Fully Self-Organized Peer-to-Peer Key Management for Mobile Ad Hoc Networks

Johann van der Merwe
University of KwaZulu-Natal
School of Electrical, Electronic and Computer Engineering
South Africa
vdmerwe@ukzn.ac.za

Dawoud Dawoud
University of KwaZulu-Natal
School of Electrical, Electronic and Computer Engineering
South Africa
dawoudd@ukzn.ac.za

Stephen McDonald
University of KwaZulu-Natal
School of Electrical, Electronic and Computer Engineering
South Africa
mcdonalds@ukzn.ac.za

ABSTRACT

Mobile ad hoc networks (MANETs) offer communication over a shared wireless channel without any pre-existing infrastructure. Forming peer-to-peer security associations in MANETs is more challenging than in conventional networks due to the lack of central authority. The main contribution of this paper is a low complexity key management scheme that is suitable for fully self-organized MANETs. The proposed peer-to-peer key management scheme uses subordinate public keys and crypto-based identifiers to eliminate any form of trusted third party. Nodes can create, disseminate and revocate their own keying material with low communication and computational overhead. We show how localized certificate exchanges on the network layer can be used to break the routing-security interdependence cycle without degrading the performance of the network. Our proposed solution is also generic since it can be deployed in any "open" mobile wireless network with symmetric or asymmetric encryption.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Distributed networks; Wireless communication; C.2.2 [**Computer System Organization**]: Network Protocols; K.6.5 [**Computing Milieux**]: Security and Protection

General Terms

Algorithms, Design, Management, Performance, Security

Keywords

Mobile ad hoc networks, network security, self-organization, peer-to-peer key management, pairwise key management, network level key distribution, identity-based cryptography, crypto-based identifiers, Mobile IPv6, subordinate public keys

WiSe'05 September 2, 2005, Cologne, Germany.

1. INTRODUCTION

Mobile ad hoc networks (MANETs) eliminate the need for pre-existing infrastructure by relying on the nodes to perform all network services. The connectivity between the nodes is sporadic due to the shared, error-prone wireless medium and frequent link breakages caused by node mobility. Fully self-organized MANETs are created solely by the end-users for a common purpose in an ad hoc fashion. Impromptu, self-organized MANETs can be informally visualized as a group of strangers, people who have never met before, coming together for a common purpose. These people have no prior relationships and share no common keying material on their nodes. Users therefore have to establish security associations between themselves after network formation without the aid of *a priori* shared keying material or any form of *off-line* trusted third party (TTP).

Several solutions for peer-to-peer key management schemes have already been proposed for MANETs [20] [10] [19] [4] [5]. From these existing solutions, [4] [5] proposed schemes for self-organized MANETs.

The main contribution of this paper is a peer-to-peer key management scheme that is suitable for fully self-organized MANETs, with any form of off-line or on-line TTP eliminated. Self-organized MANETs inherently will not find application where user access to the network is restricted. The proposed scheme is designed mainly for "open" MANETs where any person with the appropriate equipment can join or leave at random without contacting any trusted authority.

The proposed key management scheme, Self-Organized Peer-to-Peer Key Management (SelfOrgPKM), leverages a variant of the ElGamal type signature scheme, *subordinate public keys* and crypto-based identifiers [14] [12] to achieve low protocol complexity without introducing excessive communication and computational overhead. SelfOrgPKM allows nodes to initialize themselves by generating their own keying material before joining the network. The scheme's operation is fully self-organized, with the burden of key management uniformly distributed between all network participants. Each node is thus its own authority domain which, similar to previous schemes [4] [5], is our main assumption. The nodes establish security associations with their one-hop neighbors on the network layer during route establishment and on the application layer on a need to know basis. We say a bidirectional security association between two nodes, A and B, exists if the nodes have exchanged their tuples $[K_A,\ ID_A]$ and $[K_B,\ ID_B]$, where K_i is node P_i's keying

material (symmetric or asymmetric keys) and ID_i a unique network identifier/address. Note that K_i and ID_i have to be authentically bound, with the binding between the keying material and node identifier/address publicly verifiable. As with most "open" networks it is left to the user controlled applications to further bind the tuple $[K_i, ID_i]$ to a unique user name.

Subordinate public keys are defined here as public keys that are derived from a user's *self-generated* primary or base public/private key pair. We impose the following properties on subordinate public keys:

1) A valid subordinate public key can only be generated if the entity knows the base private key.

2) The user can *self-generate* a renewed subordinate public key as frequently as needed.

3) The subordinate private key must be statistically independent of the base private key and other renewed subordinate private keys, i.e. a compromised subordinate private key does not reveal any information about the user's base private key or any future renewed subordinate private keys.

4) There must exist a binding between the user's base public key and subordinate public key that supports non-repudiation.

The paper is organized as follows: in Section-2 the related work is briefly surveyed. Section-3 presents a variant on the generalized ElGamal type signatures as a strong cryptographic building block for the proposed subordinate public key generation scheme. Section-4 introduces a new subordinate public key generation scheme. In Section-5 the new peer-to-peer key management scheme, SelfOrgPKM, for self-organized MANETs is proposed. Section-6 discusses the security, performance and features of the proposed peer-to-peer key management scheme. Some conclusions are provided in Section-7.

2. RELATED WORK

The majority of existing schemes, for example [20] [10] [19], are based on variations of a distributed certificate authority (DCA) that is held responsible for vouching for the authenticity of keying material. An off-line TTP is used to initialize the DCA nodes. The collection of DCA nodes, on the other hand, can be seen as a distributed on-line TTP. In contrast to conventional networks, the certificate authority has to be distributed to avoid a single point of attack [20].

Capkun *et al.* [4] present a self-organized public key management scheme based on Pretty Good Privacy (PGP) [21]. Similar to PGP, each node disseminates its own certificates and keeps a certificate repository comprising of the certificates of nodes in its local neighborhood. Users share their certificate repositories and mutually authenticate each other's certificate by finding a certificate chain linking their certificates.

Montenegro *et al.* [12] and Bobba *et al.* [3] use crypto-based-identifiers to bind node identifiers to public keys. The notion of using partial hashes of a mobile node's public key to form its network address originated with O'Shea *et al* [14] as an authentication protocol to protect Mobile IPv6 from address falsification. The crypto-based addresses are used in [3] to protect the basic exchanges between nodes and to bootstrap the routing security mechanism, effectively breaking the routing-security interdependency cycle and solving the address ownership problem.

