Available online at www.sciencedirect.com

ScienceDirect

Computer Networks 51 (2007) 4938–4950

Computer Networks

www.elsevier.com/locate/comnet

Cryptographic techniques for mobile ad-hoc networks

Vanesa Daza [a], Javier Herranz [b], Paz Morillo [c,*], Carla Ràfols [c]

[a] *Dept. d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili (URV), Av. Països Catalans, 26, 43007 Tarragona, Spain*
[b] *IIIA, Artificial Intelligence Research Institute; CSIC, Spanish National Research Council Campus UAB s/n, 08193 Bellaterra, Spain*
[c] *Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya (UPC), C/Jordi Girona, 1-3, Campus Nord,*
Mòdul C-3, 08034 Barcelona, Spain

Received 15 January 2007; received in revised form 11 June 2007; accepted 16 August 2007
Available online 26 August 2007

Responsible Editor: X.S. Shen

Abstract

In this paper, we propose some cryptographic techniques to securely set up a mobile ad-hoc network. The process is fully self-managed by the nodes, without any trusted party. New nodes can join the network and are able to obtain the same capabilities as initial nodes; further, each node can obtain a pair of secret/public keys to secure and authenticate its communication. Two additional features of our system are that it allows to implement threshold operations (signature or decryption) involving subgroups of nodes in the network and that any subgroup with a small number of nodes (between 2 and 6) can obtain a common secret key without any communication after the set up phase.

Keywords: Mobile ad-hoc networks; Secret sharing techniques; Identity-based cryptography; Threshold cryptography

1. Introduction

A mobile ad-hoc network (also known as MANET) is a self-organized wireless network of mobile nodes without any fixed infrastructure. Nodes roam through the network, causing its topology to change rapidly and unpredictably over time. New nodes can join the network, whereas at the same time other nodes leave it or just fail to connect (temporarily) because they move to a region that is not in the cover range of the network. Nodes are typically wireless devices such as PDAs, laptops or cellular phones. From the very beginning, the use of MANETs has been appealing for both military and civilian applications, specially in the last decade because of the development of wireless LAN technology.

MANETs are also characterized for being bandwidth and energy constrained (nodes are often battery-powered devices) and are specially prone to the security threats of eavesdropping, interception, denial-of-service and routing attacks. Some of these problems may be solved or mitigated with the use of cryptographic protocols. In the recent literature many papers make specific proposals on how to

[*] Corresponding author. Tel.: +34 934016008; fax: +34 934015981.
E-mail addresses: vanesa.daza@urv.cat (V. Daza), jherranz@iiia.csic.es (J. Herranz), paz@ma4.upc.edu (P. Morillo), crafols@ma4.upc.edu (C. Ràfols).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.08.002

use well-known cryptographic techniques to secure MANETs, although the problem is far from being solved.

As an example of MANET, let us consider multiplayer computer games. With the increasing amount of mobile devices, multiplayer computer games are getting very popular. In such games, the set of players is changing during the game, the players can join or leave the game at any time, there are different teams, etc. In some of them, the decision about player admission or game strategy is taken only when a certain number of members agree. Some transmitted information has to be protected (encrypted) against other players or teams.

1.1. Our contribution

In this paper, we propose how to set up a MANET with the following properties:

1. the process is as decentralized as possible, i.e. the nodes themselves manage the whole life of the MANET;
2. new nodes can join the MANET at any moment, getting the same capabilities as the initial nodes, if desired;
3. each node can obtain a secret key/public key pair to be used in the remaining (possibly long) life of the MANET, for example to sign or decrypt messages.

The first property is achieved by using secret sharing techniques. At the end, the MANET will have a secret key/public key pair (SK, PK) such that PK is public and each node N_i holds a share $[\mathrm{SK}]_i$ of the secret key SK.

The second property comes from the use of symmetric bivariate polynomials to allow dynamic sets of nodes. Specifically, a new node N_j must contact other nodes in order to obtain its secret information, in particular its share $[\mathrm{SK}]_j$.

With respect to the third property, the obtention of individual secret/public keys, we propose two possibilities, depending on the kind of scenario where these keys are going to be used. The first one is the well-known PKI-based scenario: each node N_i can individually generate its pair (sk_i, pk_i) of secret/public keys. Then, the node must contact other nodes, which will jointly compute a valid certificate linking the identity of N_i with the public key pk_i. An alternative is to consider the identity-based scenario where the public key pk_i of each node N_i can be derived (in a public and efficient way) directly from its identity N_i. Later, node N_i can obtain the secret key sk_i which matches with pk_i by contacting some master entity. In our system, the role of this entity is distributed so that node N_i, after contacting some other nodes will be able to obtain sk_i. Our system admits some interesting extensions to threshold cryptography. Suppose the MANET is divided into different subgroups of nodes according to a common characteristic (for example, members of the same team in multiplayer computer games). Then we show how our proposal can be extended to allow threshold decryption; that is, a message intended for a certain group $\mathtt{SG}$ can be decrypted only if enough nodes in $\mathtt{SG}$ cooperate. To do this, we need an extra variable to be added to the aforementioned bivariate polynomial. Similar ideas can be used to implement also threshold signatures on behalf of the subgroup (the verifier is convinced that a number of members of $\mathtt{SG}$ have cooperated to jointly compute the signature). Furthermore, our proposal allows to reduce the number of nodes in $\mathtt{SG}$ necessary to decrypt if the sender considers that a weaker level of security is enough for a specific message.

Finally, we show that groups of up to four members in the PKI-scenario and up to six in the identity-based scenario share a common secret key without any further communication after the initialization step.

1.2. Related work

One of the main issues when applying cryptography to MANETs is how to distribute the role of the trusted authority among the nodes. Therefore, most proposals to secure mobile ad-hoc environments make use of some secret sharing technique to distribute the key of this trusted entity. The first proposal was due to Zhou and Haas [1]. They used threshold cryptography to distribute the role of the Certification Authority (CA) in a PKI scenario among a set of selected servers. However, this proposal is not suitable for a purely ad-hoc environment where those selected nodes may not always be available. Kong et al. [2] adapted this idea to distribute trust among all of the nodes. However, their specific RSA threshold scheme has been proved to be insecure [3,4]. Other works [5–7] consider an identity-based scenario and distribute the role of the master entity.

All these works, whether in a PKI or an ID-based scenario, make use of Shamir's secret sharing scheme; this implies that new nodes joining the MANET receive some secret information, but either they do not obtain the same capabilities as the initial ones or they do it at the cost of a lot of interaction among some existing nodes. We overcome this limitation with the use of other secret sharing techniques (which employ bivariate or trivariate polynomials), as we will see in Section 3.

Bivariate polynomials have already been used in other works to dynamically allow new nodes joining the network without the need of any external trusted party, inspired on the original work of [8]. Some works [9,10] constructed decentralized flexible dynamic group key distribution schemes by means of using polynomials in two variables. The goal is to generate common group secret keys. Saxena et al. [11] used this technique to establish pairwise keys in a non-interactive way in a mobile ad-hoc scenario. We stress that their work focuses on symmetric cryptography. A good explanation on the advantages of bivariate polynomials can be found in [11].

