A Proposed Public Key Management Scheme for Mobile Ad Hoc Networks

D.S.Dawoud[#1], P.Dawoud[#2] , J.Van der Merwe[#3] , C.Ndagije[#4]
[#1] Professor, Dean-Faculty of Applied Science, National University of Rwanda, Rwanda
Corresponding author email: dsdawoud@gmail.com
[#2]PhD Student, University of Waterloo, Canada,
[#3]Dr. Consultant – SA
[#4]Dr. Senior Lecturer – National University of Rwanda

ABSTRACT
The main objective of this paper is to find a key management scheme that is suitable for mobile ad hoc networks using some of the results introduced in the first part of the paper. The paper proposes a *public key management scheme*, called Ad Hoc Public Key Management (AdHocPKM). AdHocPKM integrates the advantages of distributed key generation, threshold-multisignatures, self-certified public keying and self-certificates to yield a secure, trustworthy key management service with a high availability feature. The proposed scheme is operated solely by the end-users and does not require any *offline* trusted third party or *a priori* sharing of keying material. This addresses the major problem in providing key management services for mobile ad hoc networks, without compromising security.

Key words: Mobile ad hoc networks; security; pair wise key management; public key cryptography; self-certified public keys; self-certificates;

1.0 INTRODUCTION

In a paper submitted to this conference (Dawoud et al, 2011a), we reviewed the researches and publications that provide solutions for the problem of peer-to-peer or pair-wise key management in mobile ad hoc networks. We showed that none of these proposals seem to be entirely suitable for truly unplanned or impromptu mobile ad hoc networks and using them will require a tradeoff between security and the fundamental *ad hoc* characteristics.

In ad hoc networks, to avoid a single point of vulnerability, the trusted third part (TTP) has to be distributed (Zhou & Hass, 1999). The system secret is thus divided up into shares and securely stored by the entities forming the distributed cryptosystem. The main advantage of a distributed cryptosystem is that the secret is never computed, reconstructed, or stored in a single location, making the secret more difficult to compromise, which effectively enhances security (Zhou & Hass, 1999). We discussed the idea of using a *distributed certificate authority* (DCA) as the core of the key management services. A threshold t or more out of the n shareholders forming the DCA are required to cooperatively generate a digital signature to vouch for public key certificates. We introduced in the paper (Dawoud, et al, 2011a) two new proposal: The first is *threshold multisgnature scheme and the second was threshold self-certified public key*. The two proposals are schemes needed to implement the main goal of the current paper.

The main objective of this paper is to propose a key management scheme that is suitable for mobile ad hoc networks. The proposed key management scheme, **Ad Hoc P**ublic **K**ey **M**anagement (AdHocPKM) is based on an *online*, *user empowered*, *user monitored*, *distributed* certificate authority.

2.0 PROPOSED THRESHOLD SELF-CERTIFIED PUBLIC KEY ISSUING PROTOCOL

In this section a self-certified public key generating protocol, *threshold self-certified public keying*, is proposed by modifying the scheme given in (Petersen & Horster, 1997) to make it suitable for ad hoc networks. The modification allows negotiation for a self-certified public/private key pair between a *distributed* certificate authority (DCA) and a single entity

(party *A*) in contrast to the *centralized* certificated authority used in (Petersen & Horster, 1997). The scheme also eliminates the need for a designated combiner node as shown in Figure-1.

In the proposed protocol a DCA and party *A* agree on an *implicitly* self-certified, public key for party *A*, without the DCA learning the corresponding private key. Party *A* then signs the self-certified public key y_A and certificate information *CIA* to yield a self-certificate $SelfCert_A$ that provides y_A with *explicit* authentication. The DCA is constructed as explained in (Dawoud et al, 2011a).

