Progressive Authentication in Ad Hoc Networks

Raja Rai Singh Verma, Donal O'Mahony and Hitesh Tewari
{Raja.Verma, Donal.OMahony, Hitesh.Tewari@cs.tcd.ie}
Network and Telecommunications Research Group (NTRG)
Department of Computer Science, Trinity College
Ireland

Abstract

In this paper we address the problem of trust formation and secure communications in ad-hoc networks. The dynamic nature and frequent membership changes make it difficult to form trust relationships in ad hoc networks. We intend to build such relationships by progressive exchange of certificates and the resulting trust may be complete or partial in nature. The trust formed is used by a node to access services provided by the remote node. A two-tier key formation scheme is proposed to ensure the confidentiality of the trust negotiation process and to secure the communication paths.

Keywords: Trust Negotiation, Authentication, Ad Hoc Network.

1. Introduction

Ad hoc networks have little or no fixed infrastructure making their deployment fast and easy. The frequent membership changes coupled with wireless as the principal form of communication makes the topology of an ad hoc network dynamic. Ad hoc networks are also prone to partitioning which makes building of trust relationships difficult. Trust relationships are essential to determine access control for services some of which may be scarce or sensitive. These relationships cannot be prearranged as they may change frequently due to the dynamic nature of an ad hoc network. Thus a flexible trust negotiation scheme is needed, which can build dynamic trust relationships that can be used to determine access control for services.

To illustrate the complexity in building trust relationships in ad hoc networks, consider an emergency scenario. Let there be an incident on road junction J10. First on the scene are the police patrol cars. When the seriousness of the problem is realized other specialized units like narcotic unit, paramedics and armed response units are called. At the same time the different news units also converge on the spot. Once on the spot the narcotic unit takes charge and other units coordinate with them. Any press information released by any officer on the spot has to be first checked from the leader of the narcotic unit. Any armed emergency on the spot will have to be coordinated with the armed response and patrol cars. The paramedics and the police units can call for more information from their respective bases but will have to wait for go ahead form the narcotic leader. As usual the press units will try to eavesdrop on information and would like to coordinate amongst themselves. The narcotic units will have their own information which they will want to keep among themselves. In the above mentioned scenario the services are located on both the mobile and the fixed nodes.

The above scenario calls for the same entity to enter into multiple trust negotiations. Trust is built progressively between entities with the exchange of certificates. Some certificates are confidential and are released only when partial trust has already been established. These sensitive certificates have to be adequately protected with checks in place prior to their disclosure to another node. Also during trust negotiations the same certificate can be used in different contexts and roles. In the example presented earlier, it can be seen that the police department identification can be the primary negotiating certificate for trust in one case, while in the other it may just be used in a supporting role with other certificates. The entire process of trust negotiation has to be resistant to eavesdropping, replay and impersonation.

Our scheme deals with two kinds of trust relationships, complete and partial. In our scheme there is an association between *trust levels* and the services to be provided to other nodes. So a partially trusted node gets a subset of the services provided by a host. To ensure confidentiality of the trust negotiation process we put forward a two-tier key formation procedure. The first is aimed at external attackers and the other towards internal *rogue users* [1]. The trust negotiation scheme is transparent to the user and needs minimum intervention. User intervention is confined to deadlock resolution and decisions on acceptance of new certificate issuers. To function in the hybrid scenario mentioned above the scheme should be able to interact with the conventional networks.

The paper is organised into the following five sections. Section 2 gives the background and the state of art information while Section 3 presents our scheme. Section 4 deals with the details of implementation. Section 5 contains conclusions and suggestions for future work.

2. Background

This section starts with a brief background study of the access control and trust negotiation systems. Then we move on to present criteria used to determine which key formation scheme is best for ensuring the correctness and confidentiality of the trust negotiation process.

2.1 Access Control and Trust Negotiation Schemes

Most access control systems rely on public key management systems to certify an association between an identity and a key in form of a *digital certificate*.

