile nodes

DSR launches the Route Maintenance mechanism during the usage of a route. After the reception of a *Route Error* message, the incorrect route is deleted from the node's cache who initiated the packet transmission. Not only link breaks can lead to the a *Route Error* message, but also a exceeded hop-limit. The *Route Error* message has the following entries:

- Address of the node that detected the error
- Address of the node to which the node was attempting to transmit the packet on this link

Johnson and Maltz [12] have optimized the DSR protocol. For instance the hop-limit avoids the mobile ad-hoc network is overloaded with old *Route Request* messages. In order to prevent storms of *Route Reply* messages, which occur if all neighbors of the destination node attempt to send a *Route Reply* message to initiator at the same time, the authors introduce a delays sending of *Route Reply* messages. Nevertheless, security aspects, such as authenticated *Route Request* messages are not considered in the protocol.

Hu, Perrig and Johnson extended DSR by security and introduce the protocol called *Ariadne* [7]. Such as SAODV, *Ariadne* makes use of hashchains for securing messages as well. The protocol can operate in three different manners, with pre-established symmetric keys, with digital signatures, or with the TESLA system [20] which is an authentication protocol for broadcasts. *Ariadne* allows a destination node to verify the identity of the source node. Moreover, the source node is able to authenticate all intermediate nodes on the route. Nevertheless the protocol is efficient, because of the use of symmetric cryptography. However, both the identity of the source node and the route towards the destination node rest unsecured and cause a big vulnerability to anonymity threats, which will be explained in section 2.

3.2 Proactive routing protocols

Proactive routing protocols follow the strategy to calculate routes, before they are needed. They try to maintain routin information all the time and for all nodes up-to-date. Two categories of proactive protocols can be indetified, which differ in their method of keeping the routing tables updated:

- Regular updates of routing tables
- Event-driven updates of routing tables

The second category launches an update of the routing table in the event that a change in the network topology was detected. The strategy of the protocol will determine how other nodes are informed about the new routes. On the other hand, proactive routing protocols following the regular routing table update approach, will send their topology information to other nodes at regular times. The main advantage of proactive protocols is that routes towards nodes can be assumed as known. Everytime a communication is desired, it can be started without a delay for route establishments in contrast to reactive protocols. Famous proactive routing algorithms for mobile ad-hoc networks are:

- Destination Sequenced Distance Vector Routing Protocol - DSDV (event driven)
- Optimized Link State Routing Protocol - OLSR (regular updated)

3.2.1 Destination Sequenced Distance Vector - DSDV

The Destination Sequenced Distance Vector protocol [19] was introduced in 1994 by Charles E. Perkins and Pravin Bhagwat and is an event-driven proactive ad-hoc routing protocol. It adjusts an famous distance

vestor algorithm, called distributed Bellman Ford [4] to an ad-hoc network. On the one hand nodes send updates on routing information on a regular basis but if a significant change to topology is detected, updates to routing information are transmitted immediately, even in-between the regular updates. The routing table of each nodes contain the following entries:

- All nodes with the amount of hop of the path towards them
- Sequence-numer indicating the up-to-dateness of routes

In order to reach a consistancy between routing tables of all nodes in the network, each node sends it's routing table it's neighbors. DSDV provides no security.

3.2.2 Optimized Link State Routing Protocol - OLSR

The Optimized Link State Routing Protocol (OLSR) is another proactive routing protocol [26]. In this protocol each node sends HELLO-messages on at regular basis in order to exchange neighbourhood information. Based on these information each node builds it own routing table and calculates routes to any nodes it wishes to communicate. Routing table information is updated due to one of the following events:

- Detection of a change in the neighbourhood
- Expiration of a route to a destination node
- Detection of a shorter route to the destination node

