Shared RSA Key Generation In A Mobile Ad Hoc Network*

B.Lehane and L.Doyle,
Dept. of Electrical & Electronic Engineering,
Trinity College Dublin, Ireland
Email: {lehaneb,ledoyle}@tcd.ie
and

D.O'Mahony,
Dept. of Computer Science
Trinity College Dublin, Ireland
Email:omahony@cs.tcd.ie

ABSTRACT

The use of a threshold certificate authority to provide cryptographic key management in mobile ad hoc networks has been suggested in the literature. We have designed and implemented such a key management service for our ad hoc networking test bed. In this paper we describe our use of distributed shared RSA key generation techniques to create a threshold certificate authority 'from scratch'. Our goal is to create a scaleable key management solution, which does not rely on prior infrastructure for its inception, and as such is formed in a truly ad hoc manner, compatible with the formation of the network itself.

Keywords:
Key management; Shared RSA key generation; Threshold Certificate Authority; Mobile Ad hoc network.

1 INTRODUCTION

Effective key management is the goal of any good cryptographic design [MOV97]. Providing a key management service in any network is challenging, but in an ad hoc network it is particularly difficult. Centralised servers cannot be relied on in an ad hoc network. Nodes often have no prior shared context as relationships and encounters are far more transient than on the fixed network. There are three distinct approaches to key management for ad hoc networks pursued in the literature: Key Exchange, Key Agreement and Public Key Infrastructure. We firstly will review the work on each of these approaches. We then discuss our work on the later area of Public Key Infrastructure and how we achieve a comprehensive key management scheme, which is tailored for ad hoc networks.

* 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

1.1 Key Exchange

Key exchange is the most primitive form of key management. Alice and Bob wishing to communicate over an insecure channel exchange a-priori a cryptographic key. This key can be exchanged by physical contact as suggested by Stajano [SA99] or over a secure side channel. Balfanz[BSSW02] expands this idea and suggests the use of public key exchange. Thus the side channel need only be secure against a 'man in the middle' attack [MIMwikipedia] but can tolerate eavesdropping, as the information exchanged is public. A 'man in the middle' (MIM) can be detected/avoided by using a short range Infrared or radio channel.

These principles are quite old. The use of physical key exchange must be the earliest form of key management, if it can be described as key management at all. Usually, key exchange is the most inconvenient method of creating a secure association between two communicating entities, however, in some ad hoc networking scenarios it is NOT inconvenient. Rather, this sort of 'demonstrative identification' [BSSW02] is useful, necessary and natural. Stajano gives the example of a laptop and thermometer wishing to communicate securely. The easiest way to be sure that 'this thermometer here' is talking to the laptop is to exchange a key by physical contact of the two devices. Because these two parties share no prior trust relationship or identity information, any identification other than 'this thermometer here' doesn't make sense. Thus for small personal area networks or similar scenarios, physical key exchange is logical and convenient.

1.2 Key Agreement and Group Keying

Key agreement protocols such as Diffie-Helman key agreement [DH77] are usually employed for more distant key exchange. Over greater distances the convenience of a secure side channel does not exist, and alternative methods must be used to thwart the 'man in the middle'

attack. If all parties share a common password, authenticated Diffie Helmann key exchange protocols [EKE], [SPEKE] can thwart an MIM attack. These protocols apply as readily to ad hoc networks as to fixed networks. Asokan [AG00] focuses on a method for authenticated group key agreement for use in an ad hoc network. This is a multiparty version of the EKE [EKE] authenticated key agreement protocol based on Becker's work [BW98]. Asokan envisions a small number of nodes in a conference-type scenario. For Example, a password could be written on a blackboard for all within the room to see, but which attackers outside could not see. Using this password to authenticate his group key agreement protocol a common group key could be created.