Lately in [5], Capkun *et al.* have proposed a peer-to-peer key management scheme that relies on user mobility to bring nodes within each other's transmission range which allows them to exchange their certificates without relying on a secure routing infrastructure. The fully self-organized version of the scheme requires nodes to use a secure side channel between the users' personal devices to authenticate each other and to set up shared session keys. The secret side channel can be a short range connectivity system such as infrared or a physical wire [5].

The impact of these protocols on the development of the proposed key management scheme, SelfOrgPKM, is considered in Section-6.

3. MODIFIED ELGAMAL SIGNATURE SCHEME

In this section a *modified* ElGamal type signature scheme is presented, developed from the generalized ElGamal signature introduced by Horster *et al.* [8]. The presented ElGamal variant will be used as a strong cryptographic building block for the proposed subordinate public key generation scheme in Section-4.

3.1 System parameter setup

The following system parameters are generated as usual:

p, q	two large primes, such that $q \mid (p-1)$.
g	generator of the cyclic subgroup of order q in $(Z)_p^*$.
$H(\cdot)$	collision free one-way hash function.
x_P	private key of user P.
y_P	public key of user P, where $y_P = g^{x_P} \ mod \ p$.

3.2 Signature generation

User P selects a random number $k \in [1, q-1]$ and computes a public commitment r as: $r = g^k \ mod \ p$.

User P signs an arbitrary message m by solving the following congruence:

$$s \equiv x_P + [H(m \parallel r)]k \ mod \ q \tag{1}$$

The set (s, r) is the signature of user P on message m.

3.3 Signature verification

Any outsider can use user P's public key y_P to verify the validity of the signature (s, r) for a message m by checking whether the following equation holds:

$$g^s = y_P r^{H(m \| r)} \ mod \ p \tag{2}$$

4. PROPOSED SUBORDINATE PUBLIC KEY GENERATION SCHEME

The proposed subordinate public key generation scheme is based on the modified ElGamal signature variant presented in Section-3 and borrows concepts from *parameter hidden* signature schemes [9].

The system parameters introduced in Section-3.1 are applicable. It is assumed that party A has generated its own public key/private key pair as follows: party A chooses a random number $x_A \in_R [1, q-1]$ as its private key and computes its corresponding public key as $y_A = g^{x_A} \ mod \ p$.

Party A can generate a subordinate public key from its base key pair (x_A, y_A) that satisfies the properties defined in Section-1 as follows:

- Party A chooses a random number $k_A \in_R [1, q-1]$ and computes $r_A = g^{k_A} \ mod \ p$.
- Party A computes its new subordinate private key as:

$$x'_A = x_A + H(KI_A)k_A \ mod \ q, \quad (3)$$

where the subordinate key information is defined as $KI_A = [ID_A \parallel y_A \parallel r_A \parallel SerNo \parallel IssueDate \parallel ValPeriod \parallel ExtInfo]$. Note that the contents of KI_A can be altered based on the network policy, where ID_A is the identity of party A, $SerNo$ a unique sequence number, $IssueDate$ the date of issuing the certificate, $ValPeriod$ the validity period and $ExtInfo$ some additional extension information.

- Finally party A computes its corresponding subordinate public key as:

$$y'_A = g^{x'_A} = y_A(r_A)^{H(KI_A)} \ mod \ p \quad (4)$$

Party A can renew its subordinate key pair with a self-organized subordinate key renewal procedure: party A simply chooses a new random number $k'_A \in_R [1, q-1]$ and computes its renewed subordinate private key as:

$$x''_A = x_A + H(KI'_A)k'_A \ mod \ q, \quad (5)$$

where $KI'_A = [ID_A \parallel y_A \parallel r'_A \parallel SerNo + 1 \parallel IssueDate' \parallel ValPeriod' \parallel ExtInfo']$.

5. PROPOSED SELF-ORGANIZED PEER-TO-PEER KEY MANAGEMENT SCHEME

The proposed peer-to-peer key management scheme for MANETs, called SelfOrgPKM, uses subordinate public keys (presented in Section-4) and crypto-based identifiers [14] [12] as strong cryptographic building blocks to set up security associations between nodes. The bootstrapping of the security service introduces minimal communication and computational overhead and does not require any form of off-line or on-line TTP. This property is consistent with the characteristics of self-organized, impromptu MANETs.

5.1 System Model

In the proposed scheme we consider a network of wireless nodes with low to high mobility speeds ($1m/s - 20m/s$). The medium access control (MAC) and routing mechanisms are assumed to be generic. The network is open for any user to join or leave at random without restricted access. 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 access constrained environments.

We assume that there is no pre-existing infrastructure and no form of on-line or off-line trusted authority. Before users join the network they have to determine what universal set of system parameters are used in the network. We assume that the users have system parameter sets pre-imaged on their nodes and that users publicly establish which set to use. Secure system parameter exchanges, without prior knowledge or any user interaction, is an interesting direction for future research. Each node generates a discrete logarithm public/private key pair. By hashing the base public key with a one-way collision resistant hash function, nodes obtain a unique network identifier/address. Our proposal inherits this idea from the original proposal of O'Shea *et al* [14]. The base or originally generated key pair is never used for any real communication to protect it from attacks on the cryptographic algorithms used to secure communication. The users rather derive a second or subordinate public/private key pair from their base key pair as specified in Section-4. Each user then generates a self-certificate to bind other useful information to their keying material such as a unique sequence number, expiry date etc.

Any user with a valid self-generated certificate can join the network. As in the initialization phase, each node is its own authority domain during network operation and therefore responsible for the renewal and dissemination of its own self-certificates. Nodes within each other's transmission range exchange certificates during route establishment on the network layer and users exchange their certificates on the application layer on a need to know basis, thus only when they want to communicate securely.

5.2 Adversary Model

We consider a straightforward general 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, modify the content of messages and inject them back into the wireless channel. When a node is compromised all its public and private information is exposed to the adversary.

The operation of SelfOrgPKM is divided into a node initialization phase which is executed by each node before the node joins the network and post-initialization which executes during network operation.

5.3 Initialization Phase of SelfOrgPKM

Each node P_i, for $(1 \leq i \leq n)$, creates a base public/private key pair (x_i, y_i) by choosing a random number $x_i \in_R [1, q-1]$ as its base private key and computes its corresponding public key as $y_i = g^{x_i} \ mod \ p$. It is assumed that each node has an authentic image of the system parameters, as specified in Section-3.1.