Other works that consider distributed cryptography over MANETs are the threshold signature schemes of [12–14]. They also make use of secret sharing techniques, although their approach is different to ours. In the case of [12,13] the shared secret changes according to the set of nodes that cooperate to produce a signature, while in the case of [14] a trusted dealer is necessary in the initialization step.

Finally, a work with similar goals than ours can be found in [15]. There, the authors propose how to create a self-organized public key infrastructure in a mobile ad-hoc network, without any third trusted party. However, their approach is different: authentication is based on chains of certificates, and a node A signs a certificate for another node B only if A trusts B. In some sense, this approach follows the same trust-based ideas behind PGP.

1.3. Organization

The rest of the paper is organized as follows. In Section 2, we explain some cryptographic primitives which will be used in the design of our scheme. In Section 3, we describe the different phases of our scheme: initialization phase, aggregation of nodes, obtention of secret/public keys, and threshold operations involving subgroups. In Section 4, we discuss the global security of our scheme, and some possible extensions for it. Finally, we conclude the work in Section 5.

2. Preliminaries

2.1. Secret sharing schemes

The idea of secret sharing schemes was independently introduced by Shamir [16] and Blakley [17]. A secret sharing scheme is a method by means of which a special figure, called usually *dealer*, shares a secret s among a set $\mathscr{P} = \{P_1, \ldots, P_n\}$ of n parties. Each party P_i is to receive privately from the dealer a piece of information $[s]_i$ (or *share*) of the secret s. The shares of those subsets of participants allowed to recover the secret (also known as *authorized subsets*) can be used to obtain s by means of a reconstruction process. The family $\Gamma \subset 2^{\mathscr{P}}$ of such subsets is called *access structure*. For example, given a set $\mathscr{P}$ of n parties as before, an access structure can be defined as the family of sets with at least t parties; this access structure Γ is known as (t,n)-threshold access structure.

Shamir's secret sharing scheme [16] realizes (t,n)-threshold access structures by means of polynomial interpolation. Let $\mathbb{Z}_q$ be a finite field with $q > n$ and let $s \in \mathbb{Z}_q$ be the secret. The dealer picks a polynomial $P(x)$ of degree at most $t-1$, where the constant term of $P(x)$ is s and all other coefficients are selected from $\mathbb{Z}_q$, uniformly and independently at random. That is,

$$P(x) = s + \sum_{j=1}^{t-1} a_j x^j.$$

Every party P_i is publicly associated to a field element α_i. Distinct parties are mapped to distinct field elements. The dealer privately sends to party P_i the value $[s]_i = P(\alpha_i)$, for $i = 1,\ldots,n$.

Let us see that the scheme realizes a (t,n)-threshold access structure. Without loss of generality, we can assume that the set of parties willing to recover the secret s is $\{P_1,\ldots,P_t\}$. The secret s can be obtained as $\sum_{i=1}^{t} \lambda_i [s]_i$, where $\lambda_i = \prod_{j\neq i} \frac{\alpha_j}{\alpha_j - \alpha_i}$ are the Lagrange coefficients.

It is proven that any set of less than t parties obtain no information about s, that is, any secret is equally probable given their shares.

2.2. Identity-based cryptography from bilinear pairings

Identity-based cryptography was introduced by Shamir [18] as an alternative to the traditional PKI paradigm. The motivation was to get rid of the need of digital certificates, signed by some trusted certification authority, which link the identity of a user with his public key. Indeed, to avoid impersonation attacks, one must check the validity of the corresponding digital certificate before using the public key (for encrypting or for verifying a signature). The infrastructure needed to manage such certificates is very costly and this can be especially problematic in the case of MANETs.

This is no longer a problem in identity-based cryptography, since now the public key of each user can be derived, in a public and efficient way, directly from his identity (e.g. e-mail address, IP address, etc.). Therefore, the link between identity and public key is established for free, from the beginning. Later, the user must contact some master entity in order to obtain his secret key. The master entity has his own pair of secret/public keys, and uses his secret key to compute the secret keys of the users. The main drawback of this paradigm is that the master entity knows the secret keys of all the users.

Most of the identity-based cryptographic schemes which have been proposed up to now (see for example [19]) employ bilinear pairings, which are maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, for groups $\mathbb{G}$ (additive) and $\mathbb{G}_T$ (multiplicative) of the same prime order q, with the following properties:

1. Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, for all $P, Q \in \mathbb{G}$, $a, b \in \mathbb{Z}_q$.
2. Non-degenerate: $e(P, P) \neq 1_{\mathbb{G}_T}$ for all $P \in \mathbb{G}$.
3. Computable: there exists an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}$.

2.2.1. *Baek and Zheng's threshold identity-based decryption scheme*

In [20], Baek and Zheng proposed a threshold decryption scheme, which works in identity-based scenarios. The idea is to distribute the secret key $\mathrm{SK}_{\mathtt{SG}}$ of a group of users $\mathtt{SG}$ into shares $[\mathrm{SK}_{\mathtt{SG}}]_i$ among the users of the group, according to some threshold t'. Then, the (public) identity $\mathtt{ID}_{\mathtt{SG}}$ of the group can be used to encrypt a message m, leading to a ciphertext C. To decrypt such a ciphertext and recover the original message m, at least t' members of $\mathtt{SG}$ have to cooperate by using their shares of the secret key. On the other hand, less than t' dishonest users of $\mathtt{SG}$ have no information at all about the plaintext message. The scheme consists of the following protocols (for simplicity, we describe a version which does not contain any correctness checking of the partial decryptions).

2.2.1.1. Setup. An additive group $\mathbb{G}$ of prime order q (generated by some public element P) and a multiplicative group $\mathbb{G}_T$ of the same order are chosen admitting a bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Three hash functions $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_T \to \{0,1\}^l$ and $H_3 : \mathbb{G} \times \{0,1\}^l \to \mathbb{G}$ are needed, where l is the bit-length of the messages to be encrypted.

The master entity has a secret key $s \in \mathbb{Z}_q$ which is chosen at random; the matching master public key is the element $\mathrm{PK} = sP \in \mathbb{G}$.

2.2.1.2. Key generation. Assume that the group $\mathtt{SG}$ has n users, $\mathtt{SG} = \{P_1, \ldots, P_n\}$. Let t' be the decryption threshold such that $1 \leqslant t' \leqslant n$. If $\mathtt{ID}_{\mathtt{SG}}$ is the public identifier of the group, then the master entity first computes the matching secret key $\mathrm{SK}_{\mathtt{SG}}$ of the group as $\mathrm{SK}_{\mathtt{SG}} = sH_1(\mathtt{ID}_{\mathtt{SG}}) \in \mathbb{G}$. Then, he picks $R_1, \ldots, R_{t'-1}$ at random from $\mathbb{G}$, and defines the mapping

$$R(z) = \mathrm{SK}_{\mathtt{SG}} + zR_1 + \cdots + z^{t'-1}R_{t'-1} \in \mathbb{G},$$

where the variable z takes values in $\mathbb{Z}_q$. Each user $P_i \in \mathrm{SG}$ is (publicly) assigned to a different value $z_i \in \mathbb{Z}_q$, and he receives from the master entity his share $[\mathrm{SK}_{\mathtt{SG}}]_i = R(z_i) \in \mathbb{G}$ of the secret key $\mathrm{SK}_{\mathtt{SG}}$.

