City Research Online

City, University of London Institutional Repository

Citation: Michalas, A., Oleshchuk, V. A., Komninos, N. and Prasad, N. R. (2011). Privacy-preserving scheme for mobile ad hoc networks.. Paper presented at the 2011 IEEE Symposium on Computers and Communications (ISCC), 28 June - 1 July 2011, Kerkyra, Greece.

This is the unspecified version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/2485/

Link to published version: http://dx.doi.org/10.1109/ISCC.2011.5983930

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online: http://openaccess.city.ac.uk/ publications@city.ac.uk

Privacy-preserving Trust Establishment scheme for Mobile Ad Hoc Networks

Antonis Michalas, Vladimir A. Oleshchuk, Nikos Komninos, and Neeli R Prasad

***Abstract*—This paper proposes a decentralized trust establishment protocol for mobile ad hoc networks (MANETs), where nodes establish security associations. In order to achieve privacy and security, we use homomorphic encryption and polynomial intersection so as to find the intersection of two sets. The first set represents a list of recommenders of the initiator and the second set is a list of trusted recommenders of the responder. The intersection of the sets represents a list of nodes that recommend the first node and their recommendations are trusted by the second node. In our experimental results we show that our scheme is effective even if there are 30 trusted nodes.**

***Index Terms*—Trust, Homomorphic Encryption, Reputation System, Mobile Ad-Hoc Networks.**

I. Introduction

THE past few years' computers started becoming more and more important for our daily round, until they became an inextricable part of our lives. As a result, users started to set also new demands for connectivity. Wired solutions were just not enough, since there was an increasing request on for connecting to the Internet, reading and sending E-mails and changing information from anywhere in general. The solution to that wish is offered by the so called ad-hoc networks.

Unlike common wireless networks, ad-hoc networks are characterized by the absence of any existing network infrastructure or centralized administration (decentralized wireless network) as well as the ease and speed of deployment. Such networks are highly dynamic and each node participates in the basic functions of the network like packet forwarding and routing, since there are no routers or access points. Ad-hoc networks can operate in a stand-alone way or be attached to a larger network.

The decentralized nature of ad-hoc networks makes them appropriate for a variety of applications where infrastructures that support central nodes are not suitable, and may expand the scalability compared to typical wireless networks. In addition, ad-hoc networks are suitable for situations where there is lack of an infrastructure. An ad hoc application is a self-organized application composed of mobile and autonomous devices interacting as peers, whose connections are made possible because of fairly close distances.

Applications of ad-hoc networks range from military tactical operations, to crisis management services, such as in disaster recovery, where the whole communication infrastructure is ruined and resorting communication rapidly is critical as well as data gathering from sensor networks, or even instant messaging/meeting applications.

A. Our Contribution

In this paper we present a scheme for trust-based communication between nodes of an ad-hoc network. In order to do so, we use a reputation system that each node takes into consideration before it decides to serve another node of the network. The proposed trust establishment scheme does not use any central trust authority, since this a superfluous requirement for ad-hoc networks. Apart from that, we make use of Homomorphic Encryption so as to achieve better security and privacy for our model. In addition, possible malicious nodes will be responsible for making the most expensive computations in order to acquire some of the resources of another node in the network.

Following this introduction, the paper is organized as follows. In Section 2 we briefly discuss about different attacks on ad-hoc networks, in Section 3 we analyze why a Trusted model is important for the secure function of a network, in Section 4 we examine related works that have been made in order to secure ad-hoc networks with the use of a Trusted Model. In Section 5 we describe our new technique, that combines *Homomorphic Encryption* with a *Reputation System* so as to achieve better security in an ad-hoc network, while in Section 6 we briefly discuss the advantages and disadvantages of our technique. In Section 7 we present our experimental results and we conclude the paper in Section 8.