The proposed protocol is as follows:

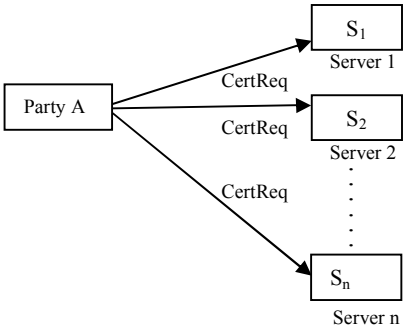

Figure 1: Distributed certificate authority (DCA) without any combiner server

1) Party *A* chooses a random number $a \in_R Z_q$ and identity ID_A, computes $r_A = g^a$ and transmits a certification request (CertReq) containing (ID_A, r_A) to *n* servers forming the DCA.
2) The DCA servers who received the certification request each prepares certification information CI_A depending on the DCA's certification policy. For example, the DCA can use party *A*'s identity, DCA's identity, party *A*'s public parameter *rA*, certificate serial number, date of issue, DCA's public key and other relevant extension information to yield: $CI_A = (ID_A \| ID_{DCA} \| r_A \| CertNo \| IssueDate \| CertPeriod \| y_{DCA} \| Ext)$.
Each of the DCA servers computes the signature parameter, $s_A = h(CI_A)$ and collaborate to generate a *threshold-multisignature* (*R, S,* $h(y)$) on s_A as explained in (Dawoud et al, 2011a). Each server P_i, $i \in \alpha$ sends (s_i, r_i), (*R, S,* $h(y)$), s_A and CI_A to party *A*, which securely store all information.
3) Party *A* verifies the threshold-multisignature, (*R, S,* $h(y)$) on signature parameter, s_A. Party *A* then computes the private key $x_A = s_A + a$. The tuple (r_A, x_A) can be seen as the signature of the DCA on the certification information CI_A. Party *A* then calculates its corresponding public key using Equation-1:

$$y_A \equiv g^{x_A} = g^{h(CI_A)} \cdot r_A \qquad (1)$$

4) Party *A* signs (CI_A, (*R, S,* $h(y)$), y_A) with its private key x_A to generate the self-certificate of y_A, with the following structure: $SelfCert_A = PS_{x_A}(CI_A \| (R, S, h(y)) \| y_A \| UserIssueDate \| Ext)$.
Party *A* publishes $SelfCert_A$, the signature of which can be verified using y_A. The public key y_A of *A*, can be publicly verified by checking that the signature (*R, S,* $h(y)$) on $h(CI_A)$ is correct and evaluating Equation-1. In the remainder of the text the proposed algorithm will be referred to as *threshold self-certified public keying*.

2.1 Self-Organized Key Renewal

The key pair (x_A, y_A) is the basic public/private key pair obtained by party *A* via the self-certified public key issuing protocol as given above in Section-2. In the self-certificate

generation protocol due to Lee *et al* (Lee & Kim, 2000) users can renew their public/private key pair and self-certificate to generate a renewed key pair (x'_A, y'_A) and a renewed self certificate $SelfCert'_A = PS_{x'_A}(CI_A \,||\, (R, S, h(y)) \,||\, y_A \,||\, y'_A \,||\, UserIssueDate' \,||\, Ext')$. This can also be referred to as *self-organized* key renewal.

To make the *user-controlled* key renewal procedure as given in (Lee & Kim, 2000) compatible with the proposed self-certified public key issuing protocol for a distributed certificate authority given in Section-2, the verification equation must be modified to the following:

$$y'_A = g^{h(CI_A)h(CI_A, r'_A)} r_A^{h(CI_A, r'_A)} r'_A \quad (2)$$

Discussion on self-organized key renewal: Important advantages of the key renewal scheme are given (Lee & Kim, 2000):

- Users can renew their own self-certified key pairs (x'_A, y'_A) without contacting the distributed certificate authority.
- An adversary not knowing party *A*'s private key x_A cannot forge a renewed key pair (x'_A, y'_A).
- If the basic key pair (x_A, y_A) is not used for any encryption (real communication) and only for key renewal, it is not vulnerable to any *shortcut* public key encryption scheme attacks (for example the adaptive chosen message attack) .
- In the case of an adversary compromising x'_A it is not possible for the adversary to gain any knowledge of any subsequent renewed key pairs (x''_A, y''_A).

The simplistic key renewal protocol therefore not only enhances security, but empowers the users to control the frequency of their key renewal. Since users are able to update their keys whenever they suspect that it has been compromised or lost, the trust in key pairs and authenticity of certificates are enhanced.

