REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
27-01-2003	Final Report	1 August 2001 - 01-Oct-02

4. TITLE AND SUBTITLE

Secure Group Management in a Coalition Environment

5a. CONTRACT NUMBER
F61775-01-WE052

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dr. Donal O'Mahony

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Ireland Trading as Trinity College Dublin
Dublin 2
Ireland

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
SPC 01-4052

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking University of Ireland Trading as Trinity College Dublin as follows: The contractor will investigate issues related to the formation and security of ad-hoc wireless networks in situations where teams are operating separated from their base location and possibly roving. The most capable group management systems available (for example, multicast protocol like VersaKey) employ a fixed group controller responsible for matching join-requests against the allowed-members list. Failure of this group manager disrupts communications and renders ineffective the entire group. Additionally, such ad-hoc schemes rely on forward confidentiality of departing nodes and backward confidentiality of newly joined nodes. Thus appropriate measures must be established to authenticate users and control access to resources and information. This effort will address these information assurance shortcomings.

15. SUBJECT TERMS

EOARD, Coalition Operations, Ad-hoc Networks, Security, Wireless Networks

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Christopher E. Reuter, Ph. D.
UNCLAS	UNCLAS	UNCLAS	UL	17	19b. TELEPHONE NUMBER *(Include area code)* +44 20 7514 4474

Contract No: F61775-01

SPC 01-4052 : Secure Group Management in a Coalition Environment

Item 0003 Final Report

Contractor: Networks & Telecommunications Research Group,

University of Dublin, Trinity College, College Green, Dublin 2, Ireland

Contact: Dr. Donal O'Mahony, Group Director

Introduction

This project set out to solve the security problems associated with communication between teams of people, affiliated to different military coalition partners, across a wireless ad-hoc network. The goals of the project were ambitious and involved carrying out background research, design of a group scheme and implementation of a prototype over a 14 month period.

Soon after the project started, we realized that before any individual can be admitted to a group, they must authenticate themselves to an entity with the power to admit them. Since we were interested in allowing individuals to maintain a multi-faceted identity, and to use this to become a part of many different groups simultaneously, we began by tackling the end-to-end authentication issue.

We have made substantial progress on this and an early prototype was demonstrated to representatives of EOARD in June 2002. We also have a comprehensive design for a system to leverage the authentication scheme to gain entry to a group which can adapt its mode of operation to suit the scale of the ad-hoc network.

Since our system depends on the use of certificates, some of which may be issued in the field, we have devised a secret sharing system that allows signature to be applied by any N of M nodes for the purposes of this certificate generation.
We have built prototypes of the authentication and secret sharing systems and in the near future we hope to complete the prototyping, integrate the components into a single system and deploy it in our campus-wide experimental wireless ad-hoc network

Progressive End-to-End Authentication

In general, nodes in an ad-hoc network will mutually authenticate each other to achieve some particular purpose. This purpose will depend on the production of a specified set of certificates and on nodes providing proof of possession of the corresponding private key. Nodes will only wish to keep the number of certificates exchanged to the minimum required to achieve the purpose

In the trust negotiation a certificate is released when its certificate release policy is satisfied. This policy usually depends on the previous certificate exchanged but certain designated certificates may be always unlocked to ensure a starting point for negotiations. The mapping between the services and nodes that they are allowed to is maintained in the service policy. This policy is released in case of a deadlock in trust negotiation between the nodes. Alternatively, a user can intervene in a deadlocked negotiation by manually releasing a certificate.

To ensure that the there is additional basis for forming trust the certificates can be cross-linked (i.e. some attribute in the certificate points to an attribute of another certificate). This is depicted in figure 1.

