A Smart Card Based Distributed Identity Management Infrastructure for Mobile Ad hoc Networks*

Eve Atallah[1,2] and Serge Chaumette[1]

[1] LaBRI, Université Bordeaux 1,
351 Cours de la Libération, 33405 Talence Cedex, France,
Serge.Chaumette@labri.fr,
WWW home page: http://www.labri.fr/perso/chaumett/
[2] XLIM, Université de Limoges,
123, avenue Albert Thomas, 87060 Limoges Cedex, France,
Eve.Atallah@labri.fr

Abstract. The network is becoming more and more versatile because of the variety of the computing resources and the communication technologies that have become available. The mobility of the nodes, in these so called Mobile Ad hoc Networks (MANets), furthermore leads to a situation where it is very difficult to establish secure community-based or even peer to peer communication channels. The basic and major problem that has to be solved is that of identity management: how to identify and authenticate an entity that is *a priori* unknown and that tries to dynamically join a community in the network? Even if we solve this problem, how to distribute these certified identities over the network? In this paper, we propose to make a clear distinction between two kinds of organization of a MANet. We consider an identity-based approach and a goal-based approach. In the identity-based approach the nodes of the network have to be precisely identified (i.e. with their real-world identity) and a central administration is therefore required. In the goal-based approach, identities are simply used to distinguish between the nodes that collaborate to a certain goal. We claim that when this second approach is considered, it is possible to support a totally distributed identity management system. Our contribution is the design and the implementation of such a system for these goal-based networks. We assume that the users who want to get involved are provided with PDAs supplied with smart cards and more precisely Java Cards, which are the basic secure bricks on which our approach relies. Of course, our approach supports the uniqueness of identities, but it furthermore enforces permanency, i.e. it prevents changing and repudiation of identity. In this paper, we describe the protocol that we have designed to support our solution and its effective implementation.

Keywords *MANets, identity management, smart cards, auto-administration*

* This work is partly supported by the ANR/SSIA project Cryscoe and by a PhD grant funded by Region Limousin.

1 Introduction

The need for identity management. Any group (either human or computer based) composed of different and independent entities requires a system to protect its assets and the realization of its founding goal (usually data sharing), by organizing rights and duties management. The success of such a system entirely depends upon the recognition of its components. It is the basis of the management of all security issues [1]. Once recognition is achieved it is for instance possible to create restricted groups with controlled access and to establish secure communication channels.

Specificity of the context. Our framework is that of MANets [2] (Mobile Ad hoc Networks) where, compared to classical networks, there cannot be any central node in charge of the administration and the organization of the network. In this context, the network dynamically evolves, based on the arrival and departure of (the devices of) the final users. Even though it is often claimed that almost nothing can be achieved in such a context, we believe that a lot can be done, provided the nature of such a network is properly taken into account. The basic idea is that there are two ways to consider the organization of a number of entities as a network:

- in **the identity based approach**, a number of nodes work altogether based on the precise knowledge of their respective real-world identities. This can be the case of a group of persons defining a proposal for some project. It might be absolutely necessary to know who is who because there may be confidential information that can only be communicated to specific partners. This is also the case of applications where things are organized around a central system, such as banking systems for instance, where the issuer of an operation must be precisely identified.
- in **the goal based approach**, a number of nodes work altogether so as to achieve a common goal. The question is not to know who is involved, but to make sure that the nodes that are involved contribute to the same precise goal. For instance, in collaborative writing like Wikipedia or in Seti like applications (even though it is in some sens centralized) there is no reason for knowing who is who in the real life. The thing that is important is the goal.[3]

Our approach relies on this subtle difference. We claim that in spontaneous mobile ad hoc networks the first approach is not feasible. Devices need to collaborate independently of any predefined realworld organization. Groups are dynamically built for a specific goal, and the nodes are given identities on the fly, the purpose of which is simply to distinguish them within the group. We can also note that a

[3] This is different from role based approaches [3, 4] where users still need to be identified based on their real world identities and where they are given roles by some sort of central administration. The choice of roles comes from the top of the organization, whereas the fact of participating to a goal comes from the bottom.

device might be willing to participate in several activities and thus join different groups, having a different identity in each group.