2.2.1.3. Encryption. Given a message $m \in \{0,1\}^l$ to be encrypted and addressed to the group $\mathtt{SG}$, the sender chooses uniformly and at random $r \in \mathbb{Z}_q$. Then he computes the value $\kappa = e(\mathrm{PK}, H_1(\mathtt{ID}_{\mathtt{SG}}))^r$ and the triple of values $U = rP$, $V = H_2(\kappa) \oplus m$, $W = rH_3(U, V)$ which define the resulting ciphertext $C = (U, V, W)$.

2.2.1.4. Threshold decryption. Given a ciphertext $C = (U, V, W)$, a member $P_i \in \mathtt{SG}$ of the group can use his secret share $[\mathrm{SK}_{\mathtt{SG}}]_i$ to compute a partial decryption, as follows. First of all, he checks if $e(P, W) = e(U, H_3(U, V))$. If this equality does not hold, he outputs 'error'. Otherwise, he computes $\kappa_i = e(U, [\mathrm{SK}_{\mathtt{SG}}]_i)$ and outputs this value as the secret partial decryption.

Let A be a set of (at least) t' users of $\mathtt{SG}$ that output their partial decryptions $\{\kappa_i\}_{P_i \in A}$. Let $\lambda_{0i}^A = \prod_{P_j \in A, j \neq i} \frac{0 - z_j}{z_i - z_j}$ be the Lagrange coefficients corresponding to this set A of users, for all $P_i \in A$. Then the players in A can recover κ as follows:

$$\begin{aligned}\prod_{P_i\in A}\kappa_i^{\lambda_{0i}^A} &= \prod_{P_i\in A} e(U,[\mathrm{SK}_{\mathtt{SG}}]_i)^{\lambda_{0i}^A} = e\left(U,\sum_{P_i\in A}\lambda_{0i}^A[\mathrm{SK}_{\mathtt{SG}}]_i\right)\\ &= e\left(U,\sum_{P_i\in A}\lambda_{0i}^A R(z_i)\right) = e(U,R(0)) = e(U,\mathrm{SK}_{\mathtt{SG}})\\ &= e(rP,sH_1(\mathtt{ID}_{\mathtt{SG}})) = e(sP,H_1(\mathtt{ID}_{\mathtt{SG}}))^r\\ &= e(\mathrm{PK},H_1(\mathtt{ID}_{\mathtt{SG}}))^r = \kappa.\end{aligned}$$

After that, the original message m is obtained as $m = V \oplus H_2(\kappa)$.

3. Our proposal

In this section, we design a method to set up a MANET. In our protocol the process is fully decentralized: the nodes themselves manage the whole life of the MANET, without any trusted third party. At any time, the new nodes joining the MANET are able to obtain the same capabilities as the initial ones after contacting only some nodes. Finally, each node can get a secret key/public key pair to be used in the remaining (possibly long) life of the MANET.

Furthermore, our system can be extended to implement threshold cryptography, namely both decryption and signature involving subgroups of nodes. We focus on the threshold decryption operation. The threshold signature case results from applying our techniques to the work in [21]. After explaining each phase of our scheme, we measure its efficiency in terms of computational and communication costs for the involved nodes.

We stress that our scheme requires at some stages authenticated and secret communication between pairs of nodes. If the involved nodes already have their pairs of secret/public keys, then they can employ well-known techniques (signature, encryption) from public-key cryptography. If not, they have to execute some protocol by means of which they authenticate each other and they obtain a common secret key to be used with some symmetric cryptosystem. Papers in the literature offer different possibilities for this task; we summarize some of these approaches:

- Some works [22–24] propose to run first a pre-authentication phase between each pair of nodes, by using some *side channel*. In this phase the nodes can agree on some short password, for example, which can be used later as the common (and authenticated) secret input for another key agreement protocol. Examples of such side channels can be a physical meeting, sending a postcard, a telephone call, or infrared communication.
- A similar approach is followed in [25], where two new ways of obtaining mutual authentication by means of some radio-channel techniques (such as distance-bounding and integrity-codes) are proposed. This work focuses on making key-agreement more user-friendly, for instance for the first protocol to be secure, it is enough that users visually verify that there are no other devices in some integrity region around them.
- Other works propose other ways to authenticate a key agreement protocol. For example, in [26] nodes can use some physical procedure to prove their identity; however, the solution is valid only for static networks. For this reason, this technique might be applied, in our scheme, only for the initialization phase, if we assume that the initial set of nodes are static during this phase.

Summing up, the problem of creating a secret and authenticated channel between any pair of nodes (without any trusted party, in particular without digital certificates) is not trivial at all, and solutions are costly and onerous. Note that many works which propose cryptographic solutions for MANETs assume at some point the existence of such channels, without discussing the costs/difficulties of implementing them.

3.1. Initialization phase

We denote as $\mathcal{N}$ the initial set of ℓnodes in the MANET; we also refer to these initial nodes as the *founding nodes* of the MANET. They jointly run the protocol described below. There are some parameters which are public: an additive group $\mathbb{G}$ of prime order q, generated by some element P, where we assume that the discrete logarithm problem (i.e., computing the integer s from the value sP) is hard. Depending on the scenario, we will need an admissible bilinear pairing $e : \mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$. Two collision-resistant hash functions $h : \{0,1\}^* \to \mathbb{Z}_q$ and $H : \{0,1\}^* \to \mathbb{G}$ are also chosen and made public. Finally, the values of the two thresholds t, t' are set. Threshold t determines the security level desired for the MANET, that is, we will assume that at most $t-1$ nodes can be dishonest. Threshold t' is chosen according to the desired security for the possible threshold operations involving subgroups of nodes. As we will see later, a necessary condition for security is $t' \leqslant t \leqslant \ell$.

We note that the admissible bilinear pairing e and the hash function H are necessary only when identity-based individual keys want to be generated or when threshold operations involving subgroups of nodes want to be allowed. Threshold t' is also only needed if we want to allow threshold operations involving subgroups of nodes.