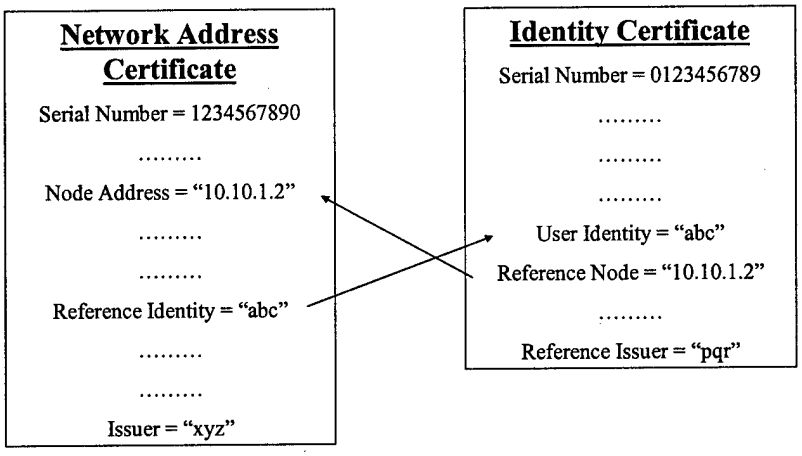


Figure 1.Cross-Referenced Certificates

Authentication in Wireless Ad-hoc Networks

In an ad-hoc network, all nodes act as routers and when establishing an end-to-end dialogue, one must be concerned about the trust level of nodes along the path. For this reason, the authentication scheme was split into two layers, with one oriented towards securing peer-to-peer communications and the other securing end-to-end communications and negotiating the trust level. These two layers are named are Peer-to-Peer NTM (PPNTM) and Remote NTM (RNTM) respectively.

The PPNTM layer encrypts messages between neighbours (i.e. all nodes within broadcast range who can reply). First the PPNTM layer has to discover its neighbours. This is done by broadcast of periodic "Hello" messages. The Traffic Encryption Key (TEK) formed in this layer has a short lifetime and key length due to the fact that nodes may be highly mobile. The RNTM layer's TEK has a larger lifetime and key length allowing it to be valid for a longer time without compromise. Additionally the

key negotiation process adds to the communication overheads so frequent TEK changes are not desirable. Note that the RNTM layer's TEK is between users of the nodes, while the PPNTM layer's TEK is between nodes. This separation of node identity from user identity makes the user migration between nodes and networks easy.

Figure 3 illustrates how the key formation will occur in a representative network. Let the network consist of five nodes A, B, C, D and R. The peer-to-peer keys K1 to K4 are formed between the respective nodes. If an external attacker "X" tries to listen to the traffic in the network it has to break the keys K1 and K2 (these are the peer-to-peer keys in "X" listening range). Even breaking these keys in real-time will yield attacker limited local information.

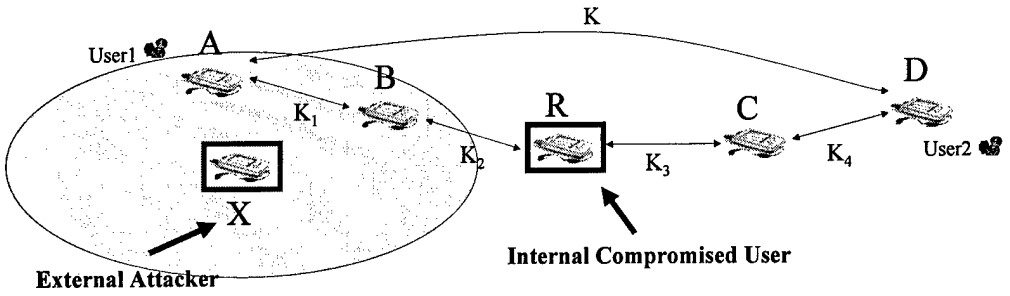


Figure 2 Key Agreement in NTM

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.

Both the layers use the Station-to-Station (STS) protocol [Dow'92] for key formation. 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 "Negotiation Wanted" message. It contains a random number UID (unique identifier) to identify the trust negotiation. This UID will be used in all subsequent messages in the trust negotiation exchange. The "Negotiation Wanted" message also contains the service(s) the node initiating the process wants to access on the remote node. The remote node replies with a "Negotiation Wanted Reply" message with the services that requires negotiation along with the service(s) which are already unlocked. These two messages carry the STS protocol elements. The third STS message is sent to complete the end-to-end key formation. All this message exchange is depicted in option 1 of Figure 4.

If a node tries to access a service which is not unlocked for it on a remote node without any prior trust negotiation then a message exchange depicted in option 2 occurs. The remote node sends a "Negotiation Required" message to the node requesting the service. This message contains the service(s) for which the negotiation is required along with a UID. This is replied to by the "Negotiation Required Reply" message for the initial node. The first two STS protocol messages ride piggyback as

depicted and then the third one is sent to complete the end-to-end key formation. Up to this point in the protocol the messages are in plaintext. The remaining messages are encrypted using the newly established TEK.

The main trust negotiation is done using the "Certificate Required" and "Certificate Required Reply" pair linked with a common Request Identifier (RID). The certificates are requested by the attribute name/value pair. It is not necessary for the reply to contain all the certificates requested. Usually each "Certificate Request" has a "Certificate Request Reply" piggyback on it. 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 case 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.