II. Attacks in Ad-Hoc Networks

Ad-Hoc Networks and more precisely Mobile Ad-Hoc Networks (MANETs) have inherently different properties than traditional wired networks and thus new kind of vulnerabilities/attacks have been arise. In this section we are going to briefly describe different types of attacks in order so as to have a picture why it is important to secure MANETs.

In such networks we find two different types of attacks, passive and active. In passive attacks, the attacker tries to collect important information about the network by eavesdropping the routing traffic. By doing that, the attacker can be in position for example to understand which nodes are playing the most crucial role in a network, and try to compromise them in order to bring the whole network down. In general passive attacks do not interfere with the stored data and there are very difficult to detect since the attacked entity is unaware of the attack.

In active attacks the attacker has to spend some of his resources so as to successfully change the content of a packet or to just disrupt other nodes of the network. This kind of

TABLE I
ATTACK CATEGORIES

Passive	Active
Traffic Analysis	Denial of Service
Passive Eavedropping	Man in the Middle Attack
	Unauthorized Access
	Replay Attack
	Session Highjacking

attacks can result to several losses for the nodes of the network. In Table I a list of both passive and active attacks are presented.

III. TRUST

Security and Trust are two notions that are strongly connected to each other. Theses two notions are overlapping but not coinciding. An entity (node) can be trusted, if operates exactly as designed and expected and works without disruption. In order to understand the term "trust an entity", we firstly differentiate two significant notions of "target of trust" and its "classification". "Target of trust" is the actual entity we trust, while the "classification" defines precisely why the entity is being trusted for. Furthermore, there might be a value of trust which describes how much we trust an entity according to some criteria [2].

Third-Party Trust refers to a situation in which two nodes trust each other even though they have not previously exchanged packets or established any kind of communication. In such a situation, these nodes can trust each other if and only if they have a relationship with another node in the network that they trust, and that node guarantees for the trustworthiness of the other node. *Direct Trust* is when two nodes have already established a trusting relationship and they can communicate to each other immediately (at least for a specific period of time), without the need of a third-party.

In wired networks is much easier to create dominant Trust Management Systems since nodes in do not have limitations such as energy consumption or limited computational resources. Furthermore, the topology of wired networks is not changing dynamically as in ad-hoc networks. So, Trust Establishment in ad-hoc networks it still remains an open issue.

IV. RELATED WORK

Trust establishment is concerning scientists for many years and lot of schemes have been developed to address that issue, not only in wired but also in wireless ad-hoc networks. We can distinguish the different approaches into two categories. Those who use a central authority and those who use a self-organized approach.

J. Sen *et al* in [3] presented a trust establishment scheme for ad hoc networks based on distributed trust model. A trust initiator was introduced only in the system-bootstrapping phase to initiate the protocol. A fully self-organized trust establishment approach was then adopted to handle the dynamic topology of the network and the membership changes of the nodes, while ensuring trust establishment among the nodes with shorter trust chains and very high probability.

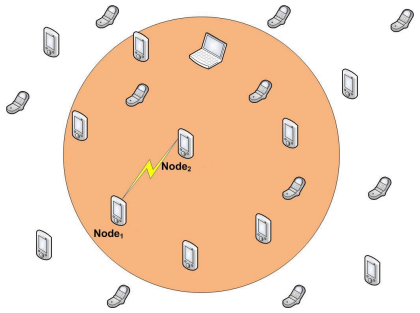

Fig. 1. Small scale ad hoc network.

C. Papageorgiou *et al* in [4] proposed a dynamic trust establishment protocol that allowed nodes of an ad hoc network to establish security associations among each other in a distributed and peer-to-peer manner. The basis of the protocol was a node-to-node security handshake using a network-wide key that every node was preconfigured with.

A.A. Pirzadapresent and C. McDonald, in [1] presented a model for trust-based communication in ad-hoc networks which introduced the notion of belief and provide a dynamic measure of reliability and trustworthiness in an ad-hoc network.