The initialization protocol is as follows:

1. Each node $N_i \in \mathcal{N}$ chooses a random trivariate polynomial $F_i(x,y,z) \in \mathbb{Z}_q[x,y,z]$, with degree at most $t-1$ in the variables x and y, degree at most $t'-1$ in the variable z, and symmetric with respect to variables x and y. Implicitly, these polynomials define a polynomial $F(x,y,z) = \sum_{N_i \in \mathcal{N}} F_i(x,y,z)$ with the same aforementioned properties as each of the polynomials F_i. Let us denote as $f_{i,0} = F_i(0,0,0)$ the constant term of polynomial $F_i(x,y,z)$, and as $s = \sum_{N_i \in \mathcal{N}} f_{i,0} = F(0,0,0)$ the constant term of polynomial $F(x,y,z)$.
2. Each node $N_i \in \mathcal{N}$ secretly sends to each of the other founding nodes $N_j \in \mathcal{N}$ the bivariate polynomial $F_{ij}(x,z) = F_i(x,h(N_j),z)$. Furthermore, node N_i includes the value $Y_i = f_{i,0}P$ in each of these messages.
3. When the previous step is completed by each node in $\mathcal{N}$, each founding node $N_j \in \mathcal{N}$ can compute its final secret information, which is the bivariate polynomial
$$S_j(x,z) = \sum_{N_i \in \mathcal{N}} F_{ij}(x,z) = \sum_{N_i \in \mathcal{N}} F_i(x,h(N_j),z) = F(x,h(N_j),z).$$

From the information received from the rest of founding nodes, each node can compute and make public the public key of the MANET, which will be

$$\mathrm{PK} = sP = \sum_{N_i \in \mathcal{N}} f_{i,0}P = \sum_{N_i \in \mathcal{N}} Y_i.$$

Note that the (implicit) matching secret key is $\mathrm{SK} = s = F(0,0,0)$. Each founding node N_j can compute from its partial information $S_j(x,z)$ a share $[s]_j = S_j(0,0) = F(0,h(N_j),0)$ of the secret key $\mathrm{SK} = s$, corresponding to a Shamir threshold secret sharing scheme defined by the t-degree polynomial $f(y) = F(0,y,0)$.

Note also that a necessary condition to securely run this initialization phase is $t \leqslant \ell$; otherwise, if all the founding nodes were dishonest, they could obtain all the secret information of the system.

3.1.1. Commitments and verifiability

For simplicity, we have not described the most complete and secure scenario for the protocol above, where each founding node N_i broadcasts commitments to the coefficients of its polynomial $F_i(x,y,z) \in \mathbb{Z}_q[x,y,z]$. Namely, if $b^{(i)}_{mnj}$ is the coefficient multiplying $x^m y^n z^j$ in $F_i(x,y,z)$, then the node makes public the commitment $C^{(i)}_{mnj} = b^{(i)}_{mnj}P \in \mathbb{G}$. These commitments can be stored in some public database (or included in every message that nodes broadcast or exchange), and are used to verify if the secret information $F_{ij}(x,z)$ that node N_i sends to other nodes is actually consistent, following the same ideas as in verifiable secret sharing schemes [27,28]. In this way, dishonest nodes can be rejected. Furthermore, the use of these commitments provides the system with some interesting extra features, as we will explain in Sections 3.4.1 and 4.

Note that, given the commitments $C^{(i)}_{mnj}$ for a given triple (m,n,j), from all the founding nodes, then $C_{mnj} = \sum_{N_i \in \mathcal{N}} C^{(i)}_{mnj}$ is the commitment to the coefficient of the monomial $x^m y^n z^j$ in $F(x,y,z)$. In a similar way, commitments C_j to the coefficients of the t'-degree polynomials $F(0,0,z)$ can be derived from the initial commitments.

In real implementations of the system, one may require the commitments and the public key PK of the MANET to be signed with the secret key SK which corresponds to PK. In our system, these signatures could be computed by any t nodes, by means of a threshold signature scheme: each node uses its share of SK to generate a partial signature; then, t partial signatures can be combined to obtain a standard valid signature. The idea of threshold signatures will appear again in Sections 3.3 and 3.4.

3.1.2. Efficiency

Each node N_i sends $t \cdot t'$ values in $\mathbb{Z}_q$ to each other node N_j. At the end of this phase, each node N_j has to secretly store also $t \cdot t'$ values in $\mathbb{Z}_q$.

Cost of the operations is low if commitments are not used. However, if commitments are used, each node has to compute $t' \cdot \frac{t(t+1)}{2}$ modular exponentiations to compute the commitments (since the polynomial is symmetric in the first two variables we do not need to commit to all $t^2 \cdot t'$ values). In this case, each verification, by N_j, of the consistency of the received secret information, from N_i, with respect to the public commitments requires $t' \cdot \frac{t(t+1)}{2}$ modular exponentiations. We emphasize that the third variable is only necessary if we want to allow threshold group cryptography. If this is not the case these calculations are still valid taking $t' = 1$.

3.2. Node aggregation

Suppose a new node N_m wants to join the MANET after the initialization phase is executed. In order to become a full member of the MANET, with the same capabilities as a founding node, it must run the following protocol:

1. N_m selects a group $\mathcal{N}_m$ of at least t existing nodes with whom N_m is able to connect. It identifies itself as N_m and requests the nodes in $\mathcal{N}_m$ to include it in the MANET.
2. If a node $N_j \in \mathcal{N}_m$ accepts to include node N_m in the MANET, it secretly sends to N_m the polynomial
$$\begin{aligned} S_j(h(N_m),z) &= F(h(N_m),h(N_j),z) \\ &= F(h(N_j),h(N_m),z) = S_m(h(N_j),z). \end{aligned}$$
Note that here we have used the symmetry of the polynomial $F(x,y,z)$ with respect to the two first variables.
3. When node N_m receives this information from t different nodes (for simplicity, we will assume that the group $\mathcal{N}_m$ contains exactly t nodes and that all of them accept to include N_m in the MANET), it can obtain its secret polynomial $S_m(x,z)$ by using Lagrange interpolation:
$$\begin{aligned} &\sum_{N_j\in\mathcal{N}_m} \prod_{N_i\in\mathcal{N}_m, i\neq j} \frac{x-h(N_i)}{h(N_j)-h(N_i)} S_j(h(N_m),z) \\ &= \sum_{N_j\in\mathcal{N}_m} \prod_{N_i\in\mathcal{N}_m, i\neq j} \frac{x-h(N_i)}{h(N_j)-h(N_i)} F(h(N_j),h(N_m),z) \\ &= F(x,h(N_m),z) = S_m(x,z). \end{aligned}$$
4. Finally, node N_m can compute its share $[s]_m = S_m(0,0)$ of the MANET's secret key $\mathrm{SK} = s$.

If commitments are being used in the system from the initialization phase, node N_j can include the commitments C_{mnj} to the polynomial $F(x,y,z)$ in the information that it sends to N_m (step 2 above). In this way, node N_m could check that the received information is consistent (also with the information received from other nodes). If this is not the case, node N_m should contact another node.

3.2.1. Efficiency

A new node N_m must communicate with t full nodes and receive t' values in $\mathbb{Z}_q$ from each of them. Again, the operations are not too expensive if commitments are not used, since Lagrange interpolation can be done quite efficiently. If commitments are used, we are almost in the same case as in the previous section: the sponsoring nodes must include commitments to $t' \cdot \frac{t(t+1)}{2}$ values in the message and the new node must perform $t' \cdot \frac{t(t+1)}{2}$ modular exponentiations to verify the consistency of the information obtained from each sponsoring node.