3.0 PROPOSED PUBLIC KEY MANAEMENT SCHEME FOR MOBILE AD HOC NETWORKS

The proposed key management scheme, Ad Hoc Public Key Management (AdHocPKM) is an integration of distributed key generation (Zhang & Imai, 2003), the new proposed threshold-multisignature scheme (Dawoud et al, 2011a), secret redistribution (Wong et al, 2002), *threshold self-certified public keying* (Dawoud et al, 2011) and self-certificates (Lee & Kim, 2000).

3.1 System Model

In the proposed scheme we consider a network of wireless nodes with low to moderate mobility speeds ($4m/s - 10m/s$). The medium access control (MAC) and routing mechanisms are assumed to be generic (for example the IEEE 802.11 MAC protocol and Ad hoc On Demand Distance Vector (AODV) routing protocol are currently practical). The network is open for any user to join at random which is inherited from the paper's main assumption that the users do not share any prior relationships. The specific application of the scheme will therefore not be for military type mobile ad hoc networks, which have a high security demand, but rather for commercial or other less resource constrained environments. Delegates meeting for the first time at a conference are a good example where the proposed key management scheme will be useful.

The first group of nodes that collaborates to form the network constructs a distributed certificate authority (DCA) using a distributed key sharing protocol. A joining node that is not part of the DCA contacts the DCA only once to obtain a self-certified public/private key pair. It is the responsibility of the DCA not to issue keying material to each unique node identity more than once. After the key negotiation process is completed a node becomes its own authority domain. Each node thus generates, disseminates and renews its own self-certificate without further interaction with the DCA.

3.2 Adversary Model

An adversary is a malicious node that uses every means available to break the proposed key management scheme. Any *active* adversary can eavesdrop on all the communication between nodes, modifies the content of the messages and injects them back into the wireless channel. When a networking node is compromised all its public and private information is exposed to the adversary. The adversary will then use this information to attempt forging the public and private information of other nodes. The operation of AdHocPKM is divided into an initialization phase and a post-initialization operation.

3.2.1 Initialization phase operation of AdHocPKM

The initial phase of the AdHocPKM scheme executes in two steps:

Step 1) Generation of distributed certificate authority: The nodes collaborate to construct the distributed certificate authority (DCA) as shown in Figure-3. As explained in (Dawoud et al, 2011a) the nodes firstly agree on system parameters. Next, each node generates and broadcasts a self-certificate as detailed in (Dawoud et al, 2011a). All nodes P_i who generated a self-certificate form the set X. The nodes must reach consensus on which P_i, $i \in X$ will participate in the distributed key generation (DKG) protocol. In (Yi & Kravets, 2003) Yi *et al* point out that the DCA should comprise of the nodes with the most computational, memory and energy resources, to take advantage of the *heterogeneity* of ad hoc networks. If the decision criteria cannot include node capability, the DCA can simply be formed by the nodes with the highest or lowest n identities (ID_i). The chosen P_i may use the DKG protocol due to Zhang *et al* (Zhang & Imai, 2003) or any other equivalent DKG protocol, to realize the *initial* share distribution between the servers forming the DCA. Note that the system parameter generation/agreement process is inherently part of the DKG protocol. All remaining nodes who do not participate in the DKG protocol can closely monitor the *public verifiable* secret distribution process. All information required to validate correct share distribution is readily available in the broadcast channel.

Step 2) Creation of self-certified public/private key pairs and self-certificates: In this step P_i, $i \notin Q$ use the proposed key issuing scheme, given in Section-5, threshold self-certified public keying, to request certification from the DCA. After completing the DKG protocol with the DCA each node P_i, $i \notin Q$ performs a self-organized key renewal procedure as explained in Section-2.1 to generate a renewed public/private key pair (x'_i, y'_i) and self-certificate $SelfCert'_i$ that can be used for real communication. The DCA servers keep records of all certificates and do not issue an initial certificate with the same identity ID_i more than once.

3.2.2 Post-initialization phase operation of AdHocPKM

The post-initialization operation of AdHocPKM considers the following instances: joining of a new node; certificate exchange and authentication; certificate revocation, update/redistribution of DCA private key shares and a compromised DCA.