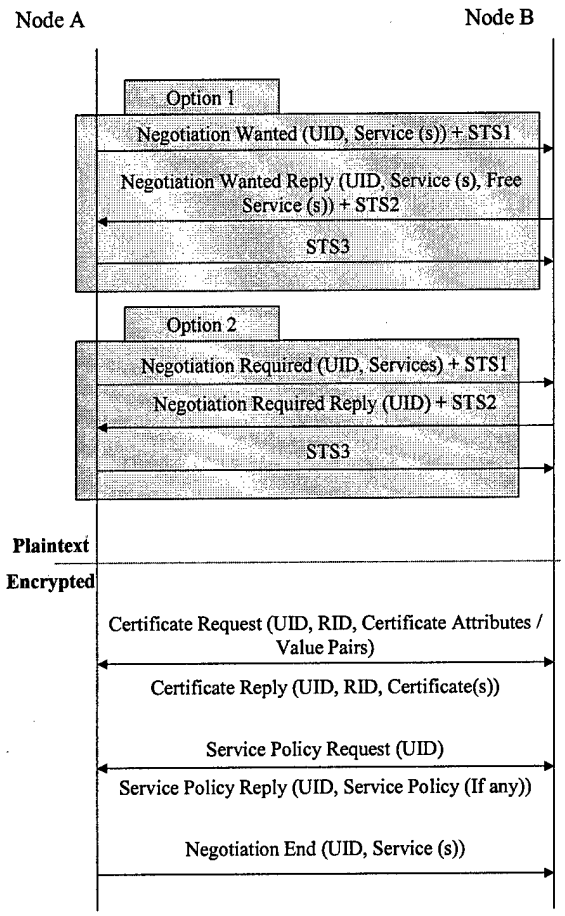


Figure 3 RNTM Layer's Trust Negotiation Protocol

Figure 4 shows the protocol stack for a networked application employing the NTM trust framework. The PPNTM layer site directly above the radio link and forms the peer-to-peer keys to encrypt communication between neighbours. The packet then passes on to the *Encryption* layer above. This layer encrypts/decrypts the data in the packets depending on the keys agreed to in PPNTM layer. In this example, the ad-hoc routing protocol used is Dynamic Source Routing (DSR) [Bjm'99] On top of the DSR layer there is the RNTM 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 proceed to the various services depending on the trust negotiation. The two sample applications are a person-to-person telephony application and an Instant Messaging like chat program.

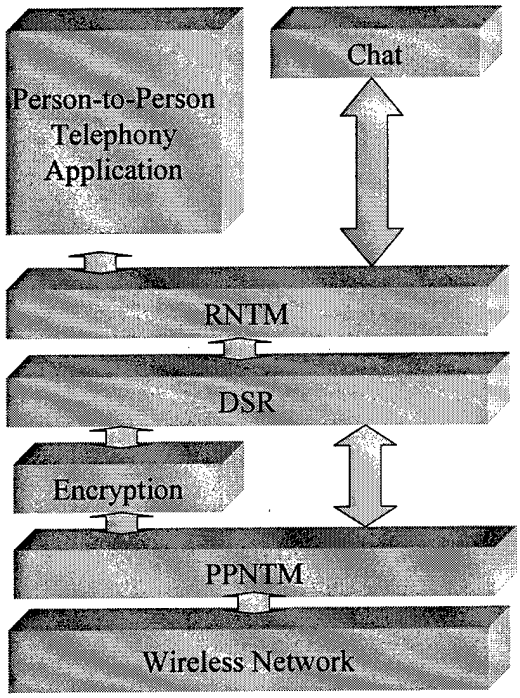


Figure 4 The Layer Structure of an NTM Application

The internal organization of the RNTM layer is shown in Figure 5. The Certificate Exchange Agent (CEA) is the heart of this layer. It receives the message from the layers above or below and acts on them according to the programmed negotiating strategy. Other data structures in the layer provide the CEA with information to aid in trust negotiation. 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). The Certificate Release Policy (CRP) contains the release policies for each of the certificate in the LCS, which may be used for trust negotiation. An association between the certificates required and the services is maintained in the Service Policy (SP). 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 identity to which it is allowed is kept in the Service Access Table (SAT). It also contains the respective end-to-end encryption keys.