Group keying allows multiparty secure communications, and hence provides group level authentication and security. However, providing keying information for individual members of the group (i.e. to allow Alice and Bob to communicate privately in the presence of other group members) requires other key agreements. Indeed networks may form where group affiliation doesn't exist, particularly in a large-scale civilian network. As such, a group key agreement is of limited utility in a non group-oriented network, such as a civilian network in which many nodes choose to communicate but some require end-to-end privacy. A public key infrastructure is better suited to this scenario.

1.3 Public Key Infrastructure

Public Key Infrastructure (PKI) is the most scaleable form of key management. Several different PKI techniques exist: [SPKI] ,[PGP], [X.509]. Various forms of these PKI techniques have been proposed for use in ad hoc networks.

Aura [AM01] proposes the use of a group oriented Public Key Infrastructure for large group formation. The leader of the group acts as a Certificate Authority (CA), which issues group membership certificates. These are SPKI-style certificates. They certify that the public key in the certificate belongs to a group member. However this again is not useful for two-party communications or non group-oriented tasks (see above).

Hubaux [HBC01] proposes a PGP type PKI. In PGP any node can issue a certificate and as such is allows a completely distributed architecture, apart from the central repository, which holds these certificates. He proposes a scheme to avoid the need for a central repository of certificates in the PGP system. This scheme involves each node keeping mini-repositories, which hold all the certificates the node issues and all the certificates issued on it. When nodes A and B meet they merge their mini-repositories. The repositories are constructed according to the 'Shortcut Hunter algorithm', devised in the paper [HBC01]. This algorithm constructs repositories such that two nodes merging repositories have a high probability of finding a chain of certificates between them if one exists. This scheme is useful in a civilian environment where delegation of trust (the notation A→B implies A trusts B) through a number of nodes is acceptable, i.e. A→B, B→C, C→D, D→E therefore A chooses to trust E, A→E.

An alternative approach is to use a Certificate Authority (CA) to issue certificates. A CA is a third party trusted by all in the system, which effectively eliminates the need for a repository of certificates. Rather than finding a certificate linking A→B→C→D→E, one simply recovers the certificate A→E. As such the CA can be seen as a one-hop shortcut through the web of trust. The problem with this is the CA must be trusted by all and becomes a single point of failure in the event of an attack.

Zhou [ZH99] suggests the use of threshold cryptography [DF87] to create a distributed Threshold Certificate Authority. A *(t,n)* Threshold Certificate Authority (TCA) is a Certificate Authority service which is provided by a threshold '*t*' of nodes from a possible set '*n*' by employing threshold cryptography [Des87], [DF89]. This TCA could act similarly to a X.509 [X.509] CA on a fixed network, issuing certificates binding public keys to Identities. By using threshold cryptography, the private key used to create signatures/certificates is shared among the '*n*' participating nodes. A valid signature/certificate created by the TCA requires the co-operation of a threshold '*t*' of participating nodes. This is done in such a way that no participant in this operation learns any more about the private key than the share it already holds. Any number $k \leq t-1$ nodes cannot produce a valid certificate. As such an attacker would need to compromise *t* nodes rather than just a single node to compromise the CA service. Thus the single point of failure is removed. To create the TCA a trusted dealer generates the keying material (i.e. public/private key pair) and shares the private key amongst the *n* players in a *(t,n)* threshold fashion.

In this paper we discuss our implementation of a Threshold Certificate Authority. Our design employs shared RSA key generation to remove the need for a trusted dealer. As such, our TCA can be created in the field without reliance on a prior security infrastructure. We seek to close the gap between the small-scale key management afforded by two-party key exchange techniques [SA99], [BSSW02], and large-scale key management afforded by a TCA-based [ZH99] PKI.

The rest of this paper is organised as follows. Section 2 discusses the need for security in ad hoc networks. Section 3 discusses PKI based key management in ad hoc networks. Section 4 discusses our PKI design. Section 5 details our conclusions and suggests areas for future work.

2 SECURITY IN AD HOC NETWORKS

In this section we briefly motivate the need for cryptographic security services in an ad hoc network.

