Secure and reliable certificate chains recovery protocol for mobile ad-hoc networks

Mawloud Omar, Hamida Boufaghes, Lydia Mammeri, Amel Taalba, Abdelkamel Tari

Secure and reliable certificate chains recovery protocol for mobile ad-hoc networks

Mawloud Omar, Hamida Boufaghes, Lydia Mammeri, Amel Taalba and Abdelkamel Tari

Laboratoire d'Informatique Médicale, Faculté des Sciences Exactes
Université de Bejaia, 06000 Bejaia, Algérie.

Abstract

The deployment of any security policy requires the definition of a trust model that defines who trusts who and how. There is a host of research efforts in the trustworthy area to securing mobile ad-hoc networks. Among the most used approaches are based on public-key certificates and gave birth to miscellaneous trust models ranging from centralized models to web-of-trust and distributed certification authorities. Certificates management in mobile ad-hoc networks is a veritable challenge because of constrictions imposed by the nature of the network. The freedom of nodes mobility involves some constraints when designing reliable certification systems. In this paper, we address this issue and we propose a secure and reliable certificate chains recovery protocol for mobile ad-hoc networks. Our proposal is based on web-of-trust in which the users ensure themselves the role of the certification service by issuing and managing the public-key certificates. The shortest and the safest certificate chains are selected in order to reduce the communication overhead and resist against compromised nodes which can generate false certificates. An analytical model is developed and simulations are performed in order evaluate the performances of our protocol, in which it demonstrates interesting results.

Keywords: Mobile ad-hoc network, certificate, public-key, web-of-trust

1. Introduction

Mobile ad-hoc networking [2] is one of the most important areas in the field of wireless communication. The premise of forming a mobile ad-hoc

network is to provide wireless communication among mobile devices anytime and anywhere with no infrastructure. These devices, such as cell phones, laptops, etc. carry out communication with other nodes that come in their radio range of connectivity. Each participating node provides services such as message forwarding, providing routing information, authentication, etc. with other nodes spread over an area. They are mostly employed in the military applications where their mobility is attractive, but have also a high potential for use in civilian applications such as coordinating rescue operations in the infrastructure-less areas, sharing content and network gaming in the intelligent transportation systems, surveillance and control in wireless sensor networks, etc.

The inherent vulnerability of mobile ad-hoc networks introduces new security problems, which are generally more prone to physical security threats. The possibility of eavesdropping, spoofing, denial-of-service and impersonation attacks increase. Similar to the fixed networks, security of mobile ad-hoc networks is considered from different points such as availability, confidentiality, integrity, authentication, non-repudiation and access control. However, the security approaches used to protect the fixed networks are not feasible due to the salient characteristics of mobile ad-hoc networks. New threats, such as attacks raised from internal malicious nodes are hard to defend. The deployment of any security service requires the definition of a trust model that defines who trusts who and how. There are research efforts in the trust model framework for securing mobile ad-hoc networks. In this paper, we focus on the category of certification-based trust models. The trust relationship among users is performed in a transitive manner, such that if A trusts B and B trusts C, then A can trust C. In this relationship, the principal B is called "trusted third party". The latter could be a central authority (like CA - Certification Authority) or a simple intermediate user in the case of web-of-trust based models [6].

The certificate chain recovery based on the web-of-trust in highly dynamic networks, such as mobile ad-hoc networks, involves two major challenges representing the main topic covered in this paper. The first problem is related to the credibility degree, which we should assign when evaluating the trust chains. This problem is specific to the web-of-trust based models, in which the users themselves, without being controlled by a central authority, establish the trust relationship propagation autonomously and in a transitive way. The transitive trust spread can cover compromised intermediate nodes, which could compromise the network security. The second problem is related

to the certificate collection service availability, which is a specific problem to the imposed constraints by mobile ad-hoc networks. Centralize certificate repositories on the servers could compromise the access availability, where the network is a subject of partitioning because of the mobility of nodes. Designing a distributed certificate collection protocol is crucial in order to overcome the availability problem, however, it opens for other issues relating to the capacity of nodes, which are constrained in terms of resources. Thus, three main criteria must be optimized, namely the computation load, the storage and the transmission.

In response to the challenges described above, we contribute through this work with a secure and reliable certificate recovery approach. The security aspect is addressed to answer the first challenge and we propose a trust chain selection mechanism based on the nodes credibility. The credibility of a node increases proportionally to the certificate number issued for that node. The proposed mechanism maximizes this criterion when selecting a trust chain in order to avoid compromised nodes to be considered. Therefore, we maximize the probability of successful signature verification of the certificate chain, and thus, decreasing the probability of reiterating the collection and verification process for another alternative chain. The reliability is addressed to answer the second challenge and we propose a negotiation protocol in order to collect information about "who trusts who". A prior analysis relating to the trust chain length is performed before executing the certificate collection process. A high chain length involves a considerable communication overhead when collecting and an expensive computation load when verifying the signatures. The storage is reduced by keeping at each node only the certificates signed by or for the node in question. The set of required certificates is collected in a distributed manner when the authentication is required between two nodes.

Our proposal allows nodes to generate, store and distribute their public-key certificates without any central server or trusted party. All the nodes have a similar role and we do not assign special functions to specific nodes. The main motivation for employing this approach comes from the self-organized nature of mobile ad-hoc networks and from the need to allow users to fully control the security settings in the network. Users public and private keys are created by the users themselves and key authentication is performed via chains of public-key certificates following the web-of-trust. Instead of storing certificates in centralized certificate repositories, certificates are stored and distributed by nodes themselves. The performance evaluation is done through both analytical modeling and simulations with comparison to con-

current approaches. The evaluated metrics are the certification success probability, the response time and the communication overhead. The certification success probability evaluates the safety metric of our protocol and it is performed through an analytical modeling using Markov chains. The response time and communication overhead evaluate the metric of the certificate chain length and are performed through simulations.

The remaining of this paper is structured as follows. In Section 2, we introduce the related work and we give a general presentation of our contributions. In Section 3, we give detailed description of our protocol. In Section 4, we present and discuss the results of performances evaluation in which we have developed both analytical model and simulations. We finally conclude this work in Section 5.

2. Related work

In the web-of-trust based models, there is no central authority. Each user acts as a certification authority independently of the other users in the network. This model is decentralized in nature and so is very adequate for mobile ad-hoc networks. In this section we survey the most relevant web-of-trust based public-key certification protocols for mobile ad-hoc networks, which are classified into two categories: proactive and reactive protocols. In the remaining of this section, we give descriptions of the protocols belonging to each category, an overall analysis and we summarize our contributions.

2.1. Proactive protocols

In this category of protocols, the process of certificate collection is executed systematically among neighboring nodes. Thus, when the node needs to verify a certificate, it is done instantly since the required chain of certificates could have been already retrieved from the network. Capkun et al. [18] have proposed a fully and self-organized protocol, which requires no central authority. Each node in the network holds a certificate repository and the certificate collection process is executed in a proactive manner, in which the certificates are exchanged among neighbor nodes at each direct contact. When a node needs to authenticate the public-key of an another node, both nodes merge their local repositories and try to find a certificate chain from the one to the other. Ren et al. [17] have proposed a modified version of the protocol of Capkun et al. by introducing a boot server to initialize the system. The boot server computes and distributes to each node a short list

with a set of bindings (nodes identifiers and public-keys). Then, each node stores it locally and generates the corresponding certificates. Thus, a web-of-trust is formed and the system becomes fully distributed, where the nodes authenticate themselves through certificate chains. Omar et al. [11] have introduced a threshold scheme within the web-of-trust. During the network initialization, nodes share the system private-key and each node holds one private-share. Instead of using the private-key for certificate signing, a node uses its private-share. Each node in the network maintains a partial view of the web-of-trust, which is updated systematically through partial certificate exchanging protocol among neighboring nodes. The public-key authentication among nodes is performed via the combination of the partial certificate chains.

