Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-019-00787-3

Secure and reliable certification management scheme for large-scale MANETs based on a distributed anonymous authority

Khaled Hamouid[1] · **Kamel Adi**[2]

Received: 29 November 2018 / Accepted: 3 July 2019

Abstract

This paper proposes a compromise-tolerant (t, n)-threshold certification management scheme for MANETs. Our solution allows to mitigate the impact of compromised nodes that participate in the certification service. In our design, certification management is achieved anonymously by an Anonymous Certification Authority (ACA). The latter is fully distributed into multiple disjointed coalitions of nodes whose structure is made hidden. This prevents an adversary from taking the control of the ACA by arbitrarily compromising t or more nodes. In other words, our proposal enhances the compromise-tolerance to more than the threshold number t of nodes without breaking down the whole certification system. As a result, our scheme requires a very smaller threshold than traditional schemes, which improves considerably the service availability. The experimental study shows a clear advantage over traditional threshold-based certification schemes by ensuring a significant positive compromise between security and availability of certification service.

Keywords MANET · Secret sharing · Certification · Anonymous signatures · Verifiable encryption · Compromise-tolerance

1 Introduction

The spectacular growth of wireless communication technologies has given rise to new network paradigms such as Mobile Ad-hoc NETworks (MANETs). The pertinence of MANETs has been, notably, marked by their ability to be self-organized and spontaneously deployed without requiring any pre-fixed network infrastructure. In fact, this feature makes them better suited than their traditional networks counterparts for a wide range of applications. However, the inherent characteristics of MANETs, such as the lack of infrastructure, the dynamic topology, the resource constraints and the wireless links introduce new security challenges which are currently in the core of a constant research work focusing on the design of suitable security mechanisms.

Certification is an important security element for any computer network, and it represents a fundamental ingredient for most of security services, such as authentication, authorization and key distribution. In traditional networks, PKIs (Public Key Infrastructures) are often used as the underlying trust infrastructure for certificate management. In PKIs, certification is achieved by a centralized Certification Authority (CA), which is a highly protected trusted third party with a high computing capacity. Unfortunately, such a model is not feasible in MANETs due to their underlying characteristics.

To overcome this issue, many researchers [3, 4, 7, 8, 12] suggested a distributed design of certification management systems based on the concept of (t, n)-threshold cryptography. In fact, *Distributed-Certification-Authority* (*DCA*) approach provides a practical solution to share the role of the CA amongst all or a subset of network nodes. However, most of proposed DCA-based schemes in MANETs have focused on the distribution mechanisms and efficiency issues, but they did not pay a lot of attention to the security of the DCA itself. Since the DCA is formed by network nodes deployed in unattended and hostile environments, attacks raised from internal compromised nodes are a serious threat which is hard to

✉ Khaled Hamouid
k.hamouid@univ-batna2.dz

Kamel Adi
kamel.adi@uqo.ca

[1] LaSTIC research laboratory, University of Batna 2, Batna, Algeria

[2] Computer Security Research Laboratory, Department of Computer Science and Engineering, University Of Quebec in Outaouais, Gatineau, Canada

Published online: 12 July 2019

defend. As DCA-based schemes are commonly based on (t, n)-threshold cryptography, their security requires setting a large threshold value. However, increasing the threshold in large-scale MANETs affects the certification service availability. Furthermore, share refreshing [3], which can be used to enhance the security, has also its limits in large-scale MANETs. Indeed, the dynamic topology, mobility, scalability, asynchronicity, and DoS attacks are some factors that may significantly increase the time interval of share refreshing, which allows an adversary to recover the shared CA's secret key.

Being motivated by this issue, we propose a new compromise-tolerant (t, n)-threshold certification management scheme for MANETs. Our solution aims to mitigate the impact of compromised nodes that participate in the distributed certification service. In our approach, certification management is achieved by an Anonymous Certification Authority (ACA), which is fully distributed into multiple (t, n)-disjointed coalitions of nodes. The coalition composition of the ACA is made hidden, and certification transactions are achieved anonymously by specific coalitions (i.e. specific sub-groups of anonymous nodes), called *valid anonymous coalitions*. The anonymity of the distributed ACA is based on pseudonyms and verifiable encryption techniques. This prevents an adversary from locating an ACA's valid coalition, and hence it cannot take the control of the ACA by arbitrarily compromising t or more nodes. More specifically, the proposed solution enhances the compromise-tolerance to more than the threshold number t of compromised nodes without breaking down the whole certification system. As a result, our proposal meets the robustness requirements without increasing the threshold. In other words, for the same security level as for traditional DCA, our model requires smaller threshold, and hence it guarantees better availability. On the other hand, our solution includes a traceability and revocation scheme which allows to revoke the anonymity of compromised nodes.

The performance and security evaluation is conducted through both numerical analysis and MATLAB simulations with comparison to concurrent similar approaches. The results show a clear advantage over traditional DCA schemes by ensuring a significant positive compromise between security and availability of certification service.

The rest of the paper is organized as follows. Section 2 presents a review of relevant works in the literature. Section 3 presents the underlying building blocks of our solution. In Section 4, we describe the proposed scheme. In Section 5, we present the underlying protocols and algorithms details. We analyze the scheme robustness trough simulations in Section 6. Section 7 discusses the advantages and limitation of our approach. Finally, we conclude the paper in Section 8.

2 Related work

Conventional certification models, such as PKI and classical PGP, are designed for traditional networks where an infrastructure, including online servers and centralized trusted authorities, is required. Obviously, such approaches are not suitable for MANETs due to their inherent characteristics. This makes the deployment of certification authorities very challenging. This section surveys existing certification schemes developed for MANET environments. These solutions are investigated with regard to efficiency, scalability, robustness and security. The proposed schemes may be classified into two main models: PGP-based certification model and Distributed Certification Authority (DCA) model. Under PGP-based certification model, nodes act as independent CAs, and hence, there is no common CA that controls the authority domain in the network. Certificates are issued in a self-organized way so that each node issues a certificate to any other it trusts. This results in a graph of certificates called *web-of-trust*. Many efforts [6, 9, 11, 17] have been devoted for the adaptation of PGP model for Ad-hoc networks. As our work focuses on DCA-based model, we do not review PGP-based schemes in this paper.

2.1 DCA-based schemes

Unlike PGP-based model, DCA-based schemes [3–5, 7, 8, 12, 18] consider a common authority domain in the network controlled by a DCA. The role of the CA is distributed on all or a subset of participant nodes based on (t, n)-threshold cryptography [16]. Under this technique, the CA's private-key is split into n shares, distributed among nodes in such a way that, any coalition of t out of n nodes can provide a certification service, while it is infeasible with less than t nodes. This approach would allow, indeed, mitigating the issue of single point of failure in MANETs by tolerating the failure of up to $(n - t)$ nodes. It is, however, vulnerable to node compromise and coalition-colluding attacks, where an adversary may controls enough compromised nodes that form the CA to corrupt the security of the whole network. Many solutions have been proposed to deal with this limitation. Li et al. [8] and Zhenhua et al. [3] suggested a solution with a periodic share updating without changing the system key. In this way, a share obtained in a period cannot be combined with another share of subsequent periods. However, these schemes are still vulnerable to coalition-colluding attacks if an adversary is capable to control any subset of t (threshold) nodes within a single updating period. This is likely possible in large scale networks where the period of share updating is quite long, giving more time to the adversary to collect enough shares. In [10], Meng et al. propose a DCA based scheme allowing to dynamically

increase the threshold value. Increasing the threshold helps to make the task of compromising the CA difficult for an adversary. Nevertheless, this results in a degradation of performances in terms of certification availability and delays. Guo et al. [4] introduce an optimal node selection mechanism for threshold certification in MANETs. The basic idea is to dynamically select the best nodes from all available ones based on their security and energy conditions. This allows the maximization of the certification success ratio while minimizing security and energy costs. However, the fact that nodes are selected based on their security levels, does not ensure that an adversary could not compromise those nodes and access to their secret shares. Moreover, there is no mechanism in this scheme to prevent a compromised node to cheat about its security and energy states in order to be selected as a server node. Park et al. [12] propose an ID-based anonymous security mechanism for cluster-based MANETs. This scheme ensures an important security property which is nodes privacy. However, the main drawback of this scheme is that, the underlying threshold signature scheme assumes that an adversary cannot compromise more than t out of n cluster-head nodes. This assumption is not realistic for real MANETs and leaves them vulnerable to coalition-colluding attacks. In our previous work [5], we have proposed SRKM (Secure and Robust Key-Management scheme for MANETs), a secure and robust DCA scheme that resists to mobile adversaries who can compromise more than the threshold number of nodes. In SRKM, network nodes that form the DCA are split into real and virtual servers, where virtual severs distribute their secret shares amongst other nodes. This creates multiple hierarchical coalitions of the DCA which makes it difficult to reconstruct the DCA key from a randomly chosen set of compromised nodes. However, the scheme is still vulnerable to coalition-colluding attacks since virtual servers may reconstruct their secret shares from nodes sharing them.

2.2 Overall discussion

One can see that the common drawback of the studied DCA schemes is that they do not resist to node compromise and coalition-colluding attacks and do not tolerate the compromise of more than the threshold number of nodes. Proposed solutions to deal with this issue, either by secret share refreshing or by threshold increasing have shown their limitations in large-scale MANETs. Our solution aims at mitigating this issue by enhancing the compromise tolerance without increasing the threshold, as this may affect the service availability. To the best of our knowledge, our proposed threshold certification scheme is the first one which can tolerate to more than the threshold number of compromised nodes forming the distributed CA without breaking down the certification system.

3 Preliminaries

In this section, we review concisely the basic building blocks employed in our work namely polynomial secret sharing and verifiable encryption schemes.

3.1 Polynomial secret sharing

In a (t, n)-threshold secret sharing scheme, a dealer D distributes a secret S amongst a set of n participants $P = \{1, 2, ..., n\}$ such that each participant holds a part s_i of the secret S. The secret S could be reconstructed only by authorized subsets of at least t participants. Authorized subsets are called t-out-of-n access structure [1]: $\Gamma = \{A \subseteq P : |A| \geq t\}$ where $1 \leq t \leq n$.

In [16] Shamir introduced a simple and elegant (t, n)-threshold secret sharing scheme based on polynomial Lagrange interpolation. Shamir's scheme consists of two algorithms:

- *Secret Distribution:* Dealer D picks a random, $(t-1)$ degree, polynomial $f(x) = \sum_{i=0}^{t-1} a_i x^i$, where $f(0) = S \in F_p$ (a finite field of size p, where p is a prime number) and $f \in F_p[x]$. Then, D generates n shares $(s_1, ..., s_n)$ such that $s_i = f(i), i = 1...n$, and distributes securely each share s_i to each participant P_i.
- *Secret Reconstruction:* With any subset $A \subseteq P$ of size t, the secret S can be reconstructed by Lagrange interpolation of t points $(i, f(i))$ as: $S = f(0) = \sum_{i \in A} f(i)\beta_i(0)$ where $\beta_i(x) = \prod_{j \in (A \setminus \{i\})} \frac{x-j}{i-j}$ are a Lagrange coefficients.