4 Trust research

Trust in fixed networks One milestone in the history of cryptography is the concept of *Pretty Good Privacy* or *PGP* [28] which made cryptography available to a wide community. Principally created for email-encryption and -signing, *PGP* functioned as a hybrid cryptosystem based on the concept of *Web of Trust*. Basically, the idea is to allow each user to operate as an autonomous certification authority, enabling them to sign and verify keys of other entities even without the maintenance of a centrally managed certification authority. This results in the establishment of various virtual interconnections of *Trust*. However, even though no central authority is needed to sign the keys, the distribution of keys is handled by a continuously accessible directory making *PGP* inadequate in mobile network settings. The core of the famous *Distributed Trust Model* [1] is the *recommendation protocol* which is always launched in the event that the *Trust Value* of a certain network entity is required. Depending on the output of this protocol *Trust* is measured and assigned into categories ranging from -1 (complete distrust) to 4 (complete trust). Evidently, distributing recommendations about entities has to be secured from unauthorized modifications and fake recommendation spreading. Unfortunately, centralized maintenance and distribution of recommendations is not feasible in mobile network settings. Furthermore recommendation based protocols are very vulnerable to *Sybil-attacks*, which is elaborated in [22]. Therefore, the new *TrustRing* idea, presented in this paper, will not involve or even consider recommended or third-party information for the purpose to calculate the *Trust-Value* of communication entities.

Audun Jøsang expresses *Trust* as *Beliefs* and uses the method of *Subjective Logic*, introduced in [13], in order to calculate the *Trust-Value* among arbitrary network entities [14]. Generally, *Belief theory* facilitates the approximate reasoning on trueness of facts principally in situations of incomplete knowledge. However, if we exemplarily consider the scenario of authenticating a network entity *B* within a mobile wireless network scenario in multi-hop transmission range by another network entity *A*, we discover that an unbroken chain of *trusted* entities is very essential, in order to reason about the real identity of *B*. The assumption of an unbroken chain within wireless and mobile network settings is a critical condition, while taking the high vulnerability to wireless link breaks of mobile networks into account [23].

Trust in mobile networks *Trust management* in mobile ad-hoc networks poses several challenges compared to *Trust* in traditional networks like the Internet or common WLAN architectures. Typically, sources of *Trust*, like *Trusted Third Parties* (*TTP*) reside on centralized servers and operate as fully-trusted and continuously accessible *Trust* evidence distribution network entities. Obviously, these centrally managed *Trusted Third Parties* are entirely important for the overall security of the network. Unfortunately, as a result these entities produce a single point of failure within the network, which means, that by compromising only this entity, the security of the whole crashes. Due to the fact that entities of dynamic and mobile wireless networks are much easier to compromise, centrally managed *Trusted Third Parties* are not adequate to function as sources of *Trust* within mobility settings.

Unfortunately, the attractiveness of mobile wireless networks of *anytime* and *anywhere* communica-

tion, is always accompanied with certain weaknesses, like for example the breakage of wireless links or the unavailability of services, making also centralized management systems inadequate. As a consequence, *Trust* management has to be organized in a distributed way and handled by network entities themselves. Accordingly, each network entity needs to individually evaluate the *Trust-Value* of another entity without referring to a global *Trust-Value* assignment system.

One novel work on *Trust* computation and distribution in mobile and dynamic networks was developed by Tao Jiang and John S. Baras [10]. It presents a methodology for distributing *Trust-Certificates* called *ABED (Ant-Based Evidence Distribution Algorithm)* by utilizing the idea of Swarm Intelligence Paradigm [3]. The proposed algorithm generates ants every time a certain certificate, which serves as a *Trust* evidence, is required. The main weakness of the ABED approach is its high vulnerability to Denial-of-Service attacks [22]. After a detailed analysis on the model it is obvious, that a malicious network entity has the capacity to send a huge amount of certificate requests for non-existing certificates simultaneously simply by spreading ants into the network.