2.2. Reactive protocols

In this category of protocols, the certificate collection process is executed on-demand. When the node needs to verify a public-key, it collects the appropriate chain of certificates in a distributed manner from the network. Funabiki et al. [12] have proposed a centralized protocol based on clustering. The certificate issuance is ensured by the nodes themselves, however they are stored at a particular node named CMN (Certificate Management Node) in each cluster. All the cluster nodes should request the CMN to collect the required certificates in order to verify the trust chain. Kitada et al. [14][15] have proposed a distributed protocol, where each node holds a local repository that contains the node's certificates signed by some other nodes and certificates delivered by the node itself for the other nodes. When a node needs to verify the public-key of an another node, it broadcasts a search request to the nodes that it directly trusts. Each intermediate node includes its own certificate in the request. Finally, the destination node adds its own certificate and sends to the source node the certificate chain. Hisham et al. [10] have proposed a modified version of the protocol of Kitada et al., in which upon receiving the different certificate chains, the source node proceeds to verify the shortest chain in order to minimize the computation overhead. Kambourakis et al. [9] have considered that the web-of-trust has the form of a binary tree. Hence, in order to respect the tree structure, each node in the network is certified by only one of its neighboring nodes and it certifies at the maximum to two of its neighboring nodes. Xia et al. [16] have proposed a certification protocol executed by the nodes themselves. Each node that first joins the network performs a proactive process by broadcasting its public-

key certificate as a request to its neighbors. With the designed neighborhood certificate distribution mechanism, each node is able to obtain all the updated public-key certificates from the neighbors within its two hops distance. For those nodes that are more than two hops away from each other, a multi-hop public-key certificate distribution is executed on-demand. Suguna et al. [7] have proposed a certification protocol, where each node generates its public-key and the corresponding private-key locally before joining the network. Public-key certificates are issued by the nodes based on nodes information about the other nodes in the network. Issued certificates to or by the user are stored in its certificate repository with a validity time. The authors of this protocol have added a stable link that gives the time of the establishment of a link between two given nodes and its failure. This value depends on the distance between two nodes, their rates of mobility and directions.

2.3. Overall analysis and our contributions

The availability of the certification service depends on the ability of each node to collect any certificate chain relating to any other node in the network. This property depends on the manner of managing the certificate repositories. In proactive protocols, each node maintains a local repository, updated systematically through a protocol of certificate exchange. This category of protocols can achieve a good level of availability since the retrieval of certificates is done locally on the node itself. However, the storage and communication overhead generated is very important and considered as the major weakness. Moreover, each node maintains an important number of certificates despite even if they are not used. In other hand, the scalability of the certification service depends strongly on the number of certificates to store at each node. The certification service may be not scalable in the category of proactive protocols due to the number of certificates which should be proportional to the network size.

We consider that the reactive protocols are the most appropriate for mobile ad-hoc networks, such no extra certificates are stored and/or managed by the nodes. Each node maintains a limited number of certificates that concern only the node itself either as the certificate issuer or holder. The certificate collection is performed once a node requires to authenticate the public-key of its interlocutor. When a node needs to verify a public-key, it collects the appropriate chain of certificates in a distributed manner. In the majority of the proposed solutions, each intermediate node embeds its own certificate

when forwarding the certification packet based mainly on its trusty neighborhood (the nodes which are directly trusted). This view, being local and very limited, decreases thoroughly the quality of the global chain. Moreover, an overhead considerable is generated following this process. In this paper, we address this issue and we propose an secure and reliable certificate chains recovery protocol. The contribution of this paper is double:

1. In design point of view, instead of each intermediate node embeds its own certificate, it includes only its own identity. Since the process of certificate transferring is expensive, we propose to reserve the process of certificate collecting once the choice is made for the most optimal chain. Therefore, upon receiving all the trust chains, the source node can select efficiently the optimal one.
2. As soon as the source node receives all the possible trust chains, it selects the most optimal one by combining two important criteria: the shortest and safest chain. The shortest chain reduces the communication overhead when collecting the certificates in the final step and also reduces the computation overhead when verifying the certificates signatures. The safest chain reduces the risk to collect false certificates, which can be generated by compromised nodes in the network.

3. Our protocol

In this section, firstly we model the considered environment and we state the problem to be solved. Then, we give detailed descriptions of the elements composing our protocol. The security analysis of our protocol against possible threats is given as soon as we describe each step.

3.1. Network model and problem statement

We consider a mobile ad-hoc network, where a set of mobile nodes are randomly located in a region and move with a random mobility pattern, i.e., nodes moving independently. To each mobile node is assigned a unique identifier. Similar to the most previous works, we assume that the traffic in the network is light and hence the communication channel is assumed to be error free and provides reliable data delivery. The network topology changes frequently because the mobility of nodes. The set of nodes in the $\mathtt{i}$'s vicinity consists of the $\mathtt{i}$'s neighbor nodes. We assume that the nodes have the same transmission power range and hence the communication links

between neighbor nodes are bidirectional. The communication among nodes is performed via multi-hop routing, and we consider that a source node could reach its interlocutor through a single and optimal routing path established by the implemented routing protocol.

We consider a web-of-trust, offline established between the users of the network. We assume that each user owns a single mobile, and hence, we will use the same identifier for the user and her node. Each mobile node holds a certificate repository, which stocks a part of the global web-of-trust including only the certificates issued to or by the corresponding user. We consider a source node and a destination node in which the source node needs to authenticate the destination node's public-key. The main issue addressed in this paper is how to find efficiently a chain of certificates from the source node to the destination, and how to collect all required certificates in the chain in order to carry out the necessary verifications? For any correct node, our main goal is to provide flexibility of the certification service to tune the protocol performance online to the trade-off between the safety and communication overhead when recovering and when collecting the required certificates.

The notations used throughout all subsequent sections are summarized in Table 1.

Term	Description
(K_i, K_i^{-1})	Respectively, the node i's public and private key
$\big(M\big)_{K_i}$	Message M encrypted by the node i's public-key
$\big(M\big)_{K_i^{-1}}$	Message M signed by the node i's private-key
$\mathcal{C}rt_i^{(j)}$	Public-key certificate issued by the node j to the node i
Θ_i	The node i's local certificates repository
ϑ_i	A cryptographic stamp signed by the node i
TC	Trust Chain, which is a sorted list including an ensemble of nodes identifiers
$\partial(V, E)$	Partial web-of-trust in which V is a set of node identifiers and E is the trust relationship between each pair of nodes
d_i^-	The number of certificates issued for the node i
Ω_{TC}	The safety degree of the trust chain TC

Table 1: Notations

3.2. Trust model and overview

Our protocol does not need any trusted third party and the certification service is assured by the nodes themselves. All the nodes have a similar role and we do not assign particular functions to specific nodes. The public-key and the corresponding private-key of each node are created locally by the node itself before joining the network. Public-key certificates are issued based on the nodes information about the other nodes in the network. If a node $\mathtt{i}$ believes that the public-key $\mathsf{K}_\mathtt{j}$ belongs to the node $\mathtt{j}$, thus the node $\mathtt{i}$ issues a public-key certificate to $\mathtt{j}$. We denote $\mathcal{C}\mathtt{rt}_\mathtt{j}^{(\mathtt{i})}$ a certificate signed by the node $\mathtt{i}$'s private-key $\mathsf{K}_\mathtt{i}^{-1}$ to represent its assurance in the binding of the node $\mathtt{j}$ and its public-key $\mathsf{K}_\mathtt{j}$, such as $\mathcal{C}\mathtt{rt}_\mathtt{j}^{(\mathtt{i})} = (\mathtt{j}, \mathsf{K}_\mathtt{j})_{\mathsf{K}_\mathtt{i}^{-1}}$. We term the node $\mathtt{i}$ by the $\mathtt{j}$'s ascending trusted node and the node $\mathtt{j}$ by the $\mathtt{i}$'s descending trusted node. We assume that there exist sparse trust relationships among nodes and any node that wishes to join the network can establish independent trust relationship with some of existing member nodes in the network.

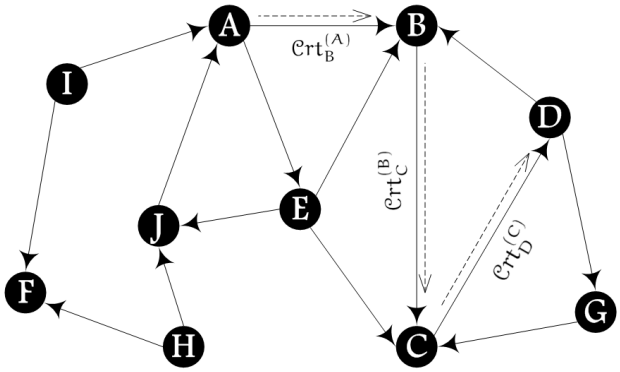

Figure 1: An example of a web-of-trust

Our protocol operates under a trust model basing on the existence of the certificates as bindings of public-keys and the corresponding user identities. The trust propagation among nodes forms a web-of-trust allowing each node to authenticate the public-key of any node in the network through a trust chain (denoted by TC). In Figure 1, we illustrate an example of a web-of-trust, in which the node A wishes to authenticate the public-key of the node D. Therefore, the node A should acquire a chain of valid certificates from the node A to the node D. For example, let's consider the certificate chain:

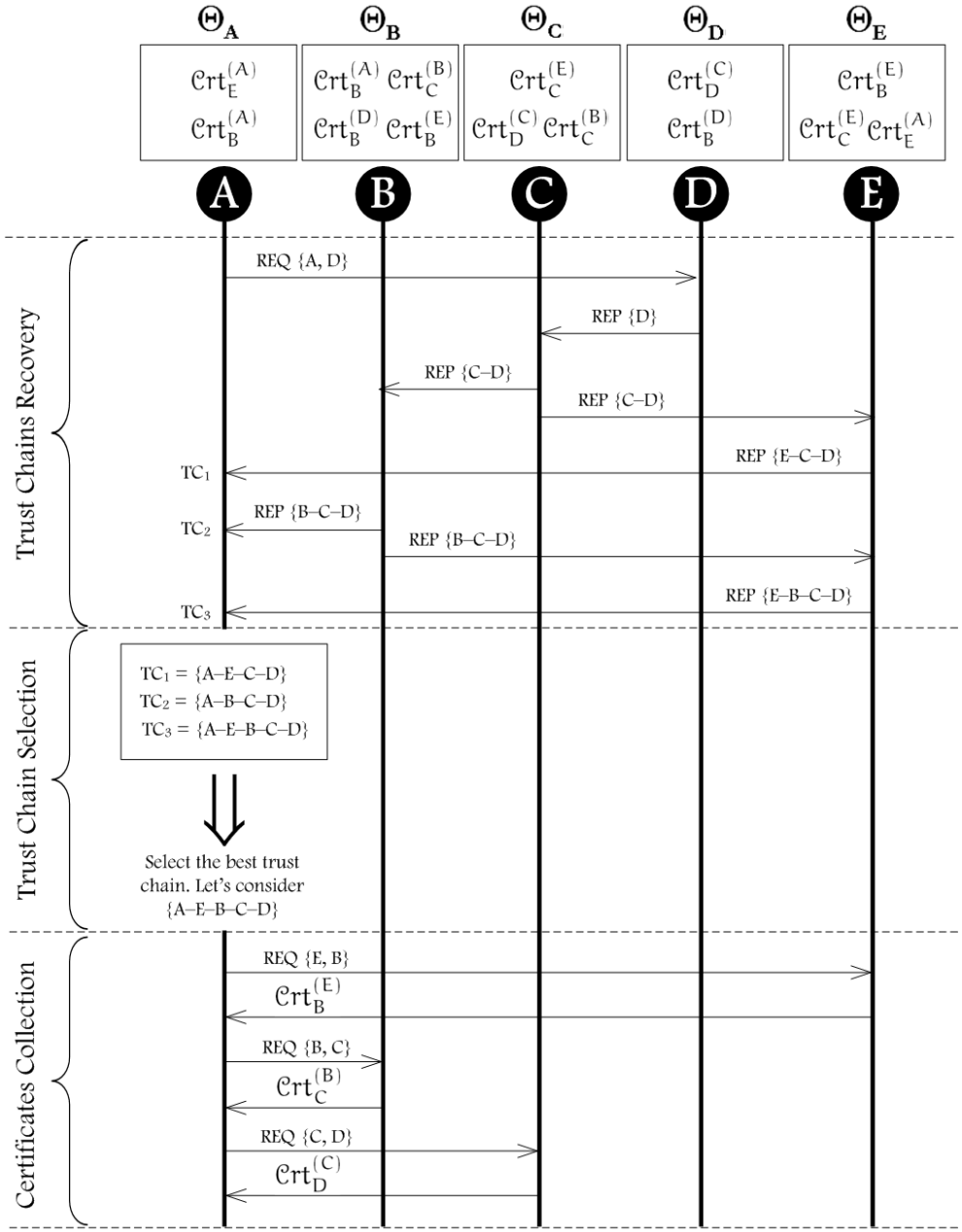

Figure 2: Overview of our protocol

$\mathcal{C}\mathtt{rt}_{\mathsf{B}}^{(\mathsf{A})}$, $\mathcal{C}\mathtt{rt}_{\mathsf{C}}^{(\mathsf{B})}$, $\mathcal{C}\mathtt{rt}_{\mathsf{D}}^{(\mathsf{C})}$. The first certificate of the chain (which is issued by the node A) will be verified by itself using its own public-key K_{A}. Each remaining certificate in the chain will be verified using the public-key figuring on the previous certificate. The last certificate in the chain holds the D's public-key.

The web-of-trust certificates are distributed among all the network nodes, in which each node $\mathfrak{i}$ maintains a part $\Theta_{\mathfrak{i}}$. Therefore, in order to collect the required chain of certificates of any pair of nodes in the network, we propose a distributed protocol executed in three phases. The first step consists of the trust chains recovery, in which the node proceeds only to collect information about the existing trust chains leading to its interlocutor. This process is executed under a distributed protocol through the intermediate nodes by exchanging only information related to "who trusts who" without certificate exchanging. With this manner, our protocol gains in performance in terms of communication overhead. The second step consists of the trust chain selection in which the node proceeds to select the most optimal trust chain among all those received. The final step consists of certificate collection, in which the node proceeds to collect and verify the validity of certificates related to the selected trust chain. In Figure 2, we illustrate an example of the protocol execution process. This example is based on the web-of-trust, illustrated in Figure 1, in which the node A wishes to authenticate the public-key of the node D. Upon receiving the request, the node D sends its identity toward its all ascending trusted nodes and each one of them adds its identity and forward it to its ascending trusted nodes. This process is repeated until to reach the node A. Therefore, the latter receives the trust chains: $\big\{\{\mathsf{A},\mathsf{E},\mathsf{C},\mathsf{D}\},\{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\},\{\mathsf{A},\mathsf{E},\mathsf{B},\mathsf{C},\mathsf{D}\}\big\}$. It selects the optimal trust chain and collects the concerned certificates using a unicast-based communication. For example, if the trust chain $\{\mathsf{A},\mathsf{E},\mathsf{B},\mathsf{C},\mathsf{D}\}$ is selected, the node A sends individual requests to the nodes E, B and C demanding respectively the certificates $\mathcal{C}\mathtt{rt}_{\mathsf{B}}^{(\mathsf{E})}$, $\mathcal{C}\mathtt{rt}_{\mathsf{C}}^{(\mathsf{B})}$ and $\mathcal{C}\mathtt{rt}_{\mathsf{D}}^{(\mathsf{C})}$. We note that the certificate which covers the E's public-key is already stored in the local repository of the node A.

3.3. Trust chains recovery

The main objective of this phase is to gather information about existing certificate chains from the source node S to the destination D. The source node S sends through a unicast-based communication a trust chain request packet $\mathtt{ReqTC}$ (Request for the Trust Chains) to the destination node D including a cryptographic stamp ϑ. A compromised node may decline to forward the request packet. The source node can detect such an attack by

setting a timer for the request and can change the route should the timer expire. A compromised node may also send fake requests in order to launch denial of service attack. Our proposal does not address directly this issue, however there is a host of research area in the literature treating this aspect [5]. Detection involves locating an attacker and taking appropriate actions as an exclusion. The monitoring nodes activity or tracing an attacker can help in detecting a denial of service attack source. Several mechanisms have been proposed to defense against such attack [4][13]. The cryptographic stamp is generated by the source node, which is signed end-to-end by the intermediate nodes when forwarding the reply packet. The destination node D responds to the request with a trust chain reply packet RepTC (Reply by the Trust Chains). The reply packet includes $\vartheta_{\mathsf{D}} = \{\vartheta\}_{\mathsf{K}_{\mathsf{D}}^{-1}}$ and $\mathsf{TC} = \mathsf{D}$ sent to all ascending trusted nodes of D. Each intermediate node i upon receiving RepTC froms its predecessor node k, it calculates ϑ_{i} by signing the received value of ϑ_{k} with its own private-key, it adds its own identifier to TC and forwards the reply packet to its all ascending trusted nodes:

$$\mathsf{i} \longrightarrow \mathsf{j} \;/\; \exists\, \mathcal{C}\mathsf{rt}_{\mathsf{i}}^{(\mathsf{j})} \in \Theta_{\mathsf{i}} : \mathsf{RepTC} = \left(\mathsf{S}, \mathsf{D}, (\vartheta_{\mathsf{k}})_{\mathsf{K}_{\mathsf{i}}^{-1}}, \{\mathsf{i}\} \cup \mathsf{TC}\right)_{\mathsf{K}_{\mathsf{i}}^{-1}}$$

This process is repeated until the packet RepTC reaches the source node S. Note that upon receiving the packet RepTC, the node j verifies if $\mathcal{C}\mathsf{rt}_{\mathsf{i}}^{(\mathsf{j})} \in \Theta_{\mathsf{j}}$ (in the normal case if $\mathcal{C}\mathsf{rt}_{\mathsf{i}}^{(\mathsf{j})} \in \Theta_{\mathsf{i}}$, then $\mathcal{C}\mathsf{rt}_{\mathsf{i}}^{(\mathsf{j})} \in \Theta_{\mathsf{j}}$). If the certificate exists, then it forwards the reply packet to its successors. Otherwise, an attack can be suspected, in which a compromised node tries to inject a fictitious reply packet. In this case, the node j should drop the received reply packet. Indeed, our protocol can suffer from such denial of service attacks launched by nodes injecting certain fictitious to cram the service of certification. Preventing such attacks is difficult because an intermediate node should anyway verify if the certificate exists in its local repository before knowing its legitimacy. Compromised nodes, when behaving as described above, cannot compromise our protocol. However, owning the identities and certificates of once legitimate nodes, they pose potential threats to the other nodes that still believe in their legitimacy. Our protocol does not directly thwart these threats; it rather relies on inputs from misbehavior detection algorithms (e.g., the reputation systems [1][3][8]) to identify and thus evicts these nodes. A compromised node may also decline to forward the reply packet. Such attack does not compromise the availability of our protocol because the destination

node responds to the trust chain request packet by a reply packed destined to its all ascending trusted nodes. Thus, if the altered trust chain information fails to be received that does not prevent the other chains to be received by the source node.