A. Josang in [5] showed that authentication can not be based only on public key certificates, but also needs to include the binding between the key used for certification and its owner, as well as the trust relationships between users. Furthermore, he developed a simple algebra scheme and described how it can be used to compute measures of authenticity.

M. Raya *et al* in [6] argued that the traditional notion of trust as a relation among entities, becomes insufficient for emerging data-centric mobile ad hoc networks and they proposed a new framework for data-centric trust establishment.

V. OUR TRUST MODEL

The proposed Trust Model does not based on any central Trust Authority, instead every node of the network, is responsible to recognize if the node who is making a request is *Trustful*, *Malicious* or there is no information about its behavior (*Uncertainty* level).

If we consider the example of Figure 1, where $node_1$ wants to communicate with $node_2$, then $node_2$ is the one who has to find if it can trust $node_1$ or not. In order to do so, when $node_2$ receives the request, must first check if it has any information about this node from a previous communication. If the answer is "yes" then it simply checks the trust level that has been assigned to $node_1$. In general we recognize the following cases:

- *Trustful* then $node_2$ serves the request.
- *Malicious* it sends a computational problem to $node_1$, and wait for a time interval t (for example $t \leq 10$sec) for $node_1$ to solve it. If $node_1$ solves the problem correct and within the time interval t then $node_2$ serves the request, otherwise drops the connection.

- *Uncertainty*, which means that they had never had a communication in the past, $node_2$ asks from $node_1$ to send which of the nodes in the network can give a recommendation for $node_1$.

So, if for example $node_3$, $node_4$ and $node_5$ had established a connection (trustful relationships) in the past with $node_1$, $node_1$ would respond to $node_2$ by sending a set with the id's of these nodes. Then if $node_2$ has previously established trustful relationships with any of these nodes ($node_3$, $node_4$ or $node_5$) it can ask them to give their opinion about $node_1$. In all cases, the total level of trust for $node_1$ will have to be *Trustful*, so as to be served immediately (i.e $node_3 \rightarrow Trust$, $node_4 \rightarrow Malicious$, $node_5 \rightarrow Trust$). In any other case, we use the options we described later in this section.

A problem that arise here is the fact that it is not secure to expose list of trusted nodes and their recommendations. It could be used by an attacker to collect information about node's trustful relations and also to create some privacy issues. Hence, we use homomorphic encryption to force the communication initiator to spend more computational resources than the one who is receiving the request.

Homomorphic encryption is a form of encryption that permits performance of a specific algebraic operation (denoted by $\otimes$) on the plain text by performing a (possibly different) algebraic operation (denoted by $\oplus$) on the corresponding ciphertext. The homomorphic cryptosystems are used as a basic building block in many secure multiparty protocols. Several such cryptosystems have been proposed in the literature [8], [9]. More formally, let us consider a public-key cryptosystem with the homomorphic property where encryption and decryption are denoted as E(.) and D(.) respectively. It means that there is an operation on encrypted data, denoted as $\oplus$, that can be used to perform summation on the encrypted data without decrypting them. Thus, we can find the encrypted sum of encrypted x and y; that is, $E(x) \oplus E(y) = E(x \otimes y)$. Consequently, we are able to multiply encrypted data if only one of the multipliers is unencrypted. Homomorphic cryptosystems proposed in the literature define operation $\otimes$ as modular multiplication while $\oplus$ is defined as modular addition or XOR. As a simple example of a homomorphic cryptosystem, we can consider the RSA cryptosystem. It is easy to see that $E(x_1) \oplus E(x_2) = (x_1^e mod n)(x_2^e mod n) = x_1^e x_2^e mod n = (x_1 x_2)^e mod n = E(x_1 \oplus x_2)$, where (e, n) is a public key. In this case, both $\oplus$ and $\otimes$ are modular multiplications. However, in the context of secure multi-party computations, the most used cryptosystems define $\otimes$ as a modular addition [8], [9].