Ad hoc networks demand increased security over and above traditional fixed and mobile networks. Firstly, communication is wireless. Wireless networks are more readily prone to eavesdropping than a wired network, as there is no physical protection of the medium. Moreover, every node in an ad hoc network has increased responsibility in comparison to a node in a traditional fixed or mobile network. This is because every node in an ad hoc network is a router. With increased responsibility typically comes the need for increased security. For example, if a node is participating in routing functions, then we may want routing updates to be authenticated so that any misbehaving/malfunctioning nodes can be identified. A lot of work on secure routing protocols exists [ZA02], [YNK01], [PH02], [AHNR02]. (Almost all assume the existence of a PKI or some form of large-scale key management system, which PKI would facilitate).

Due to the possible deployment scenarios of ad hoc networks, for example a military battlefield, security is often more important than in conventional networks. Moreover, even in civilian networks where data might not need strict confidentiality, people usually desire a degree of privacy. For example, we may be happy to let the GSM network operator hear our mobile phone calls but in contrast, we may not be happy that our neighbour hears our calls. We no longer have a neutral third party GSM operator forwarding our phone calls, instead every node in the network is forwarding our calls and hence can eavesdrop on them. As such, end-to-end security may often be required. In fact we rely on the fact that our data can be overheard by nearby nodes so that it can be transmitted through the network. Clearly with more nodes having greater access to the data sent in the network the need to secure that data by cryptographic means increases. This requires effective cryptographic key management.

3 PKI-BASED KEY MANAGEMENT

As discussed in Section 1, the literature covers several attempts to solve the problem of key management in ad hoc networks. Various different network scenarios are contemplated, from small-scale personal area networks [SA99] to large-scale environments [KZLLZ00], [ZH99], [HBC01]. In this paper we are primarily concerned with large-scale key management (which typically involves some form of Public Key Infrastructure (PKI)) and how such a scheme can be formed in an ad hoc manner. We now discuss in more detail the PKI solutions to key management mentioned in the introduction and our TCA-based PKI design.

3.1 Threshold Certificate Authority

Zhou [ZH99] suggests the use of a threshold Certificate Authority to provide scaleable key management found in a traditional Public Key Infrastructure. This TCA-based PKI provides strict trust management unlike a PGP based solution. However by distributing the CA service in a threshold manner the availability is increased and most importantly, it is less vulnerable to compromise if attacked.

One argument levied against the use of a TCA to create a PKI is that users will suffer from Denial of Service if some servers aren't available [AM01]. Aura [AM01] instead prefers a centralised CA approach. His argument is that mission critical services (e.g. 'launch missile') would be more vulnerable to Denial of Service attacks in a TCA based system. Aura [AM01] does not distinguish between the issuance of certificates and the use of certificates. While it is true to say that certificate issuance may be more vulnerable to Denial of Service attacks in a TCA scheme, usage of issued certificates is not more vulnerable. As such a Denial of Service attack on the TCA is not as grave as Aura appears to suggest.

Hubaux regards the TCA-style approach as useful only in a military context where tight security is needed. He prefers the more distributed approach of PGP with every node taking part in the PKI service. We feel that the TCA approach should not be limited to the military environment. A civilian PKI based on a TCA is just as viable, and perhaps necessary in the presence of a large number of nodes issuing false certificates as is possible in the PGP system. The fact that special nodes may be needed to provide the service is acceptable - they may do so for profit, as is the norm on the internet with commercial CA's such as [Verisign] and [Baltimore].

3.2 Dynamic Redistribution of TCA

In fact Kong et al [KZLLZ00], [LL00] propose a TCA design that does not require special nodes. By using proactive share refreshing [HJKY95] the shared private key of the TCA is redistributed throughout the network. Every trusted node in the network takes part in the TCA service. This provides a more distributed service and increases availability. Kong et al [KZLLZ00], [LL00] employ a key share dealer to create the original key shares used by the TCA. In [LL00] it is suggested that the shared dealer poses a single point of failure of the system if it is compromised. The use of shared generation of the RSA keying material by the nodes involved in the Certificate Authority service would remove this single point of failure (the dealer) from their system.