"This material is based upon work supported, in part, by the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory, under Contract No. F61775-01-WE052"

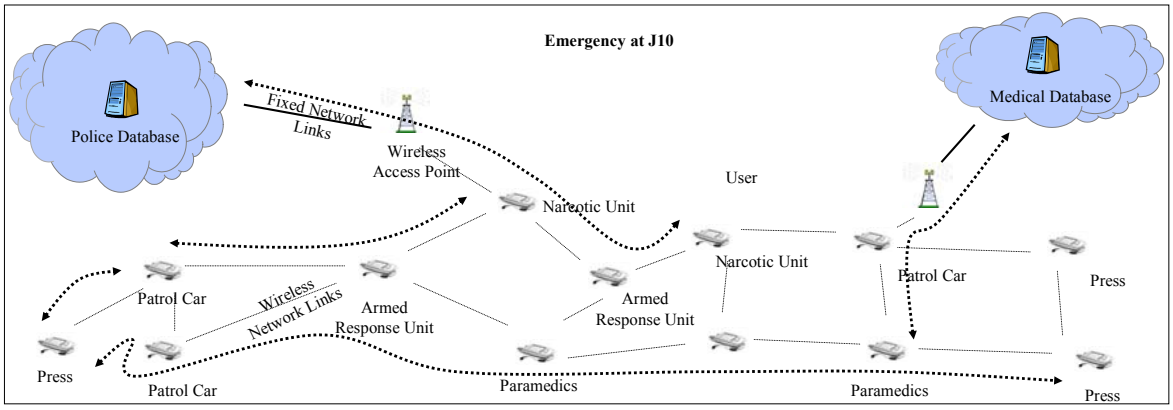

Figure 1. Sample Scenario

These certificates contain the public key and the identity along with other details cryptographically signed by a trusted third party. We classify these systems by whether they were designed for conventional networks or ad hoc networks.

2.1.1 Key Management in Conventional Networks

The two main public-key management solutions are Pretty Good Privacy (PGP) [2] and the X.509 public key infrastructure [3]. PGP has an anarchic organization in contrast to a rigid hierarchy of X.509. In PGP though there are some central certificate repositories these are not much used. In X.509 there is a hierarchy of Certification Authorities (CA) which are responsible for the issuing of certificate and their verification. A node verifies the authenticity of a certificate by using the public key of the CA. The CA may revoke a certificate and periodically release a Certificate Revocation List (CRL) containing references to the revoked certificates. Delays in the release of a CRL may lead to the acceptance of some revoked certificates by nodes in the network. In ad hoc networks this approach is difficult to operate as access to a CA cannot be guaranteed at all times to obtain the latest CRL. In PGP a certificate's trustworthiness is assigned by the user using it. This process is made difficult in PGP as most of the certificates are self-signed and their trustworthiness needs to be verified by the user. The process to estimate the trustworthiness of a certificate may be prolonged and difficult in an ad hoc network.

2.1.2 Key Management in Ad Hoc Networks

The key management approaches for ad hoc networks try to eliminate the need for a centralized CA [3]. The first approach described below emulates a conventional CA by distributing it on several nodes. In the second approach each node authenticates the other using some prefixed criteria, while in the last approach a self-organized public-key infrastructure is used.

2.1.2.1 Distributed Certification Authority

Zhou [1] has proposed a key management scheme for ad hoc networks using threshold cryptography and the public-key paradigm. The scheme provides for distribution of parts of the secret key among some special ad hoc nodes designated as *servers*. An attacker has to break into a *threshold number* of servers in order to get access to the secret key of the service. To prevent progressive compromise of servers *share refreshing* is done periodically. This scheme requires prior communications and coordination between the nodes for setting up the service. Also, in this scheme some nodes (namely the *servers*) will have to work more than other nodes. Furthermore the requirement for each server to know the public key of all nodes is difficult if the number of nodes in the ad hoc network is large.

2.1.2.2 Pre-arranged Shared Secret

This approach is based on the existence of a shared secret among the nodes in the ad hoc network. Individual nodes in the network use the shared secret to generate their respective keys. One such scheme proposed by DeCleene [4] has a hierarchical framework. Each area in the hierarchy has a controller. These area controllers re-key a node when it moves between different "areas". Another scheme proposed by Kong [5] uses the emulation of certification authority and shared secret model along with a Public Key Infrastructure (PKI) based centralized model. Initially the scheme has an aerial node acting as the centralized node for key distribution. If this aerial node is destroyed the scheme uses threshold cryptography based on secret sharing to emulate a distributed certification authority.

2.1.2.3 Self-Organized Public-Key Infrastructure

Hubaux [6] proposed a public-key distribution based trust building scheme for ad hoc networks which is similar to the PGP *web of trust* concept. The scheme differs from PGP as there are no central certificate directories for distribution of certificates. Instead a user selects a subset of certificates from its repository to disclose to the other user. Both the users then merge the received certificates with their own certificates. In order to find the public-key of a remote user the local user makes use of the Hunter Algorithm [6] on the merged certificate repository to build certificate chain(s). A certificate trust chain should lead from the local user certificate to the remote user's certificate. The local user uses the public-key contained in the remote user's certificate. The probability of finding such a certificate chain in this scheme is high but is not guaranteed. This decentralized scheme leads to disclosure of too much information about the originating node as it releases several unnecessary certificates, which may not be needed in chain formation.