3.3. Obtention of individual secret keys

We consider two possibilities for the obtention of individual secret keys, depending on whether nodes want to use PKI-based keys or identity-based keys. Note that the goal is to provide each node with a pair of secret/public keys, to be used in the long life of the MANET for signing/decrypting. The main advantage of using PKIs with respect to using symmetric cryptosystems is that each node has to store only one secret key, and not one common secret key for any other node.

3.3.1. PKI scenarios

Each node N_m can individually generate a pair $(\mathrm{sk}_m, \mathrm{pk}_m)$ of secret/public keys for some public key cryptosystem or signature scheme, like RSA. This will allow node N_m to use signature and decryption techniques in order to make its communications secure, from this moment on. However, to avoid typical impersonation attacks, it is required that some authority certifies that the secret key sk_m which matches with the public key pk_m is actually known by the node N_m. In our system, the role of this authority is shared among the nodes. They will produce a certificate linking pk_m with N_m, which in fact is a signature on the message $N_m \| \mathrm{pk}_m$, valid under the MANET's public key PK.

To do this, the nodes can use any threshold signature scheme. In such a scheme, each node N_i holding a share $[s]_i$ of the MANET's secret key can compute a partial signature on the message $N_m \| \mathrm{pk}_m$, by using this secret share, after being contacted by N_m. If the threshold signature scheme is non-interactive, this can be done without interacting with any other node. Eventually, N_m can be required to prove that it is really N_m and that it knows the secret key sk_m matching with the public key pk_m. Once t nodes have computed the partial signatures and given them to the requesting node N_m, this node can combine them to obtain a standard signature on the message $N_m \| \mathrm{pk}_m$, valid under the public key PK. Note here that if node N_m has already become a full member of the MANET, it already knows its secret information $S_m(x, z)$, in particular the share $[s]_m$, and so it must obtain only $t - 1$ valid partial signatures, because it can compute one partial signature by itself.

The final signature (or digital certificate) can be employed along with $(\mathrm{sk}_m, \mathrm{pk}_m)$, as usual in PKI-based cryptographic scenarios.

Using a non-interactive threshold signature scheme may be a more suitable option for the MANET setting, given the possible energy constraints on the nodes and the transmission cost. Taking into account that in our framework the nodes have shares $[s]_i$ of a secret key $\mathrm{SK} = s$ such that the public key is $\mathrm{PK} = sP$, we recommend to use the non-interactive threshold signature scheme proposed in [29], which perfectly fits in with this scenario. See [30] for other recent proposals of non-interactive threshold signature schemes.

3.3.2. Identity-based scenarios

The generation of the individual secret/public keys for the nodes is different in this second scenario, with respect to the first one where each node generated its pair of keys on its own. Now the public key of each node N_m is directly inferred from its identity, namely $\mathrm{pk}_m = H(N_m) \in \mathbb{G}$, where $H : \{0,1\}^* \rightarrow \mathbb{G}$ is the hash function chosen in the initialization phase. The secret key matching with this public key is $\mathrm{sk}_m = s \cdot \mathrm{pk}_m = sH(N_m)$, where s is the master secret key (also denoted as SK here). In identity-based systems with a centralized master entity, the node's secret key would be obtained by contacting the master entity. In our decentralized and self-managed system, the role of the master entity is distributed among the nodes themselves. Therefore, the node N_m will have to contact some nodes in order to obtain enough information to compute its identity-based secret key $\mathrm{sk}_m = sH(N_m)$. The details of this protocol are as follows:

1. N_m contacts a group $\hat{\mathcal{N}}_m$ of at least t full nodes. It identifies itself as N_m and requests shares of its secret key.
2. If a node $N_j \in \hat{\mathcal{N}}_m$ accepts the identification, it secretly sends to N_m the value
$$\sigma_{jm} = S_j(0,0)H(N_m) = F(0, h(N_j), 0)H(N_m) \in \mathbb{G}.$$
3. N_m needs to receive t different such values σ_{jm} (note that if node N_m is already a full member of the MANET, it can obtain one of these values for free by using its own secret polynomial $S_m(x, z)$). To simplify the notation, suppose that $\hat{\mathcal{N}}_m$ contains exactly t nodes and that all of them have sent the corresponding values σ_{jm} to N_m. Then this node can use Lagrange interpolation (with respect to the second variable of the polynomial $F(x, y, z)$) to obtain its secret key:
$$\mathrm{sk}_m = F(0,0,0)H(N_m) = sH(N_m) \in \mathbb{G}.$$

3.3.3. Efficiency

In the PKI case, a node N_m must obtain a partial signature from t other nodes. The cost will depend of the particular scheme being used. In particular the scheme of [29] is quite efficient, requiring only an evaluation of a hash function and a modular exponentiation. In the ID-based case, node N_m must also contact t nodes which will send him an element of the group $\mathbb{G}$ (typically a rational point on an elliptic curve).

3.4. Threshold operations involving subgroups

As we have already said in Section 3.1, each node N_j holds a share $[s]_j = F(0, h(N_j), 0)$ of the secret key $\mathrm{SK} = s$ of the system, corresponding to a Shamir secret sharing scheme with threshold t. Therefore, nodes can use their shares to perform some operations (signature and decryption), in such a way that the cooperation of at least t nodes is necessary to successfully complete the operation. An example is the threshold signature on the digital certificate which links a node N_m with his PKI-based public key pk_m, as we have just seen.

However, our system also allows threshold operations (signature or decryption) concerning only a subgroup of (possibly less than t) nodes in the MANET; this is actually the reason why we introduced the third variable z in the polynomials. We will concentrate in the case of threshold decryption, following the scheme due to Baek and Zheng sketched in Section 2.2.1. The same ideas can be used for identity-based threshold signatures, using the scheme in [21].

In our MANET scenario, assume that a node uses this scheme to encrypt a message for a subgroup $\mathtt{SG}$ of nodes (for example, players of a same team in a multiplayer computer game) in such a way that decryption is possible only if t' nodes of this subgroup cooperate. Now a member $N_m \in \mathtt{SG}$ willing to decrypt the message can run the following protocol to obtain its share of the secret key $\mathrm{SK}_{\mathtt{SG}}$:

1. N_m contacts a group $\tilde{\mathscr{N}}_m$ of at least t full nodes (maybe including itself, if it is already a full member of the MANET; for simplicity, we will assume that the group $\tilde{\mathscr{N}}_m$ contains exactly t members and that the process is run successfully with all of them).
2. N_m identifies itself as a member of the subgroup $\mathtt{SG}$ and requests its share of $\mathtt{SG}$'s secret key.
3. If a node $N_j \in \tilde{\mathscr{N}}_m$ accepts the identification, it secretly sends to N_m the value
$$\begin{aligned}\tau_{jm} &= S_j(0, h(N_m))H(\mathtt{ID}_{\mathtt{SG}})\\ &= F(0, h(N_j), h(N_m))H(\mathtt{ID}_{\mathtt{SG}}) \ \in \mathbb{G}.\end{aligned}$$
4. Once N_m has received t different such values $\{\tau_{jm}\}_{N_j \in \tilde{\mathscr{N}}_m}$, this node can use Lagrange interpolation (with respect to the second variable of the polynomial $F(x, y, z)$) to obtain its share
$$[\mathrm{SK}_{\mathtt{SG}}]_m = F(0, 0, h(N_m))H(\mathtt{ID}_{\mathtt{SG}}) \ \in \mathbb{G}.$$