3.3 Shared RSA Key Generation To Create Our TCA

We have implemented this shared RSA key generation functionality. We feel this functionality is necessary not only to remove the single point of failure, but also because a trusted share dealer might not be available or indeed such a trust relationship might not even exist. Using shared key generation, the formation of the TCA can happen 'in the field' without reliance on prior or external security associations like a trusted dealer. The distributed shared key generation algorithm requires secure communication channels between the participants during the computation. These secure channels can be set up 'in the field' by employing key exchange techniques such as those described by Stajano [SJ99] or Balfanz [BSSW02]. Thus, the formation of a scaleable key management architecture is not contingent on external authentication or keying information. Just as an ad hoc network may be created 'from scratch', or without prior infrastructure, so too, a Public Key Infrastructure can be created 'from scratch' without reliance on prior security infrastructure.

3.4 SPKI-style Certificates

The only other key management system in the literature, which allows for small to large scale operation is that of Hubaux [HBC01]. The trust model of Hubaux's system is based on PGP's web of trust. This as we have mentioned is not suitable for all types of network. We want to provide the more flexible and secure alternative facilitated by a certificate authority even at the expense of a less distributed service. Also, we wish to use SPKI-style [SPKI] certificates. We believe that authorisation/attribute certificates will be even more useful in an ad hoc network than on a fixed network. This is because of the lack of centralised services such as access control servers. Identity certificates are often used to verify someone's identity and make an access control decision based on this identity. This requires an online access control server. Using such a centralised facility will not be convenient in an ad hoc network. We suggest that attribute/authorisation certificates will have increased usage potential in an ad hoc networking environment because they make access control information available directly and off-line. PGP web of trust does not facilitate the use of SPKI-style certificates easily. This is one reason why we desire a TCA-based PKI.

4 IMPLEMENTATION AND DESIGN FEATURES

We combine the techniques of Boneh [BF97] with those of Catalano [CGH00] to generate our RSA key shares. We then borrow techniques described by Shoup [Sho00] to facilitate RSA signature generation to create the Certificates provided by the certification sevice.

Boneh's paper outlines an algorithm for the shared generation of a public modulus N and the creation of corresponding private key shares. They describe an (n,n) threshold system, where a partial signature from all the players in the system is required to create the full valid signature. They suggest methods described by Rabin[R98] to create a threshold (t,n) sharing of the private key. However, Rabin uses a two-level sharing scheme to achieve this. The threshold t is achieved by re-sharing each additive share using a (t,n) polynomial sharing as in Shamir [Sha79]. When a node is unavailable t of the nodes can recombine his share and act on his behalf. This scheme is undesirable because of the inefficiency of the two level sharing but more importantly because it requires a lot of interaction during certificate/signature generation. The servers must interact with each other to rebuild all the shares of the unavailable players. This is an undesirable feature for an ad hoc network.

4.1 Additive (t,n) private key sharing

In Malkin et al's [MWB99] implementation of Boneh's protocol [BF97] they provide a (t,n) service by creating multiple additive sharings, one for each possible coalition of t servers. Each player in the TCA has to maintain multiple shares rather than a single one, and during signature generation the coalition of t servers being used implies which share has to be used. This is undesirable for an ad hoc network because it creates a high level of interaction if one of the servers is unavailable. For example, Alice requires a certificate from a (3,5)-threshold Certificate Authority service. Alice decides to use servers $\{S_1, S_2, S_3\}$ to service her certificate request. She collects two of the three partial certificates needed (from S_1 and S_2), but then can't locate the third server S_3 (either un-contactable or corrupt) that she designated at

the start of the protocol. She must now specify a new coalition e.g. $\{S_1, S_2, S_4\}$ to fulfil the certificate request. Unfortunately, because the partial certificates are coalition-dependent she must begin the protocol anew. She must re-contact servers S_1 and S_2 and get new partial certificates from them. Obviously this is undesirable in an ad hoc network where the likelihood of a server going offline is relatively high.