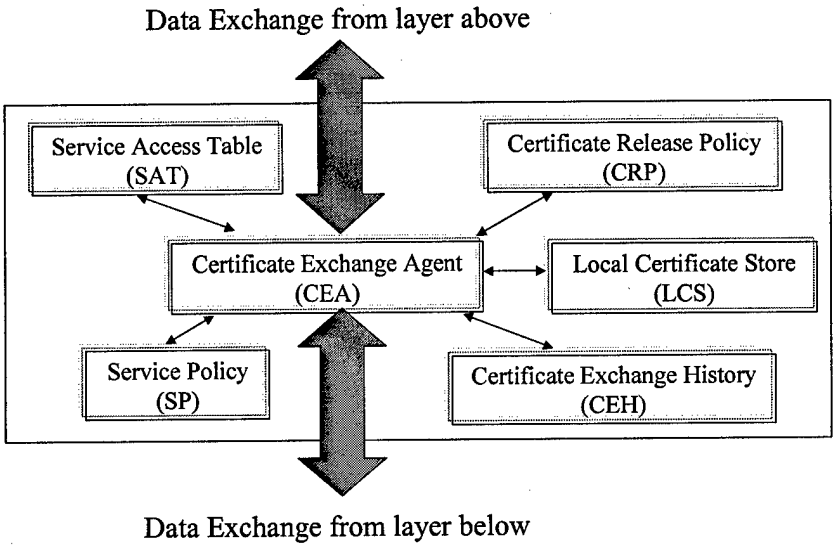


Figure 5 Internal Organization for the RNTM Layer

The RNTM layer decides which certificate to release to the remote node depending on the certificate release policy. 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. For example a certificate indicating the job title of an individual may not be released without the requestor proving (by producing the appropriate certificate) that they were affiliated to the same organization.

Forming a Group

Now that a node has an ability to progressively authenticate itself to others, we would like to use this capability to gain entry to a group in which a node can simultaneously communicate with all other group members.

We are at the early stages of research in formulating a group management scheme, Figure 1 a, b and c show the stages of group formation using a protocol which is based loosely on the Group Secure Association Key Management Protocol (GSAKMP) [Hch'00]. The membership criterion (MC) is the expression of attribute name/value pairs that a node has to satisfy to join a group. It is similar to the certificate release policy of the NTM scheme presented before. The scenario 2 represents a node wanting to join a preformed group on receipt of an advertisement

