Universidade Federal do Paraná

Departamento de Informática

Michele Nogueira Lima
Eduardo da Silva
Aldri Luiz dos Santos
Luiz Carlos P. Albini
Guy Pujolle

Survivable Public-Key Management for Self-Organized Wireless Ad Hoc Networks

Relatório Técnico
RT-DINF 002/2010

Curitiba, PR
2010

Resumo

Cryptographic techniques are at the center of security solutions for wireless ad hoc networks. Public key infrastructures (PKIs) are essential for their efficient operation. However, the fully distributed organization of these networks makes a challenge to design PKIs. Changes in network paradigms and the increasing dependency on technology require more dependable and survivable PKIs. This paper presents a survivable PKI whose goal is to allow its operation even in face of attacks or intrusions. The proposed PKI is based on the SAMNAR architecture in which an adaptive cooperation among preventive, reactive and tolerant defense lines is used to achieve survivability. The PKI employs also different evidences to prove the liability of users for their keys as well as social relationships for helping public key exchanges. Analytical and simulation results show the improvements attained by our proposal in terms of effectiveness and survivability to different attacks.

1 Introduction

Wireless ad hoc networks (WANETs) - mobile or stationary - are composed of devices (nodes) communicating among themselves in a wireless multi-hop fashion [1]. Such networks allow communication over a shared wireless channel without any pre-established infrastructure or centralized management. Due to their characteristics, WANETs are prone to different threats, for example: (*i*) wireless communication make them susceptible to interceptions, interferences or passive eavesdropping; (*ii*) multi-hop communication allows malicious or selfish behavior due to required cooperation among nodes [2].

Many solutions have been proposed to provide security on WANETs [3–5]. The majority of them apply cryptographic techniques in order to enforce integrity, confidentiality, authentication, and non-repudiation in link-layer connectivity, routing, or end-to-end communication. Cryptographic techniques rely on a keying material, which determines the functional output of cryptographic algorithms, controlling the complexity in breaking encrypted messages, authenticating nodes and users, proving their trustworthiness, and validating messages. This material can include public/private key pairs, secret keys, initialization parameters, and non-secret parameters.

To allow secure communications, cryptographic keys must be distributed and managed. A proper key management system must ensure node legitimacy, key generation, availability, storage, distribution, and revocation. However, due to the self-organization of WANETs and the lack of a central entity, designing key management systems is a challenging task. Even though several key management schemes for WANETs can be found in the literature [2], changes in network paradigms towards pervasive and dependable computing demand for designing reliable, survivable and scalable key management schames [6].

This work proposes a survivable and reliable public key infrastructure (PKI) for WANETs, called *S*ecure Group-Based PKI (SG-PKI). Its goal is to provide key management operations even in face of attacks or intrusions. SG-PKI is based on the SAMNAR architecture [7], and on groups build based on the relashionship of the users. The SAMNAR architecture offers an adaptive cooperation among preventive, reactive and tolerant defense lines to achieve survivability. It is also presented different types of evidences to prove the liability of using the relashionship of the users as basis to group formation in SG-PKM. Simulation and analytical evaluation show its effectiveness and survivability to attacks.

The paper is organized as follows: section 2 discusses related work; section 3 provides an overview of the SAMNAR architecture; section 4 presents the models and assumptions used by SG-PKM; section 5 detaisl all operation of the SG-PKM; simulation and analytical analyses are in section 6. Finally, section 8 concludes the paper and outlines future work.

2 Related Work

The first key management proposals have adapted traditional key management systems for WANET conditions [8, 9]. In general, they are based on certificate authority (CA) functionalities in order to securely distribute keys. Public key management approaches designed for WANETs can be classified in [2]: identity-based [10], chaining-based [11–13], cluster-based [14, 15], predeployment-based [16] and mobility-based [17]. Among them, the chaining-based schemes appear to be the most suitable scheme to the WANETs environment.

The *Self-Organized Public Key Management System* [11–13] is the main chaining-based key management scheme. From now on it will be called *PGP-Like*. It is a public key management scheme that uses certificate chains. Private and public keys of nodes are created by the nodes themselves following the PGP concepts [18]. In addition, each node issues public key certificates to other nodes it trusts. In PGP-Like, if a node u believes that a given public key K_v belongs to a given node v, it issue a certificate binding K_v to the node v, $(v, K_v)_{prK_u}$, where prK_u is the private key of node u. This certificate is stored in both nodes local certificate repositories. Furthermore, each node periodically exchanges its own repository with its neighbors. Each node u maintains an updated local certificate repository, G_u, and a non-updated local certificate repository, G_u^N [12]. The non-update local certificate repositories contains the certificates that have expired and they are considered revoked.

When node u wants to authenticate the public key K_v of node v, it firstly tries to find a path from vertex K_u to vertex K_v in G_u. If $\exists(K_u \rightsquigarrow K_v) \in G_u$, node u authenticates it. If $\neg\exists(K_u \rightsquigarrow K_v) \in G_u$, node u merges G_u with G_v, $G' = G_u \cup G_v$, and it tries to find $(K_u \rightsquigarrow K_v) \in G'$. If such path exists the authentication succeeds. The path found in the repositories is a certificate chain. Note that, certificate chains are weak authentications, as they assume that trust is transitive. Unfortunately, ensuring a valid transitive trust with more than two nodes in the chain is very difficult [19]. The use of certificate chains make PGP-Like highly vulnerable to impersonation attacks, as shown in [20]. An attacker, node x, can create a false identity m and issue a certificate binding k_m to m. All nodes that trust in x will also trust in m. Thus, if node x maintains a correct behavior during a considerable time, several units will, probably, trust in it, and the false identity will be spread over the network due to the certificate exchange mechanism.

Several proposals based on groups of nodes can also be found in the literature. Some of them present characteristics such as resiliency, fault-tolerance or scalability that can improve survivability [21,22]. However, they focus mainly on efficiency neither dealing with a complete survivable system, nor reaching all survivable requirements and properties.

Despite of all the mentioned works, none of them has been designed with survivability in mind. For the best of our knowledge, only few works have proposed survivable key management systems such as [23]. In that work, a survivable and efficient key management system for wireless sensor network is presented focusing on robustness and recoverability. Methods for distributing, maintaining and recovering session keys are defined to work even in case of compromised nodes. However, such scheme is only suitable to wireless sensor networks and its properties are essential to achieve a holistic survivable system [2].

3 Survivable Architecture

After definition of objectives, restrictions and funcionalities of SG-PKM, the Survivable Ad hoc and Mesh Network ARchitecture (SAMNAR) [24] was choosed to support SG-PKM. In Survivable Ad hoc and Mesh Network ARchitecture (SAMNAR) [24], a survivable architecture used as support to SG-PKM, the authors argue that survivability can be achieved with an adaptive cooperation among the three defense lines – preventive, reactive and tolerant. SAMNAR contains the **survival**, **communication** and **collect** modules as illustrated in Fig. 1. The **survival module** holds five independent components, being four ones related to survivability properties: resistance, recovery, recognition and adaptability, and the control component. The properties represent respectively the capability of the key management system to repel attacks; detect attacks or evaluate the damage extension; restore disrupted information or functionalities; and quickly incorporate lessons learned from failures and adapt to emerging threats.

In SG-PKM, the *resistance component* is composed of preventive local node mechanisms such as personal firewalls, anti-virus, anti-spyware and others. It is also reinforced by some cryptographic operations such as digital signatures and by Message Authentication Code (MAC). These mechanisms can be integrated among them or not, but in all cases they provide inputs for the control component. These inputs are information about the mechanisms such as the key length used in the cryptographic operations, the cryptographic algorithm, the last update version of the anti-virus or anti-spyware

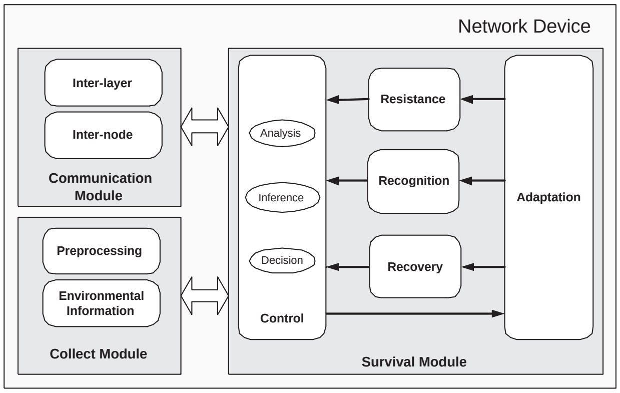

Figura 1: SAMNAR Architecture

database, statistics about attacks or intrusions and others. All resistance mechanisms are self-adjustable changing their configuration depending on the network or environmental conditions. For example, the rules of a personal firewall can be more rigorous in certain environments, while more tolerant in other ones.