A very famous *Trust* model is the *EigenTrust* algorithm described in [16]. The model targets the establishment of *Trust* within Peer-to-Peer networks. Comparable to dynamic mobile networks, centralized *Trust* management in Peer-to-Peer is not possible. The *EigenTrust* algorithm helps to reduce the amount of not authentic files within the system even in the presence of collaborating adversarial network entities. In order to reach their aim, the authors assume several peers as *pre-trusted* from the outset. These *pre-trusted* entities might be for instance the initiators of the network. One interesting aspect of this approach is the generation of a global *Trust-Value*, which represents how much all network entities trust one specific network node. This global *Trust-Value* is based on local *Trust-Values*, collected from either positive or negative transactions. Basically, the main weakness of this approach is the precondition of *pre-trusted* network entities. Nevertheless, the overall idea of the *EigenTrust* algorithm might be enhanced and tailored to the dynamic nature of mobile wireless networks, by for example introducing a random as selection of the *pre-trusted* entities. The more serious problem of the algorithm is based on the calculation of the global *Trust-Value*. Generally, the collection of information, in that case the collection of local *Trust-Values*, implicates multiple additional security problems in mobile network settings. In order to avoid *Sybil-attacks* each of these local *Trust-Values* hat to be communicated over authenticated channels, which is a critical condition, taking into account the high vulnerability to wireless links breaks in mobile network settings. For that reason, the newly elaborated concept of *TrustRings*, which is going to be presented in the following section of this paper, is completely independent of globally *Trust-Values*.

In contrast to the *EigenTrust* algorithm the authors in [11] assume that *Trust* is handled completely distributed and only restricted to local interactions. Keeping this idea in mind, they model the mobile network as an undirected graph (V,E) where the edges represent connections to exchange trust information. This means that two end-nodes of an edge are no physical neighbors in geometrical distance although they have a trust relationship in the graph. The distributed trust computation model is based on elementary voting methods, so that only entities in the neighborhood have the right to vote if the network entity is trustworthy or not. An entity tries to find the most trusted nodes in order to create a secure path for communicating to another entity. Unfortunately, also this *Trust* model is very vulnerable to *Sybil-attacks* where the attacker may fake opinions about the trustworthiness of a certain node in order to attract more traffic to it and compromise the node.

In the following section we will model the network and present the idea of *TrustRings* used to calculate *Trust-Values* in mobile and dynamic wireless network settings.

5 TrustRings Network Model

The foundation of *TrustRings* represents an egocentric network model, demonstratet in figure 11.

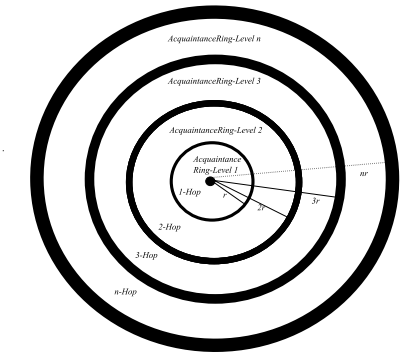

Figure 11: Network model

The TrustRing Network Model procedure is performed by each node in the network autonomously in the following way: Placing itself as the centric node in the middle of the network, first of all each node starts to build 3-dimensional spheres around itself using the multiple its own transmission range as the radius of

the sphere. According to this, the first sphere of each node is created by using exactly the transmission range (maximum 1-Hop distance) of each node. The model assumes that all nodes have the same transmission range, so that the nodes' spheres at the same Hop-distance have equal dimension. Continuing this process, the next sphere of each node is created by using the doubled transmission range (maximum 2-Hop distance), the third sphere is generated applying the triple transmission range (maximum 3-Hop distance) and so on. Figure 1 visualizes a reduced 2-dimensional view of the *TrustRing* Network Model, where spheres are represented simplified as rings leading to the name of the model. All entities within the direct (1-Hop) range from the centric node are located within the innermost sphere, named *AcquaintanceRing-Level 1*. The subsequent sphere, which is generated by the centric node, is called *AcquaintanceRing-Level 2*. By further iterating this process, *AcquaintanceRings* of different levels are created, for example in *n-Hop* distance from the centric node the *AcquaintanceRing-Level n* sphere is located. However, the assumption that a centric node *i* can communicate with node *j*, located within *i*'s *AcquaintanceRing-Level 2*, with 2 Hops while bridging the distance through an intermediate node *k*, where node *k* forwards the packet to the required destination node *j*, is generally **wrong**, which is demonstrated in figure in figure 12.