Until now we have concentrated on identity certificate based schemes. There are two other certificate types - namely the capability and property certificates. An identity certificate merely binds names to keys, while a capability certificate has embedded authorizations in it allowing the owner (client) to perform certain authorized actions on resources of the issuing server. The third and most generalized type of certificate is the property-based certificate. A property based certificate has the ability to embed arbitrary property name/value pairs into the certificate. Property based certificates are relatively new compared to the other two and can be used to express both the identity and capability certificates. The best example of identity certificate based systems is version 1 of X.509 [3]. Version 3 of X.509 which supports arbitrary attribute name/value pair is property certificate based but is primarily used as an identity certificate on the Internet. Capability certificate based systems like the IETF Simple Public Key Infrastructure (SPKI) [7] and Keynote scheme of Blaze [8] restrict the context in which a certificate can be used in authentication and authorization. The client's certificates in SPKI and Keynote systems contain embedded access permissions for services on the issuing server. Therefore the certificate is only valid on the issuing server.

2.1.3 Trust Negotiation Systems

Winsborough [9] has proposed an "Automated Trust Negotiation" scheme which tackles the problem of exchange of property certificates for authentication and authorization between strangers. Under this scheme, an unknown client can get access to server resources by presenting certificates that match those required by the server's access policies. This trust negotiation scheme allows the progressive disclosure of certificates between the client and the server. Each node has its negotiation strategy for the disclosure of certificates. Further, the scheme assumes presence of an online certification authority to authenticate the certificates. Ad hoc networks are peer-to-peer oriented where the access to an online certification authority is not guaranteed. As such this scheme is unsuitable for ad hoc networks.

2.2 Key Establishment Protocols

The correctness and confidentiality of the trust negotiation process is ensured by employing cryptographic techniques. This involves a key establishment between two entities participating in trust negotiation. In this section we will concentrate more on the key agreement protocols than on key transport protocols [10], as all parties participating should have a say in the parameters used in key computation. A good key agreement scheme [11] should have

- Minimal number of message exchanges
- Low communication overhead
- Low computational overhead using pre-computation to minimize online computational overhead

Other desirable features of a key agreement protocol should be that it there is no reliance on time stamping as clock synchronization is difficult to achieve in ad hoc networks. Mutual authentication of the two entities participating in the key exchange is desirable. A comparison of the major key agreement schemes compared by Boyeon Song [10] and that forms a basis of our selection. On further examination of the computation times of the key agreement schemes we concluded that the Diffie-Hellman Station-to-Station (STS) protocol [12] best suits our goal. The key features of the STS protocol are

- Requires only three public-key operations by each of the participants
- Both parties contribute to the parameters used in key formation
- Just three messages are exchanged between the entities
- Mutual authentication of the communication partners takes place
- Mutual key confirmation is done between the communication partners
- The protocol is resistant to the man-in-the middle attack

The trust negotiation process in our scheme is based on the scheme by Winsborough [9] which we adapt for ad hoc networks. Our scheme is also able to function for short periods of time in the absence of an online CA. This can operate with conventional X.509 or the distributed certificate authorities [1]. The message exchanges are encrypted using a key formed by a STS key agreement protocol.

3. Trust Management

The trust negotiation process between two users requires the messages to be routed through the network. Consequently the trust negotiation has to be positioned over the routing layer. However this leaves the routing information susceptible to misinformation attacks (i.e. packets may be intentionally changed or false packets inserted). Therefore a two tier solution is proposed with Neighbor Trust Management (NTM) and Remote Trust Management (RTM) layers respectively. The NTM layer encrypts messages between neighbors (i.e. all nodes within broadcast range who can reply). In the NTM layer the traffic encryption key (K_{NTM}) formed has a short lifetime and key length due to the fact that nodes may be highly mobile. The RTM layer's traffic encryption key (K_{RTM}) has a longer lifetime and key length.

Figure 2 illustrates how the key formation occurs in a representative network. The peer-to-peer keys K_1 to K_4 are formed between the respective nodes. If an external attacker "X" tries to listen to the traffic in the network it has to guess the keys K_1 and K_2 (these are the peer-to-peer keys in "X" listening range). Even breaking these keys in real-time will yield the attacker limited local information. Suppose the internal node "R" is compromised and decides to modify or falsify information it is routing between *user1* on node "A" and *user2* on node "D" the end-to-end key "K" prevents it from doing so.

