SOMA: Self-Organised Mesh Authentication

Foivos F. Demertzis, Christos Xenakis

University of Piraeus, Department of Digital Systems,
80 Karaoli Dimitriou Street, PC 18534, Piraeus, Greece
{fdemertz,xenakis}@unipi.gr
http://www.ds.unipi.gr

Abstract. Community mesh networks have emerged rapidly in every metropolis around the world, however many of the security methods applied are counter-intuitive and usually disrupt the autonomous characteristics of the mesh nodes. In SOMA we present a structured Peer-to-Peer solution providing authentication service based on a scalable, self-organized and fully distributed Web-of-Trust. Our proposal is a hybrid Public Key Infrastructure build on top of Chord, allowing each agent to place its own trust policy while keeping the autonomous characteristics the nodes intact. Our goal is to create a large-scale authentication system for mesh networks without the need of a Trusted Third Party. We leave the decision of whom to trust in each agent independently taking advantage of the overlay to alleviate the shortcomings of traditional Web-of-Trust models. This is achieved by using the overlay as a meta-structure to infer trust relationships. The possible attacks and limitations of our proposal are also investigated and discussed.

Keywords: Mesh Networks, Public Key Infrastructure, Web-of-Trust, Peer-to-Peer

1 Introduction

Community mesh networks have emerged in every metropolis around the world, where the comprising nodes share resources and services in an autonomous fashion. These mesh networks are based on independent, computational powerful, wireless ad-hoc, multi-hop nodes. Identity management is the cornerstone for the establishment of trust in any network, and as such mesh networks are of no exception. In such networks, authentication credentials are usually exchanged in a bi-directional and symmetric fashion between two or more parties. Nodes that wish to communicate securely, first need to establish some form of trust, which is based on the fact that initially the two parties are distrusted. Therefore, to establish a secure channel of communication and authenticate the exchanged data, a node first and foremost should be able to prove the authenticity of the identification credentials of the party in question.

These community mesh networks comprise of a decentralised infrastructure that can scale to many thousands. In this work, we focus on fully or semi-planned, long-term wireless ad-hoc networks similar to the many community

Wireless Metropolitan Networks (WMNs). The nodes use wireless multi-hop transmissions to communicate, based on IEEE 802.11 and are computationally strong enough, such as the ones found at the edges of community WMNs (e.g., Athens WMN [1], MIT's Roofnet [2]).

1.1 Motivation

Currently, many security models have been proposed for mesh networks, but most of these are counter-intuitive and usually disrupt the autonomous characteristics of the mesh nodes. The use of a centralised Public Key Infrastructure (PKI) approach is fundamentally incompatible with the decentralised open nature of the mesh network's topology. Many traditional PKI solutions delegate trust to a Trusted Third Party (TTP). Single Certification Authority (CA) and replicated CAs are not viable for community mesh networks. The former is not a scalable architecture and neither is fault tolerant. The latter, introduces increased security risk, by providing additional attack vectors. This is due to the fact that an adversary can gain access to the CA's private key by compromising even a single of the replicated CAs.

A plethora of the proposals use empowered nodes either involving the organisational aspects of the network or its trust management. Implementing such a solution for mesh networks that scale to many thousands would inevitably lead us to inefficiency, bottlenecks and points of failure. By delegating trust decisions to an external third party, distributed or otherwise, we would inevitably lose the individuality and autonomy of the mesh nodes. Hence, we propose SOMA a system that has self-organization, autonomy and scalability as its principal design co-ordinates.

1.2 Our Contribution

SOMA is a certificate-based authentication infrastructure that aims to create a large-scale secure authentication system for mesh networks without the need of a TTP. We aim at creating a self-organised, efficient and scalable authentication infrastructure, without sacrificing the autonomous characteristics of the nodes. Hence, our work focuses on building on top of a self-organised, structured Peer-to-Peer, Web-of-Trust [3] infrastructure. The proposed system leaves the decision of whom to interact with and why to place trust on each of the agents, independently. It provides a policy-based system, which can also be the basis for a more complex reputation based model. Such a model can include a variety of relationships and metrics, which is out of the scope of this paper.