The *recognition component* comprehends reactive mechanisms to identify malicious behaviors such as intrusion detection systems (IDSs) or reputation systems. In SG-PKM, recognition is achieved by a reputation system, though other mechanisms could also be employed. This system is responsible for evaluating the reputation level of the physical neighbors of a given node. Reputation levels are also inputs for the control component. By these mechanisms, SG-PKM can be reconfigured dynamically, i.e. parameter or threshold values could be changed based on network and environmental conditions.

The *recovery component* is responsible for providing the recovery and redundancy of the keying material. In SG-PKM, these mechanisms are applied in different operations such as certificate generation, renewal processes or public key authentication. In the certificate generation, for instance, a group of users must be created in order to have a kind of redundant witnesses of public key exchanges. This approach tries to minimize the possibility of false certificates or false identities in the system. Redundancy is also employed in the authentication process in which the system always needs to confirm a valid authentication. Some other uses of redundancy in SG-PKM are discussed in Section 5.

The *adaptation component* complements the previous ones being responsible for executing changes based on the analyses, inferences and decisions of the control component. These changes can be, for example, in the rules of the personal firewall, in the threshold value of the reputation system or in the redundancy requirements of a key management operations. Adaptation component is also responsible for learning from previous actions and, later, making same actions if the node or the network presents a similar condition.

The *control component* manages and coordinates all modules in the architecture. It receives information from communication and collects modules, as well as from the resistance, recognition and recovery components. The control component correlates and analyzes all information in order to make inferences and decisions. All decisions are sent to the adaptation component that define and send satisfactory parameter values to other modules or components.

The **communication module** is responsible by cross-layer and inter-node communications. The *inter-layer component* provides information about different network layers to to control component, that makes decisions based on all network layers. Already the *inter-node component* provides information communication, exchange and synchronization among nodes, aiming to guarantee the survivability of the whole network. Example of this information is node configurations or statistics about intrusion detections.

The **collect module** holds mechanisms to gather all data required by the survival module. It is out of the architecture scope to define the collection method. However, the survival module specifies adaptively which data and information must be collected following its requirements. The collect module is composed of the *preprocessing component* and the *environmental information component*. The first one is exploited when gathered data need to be treated before sending it to the survival module. Normalizations, previous calculations and others are examples of preprocessing used to facilitate analyses and inferences of the survival module. The second component stores information gathered periodically about network conditions, sending it to the survival module when required.

4 Notation, Models and Assumptions

Table 1 summarizes the notation used in the SG-PKM.

Network model: the multi-hop wireless ad hoc network consists in a set of n mobile or stationary nodes identified by $X_1, X_2, X_3, ..., X_n$. The network is self-organized and nodes can freely move on the given area. No support infrastructure exists neither a central control entity to manage network resources. Hence, nodes have similar functionality contributing to the network maintenance, routing process and public key management.

Two given nodes X_i and X_j have a *physical wireless link*, if their Euclidean distance is no greater than r, the communication range, and, thus, X_i and

Tabela 1: Used notation

Note	Explanation
i	a given user identity
X_i	a node identity
X	PKI nodes set
IG_w	identification of a given group w
IG	initiator group set
m	number of users in an initiator group
p_i	public key of a given user i
s_i	private key of a given user i
P_w	public key of a given initiator group w
$(P_u \rightsquigarrow P_v)$	certificate chain between P_u and P_v
S_w	private key of a given initiator group w
$C^{i}_{S_w}$	public key certificate binding the public key of a given identity i signed with the private key of a given group w
$C^{IG_z}_{S_w}$	group certificate binding the public key of the group IG_z and signed with the private key of a group IG_w
T	the expiration time of a certificate
T_{ex}	certificate exchange time
$SIGN[a]_{S_w}$	signing a given information a with S_w
$AUTH[X_i \rightsquigarrow X_v]$	X_i is authenticating p_v of X_v
$MAC(w)$	message authentication code of a given group identification
$a\|b$	a given information a is concatenated with a given information b
G_i	repository of updated certificates of X_i
G_i^N	repository of non-updated certificates of X_i
G	group certificate graph
$\lvert Z\rvert$	Size of a given set Z

X_j are called neighbors in respect to each other. A *physical path* between two nodes, for example, X_i and X_k, is a set of subsequent physical wireless links. Two nodes are physically connected if there is a physical path starting at one and ending at the other. No node has complete knowledge of the physical network topology requiring routing to communicate with nodes out of its communication range.

Trust model: Trustworthiness among nodes depends on the existing friendship of users participating on the network. If two users, e.g. i and j, trust each other, their respective devices, X_i and X_j, can exchange their public keys. A given node trusts in another only if their users have exchanged their public keys through a side channel (e.g., over an infrared channel). As in [1], trustworthiness between two nodes is considered to be bidirectional, that is, if X_i trusts in X_j, X_j also trusts in X_i. This assumption is based on statistical analysis of the “Web of Trust” among users of PGP. This analysis shows that about 2/3 of the links in a large strongly connected social network are bidirectional [25].

Friend relationships form a spontaneous network [26], being independent of the physical network and presenting social network properties such as small word [27] and scale-free phenomena [28]. The small world phenomena is found in social networks where every pair of user can be reached through a short chain of social acquaintances [27]. Already the scale-free phenomena results from the existence of few users with greater number of friends than others. Moreover, these few users will have high probability to be chosen by new ones as their friends ("the rich get richer" paradigm [28]).

Threats model: Different types of attacks can harm PKIs in WANETs. The following analysis focus on those attacks that can compromise availability, confidentiality, integrity, authenticity and non-repudiation principles in a public key management system. An attack scenario is considered as an adversary being able to compromise one or more nodes and, consequently, to avoid or delay key management system functions. Specifically, following attacksare handled: Sybil, masquerade and denial of service (DoS) attacks [4]. Other attacks are out of the scope of this paper.

Sybil: Sybil attacks occur when adversary nodes create multiple identities in the PKI in order to manipulate keys and certificates in their advantage. False node identities can operate as legitimate ones and, thus, they can violate confidentiality, authentication and non-repudiation principles.

Masquerade: a malicious node can forge the identity of a legitimate node, violating the non-repudiation and authentication principles. Malicious nodes can generate these attacks to participate in the key management as a legitimate node. Moreover, through this attack, nodes may be able to compromise the integrity and confidentiality of the messages. Masquerade attacks can also be used in the elaboration of man-in-the-middle (MITM) attacks [4].

Denial of service (DoS): a misbehavior node, malicious or not, may stop providing authentication service as well as key storage or certificate generation, distribution or revocation. Hence, it decreases the good operation of key management services. A motivation for this attack can be, for example, saving resources, such as storage or processing, while the node still takes part in the key management system. However, a given compromised node can maliciously participate in the key management system to damage it.

5 Survivable Key Management System

In this section, we introduce our survivable PKI, called SG-PKM. First, we give a brief overview of SG-PKM structure. After, we describe the PKI operations corresponding to the creation of public keys and public key certificates, certificate renew and revocation, and authentication. We explain them focusing on survivability and how it can be achieved. Explanations take into account the assumptions and models described in Section 4.

For simplicity, we assume that each honest user owns only one node in the *physical network*. Hence, a node corresponds to a user. Our PKI follows the "WAN-of-LANs" paradigm [29] meaning that it is decomposed into small groups called *initiator groups* (IGs).

Initiator groups are composed of nodes whose users have a friend relationship among them. All nodes in a group have the same role without needing group leaders. Groups are essential for joining a new node to the PKI, for issuing certificates or renewing keys. However, the maintenance of initiator groups is not critical for our PKI. It is designed in order to self-adjust to changes, and also to minimize the computational cost in maintaining groups and network overhead.

Fig. 2 illustrates two initiator groups, IG_1 and IG_2. IG_1 is composed of X_1, X_2, X_3, X_4, X_5, X_6, X_7, and X_8, whereas IG_2 is composed of X_7, X_8, X_9, X_10, X_11, X_12, X_13, and X_14. The respective users owning the nodes of IG_1 are friends as well as the users owning the nodes of IG_2. Nodes into a group reciprocally issue public key certificates among them. These certificates are represented by the double arrows meaning the existence of certificates issued mutually between two nodes. In this case, we represent also an intersection between IG_1 and IG_2 by the nodes X_7 and X_8.