Shamir's scheme presented above is *information-theoretically* secure, also termed as perfect, that is, with $(t-1)$ or fewer of points $(i, f(i))$, it is infeasible to reconstruct the polynomial $f(x)$ and so the secret S.

3.2 Verifiable encryption

Verifiable encryption protocol is a "*zero knowledge*" protocol where a prover P proves to a verifier V that an encrypted message m satisfies a particular property, without revealing any useful information about m.

Camenisch et al. [2] proposed a verifiable encryption scheme for Discrete Logarithms (DL). Verifiability is achieved by a *zero-knowledge proof protocol (PoK)* to prove properties related to discrete logarithms. Camenisch's scheme consists of an encryption scheme and zero-knowledge proof protocol, which are briefly described in the following:

- **Public-key encryption algorithm:**
  - *Key generation*: Select two random primes p', q' and compute $p = 2p' + 1, q = 2q' + 1, N = pq$. Choose $g' \in \mathbb{Z}^*_{N^2}$ and compute $g = (g')^{2N} \ (mod \ N^2)$. Picks a random $x \in [N^2/4]$ as the secret key, and compute $y = g^x \ (mod \ N^2)$ as the corresponding public-key, where $[a]$ denotes the set $\{0, ..., \lfloor a - 1 \rfloor\}$, and $\lfloor a \rfloor$ is the largest integer $\leq a$.
  - *Encryption*: To encrypt $m \in [N]$ using the public-key y, picks a random $r \in [N/4]$ and compute $u = g^r \ (mod \ N^2), e = y^r h^m \ (mod \ N^2)$, wehere $h = 1+N \ (mod \ N^2)$. Outputs the encryption of m: $c = (u, e)$.
  - *Decryption*: to decrypt a ciphertext $c = (u, e)$ using the secret key x, compute $\hat{m} = (e/u^x)^{2t} \ (mod \ N^2)$, where $t = 2^{-1} \ (mod \ N)$. The decryption is m if $\hat{m}$ is of the form h^m.
- **Zero-knowledge proof of a Discrete Logarithm (DL):** Assume a ciphertext c as an encryption of a message m under a given public-key y. Zero-knowledge proof of DL, denoted $PoK\{(m) : c = E_y(m) \wedge \delta = \gamma^m\}$, is a two-party protocol whereby the encryptor (prover) P of m proves to a verifier V that the ciphertext c is an encryption under y of $log_\gamma \delta$ (discrete logarithm of an element δ with respect to base γ), where c, δ, γ are publicly known. Figure 1 presents a simplified version of Camenisch's zero-knowledge proof of DL [3].

4 The proposed scheme

In the proposed scheme, we assume an Ad-hoc network deployed spontaneously from N mobile nodes without the assistance of any centralized trusted authority. In the following subsections, we discuss the adversary model and intended security properties, and then we describe our proposed (t, n)-threshold certification model.

4.1 Adversary model

We consider an adversary with the capability of compromising k arbitrary nodes, within a period of time, where k may exceed the certification threshold (t). Therefore, nodes are either *honest* (follow the rules) or *compromised* (may not follow the rules). We make the following assumptions about the potential misbehavior of a compromised node:

- *Corruption*: this is when a compromised node reveals its secret share to a third adversary or colludes with other compromised nodes in order to reconstruct the CA's secret key.
- *Disruption*: this is when a compromised node sends an inconsistent partial certificate to prevent other nodes from correctly reconstruct the requested certificate.

4.2 Trust model

We consider a network with a single authority domain managed by two authorities operating at two security hierarchical levels. The first level's authority, named *Anonymous Certification Authority (ACA)* is controlled by the second level's authority, named *Pseudonym Authority (PA)*. Both the ACA and the PA are fully distributed amongst network nodes based on threshold cryptography. The security in the first level allows to secure communications by ensuring authentication and authorization, based on conventional certificates, which are managed by the ACA. In this level, nodes reveal their real identities during regular networking operations. In the second level the PA ensures the security of the distributed ACA based on a pseudonym-based anonymity scheme. In this level, ACA members communicate anonymously when they are executing ACA tasks such as a certificate issuing to a network node. Each node holds a pseudonym and a certificate which

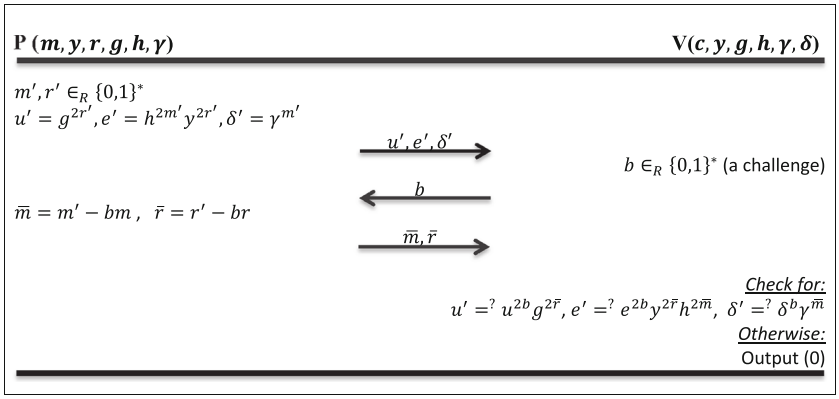

Fig. 1 Zero-knowledge proof (PoK) of a discrete logarithm

are not cryptographically linkable. The certificate is used, as conventionally assumed, for secure communications, authentication and authorization during regular networking operations where real identities are used. Pseudonyms are, however, used when nodes participate in certificate issuing while remaining anonymous.

4.2.1 Anonymous certification authority

The ACA is responsible for certification management. The ACA is fully distributed amongst network nodes based on a particular (t, n)-threshold scheme, and has a private/public keys pair denoted (s, P). Certificates can be issued by specific valid coalitions of ACA's members. During the certificate issuing process, ACA's members operate anonymously using *verifiable pseudonyms* rather than their real identities. This means that no one can identify and locate the members of a valid coalition used to provide a certification service. The ACA deployment protocol is described in Section 5.4.

4.2.2 Pseudonym authority

The PA is responsible for managing the anonymity and anonymity revocation of ACA members. More precisely, it has the ability to execute the following tasks: 1) issuing *traceable* and *verifiable* pseudonyms (PS_i) for ACA's members, 2) providing traceability and revocation of misbehaving ACA's anonymous members. To perform these tasks, the PA uses three private/public keys pairs $\{(x_1, Y_1), (x_2, Y_2), (a, A)\}$, where $x_1, x_2, a \in \mathbb{Z}_q$ (for prime q), $Y_i = g_i^{x_i}$ $(i = 1, 2)$, $A = g_2^a$, and g_1, g_2, g are generators of cyclic groups G_1, G_2, G respectively. The PA is fully distributed to all network nodes using (k, N)-threshold cryptography, where $(k < N)$ is the threshold and N is the network size. In other words, each node gets one share from each of the aforementioned PA's secret-keys, while none of them has the whole knowledge of (x_1, x_2, a). This way, PA's tasks are performed in a threshold fashion by any coalition k-out-of-N of the PA's members. Protocols and details of PA's tasks are described in Section 5.

4.3 Security properties

The proposed scheme is designed to achieve the following security properties:

- *Coalition/signer anonymity*: certificates are signed by specific subsets of nodes called *anonymous valid coalitions*, which form an *Anonymous Certification Authority (ACA)*. Nodes participate *anonymously* in threshold certificate signatures using pseudonyms rather than revealing their real identities. Coalition/signer anonymity property ensures that:
  - given a signed partial certificate, no node can identify who is the actual signer.
  - given a pseudonym, no one can deduce the identity of the pseudonym's holder. The linkage pseudonym/identity is known only to its holder or can be revealed by a specific coalition of nodes.
  - no one can determine the actual composition of a valid coalition.
- *Verifiability*: the correctness of received key shares could be verified by any node. The same is true for received partial certificates.
- *Traceability / revocation*: identity of nodes should be exposed to trace back and revoke misbehaving anonymous nodes that sign inconsistent partial certificates. Hence, with this property, it is possible to recover identities of misbehaving anonymous nodes from their pseudonyms by (and only by) a coalition of a threshold number of authorized nodes. Note that the threshold used for anonymity revocation is not the same used in the certification.
- *Unforgeability*: this property ensures that non-valid coalitions of nodes cannot recover the ACA's secret key or forge a valid certificate on behalf of the ACA.
- *Compromise tolerance*: the certification system and its secret key remain protected with a very high probability even if the number of compromised shareholder exceeds the threshold.

4.4 Scheme description and system settings

In order to enhance the robustness of traditional (t, n)-threshold certification schemes against coalition-colluding attacks, our proposed scheme increases the compromise tolerance to more than t nodes. More specifically, the ACA is distributed into multiple disjointed coalitions of nodes, denoted as *anonymous valid coalitions*. Each valid coalition constitutes a distinct (t, n)-polynomial secret sharing of the ACA. Based on these settings, certificate issuing requires the contribution of at least t nodes from the same valid coalition. Otherwise, any other subset of nodes constitutes a non-valid coalition, and cannot perform certification operations, even if their number reaches the threshold t. Furthermore, the structure of valid coalitions is masked to all, even the coalition members themselves. This prevents attackers from identifying and locating nodes forming a valid coalition. The employed anonymity mechanism is based on the *verifiable pseudonyms* technique. Figure 2 illustrates the proposed certification scheme.