At the end of the process of trust chains recovery, the source node will receive an ensemble of trust chains $\mathsf{TC}_1, \mathsf{TC}_2, ..., \mathsf{TC}_N$. A trust chain TC is an ensemble of sorted node identifiers, where the TC's ends are the source and destination nodes, such as:

$$\mathsf{TC} = \left\{ S = \mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_{|\mathsf{TC}|-1}, D = \mathrm{ID}_{|\mathsf{TC}|} \right\}$$

We denote by $\mathsf{TC}^{(k)}$ the k^{th} node in the trust chain TC. Following the latter, the source node S trusts the node which has the identity ID_2, the latter trusts the node which has the identity ID_3 and so on until to reach the destination node D. Thus, we note in the general case that following TC, the node $\mathsf{TC}^{(k)}$ trusts the node $\mathsf{TC}^{(k+1)}$. A trust chain TC corresponds to an available certificate chain allowing the source node S to reach the node D. Therefore, the number of certificates to be collected equals to $|\mathsf{TC}|-1$. Upon receiving all the trust chains, the source node constructs locally the partial web-of-trust ∂ inspired from the received TC_l. The partial web-of-trust ∂ is a directed graph, consists of a set of vertices V and a set of edges E. V represents the set of node identifiers belonging to the received trust chains, such as:

$$V = \{ i : \ i \in \bigcup_{l=1}^{N} \mathsf{TC}_l \}$$

and E represents the existing trust relationship between each pair of nodes belonging to all the received trust chains TC_l, such as:

$$E = \{ (i,j) : \ \exists \ \mathsf{TC}_l \ / \ (\mathsf{TC}_l^{(k)}, \mathsf{TC}_l^{(k+1)}) = (i,j) \}$$

In Figure 3, we illustrate an example which summarizes the steps of the partial web-of-trust construction. Following the global web-of-trust, upon receiving the reply packet, each intermediate node includes its own identity and forwards it to its ascending trusted nodes until to reach the source S (cf. Figure 3 (a) and (b)). The source node receives an ensemble of trust chains allowing it to reach the destination node (cf. Figure 3 (c)). Finally, it combines all the received trust chains and builds-up the partial web-of-trust

∂ illustrated in Figure 3 (d). Note that ∂ represents a view part of the global web-of-trust, which is used in order to get information about the existing certificates and to decide about the optimal chain before to proceed to the process of certificate transferring. The process of the optimal trust chain selection is described in the following subsection.

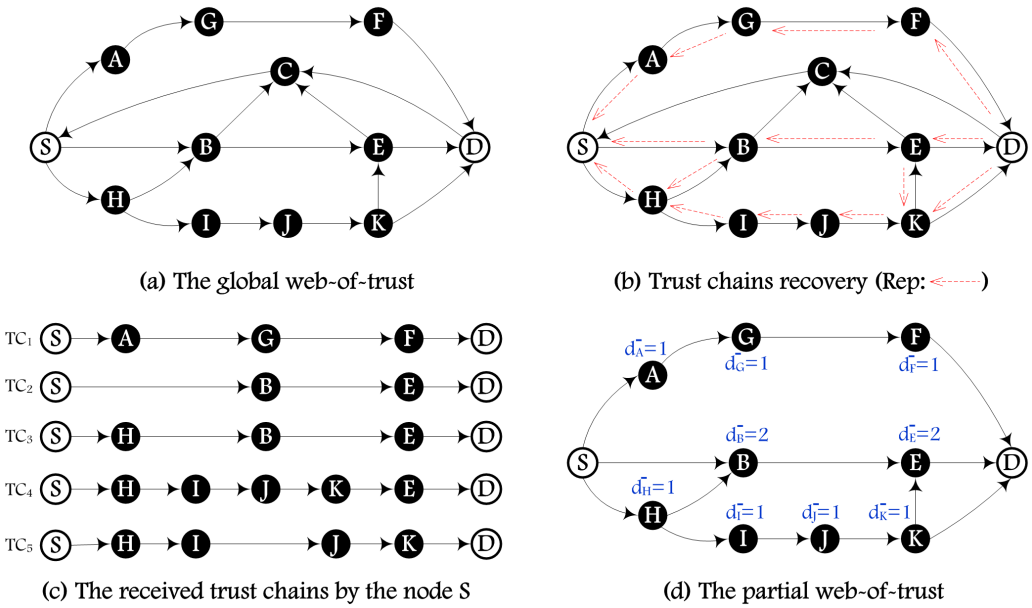

Figure 3: Partial web-of-trust construction

3.4. Trust chain selection

From the partial web-of-trust ∂, the source node S selects both the shortest and the safest trust chain, which relays it to the destination node D. We evaluate the length of a given trust chain TC by the number of certificates which should be collected. This metric allows to reduce the treatments of certificates when collecting and also reduces the computation overhead when verifying the validity of signatures. We evaluate the safety of a given trust chain TC by the safety degree of nodes which are concerned by the certificates included in this chain. The safety degree of a given node is calculated in function of the number of certificates issued to it following the partial web-of-trust. With the safest chain, we give less priority to the certificates issued by the nodes with lower safety degree. With this manner, we reduce

the risk to fail the signatures verification and hence avoid triggering another process of recovery, selection and collection of another alternative certificate chain. We calculate the safety degree of a given node i through the number of incoming edges (denoted by d_i^-) to this node following the partial web-of-trust ∂. Note that the safety degree does not imply the node trust degree or its reputation. The reputation of a node may decrease or increase depending on its behavior which can be either positive or negative. Our protocol does not address this aspect. The safety degree of a node is measured in terms of the number of nodes that believe on the validity of its public-key. If a large number of nodes believe on the validity of the public-key of a given node, it reinforces the belief that the key belongs to the node in question. This metric is very important, which we have introduced in order to measure the credibility of the certificate chain by selecting that covered by nodes having the maximum value of safety degree. The safety degree of a node cannot be negative and in the worst case, it may be equal to zero. In this case, no trust chain passes through this node. If the value of the safety degree of a node is less, this implies that its impact will be very low in terms of certificate chains number passing through this node, and hence, our protocol gives priority to the most credible chain. We calculate the safety degree Ω_{TC} of a given chain TC such as:

$$\Omega_{TC} = \min\{d_i^-\} \ / \ i \in TC$$

i.e. we refer to the node which has the lowest safety degree. From an ensemble of trust chains, the safest one is the one which has the maximal value of the lowest safety degree of the intermediate nodes. For example in Figure 3 (d), the safety degree of the node B is $d_B^- = 2$ and the safety degree of $TC = \{S, H, B, E, D\}$ is $\Omega_{TC} = 1$. In order to select the best chain, we combine the two metrics, i.e., the safety and the length of the trust chains. To do this, we minimize the trust chain length metric and we maximize the safety degree metric. Thus, we introduce the objective function $\mathcal{F}$ as:

$$\mathcal{F}(TC) = \Omega_{TC} \times (|TC| - 1)^{-1}$$

In Table 2, we present the rate of all the trust chains that connect the source node S to the destination D following the partial web-of-trust illustrated in Figure 3 (d). $TC^* = \{S, B, E, D\}$ is the optimal trust chain which follows the highest tradeoff between the length and the safety. The process of the certificate collection of the optimal trust chain is described in the

following subsection.