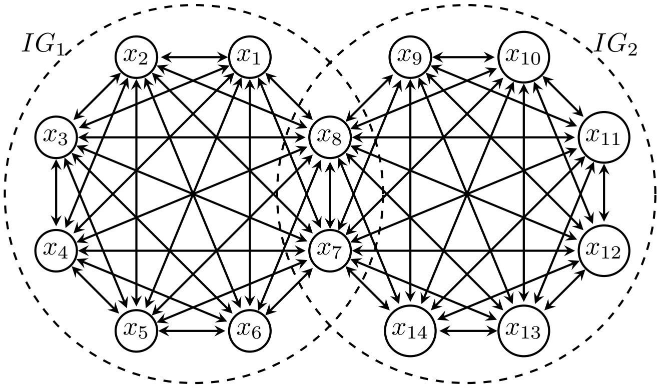

Figura 2: Initiator groups

Following ideas in [30], groups also provide evidences about the liability of nodes for their public keys and the liability of the group for their digital signatures. Our PKI aggregates different types of evidence such as node reputation and node preventive level. Moreover, a group offers a kind of testimony among its nodes reinforcing the proofs of their liability for certificates and keys. These aspects have been employed to increase the recovery property of the PKI.

Public key certificates are used to bind a public key to an identity. Hence, in our model we have two types of public key certificates: *node certificates* and *group certificates*. Node certificates bind user public keys with user identities, whereas group certificates bind group public keys with group identification. Node certificates are signed with the private key of the group in which the node participates. Group certificates are signed by the private key of another group.

A given node certificate $C^{j}_{S_w}$ is composed of an expiration time T, the node identity X_j, its public key p_j and the message authentication code (MAC) of X_i initiator group identification. All this information is signed with S_w, i.e., the private key of the group IG_w. In addition, certificates also own the X_i initiator group identification. In a nutshell, $C^{j}_{S_w}$ holds:

$$C^{j}_{S_w} = (SIGN[T \| X_j \| p_j \| MAC(IG_w)]_{S_w} \| IG_w) \tag{1}$$

Group certificates follow the same organization of node certificates. However, a given group certificate $C^{IG_z}_{S_w}$ consists of:

$$C^{IG_z}_{S_w} = SIGN[T \| IG_z \| P_z]_{S_w} \tag{2}$$

For facilitating our proposal description, an abstract model based on graph theory gives support to explain many PKI operations. This approach was used on [9,31], but, in our model, only group certificates and group public keys are represented in a graph $G(V, E)$, called *group certificate graph*. Public keys of groups compose the set of vertices V and group certificates compose the set of directed edges E.

To summarize, Fig. 3 provides an overview of presented models and their interrelation.

5.1 Creating public keys and certificates

In SG-PKM, each user individually creates its pair of keys, p_i and s_i, and stores them in the node X_i. After generating p_i, X_i needs to find $m-1$ trusted nodes in order to issue certificates for its public key. The set of m nodes, including X_i, compose an IG. These m nodes need to trust in each other and their trustworthiness follows the friend relationship existing among their users. Nodes in a given IG will exchange their public keys among themselves using a side channel such as infrared.

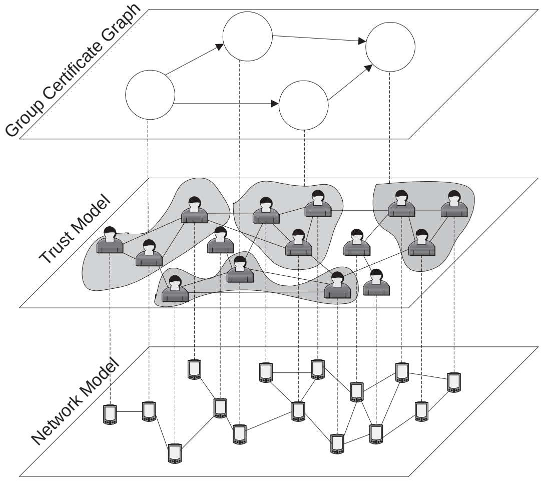

Figura 3: Interrelation among all models

The m nodes will generate together a pair of keys for their group identified by IG_w. This pair of keys can be build using any distributed key agreement scheme without a trusted third party, as [32–34]. Here, we considered Pedersen's threshold scheme (t, m) [32], in which m nodes are necessary to build P_w and S_w in a distributed way. After creation of P_w and S_w, the public key P_w is available to all network nodes and the private key S_w is distributed among m members of IG_w, following a threshold cryptographic scheme (t, m) [35]. After that, t nodes can execute cryptographic operations with these keys, such as signing or encrypting.

The distributed generation of IG_w and the use of a threshold scheme are some approaches applied to improve the tolerance against attacks in SG-PKM. Moreover, group formation based on friend relationships decreases the probability of false identities in the system. In Section 6, we evaluate the best value of m in terms of practical viability. Social networks present a high clustering coefficient demonstrating a great amount of loops of order three [28]. This means a high probability of two friends to have a friend in common.

After generating P_w and S_w, public key certificates will be issued, binding the public key of each member of IG_w with its respective identity. These certificates, called *node certificates*, are signed with the private key of the group and locally stored by nodes themselves. In the end of this phase all nodes in the IG_w will possess certificates for all nodes in the group.

The public key P_w of a given IG_w also needs to be certified. Then, groups can issue certificates among themselves binding a given P_w with its identity, called *group certificates*. IG_w can issue a certificate $C_{S_w}^{IG_z}$ for IG_z, if IG_w

believes in IG_z. A given group IG_w believes in another IG_z if at least one node in IG_w trusts two or more nodes in IG_z, or two or more nodes in IG_w also participate in IG_z. The required redundancy with two or more nodes intends to improve the reliability in evaluating public key liability.

5.2 Certificate exchange

Each node possesses two local repositories to store updated and non-updated certificates. The updated repository of a given node X_i is represented by G_i. This kind of repository holds node or group certificates that are still valid. When the certificate time T expires, it becomes a non-updated certificate and it will be moved to the non-updated repository. The non-updated repository of a given node X_i is represented by G_i^N.

Nodes periodically exchange their group certificates with their neighbors in the physical network depending on node reputation and preventive level. Initially, each node holds only the certificates of groups that it participates, and the certificates that nodes in its groups have issued for other groups. With the periodic certificate exchange, each node increases the number of group certificates in their local repositories.

Each node requests to their physical neighbors the list of group certificates they keep. This message can be sent via *piggybacking* with control messages used MAC protocol at neighbor discovery. This mechanism is presented in Algorithm **??**, as follows. A given node X_i sends to its neighbor a hash of its local repository, and requests them the missing ones. Each neighbor responds with a message containing the group certificates that node X_i does not have stored. Finally, upon receiving the neighbors' certificates, node X_i stores these certificates in its non-updated group certificate repository (G_i^N).

Certificate exchanges are performed in time interval T_{ex}. For simplicity, we assume that all nodes follow the same value of T_{ex} and that exchanges are not synchronized. Hence, if a given node X_i is sending its certificates to a node X_j, this does not mean that X_j is also sending its certificates for X_i at the same time.

5.3 Authentication

When a node X_i needs to authenticate the public key p_j of a node X_j, X_i requests to X_j the certificate issued for its public key. X_j can participate in many groups, and then replying any certificate issued to it. Hence, X_i can choose one or more certificates to validate. Into each certificate, nodes can know the identification of its initiator group. Algorithm **??** demonstrates a node X_i authenticating the certificate $C_{S_y}^{X_j}$, signed by members of group IG_y.

Supposing that X_i have chosen $C^j_{S_w}$, it will need to use P_w to validate this certificate. However, before using P_w, X_i needs to authenticate P_w. The authentication of P_w is realized by a chain of group certificates. Then, for authenticating P_w, X_i searches at least two chains of valid group certificates connecting its initiator groups to IG_w in its updated group certificate repository. If $\exists(P_y \rightrightarrows P_w) \in G_i : X_i \in IG_y$, node X_i validates the public key P_w of group IG_w and, then, validates the certificate $C^j_{S_w}$ of node X_i.

However, if $\nexists(P_y \rightrightarrows P_w) \in G_i : X_i \in IG_y$, node X_i will merge its updated group certificate repository with the updated group certificate repository of X_j ($G_1 = G_i \cup G_j$). So, X_i searches at least two chains of valid group certificates connecting its initiator groups to IG_w in G_1. Likewise, if $\exists(P_y \rightrightarrows P_w) \in G_1 : X_i \in IG_y$, node X_i validates the public key P_w of group IG_w and, then, validates the certificate $C^j_{S_w}$ of node X_i.