Our solution is based on proposed in [7] approach to find privacy preserving intersection of two sets. Assuming that the first set represents a list of recommenders of $node_1$ and the second set is a list of trusted recommenders of $node_2$. Then the intersection will represent a list of nodes that can recommend the first node and their recommendations are trusted by the second node.

A. Construction of the Polynomial

In order to find the intersection of two sets, we use the method that Freedman *et al* proposed in [7].

We assume that each node has selected public and private keys according to some homomorphic public-key crypto scheme. Let $E_{n_i}(m)$ denote an encryption of m with public key of node n_i, and $D_{n_i}(m)$ denote a decryption of m with private key of node n_i. We assume that $E_{n_i}(x + y) = E_{n_i}(x) \oplus E_{n_i}(y)$.

Lets assume that $node_1$ has k nodes that can recommend it. So the set with the recommenders of $node_1$ is defined as follows:

$$R_1 = \{n_3, \cdots, n_{k+2}\} \tag{1}$$

Because, we don't want to expose the content of R_1 to possible passive attacker(s) that may intercept the message(s) and to the receiver ($node_2$), we construct polynomial P of degree k from the set R_1, such that the roots of P will be the elements of R_1. As we can see from (2) the roots of P are the elements of set R_1.

$$P(x) = (n_3 - x) \cdot (n_4 - x) \cdots (n_{k+2} - x) \tag{2}$$

$$\Rightarrow P(x) = \sum_{i=0}^{k} a_i \cdot x^i \tag{3}$$

After $node_1$ has defined polynomial $P(x)$, it encrypts the coefficients with its public key and sends to $node_2$ the encrypted coefficients of this polynomial. $Node_2$ receives a set of the encrypted coefficients (4):

$$E_{n_1}(R_1) = \{E_{n_1}(a_k), E_{n_1}(a_{k-1}), \cdots, E_{n_1}(a_0)\} \tag{4}$$

Similarly, we suppose that $node_2$ have l nodes that it trusts, represented by set R_2:

$$R_2 = \{n_{k+2}, n_{k+3}, \cdots, n_{k+l+1}\} \tag{5}$$

The two nodes have to find the intersection of sets R_1 and R_2, in order to know which 'friends' they have in common. But this cannot be found from $node_2$ (at least not without spending lot of computational resources) since the list it received is encrypted. So $node_2$ makes use of the homomorphic encryption properties, to calculate encrypted values of $P(x)$ for each x from R_2 without knowing coefficients of $P(x)$ in clear text or result of calculation.

That is, for each node n_j from R_2, $node_2$ can calculate polynomial without decrypting coefficients as follows:

$$E_{n_1}(P_1(n_j)) = E_{n_1}(a_k) n_j^k \oplus \cdots \oplus E_{n_1}(a_1) n_j \oplus E_{n_1}(a_0)$$

So, $node_2$ creates the following set:

$$E_{n_1}(R_2) = \{E_{n_1}(P(n_{k+2})), \cdots, E_{n_1}(P(n_{k+l+1}))\}$$

$E_{n_1}(R_2)$ is sent back to $node_1$ who has to decrypt it in order to find if the two sets intersects ($P(x) = 0$ for at least one element x from R_2)

In our case it's clear that:

$$R_1 \cap R_2 = n_{k+2}, \tag{6}$$

which means that the only common 'friend' they have is n_{k+2}. So, $node_1$ will find its common friends with $node_2$ (in our example only node n_{k+2}) and it will ask them to give their reputation opinion about $node_2$. Since $node_1$ is communicating with with node(s) that have previously established a trustful relationship, the cost of the communication is significantly small. The same also holds for the next step where these nodes will communicate with $node_2$ in order to give their reputation for $node_1$.