Each node generates a unique identifier (ID_i) that is bound to its base public key y_i as follows:

$$ID_i = H(y_i) \quad (6)$$

SelfOrgPKM requires ID_i to be used as the node's network address or as a fixed part of the address. Note that this requirement places no constraint on the structure of the network addresses: the entire hash output, ID_i, can be used in MANETs with flat, static addresses or only a part of the output can be used in MANETs with dynamic addressing. We note that SelfOrgPKM can easily be extended to incorporate strong network access control at the cost of losing the self-organized feature: an off-line trusted authority can bind ID_i, y_i and a unique username by generating a certificate for P_i signed by the authority.

Each node P_i uses its base public/private key pair (x_i, y_i) to generate a subordinate public/private key pair (x'_i, y'_i) as specified in Section-4.

Note that P_i's base key pair (x_i, y_i) is never used for real communication. Rather, each P_i uses its subordinate key pair (x'_i, y'_i) for securing actual communication.

To obtain an explicitly authentic key pair each node uses

its newly obtained subordinate private key x'_A to sign the key information content, KI_i (concatenated with its subordinate public key y'_i and public commitment β'_i) via the modified ElGamal signature scheme presented in Section-3. Node P_i's self-certificate can then be defined as: $SelfCert'_i = [KI_i \parallel y'_i \parallel \alpha'_i \parallel \beta'_i]$, where (α'_i, β'_i) is the appended signature on $KI_i \parallel y'_i \parallel \beta'_i$.

5.4 Post-Initialization Phase of SelfOrgPKM

The post-initialization phase commences after network formation. Each node must perform the initialization phase, as presented in Section-5.3, before joining the network.

5.4.1 Certificate exchange and authentication

Certificate exchange takes place between nodes on a peer-to-peer, need to know basis. Nodes set up a bidirectional security association by exchanging their renewed self-certificates, $SelfCert'$. SelfOrgPKM requires all nodes to exchange self-certificates with their one-hop neighbors on the network layer. As shown in Figure-1, nodes within each other's transmission range exchange their certificates during route establishment. We will limit our scope to on-demand routing without binding our scheme to a specific routing protocol. The source node starts as usual with a broadcast route request. Two unicast messages are needed for subsequent certificate exchange (if the source and neighboring node have not done so already): one message from the neighboring node and one message from the source node. This process continues until the route request reaches the destination node, hence the route requests serve as triggers for certificate exchanges between neighboring nodes in the route discovery phase.

It is trivial to see that the one-hop network layer certificate exchange mechanism makes the security scheme independent of the routing-security interdependence cycle as defined in [3].

The example, illustrated in Figure-1, also explains certificate exchanges at the application layer: assume $Node_A$ wants to communicate securely with $Node_B$. In the first round $Node_A$ sends to $Node_B$ a $CertRequest$ requesting $SelfCert'_B$ from $Node_B$ over the established route. Note that $CertRequest$ contains the certificate $SelfCert'_A$. If $Node_B$ grants the request it replies in the second round with $SelfCert'_B$. Note that this two round procedure requires no synchrony between $Node_A$ and $Node_B$. The self-certificates and subordinate public keys of $Node_A$ and $Node_B$ are authenticated as follows:

1) Each node implicitly authenticates the base public key of its peer node by checking if Equation-6 holds.

2) Next, the peer nodes implicitly authenticate the subordinate public key of their peers by checking if Equation-4 holds.

3) Finally each node validates the self-certificate of its peer node, $SelfCert'_i$, by verifying the signature (α'_i, β'_i) on $[KI_i \parallel y'_i]$. This explicitly authenticates both the base public key y_i of $Node_i$ and the subordinate public key y'_i. If the node identifier ID_i matched the hash output the node is also assured that the identifier/address has not been spoofed by $Node_i$.

If all three of the above verification steps hold then the subordinate public key y'_i is explicitly authentic and securely bound to the base public key y_i which in turn is securely bound to the nodes *statistically* unique identifier/address.

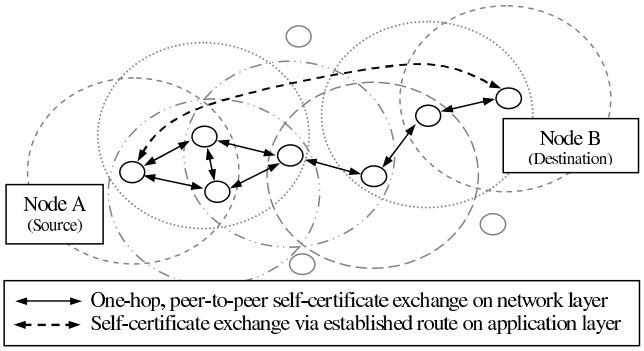

Figure 1: SelfOrgPKM certificate exchange

If the certificate transfer fails due to error-prone connectivity or unsuccessful authentication, $Node_A$ will retry by resending $CertRequest$.

For efficiency reasons nodes can use symmetric key schemes to secure all subsequent messages. This can easily be achieved by using the available authenticated public keys to establish a session key between peers.

5.4.2 Certificate revocation

SelfOrgPKM makes use of a self-revocation system based on a self-organized subordinate key renewal procedure. As mentioned in Section-5.3, nodes do not use their base key pair for any real communication, but must derive a subordinate key pair (x'_i, y'_i) from the base key pair which is then used for actual communication. This significantly reduces the chance of a successful attack on a node's base key pair. The self-organized key renewal process given in Section-4 can be used by the node to obtain a renewed key pair (x''_i, y''_i) at any point in time during the post initialization operation of the network. The node will thus derive a new private key $x''_i = x_i + H(KI'_i)k'_i \ mod \ q$ and generate a new self-certificate $SelfCert''_i = [KI'_i \parallel y''_i \parallel \alpha''_i \parallel \beta''_i]$. The renewed certificate $SelfCert''_i$ can be sent to the node's frequent contacts or offered to other nodes on communication initialization. Since nodes are responsible for their own keying material they can renew their subordinate key pair as frequently as desired. Nodes will however most likely renew their key pair in two instances: when they suspect that their subordinate private key as been compromised or when their set validity periods $ValPeriod'$ have expired.

6. DISCUSSION ON THE SECURITY AND FEATURES OF SELFORGPKM

The proposed peer-to-peer key management scheme for MANETs, presented in Section-5, makes use of subordinate public keys and crypto-based identifiers as building blocks to effectively eliminate the need for any form of off-line or on-line TTP. The availability of an off-line TTP is fundamentally against the characteristics of self-organized MANETs [4] [5]. This makes schemes such as [20] [10] [19] unsuitable for self-organized MANETs. The weaknesses of these existing schemes extend into network formation. They use a distributed certificate authority (DCA) as an on-line TTP which can be attacked. Our proposed scheme avoids these weaknesses by using a fully distributed system where each node becomes its own authority domain.