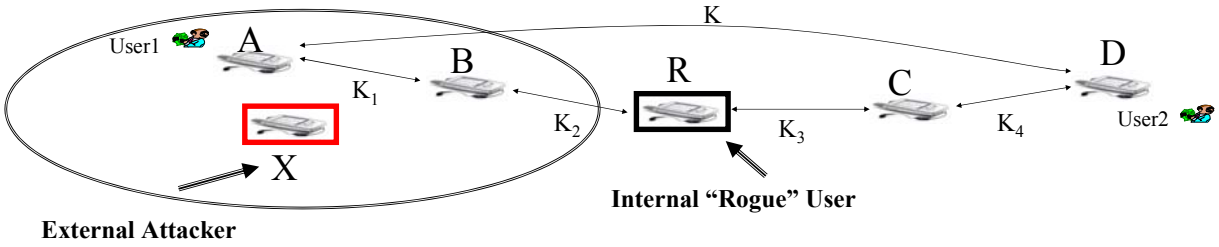

Figure 2. Key Agreement in NTM

3.1 Remote Trust Management

In the real world when two strangers meet they build trust in each other by exchanging paper credentials (e.g. business cards, passports etc.). Similarly in ad hoc networks nodes are *strangers* and to build trust they exchange property based digital certificates in our scheme. As in the real world sometimes the trust formed can be partial. Our scheme interprets partial trust as access to a subset of the services a node provides. The use of property based certificates makes it easy to encode paper credentials. In our scheme the certificates can be used in the trust negotiation in a different role instead of the context the certificate was originally issued.

If a node is relocated to a new area of operation it can build new trust relationships by exchanging certificates. This flexibility means that there is no need for reconfiguration of a node. It must be noted that in our scheme we discourage the use of self-signed certificates as these need to be individually verified by the user. This may be a difficult task in ad hoc networks.

To ensure the correctness and confidentiality of the trust negotiation process an end-to-end traffic encryption key K_{RTM} has to be formed. The authentication for the K_{RTM} formation is done using an *identity certificate*. In effect the identity certificate is a certificate designated by the user of a node to identify itself to remote users. It should contain an attribute "identity" and the corresponding value (e.g. CN="tom"). This key formed is used to encrypt the trust negotiation by exchange of certificates.

The Certificate Exchange Agent (CEA) in the RTM layer performs the trust negotiation process. It receives the message from the layers above or below and acts on them according to the programmed negotiating strategy. This strategy is determined by inputs from the other data structures in the layer shown in figure 3. The Local Certificate Store (LCS) contains the local certificates to be used in negotiation along with trusted certificate issuers and their respective Certificate Revocation List (CRL) [3]. The Certificate Release Policy (CRP) contains the release policies for each of the certificate in the LCS which may be used for trust negotiation. The Certificate Exchange History (CEH) keeps a record of the old trust negotiations for ease of future negotiations. The mapping between the services allowed and the node to which it is allowed is kept in the Service Access Table (SAT). It also contains the K_{RTM} keys that have been negotiated with a remote node. The Service Policy (SP) contains the association between the certificates required and the services available on the node.

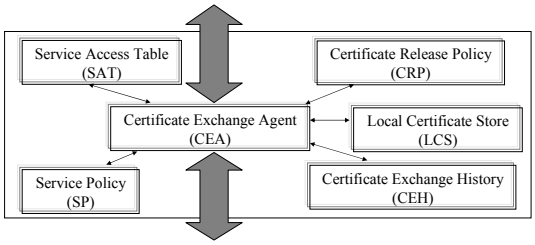

Figure 3. Internal Organization of the RTM Layer

3.1.1 Trust Negotiation Protocol

The trust negotiation between two nodes can be triggered in two ways. A node can explicitly ask for trust negotiation from a remote node by sending it a *Service Request* message. It contains a Unique Identifier (UID) to identify the trust negotiation. This UID will be used in all subsequent messages in the trust negotiation exchange. The *Service Request* message also contains the service(s) the node initiating the process wants to access on the remote node. The remote node replies with a *Service Available* message with the services that require negotiation along with the service(s) which are already unlocked. This option of starting a trust negotiation process is depicted in option 1 of Figure 4. The K_{RTM} negotiation is done simultaneously using the STS protocol. The STS key agreement protocol first and second messages (STS1 and STS2) ride piggyback on the *Service Request* and *Service Reply* messages. The third STS protocol message (STS3) is sent on its own.