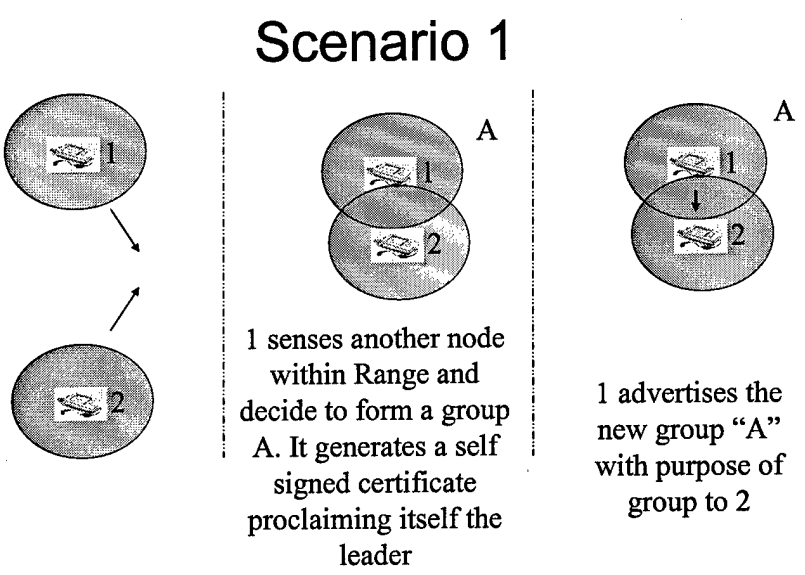


Figure 6.a Initial Stage of Group Formation

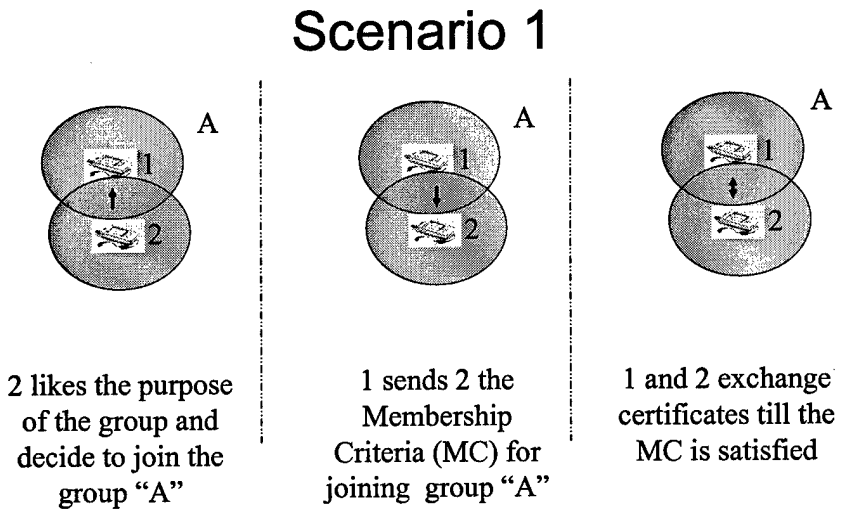


Figure 6.b Node 2 authenticates itself to Node 2 – The group owner

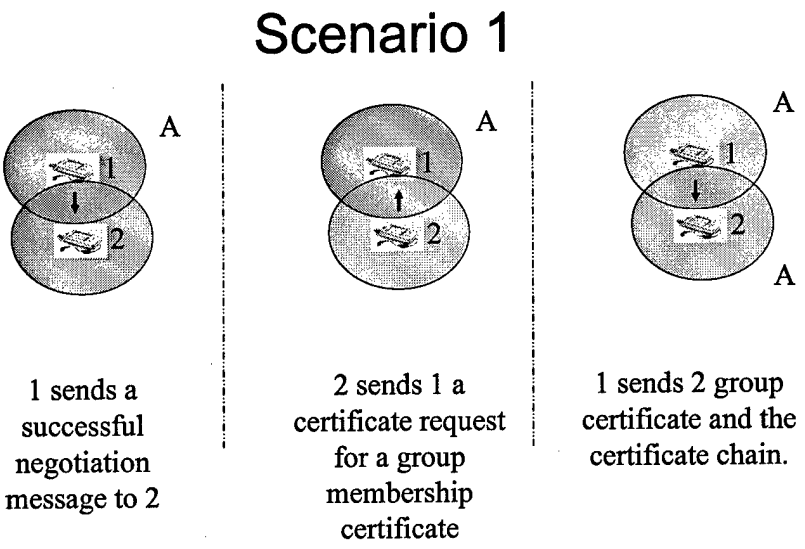
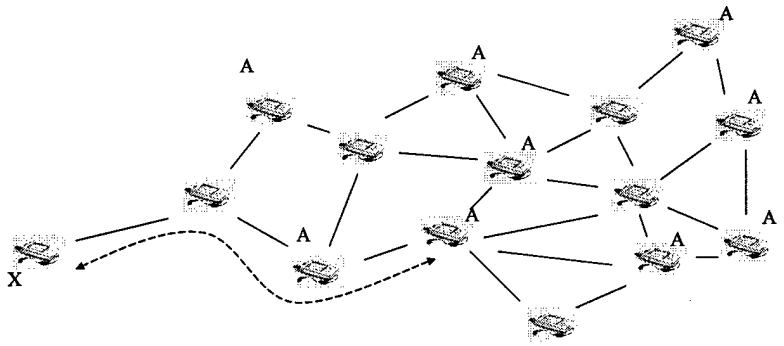


Figure 6.c Node 2 Admits Node 1 to the group



Node X gets a advertisement for Group A and then decide to join it. Note : A node not participating in group relays the group messages

Figure 7 Joining a pre-existing group

To cope with frequent membership changes three models are proposed in Figure 8. If the group is to be small and have a leader controlled existence then the first model is used. In case the group grows and it is difficult for the leader to control the group membership then the group shifts over to model two. The second model is less secure

than the first one, but at the same time is more scalable. The final model is the least secure but is most scalable. The scheme can proceed from model one to three with little reorganization but cannot easily move in the opposite direction. The exact protocol used to form a group and also the ways in which existing groups may change mode of operation is still the subject of active research.

Three Models

Central Control of Membership with group key (CCMGK) – The Group Leader decides the membership and the group key

Central Control of Memip without group key (CCMNGK) – The Group Leader decides the membership but there is no group key and scheme relies on sub-group keys

No Central Control Membership without group key (NCCNGK) – The Group Leader / Sub-Leader has control of its subgroup membership and there is no group key and the scheme relies on sub-group keys.