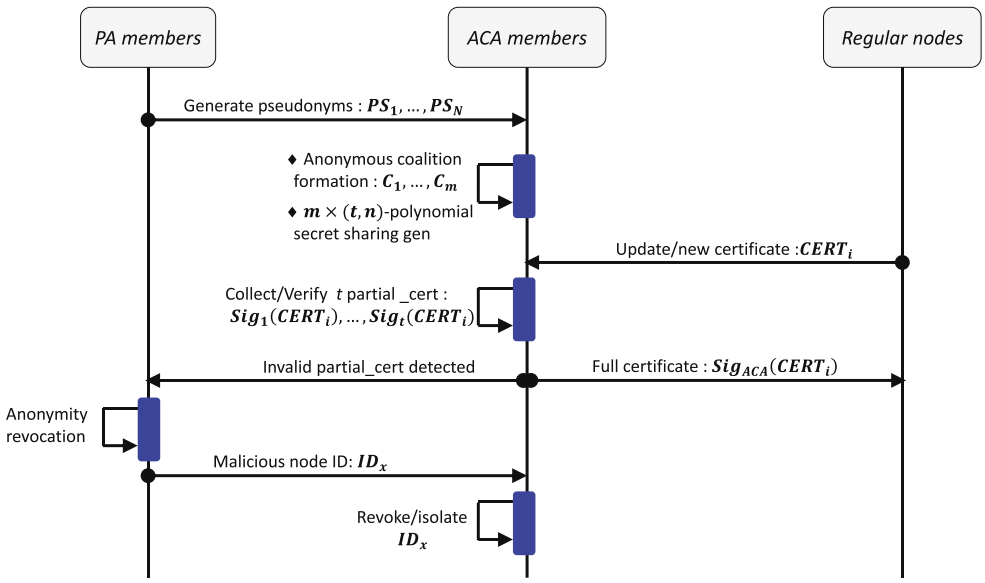

Fig. 2 Overview of proposed scheme

4.4.1 Verifiable pseudonyms

Nodes communicate with their real identities (ID_i) during all regular networking operations except when they participate in certification process, where they rather use pseudonyms (PS_i) and they cannot be identified from their pseudonyms. To that end, we propose, a protocol for generating verifiable pseudonyms (see Section 5.3 for more details) which have the following properties:

- *Verifiability*: this property ensures that any node in the network should be able to verify the validity of a pseudonym, i.e.: to verify that a given pseudonym correctly identifies its holder. A valid pseudonym can only be issued by the PA based on the real identity of the pseudonym holder.
- *Unforgeability*: no one other than the distributed PA can forge a verifiable pseudonym.
- *Traceability*: given a pseudonym PS_i, the corresponding identity ID_i can be extracted from PS_i only by a coalition k-out-of-N of the PA members. This means that no one other than the distributed PA can reveale the identities of nodes from their pseudonyms. Traceability is used to detect and revoke misbehaving anonymous nodes.

4.4.2 Threshold settings

Let V$= \{v_1, ..., v_N\}$ be the set of network nodes. The ACA's secret key s is distributed into m disjointed subsets of secret shares denoted $S_1, ..., S_m$ where $m \approx N/n$ and $n \leq N/2$ is the size of S_i $(i = 1...m)$. Each set $S_i = \{s_{i1}, ..., s_{in}\}$ is a (t, n)-polynomial secret sharing of the ACA's secret key s, where $t < n$. Thus, the global set of shares in the system is $S = \bigcup_{i=1}^{m} S_i = \{s_{ij} : i = 1...m, j = 1...n\}$, where $|S| = |V| = N \approx m \times n$.

Let $\delta : \{1, ..., N\} \longrightarrow \{ij : i \in [1, m], j \in [1, n]\}$ be the function used to distribute to each participant node v_i only one share from S which is $s_{\delta(i)}$. Applying this function to all network nodes, the latter are implicitly partitioned into groups depending on the shares they hold, i.e.: nodes that hold shares from the same subset S_i are considered to be part of the same group. Given this fact, we obtain a partitioning of nodes into m groups $C_1, ..., C_m$, called *valid coalitions*, where $C_i \subseteq V$ and $C_i \cap C_j = \phi$ $(i, j = 1, ..., m/i \neq j)$. Obviously, any t-out-of-n elements from any valid coalition C_i $(i = 1...m)$ can sign a certificate. Any other subset of nodes $B \subseteq V$, where either $|B| < t$ or $(|B| \geq t$ and $|B \cap C_i| < t$, $i = 1...m)$, is called a *non-valid coalition* and cannot produce a signature. In other words, nodes from different groups cannot form a valid coalition, and they cannot combine their shares to provide a valid certificate signature even if their number reaches the threshold.

More formally, the *general access structure* (Γ) that corresponds to the above system settings is defined as:

$$\Gamma = \{W \subseteq V : \exists i \in \{1, ..., m\}, |W \cap C_i| \geq t\}$$

4.4.3 Generation of valid coalitions

Given a secret s shared amongst a set of nodes, a *valid coalition* would be any subset of nodes whose the combination of their shares can reconstruct the secret s, whereas a *non-valid coalition* is any subset of nodes such that their shares combination gives no information about s.

The generation of multiple disjointed valid coalitions of a secret s is conducted as follows (see Fig. 3 for illustration):

- Generate a certain number ($m > 1$) of distinct polynomials of degree $t-1$: $f_i(x) = a_{i,0} + a_{i,1}x + ... + a_{i,t-1}x^{t-1}$ (for $i = 1...m$) such that $a_{1,0} = a_{2,0} = ... = a_{m,0} = s$
- Calculate $n > t$ points from each polynomial $f_i(x)$, $i = 1...m$
- Then any subset of t points from the same polynomial can be interpolated to recover s, and hence each subset of nodes from which we can get the t points is considered as a *valid coalition*.

In the proposed scheme, the ACA's private-key is distributed on the nodes, as described above, by generating $n \times m$ points, where every node is given one point as its share of the ACA's private-key. A valid coalition consists of at least t nodes that hold shares generated by the same polynomial. For instance, suppose the threshold $t = 3$, according to the example of Fig. 3, any of the following subsets is a valid coalition that can recover the ACA's key: $\{s_{11}, s_{12}, s_{13}\}$, $\{s_{11}, s_{12}, s_{14}\}$, $\{s_{22}, s_{23}, s_{24}\}$, $\{s_{31}, s_{32}, s_{33}, s_{34}\}$. Any other subset that does not include at least three points from the same polynomial cannot be used to recover the key.

5 Protocol and algorithm details

5.1 System parameters setup

The network is bootstrapped by the first nodes who meet to form the MANET. The network bootstrapping phase includes the *system parameters setup* as well as the *PA/ACA bootstrapping*. Since we adopt a pure Ad-hoc scenario, this should be done in a fully distributed way without the assistance of a centralized Trusted Authority (TA). Hence, we replace the TA by the first legitimate nodes, which collaboratively initialize and distribute the role of both PA and ACA. It is assumed that this phase is made offline where only legitimate nodes are part of the system.

Prior to PA/ACA deployment, each node establishes a list of identities of all participant nodes in the network. Let $V = \{v_1, ..., v_N\}$ be the set of participant nodes, where $N = |V|$ is the network size. Each physical node $v_i \in V$ has a unique identity ($ID_i \in \{0,1\}^*$).

Afterwards, participant nodes agree on the following system parameters:

- G, G_1, G_2, cyclic groups of order prime q, with generators g, g_1, g_2 respectively.
- $H : \{0,1\}^* \longrightarrow \mathbb{Z}_q$, a one-way hash function.
- (k, N), the threshold settings used by the distributed pseudonym authority.
- $m, (t, n)$, the threshold settings used by the ACA, where $m \approx N/n$ is the number of valid coalitions that form the ACA, $n \leq N/2$ is the size of a coalition, and $t \leq n$ is the signature threshold.

5.2 PA bootstrapping

In this phase, the PA's role is distributed to all nodes using a (k, N)-threshold cryptography (see Section 4.2.2). To initialize the PA, and since no offline TA is assumed, nodes collaboratively generate a (k, N)-threshold secret sharing for the PA's secret-keys (x_1, x_2, a) based on a distributed VSS scheme [14].

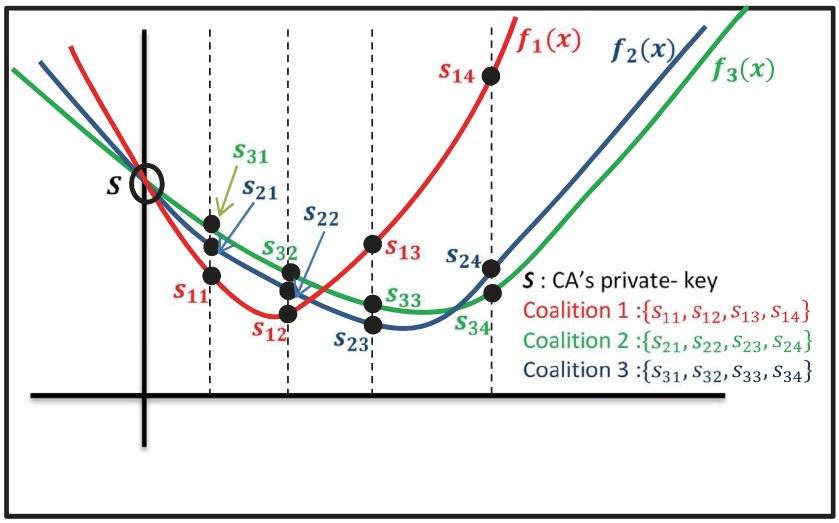

Fig. 3 Multiple polynomial secret sharing

To do this, each node $v_i \in V$ executes the following procedure:

- Picks two random, $(k-1)$'s degree, polynomials $f_i(x) = f_{i,0} + f_{i,1}x + f_{i,2}x^2 + ... + f_{i,k-1}x^{k-1}$ and $h_i(x) = h_{i,0}+h_{i,1}x+h_{i,2}x^2+...+h_{i,k-1}x^{k-1}$, where the coefficients $f_{i,j}, h_{i,j} (i, j = 0...(k-1))$ are elements of $\mathbb{Z}_q$, and $f_i(0) \neq h_i(0) \neq 0$.
- For every node $v_j \in V (v_j \neq v_i)$, send two polynomial points $f_i(ID_j)$ and $h_i(ID_j)$. Meanwhile, send also witnesses for polynomial coefficients $\{g_1^{f_{i,l}}, g_2^{h_{i,l}} : 0 \leq l \leq k-1\}$ to allow correctness verification of the sent points [13].
- Receive about $2(N-1)$ points, i.e.: $f_j(ID_i)$ and $h_j(ID_i)$(for $j = 1...N, j \neq i$) and verify their correctness using polynomial coefficient witnesses.
- Combine received points with the self-generated point $f_i(ID_i)$ to compute $x_{i1} = \sum_{j=1}^{N} f_j(ID_i)$ and $x_{i2} = \sum_{j=1}^{N} h_j(ID_i)$.
- Computes the PA's public-keys (Y_1, Y_2) as: $Y_1 = \prod_{j=1}^{N} g_1^{f_{j,0}}$, $Y_2 = \prod_{j=1}^{N} g_2^{h_{j,0}}$. Resulting keys (Y_1, Y_2) should satisfy $Y_1 = g_1^{x_1}$ and $Y_2 = g_2^{x_2}$.

In the same way, the nodes create the shares $(a_1, ..., a_N)$ for the secret a, as well as the corresponding public shares $(g_2^{a_1\beta_1(0)}, ..., g_2^{a_N\beta_N(0)})$, where $\beta_i(0)$ is Lagrange coefficient.