Joining of a new node: If a new node joins the network, the node requests neighboring nodes' self-certificates to learn the public key of the DCA. The new node then contacts the DCA for its *own* self-certified public key and generates a self-certificate as explained in Step 2 of the initialization phase.

Certificate exchange and authentication: Assume node A wants to privately communicate with node B. Node A can obtain the certificate of node B from any other node in the network (including node B or DCA servers). To authenticate the self-certificate $SelfCert'_B$ node A performs the following steps:

1) Verify the signature of node B using y'_B. This provides *key confirmation* making y'_B an *explicitly* authentic public key (Lee & Kim, 2000).

2) Use the public key of the DCA $y_{DCA} = y_Q$ to verify the signature (R, S, $h(y)$) of the DCA on the certificate information CI_B of node B.

3) Use Equation-2 to validate the authenticity of node B's public key y_B.

If the steps above yield a positive result then node *A* is guaranteed that the binding between the public key y'_B of node *B* and identity ID_B is authentic, assuming the intractability of the discrete logarithm problem.

Certification revocation: AdHocPKM has two low cost certificate revocation schemes, *Self-revocation* and *DCA-revocation*: In the *Self-revocation* scheme the users themselves can revoke their certificate $SelfCert'_i$ if the user believes that its public/private key pair (x'_i, y'_i) is no longer valid or the binding expressed in the certificate is violated. Without any interaction with the DCA the user simply renews the key pair using the *self-organized* key renewal protocol (Lee & Kim, 2000) as explained in Section-2.1. This will yield a new self-certificate $SelfCert''_i$ which contains a new key pair (x''_i, y''_i) authentically bound to the user's identity ID_i. In the *DCA-revocation* scheme the self-certificate validity period *CertPeriod* has expired and the users may request certificate renewal from the DCA. The *threshold self-certified public keying* protocol is used as given in Section-5. Note that the node will not use a new identity but transmits its expired certificate to the DCA servers instead. The user has three options to redistribute its renewed certificate: The user can broadcast the renewed certificate to the entire network, issue the renewed certificate to any node requesting communication or send the renewed certificate to frequent contacts. The latter of these would be the most appropriate if a *frequent contact list* is available. *Update/redistribution of DCA private key shares:* As pointed out in (Herzberg, 1997) the secret shares of the threshold cryptosystem have to be periodically updated after elapse of a time period *T*, to make the system proactively secure. Mobile adversaries thus have to compromise more than threshold t shareholders within *T* to break the system. Nodes that form the DCA will not always be able to provide certification services. For example nodes may be compromised, move out of transmission range, become disconnected from the network due to depleted (battery) resources or become selfish (Buttyan &Hubaux, 2003). To sustain the availability of the DCA a secret redistribution protocol can be used to redistribute the private key of the DCA to a new access structure $\Gamma_{P'}^{(n',t')}$. The distributed key generation protocol due to Zhang et al (Zhang & Imai, 2003) that is used to construct the DCA, can be easily changed into a DKG protocol with little modification. In fact, analysis by the authors have shown that the same protocol can be used for periodic secret share updates by keeping the access structure constant, i.e., $\Gamma_{P'}^{(n',t')} = \Gamma_{P}^{(n,t)}$. The system parameters (n, t) provide the key management service with a tradeoff between availability and security, where *n* is the total number of nodes forming the DCA and *t* the threshold of the cryptosystem. Increasing *n* increases availability but lowers the security of the system. Conversely, increasing *t* decreases the availability of the certification service but increases the security level. The nodes must agree on an appropriate set (n, t) which can be influenced by numerous factors, which are application dependent. This is subject to future investigations. The *secret redistribution protocol* enables the key management service to dynamically adjust the security/availability tradeoff needed for the current networking environment. The reconfiguration of the DCA makes the key management service as a whole, a scalable solution.

Compromised DCA: If network participants accumulate evidence that *t* or more of the DCA servers are compromised, the nodes distribute the evidence to the other network participants and construct a new DCA as explained in Step 1 of the initial phase operation of AdHocPKM (Section-3.3). For example two certificates with the same identity constitute public proof that the DCA has been compromised and will require an immediate reconstruction of the DCA.