Secret Splitting

The above examples assume the existence of a certification authority to bind identities or attributes to individuals or nodes. The use of a central server to provide this Certification Authority service is not adequate for an ad hoc network. A central server may be unavailable and also becomes a single point of failure in the event of attack. Replication would alleviate problems of availability but greatly worsen the threat of attack.

A more sophisticated solution is the use a threshold service, whereby a threshold of servers, provide the service of a Certificate Authority as discussed by Zhou and Hass[ZH'99]. A threshold Certificate Authority is made up of a collection of servers which each possess a portion of the Certificate Authority's overall secret key. These shares are created using threshold cryptographic techniques such that the shares can be applied individually to create partial certificates. These partial certificates are then combined to create a whole certificate as described by Desmedt [Des'87],[DF'91]. The ability to apply the shares individually means that the overall secret need never be reconstructed. As such an attacker wishing to gain control of the service to issue false certificates would be required to take over a threshold of servers rather than a single server. The size of the threshold can be chosen in accordance with the risk of attack. Techniques such as share refreshing[HJKY'95] can be employed to force the attacker to complete his attack within a certain time frame, thus offering further protection against compromise.

The Threshold Certificate Authority Service can be created with the aid of a Trusted Share Dealer. This Trusted Share Dealer generates the public/private key pair and then breaks the private key into shares, which, are distributed to the members of the Certificate Authority Service. The reliance on a Trusted Share Dealer is restrictive. Such an entity may be un-contactable or may not exist. Ideally we would like the member nodes of the Threshold Certificate Authority Service to generate the key shares themselves in a collaborative manner. We would like this collaborative shared key generation protocol to be executed in such a manner that each node involved learns nothing about the overall secret key or about any share obtained by any other node involved. Boneh and Franklin [BF'97] introduced an efficient protocol for this type of shared key generation for the RSA algorithm. We have implemented this algorithm in conjunction with another algorithm by Catalano et al[CGH'00] . Catalano et al's algorithm has previously not been implemented. Our motivation for using Catalano et al's algorithm is discussed in the implementation section below.

The shared key generation algorithm allows us to create a Threshold Certificate Authority in a cooperative manner. No trusted shared dealer is needed.
We feel that the use of shared public/private key generation bridges the gap between small-scale key management techniques for ad hoc networking (such as physical / face-to-face key exchange) described by Stajano and Anderson[StajA'99] and large-scale key management schemes, namely Public Key Infrastructures, which rely on a Threshold Certificate Authority described by Zhou[ZH'99]. The shared key generation capability allows us to dynamically create a Certificate Authority in an ad hoc network leveraging whatever security associations exist in the network. These security associations may be imported from external sources(e.g. the fixed network) in the form of shared keys or pre-issued certificates. However they may also be transient security associations created from within the network as described by Stajano[StajA'99]. He suggests that parties who come to trust each other for whatever reason will physically exchange keying material so that they can maintain this trust over the communications medium.

We feel that small communities could grow in this manner exchanging pair-wise keys but that for large communities of nodes this would quickly become unmanageable. However by using these secure links and running the shared key generation algorithm a threshold Certificate Authority could be created as we have described. Thus the formation of a large scale key management solution in the network i.e. a Public Key Infrastructure can be ad hoc.

Implementation Details

We implemented a combination of algorithms to achieve our shared RSA key generation functionality. Boneh and Franklin's techniques produce an additively shared RSA secret key and corresponding public key. We wanted to produce a polynomial sharing of the secret key. Our motivation for this was to facilitate the certification phase. Creating certificates using polynomial key shares does not require interaction on behalf of the client wishing to be certified. The client obtains the partial certificates from the different servers and when he obtains a threshold of partial certificates he combines them to create the full certificate. On the other hand , if additive key shares are used the client must know in advance which servers he is going to obtain his certificates from, he informs each server which coalition of servers he is going to use and each server produces a partial certificate tailored for that

coalition. This is undesirable for two reasons. Firstly it requires the servers to carry multiple partial shares (1 for each possible coalition of servers). Secondly and most importantly for our ad hoc networking environment, it is undesirable because the client wishing to obtain certification must know in advance which servers it is going to use. If one of the servers becomes unavailable all the partial certificates he has collected are wasted. The client must begin anew and request partial certificates from all servers in a new coalition.

We used the techniques of Catalano et al[CGH'00] to create the polynomial sharing of the private key. We implemented our solution for the specific case of a (2,3) threshold scheme. 2 is the threshold , 3 is the number of participants. We have tested shared RSA key generation between the 3 servers running on 500MHz Pentium laptops communicating over IEEE 802.11b. On average it takes under 2 minutes to produce a shared 1024 bit key on these machines. We are currently porting our code to Windows CE to test the viability of providing such a service on a more constrained platform and deploy it in our production network.