In SOMA, we make use of structured Peer-to-Peer in contrast to unstructured flooding based protocols, e.g., Gnutella [4], due to the fact that the nodes are mostly static. Therefore, using a structured Peer-to-Peer approach, such as Chord [5], a node is given access to a scalable holistic view of the mesh. In addition, building an efficient Web-of-Trust infrastructure, which exploits collective knowledge in a self-organised way, fits better to the studied autonomous network architecture. Using Chord, we gain mathematically provable scalability and great thoroughly investigated static resilience [5], which is a mandatory requirement for an identity management system. SOMA is based on a PGP-like [3] architecture, where the nodes create the public and private keys themselves. Issuing and managing of the key material is done locally and digital certificates are issued using only the collective information of the overlay routing architecture. Through certificate exchanges each node builds each keyring, storing it locally, in accordance to PGP Web-of-Trust. In contrast to hierarchical PGP, we do not use neither a central nor a distributed CA, and we avoid completely delegation of trust to a TTP. Hence, each of the agents uses its keyring independently, placing their trust depending on the identification credentials gathered. A node, after assessing the certificates on its keyring and evaluating the identity of the communicating parties, will use these credentials to establish a secure communication channel.

The basic operations and goals of our solution stem from the fact that we wish to have no empowered nodes. From a security design point of view, our goals are similar to the ones found in most public-key based schemes, authentication, data integrity, confidentiality and non-repudiation. Most community WMNs are open and each peer individually decides what services to offer in the mesh. Therefore, our authentication mechanism has to impose no barriers to entry and, in addition, keep the autonomy of the nodes intact. We take advantage of the consistent hashing, employed by many structured Peer-to-Peer overlays and we use the overlay look-up protocol to infer the trust relationships between the autonomous peers.

In the next chapter we will examine some of the related work and background that has been put into distributing PKIs. Section 3 presents our authentication system called SOMA. Section 4 presents an evaluation including discussion of the security issues and closely examines possible attack scenarios. Finally, section 5 summarizes important points and outlines the conclusions drawn.

2 Related Work

The notion of a distributed Public Key Infrastructure (PKI) to provide authentication service for ad-hoc networks has been coined around for many years. A lot of interest and research has been put into creating efficient scalable infrastructures, that do not sacrifice the autonomous characteristics of the comprising nodes. Research has mainly been put into mobile ad-hoc networks (MANETs) where mobility is a limiting factor, and in wired systems the central PKI is distributed using threshold cryptographic schemes [6]. In this section, we summarize the most important literature that has been proposed for tackling public key management in ad-hoc networks. We also do a critical appraisal on issues that we will be investigating through SOMA.

In 1999 L. Zhou and Z. Haas [7] proposed the first Distributed Certification Authority (DCA) for ad-hoc networks, based on threshold cryptography. We refer to this work because it was the foundation for many DCA implementations that followed and hence, have similar limitations abated to an extent depending on the specifics of each approach. The architecture proposed by Zhou and Haas has an arbitrary number of server-nodes and combiner nodes that constitute the DCA. Without delving into the technical details, these server-nodes would divide between them the private key of the CA. The server-nodes would follow a (n, t+1) scheme, needing t+1 out of n shares to recover the secret parts of the divided private key, as per Shamir's secret sharing scheme [6]. All the nodes in the system would also have a public key/secret key pair that the DCA would be responsible to certify. Subsequently, the server-nodes involved would sign with their share of the private key a partial signature and submit these pieces to a combiner node that produces the final signature.

There are several limitations with this approach, most importantly, it does not scale since it is semi-distributed. Furthermore, it uses empowered nodes, which are not self-organized, but rather depend on an off-line authority to empower them. The nodes require an off-line assignment of Unique Universal Identifiers (UUIDs). The proposed system does not provide a solution in case of connectivity problems between the nodes since it was out of its scope. Large-scale changes in topology have a detrimental effect both on the system security and its performance. This is due to the fact that security comes with a trade-off to availability and hence, a method needs to be devised to re-organise the number of shares, required to construct the secret key.

Similar in concept to Zhou and Z. Haas is by Kong et al. [8], which alleviates the availability issues, by not using combiner nodes, but exposes the system to Sibyl attacks [9]. More work in similar grounds has been done in [10] an extension proposed by Zhou et al. for use on the Internet. Furthermore, Yi and Kravets [11][12][13] propose a DCA based on threshold cryptography. The main points of their proposal being that the nodes requiring certification contact t+1 nodes instead and make use an efficient β-unicast method for dissemination instead of flooding.