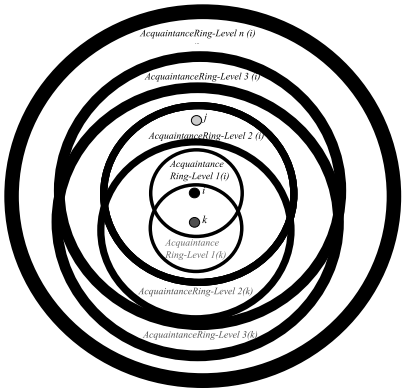

Figure 12: Network model 2

Although, node *j* is located within *i*'s *AcquaintanceRing-Level 2*, it is still not guaranteed that a 2-Hop route is available towards node *j* also in the presence of node *k* residing within the *i*'s *AcquaintanceRing-Level 1*.For this reason, only nodes within node *i*'s *AcquaintanceRing-Level 1* can be reached with 1-Hop communication.

5.1 Trust calculation

In addition to the egocentric view of the network, each network node *i* maintains a *Trust-Value-Database* to store the *Initial-Trust-Value* $\eta_{(i,j)}$ from known network entities that are labeled by an unique natural number j. As the name indicates, the *Initial-Trust-Value* $\eta_{(i,j)}$ is not the only *Trust-Value* a node *i* can have of a network entity *j*. This *Initial-Trust-Value* $\eta_{(i,j)}$ is calculated by node *i* only from **direct** and local interactions with the entity j within node *i*'s *1-Hop* communication range (or in other words within node *i*'s *AcquaintanceRing-Level 1*). Obviously, positive experiences with node *j* raise the *Initial-Trust-Value* $\eta_{(i,j)}$ whereas negative experiences with node *j* lead to a smaller *Initial-Trust-Value* $\eta_{(i,j)}$. Nodes have the ability to decide by themselves how they evaluate positive and negative experiences. Each node may use its own metric to calculate the *Initial-Trust-Value* $\eta_{(i,j)}$.

In any case, the *Initial-Trust-Value* is very essential and builds the foundation for all succeedingly calculated *Trust-Values* of the specific node to which the *Initial-Trust-Value* belongs to.

Depending on the distance of the *TrustRing* a network entity *j* is located from the centric node *i*, the *Trust-Value* $\eta^{(l)}_{(i,j)}$ (where l is the level-number of node *j*'s *AcquaintanceRing*) decreases exponentially starting from the *Initial-Trust-Value* $\eta_{(i,j)}$. Hence, the farther the location of node *j* the smaller its *Trust-Value* and the uncertain the reliable communication between *i* and *j*. Principally, the shrinking control over the communication paired with high vulnerability to wireless link breakages, makes communications towards nodes located within *AcquaintanceRing* of higher levels *l* more susceptible to breakdowns and malicious attacks. Therefore, the presented *TrustRing* Network Model fulfills the famous expression:

Trust is good, Control is better.

As a result, it is very important for the centric node *i* to adjust the *Initial-Trust-Value* $\eta_{(i,j)}$ of node *j* according to the geographical location represented as *AcquaintanceRing* of a certain level, if communication is desired. Furthermore, the decreasing control during communications between the centric node *i* and a node *j* located within the *AcquaintanceRing-Level l* leads to an increasing dependence on services of intermediate-nodes *k* located in *AcquaintanceRings* of lower levels then level *l* services simultaneously. These services might include, for example forwarding of packages or participating in the resolution of route-requests. The following function established in Definition 1 can be used to calculate the node *j*'s *Trust-Value* in different levels of *AcquaintanceRings*, if and only if the *Initial-Trust-Value* is already known from direct and local interactions between node *i* and *j*.