Secret Splitting – Future Plans

We implemented our scheme for the specific case of (2,3) threshold Certificate Authority. To expand our scheme to a (t,n) scheme would require a few changes. In particular the combination of the partial certificates would require a modification as described by Shoup[S'00].
Other augmentations could be made to make the key generation phase robust, i.e. tolerate misbehaving members of the key generation protocol. Techniques for robust shared key generation are described in Frankel et al [FMY'99].
However we are primarily concerned with the investigation of new applications for shared cryptographic key generation and threshold cryptography in general as discussed below.

New Applications for Secret Splitting

Task oriented Groups/Coalitions

The coalition of 2 groups or armies may be facilitated by threshold cryptography. The rights or powers of the coalition may be distributed using shared private keys. Decision making in the group may be based on a consensus of a threshold of members. The formation of this group could be based on the formation of a public/private key pair, which was shared in a threshold manner amongst the group members. Our shared generation algorithm could be used to achieve this. Group decisions would be based on a threshold of signatures being applied by the group.

Multicast Group Keying

Distributing symmetric traffic keys throughout a multicast group relies on the group leader sharing a long term secret key with each member of the group or on a Public Key Infrastructure being in place. This relies on central management of group membership, namely through the group leader. One area we are interested in investigating is an alternative method of administrating and distributing group keying information in a distributed fashion rather than a centralised fashion. The members of a group could collaborate to generate a shared private/public key pair. New symmetric traffic keys would be encrypted using this public key of the group. To decrypt the message to find the traffic key a threshold of group members would need to collaborate. This would remove the need for a leader who manages membership of the group. The membership management would be distributed throughout the group. This may provide a more timely mechanism for expelling and admitting members to the group and would also have other obvious benefits. The new keys would be encrypted and broad cast once, thus, individual messages to every member of the group would not be needed. Also the reliance on a central leader figure would be removed which is attractive in ad hoc networking scenario.

Ubiquitous computing applications.

The vision of ubiquitous computing is that computers will be pervasive and act often on behalf of a user without user interaction. Machines will interact almost socially perhaps akin to human interaction. New trust models will develop as discussed by Shanker and Arbaugh[SA'02]. Communities of cooperating computers may develop just as how communities develop in human society. These trusted communities will require keying material so that there interactions can remain trusted and secure. Our shared key generation algorithm will allow large communities of computers to maintain their trust in one another. Clearly in this scenario, physical key exchange will become unmanageable and large PKI's will be needed.

Deploying the System on DAWN

In order to demonstrate our work, the Networks & Telecommunications research group is building a test network called the Dublin Ad-Hoc Wireless Network (DAWN). As an ad-hoc network, there is no requirement for fixed infrastructure, but we have deployed a small number of relay nodes across the University campus which have permanent connections to power and in some cases have the capability to relay packets into the fixed network.

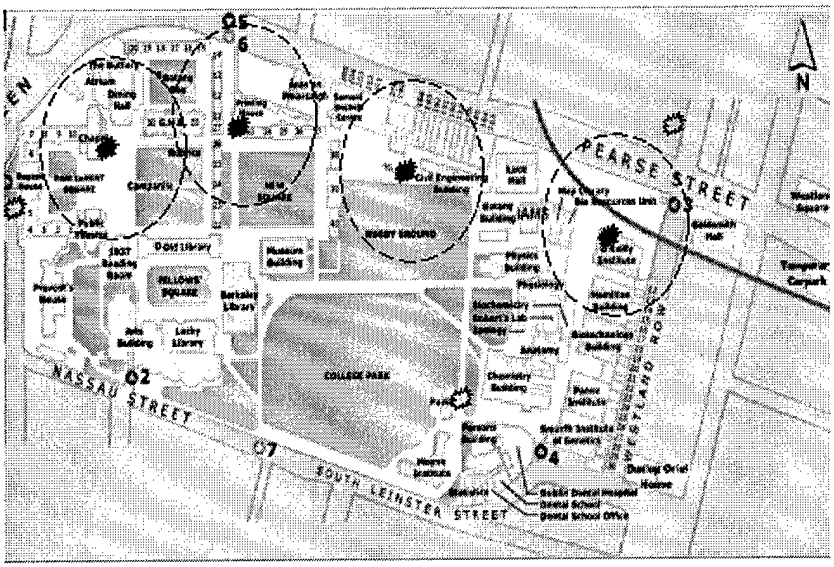


Figure 9 Phase 1 of the Dublin Ad-hoc Wireless Network (DAWN)

Presently, we have integrated the simple PPNTM into the production protocol stack and have the ability to carry voice and data traffic over an ad-hoc network with AES encryption applied to the individual links. We expect that in the coming 2 months, we will have a working implementation of PPNTM which will allow quite sophisticated end-to-end authentication between ad-hoc nodes in the network.