If even after merging the repositories, $\nexists(P_y \rightrightarrows P_w) \in G_1 : X_i \in IG_y$, node X_i it will try to find them in the union of its updated and non-updated repositories. In the successful case, X_i will need to verify if the binding between identity and public key into non-updated certificates are still valid. The validation will be detailed in Section 5.4. If none of these cases happen, X_i will not be able to authenticate the group public key or the node public key.

As an example, supposing that Fig. **??** represents the group certificate repository of a given node X_i, member of IG_1, and that it wants to authenticate a given node X_j into IG_4. Thereby, X_i must find at least two chains of valid certificate connecting P_1 and P_4 in its local group certificate repository. In the example, X_i can use the chains $P_1 \rightarrow P_2 \rightarrow P_4$ and $P_1 \rightarrow P_3 \rightarrow P_4$ to validate the public key P_4, and then, authenticate X_j.

5.4 Validation of group certificates

As mentioned, all certificates received via certificate exchange mechanism and certificates with expired lifetime are stored in the non-updated group certificate repositories. When a node X_i wants to validate a certificate, it must to send a message to all members of group that issued the certificate. The validation must be done by at least t nodes from this group. Before node certificates have their time expired, their initiator group can issue a new version of the certificate. If a subset t in a given IG_y do not have any reason to revoke a given node certificate $C^i_{S_y}$, they can issue a updated certificate, with a new expiration time. Using t nodes, instead of the m nodes of the group, minimizes the overhead in the physical network without losing redundancy feature. After updating a node certificate, one copy is sent for all nodes in IG.

The Algorithm **??** presents a pseudo-code of validation operation. In this case, node X_i sends a Validate Request (VREQ) message to all members of issuer group of $C_{S_y}^{IG_w}$ and waits for at least t Validate Reply (VREP) messages. If X_i does not receive these replies in a timeout period, it will not be able to validate the certificate.

5.5 Updating certificates

Before node certificates have their time expired, their initiator group can issue a new version of the certificate. A subset t of nodes from a group that issued a certificate can issue a new version of this certificate if they believe that the binding "user–public key" of this certificate is still valid. Using t nodes, instead of the m nodes of the group, minimizes the overhead in the physical network without losing redundancy feature. The mechanism used to update a group certificate is different from the one used to update a group certificate. Nodes and groups certificate update is presented below.

An update of a node certificate is started by node itself, that requests to other members of issuer group a new version of its certificate. If a subset t in a given IG_y does not have any reason to revoke a given node certificate $C_{S_y}^{i}$, they can issue an updated certificate, with a new expiration time. They send to node X_i a message of certificate update (*nodeRenewing message*), signed with their respective subparts of the private key of group IG_y. When node X_i receive t messages updating its certificate, it must to send a copy of updated certificate to all other members of group IG_y. Algorithm **??** demonstrates the update operation of node certificate $C_{S_y}^{i}$ of node X_i, signed by members of group IG_y.

Group certificates can also be renewed by a subset of t nodes of the group that has originally issued the certificate. In this case, if a given node X_i, member of group IG_w, needs update its group certificate $C_{S_y}^{IG_w}$, it sends a message requesting the update to all members of IG_y, and waits for at least t replies updating the certificate. Each reply message must be signed with a distinct subpart of private key S_y. In the reply message, each node also sends a list of members that have requested a validation of the certificate that is being updated.

A new version of the certificate, with a new expiration time, is sent for all nodes in the issuer group and for all nodes that have previously requested it. In order to minimize the communication overhead, node X_i can send the updated certificate only to nodes that had requested a validation of this certificate more recently. If a given node X_j does not receive an updated version of an expired certificate, it will move this certificate to its non-updated repository of X_j (G_j^N). If necessary, this certificate must be reactively updated.

Algorithm **??** presents the update operation of group certificate $C^{IG_w}_{S_y}$.

5.6 Revoking certificates

Both node and group certificates can be revoked. Moreover, two kinds of certificate revocation exist: *implicit* and *explicit*. Implicit revocations occur when the validity of the certificate expires, once certificates are issued with an expiration time. This process happens automatically and locally for all certificates stored in the updated repository of each node. No intervention of other nodes in the PKI is requested. Node or group certificates can be implicitly revoked based on the validity period of certificates.

However, many reasons may cause a certificate to become invalid prior to the expiration time. Examples of these reasons are changes in the relationship status between certificate issuer and the key pair owner (e.g., two users have no more friendship relations), and a suspicion that the private key associated with the certificate was compromised. Under such situations, the certificate issuer can to revoke explicitly the certificate.

In the explicit revocation, members of a given group IG_y can revoke a node certificate issued by them, i.e. $C^i_{S_Y}$. It is necessary at least t signatures of members of IG_y to explicitly revoke a certificate. In Algorithm **??**, a node X_v, member of a group IG_y, wants to revoke the node certificate $C^i_{S_y}$ of a node X_i. In this case, node X_v sends a revocation request to all members of IG_y. Receiving a *nodeRevocation* message, each member of IG_y decides by revoke or not the certificate based on information about node X_i. If it also has reasons to revoke the certificate, it returns a message to X_j accepting the revocation of $C^i_{S_y}$. This message must be signed with its subpart of private key of S_y.

If node X_j receives at least t messages accepting the revocation of C^iS_y, this certificate is considered revoked. After, X_j sends a revocation message of C^iS_y, signed with the private key of IG_y, to all members of group IG_y and all members of groups that have issued a group certificate to IG_y. These groups must propagate this information to all nodes that have requested a validation of certificate of group IG_y. So, all nodes that have stores the certificate of group IG_y will be noticed that this group has a compromised node.

Receiving a *nodeRevocation* message, all nodes store revoked node certificate in a local Certificate Revocation List (CRL). Each node uses information in its CRL before authenticates or provides information about a given certificate. The CRL facilitates the authentication procedure, decreasing computation costs when a node is searching valid certificates.

Algorithm **??** presents the explicit revocation of a given group certificate $C_{S_y}^{IG_w}$ by a node X_i. In this case, X_i creates a message of the type *requestRevocation*, and send it to all members of IG_y. As in node revocation, each member of IG_w receiving a *requestRevocation* message, decides to accept or not revoke the certificate. If it accepts the revocation of $C_{S_y}^{IG_w}$, it returns a message signed with its subpart of private key S_y. This reply message also contains the list of nodes that requested a validation of certificate that has been revoked.

When node X_v receives t messages accepting its revocation request of a group certificate, this certificate is considered revoked. So, X_v stores this certificate in its local CRL and, after, sends a revocation message (*groupRevocation*) to all members of IG_y and to all nodes that requested a validation of $C_{S_y}^{IG_w}$. This message is now signed with private key S_y. Each node, when receives a signed *groupRevocation* message, moves the revoked certificate to its non-updated repository and, also, stores this certificate in its local CRL.

6 Analytical Evaluations

The trust model presented in Section 4 provides support for many assumptions and operations of SG-PKM. This trust model is the base for initiator group formation and for the existence of redundant relationships among groups. Despite of forming an initiator group is a requirement for a node to participate of the PKI, this section analysis the feasability of having such groups based in a friend social network. In the same way, the viability of having the required redundancies among groups is evaluated.

For all analysis, we have used a practical example of friend social network, the PGP. As in the trust model assumed by SG-PKM, in PGP public keys are exchanged in a self-organized manner and certificates are signed based on a users' friend relationship. Hubaux et. al [36] have demonstrated that this network formed by public keys and certificates reflects existing social relationships between users. This network presents "small world" and "scale free" phenomena.

For analyzing the viability of existing initiator groups and redundant relationships among them, we use a PGP database and we have applied the methodology and metrics proposed by Latapy et. al [37]. Initially, we observe the PGP database as a symmetric graph $G_{sym} = (V, E)$, in which V is the set of public keys representing the vertices, and E is the set of certificates representing the edges. After, we have extracted maximal cliques of different sizes from G_{sym}. Cliques in a graph means a subset of vertices such that any two vertices are connected by an edge. A clique is called maximal if it is included in no other clique. In SG-PKM, cliques represent initiator groups and show that all nodes have symmetrically changed their public keys.

Table 2 presents statistics about cliques in a PGP graph with $|V| = 956$ and $|E| = 14647$. We have used algorithms proposed by Uno et. al [38] for finding cliques. We compare the number of general cliques with the number of maximal cliques. We observe that only 9 vertices, about 0.9% of the vertices in PGP graph, do not participate of groups. In general, the number of cliques with a size equal to 4, 5, or 6 is greater than others. These results confirm the potentiality of group formation using a PGP graph, proving the first assumption of SG-PKM: the group formation based on the friend relationship between users.