The existing schemes that take the characteristics of self-organized MANETs into consideration have the following main weaknesses:

1) The PGP approach presented in [4] only provides weak certificate authentication and may fail to provide certificate chains between all node pairs in the network.

2) The major weakness of the crypto-based identifier approach [14] [12] [3] is that users cannot revocate their public keys without changing their network addresses and/or identifiers [5]. In [3], the routing protocol control packets are extended by appending a users public key to route requests. With the high number of route requests sent by the on-demand routing protocols this leads to significant additional overhead.

3) The peer-to-peer key management scheme in [5] has a significant time delay in the setting up of the security associations. The self-organized scheme relies on node mobility to ensure that sufficient nodes come into physical contact. Asking of users to actively set up security associations with every node they meet may be a strong requirement.

In general all three of these approaches have strong points which are adequately highlighted by the authors. Some of these concepts have been incorporated in our proposal and clearly referenced when used.

The proposed scheme does not suffer from any of the above weaknesses: our proposal inherits the main benefits of crypto-based identifiers [14] [12] [3] as a means of solving the address ownership problem. Subordinate public keys introduced in Section-4 are used for real communication, leaving an adversary with a brute-force attack as the only option to compromise a node's base public/private key pair and/or identifier. The subordinate key pairs can also easily be renewed without having to modify the base public key pair which keeps the nodes' identifiers constant. With our approach users do not experience any noticeable time delay in the set up of security associations and distribution of their own certificates. We exploit the routing control packets to trigger localized certificate exchanges without extending the routing packets. Certificates can be explicitly authenticated with low computational overhead.

The characteristics of "open" networks inherently makes them vulnerable to Sybil attacks, as defined in [6]. The effect of nodes with multiple identities in MANETs is an interesting future research topic. More importantly, in "open" networks, nodes should not be able to impersonate other network nodes, hence the key management protocol must ensure that a node's identity/address cannot be falsified and that the binding between the node's identity and keying material cannot be forged. The user certificate must therefore be explicitly authenticated. This will be our focus in the remainder of the security discussion.

6.1 On the security of SelfOrgPKM

Considering the proposed scheme's *system model* and *adversary model*, given in Section-5.1 and Section-5.2 respectively, it becomes clear that an adversary will attack the scheme mainly in the following ways:

1) SelfOrgPKM uses a one-way collision resistant hash function to uniquely bind the nodes' public keys and network identifiers. From Equation-6, an adversary can attempt to spoof the identity/address of user P_i, by generating a forged public key y_i' that hashes to the same ID_i. This is analogous to finding a collision on a strong hash function such that $ID_i = H(y_i) = H(y_i')$.

2) The proposed key management scheme uses the subordinate public key generation scheme given in Section-4 with a twofold objective. Firstly, the scheme allows users to derive a secondary public/private key pair used for real communication thereby forcing the adversary to directly derive the private key from the public key, i.e. find $log_g\ y_i$. Secondly, it provides the nodes with an efficient computational and communication method of frequently renewing their keying material. Thus, from Equation-12, an adversary can attempt to obtain a forged subordinate public key y_i'' that satisfies $y_i'' = g^{x_i''} = y_i(r_i')^{H(KI_i')}\ mod\ p$.

3) Users generate self-certificates to bind their key pairs to their keying information KI_i as defined in Section-4. An adversary may attempt to forge a self-certificate by binding a forged subordinate public key y_i'' to the users valid keying information KI_i.

4) The only messages an adversary can intercept and alter are the certificates exchanged between nodes. While SelfOrgPKM prevents adversaries from forging certificates, mitigating all attacks on the network and lower layers that may thwart reliable communication, is not within our scope. SelfOrgPKM does however provide the network layer with cryptographic keying material without relying on a route establishment mechanism, thereby breaking the routing-security interdependency cycle [3].

The goal of the discussion on the security of SelfOrgPKM is to show that the proposed scheme is secure within our informal *adversary model*, i.e. it avoids the main angles of attack pointed out in 1) to 4) above. The security of the self-certificate generation and subordinate public key generation procedures are largely based on the security of the ElGamal type signature scheme presented in Section-3. We thus divide the security discussion into three parts:

- Firstly we show that the modified generalized ElGamal type signature variant, presented in Section-3, is secure within the combined security model, the Random Oracle and Generic Model (ROM+GM), proposed by Schnorr *et al.* [16] [17]. Although this model is mainly of theoretical interest, the proof provides some form of guarantee that the signature scheme cannot be broken based on widely accepted cryptographic assumptions. The two main assumptions in ROM+GM are the existence of an ideal hash function and ideal group G of prime order q. There exist hash functions and groups that have shown to be practical (but limited) instances of such ideal oracles.

- Secondly it is shown that verifying a user's self-certificate via verification of the ElGamal type signature on the self-certificate content $KI \parallel y'$ and validating the subordinate public key with Equation-4, yields the user's subordinate and base public key explicitly authentic.

- Lastly we investigate the probability of an adversary finding a collision on a strong hash function and thus spoofing network addresses or identifiers.

6.1.1 Security proof for the presented ElGamal signature scheme

In the following proof we refer to the combined security model, the Random Oracle and Generic Model (ROM+GM),

proposed by Schnorr *et al.* [16] [17]. Note that the formal adversary model as defined by Schnorr *et al.* is applicable.

In the first part of the security proof for the proposed peer-to-peer key management scheme, SelfOrgPKM, it will be shown that the modified generalized ElGamal type signature variant presented in Section-3, is secure against the *one-more signature forgery* attack [16] in the ROM+GM security model.

THEOREM 1. *Let a generic adversary $\mathcal{A}$ interact with a signer and be given g, the public key y and an oracle for H. $\mathcal{A}$ performs t generic steps which include l sequential signer interactions. With a probability space consisting of y, H and coin flips of the signer, it is not possible for $\mathcal{A}$ to produce $l+1$ signatures with a probability better than $\frac{\binom{t}{2}}{q}$.*

In the following proof *Lemma 1* and *Lemma 2* are those defined and proved in [16].