All of the work so far has been carried out with certificates issued to the nodes by an off-line authority prior to deployment. We would like to extend the secret splitting work to develop an operational distributed CA which could generate its public/private key pair in the field and then issue special purpose certificates to nodes for use in PPNTM-based authentication. We would like to explore applications where small teams could create a distributed CA by communicating with each other using the IrDA ports of their handhelds and then use the CA thus formed to later certify new members into a group based on a threshold of approving nodes.

We are also planning to further refine our group maintenance scheme and to deploy this in the DAWN network.

Conclusions and Future Work

The original goals of this project were ambitious and the timescale quite compressed. Within the elapsed time, we have developed an advanced progressive authentication scheme that can be used for person-to-person authentication in a military coalition environment. We have also designed and partially implemented a threshold certificate authority scheme which can be used to form CAs dynamically and thus can dynamically form teams in the field. In terms of group communication, we have embarked upon the design of a scheme that can scale from very small groups up to very large numbers of nodes.

With additional time and resources, we would like to integrate these schemes and develop a prototype system that could be tested in our DAWN network allowing a number of different usage scenarios to be exercised.

References

[Dow'92] W. Diffie, P.C. van Oorschot, and M.J. Wiener, "Authentication and authenticated key exchanges," Designs, Codes and Cryptography 2 (1992)

[Bjm'99] J. Broch, D. Johnson, and D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks,"http://www.ietf.org/internetdrafts/draft-ietfmanet-dsr-03.txt, Oct 1999. IETF Internet Draft (work in progress).

[Hch'00] H. Harney, A. Colegrove, E. Harder, U. Meth, R. Fleischer, Group Secure Association Key Management Protocol (GSAKMP), draft-irtf-smug-gsakmp-00.txt, November 2000, Work in Progress.

[BF'97] D.Boneh,M.Franklin. "*Efficient generation of shared RSA keys.*" Crypo'97, p425-439 LNCS 1294

[CGH'00] D.Catalano,R.Gennaro,S.Halevi "*Computing inverses over a shared secret modulus*". Eurocrypt'00, LNCS 1807, Pages 190-206

[Des'87] Y. Desmedt. "*Society and group oriented cryptography:A new concept*". Crypto'87 , p120-127 LNCS 293

[DF'91] Y.Desmedt,Y. Frankel "*Shared Generation of Authenticators and Signatures*" Crytpo'91 p457-469 LNCS 576

[FD'92] Y.Frankel,Y.Desmedt. "*Parallel reliable threshold multisignature",* Tech. Report: 92-04-02, University of Wisconsin-Milwaukee.

[FMY'98] Y. Frankel, P.MacKenzie,M.Yung "*Robust Efficient Distributed RSA-key Generation*" The Thirtieth Annual ACM Symposium on Theory of Computing STOC '98

[HJKY'95] A.Herzberg,S.Jarecki,H.Krawczyk,M.Yung. "Proactive Secret Sharing or: How to cope with perpetual leakage", Crypto'95 p339-352 LNCS 963

[SA'02] N.Shankar, W.Arbaugh "On Trust for Ubiquitous Computing", Workshop on Ubiquitous Computing , Sept.'02, Göteborg Sweden.

[S'00] V. Shoup "Practical Threshold Signatures", Eurocrypt'00 p207-220 LNCS 1087

[StajA'99] F.Stajano, R. Anderson "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks". 7th International Workshop on Security Protocols, Cambridge, UK, April'99.

[ZH'99] L.Zhou, Z.J.Haas "Securing Ad Hoc Networks" IEEE Networks,13(6):24-30,1999.

Declarations:

The Contractor, Trinity College, hereby declares that, to the best of its knowledge and belief, the technical data delivered herewith under Contract F61775-01-WE052 is complete, accurate, and complies with all requirements of the contract

Date:_____
Name & Title of Authorized Official:_____

I certify that there were no subject inventions to declare as defined in FAR 52.227-13, during the performance of this contract.
Date:_____
Name & Title of Authorized Official_____