The only thing we must prove now is that the shares $[\mathrm{SK}_{\mathtt{SG}}]_m$ of the secret key obtained in this way follow exactly the same distribution as in the sharing process of [20,21] depicted in Section 2.2.1. Indeed, as a result of the initialization phase, we have

$$F(0, 0, z) = s + a_1 z + \cdots + a_{t'-1} z^{t'-1},$$

for some values $a_i \in \mathbb{Z}_q$, for $i = 1, \ldots, t' - 1$. Now we can define the mapping $R(z) = F(0, 0, z)H(\mathtt{ID}\ _{\mathtt{SG}}) \in \mathbb{G}$, which has all the required properties: (1) $R(0) = sH(\mathtt{ID}_{\mathtt{SG}}) = \mathrm{SK}_{\mathtt{SG}}$; (2) the rest of terms are random elements in $\mathbb{G}$, namely $R_i = a_i H(\mathtt{ID}_{\mathtt{SG}})$ for $i = 1, \ldots, t' - 1$; (3) and finally, the share of $\mathrm{SK}_{\mathtt{SG}}$ corresponding to a node $N_m \in \mathtt{SG}$ is $[\mathrm{SK}_{\mathtt{SG}}]_m = F(0, 0, h(N_m))H(\mathtt{ID}_{\mathtt{SG}}) = R(h(N_m)) = R(z_m)$, as in the original sharing procedure, taking $z_m = h(N_m)$ for the different values assigned to the members of $\mathtt{SG}$.

Therefore, the nodes $N_m \in \mathtt{SG}$ can use their shares of the secret key to jointly decrypt messages addressed to $\mathtt{SG}$, or to jointly compute threshold signatures on behalf of this subgroup $\mathtt{SG}$, by using the schemes in [20,21]. Note that the condition $t' \leqslant t$ is required because, if $t < t'$, then t nodes could decrypt any ciphertext addressed to $\mathtt{SG}$ (in fact t nodes are enough to recover the implicit master secret key s and so break all the security of the system), contradicting in this way the required security property of threshold decryption schemes. This should not be a problem in practice, because the threshold t which protects the whole security of the system should be chosen large enough.

3.4.1. Decreasing the threshold t'

As explained in Section 3.1, the commitments corresponding to the polynomial $F(0, 0, z) = s + a_1 z + \cdots + a_{t'-1} z^{t'-1}$, i.e., the values $\mathrm{PK} = sP$ and $C_j = a_j P$ for $j = 1, \ldots, t' - 1$, can be derived from the initial commitments $C^{(i)}_{mnj}$. Now assume that a sender of a message to the group $\mathtt{SG}$ thinks that the threshold t' is too high, and that a weaker level of security is enough for the encrypted message that it wants to send to the group. Suppose that the desired threshold is in that case t'' satisfying $1 \leqslant t'' < t'$. Then, along with the standard ciphertext $C = (U, V, W)$ corresponding to the Baek–Zheng scheme described in Section 2.2.1, the sender must append other values. The idea is that the sender itself will compute $t' - t''$ valid partial decryptions of the ciphertext C, corresponding to some "dummy" nodes (out of $\mathtt{SG}$, for example); in this way, combining these partial decryptions with t'' new partial decryptions coming from $\mathtt{SG}$ will be enough to recover the message, by using the Baek–Zheng threshold decryption method.

The only point is how can the sender compute valid partial decryptions for these $t' - t''$ dummy nodes $N_i \notin \mathtt{SG}$. Recall that a correct partial decryption coming from such a node N_i is $\kappa_i = e(U, [\mathrm{SK}_{\mathtt{SG}}]_i)$, where $[\mathrm{SK}_{\mathtt{SG}}]_i = F(0, 0, z_i)H(\mathtt{ID}\ _{\mathtt{SG}}) \in \mathbb{G}$, if $z_i = h(N_i)$ is the value in $\mathbb{Z}_q$ publicly assigned to node N_i. We can write $H(\mathtt{ID}_{\mathtt{SG}}) = \alpha P$ for some (unknown) value of $\alpha \in \mathbb{Z}_q$. Then we have

$$\begin{aligned}[\mathrm{SK}_{\mathtt{SG}}]_i &= F(0, 0, z_i)H(\mathtt{ID}\ _{\mathtt{SG}}) = F(0, 0, z_i)\alpha P\\ &= \alpha\left[sP + z_i a_1 P + \cdots + z_i^{t'-1} a_{t'-1} P\right]\\ &= \alpha\left[\mathrm{PK} + z_i C_1 + \ldots + z_i^{t'-1} C_{t^{P}\prime-1}\right].\end{aligned}$$

Now, since the first part U of the ciphertext C is $U = rP$ for some value of $r \in \mathbb{Z}_q$, we can rewrite the partial decryption corresponding to the dummy node N_i as

$$\begin{aligned}\kappa_i &= e(U,[\mathrm{SK}_{\mathtt{SG}}]_i)\\ &= e\big(rP,\alpha\big[\mathrm{PK}+z_iC_1+\cdots+z_i^{t'-1}C_{t'-1}\big]\big)\\ &= e\big(\alpha P,r\big[\mathrm{PK}+z_iC_1+\cdots+z_i^{t'-1}C_{t'-1}\big]\big)\\ &= e\big(H(\mathtt{ID}_{\mathtt{SG}}),r\big[\mathrm{PK}+z_iC_1+\cdots+z_i^{t'-1}C_{t'-1}\big]\big).\end{aligned}$$

And this value can be perfectly computed by the sender node, which knows all the public commitments and has chosen the value r when it has generated the standard ciphertext C.

Summing up, if B denotes the employed set of $t'-t''$ dummy nodes and $C=(U,V,W)$ is the standard ciphertext corresponding to Baek–Zheng scheme with threshold t', the final ciphertext that the sender must broadcast in this case is $C'=(t'',C,\{\kappa_i\}_{N_i\in B})$. After that, t'' real nodes N_m of $\mathtt{SG}$ can use their shares $[\mathrm{SK}_{\mathtt{SG}}]_m$ of the secret key of $\mathtt{SG}$ (obtained as described in Section 3.4) to compute their partial decryptions and combine them with $\{\kappa_i\}_{N_i\in B}$ to recover the encrypted message.

3.4.2. Efficiency

A node N_m willing to obtain its share of the subgroup secret key must do roughly the same operations as in the previous section for the obtention of an ID-based key: contact t nodes and obtain an element of $\mathbb{G}$ from each of them, then use Lagrange interpolation. Further, the threshold group operations of Baek and Zheng's scheme require the use of pairings, which are quite expensive, although a lot of progress in their implementation has been made. Decreasing the threshold for group operations to t'' requires adding $t'-t''$ components to the ciphertext and also implies that the sender must compute $t'-t''$ pairings.