All the aforementioned proposals do not scale well and are not fully self-organized, hindering the autonomous characteristics of a mesh network. One different approach that uses certificate chaining similarly to the PGP Web-of-Trust was proposed by Hubaux et al. [14][15][16] for MANETs. They developed a self-organised solution for key management in ad-hoc networks, where each user has authority and issues public-key certificates to other users. Key authentication is done via the exchange of certificate graphs. Each user stores all the certificates from other users and evaluation of valid certificate chains is done through merging their individual certificate repositories. The certificate dissemination that takes place is done in ad-hoc flooding basis, taking into account the mobility of the users, which is integral for the convergence of the certificate graph. This introduces a delay in the initial stages of network and can cause problems when there are insufficient trust relationships. More work on certificate chaining schemes similar to Hubaux et al. was done in [17] were the authors propose a hierarchical Binary Tree based solution with certificate chaining for MANETs.

Other related work that focuses on different requirements can be found in identity based encryption schemes in [18], [19]. In addition, overlay DCAs have been proposed that use DHT (Distributed Hash Table) to provide the primitives needed to create a distributed storage of the key material and the underlying management infrastructure. In SOMA, on the other hand, we do not use storage of a DHT as a core component. A DHT solution can be found on [20], in which the authors propose a statistical quorum based mechanism to probabilistically get certificates of the overlay and decide on their authenticity. Moreover, on [21] they propose the use of threshold cryptography to distribute the CA functionality. They derive and delegate trust by distributing the certificate directory and the private key of the CA in many nodes, divided into segments of trust. In both these structured Peer-to-Peer solutions DHT provides a distributed storage for the key material and the underlying management infrastructure, distributing the directory and the private key of the CA. We, on the other hand, focus on the autonomy of the nodes, each acting independently using a Web-of-Trust.

3 SOMA

In this section we present our proposal and the rationale behind our design choices. Furthermore, we proceed in analysing the steps taken for the formation of a mesh network and the authentication protocol specifics that will be followed by the nodes.

3.1 Overview

We wish to build an authentication infrastructure in the presence of active adversaries, so that the nodes can verify the public keys and their authenticity via chains of trust. Our system is build on top of Stoica's Chord [5]. Chord is a structured overlay protocol, that provides a look up interface based on consistent hashing over a circular (modulo) identifier space. Using a key, Chord maps that key onto a node. Chord is normally used as the basis for providing some DHT functionality. On a DHT that is based on Chord, we can normally store (key,value) pairs as we would in every other locally stored hash table and retrieve the data back, based on our key. Consistent hashing uses a one-way function (e.g., SHA1, SHA2) to map both the keys and the node IDs (e.g., IP-addresses) uniformly distributed in the same identifier space. A key look-up in Chord returns the IP address for that key. It is scalable, taking O(log N) communication hops and keeps O(log N) state per node, where N is the total number of nodes in the system.

The benefits inherited from Chord can be summarised:

- A simple scalable lookup algorithm based on consistent hashing.
- A robust overlay that is robust on node joins and failures.
- A stabilisation protocol.

Our proposal is targeted on multi-hop, ad-hoc networks that display transitive trust relationships between the comprising nodes. This can be extrapolated further saying that as a community WMN, the nodes will build a PGP Web-of-Trust and exhibit characteristics, similar to a Small-World graphs [22]. Moreover, even-though the nodes themselves will not often be neighbours of one another, most nodes will be reachable from every other by a significantly smaller number of hops, compared to the network size. This is usually achieved by short-cuts between the nodes.

We use the overlay network as a meta-structure that efficiently creates the transitive relationships, so that the nodes can deduce the chains of trust between them in a self-organised and scalable manner. Hence, when the Chord protocol is used as the basis of a Web-of-Trust model and certificates are stored in local keyrings, we can safely reach the conjecture that a node would be able to find a chain of certificates in O(log N) time and in O(log N) number of certificates for any trust path given. Ideally, a peer requiring a secure authenticated service to another peer would be able to deduce a valid chain of trust, with the same complexity as would a simple look-up do in Chord. Therefore, SOMA builds a self-organised distributed authentication service, based on the aforementioned idea and tailors it to the needs of the mesh networks.