In all existing solutions, it is mandatory for all the users to be part of a human organization that the network simply reflects. The network entities are dependent of a central administration that must remain available at all time. Our contribution removes these constraints. We provide an auto-administrated architecture that enables the dynamic allocation of identities to the nodes of a MANet; it can then serve as a basis to develop higher level security mechanisms (which are out of the scope of this paper). Our system requires neither centralization of identities nor in-line administration.

2 The phases of identity management, existing solutions and MANet specific problems

In classical networks (as opposed to MANets), a trusted entity (for instance a dedicated national agency or a network administrator) validates and centralizes the identities of all the users. It delivers ID cards or certificates (X509 [5] for instance). In a MANet, there cannot be any centralization, because of the volatility of the nodes that compose the network and of the network itself. Thus, several specific difficulties appear during the different phases of identity management. In this section, we discuss these difficulties. We present a number of existing approaches that deal with the different phases (validation, certification and distribution) of identity management and explain why they do not solve the problems that we want to address. We deduce some requirements that our solution will need to cope with.

2.1 Nature of the identities and their validation

At the user level, an identity must have a public part which is called the identifier (for instance a login name, a pseudo plus a public key, etc.) and an authenticator that makes the link between the entity and its identifier (for instance a password, a private key, etc.).

Uniqueness. At least the authenticator has to be unique (and it must be kept secret to prevent impersonation). If a central authority generates the keys, the uniqueness is straightforward to achieve. The problem is more difficult to deal with in a MANet, where such a central authority does not exist. It is impossible to compare an identity with all the other identities because there is no global knowledge of the network, i.e. no entity that knows all the identities. A distributed algorithm would not work either, because a node could leave during the verification, or the network could be separated in several disconnected parts. It is thus impossible to validate the uniqueness of an identity in a MANet, unless this uniqueness is guaranteed by the nature of the identity creation process, i.e. by the identity creation algorithm. These considerations lead to the following requirement that we want to support in our solution.

Definition of 1st requirement An identity must be unique and thus a part of it has to be unique by nature of the creation process. This unique part must be kept secret.

Permanency. Once established, the identity of a device should be definitively linked to this device. When asked to confirm its identity, the device should not be able to deny it. This is the permanency condition that we want to foster.

Definition of 2nd requirement An identity can neither be modified nor dismissed. The owner of an identity must confirm it when asked for.

A solution to meet this requirement (even though partial since it does not oblige the user to confirm its identity when asked for) is proposed in [6]. Each device is given a pair of asymmetric keys created during its production phase and guaranteed to be unique. Its identifier is the public key of this pair of keys, and its validity is certified by a reliable authority that signs it. The key pair as a whole is kept secret by being stored in a secure module such as a smart card. Identities cannot be changed by the user (the CA would be required) and this is part of the permanency requirement defined above. This solution does not make it possible for a device to have several identities, it does not support non repudiation, and it depends on an administration infrastructure. It thus does not meet our goals.

2.2 Certification and distribution

Once the identities have been generated and validated, the next step consists in distributing them along with the the proof that they are valid, this assembled information being usually referred to as a certified version of the identity. Several methods exist to achieve these operations. In the rest of this section, we describe two of the major approaches that could at first glance be considered as potential solutions to solve the problem in a MANet.

Auto-organization: to mitigate the problem of the availability of nodes for the distribution of certificates, solutions such as those described in [1, 7, 8] rely on the use of a trusted node chosen using a method to establish confidence [9]. Establishing confidence requires to observe a number of nodes over a period of time and thus to recognize them. This kind of approach can therefore only be used once identities have been validated, which is precisely what we are trying to do. It thus cannot help in our context.

Signature by a Certification Authority (CA): in classical networks, the authority is centralized on one or several nodes that share the same pair of primary keys. For large networks this approach is extended based on a hierarchical organization. As MANets cannot rely on the continuing existence of any specific node, solutions to distribute this otherwise central authority were developed that use the *threshold secret sharing* principle. The key of the CA is shared by a set of n nodes and k < n partial signatures are required to reconstruct a complete signed certificate. These methods allow to admit a new node in the network as the result of the collective decision of at least k nodes. There are several partial signatures protocols that use RSA [10–13] or DSA [14, 15] keys. The distribution of the authority can be partial [16, 17] or total [18, 19], in which case every node is supplied with a partial key.