4. Security and extensions

4.1. Security discussion

In [8], Blundo et al. showed how to use bivariate polynomials $F(x,y)$ to share a secret and also to set up a pairwise key agreement in a dynamic group. The common key of parties i, j was the value $F(i,j)$ and both the secret sharing scheme and the key agreement protocol were proven unconditionally secure, that is, it was proven that non-authorized parties obtained no information on the secret or the shared key. On the other hand, the security of Baek and Zheng's threshold decryption scheme relies on the standard bilinear Diffie Hellman assumption, and the security proof can be found in [20]. The security of our proposal follows immediately from these two papers.

It is important to stress that, in order to ensure security of our scheme, a secure and authenticated channel must be established in the first step of the communication between every pair of nodes. We have discussed some existing methods to achieve such a channel at the beginning of Section 3.

4.2. Proactive security

For long-lived MANETs it may become necessary to use proactive secret sharing in order to "refresh" the shares of the secret (but not the secret). Indeed, the assumption that less than t nodes are not corrupted for the entire life of the MANET may be too strong if the MANET has a long lifetime. In this case, a more reasonable security assumption is that not more than t nodes are simultaneously dishonest in a shorter interval of time. To deal with such scenarios the notion of proactive secret sharing was proposed in [31]. The basic idea is that, if the lifetime of the MANET is divided into different periods, at the end of each of them a set of at least t nodes execute the secret sharing scheme as described in Section 3.1, but for the secret $s=0$. The new shares are now the sum of the old ones with the new shares of 0. The shared secret is the same, because the shares of the nodes implicitly define a different polynomial which takes the same value at zero.

Note that if we use verifiable secret sharing then, with our scheme, it is possible to sign the commitments to the coefficients of the new polynomial in a threshold way, ensuring its integrity and authenticity. Thus our scheme can be extended for long-lived MANETs at a reasonable cost.

4.3. Key agreement

Our system allows every pair of nodes N_i, N_j to agree on a symmetric key $d_{ij}=F(h(N_i),h(N_j),0)\in\mathbb{Z}_q$ as noted in [8]. Note that if we are in an identity-based scenario where $\mathbb{G}$ admits pairings, then two nodes N_i, N_j who have already obtained their secret keys (for example, $\mathrm{sk}_i=\mathrm{sH}(N_i)$) also share the private information

$$K_{ij}=e(sH(N_i),H(N_j))=e(H(N_i),sH(N_j)),$$

in a non-interactive way. If the identities of the nodes include some expiry date, since K_{ij} depends on the identities N_i, N_j, the key K_{nm} changes for every period. The same happens to the key d_{ij}, but only if the feature of proactive security is considered. On the other hand, the security of d_{ij} is unconditional, while the security of K_{ij} is based on computational assumptions.

If commitments are used when implementing our scheme, some other group secret keys (for groups of more than two nodes) can be defined and computed in a non-interactive way. For example each group of three or four nodes will have at least a common secret key. To see this, just note that each node N_i has a secret information $F(h(N_i),0,0) = [s]_i$, and shares with any other node N_j the secret key $d_{ij} = F(h(N_i), h(N_j),0)$. Also note that the values $[s]_iP$ and $d_{ij}P$ can be publicly derived from the commitments to the coefficients of the polynomial F. Therefore, nodes N_i, N_j, N_k can agree on the key $d_{ij}[s]_kP$ (or on $d_{ik}[s]_jP$, $d_{jk}[s]_iP$, the decision on which of the three keys is being use should be defined by a MANET rule). Note that, in order to compute this key, knowledge either on d_{ij} and $[s]_kP$ or on $d_{ij}P$ and $[s]_k$ is necessary, and this is only the case for nodes N_i, N_j and N_k. Following the same idea, it is clear that a group of four nodes N_i, N_j, N_k, N_l share the key $d_{ij}d_{kl}P$.

When using groups $\mathbb{G}, \mathbb{G}_T$ for which parings exist, this idea can be extended to groups of up to six members. As an example, consider the group of five nodes N_i, N_j, N_k, N_l, N_m. This group shares, for example, $K_{ijklm} = e(d_{ij}P, d_{kl}P)^{[s]_m}$. Because of the bilinear properties of the pairing this is equal to $K_{ijklm} = e(d_{ij}P, [s]_mP)^{d_{kl}} = e(d_{kl}P, [s]_mP)^{d_{ij}}$. To compute K_{ijklm}, knowledge of one of the integers $d_{ij}, d_{kl}, [s]_m \in \mathbb{Z}_q$ is required, while the inputs of the pairing, in $\mathbb{G}$, can be publicly derived from the commitments.

5. Conclusion

When dealing with mobile ad-hoc networks (MANETs), one usually wants the network to work without the presence of any trusted third party. A natural way of achieving this property is by sharing among the nodes the role that such a third party would play. For this, an essential tool is the use of secret sharing techniques. However, the use of standard secret sharing techniques makes dynamism difficult to achieve.

In this paper, we present a scheme which overcomes this limitation with the use of bivariate polynomials. Although these techniques are well known and proven useful to provide dynamism in admission control, they had not been used to distribute the role of a trusted authority in asymmetric cryptography. One of the most important contributions of this paper is to point out the relevance of these techniques to this setting.

As a result, we propose a scheme that achieves decentralization and full dynamism, as well as other interesting and desirable properties for a MANET: each node can obtain a pair of secret/public keys for its own use, threshold operations involving subgroups of nodes can be implemented, small groups of nodes can compute common secret keys, and the system can enjoy proactive security, which leads to a long-lived MANET.

Acknowledgments

The work of the two first authors was partially supported by Spanish *Ministry of Education and Science*, under projects SEG2004-04352-C04-01 (“PROPRIETAS”) and CONSOLIDER CSD2007-00004 (“ARES”). The work of the first author was also supported by the Government of Catalonia under grant 2005 SGR 00446 and by the UNESCO *Chair in Data Privacy*. This author is solely responsible for the view expressed in this paper, which does not necessarily reflect the position of UNESCO nor commits that organization. The work of the two last authors was partially supported by Spanish *Ministry of Education and Science*, under projects TSI2006-02731 and CONSOLIDER MTM2005-24556-E.

References

[1] L. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Network 13 (6) (1999) 24–30.

[2] H. Luo, J. Kong, P. Zerfos, S. Lu, L. Zhang, URSA: ubiquitous and robust access control for mobile ad hoc networks, IEEE/ACM Transactions on Networking 12 (6) (2004) 1049–1063.

[3] M. Narasimha, G. Tsudik, J.H. Yi, On the utility of distributed cryptography in P2P and MANETs: the case of membership control, in: Proceedings of ICNP’03, 2003, pp. 336–345.

[4] S. Jarecki, N. Saxena, J.H. Yi, An attack on the proactive RSA signature scheme in the URSA ad hoc network access control protocol, in: Proceedings of the SASN’04, 2004, pp. 1–9.

[5] A. Khalili, J. Katz, W.A. Arbaugh, Toward secure key distribution in truly ad-hoc networks, in: Proceedings of SAINT Workshops’03, vol. 22, 2003, pp. 342–346.