TC_l	$\lvert TC_l \rvert - 1$	Ω_{TC_l}	$\mathcal{F}(TC_l)$
$\{S, A, G, F, D\}$	4	1	0.25
$\{S, B, E, D\}$	3	2	0.67
$\{S, H, B, E, D\}$	4	1	0.25
$\{S, H, I, J, K, D\}$	5	1	0.20
$\{S, H, I, J, K, E, D\}$	6	1	0.17

Table 2: Trust chains evaluation

3.5. Certificate collection

Once the source node selects the optimal trust chain TC^*, then it proceeds to collect the coalition of the certificates composing this chain. Therefore, it sends an individual certification request to each node concerned by the certificate chain except that issued by itself. Each intermediate node $TC^{*(k)}$ following the trust chain TC^* will be requested for the certificate $\mathcal{C}rt_{TC^{*(k+1)}}^{(TC^{*(k)})}$. Upon receiving the certification request, each node responds to the source node S with the requested public-key certificate. When the source node S receives the chain of certificates relating to the optimal trust chain TC^*, it proceeds to authenticate the intermediate nodes by verifying the following:

$$\vartheta = \left(\ldots \left(\left(\vartheta_{TC^{*(2)}} \right)_{K_{TC^{*(3)}}} \right)_{K_{TC^{*(4)}}} \ldots \right)_{K_{TC^{*(|TC|)}}}$$

A compromised node has no possibility to usurp the identity of other nodes, because the source node can verify the identity by checking the ownership of the private-key associated with the certified public-key. If the certificate update is made regularly, a compromised node is given practically no possibility to compromise a private-key before the key expires. The stamp generated in the trust chains recovery step is signed by the intermediate nodes, however the exchanged messages do not require confidentiality, so threats such as eavesdropping and corruption cannot degrade the security of our protocol.

3.6. Certificate revocation

A node i can revoke any certificate $\mathcal{C}rt_j^{(i)}$ it issued if it considers that the public-key of the given node j is no longer valid. In this case, it removes the

revoked certificate from its local repository and sends a revocation message to the node $\mathtt{j}$ (the subject of the certificate). The latter should also remove the revoked certificate from its local repository. A compromised node may do not remove the revoked certificate. This does not influence the security of our protocol, because in the step of trust chains recovery, the issuer node verifies if the certificate belongs to its local repository, otherwise, the reply packet will be dropped. A node $\mathtt{i}$ can also revoke its own certificates if it wishes to renew its pair of keys (for example, if it believes that its private-key was compromised). In this case, it removes from its local repository all the certificates that cover its old public-key and sends a renew request $\mathtt{ReqR}$ (Request for Renew) to the nodes which issued the old ones:

$$\mathtt{i} \longrightarrow \mathtt{j} \;/\; \exists\, \mathcal{C}\mathtt{rt}_{\mathtt{i}}^{(\mathtt{j})} \in \Theta_{\mathtt{i}} : \mathtt{ReqR} = \left(\mathtt{i},\ \mathtt{j},\ \mathcal{C}\mathtt{rt}_{\mathtt{i}}^{(\mathtt{j})},\ \mathtt{K'}_{\mathtt{i}}\right)_{\mathtt{K}_{\mathtt{i}}^{-1}}$$

The certification request includes the revoked certificate $\mathcal{C}\mathtt{rt}_{\mathtt{i}}^{(\mathtt{j})}$ and the $\mathtt{i}$'s new public-key $\mathtt{K'}_{\mathtt{i}}$. When a node receives the certification request, it replaces the revoked certificate with the new one and sends the latter to the requestor node.

4. Performances evaluation

In this section, we verify the performance of our protocol with respect of three important metrics: the certification success probability, the response time and the communication overhead. The certification success probability evaluates the safety metric of our protocol and it is performed through an analytical modeling using Markov chains. The response time and communication overhead evaluate the metric of the certificate chain length and are performed through simulations. We have compared our protocol to the protocol of Kitada et al. [14] and Hisham et al. [10]. In the protocol of Kitada et al., when a source node needs to collect a certificate chain toward a destination, it broadcasts a request to the nodes that the source directly trusts. If an intermediate node receives the request, it modifies the request by adding its own certificate and broadcasts it to the nodes that it directly trusts. At the end of the process, the destination node adds its own certificate to the request and sends all the certificate chains to the source node. The protocol of Hisham et al. follows the same process, however the source node gives priority to the most shortest certificate chains.

4.1. Analytical modeling

In order to measure the certification success probability of the three protocols, we develop an analytical model based on Markov chains which is illustrated in Figure 4. We consider a set of available certificate chains $C_1, C_2, \ldots, C_N$ allowing the source node to reach its interlocutor. The main objective is to evaluate the quality of the certificate chain to be selected in order to authenticate the destination public-key. In the protocol of Kitada et al., the source node proceeds to verify the first received certificate chain. If its fails, the source node passes to the second and so on. Hence, the choice of the certificate chain depends highly to the network topology and thus we consider that the choice is done in an arbitrarily manner. The protocol of Hisham et al. selects the shortest chain and our protocol selects both the shortest and safest certificate chain.

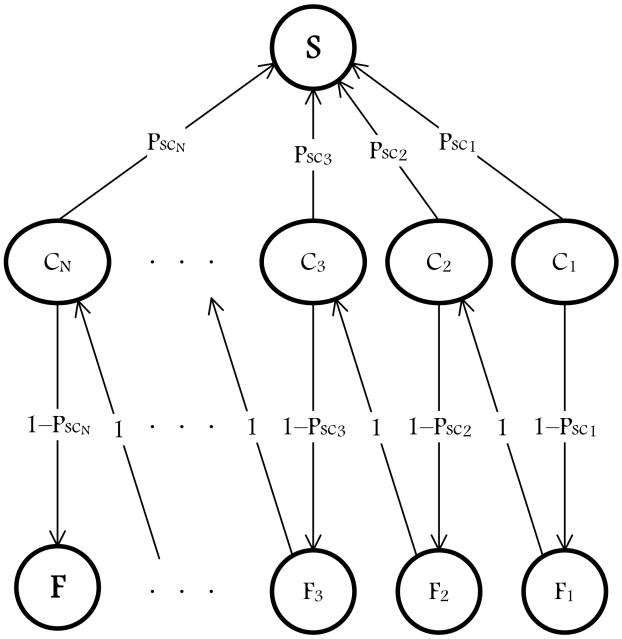

Figure 4: Markov chains model

The proposed model is generic for the three protocols in which the state C_i represents the state of the system when choosing the i^{th} certificate chain; the state F_i represents the verification failure of the i^{th} certificate chain (the system passes automatically to the verification of the next certificate chain

C_{i+1}); the state S represents the verification successful of the i^{th} certificate chain and hence the certification service successful; the state F represents the verification failure of the N^{th} certificate chain (the last available one) and hence the certification service failure. The transition probabilities are defined as follows. P_{sc_i} is the transition probability from the state C_i to the state of success S. $1 - P_{sc_i}$ is the transition probability from the state C_i to the state F_i. The probability of certification service successful at the iteration j (i.e. the success probability when verifying the j^{th} certificate chain) can be calculated as:

$$P_S^{(j)} = P_{sc_j} \times \prod_{i=1}^{j-1} (1 - P_{sc_i})$$

i.e. the verification success probability of the j^{th} certificate chain knowing that all previous ones $C_1, C_2, \ldots, C_{j-1}$ undergone failure. The certification service failure probability at the iteration j means that all the preceding $j-1$ iterations undergone failures including the iteration j. It can be calculated as:

$$P_F^{(j)} = \prod_{i=1}^{j} (1 - P_{sc_i})$$

In order to compare the three protocols, we have considered a fully connected network and we have generated randomly a set of web-of-trust. Two nodes are chosen randomly as source and destination node. Following the availability of certificate chains between the two nodes we have calculated the probability of success and failure of the service of certification. The verification success probability P_{sc_i} of a given certificate chain C_i depends highly to the safety degree of each node belonging to the chain, which is calculated as:

$$P_{sc_i} = \prod_{j=1}^{|C_i|} \frac{d_j^-}{\max\{d_k^- : k \in \text{web-of-trust}\}}$$

Figure 5 and 6 represent, respectively, the success and failure probabilities in function of the certificate graph density. The abscissa axis values represent the web-of-trust connectivity degree. For example, the value 3/4 indicates that the web-of-trust is at 75% connected. Figure 7 and 8 rep-

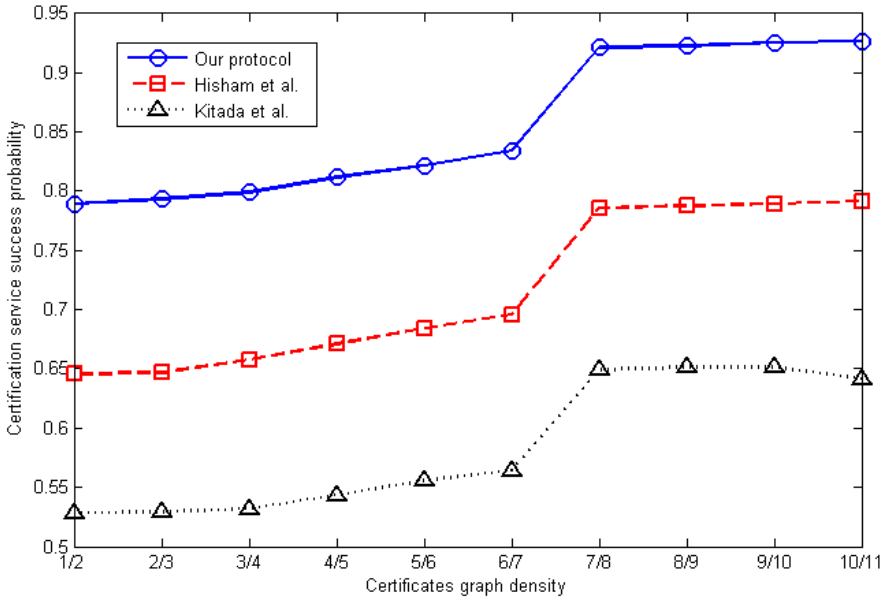

Figure 5: The certification service success probability in function of certificates graph density

resent, respectively, the success and failure probabilities in function of the number of iterations. The abscissa axis values represent the number of times the source node retries to verify a novel chain when the previous one undergoes failure. For example, the value 1 indicates that the source node verifies the first optimal chain, if it fails it passes to the best second one and so on. With comparison to the protocols of Kitada et al. and Hisham et al., our protocol achieves the best results. This is interpreted by quality of certificate chains selected which considers unlike the other protocols the degree of the certificate chain credibility.