The advantages of these solutions are as follows. First, to compromise the system several nodes must be attacked (get the CA secret key, DOS, etc.) instead of only one. Second, it increases the global availability of the certification authority (by choosing n large enough compared to k). If the distribution is partial, some devices are in charge of a more important task and thus have a specific non symmetric role, what we do not want. We furthermore have the problem of locating these nodes that share the authority. If the distribution is total there is equality between the nodes and no more localization question. In any case, an initialization step is necessary that requires the presence of an administrator and a certain number of nodes for the initial distribution of keys. The administration phase is then strongly dependent of the network specific usage and must be initiated before the network is constituted, what clearly removes the spontaneity, which is a feature that we want to support. Therefore our third requirement.

Definition of 3rd requirement The validation, certification and distribution phases cannot rely on a central administration that would reflect an ***a priori*** restricted human (i.e. real-world) organization.

3 Our solution

Based on the requirements defined above we can establish a number of features to achieve at the implementation level.

1st requirement An identity must be unique and thus a part of it has to be unique by nature of the creation process. This unique part must be kept secret.

To implement this constraint, we need a process to generate unique identities and the possibility to store them in a secure, read only area.

2nd requirement An identity can neither be modified nor dismissed. The owner of an identity must confirm it when asked for.

This requires the capability to store data in a non erasable, read only area. The confirmation of an identity must be out of the control of its owner, so that he cannot deny it or wrongly confirm an identity that he does not own.

3rd requirement The validation, certification and distribution phases cannot rely on a central administration that would reflect an ***a priori*** restricted human (i.e. real-world) organization.

As a consequence of this 3rd requirement we have to make virtual the notion of CA.

Presentation of Java Cards. Our solution satisfies these requirements by using smart cards and more precisely Java Cards[4] which provide some specific features, among which:

- their ROM memory makes it possible to install applications (in factory) that can thereafter neither be modified nor erased. Note that the ROM of the cards cannot store application data.
- they allow to store persistent data.
- the information they store can be protected by a firewall that sits between applications.

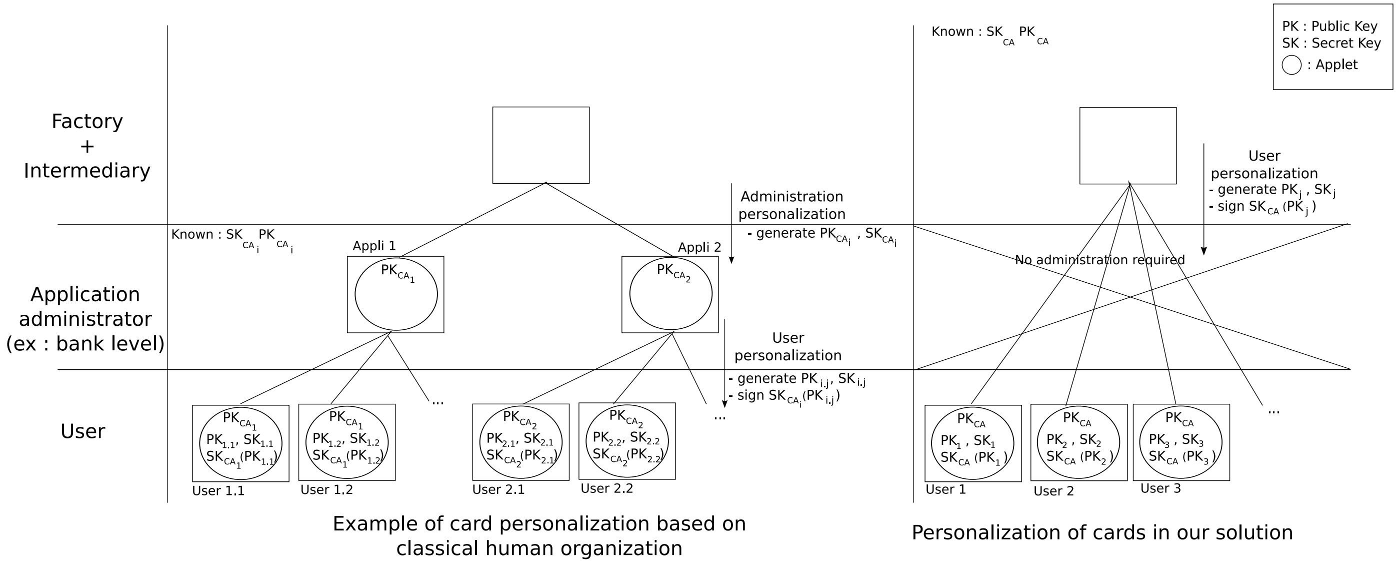
We strongly rely on the fact that smart cards are secured devices and we do not consider physical attacks like fault injection.