4.2 Non-interactive signature generation

One would like to be able to run the signing protocol without needing prior knowledge of the servers, which will service the certificate request. If the partial signature generation was independent of the coalition used then (following the example above) Alice could collect 3 partial certificates from any of the 5 servers she could locate and combine these to create the certificate. In addition, the use of multiple shares by each TCA server (i.e. different shares depending on the coalition in use) does not scale well and is more difficult to implement. To avoid multiple shares and principally to provide non-interactive signature generation we require a polynomial sharing of the private key between the members of the TCA service. Catalano [CGH00] describes how to efficiently generate a shared polynomial RSA private key. By using Boneh's [BF97] technique to collaboratively generate the shared public key and Catalano's techniques to collaboratively derive the corresponding private key shared via a polynomial we achieve our goal. These shares can not be applied directly to create RSA signatures, however by employing techniques described by Shoup [Sho02] for non-interactive RSA signature generation we are able to complete the process from key generation to signature generation.

The authors know of no other implementation of a shared key generation algorithm, which facilitates this. This work may be of independent interest in fixed networks but our principle design goal was to make a threshold Certificate Authority which was feasible for use in an ad hoc network and didn't rely on a trusted dealer for its inception.

4.3 Feasibility of Shared RSA Key Generation

We have run our shared RSA key generation protocol over WLAN on 3 machines, 2*500Mhz laptop, 1*200Mhz Compaq IPAQ and achieved average 512 bit key generation times of 2.5 minutes. The secure channels required during key generation phase can be created using pair-wise secret keys exchanged between each of the 3 nodes involved over an infrared channel. Other methods of secure association are possible as described by Stajano [SA99] and by Balfanz et al [BSSW02].

5 CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Our implementation of shared threshold RSA key generation closes the gap between local secure association and large-scale key management. We provide the means to build a large-scale key management system without reliance on prior infrastructure. Our solution to key management is truly ad hoc. We have shown that this is practical (in terms of time consumption) on the type of hand held devices in use today. We have made a number of optimisations to our protocol for shared RSA key generation to make it more suitable for an ad hoc networking environment, in particular the provision of non-interactive signatures.

5.2 Future Work

Our shared RSA key generation algorithm works in the honest but curious mode i.e. it is not robust in the face of active adversaries. This means we can't isolate and eliminate misbehaving nodes during shared key generation or signature generation. Creating a robust version of our protocol is possible, following techniques described in [FS01]. However, recent developments by Algesheimer et al [ACS02] may make an implementation of their robust key generation techniques more interesting as it is yet to be seen if they are practical in terms of efficiency. The utility of a TCA service in an ad hoc network needs to be studied. Does the service need to be online? Is certification on an offline face-to-face basis, or is distant certificate renewal useful or sensible?

REFERENCES

[ACS02] J.Algesheimer, J.Camenisch, V. Shoup. "Efficient Computation Modulo a Shared Secret with Application to the Generation of Shared Safe-Prime Products." *Crypo'02,p417-432 LNCS 2442*

[AG00] N.Asokan,P.Ginzboorg. "Key Agreement in ad-hoc networks" *Computer Communications, 23:1627-1637, 2000.*

[AHNR02] B.Awerbuch, D.holmer, C. Nita-Rotaru, H.Rubens, "An on-demand secure routing protocol resilient to byzantine failures." In *ACM Workshop on Wireless Security (WiSe'02)*, Atlanta, Georgia, September 2002

[AM01]T.Aura,S.Maki. "Towards a survivable security architecture for ad-hoc networks" Security protocols 9th international work-shop. Cambridge,UK,April'01, LNCS 2467,p63-73 2002.