At the end of the PA deployment, each node v_i keeps privately its secret shares (x_{i1}, x_{i2}, a_i), while the PA's public parameters (Y_1, Y_2, A) are published, where $A = \prod_{i=1}^{k} g_2^{a_i\beta_i(0)}$.

5.3 Pseudonym's generation and verification

To satisfy the pseudonym properties defined in Sections 4.3 and 4.4, we propose a pseudonym generation/verification protocol based on Camenisch's Verifiable Encryption scheme [2] (kindly refer to Section 3.2). Roughly speaking, a pseudonym is generated as a composite of two cryptographic functions. First, the real identity (ID_i) of node v_i is signed by the PA's members in a threshold fashion using the PA's secret-key x_2. This produces a signature denoted PS'_i. The latter is then encrypted, by its holder v_i using the PA's public-key (Y_1). This results in a verifiable pseudonym PS_i. The verification of the generated pseudonyms is achieved using our adapted version of the Camenisch's zero-knowledge proof of a Discrete Logarithm (PoK). The new version of the protocol allows nodes to produce encrypted PA's signatures of their IDs while preserving their public verifiability. In other words, a given node holding a pseudonym PS_i can prove to any other node that its pseudonym is a valid PA's signature of its ID, without revealing the latter. In the sequel, we present, in detail, the protocol. Note that, the used signature scheme is based on Schnorr's scheme [15].

5.3.1 Pseudonym generation protocol

Without loss of generality, let $B = \{v_1, v_2, ..., v_k\}$ be a coalition of k PA's members, and let $v_{j(j>k)} \in V$ be a node that want to generate its pseudonym PS_j. Then, given the PA's credentials $\{(x_{i2}, a_i : i = 1...N), Y_1, A\}$ and the setup parameters $(g_1, g_2, H(.))$, the pseudonym generation protocol proceeds as follows:

- **Step 1.** *Initialization*:
  - 1.1 v_j picks a random secret $r_j \in \mathbb{Z}_q$, and computes $u_j = g_1^{r_j}$.
  - 1.2 v_j generates a pseudonym request that includes its identity ID_j, and sends it to the PA members in B.
- **Step 2.** *PA's signature on* ID_j:
  - 2.1 Each node $v_i \in B$ computes the following partial signature on ID_j and sends it to v_j:

$$PS'_{ij} = x_{i2}.\beta_i(0).H(ID_j||A) + a_i.\beta_i(0) + \psi_i \quad (1)$$

    where $\beta_i(0) = \prod_{l=1(l\neq i)}^{k} \frac{-ID_l}{ID_i - ID_l}$ is the Lagrange coefficient, and ψ_i is a shuffling factor generated by v_i satisfying $\sum_{i=1}^{k} \psi_i + \psi_j = 0$. The shuffling factor is used to prevent potential eavesdroppers from learning PS'_j and hence correlating PS_j to ID_j. One method to generate the shuffling factor can be found in [20].
  - 2.2 v_j reconstructs the PA's signature (PS'_j, A) as:

$$PS'_j = \sum_{i=1}^{k} PS'_{ij} + \psi_j = x_2.H(ID_j||A) + a \quad (2)$$

- **Step 3.** *Verifiable Encryption of* PS'_j:
  - 3.1 Node v_j produces its verifiable pseudonym as an encryption of PS'_j using the PA's public-key Y_1:

$$PS_j = enc(PS'_j)_{Y_1} = g_2^{PS'_j}.Y_1^{r_j} \quad (3)$$

  - 3.2 v_j publishes its pseudonym (PS_j, u_j), while it keeps (r_j, PS'_j) as secret parameters.

5.3.2 Pseudonym verification protocol

Given a pseudonym (PS_j, u_j) and the PA's public parameters (Y_1, Y_2, A), any node $v_i \in V$ can verify PS_j based on the modified PoK as follows:

1) pseudonym holder v_j (Prover) randomly chooses secrets $m'_j, r'_j \in \mathbb{Z}_q$ and computes $u'_j = g_1^{r'_j}$, $e'_j = Y_1^{r'_j} g_2^{m'_j}$, $\gamma_j = g_2^{m'_j} Y_2^{m'_j}$.

2) v_j sends to node v_i (Verifier): (u'_j, e'_j, γ_j)
3) v_i chooses a challenge c and sends it to v_j.
4) v_j sends back to v_i : $\widehat{r_j} = r'_j - cr_j, \widehat{m_j} = m'_j - cPS'_j$, $\widehat{k_j} = m'_j + cH(ID_j||A)$.
5) v_i accepts the pseudonym PS_j if the following equalities hold:

$$u'_j =^? u_j^c g_1^{\widehat{r_j}}$$
$$e'_j =^? PS_j^c Y_1^{\widehat{r_j}} g_2^{\widehat{m_j}}$$
$$\gamma_j =^? g_2^{\widehat{m_j}} Y_2^{\widehat{k_j}} A^c$$

5.4 ACA bootstrapping

Once the PA is initialized and the pseudonyms are generated, the nodes proceed to the bootstrapping of the ACA. To that end, the nodes will be partitioned into m disjointed subsets $C_1, ..., C_m$ called *anonymous valid coalitions*, where each subset forms a distinct (t, n)-threshold secret sharing of the ACA's secret key (s). The (t, n)-secret sharing associated with these coalitions should be generated with accordance to the general access structure (Γ) defined in Section 4.4. In other words, secret shares from different coalitions cannot be combined in order to build the secret s.

Without loss of generality, let $PS = \{PS_1, ..., PS_N\}$ be the list of pseudonyms of participant nodes, where $N = |V|$ is the size of the network. The detailed ACA's bootstrapping protocol is described as follows:

(1) Randomized partitioning of nodes and pseudonym exchange

- Each node generates a sequence number (seq_i) for each pseudonym $PS_i \in PS$ as : $seq_i = H(PS_i||z_i)$, where $z_i \in [1, N^3]$ is a random number. The generated sequence numbers are used to sort out the nodes.
- Each node sorts out the list of pseudonyms it holds according to corresponding sequence numbers.
- Each node determines its coalition and the associated pseudonyms, where the n first pseudonyms constitute the first anonymous valid coalition (C_1), the n next constitute the second valid coalition (C_2), and so on (n is the coalition size).

(2) Polynomial generation

- Each node $v_i \in V$ generates m random polynomials of $(t-1)$ degree:

$$f_i^l(x) = f_{i,0}^l + f_{i,1}^l x + f_{i,2}^l x^2 + ... + f_{i,t-1}^l x^{t-1} \ (l = 1...m) \tag{4}$$

  where $f_{i,0}^1 = f_{i,0}^2 = ... = f_{i,0}^m \neq 0$, $m \approx N/n$ ($n \leq N/2$ is the size of a coalition).

- The ACA's private-key (s) is then:

$$s = \sum_{i=1}^{N} f_i^l(0) = \sum_{i=1}^{N} f_{i,0}^l \ (\forall l \in [1, m]) \tag{5}$$

- For each verified pseudonym $PS_j \in PS$, v_i computes m subshares $\{f_i^1(PS_j), ..., f_i^m(PS_j)\}$ as well as $\{b_{i,0}^l = g^{f_{i,0}^l}, ..., b_{i,t-1}^l = g^{f_{i,t-1}^l} : l = 1...m\}$, where $b_{i,k}^l$ $(k = 0...(t-1))$ are the public coefficients of the generated polynomial $f_i^l(x)$ from node v_i. The latter are used by other nodes to check the consistence of the received shares from v_i.
- For each verified pseudonym PS_j, v_i broadcasts $\{f_i^l(PS_j), b_{i,k}^l : l = 1...m, k = 0...(t-1)\}$ encrypted with u_j so that only node v_j can learn them.

(3) Secret share generation

- v_j decrypts the received subshares $\{f_i^l(PS_j), b_{i,k}^l : i = 1...N \ (i \neq j), l = 1...m, k = 0...(t-1)\}$ using its secret r_j, and then verifies their consistence by the following equation:

$$g^{f_i^l(PS_j)} =^? \prod_{k=0}^{t-1} \left(b_{i,k}^l\right)^{PS_j^k} \ (l = 1...m, i = 1...N, i \neq j) \tag{6}$$

- v_j computes its secret share s_j associated to its coalition from the received subshares. Let $(C_x)_{x\in\{1,...,m\}}$ be the v_j's coalition. Then, v_j's secret share, say s_j, results from summing N out of $(N \times m)$ subshares which have been generated from the x-th polynomial of each node. This is expressed by the following formula:

$$s_j = \sum_{i=1}^{N} f_i^x(PS_j) \tag{7}$$

- v_j computes also the following polynomial coefficients witnesses corresponding to its coalition C_x: $\{b_{x,0} = \prod_{i=1}^{N} b_{i,0}^x, ..., b_{x,t-1} = \prod_{i=1}^{N} b_{i,t-1}^x\}$. These parameters are used by v_j for consistence verification of partial signatures during threshold certificate issuing phase. The use of these parameters is clarified in Section 5.5.
- At the end, v_j computes its public share: $P_j = g^{s_j}$.

(4) ACA's public-key generation

- Each node $v_i \in V$ computes the public-key of the ACA as : $P = b_{x,0}$, where $b_{x,0}$ is the polynomial coefficient witness calculated previously, and

which corresponds to the v_i's coalition, namely $C_{x_{(x\in\{1,...,m\})}}$. This key should satisfy $P = g^s$.

Theorem 1 *Assume that node v_i has the share $s_i = \sum_{l=1}^{N} f_l^h(PS_i)$ and another node v_j has the share $s_j = \sum_{l=1}^{N} f_l^k(PS_j)$ $(h, k \in \{1, ..., m\})$, if $h = k$, then v_i and v_j belong to the same valid coalition. Otherwise, they belong to two distinct coalitions.*

Proof By definition of valid coalition (see Section 4.4), if two nodes v_i and v_j are in the same coalition, then $\exists g(x) \in F_p[x]$: $s_i = g(PS_i)$, $s_j = g(PS_j)$ and $s = g(0)$, where s_i, s_j are two consistent shares of s held by v_i and v_j respectively. According to our share generating protocol, we have:

$$s_i = \sum_{l=1}^{N} f_l^h(PS_i) \text{ and } s_j = \sum_{l=1}^{N} f_l^k(PS_j) \tag{8}$$

Let $g(x)$ be the polynomial resulting from the following sum:

$$\exists r \in \{1, ..., m\} : \sum_{l=1}^{N} f_l^r(x) = g(x) \tag{9}$$

Then, according to formulas (5), (9) we have $g(0) = s$, and from Eq. 9 we have:

$$g(PS_i) = \sum_{l=1}^{N} f_l^r(PS_i) \text{ and } g(PS_j) = \sum_{l=1}^{N} f_l^r(PS_j) \tag{10}$$

According to Eqs. 8 and 10, if $k = h = r$, then : $g(PS_i) = s_i$ and $g(PS_j) = s_j$. □

5.5 Certificate issuing

A certificate can be issued from at least t (the threshold) partial signatures, generated by nodes belonging to a valid coalition (see Fig. 4). Partial signatures are provided anonymously without revealing the real identities of the signers. The requester node, say (v_i), prepares its unsigned certificate $(CERT_i)$ that includes its identity, a public-key and other parameters, and sends it to any node in the network. The node receiving the request, which belongs to a given valid coalition C_x $(x \in [1, m])$, collects $(t-1)$ partial certificates from a randomly selected subgroup $B \subset C_x$ of anonymous nodes. Then, it combines them with its partial signature to produce the requested certificate. We would again point out that each node knows only the pseudonyms of the members of its coalition.