We have specifically chosen Java Cards rather than any other brand of card because they are easy to program and we have been using this technology for quite a long time in our team.

In the rest of this section we give an overview of our solution and how it is used. The lower level protocol is described in section 4.

Implementation of our solution. Each card is prepared as follows:
1) An applet (Java Card application) is installed in factory on each card. It provides the methods required to define identities and to ensure their definitive registration. The use of data stored inside a card is completely controlled by this card and is thus limited by nature to the operations that we have defined. It is then possible to register an identity permanently without any risk that it is modified or erased.

[4] Java and all Java-based marks are trademarks or registered trademarks of Sun microsystems, Inc. in the United States and other countries. The authors are independent of Sun microsystems, Inc. All other marks are the property of their respective owners.



PK : Public Key
SK : Secret Key
○ : Applet
Factory
+
Intermediary
Application
administrator
(ex : bank level)
User
Known : SK_{CA_i} PK_{CA_i}
Appli 1
Appli 2
PK_{CA_1}
PK_{CA_2}
Administration
personalization
- generate PK_{CA_i} , SK_{CA_i}
User
personalization
- generate $PK_{i,j}$, $SK_{i,j}$
- sign $SK_{CA_i}(PK_{i,j})$
PK_{CA_1} $PK_{1.1}$, $SK_{1.1}$ $SK_{CA_1}(PK_{1.1})$
PK_{CA_1} $PK_{1.2}$, $SK_{1.2}$ $SK_{CA_1}(PK_{1.2})$
PK_{CA_2} $PK_{2.1}$, $SK_{2.1}$ $SK_{CA_2}(PK_{2.1})$
PK_{CA_2} $PK_{2.2}$, $SK_{2.2}$ $SK_{CA_2}(PK_{2.2})$
...
User 1.1
User 1.2
User 2.1
User 2.2
Known : SK_{CA} PK_{CA}
User
personalization
- generate PK_j , SK_j
- sign SK_{CA} (PK_j)
No administration required
PK_{CA} PK_1 , SK_1 SK_{CA} (PK_1)
PK_{CA} PK_2 , SK_2 SK_{CA} (PK_2)
PK_{CA} PK_3 , SK_3 SK_{CA} (PK_3)
User 1
User 2
User 3

Example of card personalization based on classical human organization

Personalization of cards in our solution

Fig. 1. Compared personalization stages

2) Each card stores (installed in factory) a global public key, a specific and unique asymmetric key pair and the public key of this pair signed by the related global private key. These data can thereafter never be accessed from outside the card. Even if smart cards are considered really safe, we still want to ensure that if a card were after all compromised, security of all the identity management architecture would not fall. Therefore we do not store a secret symmetric key or a private asymmetric key[5].

This secure platform is then used as follows:

1) To communicate its identity to another card when asked for, a card encrypts it with its own private key and supplements it with its signed (by the global private key) public key. This is enough to prove that this identity has been originally provided from inside a card and that it obeys the rules that we have defined (uniqueness, permanence) [6].

2) This could then be used to exchange a session key to achieve secure communication. This would lead to good forward secrecy and fast throughput between smartcards[7].

Our solution vs other smart card infrastructures. Even though smart cards are already widely used for identification and authentication between a user and a central system, there are many differences with the context of our approach. The global key pair, the public key of which is installed on all the cards, does not depend on a specific application that the card could be used for, and thus the personalization of the card for a specific application does not take place at the same level as in classical solutions. This is a consequence of our discussion on identity vs. goal based approaches in section 1 page 2, and is described figure 1.

[5] If we had ignored this risk, we could have chosen to include the global private key in each card and then to generate the card key pair and to sign its public key inside the card.

[6] It should be noted that if a card becomes compromised and the global public key is discovered, the attacker can get to know the identities of the entities that participate in the network. This has no impact over the security or the proprieties of our infrastructure. Furthermore, the fact that cards are considered today as the most trusted available devices makes it possible to ignore this hypothetical risk.

[7] This precision was suggested by one of the reviewers of this paper.