PROOF. [following Schnorr *et al.* [16]]

As given by *Lemma A* defined below, the group element $f_{i'} = g^{\frac{s'_i}{c'_i}} g^{-\frac{x}{c'_i}} = g^{\langle \alpha_{\mathbf{i}'}, (1,x,\mathbf{k})\rangle}$ for an arbitrary $i \leq t'$. $\mathcal{A}$ receives hash query $c'_i = H(m \parallel g^{\frac{s'_i}{c'_i}} g^{-\frac{x}{c'_i}})$ and needs to find s'_i which satisfies Equation-9. The adversary $\mathcal{A}$ is thus required to solve a linear polynomial $x + c'_i\langle \alpha_{\mathbf{i}'}, (1,x,\mathbf{k})\rangle$ at $(x,\mathbf{k})$. By *Lemma 2* presented by [16], x is statistically independent from $(\alpha_{\mathbf{i}'}, (1,x,\mathbf{k}))$, excluding prior collisions $f_j = f_k$. By *Lemma 1* presented in [16], it is known that such collisions will only occur with an upper bound probability of $\frac{\binom{t'}{2}}{q}$. On the other hand, by *Lemma A*, adversary $\mathcal{A}$ must choose $c_1, \ldots, c_l$ for each signature (m'_i, c'_i, s'_i) that satisfies Equation-7 such that x cancels out. In the case of a sequential attack, without any collisions among the computed group elements $f_1, \ldots, f_{t'}$, the system of $l+1$ equations for $c_1, \ldots, c_l$ is solvable with an upper bound probability of $\frac{\binom{t''}{2}}{q}$, where t'' denotes the number of queries to H [16]. It follows from $\frac{\binom{t'}{2}}{q} + \frac{\binom{t''}{2}}{q} \leq \frac{\binom{t}{2}}{q}$, that $\frac{\binom{t}{2}}{q}$ is the highest probability for $\mathcal{A}$ to succeed in a sequential, *one-more signature* attack on the signature scheme presented in Section-3. □

LEMMA A. *Let the triplet (m'_i, c'_i, s'_i) be a signature with a probability better than $\frac{1}{q}$. The c'_i-coordinate then coincides with the value $H(m \parallel f)$ corresponding to the hash query $(m \parallel f)$. From Equation-2, $g^k = g^{\frac{s}{c}} g^{-\frac{x}{c}}$. The hash query $(m \parallel f) \in G \times M$, satisfies $c'_i = H(m \parallel f) = H(m \parallel g^{\frac{s'_i}{c'_i}} g^{-\frac{x}{c'_i}})$, where the group element $f = f'_i$ for some arbitrary $1 \leq i' \leq t'$. The parameters (m'_i, c'_i, s'_i) also satisfy:*

$$c'_i = \frac{1}{-\alpha_{i',1} + \sum_{k=1}^{l} \left[\alpha_{i',k}\, c_k^{-1}\right]} \tag{7}$$

$$s'_i = c'_i \left[\alpha_{i',0} + \sum_{k=1}^{l} \alpha_{i',k} \frac{s_k}{c_k}\right] \tag{8}$$

In the following proof, *Lemma 2* is as defined and proved in [16].

PROOF. [following Schnorr *et al.* [16]]

Since $1 \leq i' \leq t'$ denotes the index of f among the computed group elements $f_1, \ldots, f'_t$, the group element can be written as $f_{i'} = g^{\frac{s'_i}{c'_i}} g^{-\frac{x}{c'_i}} = g^{\langle \alpha_{\mathbf{i}'}, (1,x,\mathbf{k})\rangle}$. It follows from the previous equation and $k_k = \frac{s_k}{c_k} - \frac{x}{c_k}$, that:

$$s'_i = x + c'_i \log_g \left[g^{\frac{s'_i}{c'_i}} g^{-\frac{x}{c'_i}}\right] = x + c'_i\langle \alpha_{\mathbf{i}'}, (1,x,\mathbf{k})\rangle \tag{9}$$

$$\begin{aligned} s'_i &= c'_i \left[\alpha_{i',0} + \sum_{k=1}^{l} \alpha_{i',k} \frac{s_k}{c_k}\right] + \\ &\quad x \left[1 + c'_i \left[\alpha_{i',1} - \sum_{k=1}^{l} \alpha_{i',k} \frac{1}{c_k}\right]\right] \end{aligned} \tag{10}$$

In order for the generic adversary $\mathcal{A}$ to calculate the correct s'_i, $\mathcal{A}$ must find c'_i such that x cancels out. $\mathcal{A}$ must therefore select $c_1, \ldots, c_l$ that satisfies Equation-7.

If x cancels out, s'_i can be computed by $\mathcal{A}$ as specified by Equation-8.

In the case that x does not cancel out in the equality given by Equation-10, the equality will only hold with probability $\frac{1}{q}$ since x is statistically independent from non-group data by *Lemma 2* presented in [16]. □

6.1.2 On the security of the proposed subordinate public key generation scheme

We return now to a more informal approach. From any entity's perspective Equation-4 can only provide *implicit* authentication of subordinate public key y'_i, i.e. the verification procedure gives no assurance that P_i knows the corresponding private key x'_i. The authenticity of the subordinate public key only becomes explicit when P_i uses it for a cryptographic procedure which inherently provides a proof of knowledge of x'_i.

An adversary $\mathcal{A}$ that wants to produce a forged subordinate public key must compute a public key $y'_{\mathcal{A}}$ that satisfies:

$$y'_{\mathcal{A}} = y_i \cdot (r_{\mathcal{A}})^{H(KI_{\mathcal{A}})} \bmod p \tag{11}$$

$\mathcal{A}$ does not know $\log_g y'_{\mathcal{A}}$ and will consequently fail to produce a valid signature that satisfies Equation-2. This serves as motivation for introducing self-certificate generation in Section-5.3, which allows the subordinate public keys to be explicitly authenticated. It will thus be appropriate to assess the security of the proposed subordinate public key generation protocol in conjunction with the signature (α'_i, β'_i) on $m_i = [KI_i \parallel y'_i]$ as described in Section-5.3. It is noted that (α'_i, β'_i) is produced via the proposed signature scheme presented in Section-3. The verification equation on (α'_i, β'_i) is given as:

$$g^{\alpha'_i} = y'_i \cdot (\beta'_i)^{H(m_i \| \beta'_i)} \bmod p \tag{12}$$

Substituting Equation-4 into Equation-12 yields:

$$g^{\alpha'_i} = y_i \cdot (r_i)^{H(KI_i)} \cdot (\beta'_i)^{H(m_i \| \beta'_i)} \bmod p, \tag{13}$$

which has the following signature equation:

$$\begin{aligned} \alpha'_i &= x_i + H(KI_i)(k_i) \\ &\quad + H(m_i \parallel \beta'_i)(\log_g \beta'_i) \bmod q \end{aligned} \tag{14}$$

An entity which explicitly authenticates y'_A via Equation-4 and Equation-12, indirectly verifies Equation-14 in two steps. From Equation-13 and Equation-14 it is concluded that an adversary $\mathcal{A}$ can only generate a forged subordinate

public key with an upper bound probability of $\frac{\binom{t}{2}}{q}$ in the ROM+GM security model (by *Theorem 1*). Furthermore it shows that the verifier of Equation-4 and Equation-12 can be assured that the party with ID_i, generated via Equation-6, knows the base private key x_i corresponding to y_i.