Without loss of generality, suppose that $B = \{PS_j : j = 1...(t-1)\}$ and PS_t is the node who receives the request from v_i to sign its certificate $CERT_i$. The certificate is then issued as follows:

(1) Partial certificate generation

- The node PS_t broadcasts the request to the selected subgroup B. When the nodes of subgroup B are not adjacent, a trivial method is to rebroadcast the request hop by hop till it reaches all target nodes. There exist in the literature many efficient broadcasting protocols for MANETs, which are beyond the scope of this paper.
- Every node $PS_j \in B$ picks a random number $k_{ji} \in \mathbb{Z}_q$ and sends back $g^{k_{ji}}$ to PS_t.
- PS_t generates a random number k_{ti} and sends back to every $PS_j \in B$ the parameter $K_i = \prod_{j=1}^{t} g^{k_{ji}} = g^{k_i}$.
- Every node $PS_j \in B$ sends to PS_t a partial signature $(Sig_j(CERT_i), K_i)$ computed as follows:

$$Sig_j(CERT_i) = s_j.\beta_j(0).H(CERT_i||K_i) + k_{ji} \tag{11}$$

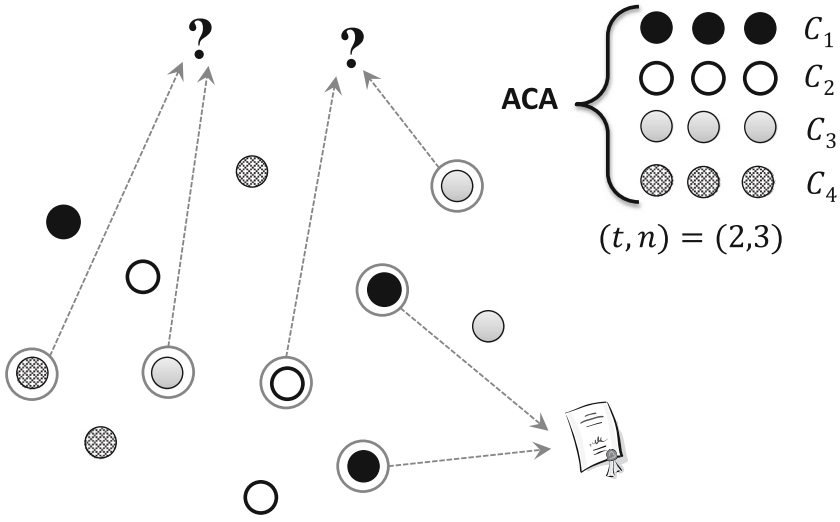

Fig. 4 Certificate issuing (C_i, $i = 1...4$ are valid Coalitions)

Where $\beta_j(0) = \prod_{l=1(l\neq j)}^{t} \frac{-PS_l}{PS_j - PS_l}$ is the Lagrange coefficient, s_j is the secret share of node PS_j and $H(.)$ is a one way hash function.

(2) Partial certificate verification

The goal of this verification is to identify possible corrupt partial certificates. For each partial certificate ($Sig_j(CERT_i)$), received from PS_j ($j = 1...t-1$), the node PS_t performs the following checks:

- verify whether PS_j is a valid ACA's member using the pseudonym verification protocol (see Section 5.3.2).
- verify the signature validity of the provided partial certificate ($Sig_j(CERT_i)$):

$$g^{Sig_j(CERT_i)} =^{?} P_j^{H(CERT_i||K_i)\beta_j(0)}.g^{k_{ji}} \quad (12)$$

where P_j is the public share of the anonymous node PS_j, as each node holds a secret/public share pair ($s_j, P_j = g^{s_j}$). The above verification is used to confirm if the provided partial certificate is signed or not by PS_j. If the verification aborts, the actual partial certificate is considered as invalid.

- verify the consistency of the partial signature with respect to the coalition (C_x):

$$P_j =^{?} \prod_{l=0}^{t-1} (b_{x,l})^{PS_j^l}$$

where $b_{x,l_{(l=1...t-1)}}$ are the polynomial coefficient witnesses associated to coalition C_x. If this verification fails, then the partial certificate is considered as inconsistent with respect to the actual coalition C_x and hence the signer PS_j is considered as compromised (see Section 4.1 for more details)

(3) Full certificate generation

- PS_t builds the certificate of v_i by combining the $(t-1)$ received partial certificates along with its own:

$$Sig_{ACA}(CERT_i) = \sum_{j=1}^{t} Sig_j(CERT_i) \quad (13)$$

- The final certificate is therefore ($Sig_{ACA}(CERT_i)$, K_i), which can be verified by anyone using the ACA public key (P), as:

$$g^{Sig_{ACA}} =^{?} P^{H(CERT_i||K_i)} K_i \quad (14)$$

It should be noted that, during certificate issuing, all transmissions amongst ACA nodes are performed anonymously over the wireless channel. We make this assumption to avoid the derivation of a pseudonym/physical-node relation by using statistical traffic analysis.

5.6 Traceability and revocation

Anonymity of the ACA members should not protect them from being traced back and revoked when they are misbehaving or compromised. To enforce this requirement, we make the anonymity revocable by the PA. More precisely, when a misbehaving anonymous ACA member is detected according to the verification process (cf. Section 5.5), its anonymity is revoked by a valid coalition of PA members (i.e.: recover its real identity).

Assume that a misbehaving node with pseudonym (PS_j, u_j) is detected by a given node v_k, then the traceability and revocation protocol proceeds as follows:

(1) Misbehavior announcement

- the denouncer node v_k issues a notice of revocation against PS_j denoted $REV(PS_j)$, which includes the pseudonym of the denounced node and a misbehavior evidence (e.g. corrupt or inconsistent partial certificate, etc.).
- v_k selects a subgroup of $(k-1)$ nodes, where k is the threshold of the PA. Suppose that this subgroup is $B = \{v_i : i = 1...(k-1)\}$.
- v_k broadcasts the revocation request $REV(PS_j)$ to all nodes in B.

(2) Anonymity revocation

- In order to prevent false accusation, each node $v_i \in B$ receiving the revocation request $REV(PS_j)$ must check the provided misbehavior evidence linked to the anonymous node PS_j. This check is performed as described in the certification issuing protocol (step (2)).
- If the verification is positive, then each node $v_i \in B$ provides to all nodes in B a partial anonymity revocation of PS_j, denoted $D_i(PS_j)$, using its PA's share (x_{i1}), such that: $D_i(PS_j) = \{PS_j, u_j^{x_{i1}\beta_i(0)}\}$ where $\beta_i(0)$ is the Lagrange coefficient.
- From the received partial revocations, each node $v_i \in B$ can compute the full anonymity revocation:

$$D_{PA}(PS_j) = \frac{PS_j}{\prod_{i=1}^{k} u_j^{x_{i1}\beta_i(0)}} = \frac{g_2^{PS'_j} Y_1^{r_j}}{u_j^{x_1}} = \frac{g_2^{PS'_j} g_1^{x_1 r_j}}{g_1^{x_1 r_j}} \quad (15)$$

- Using the computed anonymity revocation above, the nodes can figure out the identity of anonymous

node PS_j. It would be ID_j if the following verification holds:

$$D_{PA}(PS_j) = Y_2^{H(ID_j||A)} A \tag{16}$$

Kindly refer to pseudonyms generation protocol (Section 5.3) and the PA initialization (Section 5.2) for the correctness of above formulas as well as the definition of the used symbols.

(**3**) **Revocation notification**

Once the misbehaving anonymous node v_j is disclosed, it will be revoked as well as all its credentials. To that end, a revocation notice against this node is issued by PA members. The latter includes the identity of revoked node (ID_j), its public-key certificate ($CERT_j$) and its pseudonym PS_j. This procedure is described as follows:

- every node $v_i \in B$ picks a random number $r_{ij} \in \mathbb{Z}_q$ and sends back $g^{r_{ij}}$ to the denouncer node v_k.
- v_k generates a random number r_{kj} and sends back to every $v_i \in B$ the following: $R_j = \prod_{i=1}^{k} g^{r_{ij}} = g^{r_j}$
- each node $v_i \in B$ as well as v_k sign and broadcast a partial revocation notice (REV_{ID_j}) against the revoked v_j:

$$Sig_i(REV_{ID_j}) = x_{2i}\beta_i(0).H(REV_{ID_j}||R_j) + r_{ij} \tag{17}$$

where $REV_{ID_j} = \{ID_j, PS_j, CERT_j\}$

- once k partial revocation notices are issued, the full revocation notice can be reconstructed as:

$$Sig_{PA}(REV_{ID_j}) = \sum_{i=1}^{k} Sig_i(REV_{ID_j}) \tag{18}$$

- the revocation of node v_j is $(Sig_{PA}(REV_{ID_j}), R_j, REV_{ID_j})$, which will be published in the network and can be verified by any node using the public-key of the PA as:

$$g_2^{Sig_{PA}(REV_{ID_j})} =^{?} Y_2^{H(REV_{ID_j}||R_j)} R_j \tag{19}$$

6 Performance and security evaluation

In this section, we provide a performance and security evaluation of our scheme, based on numerical analysis and MATLAB simulations. As our proposal is intended to improve the robustness and the compromise-tolerance, our evaluation is focusing on the system robustness against coalition-colluding attacks. In such attacks, a coalition of corrupted nodes may conspire, using their secret shares, to compromise the distributed certification authority. In our evaluation, we measure the following metrics:

- ***The Successful Certification Ratio (SCR):*** defined as the ratio of the number of successful certification issuance operations to the total number of certification requests.
- ***The compromising probability:*** suppose that x arbitrary nodes ($t \leq x \leq (t-1)m$) are controlled by an adversary $\mathcal{A}$. Based on *Hypergeometric distribution*,[1] the probability that $\mathcal{A}$ recovers the ACA's secret-key or generates a valid certificate is defined as follows:

$$Pr[X \geq t; N, x, n_i (1 \leq i \leq m)] = \sum_{i=1}^{m} \frac{\binom{n_i}{t}\binom{N-n_i}{x-t}}{\binom{N}{x}} \tag{20}$$

where $\binom{a}{b}$ is a binomial coefficient, and X is a random variable denoting the number of nodes in the same coalition. The compromising probability can also be defined as the probability that at least t out of x nodes are in the same coalition. In the sequel, we note $Pr(t)$ this probability.