Advantages of our solution (See figure 2). We have established an auto-administration system that implements the identity management requirements that we have defined. It supports a high level of security, leaves the user total freedom, and requires no preparation once the cards are out of the factory. No user or administrator of the network has to care about key management. The final user only needs to provide an identifier. Furthermore, even once supplied with an identity, a card is not dedicated to one single application as it is the case for instance with banking cards that are specific to one single bank. It can be supplied with new identities and join other goal-based networks on the fly.

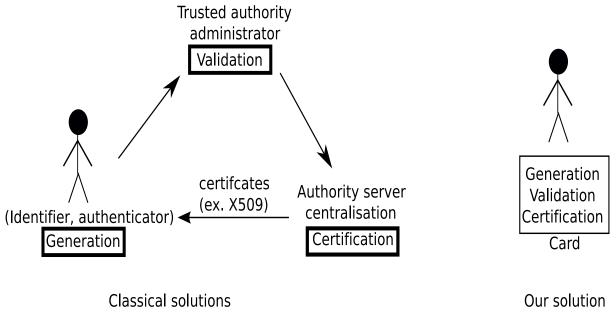


Fig. 2. A comparison of solutions for identity management in MANets

4 Our protocol

In our protocol, all communications between cards obey a number of rules:

1. All messages are encrypted by the private key of the sending card.
2. So that a message can be deciphered, it contains the public key of the sending card signed by the private global key (that signed key was stored in every card in factory).
3. A message also contains the following information:
   - the nature of the request (for instance «ACert» for a certificate request).
   - a nounce used to avoid replay. This for instance prevents repeated storage operations that would saturate the memory of cards.

Additional data can then be added according to the type of the message.

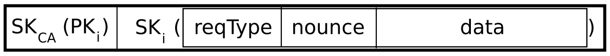


Fig. 3. Structure of a message

Creation of an identity (see Fig.4). The user first provides the card with an identifier (1). The card then verifies if an identifier has already been defined (2) and if not, generates a RSA key pair (3), associates it with the received identifier (4) and definitively loads the association in the card (5).
This complies with our 1st requirement, since thanks to the RSA key that it contains, the identity is unique by creation. The identity is protected by the card and its private part is kept secret inside the card. Permanence which is our 2nd requirement is also supported because we provide no method to modify or erase an identity[8]. This approach also complies with the 3rd requirement, since the validation of identities does not require any *a priori* restricted central administration.

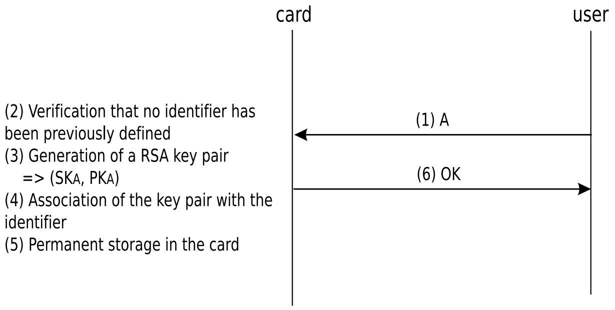

card
user
(2) Verification that no identifier has been previously defined
(3) Generation of a RSA key pair
=> (SKA, PKA)
(4) Association of the key pair with the identifier
(5) Permanent storage in the card
(1) A
(6) OK

Fig. 4. Creation of an identity

Certification and distribution (see Fig.5). First, the user sends a request to the card asking it to discover his neighbours (1). The card prepares the request (2) that only contains the request type and the nounce. It is sent (ciphered) to the neighbourhood (3)(4)(5). A card that receives this request deciphers it (6) and prepares an answer by adding its public identity (public key + identifier) to the message (7). The response is sent (ciphered) (8)(9) and the initial card receives all these incoming messages (10). It deciphers them (11) and temporarily stores all the received identities (12). The identifiers are then propagated to the user level (13).

[8] Note that swapping a smart card for a new one to get a new identity would have no consequence over the global infrastructure. This precision has been suggseted by a remark of one of the reviewers of this paper.