The SOMA architecture comprises the mesh nodes, forming a virtual ring overlay. Each node has a unique ID that encompasses its physical and logical address. A node wishing to join SOMA requires to know one node that is already in the ring (e.g., its bootstrapping node to the mesh network). Subsequently, each node is able to exchange certificates and establish trust relationships with the rest of the nodes in SOMA. The overlay structure forms the framework for the transitive trust relationships and each node exchanges certificates with the nodes that is directly responsible for routing. When a node requests a PGP chain of certificates to another node, following the overlay routing, will result to an efficient trust path discovery.

The nodes in the mesh form logical ring. The main components of a node in SOMA are listed below:

- A logical ID that is directly correlated to its physical ID.
- A PGP keyring.
- A finger table providing efficient lookups and captures the inter-node certificate relationships.
- A list of successors and a cache of IP addresses that provide resilience against attacks and failures.

In SOMA, we assume that the nodes will use TCP and are assigned static IP addresses. The network is assumed vulnerable by adversaries at all times. The TCP communication messages are reinforced with authenticated DiffieHellman session keys. The certificate exchange between two parties is done using 2-pass strong authentication. The PGP messages and certificate format details follow the OpenPGP Message Format [23] or any compliant format.

3.2 Initialization and Bootstrapping

When a new node n wishes to join the mesh network and use the authentication service, it must first create a public/secret key pair Pk_n/Sk_n . Furthermore, n will produce a self-signed certificate $Cert_n$, which is done locally and without any CA involvement. Then it bootstraps to the network, by contacting one of its direct one hop links in the mesh n'. These one hop links serve as trust-anchors and short-cuts in the overlay ring. They are considered initially trusted since usually in community WMNs for a node to join the network it has to contact another peer directly for bootstrapping, address resolution, antenna alignment etc. Therefore, we can safely assume that the node n can verify the identity of n', its introducer, as authentic at least to the extent of connecting the peer n to the mesh.

To extrapolate further on the above, the side channel (e.g., physical contact or other direct channels) between two peers, performs the role of an off-line CA. *n'* being the bootstrapping node of n will sign the authenticity of the key Pk_n and n will sign the authenticity of $Pk_{n'}$. The certificate will follow the PGP format details and *n'* will then put in its keyring the certificate $Cert_{n \Rightarrow n'}$. Additionally, n will hold in its keyring $Cert_{n' \Rightarrow n}$ as in the PGP Web-of-Trust. Hence, at the start we have direct one-hop relationships which are bi-directionally verified by n and its introducing nodes.

To join a SOMA ring a node needs to generate a new unique ID. The unique ID of node n is ID_n and will be derived by hashing the concatenation of node's n public key and its network address $ID_n = h(Pk_n||IP_{address\ of} n)$. This identifier ID_n will be the key that uniquely identifies node n in SOMA's circular identifier space. Node n will then use the produced ID_n to connect to the overlay. The authenticated introducer in SOMA is in fact what the initial contact link is to the Chord ring. *n'* will learn node's n successor by looking up ID_n and n will proceed with building its finger table, as described in the following section.

3.3 Node Join and Stabilisation

For n to join SOMA it will need to setup its finger table. A finger table is a list of pointers to IDs in the overlay. Instead of holding a single pointer to the next node, $n.finger = ID_{next\_node}$, a list of m nodes is maintained, *n.finger(m)*, with their logical inter-node distance increasing exponentially. This provides the efficient look up mechanism. Typically, on each of the entries, associations and additional information will be stored. For instance, each ID in the finger table will be associated with a corresponding certificate from the local keyring and a physical address. Moreover, the nodes will need to take past transactions into account, therefore, for each of the fingers n will record a trustworthiness metric and a cache of IP addresses encountered thus far. The finger table of a node can be viewed as a map holding the IDs that a node can use as certificate paths in SOMA. Therefore, the finger table will comprise the nodes that it has exchanged certificates with and that is responsible for their correct routing. In more details, the *n.finger(i)* is the i^{th} entry of the finger table for the node n and will hold the $n + 2^{i-1}$ ID and physical address that succeeds n on the circular identifier space, where $1 \leqslant i \leqslant 160$ (for SHA1 with arithmetic modulo 2^{160}).