4.0 DISCUSSION AND COMMENTS ON ADHOCPKM

In contrast to existing key management solutions (Zhou & Hass, 1999) (Yi & Kravets, 2004) (Yi & Kravets, 2003), the proposed scheme uses a *public verifiable*, round optimal, distributed key generation protocol (Dawoud et al, 2011a) to construct the DCA. This effectively eliminates the need for a common *offline* TTP or any *a priori* shared keying material. The DCA uses a proactively secure threshold-multisignature scheme to sign certificate information. Nodes forming the DCA are thus held accountable for their signatures

and dishonest nodes can be identified and disqualified when cheating. Repudiation of individual signatures can be proven not to be possible nor can an individual signer be impersonated without being compromised. Adversaries forming part of the distributed certificate authority cannot gain any bene fit from generating invalid signatures since they will be disqualified on their first attempt to cheat. Accordingly, it is anticipated that cheating will be discouraged. When nodes join the network they only need to contact the DCA *once* to acquire a self-certified public key pair and distribute their certificate in a self-organized way. In existing solutions the nodes need to contact the DCA every time they require certification services. This feature significantly reduces the computational and communication overhead on the network and in particular on the servers forming the DCA. The certificate revocation schemes are low cost and robust. Certificate validity period can also be increased since the nodes' basic public/private key pair (x_i, y_i) is never used for any real communication. In contrast to the certification chaining approach (Capkun et al, 2003), certification can be guaranteed between any two communication entities and the authenticity of certificates are based on the intractability of the discrete logarithm problem and not on the trust between nodes in the network. Users can explicitly authenticate certificates by verifying the self-certificate of the unknown user and evaluating Equation-2 in Section-2.1.

The proposed key management scheme, AdHocPKM, is not bound to the use of a specific *threshold-multisignature* scheme (Dawoud et al, 2011a)or *self-certified public key issuing* protocol (Section-2). These may be replaced by more secure and efficient schemes, if such schemes are available. The main concern with existing key management solutions, which rely on a distributed certificate authority (Zhou & Hass, 1999) (Yi & Kravets, 2004), is the availability of the certification service, i.e., the *Success ratio* = (*Number of successful certification requests)/(Number of total certification requests)* This problem has been extensively investigated through simulations by Yi *et al* (Yi & Kravets, 2004) and Carter *et al* (Carter et al, 2003). The simulation results in (Yi & Kravets, 2003) concluded that a *success ratio* of more than 90% was achieved under all mobility scenarios, using flooding as the communication method to reach the DCA servers. In (Carter et al, 2003) it is proved that *manycast* can be used to further increase the *success ratio*. Since the scheme proposed in this paper is using the same networking environment and communication instance as those considered in (Carter et al, 2003), it can thus be concluded that the simulation results are applicable to AdHocPKM.

5.0 CONCLUSION

This paper addresses the issue of key management in mobile ad hoc networks. The proposed scheme is only operated by the end-users and does not require any *offline* trusted third party (TTP) or *a priori* sharing of keying material. This solves the major problem of many existing key management proposals for mobile ad hoc networks (Zhou & Hass, 1999) (Yi & Kravets, 2004) (Kong et al, 2001) that use a *distributed certificate authority* (DCA). In the existing proposals the DCA needs to be empowered with a common *offline* TTP, which is not guaranteed to be available in ad hoc networks. The paper used two proposed schemes given in (Dawoud et al, 2011a). The first is the proactively secure *threshold-multisignature* scheme that does not need any trusted authority. This threshold signature scheme gives the signature verifier a guarantee that at least t of the group members participated in the generation of the group signature and also allows the verifier to establish the identities of the signers. The scheme is secure against the *universal forgery* attacks in contrast to the threshold signature schemes in (Li et al, 2001) (Wang et al, 1998) the signers are traceable. The second is the *threshold self-certified public keying protocol proposed in our paper*, which allows entities to establish a self-certified public key (Petersen & Horster, 1997) (Lee & Kim, 2000) with a *distributed* trusted authority as opposed to a *centralized* authority. The two protocols are used to introduce, in this paper, a novel public key management scheme, Ad Hoc Public Key Management (AdHocPKM) was introduced, which allows users, with the aid of a DCA, to create, store, distribute and revoke their own public key material in a self organized way yielding a trustworthy and secure key management service with a high availability feature. AdHocPKM uses self-certified public keys and self certificates to provide a mechanism that