Definition 1 *For a centric node i in a mobile wireless network, let $\eta_{(i,j)}$ be the Initial-Trust-Value of network entity j within the AcquaintanceRing-Level 1 (j is located in maximum 1-Hop distance from i). Then the Trust-Value for j, if j is located within i's AcquaintanceRing-Level n, in minimum (n-1)-Hop distance and maximum n-Hop distance from i, is calculated by i with the following function:*

$\eta^{(n)}_{(i,j)} = \eta_{(i,j)} * e^{(-0.5(n-1))}$, *where* $n \in 1, 2, 3, ...$

The table highlights the influence of the *Initial-Trust-Value* $\eta_{(i,j)}$ of node *j* calculated by *i* for the subsequent decrease of the *Trust-Value* dependent on the *Hop-Distance* from the centric node *i*.
It is noticeable that the *Trust-Values* of the functions with the *Initial-Trust-Value* $\eta_{(i,j)}$ ranging from 1 to 5 fall below 1 already after the 4^{th} Hop. By doubling the *Initial-Trust-Value* $\eta_{(i,j)}$ up to 10 the curve will fall below 1 after the 5^{th} Hop. By reapplying this process to the *Initial-Trust-Value* of 20, 6 Hops are sufficient to compute a *Trust-Values* below 1. By further increasing the *Initial-Trust-Value* up to 100 the curve will fall below 1 after the 10^{th} Hop, illustrated in the table below.

Table 1: Trust-Values depending on the number of hops from center-node i and on the Initial-Trust-Value $\eta_{(i,j)}$

# Hop	$\eta_{(i,j)}$ 1	2	3	5	10	20	100
1	1	2	3	5	10	20	100
2	0.6065	1.2131	1.8196	3.0327	6.0653	12.120	60,653
3	0.3688	0.7358	1.1036	1.8394	3.6788	7.3576	36,788
4	0.2231	0.4463	0.6694	1.1157	2.2313	4.4626	22,313
5	0.1353	0.2707	0.4060	0.6767	1.3533	2.7067	13,533
6	0.0820	0.1642	0.2463	0.4104	0.8209	1.6417	8,209
7	0.0498	0.0996	0.1494	0.2489	0.4979	0.9957	4,979
8	0.0302	0.0604	0.0906	0.1510	0.3020	0.6040	3,020
9	0.0183	0.0366	0.0550	0.0916	0.1832	0.3663	1,832
10	0.0111	0.0222	0.0333	0.0555	0.1110	0.2222	1,111
11	-	-	-	-	-	-	0,674

Deciding to select 20 for the maximum *Initial-Trust-Value* $\eta_{(i,j)}$ will allow up to 6 Hops until the *Trust-Value* will fall below 1. This aligns with practical results from simulation of, for example the topology-based routing protocols for mobile ad-hoc networks, like the *Virtual Topology Based Routing Protocol* [2], that operates up to an average Hop-bound of 4. Choosing too high values for $\eta_{(i,j)}$ results in unrealistic maximal Hop-bounds and simultaneous implications of unreliable communication due to the dramatically decrease of bandwidth within the mobile ad-hoc network [6]. This leads to the conclusion that the *Trust-Values* can range from 0 to 20 basing on previous interactions and experiences.