Hence, when n joins the ring and since each node holds O(log N) state in each finger table, to holistically capture the new entry more nodes in SOMA need to propagate the change. Similarly to the original Chord look up, for a certificate path discovery to be successful, only the successor pointers of the finger table need to be correct. The finger table mechanism serves only in lookup efficiency, not its correctness. The finger table needs to be kept up to date with new entries so that certificate look ups scale logarithmically. When node n joins, the steps taken are:

1. Initialization of the predecessor and the finger table entries of n by establishing authenticated relations through a chain of trust.
2. Update the fingers and predecessors of the existing nodes to reflect the addition of n.

The first step for the joining node n will be to initialize its finger table and find its virtual position in the ring. Node n calls $n.join(ID_{n'})$, where node *n'* is its introducer. This will make *n'* to do a Chord look-up of ID_n in the ring and return back to n the key/address of its successor s. The successor s is the first finger in the finger-table of n, $n.finger[1] = s$. In addition *n'* will provide n with a PGP certificate $Cert_{n' \Rightarrow n}$ so that it can associate its credentials with s. Finally, node n will set $n.successor \rightarrow s$

The join mechanism of SOMA changes the periodic stabilisation protocol in Chord. The procedures *stabilize*(), $notify()$ and $fix\_fingers()$ run periodically to notify the overlay nodes for the new entry, accomplishing the steps 1 and 2. After finding its successor, n will proceed with *n.stabilize*() so that it can exchange its credentials with its successor and propagate the information on the new join. We can think of *stabilize*() as a protocol working on linked list, that uses certificates for authentication and updates the linked list when there is a new entry.

The stabilization runs periodically in the background by all nodes. The first step for *n.stabilize*() is for n to notify node s that $n.successor \rightarrow s$ using the certificate provided by *n'*. Node s will proceed with checking the validity of the supplied credentials and call $s.findCertChain(n')$ (more details in section 3.4). The validation mechanism includes checking the public key, the address, the signatures, the timestamps and if any of the certificates in the chain is revoked. If s finds that all are correct, it sets $s.predecessor \rightarrow n$ else will request for another valid PGP certificate. Moreover, when the original predecessor of node s, node y runs *stabilize*(), it will ask for *s.predecessor*, s will then return to y that n is its new successor. Node y in turn will $notify()$ n. Node n will reply back providing the certificate chain by *n'*. Node y will similarly check the supplied chain leading to n. After the above stabilizations finish, n will set $n.predecessor \rightarrow y$ and y will have $y.successor \rightarrow n$.

After, the establishment of correct associations with the successor and predecessor, $fixFingers()$ which also runs periodically by the nodes, refreshes the finger table entries. The procedure is repeated for each of the non-trivial ring intervals in the finger table. Which are the ones containing non-virtual distinct nodes. The procedure of $n.fixFingers()$ is as follows: $For\ each\ i \in m$: $finger[i] = findCertChain(finger[i])$, where m=number of bits used as index (e.g. SHA1 m=160 bits). The procedure to find a certificate chain to another node is done through the $findCertChain(target\_node)$ and follows original Chord $find\_successor()$ algorithm, as described in the following section.

3.4 Certification

When n wishes to authenticate with a peer, it will require a valid chain of trust. In an established SOMA with N number of peers, n will hold all the routing information that is needed on its finger table, similarly to Chord. In addition, n will hold a keyring with all the public keys and information that will allow it to make decisions, based on a trust metric. When two nodes need to mutually authenticate, they construct a certification request and exchange their credentials using the following:

1. $A \longmapsto B : Cert_A, n_1, t_1, ID_B, data_1, (n_1, t_1, ID_B, data_1)Sk_A$
2. $B \longmapsto A : Cert_B, n_2, t_2, ID_A, data_2, (n_2, t_2, ID_A, data_2)Sk_B$

$Cert_{A,B}$ are the respective node certificates. $n_{1,2}$ are nonces and $t_{1,2}$ are expiration timestamps used to ensure freshness and that they were received in a valid time-frame. The $data_{1,2}$ can include a session key encrypted with the receiver's public key and also other information related with the origin of data and implementation specifics.