Another point of concern is that a compromise of the subordinate private keys may reveal information about the base or primary private key. From Equation-1 and Equation-3 it can be seen that the base private key x_i is blinded from the subordinate private key x'_i by the addition of a random number k_i. This is the same mechanism that is used by all ElGamal type signatures to protect private keys from being derived from valid signatures. An adversary $\mathcal{A}$ that compromises a subordinate key pair (x'_i, y'_i) therefore has the same probability of gaining knowledge of the base private key x_i as someone with a valid signature (s_i, r_i), generated via the signature scheme presented in Section-3.

6.1.3 On the security of hash based identifiers

The security of crypto-based identifiers has been extensively reviewed in [12] [11] [14] [3]. To avoid repetition we will only briefly consider the security of hash functions. As noted from the specification of the proposed key management scheme we do not bind SelfOrgPKM to any specific hash function. Such a secure hash function can be carefully chosen on deployment of the protocol. For example, currently SHA-1 [13] with a 160 bit output will provide adequate security since 2^{80} hash operations are needed to find a collision on SHA-1 using a brute-force attack. The most recent known attack on SHA-1, presented in [18], shows that a collision on SHA-1 can be found with a complexity of less than 2^{69} hash operations. We note however that an attacker also has the additional computational overhead of one full exponentiation in order to find a valid public key before computing the hash function. Clearly the additional time complexity of the exponentiation makes the spoofing of network addresses impractical.

6.2 On the efficiency of SelfOrgPKM

SelfOrgPKM is fully distributed, preserving the symmetric relationship between nodes as required in MANETs. The proposed peer-to-peer key management scheme thus places the same communication and computational burden on each node which in our view is the first step towards mitigating selfishness attacks and extending the nodes' battery life. In the next two subsections the performance of SelfOrgPKM will be investigated in an ideal network setting, hence assuming guaranteed connectivity. In Section-6.3, we evaluate the performance of SelfOrgPKM in a simulation study where factors such as connectivity and route failures (due to the error-prone wireless channel, node mobility etc.) have an impact on the system operation.

6.2.1 Efficiency of SelfOrgPKM initialization phase

The initialization phase is performed by each node before joining the network and therefore has no impact on network performance. This process should however still be as efficient as possible. Each node P_i performs 4 exponentiations (exp), 3 random number generations (R_{gen}) and 3 hash computations ($H(\cdot)$) (The 2 multiplications and 2 summations have insignificant impact on the time complexity in comparison with the exponentiations). The initialization phase has no communication cost.

6.2.2 Efficiency of SelfOrgPKM post-initialization phase

The on-line post-initialization phase of SelfOrgPKM results in little overhead for each node. A node renewing its self-certificate has to perform only 2 signature generations and 1 exponentiation to compute its renewed subordinate public key with a total cost of 3 exp, 2 R_{gen} and 2 $H(\cdot)$. Any node can verify another nodes' self-certificate with a computational cost of 3 exp and 1 $H(\cdot)$ and only 3 exp for all subsequent verifications since the base public key has to be authenticated only once.

Self-certificate exchanges on a peer-to-peer basis (on the application and network layers) are the only communication overhead imposed on the network by the proposed scheme. A certificate exchange procedure on the application and network layer only takes 2 asynchronous rounds with 1 unicast message from each node.

6.3 Performance Evaluation of SelfOrgPKM

The effectiveness of the proposed public key management scheme was investigated through simulations in the ns-2 simulator (release 2.28) [2], using the OpenSSL cryptographic library (version 0.9.7e) [1] to implement the basic modular arithmetic. The implementation supported the mathematical correctness of the cryptographic design and confirmed the low implementation complexity of the proposed scheme.

6.3.1 Simulation model

In our simulation we used the ns-2 implementation of the IEEE 802.11b physical layer and medium access control protocols. The radio model was modified to have a nominal bit-rate of 11Mb/sec while supporting a 250m transmission range.

The main objective of our preliminary simulation study was to isolate the effect of network layer certificate exchanges on the network performance under different traffic and mobility scenarios. We used the ns-2 constant bit-rate (CBR) traffic generator to simulate the connection patterns and random way point model to simulate node mobility. For all simulations the CBR packet size was set to 512 bytes, with a total of 50 networking nodes. The size of certificates was set to 460 bytes. Maximum node mobility was set to 5m/sec and 20m/sec, while the loading of the network was varied between 1 CBR packet/sec and 8 CBR packets/sec. With a total of 50 connections, each node was set to have one CBR traffic source with a single unique destination. In the first 90 sec of the total 1000 sec of simulation time all traffic sources are randomly started in order to force as many nodes as possible to participate in the network layer certificate exchange procedure. The network area for all simulations was set to 1000m x 1000m.

Section-5.4.1, explained the certificate exchange mechanism of SelfOrgPKM on the network layer. In our discussion we specified that the certificate exchange procedure of the proposed scheme is not bound to a specific routing protocol. The details of the routing level certificate exchange implementation is dependent on the route discovery mechanism of a suitable MANET routing protocol. In our simulations we chose as a network layer model the Ad Hoc On-demand Distance Vector (AODV) routing protocol [15], as included in the ns-2 distribution.

SelfOrgPKM's network layer certificate exchange procedure was easily integrated into the AODV protocol's route

discovery mechanism. AODV allows for route discovery by forwarding a route request (RREQ) from the source node to the destination node. Each intermediate node that receives the RREQ caches the broadcast ID of the RREQ and captures the reverse route in its routing table. An intermediate node with a fresh enough route will reply to the source node with a route reply message (RREP) or forward the RREQ to its neighbors. If a node receives a RREQ and has already cached the RREQ broadcast ID or was the originator of the RREQ, it will disregard the received RREQ.