[Baltimore] www.baltimore.com

[BSSW02] D. Balfanz, D.K.Smetters, P.Stewart, H.Wong "Talking to strangers:Authentication in ad-hoc wireless Networks" *NDSS '02, San Diego, California, February 2002*

[BF97] D.Boneh,M.Franklin. "Efficient generation of shared RSA keys." *Crypo'97, p425-439 LNCS 1294*

[BW98] K.Becker,U.Wille. "Communicatin complexity of group key distribution" 5th ACM conference on Computer and Communications security,San Francisco,Nov98,ACM press.

[CGH00] D.Catalano,R.Gennaro,S.Halevi "Computing inverses over a shared secret modulus". *Eurocrypt'00, LNCS 1807, Pages 190-206*

[Des87] Y. Desmedt. "Society and group oriented cryptography:A new concept". *Crypto'87 , p120-127 LNCS 293*

[DF89] Y.Desmedt,Y.Frankel "Threshold Cryptosystems", *Crypto'89, p307-315 LNCS 435*

[FS01] "Fully Distributed Threshold RSA under Standard Assumptions", *Asiacryp'01,p310-330 LNCS 2248*

[HBC01] J-P.Hubaux,L.Buttyan,S.Capkun. "The quest for security in mobile ad hoc networks", In proceedings of *MobiHOC'01*

[HJKY95] A.Herzberg,,S.Jarecki,H.Krawczyk,M.Yung. "Proactive Secret Sharing or: How to cope with perpetual leakage", *Crypto'95, LNCS 963, p339-352.*

[KZLLZ00] J.Kong,P.Zerfos,H.Luo,S.Lu,L.Zhang. "Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks". *ICNP'01.*

[LL00] H.Luo,S.Lu "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks". *Technical report UCLA-CSD-TR-200030, October'00,UCLA.*

[MIMwikipedia] http://www.wikipedia.org/wiki/Man-in-the-middle_attack

[MOV97] A.Menezes,P.vanOorschot,S.Vanstone. "Handbook of Applied Cryptography".CRC Press,1997.

[MWB99] M.Malkin,T.Wu,D.Boneh"Experimenting with shared generation of RSA key", In proceedings of the *Internet Society's 1999 Symposium on Network and Distributed System Security (SNDSS)*, pp. 43--56

[PGP] http://www.pgpi.org

[PH02] P.Papadimitratos,Z.Haas, "Secure routing for mobile ad hoc networks." In *Proceedings of the SCS Communication Networks and Distributed Systems, Modeling and Simulation Conference (CNDS'02)*, San Antonio, Texas, January 2002, pp. 27-31.

[R98] T.Rabin. "A Simplified Approach to Threshold and Proactive RSA",*Crypto'98,p 89-104, LNCS 1462*

[SA99] F. Stajano , R. Andersson "*The resurrecting duckling: Security issues in ad-hoc wireless networks* ", in the Proceedings of the 7th International Workshop on Security Protocols, LNCS, Springer-Verlag, 1999.

[Sha79] A.Shamir, "How to share a secret", Communications of the ACM22,1979,pp.612-613

[Sho00] V.Shoup, "Practical Threshold Signatures",*Eurocrypt 2000, LNCS 1807,p207-220.*

[SPKI] http://world.std.com/~cme/html/spki.html

[Verisign] www.verisign.com

[X509]http://www.ietf.org/html.charters/pkix-charter.html

[YNK01] S.Yi, P. Naldurg, R. Kravets, "Security-aware ad hoc routing for wireless networks." In *Proceedings MobiHoc'01*

[ZA02] M.Zapata, N.Asokan "Secure ad hoc on-demand distance vector routing." *ACM Mobile Computing and Communications Review 6*, 3, July 2002, 106-107.

[ZH99] L.Zhou, Z.J.Haas ."Securing Ad Hoc Networks" *IEEE Networks,13(6):24-30,1999.*