- ***The vulnerability window:*** most of (t, n)-threshold certification schemes employ a proactive secret refreshing in order to strengthen the security of the certification system against break-ins. This way, newly refreshed and old shares cannot be combined to reconstruct the system key. Therefore, the *vulnerability window* is defined as the time interval in which an attacker can collect at least (t) consistent shares [19]. This interval is not necessarily time-based, it can, for instance, be expressed by events, especially in asynchronous systems. The vulnerability window (ω) can formally be defined as follows:

$$\omega = \begin{cases} n\,\tau + (n-t)\,\tau + \Delta, & n < 2t-1 \\ 2n\,\tau + \Delta, & n \geq 2t-1 \end{cases} \tag{21}$$

Where τ is the time of one share updating, and Δ is the interval separating two consecutive share updating operations. Figure 5a and b illustrate the vulnerability window range in case of $n < 2t-1$ and $n \geq 2t-1$ respectively.

- ***The compromise tolerance:*** this measures how compromise-tolerant the certification system is. In other words, the extent to which the certification system resists to coalition-colluding attacks, when the number of compromised nodes increases. There are many factors that may affect the system robustness, against aforementioned attacks, namely the network size, potential Denial of Service (DoS) attacks, the

[1] The hypergeometric distribution is a discrete probability distribution that describes the probability of k successes in n draws, without replacement, from a finite population of size N that contains exactly K successes

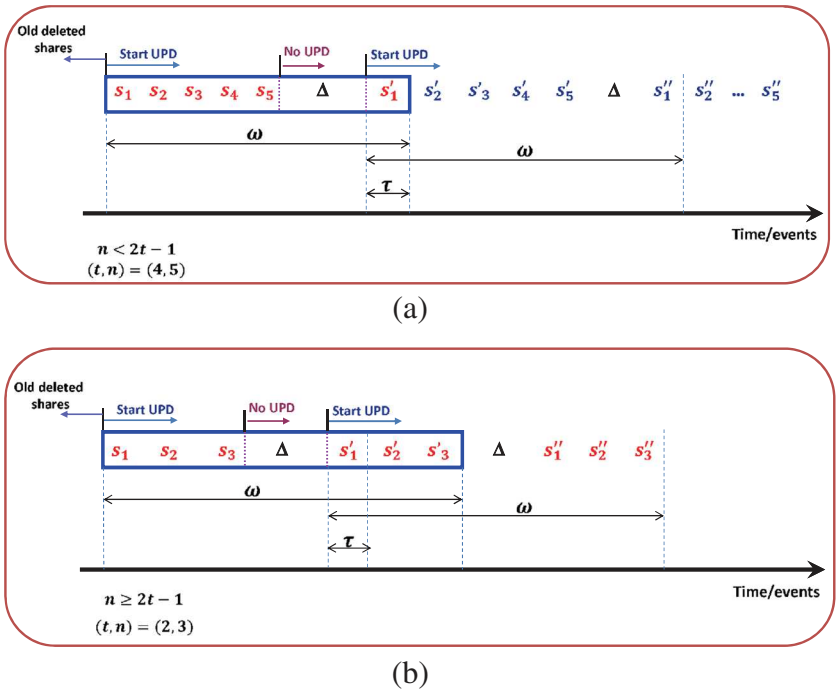

Fig. 5 Vulnerability window: **a** $n < 2t-1$ **b** $n \geq 2t-1$

number of compromised nodes and the threshold configuration. The impact of these factors on the system robustness is analyzed in our simulation. The compromise tolerance of our scheme is evaluated by measuring the probability of compromising the certification system as defined by the formula (20). Also, to allow comparison with existing schemes, we measure the vulnerability window (ω).

In order to highlight the enhancements and improvements of our solution, we compare it to recent existing (t, n)-certification schemes, and specifically those in [3, 4, 7, 8, 12] that we chose on the basis of their relevance and significance. These schemes have the following common two features: (i) any coalition t-out-of-n nodes can recover the CA's secret key and perform any related operation, (ii) nodes update, periodically, their secret shares. In the rest of this section, these schemes are referred to as traditional Distributed Certification Authority (DCA) model.

The simulations are carried out based on event scenarios and executed for 600 seconds. During this time, the arrival of certification requests is modeled by a Poisson distribution with an average interarrival time of 10 seconds. Simulation parameters are summarized in Table 1.

6.1 Effect of network size and DoS attacks on robustness

In this experiment, we evaluate the robustness of the certification system with respect to both network size increasing and DoS attacks. We consider a mobile adversary with a given capacity $T_{\mathcal{A}} = x/\rho$, which means the ability to compromise x nodes within a period of time ρ. To evaluate the robustness of traditional DCA models, we measure the vulnerability window ω with respect to an increasing network size, as well as the number of attacked nodes by DoS. We consider an inter-update time $\Delta = 120$ seconds (secret share refreshment). The results show that, in both cases ($n \geq 2t-1$) and ($n < 2t-1$), increasing the network size and DoS attacks, results in an expansion of the vulnerability window ω, as shown in Figs. 6 and 7. This is due to the fact that, the increase of both the DoS attacks and the network size will slow down the key update process, and the vulnerability window will be increased accordingly. Therefore, the traditional DCA schemes become less robust

Table 1 Simulation parameters

Parameters	Values
Simulation time	600 seconds
Network area	600 m × 600 m
Wireless channel	802.11b
Bitrate	11 Mbps
Packet size (byte)	1500
Transmission range	[100 m – 250 m]
Channel errors	0%
Number of nodes (max)	200
Mobility	Random way point
Pause time (second)	[5-20]
Speed (m/s)	[0–20]

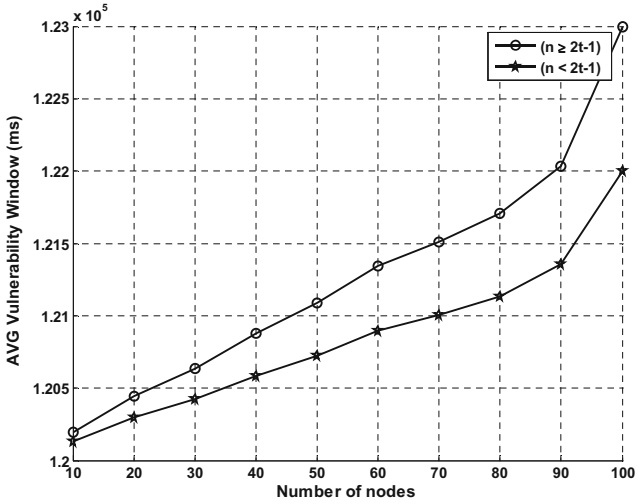

Fig. 6 Effect of network size on robustness in traditional DCA schemes

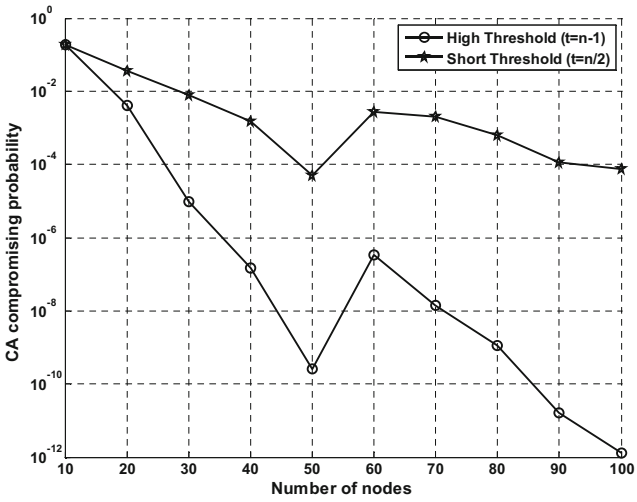

Fig. 8 Effect of network size on robustness in our scheme

when the network size and/or DoS attacks increase. This is because, when the vulnerability window increases, the capacity of the adversary ($T_{\mathcal{A}}$) become sufficient for allowing him to compromise a threshold number of nodes within a vulnerability window, and, then, compromise the certification system.

Furthermore, we evaluate the robustness of our scheme by measuring the probability of compromising the ACA with respect to the network size. However, we don't measure the vulnerability window, because our scheme is not refreshment based. In this experiment, we set $T_{\mathcal{A}} = t/\omega$. Obviously, with such an adversary capacity, the traditional DCA schemes are all beaten. However, our scheme resists to such an adversary as it can be seen from the results in Fig. 8, where it appears clearly that the compromising probability decreases with the increasing of the network size. For instance, with 100 nodes, the compromising probability become negligible. Another striking result is that even when we increase the adversary capacity up to $(t + t/2)/w$ at $N = 60$, the compromising probability marks a marginal rise and then decreases contentiously as the number of nodes increases. Furthermore, it is important to note that a high threshold (e.g. $t = n - 1$) provides more robustness, but even with a small threshold (e.g. $t = n/2$), our scheme maintains its robustness as shown in Fig. 8. Table 2 shows a comparison of our scheme versus traditional DCA model, and as we can see, the traditional DCA schemes fall out from 70 nodes in the network, while our scheme is still uncorrupted with a high probability.

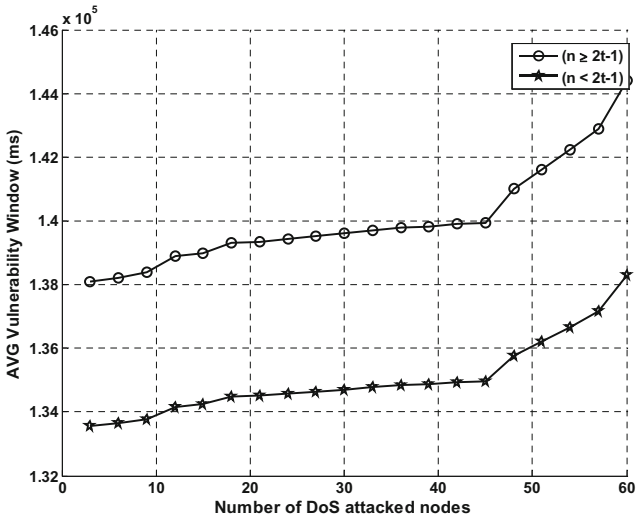

Fig. 7 Effect of DoS attacks on robustness in traditional DCA schemes

6.2 Effect of the threshold parameter on the successful certification ratio

The threshold value t has a very important impact on both robustness and availability of the certification system. A very large threshold value ensures high robustness, but the service availability and delays may not be satisfactory. Indeed, if the threshold is small, a node may get a certificate

Table 2 Robustness comparison: our scheme vs. traditional DCA (adversary capacity = t nodes per 120800 ms)

Network size	Traditional DCA	Our scheme
10	Not compromised	Not compromised
40	Not compromised	Not compromised
70	Compromised	$Pr(t) = 0.002$
100	Compromised	$Pr(t) = 7.703\text{e-}05$

Table 3 Required threshold with respect to adversary capacity and its impact on SCR

	Traditional DCA ($N = 100, n = N/2$)		Our scheme ($N = 100, m = 5, n \approx N/m$)	
Adversary capacity (x/ω)	Required threshold	SCR	Required threshold	SCR
9	10	0.90	6	0.96
19	20	0.75	10	0.91
29	30	0.69	12	0.84
39	40	0.52	14	0.83

easily and rapidly, however, the adversary can also easily collect enough shares and breaks the DCA security.