AODV's RREQ receive function was modified as follows: assume intermediate $Node_C$ broadcasts a RREQ which is received by its neighbor, $Node_D$. As usual $Node_D$ checks the broadcast ID and RREQ source address for consistency. $Node_D$ then consults its repository of received certificates to determine whether it has not already received the certificate from $Node_C$. If the query is positive, $Node_D$ has the certificate of $Node_C$ and will continue to process the RREQ as usual. If $Node_D$ has not received the RREQ before *and* did not originate the RREQ *and* does not have the certificate of the $Node_C$, we define two scenarios:

- In the first scenario $Node_D$ takes an optimistic approach: when $Node_D$ receives a RREQ from $Node_C$ it sends its own certificate to $Node_C$ and process the RREQ as normal. If $Node_C$ receives the certificate it will reply to $Node_D$ with its own certificate.

- In the second scenario $Node_D$ takes a more conservative approach: when $Node_D$ receives a RREQ from $Node_C$ and finds that it does not have $Node_C$'s certificate, $Node_D$ will send its own certificate to $Node_C$ and disregard the AODV RREQ. If $Node_C$ receives the certificate it will reply to $Node_D$ with its own certificate. AODV is unaware that the RREQ has been received and dropped by $Node_D$ and will respond to the scenario as if $Node_D$ is unavailable.

We stress again that this modification of AODV is not to make the protocol secure as a whole. Of primary interest is the impact of localized peer-to-peer certificate exchange on the network performance when integrated into a practical MANET routing protocol. The two scenarios explained above are partly applicable to previous efforts to secure AODV [7]. If the first scenario is used to distribute keying material for example to SAODV, then user P_i's base public key y_i, subordinate public key y_i' and public signature parameter r_i, will have to be appended to RREQs in order for the next hop to verify the DSA or ElGamal type signatures on the RREQ messages [7]. This will result in additional routing overhead in addition to the single or double RREQ signature extensions. In the second scenario the overhead will be eliminated at the cost of losing possible routes due to dropping the RREQ messages.

6.3.2 Simulation Results

In Figure-2, it is shown that the localized peer-to-peer network layer certificate exchange mechanism has minimal impact on network performance. The simulation results for scenarios 1 and 2 (Section-6.3.1), correspond closely with the CBR reference simulation at low mobility (maximum 5m/sec). The initial disregard of the RREQ does not significantly affect the AODV performance. This is explained by the fact that the initial received RREQ triggers a flurry of

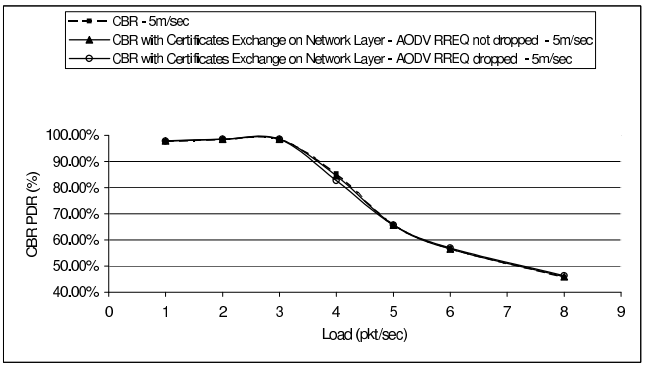

Figure 2: CBR packet delivery ratio % vs load in pkt/sec for 5m/sec mobility

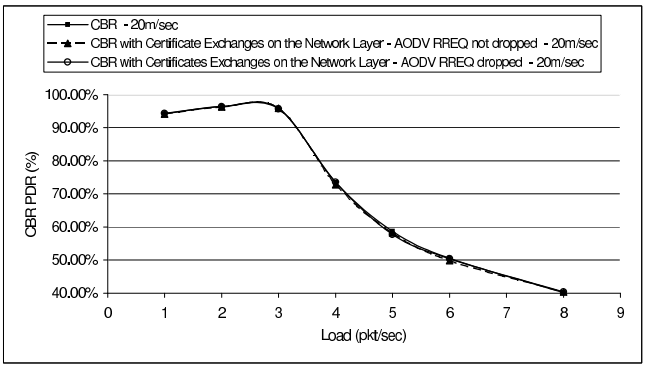

Figure 3: CBR packet delivery ratio % vs load in pkt/sec for 20m/sec mobility

certificate exchanges which allows subsequent RREQ messages to be handled as usual. This result also demonstrates an important concept: it is possible to trigger network level certificate exchanges without extending the routing protocol's control packets.

As anticipated an increase in mobility (from maximum 5m/sec to 20m/sec) degrades the overall performance of the network. Figure-3 shows that both the simulations for scenarios 1 and 2 follow the CBR reference simulation. In fact with high mobility, the correlation between all simulations becomes even closer. This supports the findings of Capkun *et al* [5] that mobility can be an advantage in securing MANETs.

Figure-4 illustrates the effect of node mobility on network performance. As the performance of our proposal degrades with the reference CBR simulation we conclude that node mobility does not have a significant impact on localized network layer unicast message exchanges.

To place the results of PDR vs. load into perspective we include the average end-to-end delay for CBR packets. Figure-5 shows that the exchange of certificates on the network layer does not add significant delay to the average delivery time of CBR packets.

7. CONCLUSION

The paper proposes a novel peer-to-peer key management scheme for fully self-organized mobile ad hoc networks, called Self-Organized Peer-to-Peer Key Management (SelfOrgPKM).

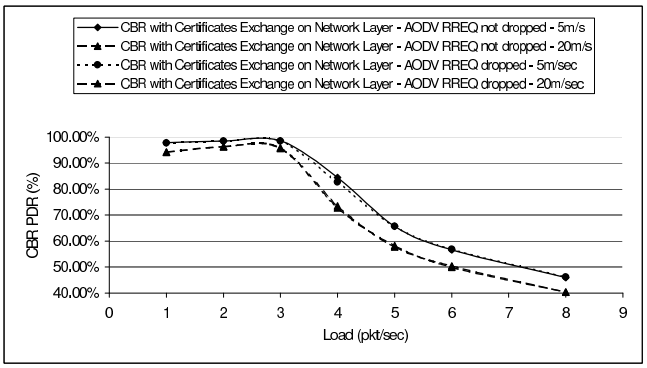

Figure 4: Effect of mobility on CBR packet delivery ratio

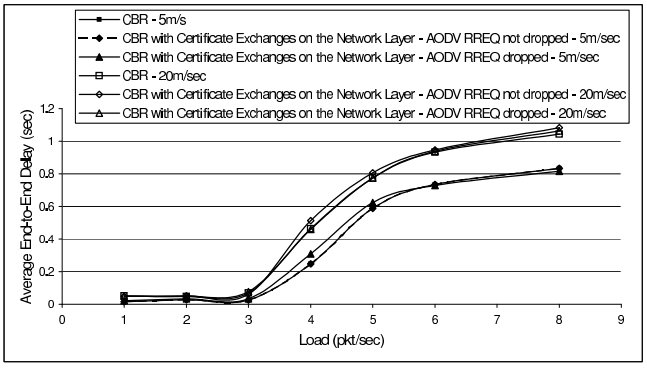

Figure 5: Average CBR packet end-to-end delay vs load in pkt/sec

The scheme has low implementation complexity and provides self-organized mechanisms for certificate dissemination and revocation without the need for any form of off-line or on-line authority.