5.2 TrustRing discovery

In the event, node *i* needs to calculate or lookup the accurate *Trust-Value* $\eta^{(n)}_{(i,j)}$ of node *j* in order to communicate within the mobile wireless network, *i* needs to determine the level of the *AcquaintanceRing* node *j* is located. This process has to be performed very carefully, because the *Trust-Value* of towards *j* shrinks with increased level of the *AcquaintanceRing*. We assume that each network entity has an unique IP address assigned, by the use of the *Distributed Protocol for Dynamic Address Assignment* [25]. Furthermore, it is obvious that in the event an entity *i* wishes to communicate with network entity *j*, *i* knows the IP address and the *Initial-Trust-Value* $\eta_{(i,j)}$ of *j*. In order to calculate *j*'s *Trust-Value* it is sufficient for entity *i* to discover the level of the *AcquaintanceRing*, in which *j* is located. It is not necessary to determine concrete coordinates of entity *j*, because the *Trust-Value* remains equal within the whole *AcquaintanceRingArea* at the same level.

One efficient mechanism was invented by Stephen Mark Huffman and Michael Henry Reifer and patented by the *United States Patent*, which allows to geolocate logical network addresses on for instance the Internet [8]. Obviously, this technology requires stationary network entities in order to be able to create the so-called *Network Topology Map*. Unfortunately, mobile ad-hoc networks are established on-the-fly without a pre-existing network infrastructure but with permanently changing and dynamic topology. Therefore, a mobile wireless network is highly dependent on cooperative behavior from network entities within their most trusted area, which is the *AcquaintanceRing-Level 1*. Consequently, in order to locate the level of the *AcquaintanceRing* of entity *j*, the network centric node *i* interviews the nodes within its *AcquaintanceRing-Level 1*, if they have any information about the location of *j*, or *i* requests them to forward the location-request message *LocReq* to their most trusted nodes within their *AcquaintanceRing-Level 1*. In return for their service, entity *i* increases the *Initial-Trust-Value* $\eta_{(i,k)}$ of the nodes *k* who participated in the *j*-location request process.

In our solution we make use of a proactive routing algorithm, such as the OLSR Optimized (Link

State Routing protocol) [26]. The core of these algorithms the calculation of routes before they are needed. Therefore finding accurate route towards the designated destination nodes is not the scope of this research.

6 Conclusion and Future Work

We described the idea of *TrustRings* which enable the calculation of *Trust-Value* for nodes in mobile wireless networks. The main concept of the presented methodology represents an egocentric view of the network. Every node assumes itself to be the middle of the network. According to this concept each node generates 3-dimensional spheres around itself, using the multiple of its maximum *1-Hop* transmission range as radius. Hence, the *TrustRings* idea allows network entities to compute the *TrustValues* towards other network participants dynamically. Based on a previously created *Initial-Trust-Value*, which is obtained by observing and measuring the *good* and *bad* experiences with the other network entity, the actual *TrustValue* will be always calculated related to the location of the nodes by using the idea of *TrustRings*. Primarily, the advantage of the *TrustRings* Network Model compared to other solutions, analyzed in section 2 of this paper, is primarily its complete independence of for example recommended third-party *TrustValues*. As a result, the *TrustRings* Network Model is resistant to *Sybil-attacks*. In our future work, we are going to implement and complete the *TrustRings* Network Model.

References:

[1] A. Abdul-Rahman and S. Hailes. A distributed trust model. *Proceedings of the 1997 workshop on New security paradigms*, 1997.

[2] I. F. Akyildiz, J. I. Pelech, and B. Yener. Virtual topology based routing protocol for multihop dynamic wireless networks. *Wireless Networks, Volume 7 , Issue 4 (August 2001)*, pages 413 – 424, 2001.

[3] B. Awerbuch, D. Holmer, and H. Rubens. Swarm intelligence routing resilient to byzantine adversaries. 2004.

[4] D. P. Bertsekas, and R. G. Gallager. Data Networks. *Data Networks,Prentice Hall, Englewood Cliffs,1987*.

[5] L. Eschenauer, V. D. Gligor, and J. S. Baras. On trust establishment in mobile ad-hoc networks. *ACM Conference on Computer and Communications Security*, pages 41–47, 2002.

[6] L. Georgiadis, P. Jacquet, and B. Mans. Bandwidth reservation in multihop wireless networks: Complexity and mechanisms. *24th International Conference on Distributed Computing Systems Workshops - W6: WWAN (ICDCSW'04)*, pages 762 – 767, 2004.