An implicit trust negotiation can be initiated if a

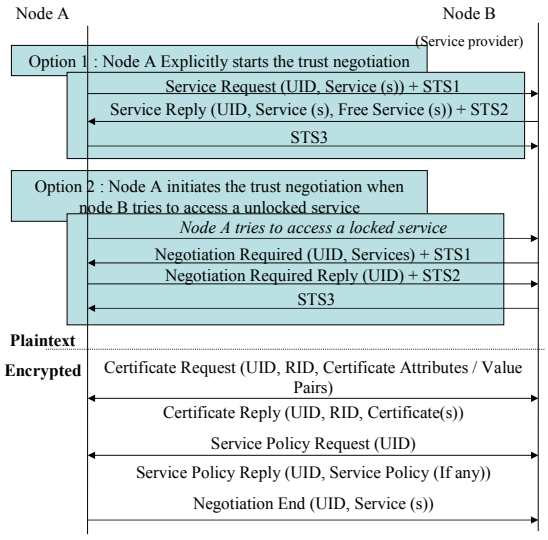

Figure 4. RTM Layer Trust Negotiation Protocol

node tries to access a locked service on a remote node. In this case the remote node commences the trust negotiation process. It sends a "Service Request" message to the node requesting the unlocked service. This message contains the requested service(s) along with a UID. The response to this is a "Service Reply" message. The first two STS protocol messages ride piggyback as depicted in option 2 of figure 4 and then the third one is sent on its own to complete the key agreement. Until this point in the protocol the messages are in plaintext. Subsequent protocol messages are encrypted using the K_{RTM} agreed in the first three messages. This also serves to authenticate the source.

The main trust negotiation is done using the *Certificate Request* and *Certificate Request Reply* pair. To match the request and reply messages a Request Identifier (RID) is used. The certificates are requested by the attribute name/value pair. It is not necessary for the reply to contain all the certificates requested. This ensures a give and take of certificates according to the certificate release policies. A deadlock in trust negotiation is detected when a certificate request is repeated. In the event of deadlock in certificate exchange either of the nodes involved can use the *Service Policy Request* message to ask for the service policy of the other node. If the disclosure policy of the service policy is satisfied the node sends the service policy using the *Service Policy Reply* message. The end of the negotiation is signalled using the *Negotiation End* message that also contains the service(s) that are unlocked by the node sending it. Either of the two nodes involved in a trust negotiation can send it any stage after the key agreement is over. This is to ensure that no false *Negotiation End* messages are sent.

3.1.2 Service Policy and Certificate Release Policy

The RTM layer decides which certificate to release to the remote node depending on the service policy and the certificate release policies. These two policies along with user input determine the negotiating strategy of the node. If the policies are well written the manual intervention is minimized. The service policy contains a mapping between service and the certificates required. Figure 5 gives a sample service policy for the scenario mentioned in the introduction section.

Service	***Certificates Required***
Level 1 Access to Information to be released to Public	Certificate showing current police departments attributes
Level 2 Access to individual actions to be performed at the site.	Certificate showing current attributes allowing it to be on site
Level 3 Access to individual narcotic case resources	Certificate having the correct current narcotic departments attributes
Level 4(Secret) Access to under investigation cases	Certificate having correct current personal attributes
Disclose Service Policy if level 2 trust is achieved	

Figure 5. Sample Service Policy for Access to Narcotic Units Mobile Node

To start the trust negotiation process one has to start from level 1 and proceed upwards. However level 4 of the service policy is a sensitive level and is only disclosed only if to level 3 trust has been reached. The service policy can only be released to the remote node if level 2 of trust has been formed. A service policy is released on demand to a remote node to resolve a deadlock situation. The deadlock is detected when a node receives two or more *Certificate Request* messages with requests for the same certificate(s). By examining a service policy from a remote node the RTM can find some local certificates that can resolve the deadlock or prompt the user to manually release some certificates to break the impasse in the trust negotiation.