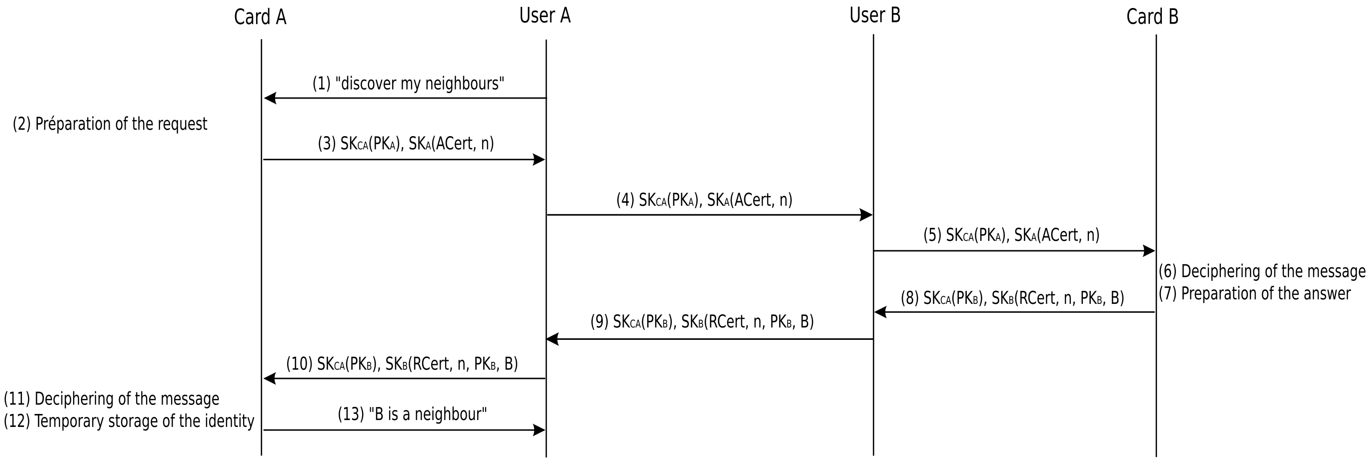

Card A
User A
User B
Card B
(1) "discover my neighbours"
(2) Préparation of the request
(3) SKCA(PKA), SKA(ACert, n)
(4) SKCA(PKA), SKA(ACert, n)
(5) SKCA(PKA), SKA(ACert, n)
(6) Deciphering of the message
(7) Preparation of the answer
(8) SKCA(PKB), SKB(RCert, n, PKB, B)
(9) SKCA(PKB), SKB(RCert, n, PKB, B)
(10) SKCA(PKB), SKB(RCert, n, PKB, B)
(11) Deciphering of the message
(12) Temporary storage of the identity
(13) "B is a neighbour"

Fig. 5. Certification and distribution of identities

In terms of robustness and security, the situation is as follows: if the communication breaks at stage (4), there are no consequences (the remote card will simply remain undiscovered). If communication breaks at stage (9), there is no real consequence either since nobody has stored anything yet and it thus cannot be an attack to saturate any of the cards. Once again, the remote card will simply not be discovered.

Once all these steps have been achieved, the deployed validated and certified identities can be used to enable secure communications.

5 Conclusion

In this paper we have presented an identity management system that we have designed for entities willing to collaborate in a goal based approach over a MANet. A prototype has been implemented on a number of Dell Axim PDAs and a draft video demo can be seen on the web at [20].

This identity management architecture sets a basis to establish higher level security features. One of its main characteristics and advantages is that it does not impose any constraint on the natural spontaneity of such dynamic networks. Thanks to the use of Java Cards, the creation and storage of (certified) identities make it possible to support the basic security requirements that we have defined (uniqueness and permanency), without any central administration or server. The fact that all the administration takes place inside the card makes the nodes of the network completely independent of any preexisting real-world group or organization. Every user (or node) thus has the possibility to create a group without any human intervention, wherever he wants, whenever he wants. This is one of the outcomes of the clear distinction that we have made between identity based networks and goal based networks.

Future work directions concern the way the goal of a group is defined and the way a node willing to join a group is allowed to enter it. Basically, the goal will be described by means of a *charter* [21] that contains a number of questions the node will have to answer. This work on charters is part of the MADNESS project carried out at XLIM, University of Limoges. Once this will be achieved, we will be able to conduct an evaluation of the global system.