4.2. Simulation results

In this subsection, we present the parameters and assumptions related to the environment of simulations, and then we present and discuss the obtained results.

4.2.1. Parameters and assumptions

We have implemented our simulations using Matlab environment [19]. We have simulated a mobile ad-hoc network with 50 to 150 nodes. Each

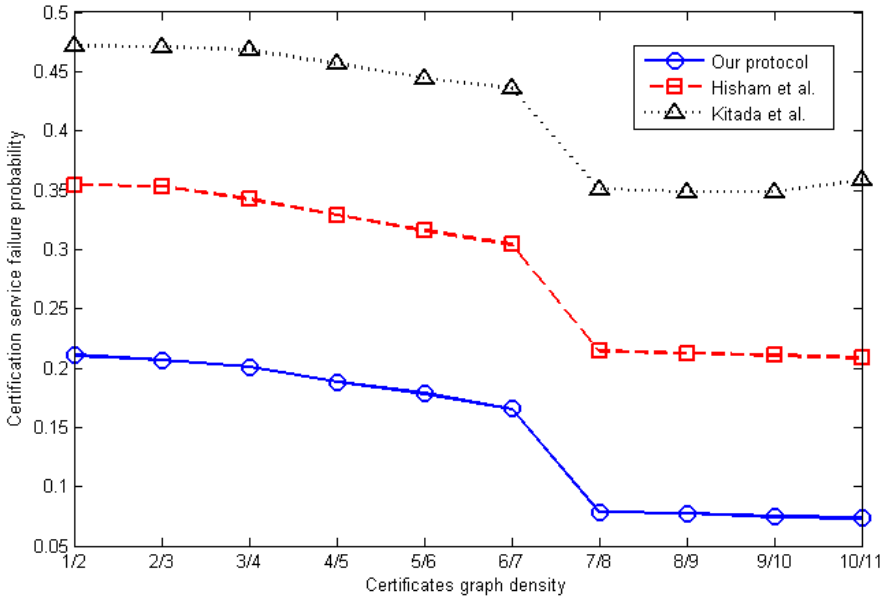

Figure 6: The certification service failure probability in function of certificates graph density

node has a nominal range of 250m and moves on a square area of $1km^2$. Each node is configured with a wireless device with a transmission speed of 50Kbps. The movement pattern is defined by the random waypoint model. A mobile node moves in the area from its current position to a new location by randomly choosing a destination coordinates and the time that it will pause when it reaches the destination. After the pause time, the node chooses a new destination and pause time. This is repeated for each node, until the end of the simulation time. Our simulator estimates if a radio link exists among the nodes according to the distance that separates them. Initial nodes positions are random on the surface. We assume that nodes have the same hardware characteristics and processing capabilities. We assume that the certification requests arrive following a Poisson law with an average inter-arrival between requests of λ. The size of a public-key certificate is assumed of 15Kbyte. We have evaluated two metrics of performance:

1. The response time, which represents the average delay from the generation of the request packet until to receive at least one valid certificate chain by the source node.

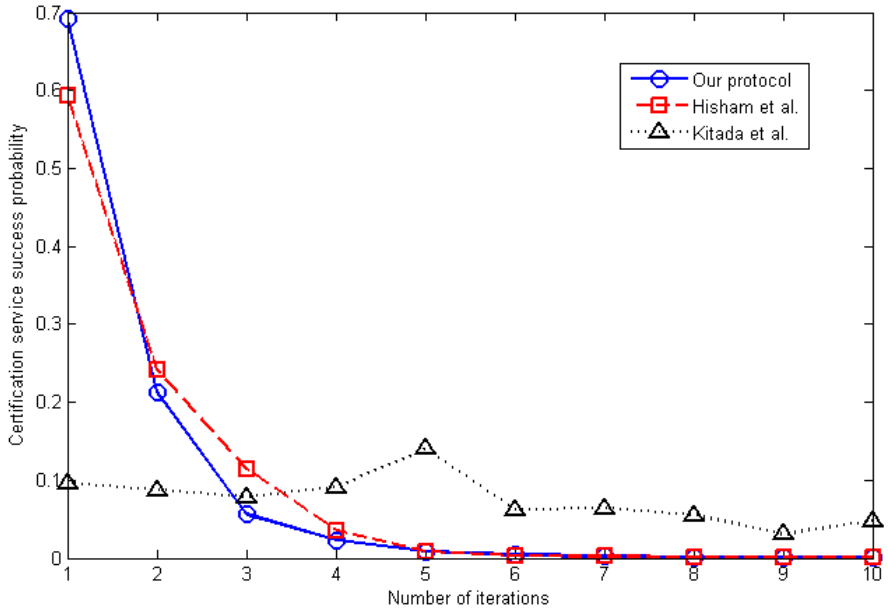

Figure 7: The certification service success probability in function of number of iterations

2. The communication overhead, which represents the total quantity of exchanged data in the network when collecting the certificates among the nodes during the simulation.

For each metric of performance, we have studied the impact of three parameters:

1. The number of ascending trusted nodes per node (denoted by η), which represents the average certificate number issued to each node in the network. This parameter allows to evaluate the performances of the protocols in function of the trust graph density.
2. The network size, which represents the number of nodes in the network. This parameter allows the evaluation of the protocol scalability.
3. The requests intensity, which represents the generated requests number per second in the network.

4.2.2. Results of comparison

Firstly, we were interested in evaluating the impact of η on the performances of the protocols. We illustrate in Figures 9 and 10, respectively, the

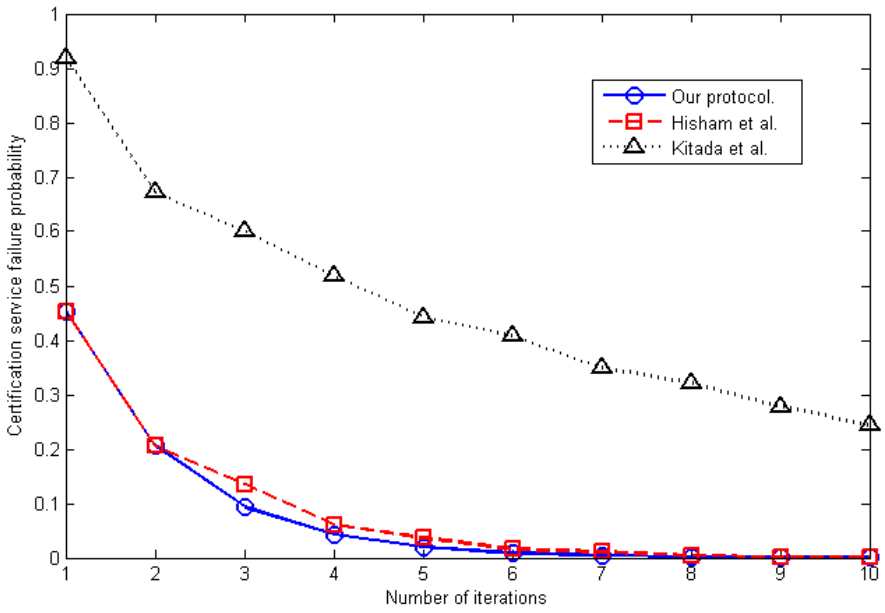

Figure 8: The certification service failure probability in function of number of iterations

comparison results in terms of response time and communication overhead. When the value of η increases, the certificates graph density increases and consequently the number of available certificate chains expands. To this effect, in the protocols of Kitada et al. and Hisham et al., the nodes spend much time when collecting the certificate chains compared to our protocol. In the latter, the nodes select the shortest chains and send, in a unicast manner, an individual request to the nodes belonging to this chain, which reduces enormously the delay of response. Moreover, it reduces the delay of the signatures verification. We note that the overhead of our protocol is much lesser than the other protocols because the size and the number of certificates collected in our protocol are diminutive. In our protocol, the nodes forward the reply packets without adding the certificates, which reduces the overhead of certificate collection.