The fully distributed scheme is superior in communication and computational overhead with respect to its counterparts [20] [10] [19] [4]. All nodes send and receive the same number of messages and complete the same amount of computation. SelfOrgPKM therefore preserves the symmetric relationship between the nodes. Each node is its own authority domain which provides an adversary with no convenient point of attack.

SelfOrgPKM solves the classical routing-security interdependency problem and mitigates impersonation attacks by providing a strong one-to-one binding between a user's certificate information and public key.

The paper also introduces two generic cryptographic building blocks as the basis of SelfOrgPKM: 1) A variant on the ElGamal type signature scheme developed from the generalized ElGamal signature scheme introduced by Horster *et al.* The modified scheme is one of the most efficient ElGamal variants, outperforming most of the other variants; and 2) A subordinate key generation scheme.

The paper introduces the novel notion of *subordinate public keys*, which allow the users of SelfOrgPKM to perform self-organized, self-certificate revocation without changing their network identifiers/addresses. Subordinate public keys therefore eliminate the main weakness of previous efforts to solve the address ownership problem in Mobile IPv6 [14] [12]. Furthermore, the main weakness of previous efforts to break the routing-security interdependence cycle in MANETs [3] is also eliminated by using a subordinate public key mechanism. The presented ElGamal variant was proved to be secure in ROM+GM (*Theorem 1*) without making any unrealistic assumptions. It was shown how the strong security of the signature scheme supports the security of the proposed subordinate key generation scheme.

The only operation of SelfOrgPKM affecting the network is the pairwise exchange of certificates. The cryptographic correctness, low implementation complexity and effectiveness of SelfOrgPKM was verified though simulation using ns-2 and OpenSSL. The simulation results show that our novel, localized certificate exchange mechanism on the network layer has negligible impact on network performance. The simulation results furthermore demonstrate that network layer certificate exchanges can be triggered without extending routing protocol control packets.

8. ACKNOWLEDGMENTS

This work was supported by ARMSCOR, the Armaments Corporation of South Africa. The authors would also like to thank:

- The anonymous reviewers for their constructive comments and suggestions.
- The authors of ns-2 and OpenSSL for making their open source software freely available.

9. REFERENCES

[1] OpenSSL cryptography library. Available at www.openssl.org.

[2] The Network Simulator - ns-2. Available at www.isi.edu/nsnam/ns.

[3] R. B. Bobba, L. Eschenauer, V. D. Gligor, and W. Arbaugh. Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks. In *proc. IEEE Global Telecommunications Conference*, December 2003.

[4] S. Capkun, L. Buttyan, and J.-P. Hubaux. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, 2003.

[5] S. Capkun, J. Hubaux, and L. Buttyan. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 2004. to appear.

[6] J. R. Douceur. The Sybil Attack. In *proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, March, 7-8 2002.

[7] M. Guerrero. Secure Ad Hoc On-demand Distance Vector (SAODV) Routing, March, 17 2005. INTERNET-DRAFT draft-guerrero-manet-saodv-03.txt.

[8] P. Horster, M. Michels, and H. Petersen. Generalized ElGamal signatures for one message block. In *proc. 2nd Int. Workshop on IT-Security*, September, 22-23 1994.

[9] P. Horster, M. Michels, and H. Petersen. Meta-Message Recovery and Meta-Blind Signature Schemes Based on the Discrete Logarithm Problem

and their Applications. In *proc. Advances in Cryptology - ASIACRYPT'94*, 1994.

[10] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-securing Ad Hoc Wireless Networks. In *proc. Seventh International Symposium on Computers and Communications (ISCC'02)*, July 1-4 2002.

[11] G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses. In *proc. Network and Distributed System Security Symposium (NDSS'02)*, February 2002.

[12] G. Montenegro and C. Castelluccia. Crypto-based Identifiers (CBIDs): Concepts and Applications. *ACM Transactions on Information and System Security*, 7(1):97–127, 2004.

[13] National Institute of Standards and Technology (NIST). Secure Hash Standard. Federal Information Processing Standards Publication FIPS PUB 180-1, U.S. Department of Commence, April,17 1995.

[14] G. O'Shea and M. Roe. Child-proof authentication for MIPv6 (CAM). *ACM SIGCOMM Computer Communication Review*, 31(2):4–8, 2001.

[15] C. E. Perkins and E. M. Royer. Ad-hoc On-demand Distance Vector Routing. In *proc. The Second IEEE Workshop on Mobile Computing Systems and Applications (IEEE WMCSA'99)*, February 1999.

[16] C. P. Schnorr and M. Jakobsson. Security of Discrete Log Cryptosystems in the Random Oracle and Generic Model. In *proc. The Mathematics of Public-Key Cryptography*, June, 12- 17 1999.

[17] C. P. Schnorr and M. Jakobsson. Security of Signed ElGamal Encryption. In *proc. Advances in Cryptology - ASIACRYPT '00*, December,3-7 2000.

[18] X. Wang, Y. L. Yin, and H. Yu. Finding Collisions in the Full SHA-1. In *proc. 25th Annual International Cryptology Conference (Crypto'05)*, August, 14-18 2005.

[19] S. Yi and R. Kravets. MOCA: Mobile certificate authority for wireless ad hoc networks. In *proc. of the 2nd Annual PKI Research Workshop (PKI 2003)*, April, 28-29 2003.

[20] L. Zhou and Z. J. Haas. Securing Ad Hoc Networks. *IEEE Network: special issue on network security*, 13(6):24–30, 1999.

[21] P. Zimmermann. *The Official PGP User's Guide*. MIT Press, 1995.