Therefore, in this section, we examine the effect of secure threshold values on the SCR. To that end, we first increase gradually the adversary capacity ($T_{\mathcal{A}}$) while determining the required minimum threshold. Then we measure the SCR with respect to these threshold values.

It is obvious that, in traditional DCA schemes, the required minimum threshold should be $t = x + 1$ to thwart an adversary with a capacity ($T_{\mathcal{A}} = x/\omega$), while in our scheme, this requires the smallest threshold value that satisfies a negligible compromising probability ($Pr(x) < 0.01$).

Table 3 compares the required minimum thresholds of our scheme to those of traditional DCA schemes, for different adversary capacities. As we can see, for the same adversary capacity as for traditional DCA, our scheme requires smaller thresholds, and hence the SCR is better in our scheme, as shown in Fig. 9.

Moreover, even when we increase the transmission range of nodes, the SCR is still better in our approach as shown in Fig. 10.

6.3 Compromise-tolerance evaluation

In this subsection, we evaluate the compromise-tolerance of our scheme, against coalition-colluding attacks with t or more compromised nodes. To that end, we measure the probability of compromising the ACA when the nodes are increasingly compromised. The measurements are carried out for different values of the threshold and the number of coalitions. In a first experiment, we set a fixed threshold $t = 20$ while varying the number m of coalitions between 3, 4 and 5. We obtained the results shown in Fig. 11, and as we can see, our scheme tolerates the compromise of a large number of nodes with negligible compromising probability ($Pr(x) \leq 0.001$). More importantly, the compromise tolerance can be enhanced by increasing the number of coalitions in the network. For instance, as it can be seen in Fig. 11, the tolerated compromise rate (with

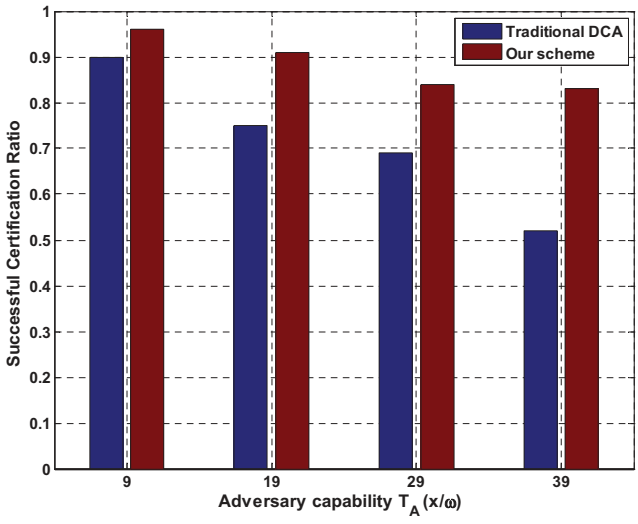

Fig. 9 Impact of security requirement on the SCR

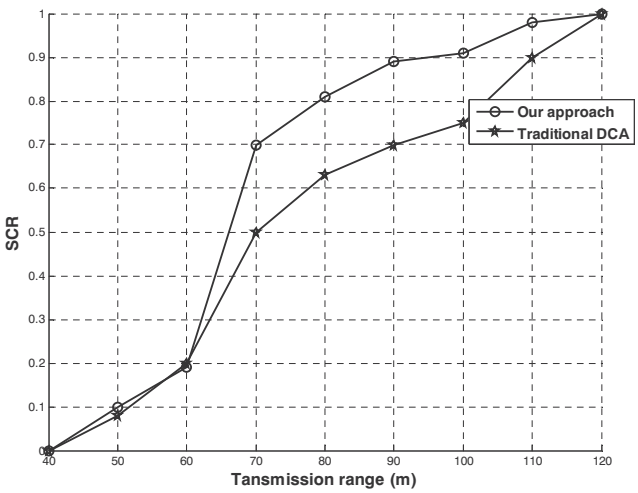

Fig. 10 Impact of transmission range on the SCR

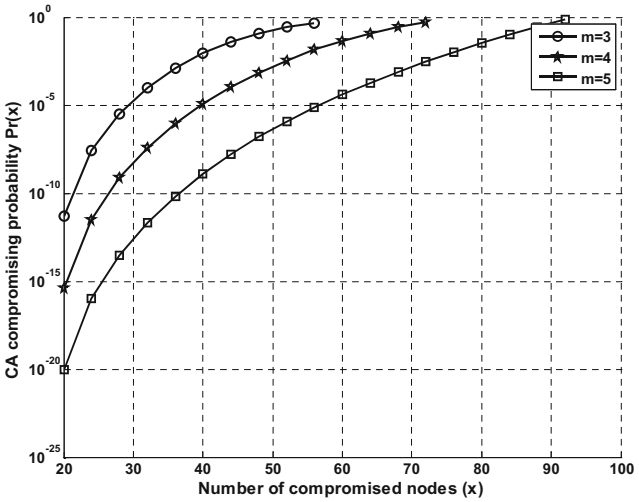

Fig. 11 Compromise tolerance in presence of compromised nodes for different coalition configurations ($N = 100, t = 20, x \leq (t-1)m$)

$Pr(x) \approx 0.001$) rises from 36% to 72% when the number of coalitions is increased from 3 to 5.

In a second experiment, we set the number of coalitions $m = 3$ and consider three values of the threshold: 10, 20 and 30. Obtained results, presented in Fig. 12, demonstrate the considerable improvement for compromise tolerance in our scheme. In particular, we can notice that, the increase of the threshold ensures a better compromise tolerance rate which reaches 70% (with $Pr(x) = 0.004$) for a threshold $t = 30$ ($t \approx N/3$).

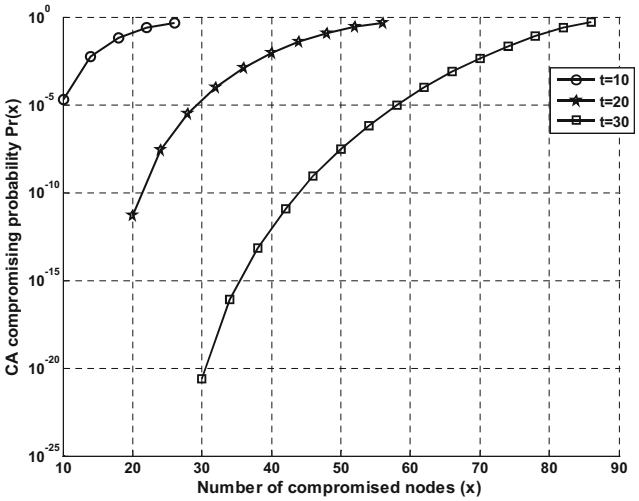

Fig. 12 Compromise tolerance in presence of compromised nodes for different threshold lengths ($N = 100, m = 3, x \leq (t-1)m$)

On the other hand, to compare our scheme to traditional DCA schemes with respect to the compromise tolerance, we set a fixed threshold $t = 10$ for a network of 100 nodes and measure the CA's compromise probability in both approaches. As we can see in the results shown in Table 4, the traditional DCA tolerates only up to $x < t$ (where x is the number of compromised nodes within a vulnerability window), while our approach can tolerate more than t compromised nodes with a very small probability that decreases when the number of coalitions increases.

Therefore, we conclude that our scheme is x-compromise-tolerant ($x \geq t$) with a non-negligible probability, compared to traditional DCA schemes which are $(t-1)$-compromise-tolerant.

6.4 The effect of coalition's configuration

In general, for best service availability, a small threshold is recommended. However, this may reduce the robustness if we consider a powerful mobile adversary. However, in our scheme, we can ensure the robustness even with a small threshold, and this can be done by increasing the number of anonymous valid coalitions. Therefore, we examine the effect of coalition's configuration, by measuring the probability of compromising the ACA with respect to the number of valid coalitions. We make this experiment with 100 and 200 nodes respectively. We choose a small threshold ranging from 10 to 20, and assume an adversary with a capacity between $10/\omega$ and $20/\omega$. Results in Fig. 13 show that the robustness can, indeed, be enhanced when we increase the number of coalitions, and this, even when we increase the capacity of the adversary. For instance, when $T_{\mathcal{A}} = 15/\omega$, we found that, the best configuration for the best robustness in this experiment is ($t = 14, m = 7$). More importantly, we found that, even when the network size increases from 100 to 200 nodes, the robustness still getting enhanced with the increasing of valid coalitions number, as shown in Fig. 14. This means that, in our scheme, one can choose a small threshold, to have a good service availability, without affecting the security of the CA.

6.5 The effect of leave operations on availability

Many nodes may stop cooperating because they are being revoked or simply they leave the network. In our scheme, the system coalitions are updated dynamically to handle these changes. We investigate this aspect in our simulation to measure how available the system is when nodes are increasingly outgoing. Overall, our results show that leaving the network does not affect significantly the availability, as it can be seen in Table 5.

Table 4 Compromise tolerance : Our approach vs. traditional DCA

	# of coalitions	The compromise probability		
		$x < t$	$x = t$	$x > t\,(x = 20)$
Traditional DCA ($n = N$)	1	0	1	1
Our approach	1	0	1	1
	3	0	1.6041e-005	0.12
	6	0	2.7757e-009	2.4751e-004
	9	0	5.7191e-012	9.3924e-007

6.6 Robustness evaluation of the pseudonym Authority

To deal with unfair/malicious revocations, either from compromised or honest-but-curious PA members, the threshold value k used by the PA should be large enough. Although the PA is based on basic (k, n)-threshold as in traditional DCA, it has not, however, the same security problems. Indeed, increasing the PA's threshold does not influence the overall certification service availability. This is because we make the role of PA to be limited, mainly, to revocation which is infrequently occurred compared to certificate renewal and generation. In contrast, the DCA is responsible for all certification and revocation operations, and obviously the increase of the threshold has an important impact on the overall certification service availability and delays. In order to evaluate these assumptions, we assess the extent to which the threshold could be increased to meet the required security level without affecting the Successful Service Ratio (SSR) in both the PA and traditional DCA. The results in Table 6 highlight that for an acceptable SSR ($\geq$ 85%), the PA may increase the threshold up to $k = 3N/4$, while the DCA cannot go beyond a threshold $k = N/3$ within a network of 100 nodes. As we can see, for the same network size and same SSR, the PA with $k = 3N/4$ may withstand compromising attacks, which are 20% more powerful than can do the DCA with $k = N/8$. As a result, for our model validation, we set $k = 3N/4$ which we believe that it has not a significant effect on the performance of the overall approach.