(Organization="LA Police" AND Certificate Issuer="LA Police") AND (Department="Narcotics" AND Certificate Issuer="LA Police") AND ((Personal Type="Detective" OR Personnel="Sergeant") AND Duty="Patrolling" AND Time="Night" AND Certificate Issuer="Personnel Department")

Figure 6. Certificate Release Policy for Narcotic Officers Identification Certificate

Each local certificate to be used in trust negotiation process has a certificate release policy attached to it. This policy takes the form of a simple set of rules specifying the set of assertions that must be met before the certificate is released. A typical "Certificate Release Policy" for network administration certificate for an online examination is shown if Figure 6. This certificate can be released to the node which exhibits that it belongs to "LA Police" department and "Narcotics" department. In addition the node has to prove that it is a detective or sergeant assigned to patrolling duties at night. The certificate issuers are also specified. The trust in the certificates in the RTM layer depends on the trust model used. These models enable our scheme to operate in absence of the certification authority for some time.

3.2 Neighbour Trust Management

This layer's primary objective is to protect the routing mechanism by forming a peer-to-peer encryption key *(K_{NTM})*. The key formed also protects neighbourhood communications. To discover the neighbours the first message of the STS protocol is broadcasted as a "Hello". A node will reply to the periodic "Hello" message if it does not already have a K_{NTM} with the broadcasting node. If the node receiving the broadcast already has a K_{NTM} agreed on then it just refreshes the information that the concerned key is still valid. There is a timer attached to each K_{NTM} formed. It is refreshed each time an encrypted data transfer is done between nodes. If this timer expires due to a long period of inactivity the key is discarded.

The two assumptions made in this layer are: firstly all the links in the network are bi-directional and secondly every node has a *network address* certificate to be used for authentication in the K_{NTM} formation. This network address certificate has embedded permissions in it allowing the node to use certain addresses. The trust in the network address certificate will be determined by using one of the three trust models described later in this section.

3.3 Trust Models and Manual Intervention

Our scheme in its normal configuration has the CRL for all the trusted certification authorities. A CRL contains the references to the certificate revoked

by the certificate authority. These are issued periodically so access to the trusted certificate authorities is not required at all times. But in an ad hoc network access to an online certification authority cannot be guaranteed at all times so some updates of CRL's may be missed. Also access to a new certification authority may be wanted when the user decides to trust it. During such times our scheme uses the following trust models described below.

The first model is the *Probabilistic Model*. Initially each of the certificates issued by a certificate issuer has the *trust value* 1 associated with it. A *distrust value* "p" (0<p<1) is subtracted from the *trust value* each time the periodic update of the CRL is missed. If an updated CRL is received the *trust value* associated is reset to 1 again. A negotiation is valid if the average of *trust values* of all certificates involved in the negotiation has to be greater than the *threshold trust value*. The NTM layers key (K_{NTM}) formation is not done until the *trust value* of the remote network address certificate is more than *threshold trust value*. For example: let a node use local certificate C_1, C_2 and C_3 in trust negotiation with *trust values* of 0.4, 0.6 and 0.55. It received certificates RC_1, RC_2 RC_3 and RC_4 with *trust values* 0.4, 0.7, 0.7 and 0.6 respectively. The average of trust values is 0.56. If the *threshold trust value* is set to 0.55 the above mentioned trust negotiation is successful.

The second model is called the *Weight Model,* which can be used in conjunction with the *Probabilistic model*. In this model each certificate participating in trust negotiation is assigned a "weight". This allows some certificates to be more valuable in a trust negotiation than the others. A *threshold weight value* set by user which has to be exceeded by the average of all weights used in the negotiation. The NTM layer is not affected by weight model and behaves in the same way, as it would do in the Conventional Model. An example of use *Probabilistic model* of along with the *Weight model*: let a node use certificate C_1, C_2 and C_3 in trust negotiation with *trust values* at that time 0.4, 0.6 and 0.55. It received certificates RC_1, RC_2 RC_3 and RC_4 with *trust values* 0.4, 0.7, 0.7 and 0.6 respectively. Let the *weight* attached to C_1, C_2 and C_3 be 2, 3 and 1 respectively. The received certificates RC_1, RC_2 RC_3 and RC_4 all have weights 1. Therefore using the formula, $\frac{\sum Pi*Wi}{\sum Wi}$, where P_i and W_i are the *trust value* and *weight* for certificate "i". This weighted mean is 0.555. If the *threshold weight value* is set to 0.55 the trust negotiation is successful.

In a typical scenario the node starts with *Conventional Model* and progresses to *Probabilistic Model* if some of the scheduled CRL updates are missed. The *Weight Model* can be used when the user thinks that some certificates are more important than others in a trust negotiation. Our scheme can switch between these configurations on its own depending on parameters set by the user. But if the user wants to control the switchover it can do it manually.