Tabela 2: Clique statistics for a PGP graph

Clique Size	**# of Cliques**	**# of Maximal Cliques**
1	956	9
2	14647	1921
3	47661	4460
4	78016	6599
5	77160	6395
6	49150	4893
9	716	351

In order to evaluate redundancies in PGP graph, we have transformed G_{sym} in a bipartite graph $G_b = (\top, \bot, E)$. In G_b, $\top$ and $\bot$ are disjoint set of vertices and $E \subseteq \top$ x $\bot$. Following the methodology presented in [37], $\top$ is a set of vertices representing maximal cliques of the graph. The $\bot$ is the set of vertices participating in cliques. Relating these concepts to SG-PKM, $\bot$ are public keys representing the nodes or users, and $\top$ are initiator groups. Edges represent the participation of nodes or users into initiator groups.

First of all, we have verified basic statistics in PGP graph. In Fig. 4, we observe the distribution of vertex degree. Vertex degree represents the number of neighbors of a given vertex. As observed in other social networks [37], PGP graph also follows the power law for the bottom degree distribution, while the top degree distribution is Poisson shaped.

We use the redundancy coefficient of a given node v, $rc(v)$, to analyze the redundancy between initiator groups in PGP. The $rc(v)$ is a fraction of pairs of neighbors of v linked to another node than v. Being $N(v)$ the set of neighbors of a given node v, redundancy coefficient is defined as presented in eq. 3.

$$rc(v) = \frac{|\{\{u, w\} \subseteq N(v) : \exists(z, u) \in E,\ \exists(z, w) \in E\}|}{\frac{|N(v)(N(v)-1)|}{2}} \tag{3}$$

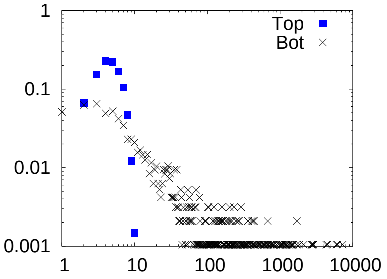

Figura 4: Degree distributions

In Fig. 5, we observe cumulative distributions of redundancy coefficient for $\top$ and $\perp$ nodes. For $\perp$ nodes, 60% of them has redundancy coefficient equal or higher than 80%, whereas 80% of these nodes has redundancy coefficient equal or higher than 50%. This shows the high redundancy of PGP graphs. As expected, the redundancy coefficient is lower for $\top$ nodes.

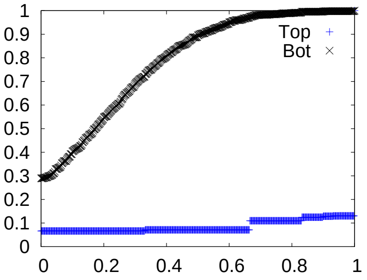

Figura 5: Redundancy distributions

6.1 Surviving threats

Following the threat model of Section 4, malicious users can compromise a PKI in many ways. First of all, a dishonest user may try to trick other users into believing in a false user-key binding by issuing false node certificates. For example, the user may issue a certificate that binds a key p_v to a user f instead of user v.

In our PKI, the probability of the dishonest user to have success is minimal. First, for using the false node certificate, the dishonest user needs to validate it. Knowing that the certificate must be signed with a private key of a given initiator group and, for validating it, a group public key will be used after its validation, if the false node certificate is not signed, it will not be validated.

Supposing that the dishonest user/node has generated $m-1$ false identities and created its own group, this group will need to be trusted by another group. That is, at least two nodes of the false group must participate in an honest group, and hence, the dishonest user will need to convince $m - 2$ honest users, in the worst case.

Another situation is the dishonest user to convince only one user in the honest group to trust in at least two users in the false group. However, for this, the honest user will be based on the preventive levels of the correspondent nodes and on their reputation levels. Considering that the reputation level will be calculated following recommendations of different and random nodes, convincing an honest node to trust false identities can be more difficult than the first situation.

Considering that the false group manages to get a certificate for its public key. The validation of this certificate for its posterior use is another obstacle for the dishonest user. In our PKI, at least two disjoint chains of certificates must be found in the group certificate graph for validating a group certificate. It means that at least two different groups must have issued certificates for a false group. For achieving this, the dishonest user needs to persuade many other honest users, decreasing the probability of a false node certificate being successfully authenticated.

Other threats can happen in a given PKI such as masquerade or DoS attacks. Masquerade attacks are prevented in our proposal by the formation of initiator groups where users know well the identity of the others. Our PKI can easily survive to DoS attacks, once that preventive mechanisms will minimize the possibility of individual nodes to be compromised; misbehaviors such as lack of cooperation can be detected by reputation systems or other mechanisms; and the existing redundancy and fully distribution contributes to increase the tolerance to successful attacks or intrusions.

6.2 Communication cost

In this section, we analyze the communication overhead of SG-PKM, generated by certificate authentication, revocation and renewing operations. All theses communication costs are measured in quantity of messages.

6.2.1 Authentication

In SG-PKM, when node X_i wants to authenticate the certificate $C^v_{S_y}$ of a given node X_v, most operations must be realized locally, by X_i itself. As discussed in Section 5.3, firstly node X_i searches two valid chains in G_i from its initiator groups to the group IG_y. If $\nexists(P_x \rightrightarrows P_y) \in G_i : X_i \in IG_x$, it will create $G_1 = G_i \cup G_v$, requesting G_v from X_v. If $\exists(P_x \rightrightarrows P_y) \in G_1 : X_i \in IG_x$, communication overhead to authenticate the certificate $C^v_{S_y}$, denoted by $ACO(C^v_{S_y})$ is:

$$ACO(C^v_{S_y}) = (UR\_Req + m.UR\_Rep) \ . \ \Delta h_{X_i - X_v} \tag{4}$$

in which $\Delta h_{X_i - X_v}$ is the average number of hops between X_i and X_v, and UR_Req and UR_Rep are, respectively, the request and reply messages of certificates from G_v.

However, if $\nexists(P_x \rightrightarrows P_y) \in G_1 : X_i \in IG_x$, node X_i will use information from its non-updated repository, creating $G_2 = G_i \cup G^N_i$. Se $\exists(P_x \rightrightarrows P_y) \in G_2 : X_i \in IG_y$, for each non-updated group certificate used to form the two chains, node X_i must request the validate for the issuers of the certificate. Thereby, the total cost to authenticate depends on the quantity of non-updated group certificates in the found chains. In SG-PKM, the overhead communication to validate a given group certificate $C^{IG_w}_{S_y}$, denoted by $VCO(C^{IG_w}_{S_y})$, is:

$$VCO(C^{IG_w}_{S_y}) = (m.VREQ + m.VREP) \ . \ \Delta h \tag{5}$$

in which Δh is the average number of hops between PKI nodes. As two messages are needed for each member of issuer group, the cost of validating a group certificate is $O(2m)$ messages.

Finally, the total overhead to authenticate a certificate $C^v_{S_y}$, denoted by $TACO(C^v_{S_y})$, in the worst case, is:

$$TACO(C^v_{S_y}) = ACO(C^v_{S_y}) + k.VCO(C^{IG_w}_{S_y}) \tag{6}$$

in which k is the quantity of non-updated certificates found in the group certificate chains, and necessary to authenticate a public key of group IG_y.

6.2.2 Revocation

If a given node X_i wants to revoke a certificate of a given node X_j, and both are members of group IG_y, X_i must to send a message requesting the revocation of certificate $C^j_{S_y}$ to all all other members of IG_y. Then, it waits for at least t messages accepting the certificate revocation. After, it sends a message informing about the revocation to all other members of IG_y and to all members of groups (IG_b) that issued a certificate to IG_y.

After, members of groups that issued a certificate to IG_y disseminate this message to all nodes that have requested a validation of certificate of IG_y, informing about the presence of a revoked node certificate in this group. Let be L as the list of nodes that requested a validation of certificate of IG_y, so the communication overhead to node X_i revoke the certificate $C^j_{S_y}$, denoted by $RCO(C^j_{S_y})$, is:

$$RCO(C^j_{S_y}) = (3(|IG_y - X_i|) + |IG_y : IG_y \rightarrow IG_b \in G| + |L|) \ . \ \Delta h \tag{7}$$

in which constant 3 represents the three messages exchanged between X_i and the other members of IG_y. The total cost depends on the number of nodes have requested a validation of the certificate of IG_y.