enhances the trust of users in the key management service. AdHocPKM uses a *publicly verifiable* distributed key generation protocol, a *publicly verifiable* secret redistribution protocol and threshold digital signature scheme with *traceable* signers, allowing all operations of the DCA to be monitored by all network participants. A cheating shareholder can be detected and disqualified. The *secret redistribution* protocol makes the DCA dynamically scalable and allows the key management service to adapt to the networking environment. The proposed scheme's certificate service focuses primarily on establishing keying material for the routing infrastructure. AdHocPKM do not explicitly define application level key establishment techniques. The proposed scheme can easily be integrated with those defined in (Capkun et al, 2004) to effectively set up keying material on both the network and application layer.

6.0 REFERENCES

Bechler M., H.-J. Hof, D. Kraft, F. Pahlke, and L. Wolf, 2004 "A Cluster-Based Security Architecture for Ad Hoc Networks," in *proc. 23rd Conf. of the IEEE Communications Society (INFOCOM'04)*, March, 7-11 2004.

Buttyan L. and J. P. Hubaux, 2003 "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592.

Capkun S., L. Buttyan, and J.-P. Hubaux, 2003,"Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64.

Carter C., S. Yi, P. Ratanchandani, and R. Kravets, 2003 "Manycast: exploring the space between anycast and multicast in ad hoc networks," in *proc. 9th Annual International Conf. on Mobile Computing and Networking (MOBICOM'03)*.

Dawoud,D.S., P.Dawoud, J.Van der Merwe, C.Ndagije ,2011"A new Multithreshold Scheme for Mobile ad hoc Networks," aet2011, Uganda

Herzberg, A., M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung, 1997, "Proactive Public Key and Signature Systems," in *proc. 4th ACM Conf. on Computer and communications security*, April, 1-4 1997.pp. 1120–1121.

Hwang T.,C.-M. Li, T. Hwang, and N.-Y. Lee, 1994, "Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders," in *proc. Advances in Cryptology - EUROCRYPT'94*.

Khalili A., J. Katz, and W. A. Arbaugh, 2003, "Towards secure key distribution in Truly Ad-Hoc Networks," in *proc. IEEE Workshop on Security and Assurance in Ad-Hoc Networks*.

Kong J., P. Zerfos, H. LUO-----, S. Lu, and L. Zhang, 2001, "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks," in *proc. Ninth International Conf. on Network Protocols (ICNP'01)*.

Lee W.-B. Lee and C.-C. Chang, 1999 "(t, n) Threshold Digital Signature with Traceability Property," *Journal of information Science and Engineering*, vol. 15, no. 5, pp. 669–678.

Li ,Z.-C., J.-M. Zhang, J. LUO-----, W. Song, and Y.-Q. Dai, 2001, "Group-oriented (t, n) threshold digital signature schemes with traceable signers," in *proc. Topics in Electronic Commerce, Second International Symposium, ISEC 200*.

Ngai , E. C. H., M. R. Lyu, and R. T. Chin,2004 "An Authentication Service Against Dishonest Users in Mobile Ad Hoc Networks," in *proc. IEEE Aerospace Conf.*.

Petersen ,H. and P. Horster, 1997,"Self-certified keys - Concepts and Application," in *proc. Third Conf. on Communication and Multimedia Security*.

Wong ,T. M., C. Wang, and J. M. Wing, 2002, "Verifiable Secret Redistribution for Archive System," in *proc. First International IEEE Security in Storage Workshop*.

Zhang, R. and H. Imai, 2003 "Round Optimal Distributed Key Generation of Threshold Cryptosystem Based on Discrete Logarithm Problem," in *proc. Applied Cryptography and Network Security (ACNS'03*.

Zhou ,L. and Z. J. Haas, 1999,"Securing Ad Hoc Networks," *IEEE Network: special issue on network security*, vol. 13, no. 6, pp. 24–30.