User intervention in our scheme is primarily provided for deadlock resolution in trust negotiation. A user can manually release a certificate or service policy to a remote node. This gives the user the capability to do the entire negotiation manually. The user can also edit the service policy and the individual certificate release policies.

4. Implementation

This section describes how our system is implemented along with some measurements of time taken to do the cryptographic operations required. The trust management scheme is implemented using the NTRG mobile test-bed with its generic stack structure [14]. In general the bottom layer(s) of the stack interacts with underlying hardware while the top layer(s) are the application which interacts with the user. This generalized stack allowed us to develop applications on Pocket PC's (Win CE) as well as fixed Workstations (Win32).

Figure 7 shows the protocol stack for a networked application employing our trust framework. The 802.11 layer is a radio interface to an IEEE 802.11b network card using broadcast IP, while the Ethernet layer is an interface to an Ethernet card. If the node is providing a bridge between the fixed network and the ad hoc network both the 802.11 and the Ethernet layers are present in the application. In case of mobile nodes only the 802.11 layer is present. The NTM layer forms the peer-to-peer keys to encrypt communication between neighbours. The next layer in the stack is the DSR layer which is an implementation of the Dynamic Source Routing (DSR) [13] Ad-Hoc Networking Protocol. On top of the DSR layer there is the RTM layer which is responsible for trust negotiation, end-to-end key formation and service availability. This layer decides if the packet coming from below can go up to which service depending on the trust negotiation. The two services provided currently are the person-to-person telephony application and the Instant Messaging like chat program in the audio and the chat layers respectively.

The STS Key agreement in NTM layer takes

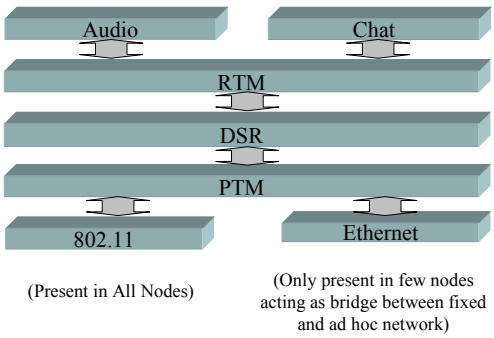

Figure 7. Protocol Stack with NTM Scheme

typically 60 ms on Compaq iPAQ H3630 Palmtop using 512-bit modulus certificates for 128-bit session key. The worst case scenario for the NTM layer is when all nodes in an ad hoc network are clustered together in range of each other. In a test of this worst case scenario with 6 mobile nodes with the above-mentioned key sizes it took 1014 ms for the key agreement to complete at the NTM layer. The NTM layer typically has small memory requirements. For each 128-bit key negotiated it has to store 24 bytes (4 bytes node name + 16 bytes key + 4 bytes Timestamp).

The trust negotiation process for one of the scenarios outlined in introduction section is shown in

Figure 8. In this instance the identity certificate used for the STS key agreement is 1024-bit. The node which starts the negotiation spends 250ms on computation for the key formation. On the other node providing the service similar process takes 255ms. Main trust negotiation using 1024-bit certificate takes 51 ms and 44 ms respectively. Overall the entire process ends in 305ms on the node asking for the service. The node providing the service takes 301 ms for the entire process. Each of the three nodes in the example takes an average of 2ms to process the message through the stack structure.