[6] J. Pan, L. Cai, X. Shen, J.W. Mark, Identity-based secure collaboration in wireless ad hoc networks, Computer Networks 51 (3) (2007) 853–865.

[7] N. Saxena, G. Tsudik, J.H. Yi, Identity-based access control for ad hoc groups, in: Proceedings of ICISC'04, LNCS, vol. 3506, Springer-Verlag, 2005, pp. 362–379.

[8] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly-secure key distribution for dynamic conferences, in: Proceedings of Crypto'92, LNCS, vol. 740, Springer-Verlag, 1993, pp. 471–486.

[9] J. Anzai, N. Matsuzaki, T. Matsumoto, A quick group key distribution scheme with entity revocation, in: Proceedings of Asiacrypt'99, LNCS, vol. 1716, Springer-Verlag, 1999, pp. 333–347.

[10] V. Daza, J. Herranz, G. Sáez, Constructing general dynamic group key distribution schemes with decentralized user join, in: Proceedings of ACISP'03, LNCS, vol. 2727, Springer-Verlag, 2003, pp. 464–475.

[11] N. Saxena, G. Tsudik, J.H. Yi, Efficient node admission for short-lived mobile ad hoc networks, in: Proceedings of ICNP'05, 2005, pp. 269–278.

[12] G. Di Crescenzo, G.R. Arce, R. Ge, Threshold cryptography for mobile ad hoc networks, in: Proceedings of SCN'04, LNCS, vol. 3352, Springer-Verlag, 2005, pp. 91–104.

[13] G. Di Crescenzo, R. Ge, G.R. Arce, Improved topology assumptions for threshold cryptography in mobile ad hoc networks, in: Proceedings of SASN'05, 2005, pp. 53–62.

[14] R. Di Pietro, L. Mancini, G. Zanin, Efficient and adaptive threshold signatures for ad hoc networks, Electronic Notes on Theoretical Computer Science 171 (1) (2007) 93–105.

[15] S. Capkun, L. Buttyán, J.-P. Hubaux, Self-organized public-key management for mobile ad hoc networks, IEEE Transactions on Mobile Computing 2 (1) (2003) 52–64.

[16] A. Shamir, How to share a secret, Communications of the ACM 22 (1979) 612–613.

[17] G.R. Blakley, Safeguarding cryptographic keys, in: Proceedings of the National Computer Conference, American Federation of Information, Processing Societies Proceedings, vol. 48, 1979, pp. 313–317.

[18] A. Shamir, Identity-based cryptosystems and signature schemes, in: Proceedings of Crypto'84, LNCS, vol. 196, Springer-Verlag, 1984, pp. 47–53.

[19] D. Boneh, M.K. Franklin, Identity-based encryption from the Weil pairing, SIAM Journal on Computing 32 (3) (2003) 586–615.

[20] J. Baek, Y. Zheng, Identity-based threshold decryption, in: Proceedings of PKC'04, LNCS, vol. 2947, Springer-Verlag, 2004, pp. 262–276.

[21] J. Baek, Y. Zheng, Identity-based threshold signature scheme from the bilinear pairings, in: Proceedings of ITCC'04, vol. 1, 2004, pp. 124–128.

[22] D. Balfanz, D. Smetters, P. Stewart, H. Wong, Talking to strangers: authentication in ad hoc wireless networks, in: Proceedings of NDSS'02, The Internet Society, 2002.

[23] S. Capkun, J.-P. Hubaux, L. Buttyán, Mobility helps security in ad hoc networks, in: Proceedings of MobiHoc'03, 2003, pp. 46–56.

[24] S. Vaudenay, Secure communications over insecure channels based on short authenticated strings, in: Proceedings of Crypto'05, LNCS, vol. 3621, Springer-Verlag, 2005, pp. 309–326.

[25] M. Cagalj, S. Capkun, J.-P. Hubaux, Key agreement in peer-to- peer wireless networks, Proceedings of the IEEE, Special Issue in Security and Cryptography 94 (2) (2006) 467–478.

[26] V. Bhargava, M.L. Sichitiu, Physical authentication through localization in wireless local area networks, in: Proceedings of IEEE Global Telecommunications Conference, Globecom 2005, vol. 5, 2005, pp. 2658–2662.

[27] P. Feldman, A practical scheme for non-interactive verifiable secret sharing, in: Proceedings of FOCS'87, 1987, pp. 427–437.

[28] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in: Proceedings of Crypto'91, LNCS, vol. 576, Springer-Verlag, 1991, pp. 129–140.

[29] A. Boldyreva, Threshold signatures multisignatures and blind signatures based on the Gap-Diffie-Hellman-Group signature scheme, in: Proceedings of PKC'03, LNCS, vol. 2567, Springer-Verlag, 2003, pp. 31–46.

[30] I. Damgård, N. Fazio, A. Nicolosi, Non-interactive zero-knowledge from homomorphic encryption, in: Proceedings of TCC'06, LNCS, vol. 3876, Springer-Verlag, 2006, pp. 41–59.

[31] A. Herzberg, S. Jarecki, H. Krawczyk, M. Yung, Proactive secret sharing or: how to cope with perpetual leakage, in: Proceedings of Crypto'95, LNCS, vol. 963, Springer-Verlag, 1995, pp. 339–352.

Vanesa Daza graduated in Mathematics in University of Barcelona (UB, Barcelona, Spain) in 1999 and received her Ph.D. in Mathematics in Technical University of Catalonia (UPC, Barcelona, Spain) in 2004. Afterwards, she joined a crypto-based security company as a Senior Researcher. Currently she holds a post-doc position in the CRISES research group in the Rovira i Virgili University (URV, Tarragona, Spain). Her research interests are mainly related with distributed cryptography and its applications to ad-hoc networks.

Javier Herranz graduated in Mathematics in 2000 and obtained his Ph.D. in Applied Mathematics in 2005, in the Technical University of Catalonia (UPC, Barcelona, Spain). After that, he spent 9 months in the École Polytechnique (France) and 9 months in the Centrum voor Wiskunde en Informatica (CWI, The Netherlands), as a post-doctoral researcher, granted with an ERCIM fellowship (May 2005–October 2006). Currently, he is a post-doctoral researcher in the institute IIIA (Bellaterra, Spain). His research interests are mostly related to cryptography, specially to digital signatures.

Paz Morillo is associate professor in Applied Mathematics in the Technical University of Catalonia (UPC, Barcelona, Spain). She graduated in Mathematics and received her Ph.D. in Computer Science. Her first research was in the field of Graph Theory and its application to the design of interconnection networks. In 1992, she founded the group of Mathematics Applied to Cryptography (MAK) at the Technical University of Catalonia. Her research interests are mainly related to the application of elliptic curves to cryptography, design of provable secure cryptosystems and distributed cryptography.

Carla Ràfols graduated in Mathematics in 2003 from the University of Barcelona. She is currently a Ph.D. student of Dr. Paz Morillo at the Mathematics Applied to Cryptography Group at the Technical University of Catalonia (UPC). Her research interests include hard-core predicates and provably secure cryptographic protocols.