VI. Security Discussion

Reputation adjustment is a very crucial issue in trust establishment. There should be a standard way known to all nodes so as to facilitate the assignment of the reputation values for different kind of activities. In our approach, we have only three levels of reputation (Trustful, Malicious and Uncertainty) and we do not use scoring like many other techniques. However more advanced techniques proposed in literature can be used here. Apart from that, in order to compete with situations where trusted nodes have been compromised, we make use of a time trust interval t_{Trust}. If a node is considered to be trusted and initiates a connection after t_{Trust} passed, then its reputation level is considered as Neutral. This technique provides security in two different directions. First we make use of the concept that if someone is trusted "today", it does not mean that he should be also trusted "tomorrow". The second issue that we solve overcomes a security flaw that exists in many systems with reputation scoring. If we use a score for each known node, then an attacker could either compromise a sufficient number of nodes with very high scores or alternately could actively participate in the network for a period of time for the purpose of gaining reputation points. These two methods could lead to powerful attacks, as each node's reputation will take time to decrease, allowing the attacker to operate with impunity for a longer period of time before he is labeled as malicious.

In addition to three tier scoring, we propose a security system based on social networks approaches with a feedback structure. Initial communication between nodes will be determined according to mutually established "friends" on the network as well as the requesting node's feedback by these friends. To facilitate secure communication of information about other members of the network (the "friends"), we use homomorphic based encryption scheme. In this method, the requesting node sends an encrypted list of its friends to the target node. The target node, in turn, adds its own list of friends to the encrypted message and sends it back. The properties of homomorphic encryption ensure that this stage requires negligible resources from the target node. Having received the encrypted combined lists of friends, the initiating node checks the lists to find any mutual friends. Having found mutual friends, that node proceeds to communicate with them, requesting that they in turn verify to the target node that it is to be trusted. That all nodes involved in this procedure are trusted friends ensures a quick resolution of the security check. If the initiating node did not find any mutual friends in the lists, it informs the target node about it, resulting in the usual communication between neutral nodes.

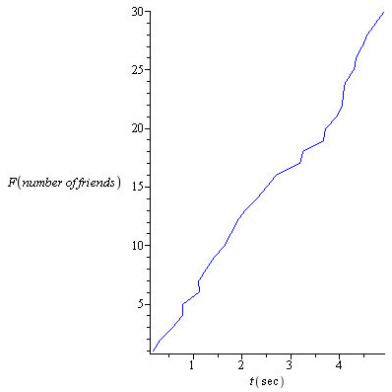

Fig. 2. Total time for two nodes to find their common friends.

VII. Experimental Results

In this section we describe the implementation of the proposed Trust Establishment scheme.

In order to measure the effectiveness of our solution, we have used the Smart Dust simulator written in Java. Our testbed consists of a laptop computer with Intel Core Duo CPU P7450 at 2.13GHz, 6.00 GB of RAM running Windows 7 64-bit. For the purpose of our experiments, we constructed an instance of the Paillier's cryptosystem with 512 $bits$ of modulus. Our experiments were implemented in a medium scale ad hoc network as seen in Figure 1. When nodes wish to communicate with their neighbors, they broadcast a message, which is processed by the neighboring nodes.

In our experiment we first of all measured the total time needed for the encryption, decryption as well as the intersection of the polynomials in order to find the common friends. The results of this experiment can be found in Figure 2. Furthermore, Table II shows some representative results from this experiment. As we can see, in the case where the polynomial is of degree 5 (which means that it has 5 friends) the time until $node_1$ finds the common friends with $node_2$ is $0.78sec$, while in the extreme case where the degree of the polynomial is 30 the time is $4.92sec$ which is also an affordable time even for devices like PDA's and mobile phones.