In the second step, we were interested in evaluating the impact of the network size on the performances of the protocols. We illustrate in Figures 11 and 12, respectively, the comparison results in terms of response time and communication overhead. The response time and the communication overhead increase when the network size increases. Indeed, when the network

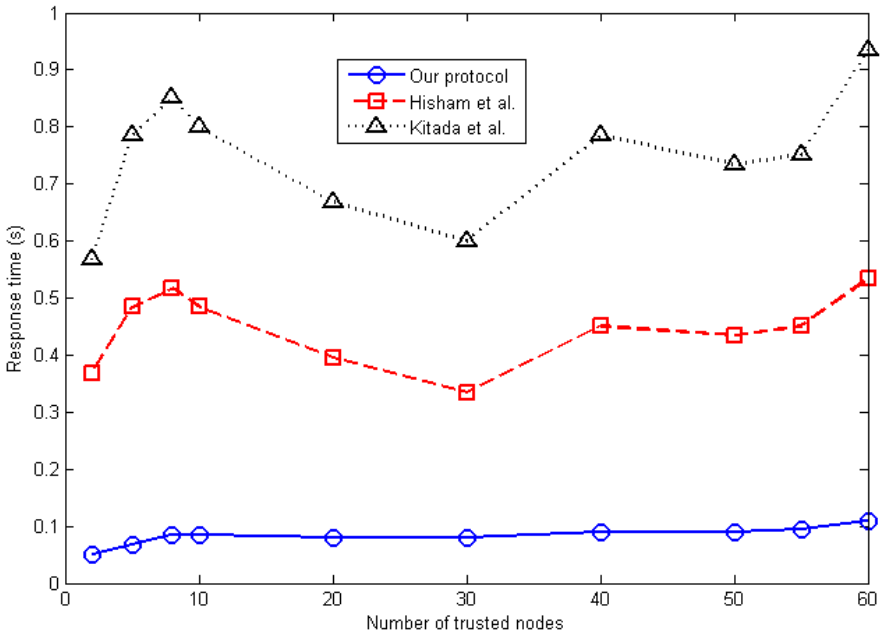

Figure 9: Response time in function of η

size increases the ratio of η versus the number of nodes in the network decreases, which reduces and increases, respectively, the available number and the lengths of certificate chains. This parameter has no much impact on the performances of our protocol, in which only information about the existing trust chains are exchanged instead of the certificates. Once the most optimal chain is found, the node proceeds to the collection. Whatever the trust chain length, the results in terms of response time and communication overhead following our protocol are better than the other protocols. In the protocols of Kitada et al. and Hisham et al., the nodes exchange a large number of certificates end-to-end until the source node receives all the certificate chains. Therefore, the generated communication overhead, computation and communication time significantly increase when the network size increases which affects the scalability of the protocols of Kitada et al. and Hisham et al.

In the last step, we have evaluated the impact of request intensity on the performances of the protocols. In order to evaluate this parameter, we have

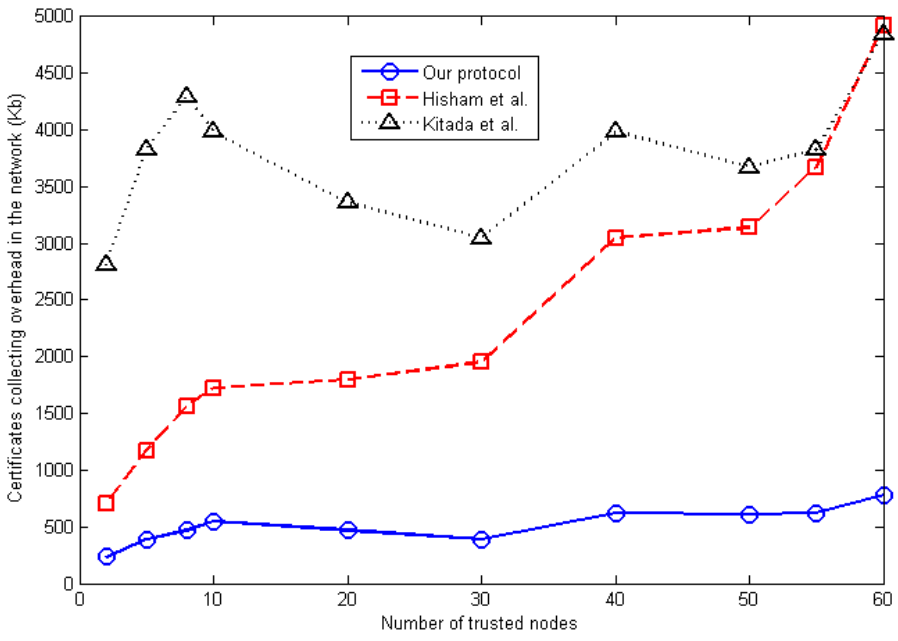

Figure 10: Certificate collecting overhead in the network in function of η

varied the inter-arrival of requests λ. We illustrate in Figures 13 and 14, respectively, the comparison results in terms of response time and communication overhead. The response time and communication overhead of the three protocols increase when the value of λ increases since the treated data size increases per period of time with a frequency of λ requests per second. Our protocol achieves the best results, in which the performance results gap among the three protocols is due to the difference in terms of the exchanged data size in order to complete the certificate collection process.

5. Conclusion

In this paper, we have focused on certificate management in mobile ad-hoc networks and we have proposed a secure and reliable certificate chains recovery protocol for mobile ad-hoc networks in order to resist against compromised nodes, which may deliver false certificates. Our protocol collects and selects the shortest and safest certificate chains. We evaluate the length

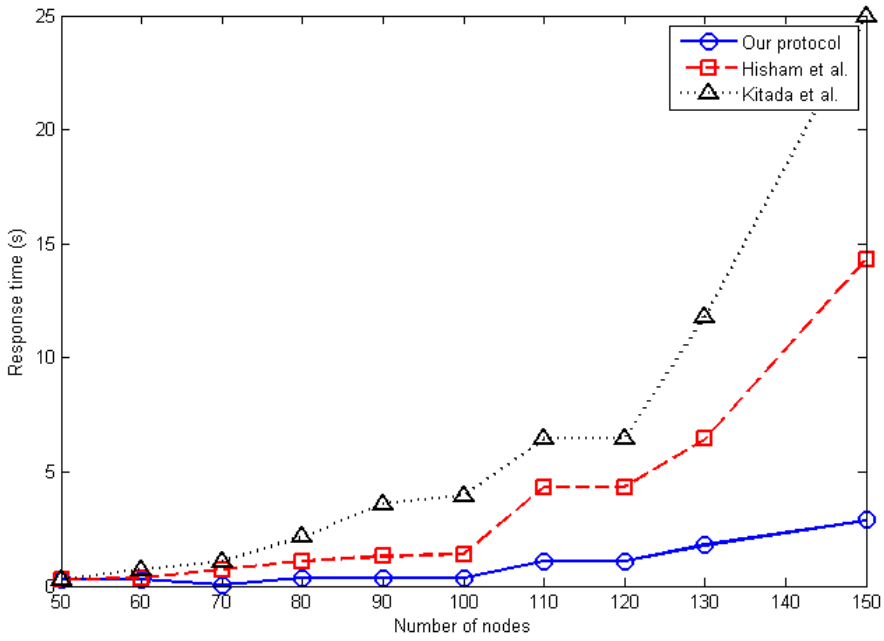

Figure 11: Response time in function of the network size

of trust chains through the certificate number which should be collected. This metric allows to reduce the treatment of certificates when collecting and also reduces the computation overhead when verifying the validity of signatures. We evaluate the safety of trust chains by the safety degree of the nodes concerned by the certificate chain. With the safest chain we give less priority to the certificates issuing by the lower nodes safety degree. Hence, the risk to fail the signatures verification is diminished, which reduces the possibility to trigger another process of recovery, selection and collection of another alternative certificate chain. Moreover, we have developed both an analytical model using Markov chains and simulations in order to evaluate the success probability of certification service, the response time and the communication overhead. Our protocol has been compared to two concurrent protocols, in which it demonstrates interesting results.