[7] YC. Hu, and A. Perrig, and DB. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks. Springer Netherlands: vol 11 (pages: 21-38)* .

[8] S. M. Huffman and M. H. Reifer. United states patent: Method for geolocating logical network addresses (6,947,978). 2005.

[9] P. Jacquet and P. Mhlethaler and A. Qayyum Optimized Link State Routing Protocol. *Published Online,http://www.ietf.org/proceedings/98dec/I-D/draft-ietf-manet-olsr-00.txt*,1998.

[10] T. Jiang and J. S. Baras. Ant-based adaptive trust evidence distribution in manet. *Proceedings of MDC*, 2004.

[11] T. Jiang and J. S. Baras. Cooperative games, phase transition on graphs and distributed trust in manet. *In Proceedings of 43rd IEEE Conference on Decision and Control*, 2004.

[12] DB. Johnson, and DA. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing, Kluwer Academic Publishers, 1996,http://www.ics.uci.edu/ atm/adhoc/paper-collection/johnson-dsr.pdf*.

[13] A. Josang. An algebra for assessing trust in certification chains. *Proceedings of the Network and Distributed Systems Security*, 1999.

[14] A. Josang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. *Proceedings of Australasian Computer Science Conference*, 2006.

[15] A. Josang, C. Keser, and T. Dimitrakos. Can we manage trust? *Proceedings of iTrust*, 2005.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. *In Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[17] P. Lamsal. Understanding trust and security. *Department of Computer Science, University of Helsinki, Finland*, 2001.

[18] C. E. Perkins and E. M. Royerl. Ad-hoc On-Demand Distance Vector Routing. *In the Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, WMCSA, New Orleans, Lousiana, USA*.

[19] C.E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *ACM Conference on Communications Architectures, Protocols and Applications, SIGCOMM '94, London, UK*.

[20] A. Perrig, R. Canetti, B. Briscoe, J. Tygar, and D. X. Song. TESLA: Multicast Source Authentication Transform. Work in progress. *Internet Engineering Task Force, 2005*.

[21] A. A. Pirzada and C. McDonald. Establishing trust in pure ad-hoc networks. *ACM International Conference Proceeding Series; Vol. 56*, pages 47 – 54, 2004.

[22] D. Spiewak and T. Engel. Trust as foundation for follow-on security mechanisms in manets. *WSEAS Transactions on Communications, Issue 1, Volume 6*, pages 125–131, 2007.

[23] D. Spiewak, T. Engel, and V. Fusenig. Unmasking threats in mobile wireless ad-hoc networks settings. *WSEAS Transactions on Communications, Issue 1, Volume 6*, pages 104–110, 2007.

[24] D. Spiewak, and T. Engel. Trusting the TrustModel in mobile wireless ad-hoc network settings. *In the Proceedings of the 5th Int.Conf. on WSEAS INFORMATION SECURITY and PRIVACY (ISP'06), November 20-22, 2006, Venice, Italy, ISBN 960-8457-56-4*.

[25] M. R. Thoppian and R. Prakash. A distributed protocol for dynamic address assignment in mobile ad hoc networks. *IEEE Transactions on Mobile Computing Vol. 5, No. 1*, pages 4 – 19, 2006.

[26] C-K.Toh A Novel Distributed Routing Protocol To Support Ad hoc Mobile Computing *n the Proceedings of IEEE 15th Annual International Phoenix Conference on Computers and Communications, IEEE IPCCC 1996, March 27-29, Phoenix, AZ, USA*.

[27] G. Zapata, Manel. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. *Technical University of Catalonia (UPC): NTERNET-DRAFT draft-guerrero-manet-saodv-06.txt, September 2006*.

[28] P. R. Zimmermann. The official pgp user's guide. *Department of Computer Science, University of Helsinki, Finland, MIT Press*.