TABLE II
Results from Figure 2

Friends	Time (sec)
5	0.78
10	1.64
15	2.50
20	3.71
25	4.29
30	4.92

In our next experiment we measured the time that $node_2$ (the node who is receiving a request) has to spend in order to add its friends to the encrypted list of $node's_1$ friends that received. The results are presented in Figure 3 while Table III

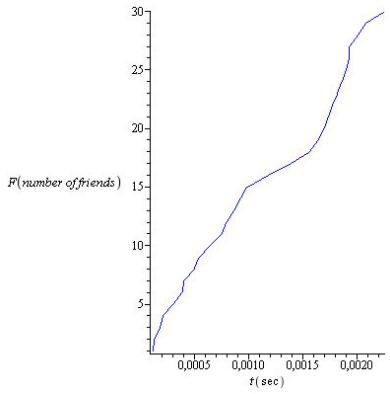

Fig. 3. Total time for $node_2$ to add its friends to the encrypted list of $node_1$.

shows some representative results from this experiment. From these results we can see that the time $node_2$ spends in order to share its friends list with $node_1$ is significant small, since even for the case where the polynomial is of degree 30 the time that $node_2$ spends is much less than $1sec$. This means that each node that receives a request is spending very little of its resources while it is also protected from possible attacks.

TABLE III
RESULTS FROM FIGURE 3

Friends	**Time (sec)**
5	0.000296418
10	0.00063518
15	0.000982122
20	0.001692848
25	0.001897043
30	0.002251126

VIII. CONCLUSION

In this paper, we proposed a robust Trust Establishment scheme for securing mobile ad hoc networks. Our scheme make use of public-key cryptography in order to protect/enhance privacy between nodes. Furthermore, it forces possible malicious nodes to spend some of their resources, so as to prove that they wish to be served as well as to improve their reputation in the network. In order to do that, we use homomorphic encryption and polynomial intersection, which provide each node with a higher level of security, since they do not exchange sensitive information in plain text. Our approach, aims at building confidence measures regarding route trustworthiness in nodes that are dynamically computed and modified based on effort - computational resources each node spends and passively observed by reputation from other nodes in the network.

REFERENCES

[1] A.A. Pirzada and C. McDonald, *Establishing trust in pure ad-hoc networks*, In ACSC '04: Proceedings of the 27th Australasian conference on Computer science (2004), pp. 47-54. Dunedin, New Zealand: Australian Computer Society, Inc., 2004.

[2] Kui Ren and Tieyan Li and Zhiguo Wan and Feng Bao and Robert H. Deng and Kwangjo Kim, *Highly reliable trust establishment scheme in ad hoc networks*, In Computer Networks (2004), volume 45, pp. 687 - 699. Elsevier, 2004.

[3] J. Sen, P.R Chowdhury and I Sengupta, *A Distributed Trust Establishment Scheme for Mobile Ad Hoc Networks*, International Conference on Computing: Theory and Applications (ICCTA'07), pp.51-58. Kolkata, India, 2007

[4] C. Papageorgiou, and K. Birkos and T. Dagiuklas and S. Kotsopoulos, *Dynamic trust establishment in emergency ad hoc networks*, IWCMC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing, pp.26-30 Leipzig, Germany, ACM, 2009.

[5] A. Josang, *An Algebra for Assessing Trust in Certification Chains*, Proceedings of the Network and Distributed Systems Security Symposium (NDSS'99). The Internet Society 1999.

[6] M. Raya and P. Papadimitratos and V.D. Gligor and J-P. Hubaux, *On DataCentric Trust Establishment in Ephemeral Ad Hoc Networks*, In IEEE INFOCOM, Phoenix, USA, 2008

[7] M. Freedman and K. Nissim and B. Pinkas, *Efficient Private Matching and Set Intersection*, In 'Proc. of Eurocrypt'04', Vol. 3027, LNCS, pp.1-19 Springer-Verlag, 2004

[8] P. Paillier, *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, In Advances in Cryptology EUROCRYPT '99, pp.223-238, Springer Berlin Heidelberg, 1999

[9] Naccache, D, Stern, J. 1998. A new public key cryptosystem based on higher residues. In: Proceedings of the 5th ACM Conference on Computer and Communications Security, CCS 98, pp.59-66, San Francisco, CA, 2-5 November 1998. New York, NY, ACM Press.