When n wishes to verify the credentials of a node p, it will have to get a trust chain leading to p. Therefore n will first check if p is already in its finger table or if p is an introducer for n. If any of the two hold true, then n will have a chain of certificates from the stabilisation protocol leading to p that can be directly verified for its validity. If ID_p is not included in its finger table, then node n will make a certificate chain lookup to find the PGP chain to p.

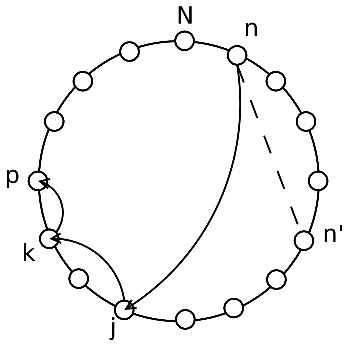

Fig. 1. An example of the logical representation of SOMA ring with N nodes.

Following the above, n with ID_n has introducer ID_n' and wants to authenticate the certificate of p (Fig. 1). Node n finds a certificate chain to ID_p, by sending a *findCertChain()* request to an intermediate node *j*, *j.findCertChain(p)*. Node j is chosen because ID_j is the closest finger that precedes ID_p in n's finger table and hence in the circular identifier space. Node n has already exchanged credentials with node j at join/stabilisation. Upon receiving the request by n, j will look in its finger table finding that the closest preceding finger is k and will reply with this information back to n providing also the certificate containing $Cert_{j \Rightarrow k}$. Finally, node n will ask $k.findCertChain(ID_p)$ similarly returning to n the final link $Cert_{k \Rightarrow p}$. As a result, n receives on each query a certificate that will allow it to build a chain of trust up until ID_p, with each time reducing the distance half-way through. Node n logarithmically gets closer to the target ID_p, building the needed trust path.

3.5 Revocation

In every PKI, dealing with the revocation of certificates can be a technical challenge. In SOMA a node can implicitly or explicitly revoke its published certificate. An implicit revocation by a node takes place when a certificate is allowed to naturally expire. Each issued certificate has a predetermined lifetime. When its expiration time passes it will no longer pass any validation tests taken by the nodes and hence, the certificate will be considered implicitly revoked. If a node is aware that its private key has been compromised, it can explicitly revoke the certificate for the public key in question. This is done by using a revocation certificate and without introducing a complex reputation scheme. Each node, creates and stores safely (e.g. in external media) a revocation certificate that will be used as insurance in case of a key compromise. The revocation certificate will be used to effectively separate the ID/node relationship for that node in the ring.

The node that revoked its certificate does not need to send the revocation request certificate to all the nodes in SOMA. The only interested parties in the revocation scheme are the ones that point in their finger table to the revoked key. Therefore, the node only needs to provide the revocation certificate to its predecessor and exchange it through the stabilisation protocol with all the nodes that update their finger table to the node itself. In this way, the Certificate Revocation List (CLR) is build conjointly with the keyring for each node and will propagate to all the nodes that require it. Furthermore, when a node requires to check if a certificate is currently revoked, then it only needs to proceed with the normal lookup operation. The nodes along the SOMA ring will provide the revocation certificate if has been published.

4 Evaluation

4.1 Security Analysis

The design characteristics of SOMA focus on decentralisation through self-organisation, scalability and robustness against malicious behavior. In this section we analyse the shortcomings and possible attack scenarios regarding the certificate mechanism, the control and use of key material and, finally, the overlay routing itself.

Node Join: When a node joins the ring, a malicious bootstrapping node could attempt to provide false credentials to the rest of the SOMA nodes. This attack would be negated in our case, since the node's public key and physical IP address is connected with its logical ID on the ring. Any false credentials provided by the bootstrapping node would require a valid certification chain, which it can not forge due to the certificate digital signatures.

Certificate Chain: A node wanting to find a certificate chain to another node, needs to authenticate first, a chain of intermediate nodes. These intermediate nodes have valid IDs and the digital certificates provide authentication and non-repudiation for the each node in the certificate path. The consistent hashing mechanism between the physical and the logical address provides a simple defence against impersonation and Sibyl attacks. If one of the nodes, misbehaves or simply creates multiple identities, it will be detected, since the certificates are bound to their address. Therefore, this node can simply be ignored and move on the previous node preceding the target node in the finger table. As long as a single node in the finger table follows the protocol the authentication can proceed. Moreover, an integrated reputation model would help in dealing with misbehaving nodes accordingly to the predefined rules.