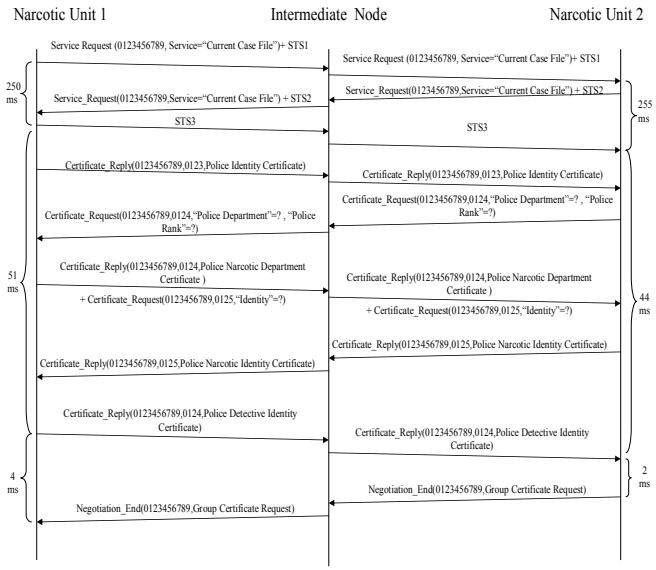

Figure 8.Trust Negotiation

In the above-mentioned example the key negotiated for encrypting the traffic is 256-bit. Therefore each negotiated key takes 40 bytes (4 bytes node name + 32 bytes key + 4 bytes Timestamp) of storage. The size of the SAT is n*32+m*128+m*n+320*n bits where m=#of services the node provides and n=#of nodes with which trust has been established. Each node and service name is 4 byte and 16 byte respectively. Let us take a simplified example when a node provides 4 services and has negotiated trust with 5 remotes nodes. Then size of SAT is 2292 bits or 287 bytes. Some of the cryptographic benchmarks obtained during implementation using RSA parameter e=0x10001 and using CryptoAPI are given in table 1.

Cryptographic Operation		Average Time
RSA Verification	512 Bit	1.339 ms for 1 iteration
	1024 Bit	5.509 ms for 1 iteration
	2048 Bit	15.256 ms for 1 iteration
DES Encryptions/Decryptions		4213 ms for 100000 iterations

Table 1. Cryptographic Benchmarks

5. Conclusion and Future Work

This paper presented a multi-faceted trust negotiation scheme for ad hoc networks by progressively exchanging property based certificates. The use of property based certificates makes the processes of trust negotiation similar to real world. Also there is a separation between the roles of a certificate in trust negotiation with respect to the original purpose of issue. Such flexibility lets a node operate in a new area of operation without reconfiguration. Additionally sensitive confidential certificates can be protected by disclosing them only if some partial trust already has been built. The trust negotiation process results in granting of access to services. This negotiation is done by taking inputs from the service policy and the certificate release policies. Facilities for deadlock resolution are provided at the protocol and the user level. Also the scheme is able to operate for some time in absence of an online certification authority. To protect the trust negotiation and the underlying routing process we have proposed a two-tier key formation. This key formation makes the trust management scheme resistant to eavesdropping, re-duplication and impersonation attacks.

A prototype of the system has been implemented and further work is continuing on development of the policy languages. Early benchmarks prove that the cryptographic operations are easily handled by the commodity handheld PC's. This work can be extended to provide checking of trust along the route in an ad hoc network. Furthermore it can be used to determine the membership criteria in group formation.

6. Bibliography

[1] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," IEEE Network Magazine, Vol. 13, No. 6, Nov 1999.

[2] Philip Zimmermann, "The Official PGP User's Guide", MIT Press, June 1995.

[3] Public-Key Infrastructure (X.509), http://www.ietf.org/html.charters/pkix-charter.html

[4] B. DeCleene et al, "Secure Group Communications for Wireless Networks", MILCOM 2001.

[5] Jiejun Kong et al, "Adaptive Security for Multi-layer Ad-hoc Networks", Special Issue of Wireless Communications and Mobile Computing, John Wiley InterScience Press, 2002.

[6] Jean-Pierre Hubaux, L. Buttyan and S. Capkun "The Quest for Security in Mobile Ad Hoc Networks" Proc. of the ACM MobiHoc 2001, Long Beach, CA, USA, Oct. 2001.

[7] Simple Public Key Infrastructure (SPKI), http://www.ietf.org/html.charters/spkicharter.html.

[8] M. Blaze, J. Feigenbaum, and A. D. Keromystis, "The KeyNote Trust Management for Public-Key Infrastructures" Cambridge 1998 Security Protocols International Workshop, England, 1998.

[9] W. Winsborough, K. Seamons and V. Jones, "Automated Trust Negotiation", DARPA Information Survivability Conference and Exposition, Hilton Head Island, SC, Jan. 2000.

[10] Boyeon Song and Kwangjo Kim, "Comparison of Existing Key Establishment Protocol", KIISC (CISC 2000), Vol. 10, No. 1, pp. 677-690, Nov. 2000.

[11] S. Blake-Wilson and A. Menezes, "Authenticated Diffie-Hellman Key Agreement Protocols", Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), LNCS 1556, pp.339-361, 1999.

[12] W. Diffie, P.C. van Oorschot, and M.J. Wiener, "Authentication and authenticated key exchanges," Designs, Codes and Cryptography 2 (1992).

[13] J. Broch, D. Johnson, and D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks," IETF Internet Draft (work in progress), ttp://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-08.txt.

[14] O'Mahony, D. & Doyle, L., "Architectural Imperatives for 4th Generation IP based Mobile Networks", Fourth international symposium on wireless personal multimedia communications, Sep. 2001, Aalborg, Denmark.