7 Discussion

Most of proposed DCA schemes have focused on efficiency and infrastructureless issues, but they did not pay more attention to the security of the DCA itself. As shown in our reported results, the robustness of DCA-based schemes is

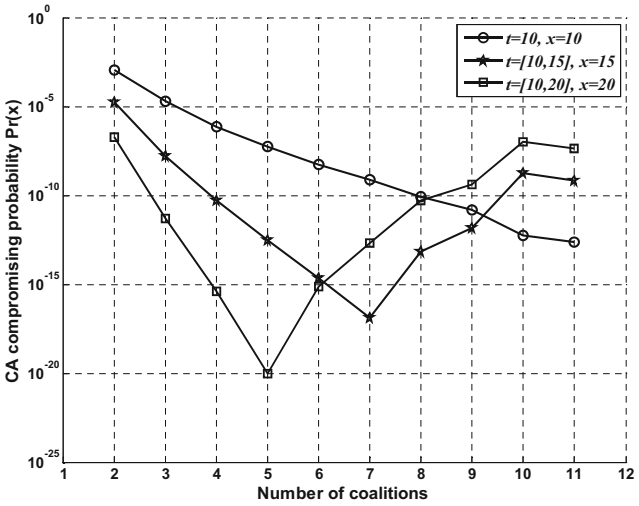

Fig. 13 Coalition's configuration impact ($N = 100$)

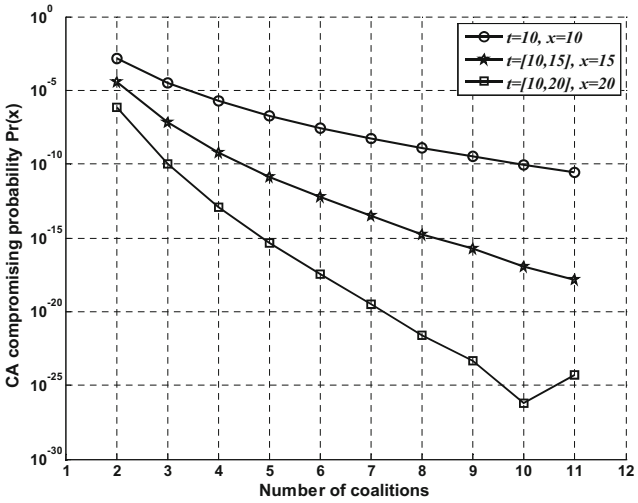

Fig. 14 Coalition's configuration impact ($N = 200$)

Table 5 Impact of network leave operation ($N = 100, m = 5, n = N/m, t = 10$)

AVG interleave time (s)	AVG rate of outgoing nodes (%)	SCR
30	25	0.95
25	27	0.95
20	31	0.94
15	43	0.93
10	54	0.88

breaking down when the network is scaled up, especially with a threshold that is not large enough. On the other hand, increasing the threshold affects the availability as we can observe in the performance analysis where the SCR turns down to 0.52 when the threshold is increased to ($t = 40, n = 50, N = 100$). In that respect, our approach shows a clear advantage over traditional DCA schemes by ensuring a significant positive compromise between security and availability. Our experimental study highlights that, for the same security level as for traditional DCA, one can reduce the threshold to maximize the SCR, while keeping a high security against DCA compromising. For instance, our scheme can resists 50% more powerful attacker than traditional scheme with an SRC = 0.91. These positive results takes more significant when the network is scaled up. The compromise tolerance is also enhanced by preventing the attacker from compromising the DCA even if a large number (that may exceeds the threshold) of nodes are compromised. More specifically, our approach allows up to t' compromised nodes with a negligible probability, where $t \leq t' \leq (t-1)m$ and m is the number of coalitions.

On the other hand, security reinforcement comes always with a cost. Theoretically, our approach seems to require slightly longer delays for certificate issuance than those of traditional schemes for the same threshold t, particularly, if t nodes are not neighbors. This is due, indeed, to the fact that t nodes should be from the same coalition. However, in practice, delays depend on many other factors such as the underlying communication protocols. Indeed, the latter has a significant impact on the delays whatever the certification method used. Furthermore, one can reduce the threshold in our approach, which consequently reduce the delays without affecting the robustness.

8 Conclusion

In this paper, we proposed a secure distributed certification management scheme with Anonymous Certification Authority (ACA). The proposed scheme introduces a new threshold certification model which is intended to enhance the system's security and robustness. In this new model, certification tasks are achieved anonymously with the help of a distributed anonymous authority. This means that, no one can identify or locate the signing nodes of a given certificate. Furthermore, although the role of the ACA is distributed amongst all network nodes, certification tasks may be achieved only by specific t-out-of-n subsets of nodes called valid coalitions. This means that, if an adversary controls a subset of t nodes, this does not mean that he can recover the system's key and compromise the CA. The adversary is challenged to locate and attack selectively a valid coalition of the ACA members in order to compromise the whole ACA. However, this is a quite difficult task, because our scheme ensures that no meaningful information is revealed to the adversary about the coalition formation. Subsequently, with t or more compromised nodes, the probability that the ACA becomes compromised is negligible. With these important features, the suggested approach provides robust and secure solution against coalition-colluding attacks which are considered as an important issue that is not solved by current schemes. In order to prevent impersonation, we proposed a protocol for pseudonym verification which allows verifying the validity of anonymous signers without revealing their identities. We also proposed a threshold revocation scheme whereby misbehavior ACA anonymous members are traced back and revoked. Through a MATLAB simulation, we highlighted the strengths of our proposed scheme, compared to exiting relevant works.

Table 6 Robustness comparison: PA vs. traditional DCA

Threat level $T_{\mathcal{A}}$ (x/w)	Suitable secure threshold	SSR (DCA)	SSR (PA)
10	$N/8$	0.90	0.96
30	$N/3$	0.69	0.92
40	$N/2$	0.49	0.88
50	$3N/4$	0.43	0.85

In a feature work, we plan to extend our anonymity mechanism to deal with the location privacy preserving issue in Ad-hoc networks. Indeed, location privacy is a major concern in these environments especially in applications such as sensor networks and vehicular networks.

References

1. Beimel A (2011) Secret-sharing schemes: a survey. In: Coding and cryptology - third international workshop, IWCC 2011, Qingdao, China, May 30–June 3, 2011. Proceedings, pp 11–46
2. Camenisch J, Shoup V (2003) Practical verifiable encryption and decryption of discrete logarithms. In: Advances in cryptology - CRYPTO 2003, 23rd annual international cryptology conference, Santa Barbara, California, USA, August 17–21, 2003, Proceedings, pp 126–144
3. Chen Z, Li S, Wu Q, Huang Q (2015) A distributed secret share update scheme with public verifiability for ad hoc network. Security and Communication Networks 8(8):1485–1493
4. Guo Y, Ma J, Chao W, Yang K (2013) Incentive-based optimal nodes selection mechanism for threshold key management in manets with selfish nodes. Int J Distrib Sens Netw, 2013
5. Hamouid K, Adi K (2010) Secure and robust threshold key management (SRKM) scheme for ad hoc networks. Security and Communication Networks 3(6):517–534
6. Hamouid K, Adi K (2015) Efficient certificateless web-of-trust model for public-key authentication in MANET. Comput Commun 63:24–39
7. Kobayashi K, Totani Y, Utsu K, Ishii H (2016) Achieving secure communication over MANET using secret sharing schemes. J Supercomput 72(3):1215–1225
8. Li L, Liu R (2010) Securing cluster-based ad hoc networks with distributed authorities. IEEE Trans Wirel Commun 9(10):3072–3081
9. Maity S, Hansdah R (2014) Self-organized public key management in manets with enhanced security and without certificate-chains. Comput Netw 65(0):183–211
10. Meng X, Li Y (2012) A verifiable dynamic threshold key management scheme based on bilinear pairing without a trusted party in mobile ad hoc network. In: 2012 IEEE international conference on automation and logistics, Zhengzhou, China, August 15-17, 2012, pp 315–320
11. Omar M, Boufaghes H, Mammeri L, Taalba A, Tari A (2016) Secure and reliable certificate chains recovery protocol for mobile ad hoc networks. J Netw Comput Appl 62:153–162
12. Park Y, Park Y, Moon S (2013) Anonymous cluster-based manets with threshold signature. Int J Distrib Sens Netw 2013
13. Pedersen TP (1991) Non-interactive and information-theoretic secure verifiable secret sharing. In: Advances in cryptology - CRYPTO '91, 11th annual international cryptology conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings, pp 129–140
14. Pedersen TP (1991) A threshold cryptosystem without a trusted party (extended abstract). In: Advances in cryptology - EUROCRYPT '91, workshop on the theory and application of of cryptographic techniques, Brighton, UK, April 8-11, 1991, Proceedings, pp 522–526
15. Schnorr CP (1990) Efficient identification and signatures for smart cards. Springer, Berlin, pp 239–252
16. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
17. Yao L, Deng J, Wang J, Wu G (2015) A-CACHE: an anchor-based public key caching scheme in large wireless networks. Comput Netw 87:78–88
18. Zhou L, Haas Z (1999) Securing ad hoc networks. Network, IEEE 13(6):24–30
19. Zhou L, Schneider FB, van Renesse R (2005) APSS: Proactive secret sharing in asynchronous systems. ACM Trans Inf Syst Secur 8(3):259–286
20. Zhu B, Bao F, Deng RH, Kankanhalli MS, Wang G (2005) Efficient and robust key management for large mobile ad hoc networks. Comput Netw 48(4):657–682

Khaled Hamouid is a lecturer at Computer Science department, University of Batna 2. He got his Ph.D. degree conjointly from university of Bejaia (Algeria) and university of Quebec in Outaouais (Canada). He is member of both LRSI laboratory (UQO, Canada) and LaSTIC research laboratory (Univ. of Batna 2). His research interests include security, trust and privacy in P2P, Mobile and Wireless Networks.

Kamel Adi is a full professor at Université du Québec en Outaouais, Canada. He is the director of Computer Security Research Laboratory. His research activities focus on the development and application of formal methods and rigorous resolution of issues related to the security of computers and computer networks: cryptographic protocols, electronic commerce protocols, intrusion detection, firewall, detection malicious code, etc.