Denial of Service: If a node joins a SOMA ring, where the majority of the nodes are malicious, then its identity even though could not be forged, the authentication service for that node could be potentially disrupted through denial of service (DoS). Against DoS attacks, SOMA is resilient due to the fact it uses consistent hashing to distribute the logical identities of the nodes. Therefore, malicious nodes would require a large majority to be able to control the full certificate log n path. This is due to the fact that the IDs are mapped using consistent hashing where the standard hardness assumptions for the chosen hash function apply.

Credential Exchange: An attack directly on the certificate exchange is thwarted by the use of nonces and timestamps, which ensure freshness, and prevent man in the middle attacks (MITM). Additionally, the inclusion of origin and target data safeguards against certificate hijacking and all forms of impersonation.

Overlay attack: Attacks on the routing protocol, itself, can be hard to avoid if the majority of nodes are malicious. Even though impersonation is averted through the logical-physical address relationship of the public key certificates, a DoS attack could potentially disrupt the overlay as a whole. In such a case, a reputation model would provide the necessary insight to marginalise or expel malicious nodes. Attacks on the network infrastructure itself could include churn attacks and potential network failures from the malicious nodes. Against such attacks, SOMA is resilient by requiring at least one correct node in the finger table for correct routing. In addition, instead of using a single successor, employing successor lists will provide additional routes and mitigate the effects of overlay attacks.

4.2 Critical Appraisal

SOMA provides authentication service, data integrity, confidentiality and non-repudiation for the nodes. But to have, a fully self-organised network, a policy model such as the one proposed is not enough to guarantee sound decision making. A reputation system would need to be integrated in the model as a monitoring mechanism. This integration can be easily achieved in the form of a trustworthiness metric in the finger table of the nodes. The reputation model would provide additional insight and better informed reasoning that would be based on predefined rules for trust establishment. Our system is designed so that it can incorporate a reputation model for the nodes to decide whether or not to place their trust. Such an extension can be through observations or recommendations by the peers themselves. The integration of such a model on top of SOMA will be investigated in future work.

SOMA is a hybrid solution based on structured Peer-to-Peer to tackle the limitations that arise in threshold cryptography, random walks and pure Web-of-Trust solutions. It utilises the collective knowledge gained by structured peer-to-peer overlays and combines it with a certificate chaining solution to create an autonomous self-organised authentication system suitable for our goals. By taking advantage of the mathematical guaranties of the Chord infrastructure, path discovery for the chain of trust building becomes scalable to many thousands of nodes and with bounded path traversal latency. Chord is application agnostic and utilises consistent hashing in the same way we apply it in SOMA, therefore SOMA can guarantee O(log N) authenticated certificate chain depth, routing efficiency for certificate discovery of O(log N) communication hops and O(log N) state per node as it was seen in the certification example in (Fig. 1).

In SOMA, there are no centralised authorities and no empowered nodes because it would limit the autonomy of the nodes and disrupt the nature of a mesh network. We do not adopt any form of threshold cryptography, since it would introduce an unnecessary trade-off between security and scalability. To extrapolate further on the (n, t+1) schemes, an open mesh network has many thousands of nodes and hence, we can not assume that an adversary would not compromise t+1 nodes in any given time frame. Moreover, we would need appropriate adaptive control mechanisms to calculate the thresholds, depending on the dynamics of the network topology, churn and number of nodes in the system. Therefore, we avoid the risk that complex voting or threshold based schemes introduce.

5 Conclusions

This paper has proposed a large-scale authentication system for mesh networks without the need of a TTP. SOMA builds on top of Chord to provide us with performance guaranties. Our proposal, allows each node to decide its own trust policy while keeping its autonomy intact. We have shown how the overlay can be used as a meta-structure to infer trust relationships and not as the means to provide distributed directory storage. Wireless ad-hoc networks have inherent vulnerabilities, the nodes can always be considered susceptible to security attacks and to physical capture. Thus, in creating SOMA we took into account the security of the network as a whole and provided the building blocks for establishing trust without relying on centralised authorities. Finally, the possible attacks and limitations of our proposal were discussed, providing the grounds for future work.