To revoke explicitly a group certificate, a member of certificate issuer group requests to all other members of this group, the certificate revocation. Then, it waits for the reply of at least t nodes accepting the certificate revocation. After, this node sends a message to all other members of its group e to all nodes that requested a validation of this certificate. Let be L as the list of nodes that requested an validation of $C_{S_y}^{IG_w}$, so the communication overhead to revoke a group the certificate $C_{S_y}^{IG_w}$, denoted by $RCO(C_{S_y}^{IG_w})$, is:

$$RCO(C_{S_y}^{IG_w}) = (3(|IG_y - X_i|) + |L|) \ . \ \Delta h \tag{8}$$

As in the node certificate revocation, the total cost depends on the quantity of nodes that had requested validation of the certificate has been revoked.

6.2.3 Update

When a given node X_i wants update its own certificate $C_{S_y}^{i}$, it sends a message to all other members of IG_y, and waits for at least t renewing replies of its certificate. After, it sends the new certificate version to all members of IG_y. Thus, the communication overhead to renew a node certificate $C_{S_y}^{i}$, denoted by $UCO(C_{S_y}^{i})$, is:

$$UCO(C_{S_y}^{i}) = (3|IG_y - x_i|) \ . \ \Delta h \tag{9}$$

For the renewing of group certificate of IG_w ($C_{S_y}^{IG_w}$, node X_i send a message requesting the certificate renewing for all other members of IG_y, and waits for at least t replies renewing the certificate. After, it sends the new version of the certificate to all members of IG_y and IG_w, and to all nodes that requested a validation of certificate has been renewed. Let be L as the list of nodes that requested an updated of $C_{S_y}^{IG_w}$, the overhead communication to update/renew a group certificate $C_{S_y}^{IG_w}$, denoted by $UCO(C_{S_y}^{IG_w})$, is:

$$UCO(C_{S_y}^{IG_w}) = (3|IG_y - X_i| + |IG_z - IG_y| + |L|) \ . \ \Delta h \tag{10}$$

In this case, the communication cost is proportional to the number of nodes that requested the validation of certificate has been revoked. To minimize this cost, node X_i could to send this message only to nodes that requested the validation more recently, or even does not send the renewing message. So, nodes themselves must to verify the certificate validity, when needed.

7 Simulative evaluation

The goals of this Section are present metrics and simulation environment used to evaluate our PKI, and discuss the simulation analyses of our PKI in face of DoS and Sybil attacks.

7.0.4 Metrics

For evaluating GP-PKM, the following metrics were used: *Group Certificate Exchange Convergence* (CE), *Ratio of User Authentication* (UA), *Group Reachability* (GR), *Non-Compromised Group* (NCG) and *Non-Compromised Authentication* (NCA). CE, UA and GR are used to evaluate scenarios under DoS attacks, whereas NCG and NCA are used to evaluate scenarios under Sybil attacks. These metrics are defined, following notations in Table 1, as:

- CE is the average percentage of group certificates in the local repositories of the nodes at time t. It also represents the time needed by all nodes have all issued group certificates in their repositories. The ideal value for this metric is 100%, however some conditions such as the PKI initialization, groups formation, attacks and others can decrease this percentage. CE can be defined as follows:

$$CE(t) = \frac{\sum\limits_{i \in X} CE_i(t)}{|X|} \quad \text{in which} \tag{11}$$

$$CE_i(t) = \frac{\sum\limits_{IG_w, IG_y \in IG} (P_w \to P_y) \in (G_i \ \cup \ G_i^N)}{\sum\limits_{IG_z, IG_x \in IG} (P_z \to P_x) \in G} \tag{12}$$

- UA is the average percentage of user authentications after the convergence time of SG-PKM. This metric is quantified by the certificate chains in updated and non-updated repositories of a node X_i. User authentications are accounted only if two or more disjoint certificate chains are found for authenticating the node. Under attack, this metric will also indicate the survivability of the PKI, evaluating if nodes will be able to authenticate others even in face of DoS attacks. UA can be defined as follows:

$$UA = \frac{\sum\limits_{i \in X} UA_i}{|X|} \quad \text{in which} \tag{13}$$

$$UA_i = \sum_{j \in X} (X_i \rightsquigarrow X_j) \in (G_i \ \cup \ G_j \cup \ G_i^N) \tag{14}$$

- GR is the average percentage of certificate chains for achieving group certificates in the updated and non-updated group repositories of a node X_i at time t. The difference in relation to UA is that here we quantify only group certificates without needing to find two or more disjoint certificate chains for authentication. Let IG_{X_i} as the initiator groups of X_i, so GR can be defined as follows:

$$GR(t) = \frac{\sum\limits_{i \in X} GR_i(t)}{|X|} \quad \text{in which} \tag{15}$$

$$GR_i(t) = \sum_{\substack{IG_w \in IG_{X_i} \\ IG_z \in IG}} (P_w \rightsquigarrow P_z) \in (G_i \ \cup \ G_i^N) \tag{16}$$

- NCG is the percentage of non-compromised groups even in the presence of dishonest nodes in the network. This metric represents the survivability of the PKI against Sybil attacks. Let be IG as the PKI groups set, NCG can be defined as:

$$NCG = \frac{\sum\limits_{IG_w \in IG} NCG_w}{|IG|} \quad \text{in which} \tag{17}$$

$$NCG_w = \begin{cases} 1 & \text{if} \quad \nexists \ f \in IG_w : f \text{ is a false identity} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

- NCA is the percentage of groups that do not have their public key authentication compromised by dishonest nodes. This metric represents the survivability against Sybil attacks of the authentication process. Let be F the set of Sybil nodes, NCA can be defined as follows:

$$NCA = \frac{\sum\limits_{i \in X} NCA_i}{|X|} \quad \text{in which} \tag{19}$$

$$NCA_i = \begin{cases} 1 & \text{if} \quad \nexists \, (P_i \rightsquigarrow P_f) \quad \forall f \in F \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

7.0.5 Environmental setup

We use the Network Simulator(NS) version 2.30 to evaluate the performance and survivability of SG-PKM. Simulations have been done in the presence of DoS and Sybil attacks. To evaluate SG-PKM, a DoS attacker do not collaborate with the PKI services, mainly in the certificate exchange mechanism.

In the simulations, 100 nodes use the IEEE 802.11 with distributed coordination function (DCF) as medium access control protocol. Their radio propagation follows two-ray ground propagation model and the communication range is 50m and 120m. Nodes move on an area of about 1000m x 1000m and 1500m x 300m, following the random waypoint model with a maximal speed of 5 m/s, 10 m/s and 20 m/s, and pause time of 20s. The total time of simulations is 3000s and results are averages of 35 simulations with 95% confidence interval.

Public and private keys are created by nodes only during group formation. Certificates are also issued during group formation and there is no misbehavior detection mechanism in the network. Certificate exchange interval T_{ex} is 60 seconds. These characteristics were implemented in this way for simplicity, not affecting survivability or effectiveness analyses.

According to Table 2, social networks present a great number of cliques with a size equal to 3, 4, 5, and 6. We evaluate SG-PKM varying values of group sizes (m) between 3 and 6. The goal is to verify the impact of the initiator group size in the effectiveness and survivability of SG-PKM. For simplicity, trust relationships are formed following the model proposed by [39].

Table 3 presents a comparison between relevant values founded in PGP graphs and generated graphs. It was considered following parameters: the clustering coefficient, that is the probabity of graph vertices forming a clique, the redundancy between cliques, that is the percentage of neighboors pairs of IG_y connected with each other, and the distance between nodes, that is the average size of relationship chains between two any nodes of PKI. Note that parameters in PGP and generated graphs are similar, that means that used graphs present the expected social behavior.