References

1. Hubaux, J., Buttyan, L., Capkun, S.: Self-organized public-key management for mobile ad hoc networks. In: Proceedings of the ACM International Workshop on Wireless Security. Volume 2., IEEE Transactions on Mobile Computing (january 2003) 52–64
2. Chlamtac, I., Conti, M., Liu, J.: Mobile ad hoc networking: Imperatives and challenges. Elsevier Ad Hoc Networks Journal **1** (july 2003) 13–64
3. Ferraiolo, D., Cugini, J., Kuhn, D.: Role based access control: Features and motivations. In: Proceedings of the 11th Annual Conference on Computer Security Applications, IEEE Computer Society Press, Los Alamitos, CA (1995) 241–248

4. Ferraiolo, D., Gavrila, S., Kuhn, D., Chandramouli, R.: Proposed NIST standard for role-based access control. Information and System Security **4** (august 2001) 224–272
5. IETF: ITU-t recommendation X.509 (revised) - information technology - open systems interconnection - the directory: Public-key and attribute certificate frameworks (2000) ISO/IEC 9594-8.
6. Kargl, F., Schlott, S., Weber, M.: Identification in ad hoc networks. In: Proceedings of the 39th Annual Hawaiian International Conference on System Sciences. Volume 9., IEEE Computer Society, Washington, DC (January 2006)
7. Capkun, S., Hubaux, J., Buttyan, L.: Mobility helps security in ad hoc networks. In: Proceeding of the 4th ACM international Symposium on Mobile Ad Hoc Networking and Computing MobiHoc'03, Annapolis, Maryland, USA, ACM Press, New York, NY (June 2003) 46–56
8. Garfinkel, S.: PGP : Pretty Good Privacy. O'Reilly & Associates, Sebastopol, California (1995)
9. Marias, G., Papapanagiotou, K., Tsetsos, V., Sekkas, O., Georgiadis, P.: Integrating a trust framework with a distributed certificate validation scheme for MANETs. EURASIP Journal on Wireless Communications and Networking (2006)
10. Y. Frankel, P. Gemmel, P.M., Yung, M.: Proactive RSA. In: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology (Crypto'97). Volume 1294., Lecture Notes In Computer Science (August 1997) 440–454
11. Y. Frankel, P.M., Yung, M.: Adaptive security for the additive-sharing based proactive RSA. In: Proceedings of the International Workshop on Practice and Theory in Public Key Cryptography. Volume 1992., Lecture Notes In Computer Science (February 2001) 240–263
12. Rabin, T.: A simplified approach to threshold and proactive RSA. In: Proceedings of the 18th Annual international Cryptology Conference on Advances in Cryptology (Crypto'98). Volume 1462., Lecture Notes In Computer Science (august 1998) 89–104
13. Shoup, V.: Practical threshold signatures. In: Proceedings of EUROCRYPT'00. Volume 1807., Lecture Notes in Computer Science (2000) 207–220
14. M. Narasimba, G.T., Yi, J.: On the utility of distributed cryptography in P2P and MANets : the case of membership control. In: Proceedings of the 11th IEEE International Conference on Network Protocol (ICNP), IEEE Computer Society, Washington, DC (november 2003) 336–345
15. N. Saxena, G.T., Yi, J.: Admission control in peer-to-peer : Design and performance evaluation. In: Proceedings of the 1st ACM Workshop on Security of Ad hoc and Sensor Network (SASN'03), Fairfax, Virginia, ACM Press, New York, NY (october 2003) 104–113
16. L. Zhou, Z.H.: Securing ad hoc networks. IEEE Network **13** (november/december 1999) 24–30
17. Yi, S., Kravets, R.: MOCA : Mobile certificate authority for wireless ad hoc networks. In: Proceedings of the 2nd Annual PKI Research Workshop (PKI'03). (2003)
18. Luo, H., Lu, S.: Ubiquitous and robust authentication services for ad hoc wireless networks. Technical report, Dept. of Computer Science, UCLA (October 2000)
19. Kong, J., Zerfos, P., Luo, H., Lu, S., Zhang, L.: Providing robust and ubiquitous security support for mobile ad-hoc networks. In: Proceedings of the 9th International Conference on Network Protocols (ICNP'01), IEEE Computer Society, Washington, DC (2001) 251–260

20. Atallah, E., Chaumette, S.: http://www.labri.fr/perso/chaumett/pda_0001.wmv.
21. Atallah, E., Bonnefoi, P.F., Burgod, C., Sauveron, D.: Mobile ad hoc network with embedded secure system. In: Proceedings of the seventh edition of e-Smart conference, Nice, France (September 2006)