References

1. AWMN: Athens Wireless Metropolitan Network, `www.awmn.gr`
2. Bicket, J., Aguayo, D., Biswas, S., Morris, R.: Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In: 11th Annual International Cconference on Mobile Computing and Networking (2005)
3. Zimmermann, P.: The Official PGP Users Guide. The MIT Press, Cambridge, MA (1995)
4. Gnutella RFC v0.4, `http://rfc-gnutella.sourceforge.net/index.html`
5. Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In: ACM SIGCOMM Technical Conference (2001)
6. Shamir, A.: How to Share a Secret, ACM, vol. 22, Issue 11,pp. 612 - 613, (1979)
7. Zhou, L., Haas, Z.: Securing Ad-hoc Networks. IEEE Network, vol. 13, no 6, pp. 2430, November/December (1999)
8. Kong, J., Zerfos, P., Luo, H., Lu, S., Zhang, L.: Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks. In: 9th International Conference on Network Protocols(ICNP), November (2001)
9. Douceur, J. R.: The Sybil Attack. Springer Berlin / Heidelberg, Microsoft Research, One Microsoft Way, Redmond, WA, 98052-6399, USA (2002)
10. Zhou, L., Schneider, F., Renesse, R.: COCA: A Secure Distributed Online Certification Aauthority. ACM Transactions on Computer Systems (TOCS), vol. 20, Issue 4, pp. 329 - 368, November (2002)
11. Yi, S., Kravets, R.: Practical PKI for Ad-hoc Wireless Networks. Department of Computer Science, University of Illinois, USA (2001)
12. Yi, S., Kravets, R.: Key Management for Heterogeneous Ad-hoc Wireless Networks. In: 10th IEEE International Conference on Network Protocols (ICNP), (2002)
13. Yi, S., Kravets, R.: MOCA: Mobile Certicate Authority for Wireless Ad-hoc Networks. In: 2nd Annual PKI Research Workshop, (2003)
14. Capkun, S., Hubaux, J., Buttyn, L.: Mobility Helps Security in Ad-hoc Networks. In: Mobile Ad Hoc Networking and Computing(MobiHoc), (2003)
15. Capkun, S., Buttyn, L., Hubaux, J.: Self-organized Public-key Management for Mobile Ad-hoc Networks. IEEE Transactions on Mobile Computing, Vol. 2, Issue 1, pp. 52 - 64, January- March (2003)
16. Capkun, S., Hubaux, J., Buttyn, L.: Mobility Helps Peer-to-Peer Security. IEEE Transactions on Mobile Computing , vol. 5, Issue 1, pp. 43-51, (2006)

17. Kambourakis G., Konstantinou, E., Gritzalis, S., ”Binary Tree Based Public-Key Management for Mobile Ad Hoc Networks”, Proceedings of the ISWCS’08 5th IEEE International Symposium on Wireless Communications Systems, pp. 687-692, October 2008, Reykjavik, Iceland, IEEE.
18. Boneh, D., Franklin, M.: Identity-based Encryption from the Weil Pairing Advances in Cryptology. In: CRYPTO 01, (2001)
19. Bobba, R. B., Eschenauer, L., Gligor, V., Arbaugh, W.: Global Telecommunications Conference. GLOBECOM ’03. IEEE Publication Date 1-5, December (2003)
20. Aberer, K., Datta, A., Hauswirth, M.: A Decentralised Public Key Infrastructure for Customer-to-Customer E-commerce. International Journal of Business Process Integration and Management, vol. 1, No. 1, pp. 26-33, (2005)
21. Avramidis, A., Kotzanikolaou, P., Douligeris, C.: Chord-PKI: Embedding a Public Key Infrastructure into the Chord Overlay Network. Springer Berlin / Heidelberg, vol. 4582/2007, ISSN: 0302-9743, pp. 354-361, (2007)
22. Watts, D.J., Strogatz, S. H.: Collective Dynamics of ’small-world’ Networks. Nature, vol. 393, doi: 10.1038/30918, 440442, June (1998)
23. Open PGP November 2007/RFC 4880, http://tools.ietf.org/html/rfc4880