Tabela 3: Comparison between PGP and generated graphs

Parameters	**PGP graphs**	**Generated graphs**
clustering coeficient	0.030	0.037
redundacy between cliques	0.213	0.282
distance between nodes	3.739	3.726

7.1 Results

Initially, we compare the effectiveness of our PKI scheme by means of the CE metric. Fig. **??** shows results comparing PGP-Like and our PKI with initiator groups with 3, 4, 5 and 6 members, in a scenario without attackers, and in scenarios with 5%, 10% and 20% of misbehavior nodes. In this case,

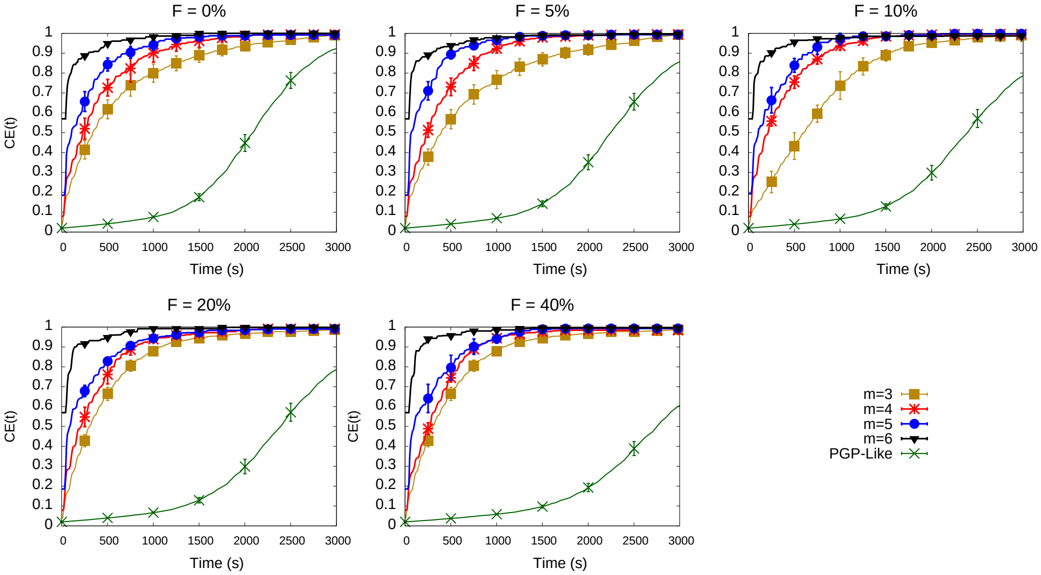

Figura 6: Comparing convergence time of CE under DoS attacks

we consider that a misbehavior node issues certificates and forms groups, but does not cooperate in the certificate exchange mechanism. That is, it requests and stores certificates in its local certificate repository, but does not reply the requests.

In our PKI scheme, the CE reaches 100% before PGP-Like, independently of the groups size and the number of misbehavior nodes. When m is equal to 6, CE reaches 100% approximately after 500 sec. of network lifetime. Already for m equal to 3, 4 and 5, 100% of CE is achieved before 300 sec. of network lifetime. Again, this behavior is independent of the percentage of attackers. Emphazing, higher CE value, higher is the probability of a node to find a path of group certificates its local repository in the the authentication process. However this does not mean that all groups will be reachable or be able to authenticate all other certificate groups, because of the redundancy necessary for authentication.

Fig. 7 presents results for GR in scenarios with 0%, 5%, 10% and 20% of attackers. As expected, we observe that, independtly of percentage of attackers, GR presents same behavior. In our simulations, and with m equal to 3, 4 and 5, GR reaches 100% approximately after 200 sec. of simulation lifetime. Only when m is equal to 6, GR does not reach 100%, but presents values close to 90%. This behavior occurs because of the difficult to form and intersect groups with 6 members.

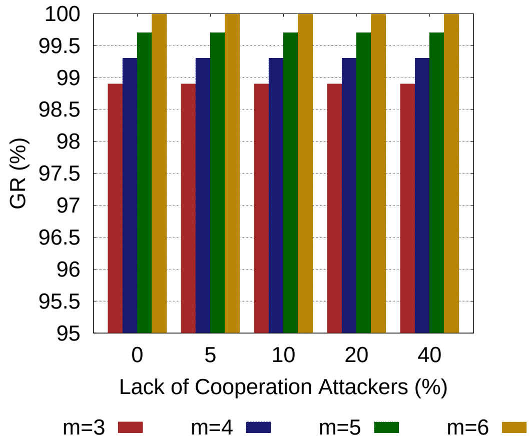

Figura 7: Comparing convergence time of GR under DoS attacks

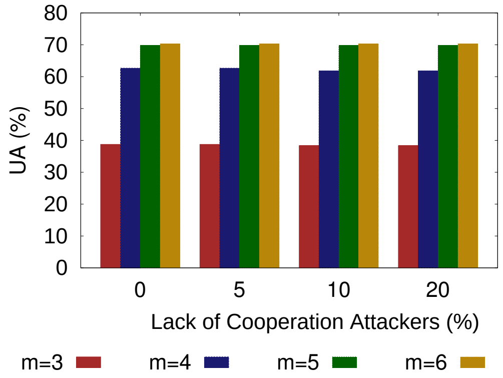

Figura 8: UA under DoS attacks

Fig. 8 compares UA, after convergence time, considering different group sizes and percentage of misbehavior nodes. Results shows that UA present same values indepently of quantity of attackers. We observe the strong influence of initiator group size in the percentage of authentications. We show that while group size increases, the percentage of user authentication also increases. Note that when m is equal to 6 or 5, UA reaches 70% of valid user authentications, while when m is 3, this value is about 40%.

Further, results also show that higher percentage of attacks do not result in a reduction of the UA when compared with the results without attacks. This behavior shows the survivability of our PKI to DoS attacks. Though lower initiator groups present lower UA, no difference is observed between its results under 0% of attack and other percentages.

Fig. 9 and Fig. 10 show through the metrics NCG and NCA the sur-

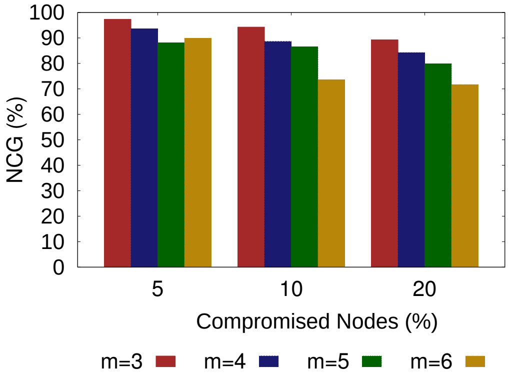

Figura 9: Non-Compromised groups under Sybil attacks

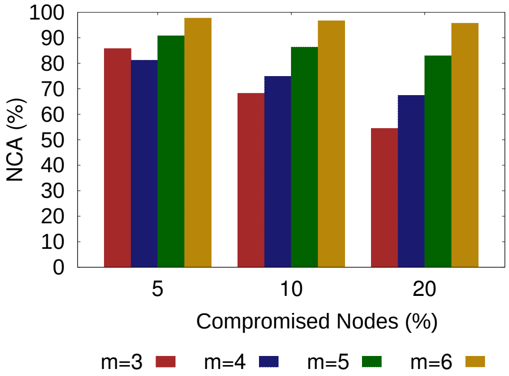

Figura 10: Non-compromised node authentication under Sybil attacks

vivability of our PKI to Sybil or masquerade attacks. In our simulation, malicious nodes create fake nodes or impersonate authentic identities and form groups with them. After, they try to compromise authentic nodes and persuade them to issue certificates to the false groups. The objetive of malicious nodes is to compromised a great number of PKI nodes. If two nodes of a same group are compromised, this group can issue a certificate to the false group. Higher number of compromised groups, higher is the probability of a false identity be authenticate by a valid node.

Fig. 9 shows the survivability of our PKI to the Sybil and masquerade attacks. Results show that with a percentage of 5% of attacker, indepently of the group size, more than 90% of groups are not demaged. When m is 3 this value is close to 99%. When the percentage of attacker is 10% and m is 3, NCG is about 95%. This value decreases a lot when m is equal to 4 and 5, being close to 90%. Only with m is 6, NCG is still about 70%. This occurs because in higher groups the probability of find two or more nodes is higher, then a malicious nodes are able to compromise more groups.

Already when the percentage of atacker is 20%, more groups are driven to issue certificates to a false group, but the results still show the survivability of our PKI. When m is 3 almost 90% of groups are not affected, and whem m is equal to 4 and 5 this value is about 85% and 80%, respectively. Only when m is 6, NCG presents a lower value, about 70%.

Finally, Fig. 10 presents the impact of Sybil attack and the group size to the authentication process. Results show that when $m = 6$, the percentage of valid nodes that do not authenticate false identities is about 98% when PKI is under 5% of attackers. This value is close to 97% with 10% of attackers and higher than 95% when percentage of attackers is 20%.

When our PKI is under 5% of attackers, the percentage of valid nodes that do not authenticate a false identity is higher than 80%. When $m = 5$ this value is about 90%. When the percentage of attackers is 10% and m is 5 or 6, NCA is yet higher than 80%. With m equal to 4 or 3, this value is 74% and 68%, respectively. Already when PKI is under a high number of attackers (20%) the NCA presents a value lower than 70%, to m equal to 3 or 4. But with $m = 5$ this value is yet higher than 80% and with $m = 6$ it is about 95%.