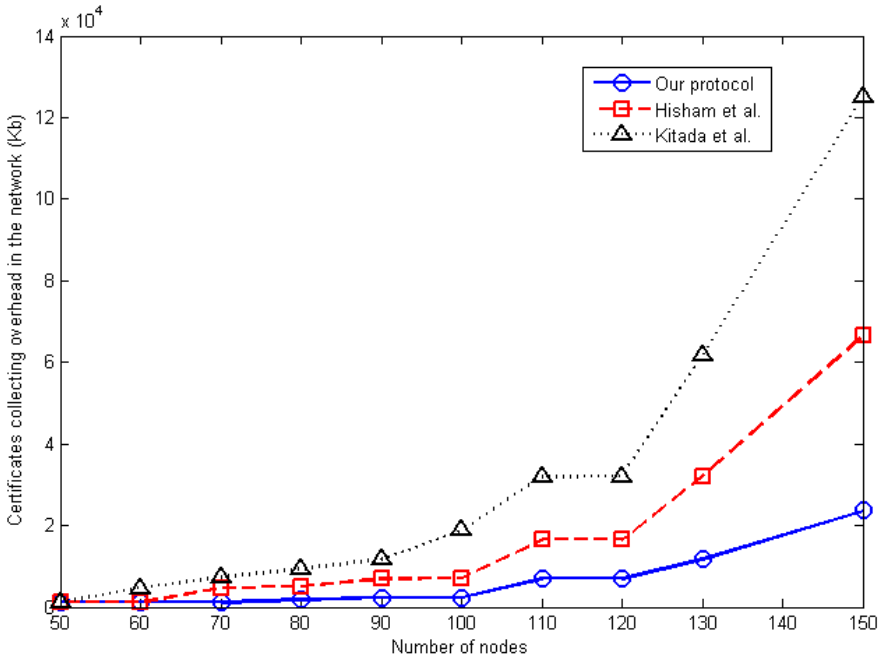

Figure 12: Certificate collecting overhead in the network in function of the network size

6. Acknowledgment

This work was carried out in the framework of research activities of the laboratory LIMED, which is affiliated to the Faculty of Exact Sciences of the University of Bejaia.

References

[1] Ferraz H-G, Velloso P-B, Duarte O-C. An accurate and precise malicious node exclusion mechanism for ad hoc networks. Journal of Ad hoc Networks; 2014.

[2] Basagni S, Conti M, Giordano S, Stojmenovic I. Mobile ad hoc networking: the cutting edge directions. John Wiley and Sons Inc., Hoboken, New Jersey; 2013.

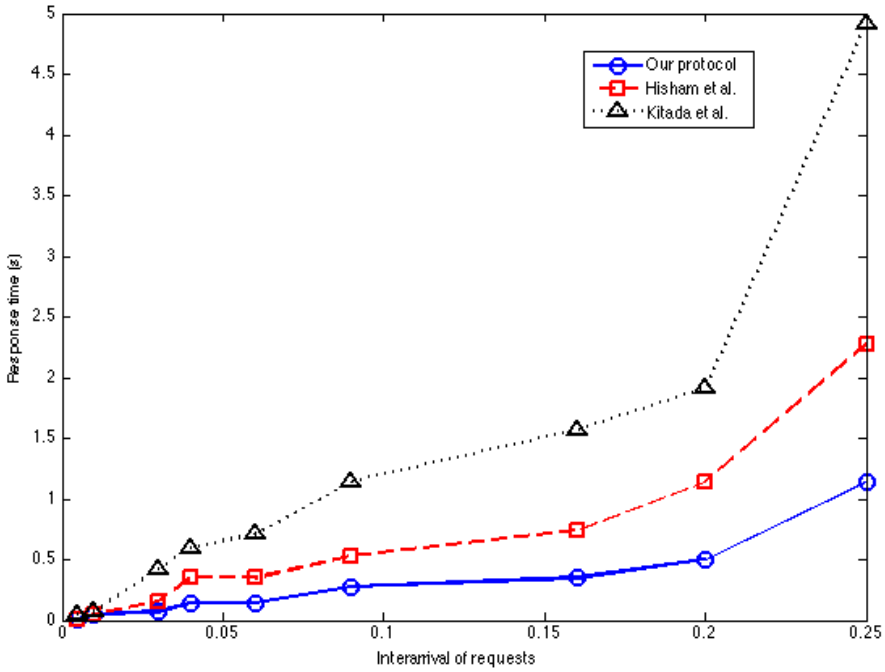

Figure 13: Response time in function of λ

[3] Braga B-B, Chaves I-A, de-Oliveira C-T, Andrade R-M-C, de-Souza J-N, Martin H, Schulze B. RETENTION: A reactive trust-based mechanism to detect and punish malicious nodes in ad hoc grid environments. Journal of Network and Computer Applications; 2013.

[4] Wang W, Wang H, Wang B, Wang Y, Wang J. Energy-aware and self-adaptive anomaly detection scheme based on network tomography in mobile ad hoc networks. Journal of Information Sciences; 2013.

[5] Jhaveri R-H, Patel S-J, Jinwala D-C. DoS attacks in mobile ad hoc networks: a survey. In proceedings of the Second International Conference on Advanced Computing and Communication Technologies; 2012.

[6] Omar M, Challal Y, Bouabdallah A. Certification-based trust models in mobile ad hoc networks: a survey and taxonomy. Journal of Network and Computer Applications; 2012.

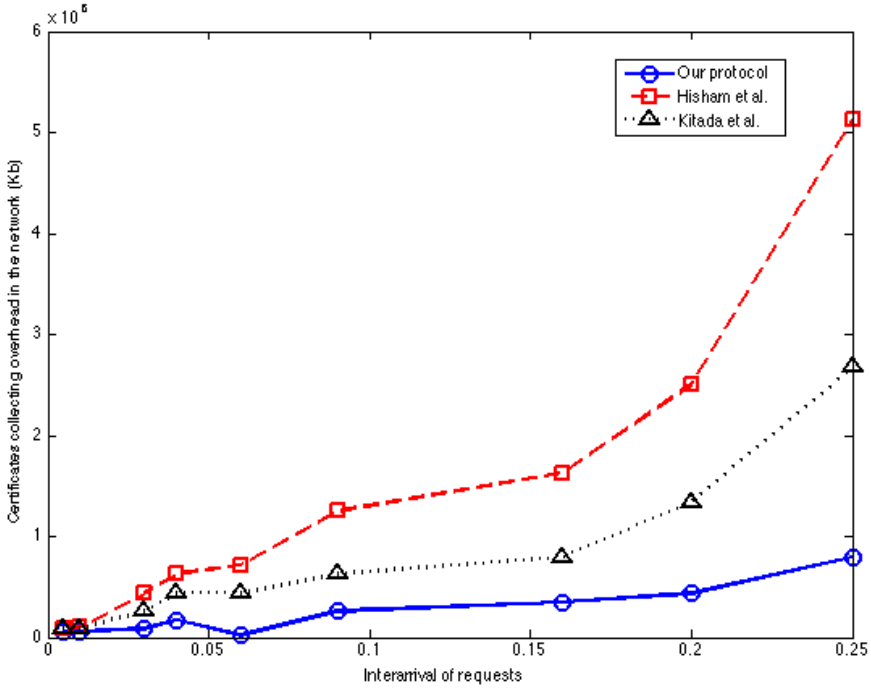

Figure 14: Certificate collecting overhead in the network in function of λ

[7] Suguna M, Subathra M. Establishment of stable certificate chains for authentication in mobile ad hoc networks. In proceedings of the International Conference on Recent Trends in Information Technology; 2011.

[8] Wang J, Liu Y, Jiao Y. Building a trusted route in a mobile ad hoc network considering communication reliability and path length. Journal of Network and Computer Applications; 2011.

[9] Kambourakis G, Konstantinou E, Douma A, Anagnostopoulos M, Fotiadis G. Efficient certification path discovery for MANET. EURASIP Journal of Wireless Communications and Networking; 2010.

[10] Hisham D, James I. On demand self-organized public key management for mobile ad hoc networks. In proceedings of the Vehicular Technology Conference; 2009.

[11] Omar M, Challal Y, Bouabdallah A. Reliable and fully distributed trust model for mobile ad hoc networks. Journal of Computers and Security; 2009.

[12] Funabiki S, Isohara T, Kitada Y, Takemori K, Sasase I. Public key management scheme with certificate management node for wireless ad hoc networks. In Proceedings of the International Multi-conference on Computer Science and Information Technology; 2006.

[13] Yi P, Hou Y-F, Zhong Y, Zhang S, Dai Z. Flooding attack and defense in ad hoc networks. Journal of Systems Engineering and Electronics; 2006.

[14] Kitada Y, Arakawa Y, Takemori K, Watanabe A, Sasase I. On demand distributed public key management using routing information for wireless ad hoc networks. Journal of IEICE Transactions on Information and Systems; 2005.

[15] Kitada Y, Watanabe A, Takemori K, Sasase I. On demand distributed public key management for wireless ad hoc networks. In proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing; 2005.

[16] Xia L, Steven G, Jill S. On demand public key management for wireless ad hoc networks. In proceedings of the Australian Telecommunication Networks and Applications Conference; 2004.

[17] Ren K, Li T, Wan Z, Bao F, Deng R, Kim K. Highly reliable trust establishment scheme in ad hoc networks. Journal of Computer Networks; 2004.

[18] Capkun S, Batten L, Hubaux J. Self-organized public key management for mobile ad hoc networks. Journal of IEEE Transactions on Mobile Computing; 2003.

[19] http://www.mathworks.com/