8 Conclusion

This work presented a survivable PKI for WANETs. Its goal is to make public key management system able to provide its services even in face of attacks or intrusions. Our PKI is based on the coordinated integration among preventive, reactive and tolerant defense lines, being self-adapted to different physical network conditions. It attains the survivability properties by different mechanisms such as the employment of different evidences to prove the liability of users for their public keys, the formation of initiator groups based on social relationships, and the use of redundancy in many PKI operations.

Simulation results showed the survivability of our PKI under high percentage of attacks and also its resistance against Sybil attacks. Results presented relevant effectiveness of our proposal taking only few minutes to achieve the maximum convergence of all certificates into all nodes of the system. As future works, we plan to evaluate performance aspects in relation to the physical network and to also quantify communication costs. If necessary, we will propose mechanisms to minimize communication costs in group maintenance.

Referências

[1] C. Zhang, Y. Song, and Y. Fang, “Modeling secure connectivity of self-organized wireless ad hoc networks,” in *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’08).* Los Alamitos, CA, USA: IEEE Communications Society, 2008, pp. 251–255.

[2] J. van der Merwe, D. Dawoud, and S. McDonald, “A survey on peer-to-peer key management for mobile ad hoc networks,” *ACM Computing Surveys*, vol. 39, no. 1, pp. 1–45, 2007.

[3] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, “Security in mobile ad hoc networks: challenges and solutions,” *IEEE Wireless Communications*, pp. 38–47, Feb 2004.

[4] D. Djenouri, L. Khelladi, and N. Badache, “A survey of security issues in mobile ad hoc and sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 7, no. 4, pp. 2–28, 2005.

[5] M. Raya and J.-P. Hubaux, “The security of vehicular ad hoc networks,” in *Proceedings of the 3rd ACM Workshop on Security of Ad hoc and Sensor Networks SASN ’05.* New York, NY, USA: ACM, 2005, pp. 11–21.

[6] L. Hoffman, “In search of dependable design,” *Communications of the ACM*, vol. 51, no. 7, pp. 14–16, 2008.

[7] M. N. Lima, G. Pujolle, E. Silva, A. L. Santos, and L. C. P. Albini, “Survivable keying for wireless ad hoc networks,” in *Proceedings of the 2009 IFIP/IEEE International Symposium on Integrated Network Management (IM ’09).* New York, NY, USA: IEEE Communications Society, Jun 2009, pp. 606–613.

[8] L. Zhou and Z. J. Haas, “Securing ad hoc networks,” *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[9] S. Capkun, L. Buttyan, and J.-P. Hubaux, “Self-organized public-key management for mobile ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64, 2003.

[10] A. Khalili, J. Katz, and W. A. Arbaugh, “Toward secure key distribution in truly ad-hoc networks,” in *Proceedings of the 2003 Symposium*

on Applications and the Internet Workshops (SAINT 2003 Workshops). Washington, DC, USA: IEEE Computer Society, 2003, p. 342.

[11] J.-P. Hubaux, L. Buttyán, and S. Čapkun, "The quest for security in mobile ad hoc networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & computing (MobiHoc 2001)*, 2001, pp. 146–155.

[12] S. Čapkun, L. Buttyán, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64, 2003.

[13] S. Čapkun, J.-P. Hubaux, and L. Buttyán, "Mobility helps peer-to-peer security," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 43–51, 2006.

[14] E. C. H. Ngai and M. R. Lyu, "Trust- and clustering-based authentication services in mobile ad hoc networks," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW 2004).* Washington, DC, USA: IEEE Computer Society, 2004, pp. 582–587.

[15] E. C. H. Ngai, M. R. Lyu, and R. T. Chin, "An authentication service against dishonest users in mobile ad hoc networks," in *Aerospace Conference 2004*, vol. 02. Big Sky, MT: IEEE, mar 2004, pp. 1275–1285.

[16] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and communications security (CCS 2002).* New York, NY, USA: ACM Press, 2002, pp. 41–47.

[17] S. Čapkun, J.-P. Hubaux, and L. Buttyán, "Mobility helps security in ad hoc networks," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing.* New York, NY, USA: ACM Press, 2003, pp. 46–56.

[18] P. R. Zimmermann, *The official PGP user's guide.* Cambridge, MA, USA: MIT Press, 1995.

[19] B. Christianson, "Why isn't trust transitive," in *Proceedings of the International Workshop on Security Protocols (WSP 1996).* IEEE Computer Society, 1996.

[20] E. Silva, A. L. dos Santos, L. C. P. Albini, and M. N. Lima, "Quantify misbehavior attacks against the self-organized public key management on manets," in *Proceedings of the International Conference on Security and Cryptography (SECRYPT 2008)*, 2008, pp. 128–135.

[21] J. Salido, L. Lazos, and R. Poovendran, "Energy and bandwidth-efficient key distribution in wireless ad hoc networks: a cross-layer approach," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1527–1540, 2007.

[22] B. Wu, J. Wu, E. B. Fernandez, M. Ilyas, and S. Magliveras, "Secure and efficient key management in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 30, no. 3, pp. 937–954, 2007.

[23] M. Chorzempa, J.-M. Park, and M. Eltoweissy, "Key management for long-lived sensor networks in hostile environments," *Computer Communication*, vol. 30, no. 9, pp. 1964–1979, 2007.

[24] M. N. Lima, H. W. da Silva, A. L. dos Santos, and G. Pujolle, "An architecture for survivable mesh networking," in *Proceedings of the 2008 IEEE Global Communications Conference (GLOBECOM '08)*. Los Alamitos, CA, USA: IEEE Communications Society, 2008, pp. 688–692.

[25] "Keyanalyze - analysis of a large OpenPGP ring," 2008, access: August 2008. [Online]. Available: http://dtype.org/keyanalyze/

[26] L. Feeney, B. Ahlgren, and A. Westerlund, "Spontaneous networking: an application oriented approach to ad hoc networking," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 176–181, June 2001.

[27] J. Wu and D. J. Watts, "Small worlds: the dynamics of networks between order and randomness," *ACM SIGMOD Record*, vol. 31, no. 4, pp. 74–75, 2002.

[28] L. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas, "Characterization of complex networks: A survey of measurements," *Advances In Physics*, vol. 56, pp. 167–242, 2007.

[29] A. N. Bessani, P. Sousa, M. Correia, N. F. Neves, and P. Verissimo, "The crutial way of critical infrastructure protection," *IEEE Security & Privacy*, vol. 6, no. 6, pp. 44–51, 2008.

[30] U. Maurer, "New approaches to digital evidence," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 933–947, Jun. 2004.

[31] ——, "Modelling a public-key infrastructure," in *Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS '96).* London, UK: Springer-Verlag, 1996, pp. 325–350.

[32] T. P. Pedersen, "A threshold cryptosystem without a trusted party," in *Proceedings of Advances in Cryptology (EuroCrypt '91)*, ser. Lecture Notes in Computer Science, vol. 547. London, UK: Springer, 1991, pp. 522–526.

[33] T.-Y. Chang, C.-C. Yang, and M.-S. Hwang, "A threshold signature scheme for group communications without a shared distribution center," *Future Generation Computer System*, vol. 20, no. 6, pp. 1013–1021, 2004.

[34] H. Ghodosi and R. Safavi-naini, "Dynamic threshold cryptosystems: A new scheme in group oriented cryptography," in *Proceedings of the International Conference on the 1st Theory and Applications of Cryptology (PRAGOCRYPT '96).* Prague, Czech: Czech Technical University Publishing House, 1996, pp. 370–379.

[35] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[36] S. Čapkun, L. Buttyán, and J.-P. Hubaux, "Small worlds in security systems: an analysis of the PGP certificate graph," in *Proceedings of the 2002 Workshop on New Security Paradigms (NSPW '02).* New York, NY, USA: ACM, 2002, pp. 28–35.

[37] M. Latapy, C. Magnien, and N. D. Vecchio, "Basic notions for the analysis of large two-mode networks," *Social Networks*, vol. 30, no. 1, pp. 31–48, 2008.

[38] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, "A new algorithm for generating all the maximal independent sets," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 505–517, 1977.

[39] F. Viger and M. Latapy, "Efficient and simple generation of random simple connected graphs with prescribed degree sequence," in *Proceedings of 11th Annual International Conference of Computing and Combinatorics (COCOON 2005)*, ser. LNCS, vol. 3595. Springer, 2005, pp. 440–449.